# Transient Dynamic Finite Element Modelling of Flexible Rotor Systems 

## with Nonlinear Fluid Film Bearings and Faults

by
Armand Krüger

Submitted in fulfilment of part of the requirements for the degree of Master of Engineering in the Faculty of Engineering, Built Environment and Information Technology

University of Pretoria: Department of Mechanical and Aeronautical Engineering
Supervisor: Prof. P.S. Heyns
Co-Supervisor: Dr. A.J. Oberholster

Title: Transient dynamic finite element modelling of flexible rotor systems with nonlinear fluid film bearings and faults
Author: A. Krüger
Supervisor:
Co-supervisor
Prof. P.S. Heyns
Dr. A.J. Oberholster
Year: 2014
This dissertation forms part of a research project assigned to the University of Pretoria by Eskom (the primary electricity utility in South Africa). The project aims to address, amongst others, the limitations imposed by shaft runout on the usable frequency range of diagnostic data measured by eddy current proximity probes on turbogenerator shafts. This research includes an experimental investigation into the effects of artificially induced faults on a laboratory-scale rotor system, the development and analysis of a mathematical (numerical) model of this rotor system and the development of data processing techniques (including artificial intelligence) to determine the rotor's condition, faults and diagnostic signal parameters from both the experimental and numerical results. Furthermore, a methodology is to be developed to perform runout compensation in an unsupervised manner. These techniques are then to be implemented for proximity probe vibration data measured on turbogenerators.

As part of the research project, this dissertation specifically focuses on the development and rotor dynamic analysis of numerical (finite element) models of the experimental (laboratory-scale) rotor system (using finite element software MSC.Nastran), including gyroscopic effects, a nonlinear force model for the hydrodynamic journal bearing of the rotor system (capable of capturing oil whirl and oil whip instabilities) as well as simulated faults (such as unbalance and rotor-stator rubbing). Since MSC.Nastran does not have a built-in nonlinear hydrodynamic journal bearing model, a custom model of such a bearing was developed and incorporated into the finite element solver, further expanding its already powerful rotor dynamic modelling capabilities. Rotor dynamic analyses performed include the calculation of critical speeds (synchronous complex modes analysis), Campbell diagrams (asynchronous complex modes analysis), steady state frequency response due to unbalance (synchronous frequency response analysis) and nonlinear transient response during rotor run-up. Amongst others, this dissertation explores the seemingly largely unexplored/undocumented capability of finite element software MSC.Nastran to perform rotor dynamic analyses using rotor models constructed with three-dimensional elements. Software (MATLAB code) was also developed to perform post-processing of the simulation results as well as signal processing for investigating the spectral content of transient results.

The support structure of the laboratory-scale rotor system was experimentally characterised and an experimental modal analysis was performed on the rotor (excluding its support structure) and its results used to update the finite element rotor models. The transient dynamic response of the experimental rotor system during run-up due to unbalance and rubbing was also analysed in order to validate the developed numerical rotor system models. The numerical results are found to be in good agreement with the experimental results.

## Acknowledgements

I would like to convey my gratitude to the following people for their contributions and support in completing this work:

- Prof. Stephan Heyns for his invaluable guidance as supervisor and whose immense enthusiasm and expertise in the field of mechanical vibrations inspired me to pursue this field of study.
- Dr. Abrie Oberholster for his guidance, advice and assistance as co-supervisor, especially with regards to some of my experimental work.
- Mr. George Breitenbach who made the Sasol laboratory and its equipment available to me and assisted in setting up and conducting experiments in the laboratory. His recommendations, sound advice and creative workmanship skills brought my experimental setup ideas to life.
- Mr. Herman Booysen for his assistance with and troubleshooting of electronic equipment. Also, Mr. Johann Clarke for assisting with and arranging the use of an eDAQ data acquisition system.
- Mr. Gerrit Visser who provided technical support for the finite element software, specifically for the utilization of subroutines as a means of including custom bearing models into finite element models.
- My parents who have always encouraged me to give my best in everything I do and to develop my God-given talents to their full potential. Also, my friends for their personal support and understanding.
- My wife Leonie for her personal support and encouragement and especially for her patience and tolerance of my frequent unavailability during this time.

TABLE OF CONTENTS
Abstract ..... i
Acknowledgements ..... ii
List of Figures ..... ix
List of Tables ..... xvi
List of Symbols ..... xviii
List of Abbreviations/Acronyms ..... xxiii
1 Introduction ..... 1
1.1 Background ..... 1
1.2 Literature Study ..... 2
1.2.1 Rotor Dynamics and Faults ..... 2
1.2.1.1 Mass Imbalance ..... 2
1.2.1.2 Whirl and Critical Speeds ..... 3
1.2.1.3 Gyroscopic Effects ..... 4
1.2.1.4 Fluid Film Bearings, Oil Whirl and Oil Whip ..... 6
1.2.1.5 Rubbing and Dry Whip ..... 7
1.2.1.6 Misalignment and Bearing Anisotropy ..... 8
1.2.1.7 Damping ..... 8
1.2.1.8 Other Rotor Faults, Nonlinear Excitations and Rotor Dynamics ..... 8
1.2.2 Modelling Fluid Film Bearings (Fluid Film Lubrication Theory) ..... 9
1.2.2.1 The Reynolds Equation ..... 9
1.2.2.2 Boundary Conditions ..... 11
1.2.2.3 Numerically Solving the Reynolds Equation ..... 12
1.2.3 Modelling Rotor Systems (Background) ..... 14
1.2.4 Finite Element Modelling of Rotor Systems with Fluid Film Bearings ..... 17
1.2.5 Rotor Dynamics in MSC.Patran and MSC.Nastran ..... 19
1.2.5.1 Rotor Dynamic Modelling Capabilities ..... 19
1.2.5.2 Rotor Dynamics Module Limitations ..... 20
1.2.5.3 Input Files, Solution Sequences, Superelements and Rotor Dynamics Entries ..... 20
1.2.6 Model Updating with MSC.Nastran Design Sensitivity and Optimization ..... 23
1.2.7 User Defined Services in MSC.Nastran ..... 23
1.2.7.1 Build Environment ..... 24
1.2.7.2 Building the Service ..... 24
1.2.7.3 Using the Service ..... 25
1.3 Scope of Work ..... 26
1.4 Overview of Chapters ..... 27
2 Experimental Rotor System Setup and Characterisation ..... 29
2.1 Experimental Rotor System ..... 29
2.1.1 Bently Nevada Rotor Kit ..... 29
2.1.2 Rotor System Dimensions and Coordinate Systems ..... 30
2.2 Rotor Characterisation ..... 31
2.2.1 Mass Densities ..... 31
2.2.2 Experimental Modal Analysis ..... 32
2.2.2.1 Experimental Setup ..... 32
2.2.2.2 Analysis ..... 33
2.2.2.3 Results ..... 35
2.2.2.4 Elastic Moduli ..... 37
2.3 Support Structure Characterisation ..... 37
2.3.1 Bush Bearing Characterisation. ..... 37
2.3.1.1 Experimental Setup ..... 37
2.3.1.2 Analysis and Results ..... 39
2.3.1.3 Stiffness and Damping ..... 44
2.3.2 Coupling Lateral Characterisation. ..... 46
2.3.2.1 Experimental Setup ..... 46
2.3.2.2 Analysis and Results ..... 48
2.3.2.3 Stiffness and Damping ..... 50
2.3.3 Coupling Torsional Characterisation ..... 51
2.3.3.1 Experimental Setup ..... 51
2.3.3.2 Analysis and Results ..... 53
2.3.3.3 Stiffness and Damping ..... 54
2.3.4 Journal Bearing Characteristics ..... 56
2.3.4.1 Geometrical Parameters ..... 56
2.3.4.2 Lubrication ..... 56
3 Numerical Rotor System Models ..... 57
3.1 Finite Element Rotor Models ..... 57
3.1.1 1D Rotor Model ..... 58
3.1.1.1 Normal Modes. ..... 62
3.1.1.2 Model Updating ..... 64
3.1.1.3 Complex Modes ..... 66
3.1.2 3D Rotor Model ..... 66
3.1.2.1 Normal Modes ..... 69
3.1.2.2 Model Updating ..... 71
3.1.2.3 Complex Modes. ..... 73
3.2 FE Rotor System Models with Linear Bearings ..... 74
3.2.1 $1 D$ Rotor System Model with Linear Bearings ..... 74
3.2.1.1 Normal and Complex Modes ..... 76
3.2.1.2 Critical Speeds ..... 77
3.2.1.3 Campbell Diagram ..... 79
3.2.1.4 Frequency Response ..... 80
3.2.1.5 Transient Response ..... 84
3.2.1.6 Gravity Loading and Initial Conditions ..... 93
3.2.1.7 Flat Surface Rubbing ..... 99
3.2.1.8 Radial Rubbing ..... 104
3.2.1.9 Bearing Anisotropy ..... 111
3.2.2 $3 D$ Rotor System Model with Linear Bearings ..... 111
3.2.2.1 Normal and Complex Modes ..... 114
3.2.2.2 Critical Speeds ..... 115
3.2.2.3 Campbell Diagram ..... 118
3.2.2.4 Frequency Response ..... 119
3.2.2.5 Transient Response ..... 122
3.2.2.6 Gravity Loading and Initial Conditions ..... 125
3.2.2.7 Flat Surface Rubbing ..... 128
3.2.2.8 Radial Rubbing ..... 131
3.3 Journal Bearing Model ..... 136
3.3.1 Boundary Value Problem ..... 136
3.3.2 Finite Difference Method ..... 137
3.3.3 Bearing Forces ..... 140
3.3.4 Journal Bearing Model in MATLAB ..... 141
3.3.5 Other Journal Bearing Models ..... 147
3.4 Incorporation of the Journal Bearing Model into the Rotor System Models ..... 149
3.4.1 Build Environment ..... 149
3.4. $\quad$ Journal Bearing Model in FORTRAN ..... 150
3.4.3 Modifications to the FE Rotor System Models ..... 151
3.4.3.1 1D Rotor System Model Modifications ..... 152
3.4.3.2 3D Rotor System Model Modifications ..... 153
3.5 FE Rotor System Models with Nonlinear Journal Bearings ..... 155
3.5.1 1D Rotor System Model with a Nonlinear Journal Bearing. ..... 155
3.5.1.1 Transient Response without Gravity Loading ..... 155
3.5.1.2 Transient Response with Gravity Loading and Initial Conditions ..... 159
3.5.2 3D Rotor System Model with a Nonlinear Journal Bearing. ..... 165
3.5.2.1 Transient Response without Gravity Loading ..... 165
3.5.2.2 Transient Response with Gravity Loading and Initial Conditions ..... 167
4 Rotor System Experimental Analysis ..... 173
4.1 Rotor Balancing ..... 173
4.2 Runout Compensation ..... 175
4.2.1 Method 1: DC Component Unimportant ..... 175
4.2.2 Method 2: DC Component Important ..... 176
4.3 Transient Unbalance Response Analysis ..... 178
4.3.1 Experimental Setup ..... 178
4.3.2 Runout Measurement ..... 181
4.3.3 Vibration Measurement ..... 185
4.3.3.1 Unbalance Load Case $1: 1 \mathrm{~g}$ weight at $90^{\circ}$ applied to disk 2 ..... 186
4.3.3.2 Unbalance Load Case 1: 2 g weight at $90^{\circ}$ applied to disk 2 ..... 192
4.4 Transient Flat Surface Rubbing Analysis ..... 196
4.4.1 Experimental Setup ..... 196
4.4.2 Runout Measurement ..... 196
4.4.3 Vibration Measurement ..... 197
5 Experimental vs. Numerical Results ..... 200
5.1 Transient Unbalance Response ..... 200
5.2 Transient Flat Surface Rubbing ..... 206
6 Conclusions ..... 207
References ..... 210
Appendix A : Rotor System Drawings, Diagrams and Images ..... 215
A. 1 Experimental Rotor System ..... 215
A. 2 1D Rotor System Model Schematic Diagrams ..... 217
A. 3 3D Rotor System Model Schematic Diagrams ..... 219
A. 4 3D Rotor Model in MSC.Patran ..... 221
Appendix B : MATLAB Programs - Experimental Analyses ..... 224
B. 1 Bush Bearing Characterisation ..... 224
B. 2 Coupling Lateral Characterisation ..... 231
B. 3 Coupling Torsional Characterisation ..... 235
B. 4 Rotor System Experimental Analysis ..... 240
B. 5 Generic Programs/Functions ..... 260
Appendix C : MATLAB Programs - Numerical Modelling ..... 268
C. 1 MSC.Nastran Results Processing ..... 268
C. 2 Rigid Rotor Model ..... 314
C. 3 Journal Bearing Models ..... 315
Appendix D : MSC.Nastran Input Files and Entry Additions/Modifications ..... 331
D. 1 1D (Free-Free) Rotor Model ..... 331
D. 2 3D (Free-Free) Rotor Model ..... 337
D. 3 1D Rotor System Model with Linear Bearings ..... 341
D. 4 3D Rotor System Model with Linear Bearings ..... 372
D. 5 1D Rotor System Model with Nonlinear Journal Bearing ..... 395
D. 6 3D Rotor System Model with Nonlinear Journal Bearing ..... 402
Appendix E : MSC. Nastran UDS Programs ..... 409
E. 1 FORTRAN Journal Bearing Model Subroutines ..... 409
E. 2 C++ and F Source Code ..... 416
E. 3 Build Configuration Files ..... 417
Appendix F : Additional Results ..... 419
F. 1 Experimental Modal Analysis. ..... 419
F. 2 Bush Bearing Characterisation. ..... 425
F. 3 Coupling Lateral Characterisation. ..... 426
F. 4 Coupling Torsional Characterisation ..... 428
F. 5 1D (Free-Free) Rotor Model - Normal Mode Shapes ..... 429
F. 6 3D (Free-Free) Rotor Model - Normal Mode Shapes (Before Updating) ..... 430
F. 7 1D Rotor System Model with Linear Bearings ..... 431
F. 8 3D Rotor System Model with Linear Bearings ..... 438
Appendix G : Calibration Certificates and Technical Data Sheets ..... 442
G. 1 Journal Bearing Lubricant - Shell Tellus S2 V 46 Physical Properties ..... 442
G. 2 Proximity Probe Calibration Certificate ..... 442
Appendix H : Equations and Derivations ..... 443
H. 1 The atan2 Function ..... 443
H. 2 2D Central Difference Equations for $1^{\text {st }}$ and $2^{\text {nd }}$ Order Partial Derivatives ..... 443
H. 3 Newton-Raphson Method for Determining the Static Equilibrium Position of a Finite Hydrodynamic Journal Bearing ..... 443
H. 4 Analytical Solution for the Short Bearing Approximation ..... 445
Appendix I: Finite Element Rotor System Models Tree Diagrams ..... 448
I. 1 1D Rotor System Model ..... 448
I. 2 3D Rotor System Model ..... 453
LIST OF FIGURES
Figure 1.1 - Unbalance Force ..... 2
Figure 1.2 - Orbit Plot ..... 3
Figure 1.3 - The Gyroscopic Effect ..... 4
Figure 1.4 - Typical Campbell Diagram (Rao \& Sharan, 1994) ..... 5
Figure 1.5 - Oil Whirl \& Oil Whip: Hilbert Spectrum Plot (Fan et al., 2011) ..... 7
Figure 1.6 - Two Basic Types of Misalignment ..... 8
Figure 1.7 - Journal Bearing Parameters ..... 10
Figure 1.8-2D Grid ..... 12
Figure 1.9 - Linear Fluid Film Bearing Model ..... 18
Figure 1.10 - MSC.Nastran UNBALNC Entry Coordinate System \& Parameters ..... 22
Figure 1.11 - Flow Diagram of NLRSFD UDS Processing within the MSC.Nastran Application ..... 25
Figure 2.1 - Experimental Rotor System: Bently Nevada Rotor Kit ..... 29
Figure 2.2 - Rotor System Dimensions ..... 30
Figure 2.3 - Solid Model of Coupling with Setscrew ..... 32
Figure 2.4 - Solid Model of Entire Rotor ..... 32
Figure 2.5 - Experimental Modal Analysis Setup ..... 33
Figure 2.6 - Experimental Modal Analysis Measurement Points Coordinates ..... 34
Figure 2.7 - Free-Free Rotor Point 11 Vertical ( $y$-Direction) FRF ..... 35
Figure 2.8 - Free-Free Rotor Point 11 Horizontal (z-Direction) FRF ..... 36
Figure 2.9 - Free-Free Rotor Average Vertical FRF ..... 36
Figure 2.10 - Free-Free Rotor Average Horizontal FRF ..... 36
Figure 2.11 - Bush Bearing Characterisation Experimental Setup Schematic ..... 38
Figure 2.12 - Bush Bearing Characterisation Experimental Setup ..... 39
Figure 2.13 - Bush Bearing Characterisation Step Response of LVDTs (Load Case 4, Run 1) ..... 40
Figure 2.14 - Bush Bearing Characterisation Step Response of Bearings (Load Case 4, Run 1) ..... 41
Figure 2.15 - Bush Bearing Characterisation Isolated Step Response \& Linear Trends (Load Case 4, Run 1) ..... 41
Figure 2.16 - Bush Bearing Characterisation Windowed Step Response (Load Case 4, Run 1) ..... 42
Figure 2.17 - Bush Bearing Characterisation Windowed Step Response DFTs (Load Case 4, Run 1) ..... 42
Figure 2.18 - Bush Bearing Characterisation 2DOF \& SDOF Models ..... 43
Figure 2.19 - Bush Bearing Characterisation Filtered Step Response (Load Case 4, Run 1) ..... 43
Figure 2.20 - Bush Bearing Characterisation Filtered Step Response DFTs (Load Case 4, Run 1) ..... 44
Figure 2.21 - Bush Bearing Characterisation Filtered Step Response Curve Fitting (Load Case 4, Run 1) ..... 45
Figure 2.22 - Coupling Lateral Characterisation Experimental Setup Schematic ..... 47
Figure 2.23 - Coupling Lateral Characterisation Experimental Setup ..... 47
Figure 2.24 - Coupling Lateral Characterisation Windowed Step Response \& DFT (Load Case 1, Run 2) ..... 49
Figure 2.25 - Coupling Lateral Characterisation Filtered Step Response \& DFT (Load Case 1, Run 2) . 50
Figure 2.26 - Coupling Lateral Characterisation Filtered Step Response Curve Fitting (Load Case 1,Run 2)50
Figure 2.27 - Coupling Torsional Characterisation Experimental Setup Schematic ..... 52
Figure 2.28 - Coupling Torsional Characterisation Experimental Setup ..... 52
Figure 2.29 - Coupling Torsional Characterisation Step Response (Load Case d, Run 2) ..... 54
Figure 2.30-Coupling Torsional Characterisation Isolated \& Windowed Step Response (Load Case d, Run 2) ..... 54
Figure 2.31 - Coupling Torsional Characterisation Filtered Step Response (Load Case d, Run 2) ..... 54
Figure 2.32 - Coupling Torsional Characterisation Filtered Step Response Curve Fitting (Load Case d, Run 2) ..... 55
Figure 2.33 - Telescopic Gauge \& Micro Meter for Measuring Journal Bearing Clearance ..... 56
Figure 3.1 - Solid Models used to Determine Effective Inertial Properties for 1D Rotor Model ..... 59
Figure 3.2-1D Rotor System Model Dimensions ..... 60
Figure 3.3-1D Rotor Model in MSC. Patran ..... 61
Figure 3.4-1D Rotor Model Shaft Beam Elements Geometrical Properties ..... 62
Figure 3.5-1D Rotor Model 1 ${ }^{\text {st }}$ Symmetric Lateral Mode Shape Pair - 85.477 Hz ..... 63
Figure 3.6-1D Rotor Model $2^{\text {nd }}$ Symmetric Lateral Mode Shape Pair - 188.7 Hz ..... 63
Figure 3.7-1D Rotor Model 3 ${ }^{\text {rd }}$ Symmetric Lateral Mode Shape Pair - 458.33 Hz ..... 64
Figure 3.8 - Free-Free Natural Frequencies: 1D Rotor Model vs. Experimental Modal Analysis ..... 65
Figure 3.9-3D Rotor System Model Dimensions ..... 68
Figure 3.10-3D Rotor Model in MSC.Patran ..... 69
Figure 3.11 - 3D Rotor Model $1^{\text {st }}$ Symmetric Lateral Mode Shape Pair -94.028 Hz ..... 70
Figure 3.12-3D Rotor Model $2^{\text {nd }}$ Symmetric Lateral Mode Shape Pair - 212.18 Hz ..... 70
Figure 3.13-3D Rotor Model $1^{\text {st }}$ Torsional Mode Shape (Mode 11) - 260 Hz ..... 71
Figure 3.14-3D Rotor Model $3^{\text {rd }}$ Symmetric Lateral Mode Shape Pair -488.71 Hz ..... 71
Figure 3.15 - Free- Free Natural Frequencies: Initial \& Updated 3D Models vs. Experiment ..... 73
Figure 3.16-1D Rotor System Model with Linear Bearings ..... 74
Figure 3.17-1D Rotor System Model with Linear Bearings in MSC.Patran ..... 76
Figure 3.18 - Linear 1D Rotor System 1 ${ }^{\text {st }}$ BW \& FW Lateral Critical Speed Mode Shape Pair ..... 78
Figure 3.19 - Linear 1D Rotor System 2 ${ }^{\text {nd }}$ BW \& FW Critical Speed Mode Shape Pair ..... 78
Figure 3.20 - Linear 1D Rotor System $3^{\text {rd }}$ BW \& FW Critical Speed Mode Shape Pair ..... 79
Figure 3.21-1D Rotor System Model with Linear Bearings Campbell Diagram ..... 80
Figure 3.22 - Linear 1D Rotor System Bush Bearing Displacement Frequency Response ..... 82
Figure 3.23 - Linear 1D Rotor System Journal Displacement Frequency Response ..... 82
Figure 3.24 - Linear 1D Rotor System Disk 1 Displacement Frequency Response ..... 82
Figure 3.25 - Linear 1D Rotor System Journal Cascade Displacement Orbits ( $25-40 \mathrm{~Hz}$ ) ..... 83
Figure 3.26 - Linear 1D Rotor System Journal Cascade Displacement Orbits (120-160 Hz) ..... 83
Figure 3.27 - Linear 1D Rotor System Run-Up Bush Bearing Displacements ..... 85
Figure 3.28 - Linear 1D Rotor System Run-Up Journal Displacements ..... 85
Figure 3.29 - Linear 1D Rotor System Run-Up Disk 1 Displacements ..... 86
Figure 3.30-Linear 1D Rotor System Run-Up Displacement Orbits ..... 86
Figure 3.31 - Linear 1D Rotor System Run-Up Disk 1 2D Displacement ..... 87
Figure 3.32 - Linear 1D Rotor System Run-Up Journal Displacements (1-3 s) ..... 87
Figure 3.33 - Linear 1D Rotor System Run-Up Journal Displacement Magnitude (1-3 s) ..... 88
Figure 3.34 - Linear 1D Rotor System Run-Up Journal 2D Displacement (1-3 s) ..... 88
Figure 3.35 - Linear 1D Rotor System Run-Up Bush Bearing Vertical Displacement Spectrogram ..... 89
Figure 3.36 - Linear 1D Rotor System Run-Up Bush Bearing Forces ..... 90
Figure 3.37 - Linear 1D Rotor System Run-Up Journal Bearing Forces ..... 91
Figure 3.38 - Linear 1D Rotor System Run-Up 2D Journal Bearing Forces ..... 91
Figure 3.39 - Linear 1D Rotor System Run-Up Coupling Connection Spring Torque ..... 91
Figure 3.40 - Disks 1 \& 2 Run-Up Unbalance Forces ..... 92
Figure 3.41 - Disks 1 \& 2 Run-Up Unbalance Force Magnitudes ..... 92
Figure 3.42 - Disks 1 \& 2 Run-Up 2D Unbalance Forces. ..... 93
Figure 3.43-1D Rotor System Model with Linear Bearings Static Deformation due to Gravity ..... 94
Figure 3.44 - Linear 1D Rotor System with Gravity: Bush Bearing Displacements ..... 95
Figure 3.45 - Linear 1D Rotor System with Gravity: Journal Displacements ..... 95
Figure 3.46 - Rigid Rotor with Gravity Loading as an External Force ..... 96
Figure 3.47-1D Rotor System Model with Linear Bearings Static Deformation due to Simplified Gravity ..... 97
Figure 3.48 - Linear 1D Rotor System with Simplified Gravity: Bush Bearing Displacements ..... 98
Figure 3.49 - Linear 1D Rotor System with Simplified Gravity: Journal Displacements ..... 98
Figure 3.50 - Simple Flat Surface Rubbing Model ..... 99
Figure 3.51 - Rotor-Stator Contact Force as a Function of Penetration ..... 100
Figure 3.52 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Displacements ..... 101
Figure 3.53 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Displacement Orbit ..... 101
Figure 3.54 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Angular Displacement ..... 101
Figure 3.55 - Linear 1D Rotor System Flat Surface Rubbing Node 27 2D Displacement ..... 102
Figure 3.56 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Vertical Displacement Spectrogram ..... 102
Figure 3.57 - Linear 1D Rotor System Flat Surface Rubbing Forces \& Node 27 Displacements ..... 103
Figure 3.58 - Linear 1D Rotor System Flat Surface Rubbing Force \& Node 27 Displacement Magnitudes ..... 104
Figure 3.59 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Axial Moment ..... 104
Figure 3.60 - Simple Radial Rubbing Model ..... 105
Figure 3.61-1D Rotor System Model with Linear Bearings \& Radial Rubbing ..... 106
Figure 3.62 - Linear 1D Rotor System Radial Rubbing Node 27 Displacements ..... 107
Figure 3.63 - Linear 1D Rotor System Radial Rubbing Node 27 Displacement Orbit ..... 108
Figure 3.64 - Linear 1D Rotor System Radial Rubbing Node 27 Angular Displacement ..... 108
Figure 3.65 - Linear 1D Rotor System Radial Rubbing Node 27 2D Displacement ..... 108
Figure 3.66 - Linear 1D Rotor System Radial Rubbing Node 27 Vertical Displacement Spectrogram ..... 109
Figure 3.67 - Linear 1D Rotor System Radial Rubbing Forces ..... 110
Figure 3.68 - Linear 1D Rotor System Radial Rubbing 2D Force ..... 110
Figure 3.69 - Linear 1D Rotor System Radial Rubbing Force \& Node 27 Displacement Magnitudes . ..... 110
Figure 3.70 - Linear 1D Rotor System with Bearing Anisotropy: Journal Cascade Displacement Orbits111
Figure 3.71-3D Rotor System Model with Linear Bearings ..... 112
Figure 3.72-3D Rotor System Model Bearings \& Coupling Acting Centre MPCs ..... 113
Figure 3.73 - Linear 3D Rotor System $1^{\text {st }}$ Torsional Mode Shape (Mode 1) - 0 Hz ..... 116
Figure 3.74 - Linear 3D Rotor System $1^{\text {st }}$ BW \& FW Lateral Critical Speed Mode Shape Pair ..... 116
Figure 3.75 - Linear 3D Rotor System $2^{\text {nd }}$ Torsional Mode Shape (Mode 2) - 65.617 Hz ..... 117
Figure 3.76 - Linear 3D Rotor System $2^{\text {nd }}$ BW \& FW Lateral Critical Speed Mode Shape Pair ..... 117
Figure 3.77-Linear 3D Rotor System $3^{\text {rd }}$ BW \& FW Lateral Critical Speed Mode Shape Pair ..... 117
Figure 3.78 - Linear 3D Rotor System $4^{\text {th }}$ BW \& FW Lateral Critical Speed Mode Shape Pair ..... 118
Figure 3.79-3D Rotor System Model with Linear Bearings Campbell Diagram ..... 119
Figure 3.80 - Linear 3D Rotor System Bush Bearing Displacement Frequency Response ..... 120
Figure 3.81 - Linear 3D Rotor System Journal Displacement Frequency Response ..... 120
Figure 3.82 - Linear 3D Rotor System Disk 1 Displacement Frequency Response ..... 121
Figure 3.83 - Linear 3D Rotor System Journal Cascade Displacement Orbits ( $25-40 \mathrm{~Hz}$ ) ..... 121
Figure 3.84 - Linear 3D Rotor System Journal Cascade Displacement Orbits (115-155 Hz) ..... 122
Figure 3.85 - Linear 3D Rotor System Run-Up Bush Bearing Displacements ..... 123
Figure 3.86 - Linear 3D Rotor System Run-Up Journal Displacements ..... 123
Figure 3.87 - Linear 3D Rotor System Run-Up Disk 1 Displacements ..... 124
Figure 3.88 - Linear 3D Rotor System Run-Up Journal 2D Displacement (1-3 s) ..... 124
Figure 3.89 - Linear 3D Rotor System Run-Up Bush Bearing Vertical Displacement Spectrogram ..... 125
Figure 3.90-3D Rotor System Model with Linear Bearings Static Deformation due to Gravity ..... 126
Figure 3.91 - Linear 3D Rotor System with Simplified Gravity: Journal Displacements ..... 126
Figure 3.92-3D Rotor System Model with Linear Bearings Static Deformation due to Simplified Gravity ..... 127
Figure 3.93 - Linear 3D Rotor System with Simplified Gravity: Journal Displacements ..... 128
Figure 3.94 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 Displacements ..... 129
Figure 3.95 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 Displacement Orbit ..... 129
Figure 3.96 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 2D Displacement ..... 130
Figure 3.97 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 Vertical Displacement Spectrogram ..... 130
Figure 3.98 - Linear 3D Rotor System Flat Surface Rubbing Forces \& Node 1415 Displacements ..... 131
Figure 3.99-3D Rotor System Model with Linear Bearings \& Radial Rubbing ..... 132
Figure 3.100 - Linear 3D Rotor System Radial Rubbing Node 1415 Displacements ..... 133
Figure 3.101 - Linear 3D Rotor System Radial Rubbing Node 1415 Angular Displacement ..... 133
Figure 3.102 - Linear 3D Rotor System Radial Rubbing Node 1415 2D Displacement ..... 133
Figure 3.103 - Linear 3D Rotor System Radial Rubbing Node 1415 Vertical Displacement Spectrogram ..... 134
Figure 3.104 - Linear 3D Rotor System Radial Rubbing Forces ..... 135
Figure 3.105 - Linear 3D Rotor System Radial Rubbing Force \& Node 1415 Displacement Magnitudes ..... 135
Figure 3.106 - Grid on the Flattened Outer Surface of the Bearing Journal ..... 138
Figure 3.107 - Continuous Grid on the Outer Surface of the Bearing Journal ..... 139
Figure 3.108 - Journal Bearing Pressure Distribution (Cavitation Ignored) ..... 142
Figure 3.109 - Journal Bearing Pressure Distribution (Cavitation Included) ..... 143
Figure 3.110 - Journal Bearing Geometry and Pressure Distribution (Cavitation Ignored) ..... 144
Figure 3.111 - Journal Bearing Geometry and Pressure Distribution (Cavitation Included) ..... 144
Figure 3.112 - Journal Bearing Pressure Distribution Contours (Cavitation Boundaries) ..... 145
Figure 3.113-3D Pressure Distribution around Journal ..... 146
Figure 3.114 - Journal Static Equilibrium Path/Locus as Bearing Load Increases ..... 147
Figure 3.115 - Comparison of Different Journal Bearing Models ..... 148
Figure 3.116 - UDS Build Environment Source Tree ..... 149
Figure 3.117-1D Rotor System Model with a Nonlinear Journal Bearing ..... 152
Figure 3.118-3D Rotor System Model with a Nonlinear Journal Bearing ..... 153
Figure 3.119-3D Rotor System Model Modified Journal Bearing Force Transfer Structure ..... 154
Figure 3.120 - Global \& Local Coordinate Systems of the Rotor System \& Journal Bearing Respectively ..... 155
Figure 3.121-1D Rotor with Nonlinear Journal Bearing: Journal Displacements ..... 156
Figure 3.122-1D Rotor with Nonlinear Journal Bearing: Journal Displacement Orbit ..... 157
Figure 3.123-1D Rotor with Nonlinear Journal Bearing: Journal 2D Displacement ..... 157
Figure 3.124-1D Rotor with Nonlinear Journal Bearing: Disk 1 Displacements ..... 158
Figure 3.125-1D Rotor with Nonlinear Journal Bearing: Disk 1 Vertical Displacement Spectrogram ..... 158
Figure 3.126 - Basic Iterative Procedures to Determine Equilibrium State of Rotor with Nonlinear Bearing ..... 160
Figure 3.127-1D Rotor with Nonlinear Journal Bearing \& Gravity: Bush Bearing Displacements ..... 162
Figure 3.128-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements ..... 162
Figure 3.129-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements \& Clearance ..... 163
Figure 3.130-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacement Orbit ..... 163
Figure 3.131-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal 2D Displacement ..... 164
Figure 3.132-1D Rotor with Journal Bearing \& Gravity: Journal Vertical Displacement Spectrogram ..... 164
Figure 3.133-1D Rotor with Journal Bearing \& Gravity: Journal Horizontal Displacement Spectrogram ..... 165
Figure 3.134-3D Rotor with Nonlinear Journal Bearing: Journal Displacements ..... 166
Figure 3.135-3D Rotor with Nonlinear Journal Bearing: Journal 2D Displacement ..... 166
Figure 3.136-3D Rotor with Nonlinear Journal Bearing: Disk 1 Displacements ..... 167
Figure 3.137-3D Rotor with Nonlinear Journal Bearing: Disk 1 Displacement Spectrogram ..... 167
Figure 3.138-3D Rotor with Nonlinear Journal Bearing \& Gravity: Bush Bearing Displacements ..... 169
Figure 3.139-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements ..... 169
Figure 3.140-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements \& Clearance ..... 170
Figure 3.141-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacement Orbit ..... 170
Figure 3.142-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal 2D Displacement ..... 171
Figure 3.143-3D Rotor with Journal Bearing \& Gravity: Journal Vertical Displacement Spectrogram ..... 171
Figure 3.144-3D Rotor with Journal Bearing \& Gravity: Journal Horizontal Displacement Spectrogram ..... 172
Figure 4.1 - Two-Plane Balancing: Balancing Planes, Bearings \& Phase Reference ..... 173
Figure 4.2 - Two-Plane Balancing: Instrumentation ..... 174
Figure 4.3 - Runout \& Vibration Measurement when DC Component Unimportant ..... 176
Figure 4.4 - Runout \& Vibration Measurement when DC Component Important ..... 177
Figure 4.5 - Rotor System Transient Unbalance Response Experimental Setup Schematic ..... 178
Figure 4.6 - Rotor System Transient Unbalance Response Experimental Setup: Overall ..... 179
Figure 4.7 - Rotor System Transient Unbalance Response Experimental Setup: Sensors \& Fluid Film ..... 180
Figure 4.8 - Journal Bearing Housing-Proximity Probes Gaps \& Bearing Clearance Measurement ..... 180
Figure 4.9 - Rotor System Bush Bearing y-Probe Runout Gap Signal ..... 181
Figure 4.10 - Rotor System Journal Bearing y-Probe Runout Gap Signal. ..... 182
Figure 4.11 - Rotor System Slow-Roll Driving End Tachometer Signal ..... 182
Figure 4.12 - Rotor System Slow-Roll Spin Speed ..... 183
Figure 4.13 - Rotor System Bush Bearing Runout Gap Signal Synchronous Ave. ..... 184
Figure 4.14 - Rotor System Journal Bearing Runout Gap Signal Synchronous Ave ..... 184
Figure 4.15 - Rotor System Bush Bearing y- \& z-Probe Runout Phase Correlation ..... 184
Figure 4.16 - Rotor System Journal Bearing y- \& z-Probe Runout Phase Correlation ..... 185
Figure 4.17 - Rotor System Bush Bearing \& Journal Bearing y-Probe Runout Radii ..... 185
Figure 4.18 - Rotor System Run-Up Bush Bearing Gap Signals - 1 g Unbalance ..... 187
Figure 4.19 - Rotor System Run-Up Journal Bearing Gap Signals - 1 g Unbalance ..... 187
Figure 4.20 - Rotor System Run-Up Non-Driving End Tachometer Signal ..... 187
Figure 4.21 - Rotor System Run-Up Spin Speed - 1 g Unbalance Test ..... 188
Figure 4.22 - Rotor System Run-Up Thermocouple Signal - 1 g Unbalance ..... 188
Figure 4.23 - Rotor System Run-Up Bush Bearing Displacements - 1 g Unbalance ..... 189
Figure 4.24 - Rotor System Run-Up Journal Bearing Displacements \& Clearance - 1 g Unbalance. ..... 189
Figure 4.25 - Rotor System Run-Up Journal Bearing Displacement Orbits - 1 g Unbalance ..... 190
Figure 4.26 - Rotor System Run-Up Journal Bearing 2D Displacement - 1 g Unbalance. ..... 190
Figure 4.27 - Rotor System Run-Up Journal Bearing Vertical Displacement Spectrogram - 1 g
Unbalance ..... 191
Figure 4.28 - Rotor System Run-Up Journal Bearing Horizontal Displacement Spectrogram - 1 gUnbalance191
Figure 4.29 - Rotor System Run-Up Spin Speed - 2 g Unbalance Test ..... 192
Figure 4.30 - Rotor System Run-Up Thermocouple Signal - 2 g Unbalance ..... 192
Figure 4.31 - Rotor System Run-Up Bush Bearing Displacements - 2 g Unbalance ..... 193
Figure 4.32 - Rotor System Run-Up Journal Bearing Displacements \& Clearance - 2 g Unbalance ..... 193
Figure 4.33 - Rotor System Run-Up Journal Bearing Displacement Orbits - 2 g Unbalance ..... 194
Figure 4.34 - Rotor System Run-Up Journal Bearing 2D Displacement - 2 g Unbalance ..... 194
Figure 4.35 - Rotor System Run-Up Journal Bearing Vertical Displacement Spectrogram - 2 gUnbalance195
Figure 4.36 - Rotor System Run-Up Journal Bearing Horizontal Displacement Spectrogram - 2 gUnbalance195
Figure 4.37 - Rotor System Flat Surface Rubbing Response Experimental Setup ..... 196
Figure 4.38 - Rotor System Rubbing Location Runout Gap Signal Synchronous Ave ..... 197
Figure 4.39 - Rotor System Journal Bearing Runout Gap Signal Synchronous Ave. - Rubbing Experiment ..... 197
Figure 4.40 - Rotor System Run-Up Rubbing Location Gap Signal ..... 198
Figure 4.41 - Rotor System Run-Up Spin Speed - Rubbing Experiment ..... 198
Figure 4.42 - Rotor System Run-Up Rubbing Location Displacement ..... 198
Figure 4.43 - Rotor System Flat Surface Rubbing Vertical Displacement Spectrogram at Rubbing Location ..... 199
Figure 5.1 - Experimental vs. 1D Model Run-Up Unbalance Response Displacements - 1g Unbalance ..... 202
Figure 5.2 - Experimental vs. 1D Model Run-Up Unbalance Response Displacements - 2g Unbalance ..... 202
Figure 5.3 - Experimental vs. 1D Model Run-Up Unbalance Response 2D Displacements - 1gUnbalance203
Figure 5.4 - Experimental vs. 1D Model Run-Up Unbalance Response 2D Displacements - 2g Unbalance ..... 203
Figure 5.5 - Experimental vs. 1D Model Run-Up Unbalance Response Orbits - 2g Unbalance ..... 203
Figure 5.6 - Experimental vs. 3D Model Run-Up Unbalance Response Displacements - 1 g Unbalance204
Figure 5.7 - Experimental vs. 3D Model Run-Up Unbalance Response Displacements - 2 g Unbalance204
Figure 5.8 - Experimental vs. 3D Model Run-Up Unbalance Response 2D Displacements - 1g
Unbalance ..... 205
Figure 5.9 - Experimental vs. 3D Model Run-Up Unbalance Response 2D Displacements - 2gUnbalance205
Figure 5.10 - Experimental vs. 3D Model Run-Up Unbalance Response Orbits - 2g Unbalance ..... 205
Figure 5.11 - Experimental vs. 1D Model Run-Up Rubbing Location Displacements ..... 206
LIST OF TABLES
Table 2.1 - Measured Rotor Masses ..... 31
Table 2.2 - Approximate Rotor Material Densities ..... 32
Table 2.3 - Experimental Modal Analysis Laser Vibrometer Settings ..... 34
Table 2.4 - Free-Free Rotor Experimental Modal Analysis Natural Frequencies ..... 37
Table 2.5 - Bush Bearing Characterisation Static Load Cases ..... 39
Table 2.6 - Bush Bearing Characterisation Natural Frequencies \& Stiffness Ratios ..... 46
Table 2.7 - Bush Bearing Characterisation Results ..... 46
Table 2.8 - Coupling Lateral Characterisation Static Load Cases ..... 48
Table 2.9 - Coupling Lateral Characterisation Laser Vibrometer Settings ..... 48
Table 2.10 - Coupling Lateral Characterisation Results ..... 51
Table 2.11 - Coupling Torsional Characterisation Static Load Cases ..... 53
Table 2.12 - Coupling Torsional Characterisation Results ..... 55
Table 3.1 - System of Consistent Units used for FE Modelling ..... 58
Table 3.2 - Effective Inertial Properties for 1D Rotor Model ..... 59
Table 3.3-1D Rotor Model Shaft Beam Elements Geometrical Properties ..... 61
Table 3.4-1D Rotor Model Concentrated Mass Elements Properties ..... 62
Table 3.5-1D Free-Free Rotor Model Natural Frequencies ..... 63
Table 3.6 - Ave. Experimental/Test Natural Frequencies with Corresponding Weighting Values ..... 65
Table 3.7-3D Rotor Model Equivalent Simplified Axisymmetric Parts Properties ..... 67
Table 3.8-3D Free-Free Rotor Model Natural Frequencies (Before Updating) ..... 69
Table 3.9 - Free-Free Natural Frequencies: Initial \& Updated 3D Models vs. Experimental Modal Analysis ..... 72
Table 3.10 - Rotor System Model Linear Support Element Properties ..... 75
Table 3.11 - Rotor System Model Disk Imbalance Properties ..... 76
Table 3.12-1D Rotor System Model with Linear Bearings Natural Frequencies ..... 77
Table 3.13-1D Rotor System Model with Linear Bearings Critical Speeds ..... 77
Table 3.14-1D Rotor System Model Linear Support Deflections \& Reaction Forces due to Gravity ..... 94
Table 3.15-1D Rigid Rotor System Linear Support Deflections \& Reaction Forces due to Simplified Gravity ..... 97
Table 3.16-3D Rotor System Model with Linear Bearings Undamped Natural Frequencies ..... 114
Table 3.17-3D Rotor System Model with Linear Bearings Damped Natural Frequencies ..... 115
Table 3.18 - 3D Rotor System Model with Linear Bearings Critical Speeds ..... 115
Table 3.19 - Comparison between 1D \& 3D Rotor System Models with Linear Bearings Critical Speeds ..... 118
Table 3.20-3D Rotor System Linear Support Deflections \& Reaction Forces due to Gravity ..... 126
Table 3.21-3D Rigid Rotor System Linear Support Deflections \& Reaction Forces due to Simplified Gravity ..... 127
Table 3.22 - Journal Bearing Demonstration Input Parameters ..... 142
Table 3.23 - FiniteHDJB.f90-ext_nlrsfd.F Input \& Output Parameter Mapping ..... 150
Table 3.24-1D Rotor System with a Nonlinear Journal Bearing Simplified Gravity Loading @ 2500 rpm ..... 160
Table 3.25-3D Rotor System with a Nonlinear Journal Bearing Simplified Gravity Loading @ 2500 rpm ..... 168
Table 4.1 - Two-Plane Balancing Measurements \& Results ..... 175
Table 5.1 - Applied Disk 2 Imbalances from Experimental Analysis used in Rotor System Models ... 201
Table 5.2 - Approximate Residual Unbalance of the Experimental Rotor System ..... 201

## List OF Symbols

Note: the same symbol might have different meanings in different contexts (i.e. in different sections)

| Symbol | Description | Unit |
| :---: | :---: | :---: |
| General: |  |  |
| $L$ | Angular momentum vector | Nm.s |
| I | Mass moment of inertia | kg. $\mathrm{m}^{2}$ |
| $\tau$ | Torque vector | Nm |
| $\omega, \boldsymbol{\Omega}$ | Angular velocity vector | rad/s |
| [M] | Mass matrix | kg; kg. $\mathrm{m}^{2}$ |
| [C] | Damping matrix | $\mathrm{Ns} / \mathrm{m}$; |
|  |  | Nm.s/rad |
| [G] | Gyroscopic matrix | Nm. $\mathrm{s}^{2} / \mathrm{rad}$ |
| [K] | Stiffness matrix | $N / m$; |
|  |  | $\mathrm{Nm} / \mathrm{rad}$ |
| $\{u\}$ | Displacement vector | $m ; r a d,{ }^{\circ}$ |
| $\{\dot{u}\}$ | Velocity vector | $\mathrm{m} / \mathrm{s} ; \mathrm{rad} / \mathrm{s}$ |
| $\{\ddot{u}\}$ | Acceleration vector | $\mathrm{m} / \mathrm{s}^{2}$; |
|  |  | $\mathrm{rad} / \mathrm{s}^{2}$ |
| $\{F\}$ | External force/torque vector | N; Nm |
| $\zeta$ | Damping coefficient | - |
| $\omega_{n} ; f_{n}$ | Natural frequency | rad/s; Hz |
| $\omega_{d} ; f_{d}$ | Damped natural frequency | rad/s; Hz |
| $\boldsymbol{F}_{n} \& \boldsymbol{F}_{t}$ | Normal and tangential force vector components respectively | $N$ |
| $k$ | Stiffness coefficient | $N / m$ |
| c | Damping coefficient | Ns/m |
| $t$ | Time | $s$ |
| $x, y, z$ | Translational coordinates | $m$ |
| $\theta$ | Angular coordinate | rad, ${ }^{\circ}$ |
| $\dot{x}, \dot{y}, \dot{z}$ | Translational velocities | $\mathrm{m} / \mathrm{s}$ |
| $\mu$ | Dynamic viscosity | Pa.s |
| $\rho$ | Mass density | $\mathrm{kg} / \mathrm{m}^{3}$ |
| $v$ | Kinematic viscosity | $\mathrm{m}^{2} / \mathrm{s}$ |
| $\phi$ | Phase angle | rad, ${ }^{\circ}$ |
| $c_{0}$ | Critical damping coefficient | Ns/m |
| T | Period | $s$ |
| E | Modulus of elasticity | $P a$ |
| $m$ | Mass | kg |
| $\boldsymbol{F}_{u}$ | Unbalance force vector | $N$ |
| $I_{x x}, I_{y y}, I_{z z}$ | Mass moments of inertia in the $x$-, $y$ - and $z$-directions respectively | kg. $\mathrm{m}^{2}$ |
| $g$ | Gravitational acceleration | $\mathrm{m} / \mathrm{s}^{2}$ |
| $f$ | Frequency | Hz |
| $\omega ; N$ | Rotor/journal spin speed | rad/s;rpm |
| $\dot{\omega}$ | Rotor spin acceleration | $\mathrm{rad} / \mathrm{s}^{2}$ |


| Fluid Film/Journal Bearing: |  |  |
| :---: | :---: | :---: |
| $A_{1 i}$ to $A_{4 i}$; | Reynolds equation finite difference approximation coefficients | - |
| $P_{\# \# i} \& f_{i}$ |  |  |
| $p_{i, j}$ | Fluid film pressure at discrete coordinates $\theta_{i}$ and $z_{j}$ | $P a$ |
| $p_{i, j}^{G S}$ | Gauss-Seidel updated fluid film pressure at discrete coordinates $\theta_{i}$ and $z_{j}$ | $P a$ |
| $\alpha_{o p t}$ | Optimal over-relaxation factor | - |
| $\theta^{\prime}$ | Angular coordinate ( $\theta^{\prime}=0$ @ line of centres) | rad, ${ }^{\circ}$ |
| $\theta_{i} \& z_{j}$ | Discrete $\theta$ - and $z$-coordinates at grid point numbers $i$ and $j$ respectively | $\mathrm{rad},{ }^{\circ}$ \& $m$ |
| $\Delta \theta \& \Delta z$ | Constant grid spacing sizes along the $\theta$ - and $z$-coordinates respectively | $\mathrm{rad},{ }^{\circ}$ \& $m$ |
| $\theta_{1} \& \theta_{2}$ | Angular locations of the starting and break-down boundaries of the load carrying region respectively (cavitation boundaries) | rad, ${ }^{\circ}$ |
| D | Bearing housing inner diameter | $m$ |
| $d$ | Journal diameter | $m$ |
| $h$ | Fluid film thickness | $m$ |
| $c$ | Radial clearance | $m$ |
| $e$ | Eccentricity | $m$ |
| $\varepsilon$ | Eccentricity ratio | - |
| $i \& j$ | Discrete grid point numbers (integers) along the $\theta$ - and $z$ coordinates respectively | - |
| $k$ | Iteration/index number (integer) | - |
| $l$ | Bearing length | $m$ |
| $n \& m$ | Number of mesh divisions/sections in the $\theta$ - and $z$-directions respectively | - |
| $p$ | Fluid film pressure | $P a$ |
| $r$ | Journal radius | $m$ |
| $R$ | Bearing housing inner radius | $m$ |
| $\boldsymbol{W}$ | Rotor weight load vector at the journal centre | $N$ |
| $\alpha$ | Over-relaxation factor | - |
| $\phi$ | Attitude angle | rad, ${ }^{\circ}$ |
| $k_{x} \& k_{y}$ | Direct linearized stiffness coefficients in the $x$ - and $y$-directions respectively | $N / m$ |
| $k_{x y} \& k_{y x}$ | Cross-coupled linearized stiffness coefficients in the $x$ - and $y$ directions respectively | $N / m$ |
| $c_{x} \& c_{y}$ | Direct linearized damping coefficients in the $x$ - and $y$-directions respectively | $N s / m$ |
| $c_{x y} \& c_{y x}$ | Cross-coupled linearized damping coefficients in the $x$ - and $y$ directions respectively | $N s / m$ |
| $F_{x} \& F_{y}$ | Fluid film forces in the $x$ - and $y$-directions respectively | $N$ |
| $p_{\text {cav }}$ | Cavitation pressure | $P a$ |
| Rotor System: |  |  |
| $e$ | Shaft eccentricity from rubbing housing centre | $m$ |
| $r$ | Shaft radius | $m$ |


| $f_{c} ; N_{c}$ | Critical speed | $\mathrm{rad} / \mathrm{s} ; \mathrm{Hz}$ |
| :---: | :---: | :---: |
| $g_{\text {ROy }} \& g_{\text {ROz }}$ | Gaps measured by $y$-and $z$-probes respectively during slow-roll (runout) | $m$ |
| $g_{y} \& g_{z}$ | Gaps measured by $y$-and $z$-probes respectively during vibration measurement | $m$ |
| $r_{R O y} \& r_{R O z}$ | Shaft/journal radii, including runout, in the $y$ - and $z$-directions respectively | $m$ |
| $v_{R O} \& w_{R O}$ | Deviations from ideal shaft/journal radius due to runout in the $y$ and $z$-directions respectively | $m$ |
| $g_{h y} \& g_{h z}$ | Gaps between the journal bearing housing and the $y$-and $z$-probes respectively | $m$ |
| $y_{R O} \& z_{R O}$ | Journal bearing centre coordinates during slow-roll (runout) | $m$ |
| $m_{j}$ | Journal mass | kg |
| $m_{c}$ | Coupling mass | kg |
| $m_{d 1} \& m_{d 2}$ | Mass of disks 1 and 2 respectively | kg |
| $E_{1} \& E_{2}$ | Moduli of elasticity of the shaft and coupling respectively | $P a$ |
| $k_{b y} \& k_{b z}$ | Bush bearing vertical and horizontal stiffness coefficients respectively | $N / m$ |
| $k_{j y} \& k_{j z}$ | Journal bearing vertical and horizontal stiffness coefficients respectively | $N / m$ |
| $k_{c y} \& k_{c z}$ | Coupling connection vertical and horizontal stiffness coefficients respectively | $N / m$ |
| $k_{c \theta}$ | Coupling connection torsional stiffness coefficient | $\mathrm{Nm} / \mathrm{rad}$ |
| $c_{b y} \& c_{b z}$ | Bush bearing vertical and horizontal damping coefficients respectively | Ns/m |
| $c_{j y} \& c_{j z}$ | Journal bearing vertical and horizontal damping coefficients respectively | $N s / m$ |
| $c_{c y} \& c_{c z}$ | Coupling connection vertical and horizontal damping coefficients respectively | Ns/m |
| $C_{c \theta}$ | Coupling connection torsional damping coefficient | Nm.s/rad |
| $\theta_{u 1} \& \theta_{u 2}$ | Imbalance angular positions for disks 1 and 2 respectively | rad, ${ }^{\circ}$ |
| $m_{u 1} \& m_{u 2}$ | Imbalance masses for disks 1 and 2 respectively | kg |
| $F_{u 1} \& F_{u 2}$ | Unbalance forces for disks 1 and 2 respectively | $N$ |
| $r_{u 1} \& r_{u 2}$ | Imbalance radii for disks 1 and 2 respectively | $m$ |
| $F_{b y} \& F_{c y}$ | Bush bearing and coupling vertical forces respectively | $N$ |
| $F_{j y} \& F_{j z}$ | Journal bearing vertical and horizontal forces respectively | $N$ |
| $l_{b}, l_{j} \& l_{c}$ | Distances between the acting centres of the bush bearing, journal bearing and coupling and the rotor system global coordinate system $y$-axis respectively | $m$ |
| $l_{C G}$ | Distance between the rotor centre of gravity and the rotor system global coordinate system $y$-axis | $m$ |
| $y_{j} \& z_{j}$ | Journal bearing acting centre vertical and horizontal displacements respectively | $m$ |
| $y_{b} \& y_{c}$ | Bush bearing and coupling acting centres vertical displacements respectively | $m$ |
| $p$ | Rotor-stator penetration | $m$ |


| $\mu_{f}$ | Dynamic friction coefficient | - |
| :---: | :---: | :---: |
| $F_{r y} \& F_{r z}$ | Vertical and horizontal rubbing forces respectively | $N$ |
| $M_{r x}$ | Axial rubbing moment | Nm |
| $\theta_{r}$ | Shaft-rubbing housing line of centres angular coordinate | rad, ${ }^{\circ}$ |
| $F_{r h y} \& F_{r h z}$ | Vertical and horizontal rubbing forces on the rubbing housing respectively | $N$ |
| $M_{r h x}$ | Axial rubbing moment on the rubbing housing | $N m$ |
| $k_{r h y} \& k_{r h z}$ | Rubbing housing vertical and horizontal stiffness coefficients respectively | $N / m$ |
| $\{P(t)\}$ | Time-dependent load vector | $N$ |
| $\{A\}$ | Constant-magnitude factor/coefficient | - |
| $k_{j h y} \& k_{j h z}$ | Journal bearing housing vertical and horizontal stiffness coefficients respectively | $N / m$ |
| Rotor Support Structure Characterisation: |  |  |
| $x_{1} \& x_{2}$ | Displacements at the centres of bush bearings 1 and 2 respectively | $m$ |
| $x_{A} \& x_{B}$ | Displacements measured by LVDTs A and B respectively | $m$ |
| $l_{A, 1}$ | Distance between bush bearing 1 and LVDT A | $m$ |
| $l_{B, 2}$ | Distance between bush bearing 2 and LVDT B | $m$ |
| $l_{1,2}$ | Distance between bush bearings 1 and 2 | $m$ |
| $l_{o f f}$ | Offset distance between sleeve and coupling acting centre | $m$ |
| $l_{w}$ | Distance between wheels attached to the driven coupling half | $m$ |
| $l_{m 1} \& l_{m 2}$ | Distances from weight hangers 1 and 2 to the coupling centreline respectively | $m$ |
| $l_{\text {boltG1 }}$ \& | Distances from the centres of gravity of bolts 1 and 2 to the | $m$ |
| $l_{\text {bolt }{ }^{\text {2 }} \text { }}$ | coupling centreline respectively |  |
| $\delta$ | Logarithmic decrement | - |
| $k_{\theta}$ | Angular stiffness coefficient | Nm/rad |
| $c_{\theta}$ | Angular damping coefficient | Nm.s/rad |
| $\boldsymbol{F}_{l}$ | Static load vector | $N$ |
| $k_{1} \& k_{2}$ | Stiffness coefficients of bush bearings 1 and 2 respectively | $N / m$ |
| $c_{1} \& c_{2}$ | Damping coefficients of bush bearings 1 and 2 respectively | $\mathrm{Ns} / \mathrm{m}$ |
| $r_{k}$ | Stiffness ratio | - |
| $m_{s}$ | Shaft mass | kg |
| $m_{l}$ | Static shaft load mass | $k g$ |
| $k_{\text {eff }}$ | Effective stiffness coefficient | $N / m$ |
| $c_{e f f}$ | Effective damping coefficient | $\mathrm{Ns} / \mathrm{m}$ |
| $X_{0}$ | Displacement amplitude | $m$ |
| $\theta_{0}$ | Angular displacement amplitude | rad, ${ }^{\circ}$ |
| $x_{\text {exp }}$ | Exponential decay factor | $m$ |
| $a \& b$ | Exponential function coefficients | $m \& \mathrm{rad} / \mathrm{s}$ |
| $\boldsymbol{F}_{w}$ | Wheel force vector | $N$ |

## Model Updating:

$E F \quad$ Error function
$w t$ \& $w m \quad$ Overall test result and model weighting factors respectively

| $w r_{k} \&$ | Weighting factors for response $k$ and model parameter $k$ | - |
| :---: | :--- | :---: |
| $w p_{k}$ | respectively |  |
| $r t_{k} \& r m_{k}$ | $k^{\text {th }}$ test and model responses respectively | - |
| $m p_{k} \& m p_{k i}$ | $k^{\text {th }}$ current and initial model parameters respectively | - |
| $f_{n k t} \& f_{n k m}$ | $k^{\text {th }}$ test and model natural frequencies respectively | $P a$ |
| $E_{i}$ | Initial modulus of elasticity | $P a$ |

List of Abbreviations/Acronyms

| OD | Zero-dimensional |
| :---: | :---: |
| 1D | One-dimensional |
| 2D | Two-dimensional |
| 2DOF | Two degree of freedom |
| 3D | Three-dimensional |
| BC | Boundary condition |
| BVP | Boundary value problem |
| BW | Backward whirling |
| C.G. | Centre of gravity |
| CDL | Component definition language |
| CFD | Computational fluid dynamics |
| CMS | Component modal synthesis |
| CSV | Comma-separated-value |
| DC | Direct current |
| DFT | Discrete Fourier transform |
| DLL | Dynamic-link library |
| DOF | Degree of freedom |
| EOM | Equation of motion |
| FE | Finite element |
| FEA | Finite element analysis |
| FEM | Finite element method |
| FFT | Fast Fourier transform |
| FMS | File management section |
| FRF | Frequency response function |
| FSI | Fluid-structure interaction |
| FW | Forward whirling |
| HHT | Hilbert-Huang transform |
| IDL | Interface definition language |
| LVDT | Linear variable displacement transducer |
| MPC | Multi-point constraint |
| PFA | Polytec File Access |
| PSD | Power spectral density |
| SCA | Simulation component architecture |
| SDK | Software development kit |
| SDL | Service definition language |
| SDOF | Single degree of freedom |
| SI | International system of units |
| SOR | Successive over-relaxation |
| STFT | Short-time Fourier transform |
| TMM | Transfer matrix method |
| UDS | User defined service |

## 1 InTRODUCTION

### 1.1 Background

Eddy current proximity probes are used worldwide to measure lateral vibration information on turbogenerator shafts for diagnostic purposes. The signal from such a proximity probe is a function of the distance between the probe and shaft surface, as well as the electrical conductivity and magnetic permeability of the shaft material (Littrell, 2005). If a shaft is not perfectly round or if its electrical properties vary along its circumference, a proximity probe will measure these variations known as mechanical and electrical runout respectively. Runout (which can be measured on a slowrotating shaft - usually at less than $10 \%$ of the shaft operational speed) is not part of the desired vibration data and is therefore often compensated for by subtracting it from the total measured signal (usually measured at the shaft operational speed). Rotor vibration may, however, cause this compensation to be inaccurate above a critical speed if runout was measured below the critical speed. On the other hand, thermal expansion from low to high shaft speed could lead to the situation where runout is measured at a specific shaft section at low speed while the probes measure a different section at operational speed. Even though proximity probes have a frequency range of typically 10 kHz , these runout compensation inaccuracies reduce the useful frequency range (dynamic range) of proximity probes to approximately ten or fewer harmonics of the shaft speed, severely limiting their diagnostic capabilities (Randall, 2011).

Furthermore, power plant personnel are often required to manually identify faults on turbogenerators from proximity probe vibration data (using tools such as shaft orbit charts). This is a time consuming task which requires training, experience and is susceptible to human error (such as incorrect orbit plot interpretation, resulting in incorrect fault identification). An automated fault detection system could therefore be greatly advantageous from a diagnostic point of view.

This dissertation forms part of a research project that aims to address these limitations by investigating the optimization of information from eddy current proximity probes currently in use by Eskom ${ }^{1}$ for diagnostic purposes. This research project includes a comprehensive development program in which a laboratory-scale rotor system, supported by at least one hydrodynamic journal bearing, needs to be used as a test platform and the effects of artificially induced faults on this system need to be investigated. A mathematical model of this rotor system (including simulated faults) also needs to be developed to serve as a benchmark for data measured on the rotor system. Following this, data processing techniques (including artificial intelligence in the form of neural networks) need to be developed to, amongst other capabilities, determine the rotor system condition and possible faults from measured or simulated data. Suitable diagnostic signal parameters are to be identified and trended. Finally, in order to increase the useful proximity probe frequency bandwidth, a methodology needs to be developed to perform runout compensation in an unsupervised manner. The focus of this dissertation is specifically on the development, analysis and validation of mathematical models of the laboratory-scale rotor system.

[^0]
### 1.2 Literature Study

### 1.2.1 Rotor Dynamics and Faults

Rotor dynamics concerns the dynamic behaviour and analysis of rotating structures and may be divided into two main categories: torsional and lateral. The former concerns the dynamics and analysis of torsional excitations and vibrations (e.g. turbine blade vibration) while the latter concerns the dynamics and analysis of radial/bending vibrations of rotors, i.e. the radial-plane orbital motion of the rotor spin axis. The focus of this dissertation is mainly on lateral rotor dynamics.

This section provides a brief description of a few basic concepts in lateral rotor dynamics and also discusses certain faults and other phenomena which may have an effect on the lateral dynamics of rotors.

### 1.2.1.1 Mass Imbalance

Imperfections will always be present in rotors due to manufacturing errors, despite high tolerances and sophisticated machinery used. Therefore, rotors often possess mass imbalance due to these imperfections and a mass imbalance results in (or is usually interpreted as) an unbalanced excitation force (often simply referred to as unbalance) which rotates with the rotor, i.e. at a frequency equal to the rotor spin speed (Rao \& Sharan, 1994).

Mass imbalance at a certain cross-section of a rotor may be represented by an equivalent concentrated mass $m_{u}$ with an eccentricity $r_{u}$ as shown in Figure 1.1.


Figure 1.1 - Unbalance Force

For a rotor with spin speed $\omega$ (which, in general, is a function of time), the unbalance force $\boldsymbol{F}_{u}$ is given (as a function of time $t$ ) by:

$$
\begin{equation*}
\boldsymbol{F}_{u}(t)=\boldsymbol{F}_{n}(t)+\boldsymbol{F}_{t}(t) \tag{1.1}
\end{equation*}
$$

where $\boldsymbol{F}_{n}(t)$ and $\boldsymbol{F}_{t}(t)$ are its centrifugal (or normal) and tangential components. In a stationary Cartesian coordinate system defined as shown in Figure 1.1 (coordinates $x$ and $y$, its origin at the geometric centre of the rotor and $\theta(t)$ the angle between the positive $x$-axis and the line connecting the geometric centre with the imbalance mass), $\boldsymbol{F}_{n}(t)$ and $\boldsymbol{F}_{t}(t)$ may be written as...

$$
\begin{align*}
& \boldsymbol{F}_{n}(t)=\left\{\begin{array}{l}
F_{n x}(t) \\
F_{n y}(t)
\end{array}\right\}=m_{u} r_{u} \omega^{2}(t)\left\{\begin{array}{c}
\cos \theta(t) \\
\sin \theta(t)
\end{array}\right\} \\
& \boldsymbol{F}_{t}(t)=\left\{\begin{array}{l}
F_{t x}(t) \\
F_{t y}(t)
\end{array}\right\}=m_{u} r_{u} \dot{\omega}(t)\left\{\begin{array}{c}
\sin \theta(t) \\
-\cos \theta(t)
\end{array}\right\} \tag{1.2}
\end{align*}
$$

Therefore:

$$
\begin{align*}
\boldsymbol{F}_{u}(t)=\left\{\begin{array}{l}
F_{u x}(t) \\
F_{u y}(t)
\end{array}\right\} & =\left\{\begin{array}{l}
F_{n x}(t)+F_{t x}(t) \\
F_{n y}(t)+F_{t y}(t)
\end{array}\right\} \\
& =\left\{\begin{array}{l}
m_{u} r_{u} \omega^{2}(t) \cos \theta(t)+m_{u} r_{u} \dot{\omega}(t) \sin \theta(t) \\
m_{u} r_{u} \omega^{2}(t) \sin \theta(t)-m_{u} r_{u} \dot{\omega}(t) \cos \theta(t)
\end{array}\right\} \tag{1.3}
\end{align*}
$$

For a rotor with a constant spin speed $\omega(\dot{\omega}=0)$, the unbalance force has no tangential component and is therefore given by:

$$
\left.\boldsymbol{F}_{u}\right|_{\dot{\omega}=0}=\left\{\begin{array}{l}
F_{u x}  \tag{1.4}\\
F_{u y}
\end{array}\right\}=\left\{\begin{array}{l}
m_{u} r_{u} \omega^{2} \cos \theta \\
m_{u} r_{u} \omega^{2} \sin \theta
\end{array}\right\}
$$

### 1.2.1.2 Whirl and Critical Speeds

Rotors with flexible shafts experience whirling due to excitations such as unbalance. Whirl is defined as the rotation or orbital motion of the plane formed by the line of centres of the bearings and the bent centreline of the shaft. Unbalance causes whirling at a frequency equal to the rotor spin speed (synchronous). Whirl frequency may also be different from rotor spin speed (asynchronous). Asynchronous whirl may be due to an asynchronous excitation or may occur at a natural whirl frequency of the rotor system. Natural whirl frequencies are dependent on rotor spin speed due to the gyroscopic effects found in rotating structures (discussed in the next section section 1.2.1.3). Critical speeds are shaft spin speeds at which the synchronous whirl amplitude reaches a maximum, i.e. the spin speed of the rotor equals the natural whirl frequency of the rotor and resonance occurs (S.S. Rao, 2004).

An orbit plot is often used to illustrate the orbital motion of a whirling shaft. It is a two-dimensional graph usually generated by plotting the displacement signals from two mutually-perpendicular lateral planes (e.g. the horizontal and vertical planes of a horizontally orientated rotor system) against each other (one signal on the horizontal axis and the other on the vertical axis - refer to Figure 1.2 as an example). The time components of the signals are therefore not visible.


Figure 1.2-Orbit Plot

The trend in the design of rotating machinery, such as steam turbines used in power plants, is towards lighter rotors rotating at higher speeds for increased efficiency and lower costs (Mohiuddin et al., 1998). Whirl is particularly evident in rotor systems with heavy disks mounted on lightweight flexible shafts and since unbalance is usually the main cause of rotor vibration, it should always be considered when designing or modelling rotor systems.

### 1.2.1.3 Gyroscopic Effects

The gyroscopic effect (an intrinsic characteristic of gyroscopes) may be described by the behaviour of a spinning disk, such as the disk shown in Figure 1.3. The fundamental equation describing the behaviour of this disk is:

$$
\begin{equation*}
\boldsymbol{\tau}=\frac{d \boldsymbol{L}}{d t} \tag{1.5}
\end{equation*}
$$

where $\boldsymbol{\tau}$ is a torque vector, $\boldsymbol{L}$ is the angular momentum vector of the disk and $t$ is time. For rotation about a fixed spin axis, the angular momentum is given by:

$$
\begin{equation*}
L=I \omega \tag{1.6}
\end{equation*}
$$

where $I$ is the mass moment of inertia about the spin axis and vector $\omega$ is the angular velocity of spin (or spin speed). From eqn. (1.5), it follows that an input torque, $\boldsymbol{\tau}$, applied perpendicularly to the spin of the disk (therefore also perpendicular to its angular momentum $\boldsymbol{L}$ ) causes an output torque perpendicular to both the input torque and the spin axis - this is illustrated in Figure 1.3. Rotation caused by this output torque is known as precession (Oak, 2008). The angular velocity of precession, vector $\Omega$, can be obtained from:

$$
\begin{equation*}
\boldsymbol{\tau}=\boldsymbol{\Omega} \times \mathbf{L} \tag{1.7}
\end{equation*}
$$



Figure 1.3-The Gyroscopic Effect

Most modern rotors operate at spin speeds much higher than their first critical speed and sometimes even higher than their third critical speed (Rao \& Sharan, 1994). Since the gyroscopic
effect is dependent on rotor angular momentum, the influence of gyroscopic moments may be negligible on the dynamics of a system at low spin speeds and also on the dynamics of rotors with low mass moments of inertia about their lateral axes. Such a rotor responds virtually the same as it would if it were stationary. At higher spin speeds, however, gyroscopic effects may significantly influence the dynamics of a rotor (assuming the rotor has sufficient mass moment of inertia about its lateral axes) and cause deviations from the stationary system response. The gyroscopic effect causes coupling between the rotor response in orthogonal planes which, in turn, splits each natural whirl frequency of the system into a forward (whirling in the same direction as shaft rotation) and backward (whirling in the opposite direction of shaft rotation) frequency pair. The frequency response of a high speed rotor system is therefore speed-dependent (Movahhedy \& Mosaddegh, 2006). The forward whirling natural frequency usually increases (i.e. the effective stiffness of the system increases) with an increase in shaft spin speed while the backward whirling natural frequency decreases. In a certain mode of vibration, if a disk of a high speed rotor is mounted at a nodal point of the system (where there is no translation but precession corresponding to shaft slope changes) the disk will experience considerable gyroscopic effects. If a disk is located at an anti-nodal point (where there is no slope change but pure translation) the disk will experience no gyroscopic effects (Rao \& Sharan, 1994). Due to the distinct influence of gyroscopic effects on the dynamics of high speed rotors, it is important to include these effects when modelling such rotor systems.

A Campbell diagram illustrates the natural whirl frequencies of a rotor as a function of rotor spin speed and may be a useful tool for visualising the influence of gyroscopic effects on the natural whirl frequencies of a rotor. Figure 1.4 shows an example of such a diagram. The natural whirl frequencies at a specific spin speed are usually determined by performing a complex modes analysis (a method usually implemented for calculating damped natural frequencies and mode shapes). These natural whirl frequencies at different spin speeds are then combined to create a Campbell diagram. Critical speeds may be deduced from a Campbell diagram by adding a line on which spin speed equals whirl frequency - where this line crosses a natural whirl frequency line, a critical speed is located.


Figure 1.4 - Typical Campbell Diagram (Rao \& Sharan, 1994)

### 1.2.1.4 Fluid Film Bearings, Oil Whirl and Oil Whip

The two predominant types of fluid film bearings are hydrodynamic and hydrostatic bearings. Within these bearings, the two load-carrying surfaces (located on the bearing journal and housing) are separated by a film of lubricating fluid (usually oil). Within hydrodynamic bearings, this film is generated by the rotating journal or shaft which forces the fluid into a wedge-shaped area at a velocity high enough to generate the required pressure to separate the load-carrying surfaces. Hydrostatic bearings utilise pumps to generate the fluid pressure required to separate the loadcarrying surfaces. Unlike hydrodynamic bearings, hydrostatic bearings therefore do not require shaft rotation to form a separating film of fluid (Budynas \& Nisbett, 2008). Turbomachines are typically supported on oil-lubricated hydrodynamic journal bearings as these bearings are relatively simple to implement and provide sufficient load-carrying capabilities with minimal frictional losses at high spin speeds.

The forces within a fluid film bearing (due to the pressurized fluid) are nonlinear functions of journal displacement, velocity and spin speed. These bearings possess both direct and crosscoupled stiffness and damping properties due to the influence of journal displacement and velocity, in any direction, on the fluid pressure. The cross-coupled forces are capable of inserting energy into the system and if this energy exceeds the energy dissipated by the direct damping of the bearing, an unstable ${ }^{2}$ rotor motion may occur. The spin speed threshold for this type of instability depends on bearing dimensions, fluid viscosity, bearing load and a number of other parameters (Nelson, 2007). The fluid itself flows around the journal at an average speed of approximately half the journal surface speed. This fluid flow could therefore lead to the occurrence of oil whirl which is a sub-synchronous (below shaft spin speed) whirling motion at a frequency generally between 40 and $48 \%$ of the shaft frequency (usually occurs in lightly loaded hydrodynamic bearings). If the shaft speed is increased to approximately twice a critical speed of the rotor and the oil whirl frequency therefore coincides with the critical speed, the oil whirl motion "locks" onto the critical speed and the instability, oil whip, occurs.

Oil whip is caused by the wedge-shaped area (into which the oil is forced) moving around the journal at the same speed as the oil flow (i.e. half the spin speed). This results in an instant loss of hydrodynamic pressure. During oil whirl the journal centre is usually close to the housing centre and the bearing stiffness is therefore much lower than the shaft stiffness - the dynamics are therefore dominated by the bearing properties. During oil whip, the journal is close to or in contact with the housing wall and the bearing stiffness (from the housing) is therefore much higher than the shaft stiffness - the dynamics are then mainly dominated by the shaft mass and stiffness. In the latter case, the journal vibration amplitude is limited only by the bearing clearance while the shaft vibration amplitude may become very large since the excitation occurs close to a critical speed of the rotor (Nelson, 2007; Berry, 2005; De Castro et al., 2008; Holmes, 1980).

As an example to illustrate the oil whirl and oil whip phenomena, Figure 1.5 shows the Hilbert spectrum of a rotor run-up vibration signal. The method used to generate this spectrum is the Hilbert-Huang transform (HHT) - used for analysing nonlinear and non-stationary signals (Huang et al., 1998). The Hilbert spectrum in Figure 1.5 (horizontal axis: vibration frequency and vertical axis: rotor spin speed) clearly shows frequencies at which (synchronous) whirl due to unbalance, oil whirl and oil whip occur as a function of the shaft spin speed. The red-coloured unbalance whirl

[^1]region at approximately 2700 cycles per minute indicates resonance or a critical speed of the rotor. The oil whirl is seen to occur at approximately half the shaft spin speed up to just below twice the critical speed where it transforms into oil whip at a constant frequency, independent of shaft speed. It is clear that oil whirl and oil whip may have a pronounced effect on the dynamics of a rotor and should therefore form part of a model which strives to accurately represent a physical rotor system.


Figure 1.5 - Oil Whirl \& Oil Whip: Hilbert Spectrum Plot (Fan et al., 2011)

### 1.2.1.5 Rubbing and Dry Whip

Rubbing, where rotor-stator contact occurs, is usually caused by excessive rotor flexural (lateral) vibration. Within turbomachines, such as steam turbines, blades may come into contact with the outer casing due to rotor vibration and small clearances - the trend for the design of these rotors is towards smaller clearances for increased efficiency. Rubbing is typically characterised by two types of forces: impact (in the radial direction) and friction (in the tangential direction) between rotor and stator (Yungong et al., 2011). The friction component of rubbing can cause backward whirling. Rubbing may be partial (intermittent) or full (sustained rotor-stator contact), depending on the rotor vibration amplitude, stator stiffness, friction, etc. The stator stiffness has a significant effect on the amplitude of rotor vibration at which full rubbing is initiated (Dai et al., 2002).

The whirling motion of a rotor subject to partial rubbing may turn into dry whip at a higher frequency ${ }^{3}$ when full rubbing occurs (Dai et al., 2002). During oil whip in a fluid film bearing, the bearing journal may come into contact with the housing and also cause backward whirl and dry whip (Berry, 2005).

Rubbing is a highly nonlinear phenomenon which could lead to catastrophic failure and should therefore be considered when modelling rotor systems.

[^2]
### 1.2.1.6 Misalignment and Bearing Anisotropy

Other than unbalance, one of the most common sources of rotor vibration is misalignment. Shaft misalignment is encountered when an eccentricity exists between two connected shafts, usually due to improper machine assembly (Kr. Jalan \& Mohanty, 2009). The two basic types of misalignment are parallel and angular, as demonstrated in Figure 1.6. Generally, misalignment is a combination of these two types.

a) Parallel misalignment


Figure 1.6-Two Basic Types of Misalignment

Misalignment causes reaction forces and moments in couplings and bearings which may have both static and dynamic components. The effects of the dynamic components are often observed as vibrations at even multiples of the shaft speed ( $2 \times$ spin speed, $4 \times$ spin speed, etc.), mainly due to the universal joint effect (Xu \& Marangoni, 1994). Abnormal preloads on bearings, due to the static reaction force components, may cause an increase in bearing stiffness along the misalignment direction. This anisotropy in bearing stiffness causes elliptical orbits near the critical speeds of a rotor (which is typically associated with misalignment). For speeds just below a critical speed, the major axis of the elliptical orbit is usually parallel to the direction of misalignment (i.e. the static reaction force) while the major axis coincides with the direction of misalignment at just speeds above the critical speed. Although misalignment is a common rotor fault that may have a considerable influence on the dynamics of a rotor system, it is rarely encountered in rotor system models due to its complexity in modelling (Y.-S. Lee \& C.-W. Lee, 1999).

### 1.2.1.7 Damping

Damping in rotor systems may be classified as either internal or external. The former is primarily due to friction at component interfaces of the rotor. A small amount of internal damping may also be attributed to hysteretic damping inherent in the rotor material. The latter may arise from various sources such as bearings or squeeze film dampers - i.e. support structure components (Nelson, 2007; Xiang et al., 2009).

Due to the spin of a rotor, internal damping forces are tangential and increase with rotor spin speed. These forces insert power into the rotor during each revolution. If this energy inserted by internal damping exceeds the energy extracted from the system by external damping, instability occurs (Nelson, 2007; Chouksey et al., 2012). The influence of rotor damping may therefore be of great significance when investigating rotor instability thresholds.

### 1.2.1.8 Other Rotor Faults, Nonlinear Excitations and Rotor Dynamics

Steam flowing through turbine blades is another source of nonlinear excitation which may influence the stability of the rotor system, especially in ultra-supercritical steam turbines (Li et al.,
2011). The interactions between the aerodynamics, stiffness and inertia of blades may result in an instability (or self-excited vibration) know as (aerodynamic) blade flutter which could lead to highcycle fatigue (RPM Turbo, 2005).

Blade loss is a highly transient occurrence due to the large sudden imbalance and asymmetry resulting from the lost blade that usually causes severe rotor vibration, impact and fan-case interactions or rubbing (Mohiuddin et al., 1998).

Transient rotor response is encountered during critical speed transitions (such as rotor run-up and run-down). It is well known that increasing a rotor's angular acceleration when crossing through a critical speed will limit the amplitude of vibration at that critical speed (Genta \& Delprete, 1995) since the unbalance excitation does not spend "enough time" at the critical speed to allow sufficient transfer of energy into that vibration mode of the rotor.

### 1.2.2 Modelling Fluid Film Bearings (Fluid Film Lubrication Theory)

### 1.2.2.1 The Reynolds Equation

The most widely used fluid film lubrication theory for fluid film or journal bearings (which may be used to predict the behaviour of such bearings) is based on the Reynolds equation developed by Osborne Reynolds in 1886. The classical theory involves assuming that the lubricating fluid film is so thin compared to the bearing radius that the curvature may be neglected. Other assumptions are:

- The fluid may be described as continuous
- The Navier-Stokes equations hold
- The fluid is incompressible
- The viscosity is constant throughout the fluid film
- The flow within the film is laminar
- The fluid inertial forces are negligible

The Reynolds equation (in cylindrical coordinates) for a bearing in static equilibrium is given by:

$$
\begin{equation*}
\frac{1}{r^{2}} \frac{\partial}{\partial \theta}\left(\frac{h^{3}}{\mu} \frac{\partial p}{\partial \theta}\right)+\frac{\partial}{\partial z}\left(\frac{h^{3}}{\mu} \frac{\partial p}{\partial z}\right)=6 \omega \frac{\partial h}{\partial \theta} \tag{1.8}
\end{equation*}
$$

and for a moving journal (San Andrés, 2006):

$$
\begin{equation*}
\frac{1}{r^{2}} \frac{\partial}{\partial \theta}\left(\frac{h^{3}}{6 \mu} \frac{\partial p}{\partial \theta}\right)+\frac{\partial}{\partial z}\left(\frac{h^{3}}{6 \mu} \frac{\partial p}{\partial z}\right)=\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t} \tag{1.9}
\end{equation*}
$$

with angular coordinate $\theta$ as shown in Figure 1.7 and where $z$ is the coordinate in the axial direction (with $z=0$ at the middle of the bearing), $h$ is the film thickness, $r$ is the journal radius, $\mu$ is the fluid dynamic viscosity, $p$ is the pressure (a function of $\theta$ and $z$ ), $\omega$ is the journal spin speed and $t$ denotes time. The length of the bearing is denoted $l$ and the ends of the bearing are thus located at $z=-l / 2$ and $z=l / 2$. Figure 1.7 also shows additional Cartesian coordinates $x$ and $y$ where the centre of the coordinate system defined by these coordinates is located at the centre of the housing. These coordinates also specify the position of the journal centre relative to the housing.


Figure 1.7 - Journal Bearing Parameters

Eqn. (1.8) applies for static equilibrium of the bearing journal where the weight of the rotor at the journal bearing, denoted by $\boldsymbol{W}$, is usually balanced by the fluid film pressure (i.e. the forces generated by this pressure). The angle between the line of centres and $\boldsymbol{W}$ (or the $y$-axis), denoted as $\phi$, is known as the attitude angle and is given by:

$$
\begin{equation*}
\phi=\operatorname{atan} 2(x, y)-3 \pi / 2 \tag{1.10}
\end{equation*}
$$

where function $\operatorname{atan} 2(x, y)$ is given in Appendix H.1. Angle $\theta^{\prime}=\theta-\phi-\pi / 2$ is a coordinate which is equal to zero at the line of furthest separation between the journal and housing (i.e. at the line of centres). The bearing eccentricity $e$ is given by:

$$
\begin{equation*}
e=\sqrt{x^{2}+y^{2}} \tag{1.11}
\end{equation*}
$$

while the radial clearance $c$ is given by:

$$
\begin{equation*}
c=\frac{(D-d)}{2}=\frac{D}{2}-r \tag{1.12}
\end{equation*}
$$

where $D$ is the inner diameter of the housing and $d$ is the diameter of the journal.

Equation (1.9) is an elliptic nonhomogeneous second-order partial differential equation which has no known analytical solution and therefore needs to be solved numerically (Vignolo et al., 2011). Modifying and solving this equation for specific purposes is still an active area of on-going research. Analytical solutions may be obtained by making certain extreme approximations with regards to the bearing geometry: Ocvirk's (1952) 'short bearing approximation' assumes the bearing length is negligible compared to the radius and the circumferential pressure gradient $(\partial p / \partial \theta)$ is thus much smaller than the pressure gradient in the axial direction $(\partial p / \partial z)$. Therefore, the circumferential pressure gradient is set equal to zero $(\partial p / \partial \theta=0)$. This approximation is reasonably accurate for
small eccentricities and finite-length journal bearings with $l / d<1 / 4$ (Gunter et al., 1975; Vignolo et al., 2011). Sommerfeld's (1904) 'long bearing approximation' assumes infinite bearing length which means that side leakage of the lubricating fluid may be neglected. If side leakage is neglected, the pressure gradient in the axial direction is much smaller than the circumferential pressure gradient ( $\partial p / \partial z \ll \partial p / \partial \theta)$. The pressure gradient in the axial direction is, therefore, set equal to zero $(\partial p / \partial z=0)$.

### 1.2.2.2 Boundary Conditions

In order to solve the Reynolds equation for a specific bearing (a boundary value problem), pressure boundary conditions (BCs) - pertaining to the physical properties of the bearing - are required.

A common set of BCs is encountered when full fluid leakage is allowed at the edges of a journal bearing (also known as full end-leakage) and the fluid film pressure at these edges are assumed to be equal to atmospheric pressure. Therefore, when working with gauge pressures, the BCs at these edges may be written as:

$$
\begin{equation*}
p(\theta,-l / 2)=p(\theta, l / 2)=0 \tag{1.13}
\end{equation*}
$$

Cavitation usually occurs in hydrodynamic journal bearings. Although cavitation may be influenced by the gas and solid contents of the fluid (Gunter et al., 1975), the effect of cavitation is often dealt with by equating any negative pressures in the bearing to zero (atmospheric pressure). The locations of the cavitation boundaries for the short bearing approximation can therefore be found by setting the pressure equation equal to zero. For static equilibrium of the bearing $(\partial h / \partial t=0)$, this yields cavitation boundaries located at the lines of closest and furthest separation between the journal and housing - i.e. at $\theta^{\prime}=0$ and $\theta^{\prime}=\pi=180^{\circ}$ (San Andrés, 2006). The applicable cavitation BCs are known as Gümbel or half-Sommerfeld BCs and are given by:

$$
\begin{equation*}
p\left(\theta_{1}, z\right)=p\left(\theta_{2}, z\right)=0 \tag{1.14}
\end{equation*}
$$

where $\theta_{1}$ and $\theta_{2}$ are respectively the angular locations of the starting and break-down (or filmrupture) boundaries of the load-carrying (positive-pressure) region of the bearing. The addition of journal translational velocity results in a deviation from these boundary locations, but the size of the cavitation region still remains half of the bearing (i.e. $\theta_{2}=\theta_{1}+\pi$ ). These BCs and boundary locations are often also used for finite-length bearing fluid film pressure calculations.

Gümbel BCs are, however, not in complete agreement with experimental data and more realistic BCs are the so-called Reynolds or Swift-Stieber BCs (Szeri, 2011) where the pressure as well as its gradient is assumed to be zero at the break-down boundary. Unlike Gümbel boundaries, this boundary is curved for a finite bearing (Curzon, 1986) and its location and shape needs to be determined. This type of boundary is known as a 'free' or 'floating' boundary ${ }^{4}$. Along with these $B C s$, the pressure is usually set equal to zero at the starting boundary of the load-carrying region (the location of which is usually calculated from the short bearing approximation). These BCs may be expressed as follows...

[^3]\[

$$
\begin{gather*}
p\left(\theta_{1}, z\right)=0 \\
p\left(\theta_{2}(z), z\right)=\frac{\partial p}{\partial \theta}\left(\theta_{2}(z), z\right)=0 \tag{1.15}
\end{gather*}
$$
\]

Since $\theta_{2}$ is a curved boundary location, it is a function of $z$.
Since the journal surface is cylindrical, the pressure distribution around the journal is periodic. Furthermore, the pressure distribution is assumed to be continuous. Thus, the following pressure continuity BCs apply:

$$
\begin{align*}
p(\theta, z) & =p(\theta+2 \pi, z) \\
\frac{\partial p}{\partial \theta}(\theta, z) & =\frac{\partial p}{\partial \theta}(\theta+2 \pi, z) \tag{1.16}
\end{align*}
$$

### 1.2.2.3 Numerically Solving the Reynolds Equation

A relatively simple numerical method often employed to solve partial differential equations, such as the Reynolds equation, is the finite difference method. This method entails discretising the solution domain (over which the differential equation is to be solved) by replacing it with a finite number of points known as grid points - refer to Figure 1.8 for an example of a two-dimensional (2D) grid. These grid points are usually equally spaced along each independent variable or coordinate (for the 2D example: $i$ and $j$ are integers that denote the discrete grid points along independent variables $x$ and $y$ respectively and $\Delta x$ and $\Delta y$ are the constant spacing sizes between grid points along $x$ and $y$ respectively). Derivatives within the differential equation (and, when applicable, BCs ) at each grid point are approximated from linear combinations of function values (function $z=f(x, y) \rightarrow z_{i, j}=f\left(x_{i}, y_{j}\right)$ in the 2 D example - Figure 1.8 ) at surrounding grid points. This results in a system of linear equations, known as finite difference equations, which need to be solved simultaneously to determine approximate values of the function values at the grid points.


Figure 1.8-2D Grid

Three general types of derivative approximations are encountered: forward difference, backward difference and central difference approximations. These approximations can be derived from the Taylor series expansion theorem. Central difference approximations are the most accurate of the three (Saad, 2003). The function to be solved may be one-dimensional (one independent variable) or multi-dimensional (more than one independent variable). Central difference equations for both first and second order derivatives of a 2D function are included in Appendix H.2.

There are two general methods available for solving the finite difference equations from the Reynolds equation: direct and iterative methods. Gaussian elimination and matrix inversion are amongst the direct methods. When employing a computer to solve finite difference equations, direct methods require the storage of coefficients in sparse matrices which could easily become very large. As an example, consider a 2D grid with $N$ and $M$ points in the directions of the first and second independent variables respectively - the number of rows and columns of the coefficient matrix are each proportional to $(N M)^{2}$. Furthermore, direct methods require that the BCs be known on a rectangular domain with its sides parallel to the two independent variables. On the other hand, iterative methods are simpler to program and require less storage to implement on a computer. Iterative methods perform 'sweeps' through finite difference equations, computing new function values from previous or current surrounding function values until convergence is reached. These methods are also capable of handling free boundaries, such as boundaries subject to Reynolds BCs, with relative ease. Reynolds BCs are automatically satisfied (within the rectangular surface area) by equating all negative pressures from each iteration to zero before the next iteration. Gümbel BCs are automatically satisfied by equating all negative pressures to zero after the complete pressure distribution has been solved (Holmes, 1975; Singhal, 1981; Lin \& Wang, 1990).

The finite difference approximation of the Reynolds equation (where $p(\theta, z)$ is the function to be solved) may be written in the form (Singhal, 1981):

$$
\begin{equation*}
p_{i, j}=A_{1 i} p_{i-1, j}+A_{2 i} p_{i+1, j}+A_{3 i}\left(p_{i, j-1}+p_{i, j+1}\right)+A_{4 i} \tag{1.17}
\end{equation*}
$$

where coefficients $A_{1 i}, A_{2 i}, A_{3 i}$ and $A_{4 i}$ are functions of $\theta_{i}$. A simple iterative method that may be used to solve the pressures at each discrete point is the Jacobi method. To implement this method, initial pressure values are assumed and then new pressure values are repeatedly (i.e. iteratively) calculated at each grid point from the previous surrounding pressure values until convergence is reached. This is symbolically given as:

$$
\begin{equation*}
p_{i, j}^{(k+1)}=A_{1 i} p_{i-1, j}^{(k)}+A_{2 i} p_{i+1, j}^{(k)}+A_{3 i}\left(p_{i, j-1}^{(k)}+p_{i, j+1}^{(k)}\right)+A_{4 i} \tag{1.18}
\end{equation*}
$$

where integer $k+1$ refers to the new iteration and $k$ refers to the previous iteration. The convergence rate of this method may be improved by using the updated pressure values as soon as they have been calculated instead of using only pressure values from the previous iteration. This is known as the Gauss-Seidel method. Assuming that for each value of $i$, the pressures are calculated for all the values of $j$ in succession before moving to the next value of $i$ (or visa versa), the iteration formula may be written as:

$$
\begin{equation*}
p_{i, j}^{(k+1)}=A_{1 i} p_{i-1, j}^{(k+1)}+A_{2 i} p_{i+1, j}^{(k)}+A_{3 i}\left(p_{i, j-1}^{(k+1)}+p_{i, j+1}^{(k)}\right)+A_{4 i} \tag{1.19}
\end{equation*}
$$

With each iteration of the Gauss-Seidel method, the pressure values move closer to the final solution. This method is, however, conservative in the incremental quantity it allows the pressure to move towards the final solution. The successive over-relaxation (SOR) method improves upon the convergence rate of the Gauss-Seidel method by increasing this incremental quantity. The SOR method entails calculating the difference between the pressure value from the Gauss-Seidel method and the previous pressure value, multiplying this difference with a factor greater than one and then adding the result to the previous pressure value:

$$
\begin{align*}
& p_{i, j}^{(k+1)}=p_{i, j}+\alpha\left[p_{i, j}^{G S}-p_{i, j}^{(k)}\right] \\
&=p_{i, j}^{(k)}+ \alpha\left[A_{1 i} p_{i-1, j}^{(k+1)}+A_{2 i} p_{i+1, j}^{(k)}+A_{3 i}\left(p_{i, j-1}^{(k+1)}+p_{i, j+1}^{(k)}\right)\right.  \tag{1.20}\\
&\left.+A_{4 i}-p_{i, j}^{(k)}\right]
\end{align*}
$$

where $p_{i, j}^{G S}$ is the Gauss-Seidel updated pressure value and $\alpha$ is the over-relaxation factor and is usually between 1 and 2 (a value equal to 1 will yield the same convergence rate as the GaussSeidel method and a value equal to or greater than 2 will cause divergence). For any particular system of equations, a unique over-relaxation factor exists which provides the fastest rate of convergence. For the Reynolds equation, this optimal factor is given by (Singhal, 1981):

$$
\begin{equation*}
\alpha_{o p t}=2 \frac{1-\sqrt{1-B^{2}}}{B^{2}} \tag{1.21}
\end{equation*}
$$

with

$$
\begin{equation*}
B=1-\frac{\pi^{2}}{2} \frac{4+\left(\frac{\pi D}{L}\right)^{2}}{m^{2}+\left(\frac{\pi D n}{L}\right)^{2}} \tag{1.22}
\end{equation*}
$$

where $n$ and $m$ are the number of mesh divisions or sections (one less than the number of mesh points) in the directions of $\theta$ and $z$ respectively. For the definition of a complete mesh, the integers $i$ and $j$ therefore start at 0 and continue up to $n$ and $m$ respectively.

### 1.2.3 Modelling Rotor Systems (Background)

The first rotor model capable of demonstrating stable supercritical operation was proposed by August Föppl in 1895. His model was however published in a German civil engineering journal and went unnoticed by the rotor dynamic community of that time. In 1919 Henry Jeffcott published the same type of model, with added damping, in an English journal. The symmetric single degree of freedom rotor model consisted of a single rigid imbalanced disk that is mounted in the middle of a flexible shaft supported by rigid bearings at each end. Modified versions of this model have also appeared in literature, where the most common variation replaces the rigid bearings with flexible bearings. This type of simplified rotor model is referred to as a Jeffcott rotor and even though it is an oversimplification of real rotors (single rotor, excluding gyroscopic effects, etc.), it is often the starting point in the investigation of rotor-related phenomena (Nelson, 2007).

Ever since then, various rotor models and methods for solving these models have been developed. There are two major classes of rotor models, namely continuous and discretised models. The former are analytical models where rotor systems are treated as distributed parameter systems. The equations of motion (EOMs) of these systems are partial differential equations. The latter
model class involves the discretisation of a rotor system so that it is approximated by finite degrees of freedom (an example of this is the very popular finite element method). The EOMs of these systems are ordinary differential equations (Mohiuddin et al., 1998). The required dynamic information may be obtained from these models using various analytical (if possible) and numerical methods. The EOMs of discrete rotor systems (in the time domain) may be expressed in the following general matrix-vector form:

$$
\begin{equation*}
[M]\{\ddot{u}\}+([C]+\omega[G])\{\dot{u}\}+[K]\{u\}=\{F\} \tag{1.23}
\end{equation*}
$$

where $\{u\}$ is the displacement vector with a component for each degree of freedom (DOF) of the rotor system, $[M]$ is the mass matrix of the system, $[C]$ is the system damping matrix (which may contain internal and external damping coefficients), $[G]$ is the gyroscopic matrix of the rotor ( $\omega$ is the shaft spin speed), $[K]$ is the system stiffness matrix and $\{F\}$ is the external force vector, which may include unbalance and other excitation forces. If a bearing is modelled using linear stiffness and damping coefficients, these coefficients are usually included in $[C]$ and $[K]$. On the other hand, if a nonlinear force model is used for a bearing, these forces are usually included in $\{F\}$ as "external" forces. Vector $\{F\}$ may also include the rotor weight (which has a significant influence on fluid film forces in horizontal rotor systems).

The rotor system models found in literature include a wide variety of different rotors, rotor system components, bearing models, dynamic effects, faults, excitations and analysis techniques. Most new studies focus on the influence of only one particular feature or aspect on the dynamics of a rotor system. Generalised models, including a combination of different rotor dynamic aspects, are uncommon. The following paragraphs provide and discuss a few examples of these various studies.

The simplest bearing model used in rotor system models, assumes that the bearings act as rigid point constrains that resist any translation but allows for rotation. Simple flexible bearings are usually modelled with direct linear translational stiffness coefficients (bending within bearings is therefore neglected). As discussed before (refer to section 1.2.1.4), the direct and cross-coupled stiffness and damping forces of fluid film bearings are nonlinear functions of the journal displacement, velocity and spin speed. As a simplification, hydrodynamic bearings are often modelled using linear approximations of their stiffness and damping forces which are only accurate for very small displacements around the static equilibrium position of the journal. These linear models are defined in terms of four stiffness coefficients (two direct and two cross-coupled) and four damping coefficients (two direct and two cross-coupled). These eight coefficients are either derived from the Reynolds equation of lubrication (Dostal et al., 1974; J.S. Rao, 1996; Y.-H. Lin \& S.C. Lin, 2001; Roques et al., 2010; Chouksey et al., 2012) or are experimentally determined (Qiu \& Tieu, 1997; Tiwari \& Chakravarthy, 2006; Tiwari \& Chakravarthy, 2009). Nonlinear fluid film bearings may be modelled by solving the Reynolds equation - this is however usually much more computationally expensive than the use of linear approximations.

As stated before, Jeffcott rotor models are useful as a starting point for rotor dynamic investigations. J.S. Rao (1982) derived analytical expressions for the unbalance response of a Jeffcott rotor supported on hydrodynamic bearings to determine conditions for backward synchronous whirling. Linearized bearing coefficients, which are shaft spin speed dependent, were used in the analysis. Another study was conducted by J.S. Rao (1985) to investigate the instability of rotors mounted on fluid film bearings with negative cross-coupled stiffness coefficients by using a Jeffcott rotor model. Choy and Padovan (1987) numerically investigated the nonlinear transient
dynamics of rotor-casing rubbing using a Jeffcott rotor. The rubbing model includes radial elastic impact forces and tangential Coulomb friction forces. Different major stages/phases of rubbing were analysed, including rub initiation, the onset of backward whirling and steady state rubbing due to system damping. The nonlinear characteristics of rubbing on a Jeffcott rotor were also numerically investigated by Cao et al. (2011), but with fractional order damping included in the rotor model. Cheng et al. (2007) investigated the dynamic behaviour of a rotor-bearing-seal system subjected to nonlinear forces from fluid film bearings and seals. The short bearing approximation was used and the transient response was calculated with the fourth-order Runge-Kutta method.

Rigid rotors are sometimes used when the stiffness of the support structure is much lower than that of the rotor itself and the spin speed is sufficiently lower than the frequency of the lowest flexible mode of the rotor. The numerical modelling of an overhung rigid rotor in a single-aerostatic bearing (which employs the load-carrying capability of compressed air) was performed by Frew and Scheffer (2008). The main focus was on developing an aerostatic bearing model and modified Euler equations to build the numerical model of the entire system, including gyroscopic effects. Friswell et al. (2001) investigated the bifurcations in Campbell diagrams which may be produced by viscous damping in bearings. A rigid rotor on simple flexible supports was modelled, including gyroscopic effects.

Gyroscopic effects are occasionally excluded from rotor models due to the negligible effect it has on the dynamics of the specific problem investigated and/or due to the unwanted complexity that its inclusion may add. Sharan and J.S. Rao (1985) presented the use of the method of influence coefficients to determine the unbalance response of a rotor supported by linearized fluid film bearings. Gyroscopic effects were excluded based on previous work by Ahuja and Sharan (1984) which concluded that in the presence of heavy damping from the hydrodynamic bearings, the gyroscopic effects have a negligible influence on the response of the rotor.

Lee and Jei (1988) performed modal analyses and unbalance response analyses of a continuous rotor system. Lee and Jei argued that, although the solution techniques for continuous rotors are limited to relatively simple rotors, this method may yield more accurate results and essential information on the effects of parameters on system behaviour, which is often difficult to observe and investigate with discrete methods such as the finite element method. The model consists of a continuous Rayleigh shaft and includes gyroscopic effects and the influence of asymmetry in the BCs (the support structure) on the dynamics of the system. These same authors also developed a continuous rotor model to analyse a system with an asymmetric rotor (Jei \& Lee, 1992). The model consists of asymmetric Rayleigh shafts, asymmetric rigid disks and isotropic bearings. Abdul Azeez and Vakakis (1999) performed numerical and experimental analyses to determine the transient response of a continuous overhung rotor undergoing vibro-impacts due to a defective bearing.

Amongst the most prevalent discretisation methods are the transfer matrix method (TMM) and the finite element method (FEM). With the radical development of computer hardware and software over the past years, the computational time required for discretisation methods has been drastically reduced and these methods are easily applied to analyse complex rotor systems. These methods (especially the FEM) have therefore become very popular. Furthermore, with the trend towards faster and lighter turbomachines with smaller clearances, it is important to be able to accurately model these complex systems using discrete methods.

The TMM solves dynamic problems in the frequency domain. This method comprises a marching procedure which starts at the BCs on one side of a dynamic system, then marches along the structure to the other side of the system. The system state at a specific point is transferred between successive points through transfer matrices and the primary advantage of this method is that, unlike the FEM, it does not require storage and solving of large system matrices and vectors (Hsieh et al., 2006). Dostal et al. (1974) utilised the TMM to investigate the onset of instability due to (linearized) fluid film bearings and the effect of external damping in delaying the onset of this instability. Later (1977), these same authors investigated the effect of intermediate, external damping as a means to controlling the synchronous response of a shaft supported on fluid film bearings using the TMM. Hsieh et al. (2006) developed a modified TMM to analyse coupled lateral and torsional vibrations of a symmetric rotor system subjected to an external torque. Timoshenko beam elements were utilised to model the shaft (a formulation which includes gyroscopic effects, amongst others).

Ruhl and Booker (1972) are known to be the first researchers to utilise the FEM in the field of rotor dynamics. Padovan et al. (1984) extended the FEM to handle the highly nonlinear forces of fluid film bearings and squeeze film dampers. Implicit and explicit integration schemes are compared in this study. Xiang et al. (2009) formulated wavelet-based rotating shaft elements to model rotor systems which include gyroscopic moments, transverse shear deformations and internal viscous and hysteretic damping. The influences of internal rotor material damping and linearized fluid film bearing forces on the modal behaviour of a rotor system were studied by Chouksey et al. (2012). Genta (1988) described EOMs, based on the FEM and the use of complex coordinates, which are capable of accounting for asymmetry in either the rotor, the support structure or both. In the case where both the support structure and rotor are asymmetric, a series solution was developed to solve the EOMs. The model also includes gyroscopic effects and internal viscous and hysteretic damping. Both De Castro et al. (2008) and Li et al. (2011) developed finite element (FE) rotor system models including gyroscopic effects and nonlinear fluid film bearing forces and investigated the oil whirl and oil whip phenomena. Both research groups used the short bearing approximation to model the fluid film bearings. The latter research group (Li et al., 2011) included a nonlinear steam excitation force at the seals of the rotor system. The commercial finite element analysis (FEA) software package, MSC.Nastran, was utilised by Surial and Kaushal (1999) to perform a critical speed (modal) analysis as well as an unbalance response analysis of gas turbine engines. Gyroscopic effects were included (via the Rotor Dynamics capability of MSC.Nastran) and the rotors and casings were connected via linear spring elements (direct stiffness only).

When the dynamics of rotor systems are analysed, constant shaft speed is usually assumed. Genta and Delprete (1995) extended the usual mathematical FE models to include the transient effects of rotors with varying spin speed. Roques et al. (2010) constructed a FE model of a turbogenerator in order to investigate the spin speed transient response of the system with rotor-stator rubbing caused by a blade-off imbalance. The angular deceleration of the rotor due to the rubbing was assessed and the spin speed was therefore considered as an unknown in the problem.

### 1.2.4 Finite Element Modelling of Rotor Systems with Fluid Film Bearings

As discussed in the preceding section (section 1.2.3), the FEM has become a very popular method for modelling rotor systems. Rotor dynamic analyses with finite elements can easily become computationally intensive (especially nonlinear transient dynamic response analyses) and therefore
one-dimensional (1D) and scalar elements are often used, instead of two-dimensional (2D) and three-dimensional (3D) elements, to model rotor systems.

1D rotor system models with fluid film bearings typically consist of:

- Shafts modelled with beam elements that consider mass moment of inertia and shear deformation.
- Disks (assumed to be rigid) modelled with concentrated mass elements that capture the inertial properties of the disks.
- Fluid film bearings modelled with linear translational spring and damper elements.

A linearized fluid film bearing model typically requires linear spring and damper elements with both direct and cross-coupled stiffness and damping coefficients derived at the equilibrium position of the bearing under static loading and constant spin speed. Figure 1.9 schematically illustrates such a model.


Figure 1.9-Linear Fluid Film Bearing Model

The direct stiffness and damping coefficients are denoted $k_{x}, k_{y}, c_{x}$ and $c_{y}$ while the cross-coupled coefficients are denoted $k_{x y}, k_{y x}, c_{x y}$ and $c_{y x}$. The cross-coupled coefficients are usually not symmetric (i.e. $k_{x y} \neq k_{y x}$ and $c_{x y} \neq c_{y x}$ ). As stated previously (refer to section 1.2.3), these linear approximations are only accurate for small displacements around the equilibrium position and if larger displacements need to be investigated, nonlinear fluid film bearing models should rather be considered.

The base or foundation is sometimes included in the rotor system model as a system of springs, dampers and concentrated mass elements. Gyroscopic effects are often included as speeddependent skew-symmetric off-diagonal terms in the damping and stiffness matrices of the rotor system.

### 1.2.5 Rotor Dynamics in MSC.Patran and MSC.Nastran

### 1.2.5.1 Rotor Dynamic Modelling Capabilities

In recent years, MSC.Software incorporated a rotor dynamic analysis capability into the FEA solver MSC.Nastran ${ }^{5}$ which accounts for the dynamics involved in structures with rotating components. This product module, MSC.Nastran Rotor Dynamics, was developed with input from companies such as Boeing Commercial Airplane Group, Rolls Royce and NASA and uses standard solution sequences from MSC.Nastran (Lawrence, 2004).

The capabilities of the product module include:

- The simulation of:
- Unbalance response
- General nonsynchronous excitation
- Manoeuvre loads (aeronautical) with rotor-stator rubbing prediction
- Blade-out response (including rubbing)
- Systems with multiple rotors
- The calculation of natural whirl frequencies and critical speeds
- Inclusion of internal viscous, structural (hysteretic), Rayleigh and modal damping (and the destabilising effect due to internal damping)
- A nonlinear force model for squeeze film dampers

MSC.Nastran Rotor Dynamics can be used to model rotors with a constant spin speed (usually steady state) or with varying spin speed (usually transient) and supports the use of standard 1D, 2D and 3D elements from MSC.Nastran.

Complex modes (complex eigenfrequency) analysis can be performed to calculate the natural whirl frequencies of a rotor at a specific spin speed or to calculate the critical speeds of a rotor. Selecting the asynchronous option calculates natural whirl frequencies while the synchronous option calculates critical speeds. Campbell diagrams may also be generated using the asynchronous option.

Frequency response analysis can be performed to determine the response of a rotor to excitation that is synchronous with rotor spin speed (synchronous option) or to arbitrary excitation with the rotor spinning at a specific speed (asynchronous option).

Static analysis can be used to determine manoeuvre loads due to the yaw and pitch of an aircraft and predict rubbing of the rotor blades with the engine casing due to excessive deflection from these loads (MSC.Nastran 2004 Reference Manual).

Linear and nonlinear transient response analyses may be performed to determine the response of a rotor due to unbalance or to other static or dynamic loads (if nonlinear loads are applied to the rotor system, a nonlinear transient response analysis must be performed). Fan blade-out analysis (which involve contact and may be highly nonlinear) can be performed on jet engines by using an explicit solver (and a model with a fine mesh) to determine the system loads and damage during the fan blade-out event and then using an implicit solver (and a model with a coarse mesh) to

[^4]determine the subsequent response of the engine due to these loads (MSC.Nastran 2012 Demonstration Problems).

In order to perform analyses including rotor dynamics, the use of a number of Rotor Dynamics module-specific MSC.Nastran input file entries (that specify spin rate, unbalance, etc.) is required. MSC.Patran ${ }^{6}$ is capable of setting up these input files required by MSC.Nastran, including most of the Rotor Dynamics entries (pre-processing) and is then also able to perform post-processing of the results.

### 1.2.5.2 Rotor Dynamics Module Limitations

All rotor dynamic analyses are performed in a stationary coordinate system (stationary relative to the support structure of a rotor system). The rotors analysed are therefore required to be axisymmetric, i.e. symmetric about their spinning axes (excluding mass imbalance) ${ }^{7}$.

The spin axis of a rotor must be defined by collinear node (grid) points and the rotor must be modelled as either a line model (confined to the grid points on the spin axis) or a 3D model reduced to a line model using superelements (discussed in the next section). If static (Guyan) reduction is used, this means that calculated rotor displacements and stresses are only accurate on the spin axis of the rotor analysed (MSC.Nastran 2004 Reference Manual; MSC.Nastran 2012 Quick Reference Guide).

### 1.2.5.3 Input Files, Solution Sequences, Superelements and Rotor Dynamics Entries

This section summarizes, amongst others, a few important basic concepts/features of MSC.Nastran required for rotor dynamic analysis.

## * Input Files

The input files of MSC.Nastran are text-based files, usually with extensions *.bdf or *.dat. These files contain all the information needed by MSC.Nastran to perform FEAs on a FE model and are structured as follows (MSC.Nastran 2004 Reference Manual):

- NASTRAN statement (optional): Used only in special circumstances.
- File Management Section - FMS (optional): Provides database initialization, job identification and restart conditions. Also specifies external files, services, etc. to be used.
- Executive Control Section: Selects the type of solution to be performed (which solution sequence to be used) and general conditions under which to perform the analysis.

```
CEND (required delimiter - end of Executive Control Section)
```

- Case Control Section: Defines the subcase structure for the analysis, selects loads and constraints and requests specific data output formats. A problem may be divided into different subcases with different constraints and/or loads.

```
BEGIN BULK (required delimiter - beginning of Bulk Data Section)
```

- Bulk Data Section: Contains the model geometry, element connectivity, element properties, material properties, constraints, loads, etc.
ENDDATA (required delimiter - end of input file)

[^5]
## * Solution Sequences

Linear dynamic problems can be solved by MSC.Nastran using either a direct formulation or modal formulation. The direct formulation solves one second order differential equation for each DOF of the model while the modal formulation solves one decoupled/coupled modal equation for each mode considered in the analysis (the user may choose how many modes to be used in the analysis). Nonlinear transient response analyses are only performed using the direct formulation.

Basic solution sequences of MSC.Nastran often used in rotor dynamics problems are (MSC.Nastran 2004 Reference Manual):

- SOL 101 \& 103: linear static and normal modes (real eigenvalue) analyses respectively
- SOL 107 \& 110: direct and modal complex modes (complex eigenvalue) analyses respectively
- SOL 108 \& 110: direct and modal frequency response analyses respectively
- SOL 109 \& 112: direct and modal transient response analyses respectively
- SOL 129: direct nonlinear transient response analysis

A general nonlinear solution sequence SOL 400 was introduced (as a pre-release) in the 2004 release of MSC.Nastran which includes nonlinear static analysis and nonlinear (implicit) transient response analysis and all the linear sequences from statics, modes, frequency response and transient response. Future releases of SOL 400 could also include nonlinear buckling analysis and nonlinear normal modes analysis. SOL 400 offers a STEP Case Control command that allows the results from one analysis to be used as initial conditions for another analysis. SOL 400 may be used to simulate solution sequences such as SOL 129, but has additional integration schemes, nonlinear iteration algorithms and convergence criteria. Refer to the MD Nastran 2006 Release Guide or an MSC.Nastran Quick Reference Guide (from the year 2006 or later) for more information on these transient response analysis features.

## * Superelements

Superelements may be used to substructure a FE model where each superelement is processed independently. The processing of a superelement results in reduced matrices that represent the properties of the superelement where it is connected to the rest of the model (at its boundary). The reduced matrices from all the superelements of a model are assembled into the residual structure and the problem is solved. Thereafter, data recovery of each superelement is done (to expand the solution to the superelements) by reversing the reductions performed. The nodes at the boundary of a superelement are known as its exterior nodes while the rest of the nodes of the superelement are its interior nodes.

The default reduction method used is static (or Guyan) reduction. In static analysis, data recovery yields the exact solution while in dynamic analysis, only the stiffness matrix reduction is exact - the mass matrix and damping matrix reduction is approximate. A method known as component modal synthesis (CMS) may be utilised to improve these approximations (MSC.Nastran 2001 Superelement User's Guide).

## * Rotor Dynamics Entries

A list of the main Rotor Dynamics entries with a short description of each entry follows:

- RGYRO: A Case Control command that activates the rotor dynamic capability. For complex eigenvalue, frequency response and static analyses, this Case Control command selects the RGYRO Bulk Data entry to specify synchronous or asynchronous analysis, the reference rotor (required in systems with multiple rotors) and its spin speed. For transient response analysis, this entry selects the UNBALNC Bulk Data entry (described below) and if there is no UNBALNC entry, RGYRO may be set to YES to include gyroscopic effects in the analysis.
- ROTORG: A Bulk Data entry which specifies node (grid) points that compose the rotor line model.
- ROTORSE: A Bulk Data entry which specifies a superelement that composes a rotor model with its boundary as the rotor spin axis (line model). Both static reduction and component mode reduction (CMS) is supported when using this entry.
- RSPINR: A Bulk Data Entry which specifies the relative spin rates between rotors (relative to the reference rotor) for complex eigenvalue, frequency response and static analyses.
- RSPINT: A Bulk Data Entry which specifies rotor spin rates for transient response analysis.
- UNBALNC: A Bulk Data entry which specifies an unbalance load for a transient or frequency response analysis in terms of a rotating cylindrical coordinate system with its zaxis as the rotor spin axis as shown in Figure 1.10. The imbalance mass (MASS field on UNBALNC entry) is denoted $m$, the radial mass offset (ROFFSET field) is denoted $r_{\text {offset }}$ and the angular position of the mass (THETA field) is denoted $\theta$. Unbalance forces are calculated in a similar fashion as described in section 1.2.1.1.
- CAMPBELL: Case Control command that selects the CAMPBLL Bulk Data entry.
- CAMPBLL: A Bulk Data entry which activates the generation of a Campbell diagram in complex modes analysis.
- NLRSFD: A Bulk Data entry which invokes a nonlinear force model for a squeeze film damper in a nonlinear transient response or nonlinear frequency response analysis.

More detailed information on these entries can be found in an MSC.Nastran Quick Reference Guide (the 2012 version was consulted in this case).


Figure 1.10 - MSC. Nastran UNBALNC Entry Coordinate System \& Parameters

Rotor damping may be included as viscous, structural (hysteretic), Rayleigh or hybrid (modal) damping. The EOMs solved during rotor dynamic analyses are in the form of eqn. (1.23). Appropriate terms are included in the damping matrix for each of these damping types and circulation terms are also added to the stiffness matrix. Structural rotor damping may be specified on either of the RSPINR or RSPINT entries or as a material property. For frequency response analysis, damping forces are (by default) calculated as functions of the rotor spin speed. For complex eigenvalue and transient response analyses, damping forces are calculated at an average excitation frequency that needs to be specified by the user. Further information on the EOMs used in Rotor Dynamics and how rotor damping is included is available in the MD Nastran 2006 Release Guide or an MSC.Nastran Quick Reference Guide (2006 or later).

### 1.2.6 Model Updating with MSC.Nastran Design Sensitivity and Optimization

Results from a FE model may not match the test results from experiments performed on the modelled structure. This could be due to modelling uncertainties such as BCs, material properties, modelling assumptions, etc. Even though experimental results also contain uncertainties, these results are often assumed to be more reliable. FE models are therefore usually updated to minimise the difference between the modelled and experimental results (Heyns, 2011; Blakely, 1991). MSC.Nastran has a Design Optimization capability which may be used for model updating. This capability is based on design sensitivity analysis which computes the rates of change of responses (such as natural frequencies) due to changes in design/model parameters (such as material properties). These rates of change are used to improve the efficiency of an optimization task (MSC.Nastran 2012 Design Sensitivity and Optimization User’s Guide).

A model updating approach that incorporates uncertainties in both model parameters and experimental/test results is the Bayesian Parameter Estimation approach. The following error function, including weighting/confidence factors, is minimised during this approach:

$$
\begin{equation*}
E F=w t \sum_{k} w r_{k}\left(r t_{k}-r m_{k}\right)+w m \sum_{k} w p_{k}\left(m p_{k}-m p_{k i}\right) \tag{1.24}
\end{equation*}
$$

where $w t$ denotes the overall test results weighting factor, $w r_{k}$ denotes the weighting factor for response number $k, r t_{k}$ and $r m_{k}$ are the test and model responses respectively, $w m$ denotes the overall model weighting factor, $w p_{k}$ denotes the weighting factor for model parameter $k$ and $m p_{k}$ and $m p_{k i}$ are the current and initial model parameters respectively. This error function may be minimised using MSC.Nastran solution sequence SOL 200 where the error function equation is specified via the DEQATN Bulk Data entry.

### 1.2.7 User Defined Services in MSC.Nastran

MSC.Software's Simulation Component Architecture (SCA) is a framework that provides users with the software's simulation technology as reusable software components. This framework provides the capability for user applications to extend or customize simulation components via User Defined Services (UDSs). The UDS capability allows users to utilise their own subroutines or applications within an MSC.Nastran execution. Specifically, this capability may be used to incorporate a userdefined nonlinear journal bearing model (via NLRSFD nonlinear forces) into an MSC.Nastran model (refer to sections 3.4 and 3.5). Creating and using a UDS in MSC.Nastran involves setting up a build environment, building the service (in the form of a dynamic-link library or DLL) and then using the
service within the MSC.Nastran application at runtime (MSC Software Development Kit 2012 User's Guide; MSC.Nastran 2012 User Defined Services).

### 1.2.7.1 Build Environment

To build a UDS, a build environment is required which needs to be configured by the user (based on the required implementations).

Software required, in addition to MSC.Nastran, for Microsoft Windows:

- MSC software development kit (SDK)
- Microsoft Visual Studio (Standard or Express Edition, incl. C++ compiler) ${ }^{8}$
- Fortran compiler (optional if services are developed in FORTRAN)

The software versions that should be used specifically with MSC.Nastran 2012:

- MSC SDK 2012
- Microsoft Visual Studio 2010 (with .Net Framework 4)
- Intel Visual FORTRAN Compiler 12.0 (included in Intel Composer XE 2011)

The SCA build system is used to build SCA framework components. This system utilises the SCons utility to do the builds. SCons is an open source software build tool provided with the MSC SDK (customized for the SCA environment - implemented as a Python script and a set of modules).

The build environment requires a source tree (folder structure) which contains the source code for the components to be built. Build configuration files are required in the source tree to control the configuration of the build system. The three types of build configuration files are:

- SConstruct: the master configuration file - required in the root directory of the source tree
- SConopts: the build options file - required in the root directory of the source tree
- SConscript: configuration file required in every directory/folder to be processed

Furthermore, the build environment requires a service definition language (SDL) file and a component definition language file (CDL). These two files must be located in the source tree and correspond to one or more interface definition language (IDL) files provided with the installation of MSC.Nastran.

Users may provide their own subroutines in either FORTRAN or C++ code. For implementation in FORTRAN, a user may utilise template files provided with the installation of MSC.Nastran (directory: <MSC.Software base directory> ${ }^{\text {MSC_Nastran|<version>|<version> }}$ |nast|services) and modify the F file (*.F) to call a FORTRAN user subroutine (*.f or *.f90). For implementation in C++, the user may also utilise the template files or generate the required source code skeleton files with the genskeleton utility provided with the MSC SDK and modify the generated CPP file (*.cpp). Refer to the MSC.Nastran 2012 User Defined Services and MSC Software Development Kit 2012 User's Guides for more detailed information on the build system and build environment.

### 1.2.7.2 Building the Service

The SCons utility may be run from the root directory of the source tree to build the UDS. If the PATH environment variable has been modified to include the directory of the SCons utility

[^6](<MSC.Software base directory> $|S D K|<v e r s i o n>\mid T o o l s)$, the SCons utility may be run by simply entering the "scons" command in the Windows Command Prompt (after changing the directory to the root of the source tree). When executed, the build system compiles and links all necessary files and then creates the DLL (along with other temporary files) which may then accessed and used by MSC.Nastran. In case the compiled service needs to be modified, the SCons utility may simply be re-executed after the required modifications have been made to the source files - the build system then automatically updates the required files and replaces the DLL.

### 1.2.7.3 Using the Service

In order to access the built UDS, the location of the DLL, service catalogue and service resource directory needs to be defined. This is done by modifying the following environment variables:

- SCA_LIBRARY_PATH: points to the location of the DLL
- SCA_SERVICE_CATALOG: points to the location of the SCA service catalogue (SCAServiceCatalog.xml)
- SCA_RESOURCE_DIR: points to the location of the SCA resource directory

The MSC.Nastran CONNECT SERVICE File Management statement is required to identify and connect to the UDS within MSC.Nastran. To trigger a specific service implementation, an appropriate Bulk Data entry needs to be associated with the group name specified in the CONNECT FMS statement. For example, the NLRSFD Bulk Data entry has additional parameters for the use of an external NLRSFD user subroutine (refer to an MSC.Nastran Quick Reference Guide for more detailed information on these parameters - the 2012 version was consulted in this case). Figure 1.11 shows a flow diagram of how a NLRSFD UDS is processed by the MSC.Nastran application during runtime. The NLRSFD entry is processed during every iteration of a transient response analysis.


Figure 1.11 - Flow Diagram of NLRSFD UDS Processing within the MSC. Nastran Application

### 1.3 Scope of Work

This dissertation entails the development of numerical models of an experimental rotor system with one hydrodynamic journal bearing and concerns only the lateral (bending) dynamics of the rotor system. The following rotor system features, dynamics and faults are included in the models:

- Gyroscopic effects
- Unbalance forces
- Nonlinear hydrodynamic journal bearing forces (capable of inducing oil whirl and oil whip)
- Rotor-stator rubbing
- Rotor and support structure damping
- Bearing anisotropy (which may be the results of misalignment)
- Rotor spin speed acceleration (run-up)

The hydrodynamic journal bearing is modelled by solving the Reynolds equation. This nonlinear force model has been developed for a finite-length journal bearing and includes the effects of journal velocity and fluid cavitation.

The work required for this study may be broken down as follows:

* Experimental rotor system setup, characterisation and analysis:
- If required, perform repairs to the existing experimental rotor system.
- Perform an experiment(s) to characterise individual components of this rotor system.
- Set up this rotor system to investigate artificially induced faults (such as unbalance).
- Equip the rotor system with eddy current proximity probes and any other measuring equipment that may be required to measure vibrations caused by faults.
- Conduct an experimental analysis to investigate the effects of artificially induced faults on the response of the rotor system.
* Develop a numerical model of the experimental rotor system:
- Construct a numerical model(s) of the experimental rotor system based on measured geometry, assumed materials, assumed lubricant properties (for the journal bearing) and experimentally determined component dynamic characteristics. The rotor and its support structure are to be modelled using FEA software including a nonlinear journal bearing model (which may need to be modelled with different software).
- Apply simulated faults to this model and determine its dynamic response under the influence of these faults.
- Validate and update the rotor system models.


### 1.4 Overview of Chapters

Chapter two begins by describing the experimental setup of the laboratory-scale rotor system investigated, including its components, layout and dimensions. The rotor is supported by two different bearings: a bush bearing and a cylindrical hydrodynamic journal bearing. The coordinate systems used throughout this dissertation for the rotor system are also defined. Before the experimental rotor system could be modelled, certain components of the system had to be characterised. Characterisation of the rotor itself (excluding the support structure) is described whereby its material densities were approximated from the weights of the rotor components and its elasticity properties (moduli of elasticity) were approximated from the assumed materials and results of an experimental modal analysis performed on the rotor. Next, the characterisation of the bush bearing is described whereby a custom experiment was performed to approximate linear dynamic stiffness and damping coefficients for lateral translation within the bush bearing. Custom experiments were also performed to approximate both lateral and torsional linear dynamic stiffness and damping coefficients for the coupling. Finally, the geometric properties of the journal bearing are specified and the dynamic viscosity of its lubricating fluid approximated.

Chapter three entails the numerical modelling of the experimental rotor system. Both 1D and 3D FE rotor system models were developed (using MSC.Nastran). A nonlinear hydrodynamic journal bearing model was developed separately from the FE rotor system models (since the version of MSC.Nastran used does not have a built-in cylindrical hydrodynamic journal bearing model) after which it was integrated into these models. For this and other reasons, this chapter is structured into five main sections:

- FE rotor models (without the support structure: free-free)
- FE rotor system models with linear bearings
- Journal bearing model
- Incorporation of the journal bearing model into the FE rotor system models
- FE rotor system models with nonlinear journal bearings

Model updating performed on the free-free rotor models using the natural frequencies determined from the experimental modal analysis is described. The rotor dynamic analyses performed on the rotor system models with linear bearings (by employing the Rotor Dynamics capability of MSC.Nastran) are discussed, including how these models were constructed, how the analyses were setup, MATLAB programs which were written for post-processing and signal processing of the results, etc. These rotor dynamic analyses include critical speed analysis, Campbell diagram generation, frequency response analysis and nonlinear transient response analysis. Furthermore, the steps required to include gravity loading (required for nonlinear journal bearing models), rotorstator rubbing and bearing anisotropy into these models were described, along with the effects that these features have on the dynamics of the rotor system models. The third main subsection describes the development of a general cylindrical finite-length hydrodynamic journal bearing model whereby the pressure distribution is solved by employing the finite difference numerical method and the resulting fluid film pressure distribution is numerically integrated over the journal surface to obtain the fluid film forces. MATLAB programs written to implement this model are also briefly discussed. The requirements and model modifications for incorporating the developed journal bearing model into the FE rotor system models as a user-defined FORTRAN subroutine are
described after which the nonlinear transient dynamic response analyses performed on the rotor system models with the integrated nonlinear journal bearing model are discussed.

The experimental analyses performed on the laboratory-scale rotor system with artificially induced faults are described in chapter four. Specifically, the transient dynamic response of the rotor system due to unbalance during run-up is investigated as well as the rotor system response during rotor-stator rubbing with a flat stator surface. MATLAB programs written to perform runout compensation on the measured vibration signals (using runout signals measured during slow-roll) are briefly discussed.

Chapter five compares the results from the numerical rotor system models to the results from the experimental analysis performed on the rotor system and comments on the capabilities of the numerical models to capture the required dynamics of such a rotor system.

Chapter six provides concluding remarks regarding the study and its results and also provides recommendations for possible future work.

## 2 Experimental Rotor System Setup and Characterisation

In this chapter, the experimental (laboratory-scale) rotor system setup is described and the characterisation of individual components of the system is discussed.

### 2.1 Experimental Rotor System

### 2.1.1 Bently Nevada Rotor Kit

Figure 2.1 shows the Bently Nevada Rotor Kit which includes a rotor, a motor ( 75 W DC; maximum speed: 10000 rpm ), a flexible coupling, an iolite bronze bush bearing, a cylindrical hydrodynamic journal bearing (including an oil-circulator and -sump) as well as ports for eddy current proximity probes. For the purposes of this study, the rotor (rotating part of the system) is considered to consist of the driven shaft (excluding the motor/driving shaft), two disks, the journal (rotating part of the journal bearing) and the driven half of the coupling (the coupling half connected to the motor shaft is excluded) while the rest of the components are considered to form part of the support structure. The rotor and support structure are collectively referred to as the rotor system.


Figure 2.1 - Experimental Rotor System: Bently Nevada Rotor Kit

Since the aim of the Eskom project is to develop data processing techniques for proximity probe data measured on turbomachinery, it is essential that the experimental rotor system used to develop these data processing techniques possesses certain features, dynamic characteristics and faults (which may be artificially induced) typically encountered in turbomachinery. The Bently Nevada rotor kit is therefore a fitting experimental rotor system since it includes a rotor comprising heavy disks mounted on a flexible shaft (as is typically the case with turbines) which is supported

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems
on a hydrodynamic journal bearing (also typically found in turbomachinery). Furthermore, the following faults may be artificially induced on this rotor system:

- Unbalance
- Fluid-induced instabilities ${ }^{9}$
- Rotor-stator rubbing
- Slight shaft misalignment


### 2.1.2 Rotor System Dimensions and Coordinate Systems

Figure 2.2 shows important dimensions of the assembled rotor system in its 'undeformed' shape or state, where the spin axis of the rotor coincides with the line of centres of the bush and journal bearings. More detailed dimensions (in the form of detail drawings) are provided in Appendix A.1. The disks and bush bearing were arbitrarily positioned along the shaft.


Figure 2.2 - Rotor System Dimensions

Figure 2.2 also shows the stationary global coordinate system used during the analysis of this rotor system. The $x$-axis of this Cartesian coordinate system coincides with the line of centres of the bush and journal bearings (also the spin axis of the undeformed rotor) and the $y z$-plane coincides with

[^7]the shaft shoulder near the coupling. Since the rotor is not completely axisymmetric, it is also useful to define a local rotating coordinate system for the rotor (which rotates with the rotor): the $x$-axis and origin of this local coordinate system coincides with that of the global coordinate system while the $y$-axis is parallel with the centrelines of the setscrews that keep the disks and coupling in position on the shaft. Therefore, when the rotor is positioned as in Figure 2.2, the local and global coordinate systems coincide. The $z$ - and $y$-directions of the global coordinate system are also referred to in this dissertation as the horizontal and vertical directions respectively. The $x$-direction of both the local and global coordinate systems is referred to as the axial direction. The global and local $y$ - and $z$-directions are sometimes referred to as lateral directions.

### 2.2 Rotor Characterisation

In this section, the characterisation of the rotor is discussed whereby its material densities are approximated and its natural frequencies (which were later used to approximate its material elastic properties) are determined.

### 2.2.1 Mass Densities

In order to calculate approximate material mass densities of the rotor components, the rotor was weighed on a scale. The setscrews (which keep the disks and coupling in position) are assumed have the density of typical carbon steel (i.e. $7800 \mathrm{~kg} / \mathrm{m}^{3}$ ). Since the coupling material seems different from the material of the rest of the rotor, the coupling was weighed separately from the rest of the rotor (i.e. the rest of the rotor was assumed to be made of the same material). The measured masses are given in Table 2.1.

Table 2.1 - Measured Rotor Masses

| Rotor Section | Mass $[g]$ |
| :--- | :---: |
| Coupling (incl. setscrew) | 65.4 |
| Rotor without coupling (incl. setscrews) | 1957 |
| Entire rotor: | $\mathbf{2 0 2 2 . 4}$ |

A solid model of the rotor system was created in SolidWorks (based on rotor system dimensions measured by hand) to aid in determining approximate material densities for the rotor (the detailed drawings in Appendix A. 1 were created from this solid model). Assuming that the rotor materials are isotropic and have uniform density, the Mass Properties tool of SolidWorks could be utilised to determine the approximate densities of the rotor components. First, the coupling with its setscrew was considered as shown in Figure 2.3: the density of the coupling material was adjusted (while keeping the setscrew density at $7800 \mathrm{~kg} / \mathrm{m}^{3}$ ) on a trial-and-error basis until the calculated mass of the coupling with its setscrew matched the mass in Table 2.1. Next, the entire rotor was considered as shown in Figure 2.4: the density of the shaft, disks and journal (assumed to be the same for all these components) was adjusted until the calculated mass of the rotor matched the mass in Table 2.1.


Figure 2.3 - Solid Model of Coupling with Setscrew


Figure 2.4 - Solid Model of Entire Rotor

Table 2.2 lists the resulting approximate component densities as well as assumed materials (based on the densities and visual appearances of the components).

Table 2.2 - Approximate Rotor Material Densities

| Rotor Component(s) | Material | Mass Density $\left[\mathrm{kg} / \mathrm{m}^{3}\right]$ |
| :--- | :---: | :---: |
| Coupling | Iron | 6206 |
| Shafts, Disks \& Journal | Steel | 7897 |
| Setscrews | Steel | 7800 |

For these material densities, the centre of gravity (C.G.) coordinates (in the global coordinate system) were determined (with the SolidWorks Mass Properties tool) to be (239.1,0,0) mm .

### 2.2.2 Experimental Modal Analysis

An experimental modal analysis was conducted on the approximately unconstrained (free-free) rotor to obtain its natural frequencies and mode shapes which were used to validate and update the numerical (FE) rotor models (refer to sections 3.1.1.2 and 3.1.2.2).

### 2.2.2.1 Experimental Setup

Figure 2.5 shows the rotor suspended from a steel frame by highly flexible rubber bands and supported in the global horizontal direction by soft sponge beams (which serve the purpose of providing a measure of flexible constraint that keeps the rotor from swinging "freely" under gravity
if excited in the horizontal or axial direction). Figure 2.5 also shows the modal hammer used to excite the rotor and the Polytec PSV 400 scanning laser vibrometer used to measure the vibration response. The Polytec data acquisition system recorded both the excitation force from the modal hammer and the vibration response of the rotor measured by the laser vibrometer at various points along the axis of the rotor. This system also includes a computer with software (Polytec PSV 8.8) that controls the operation of the laser, calculates frequency response functions (FRFs) from recorded data and calculates and displays the operational deflection shapes (ODSs) of the measured structure.


Figure 2.5 - Experimental Modal Analysis Setup

### 2.2.2.2 Analysis

The rubber bands and sponge beams have negligible stiffness compared to the rotor and the modes resulting from this experiment may therefore be assumed to represent the free-free modes of the rotor. Since the rotor is not completely axisymmetric, it exhibits resonant frequencies in the local $y$-direction of the rotor that differ from the resonant frequencies in the local $z$-direction. The modes of the rotor were therefore analysed separately for excitation and response in the local $y$ and $z$-directions of the rotor. Figure 2.6 shows the measurement points along the spin axis of the rotor that were considered during the experimental modal analysis. The assumption is made that the response on the outer surface of the rotor, for a specific axial location or coordinate, accurately represents the response at the spin axis (centreline) of the rotor ${ }^{10}$.

[^8]

Figure 2.6-Experimental Modal Analysis Measurement Points Coordinates

Two different types of scanning measurements were performed: for excitation and response in the local $z$-direction (referred to here as the horizontal direction) and the local $y$-direction (referred to here as the vertical direction). For both the horizontal and vertical scans, the rotor was excited at point 11 in the horizontal and vertical directions respectively, while the response was measured at points 1 to 19 for the horizontal scan and points 2 to 19 for the vertical scan (point 1 does not fall on the rotor in the vertical direction - refer to Figure 2.6). During the scanning measurements, the Polytec PSV software calculated an approximate FRF at each of the measurement points by performing complex averaging of FRFs (determined using the $H_{1}$ FRF estimator) from five measurements per point. Other signal recording and processing settings that were used during the analysis are given in Table 2.3.

Table 2.3 - Experimental Modal Analysis Laser Vibrometer Settings

| Channels | Range | Sensitivity |
| :--- | :---: | :---: | :---: |
| Vibrometer | 10 V | $50 \mathrm{~mm} / \mathrm{s} / \mathrm{V}$ |
| Modal Hammer | 500 mV | $2.15 \mathrm{mV} / \mathrm{N}$ |
| Frequency | 2 kHz |  |
| Bandwidth | 6400 |  |
| FFT Lines | 5.12 kHz |  |
| Sampling Frequency | $312.5 \mathrm{mHz}(T=3.2 \mathrm{~s})$ |  |
| Frequency Resolution | Type | Shape Factor |
| Windowing | Exponential | 4 |
| Vibrometer | Force | 5 |
| Modal Hammer |  |  |

The FRFs resulting from the scanning measurements were stored in SVD (*.svd) file format and a generic MATLAB program PolytecScanFFTPlot.m (included in Appendix B.5) was written to load, plot and store these FRFs. This program also calculates the average of the loaded FRFs which may be used as a mode indicator function. This program requires an appropriate version of Polytec File Access ${ }^{11}$ (PFA) software (that corresponds with the version of Polytec PSV or Vibsoft used to acquire the measured data) and custom MATLAB functions GetPointData.m, GetFFTAcqProps.m, GetXYZCoordinates.m and GetVideolmage.m (installed with the PFA software) to be in the working directory. PolytecScanFFTPlot.m plots the FRF for each point and then (optionally) stores this data in ASCII (*.asc) file format, which may be used in OROS Modal software - the capabilities of which include animating the ODSs of a structure and performing a complex FRF fit to estimate the modal properties of a structure.

### 2.2.2.3 Results

The resulting FRFs for point 11 in the rotor local $y$ - and $z$-directions are shown in Figure 2.7 and Figure 2.8 respectively. The rest of the FRFs are included in Appendix F.1. Figure 2.9 and Figure 2.10 show the average FRFs for all measurement points (as mode identification functions) for the vertical and horizontal directions respectively.


Figure 2.7 - Free-Free Rotor Point 11 Vertical (y-Direction) FRF

[^9]

Figure 2.8 - Free-Free Rotor Point 11 Horizontal (z-Direction) FRF


Figure 2.9 - Free-Free Rotor Average Vertical FRF


Figure 2.10 - Free-Free Rotor Average Horizontal FRF

The flexural natural frequencies identified from Figure 2.9 and Figure 2.10 are given in Table 2.4. The fourth natural frequency in the $z$-direction is not very noticeable - this may be due to the orientation of the disk setscrews (parallel to the local $y$-axis) which do not provide resistance against rotation of the disks in the local $y$-direction of the rotor.

Table 2.4 - Free-Free Rotor Experimental Modal Analysis Natural Frequencies

| Mode | Horizontal Natural <br> Frequency $[\mathrm{Hz}]$ | Vertical Natural <br> Frequency $[\mathrm{Hz}]$ |
| :---: | :---: | :---: |
| 1 | 84.6 | 85 |
| 2 | 188.4 | 193 |
| 3 | 447.5 | 465.5 |
| 4 | 660.6 | 740 |
| 5 | 859.1 | 930.6 |

### 2.2.2.4 Elastic Moduli

Since the shaft, disks and journal are assumed to consist of the same material (carbon steel), these components are also assumed to have the same elastic modulus while the coupling is assumed to have a different elastic modulus since it consists of a different material (iron). As initial guesses for the numerical rotor models (discussed in chapter 3 ), the elastic moduli of the coupling and the rest of the rotor (shaft, disks and journal) were set to 120 GPa (the approximate elastic modulus of iron) and 210 GPa (the approximate elastic modulus of steel) respectively. Model updating was performed on the FE rotor models (refer to sections 3.1.1.2 and 3.1.2.2) whereby these initial values were updated to minimise the difference between the lateral natural frequencies from these models and the experimental modal analysis (i.e. the frequencies listed in Table 2.4). The mode shapes from the FE rotor models were also visually compared to the ODSs of the experimental rotor at its natural frequencies and were seen to be in good agreement.

### 2.3 Support Structure Characterisation

The support structure is characterised by determining linear approximations for the stiffness and damping coefficients (assuming viscous damping) of the bush bearing and coupling. Characteristics of the journal bearing are also provided.

### 2.3.1 Bush Bearing Characterisation

As part of the bush bearing, there is an O-ring located between the bush and the housing. Loads applied to the bush are therefore transferred to the housing via this O-ring. The O-ring consists of rubber (most likely a synthetic rubber). The stiffness and damping characteristics of rubber are nonlinear and depend on the magnitude and frequency of the load applied to it (amongst others). The approximate linear dynamic stiffness and damping coefficients of the bush bearing should therefore be determined more or less at the operating load of the bearing. The step response of the bush bearing was investigated, from which linear approximations of both the dynamic stiffness and viscous damping coefficients could be found.

### 2.3.1.1 Experimental Setup

Figure 2.11 shows a schematic of the bush bearing characterisation experimental setup while Figure 2.12 shows photos of this setup. Instead of attempting to balance a loaded shaft on one bearing, two bush bearings (the second one also from the rotor kit) were used to support the shaft. The bearings were mounted on the base in an upside-down orientation and the shaft and the shaft load were centred between the bearings to apply the same load to both bearings. The step input load was suspended from a hook by a fishing line. Two linear variable displacement transducers (LVDTs), mounted on the rotor kit base, were utilised to measure the displacements of the shaft.

The measurements from these transducers were digitised with an HBM Spider8 data acquisition system and stored on a computer in text file format (tab delimited). The bearings were denoted bearings 1 and 2 (where bearing 1 was the bearing to be characterised) and the LVDTs were denoted LVDT A and LVDT B, positioned next to bearings 1 and 2 respectively.


Figure 2.11 - Bush Bearing Characterisation Experimental Setup Schematic


Figure 2.12 - Bush Bearing Characterisation Experimental Setup

### 2.3.1.2 Analysis and Results

The base, bearing pedestals, shaft and static load weight hanger (as configured in this setup) were assumed to be rigid compared to the O-rings in the bush bearings. The displacement responses measured by the LVDTs are thus assumed to result only from the deformation of the O-rings and bushes. The components that were free to move/vibrate (relative to the base) as the bearings deform, due to the step input load, are indicated in blue in Figure 2.11 - these components also comprise the static load applied to the bearings (although the mass of the bushings were excluded from calculations). The step input load was provided to the bearings by severing the fishing line with a flame. The step response of the system was recorded and investigated for four different static load cases. The total static load and vibrating mass of the system were varied by using four different sets of weights, as listed in Table 2.5. The shaft, static load weight hanger and the hook with a magnet weighed $67.2 g, 51 g$ and $39.2 g$ respectively (157.4 $g$ in total).

Table 2.5 - Bush Bearing Characterisation Static Load Cases

| Load <br> Case | Static Load <br> Weight(s) $[\mathrm{kg}]$ | Total Vibrating <br> Mass $[\mathrm{kg}]$ | Total Static <br> Load/Bearing $[\mathrm{N}]$ |
| :---: | :---: | :---: | :---: |
| 1 | 0.5 | 0.6574 | 3.225 |
| 2 | 1 | 1.1574 | 5.677 |
| 3 | 1.5 | 1.6574 | 8.130 |
| 4 | 2 | 2.1574 | 10.582 |

A MATLAB program BushBearingChar.m (included in Appendix B.1) was written to load the raw recorded LVDT signals (stored in text file format), convert these signals from millivolts ( mV ) to millimetres ( mm ) using sensitivities of -40.098 and $-40.589 \mathrm{mV} / \mathrm{mm}$ for the LVDTs (determined during calibration of these sensors), set the displacements before the step response equal to zero and plot the resulting signals. The TXT filenames need to adhere to a naming convention (including
the size of the static load weight and measurement/run number - specified by the user of the program in the code itself ${ }^{12}$ ) that is used when loading the files.

Figure 2.13 shows the displacement signals measured by the LVDTs (from 3 to $4.5 s$ ) for the first recorded measurement of load case 4 (used as an example for discussion throughout this section) .


Figure 2.13 - Bush Bearing Characterisation Step Response of LVDTs (Load Case 4, Run 1)

A slight difference is noted between the displacement magnitudes measured by the two LVDTs which indicates a difference in the dynamic stiffness properties of the two bearings. Assuming the shaft is straight and rigid, the displacements at the 'acting centres ${ }^{13}$ of the bearings may be calculated by interpolation from the LVDT displacements:

$$
\begin{align*}
& x_{1}(t)=\frac{l_{A, 1}}{l_{A, 1}+l_{1,2}+l_{B, 2}}\left[x_{B}(t)-x_{A}(t)\right]-x_{A}(t)  \tag{2.1}\\
& x_{2}(t)=\frac{l_{B, 2}}{l_{A, 1}+l_{1,2}+l_{B, 2}}\left[x_{A}(t)-x_{B}(t)\right]-x_{B}(t)
\end{align*}
$$

where $x_{1}$ and $x_{2}$ are the displacements at the centres of bearings 1 and 2 respectively, $x_{A}$ and $x_{B}$ are the displacements measured by LVDTs A and B respectively, $l_{A, 1}$ is the horizontal distance between LVDT A and the centre of bearing $1, l_{1,2}$ is the horizontal distance between the centres of bearings 1 and 2 and $l_{B, 2}$ is the horizontal distance between bearing 2 and LVDT B. The values of $l_{A, 1}, l_{1,2}$ and $l_{B, 2}$ are $26 \mathrm{~mm}, 57 \mathrm{~mm}$ and 28 mm respectively (calculated from the dimensions shown in Figure 2.11). BushBearingChar.m performs these interpolations and the resulting bearing displacements (between 3 and $4 s$ ) are shown in Figure 2.14.

[^10]

Figure 2.14 - Bush Bearing Characterisation Step Response of Bearings (Load Case 4, Run 1)

Another MATLAB program BushBearingCharSP.m (also included in Appendix B.1) was written to perform further signal processing on the data initially processed with BushBearingChar.m. To remain within a relevant frequency range, this program resamples the bearing signals if a sampling frequency of more than 1600 Hz is detected (different sampling frequencies were used for different measurements). The bearing signals are then cropped (with user-defined limits specified in the code itself) to extract only the step response sections of the signals for further analysis. Next, the discrete Fourier transforms (DFTs) of these signals are determined (computed with the fast Fourier transform [FFT] algorithm - the MATLAB function dft.m that was written to apply this algorithm is included in Appendix B.5). The program 'detrends' the signals by removing piecewise linear trends from the signals with user-defined breakpoints between adjacent linear segments (to remove the 0 Hz or DC values from the DFTs). After this, an exponential window (with a userspecified shape factor) is applied to the response signals to ensure periodicity (which reduces leakage). A MATLAB function expwindow.m (included in Appendix B.5) was written to apply this window. Again, the DFTs of these detrended and windowed signals are determined.

Figure 2.15 shows the isolated/cropped bearing displacements for the first recorded measurement of load case 4, including the piecewise linear trends (with 15 break points) removed by BushBearingCharSP.m. Figure 2.16 shows the detrended and windowed signals while Figure 2.17 shows the DFTs of these signals.


Figure 2.15 - Bush Bearing Characterisation Isolated Step Response \& Linear Trends (Load Case 4, Run 1)


Figure 2.16 - Bush Bearing Characterisation Windowed Step Response (Load Case 4, Run 1)


Figure 2.17 - Bush Bearing Characterisation Windowed Step Response DFTs (Load Case 4, Run 1)

Figure 2.17 shows clear peaks at approximately 68 Hz which indicates a damped dynamic natural frequency ( $f_{d}$ ) of the system ${ }^{14}$. Although both bearings were subject to approximately the same load, the linear trend values of bearing 2 and the vibration magnitude of bearing 2 at the damped natural frequency are higher than for bearing 1 (refer to Figure 2.15 and Figure 2.17) - this indicates that bearing 1 has a higher stiffness than bearing 2. Assuming bearings 1 and 2 have linear stiffness coefficients $k_{1}$ and $k_{2}$ respectively, the same constant load $F_{l}$ applied to both bearings cause different deflections, denoted $x_{1}$ and $x_{2}$ respectively. From Hooke's law:

$$
\begin{equation*}
k_{1}=\frac{F_{l}}{x_{1}} \quad \text { and } \quad k_{2}=\frac{F_{l}}{x_{2}} \tag{2.2}
\end{equation*}
$$

Dividing $k_{2}$ by $k_{1}$ yields:

$$
\begin{equation*}
r_{k}=\frac{k_{2}}{k_{1}}=\frac{F_{l} / x_{2}}{F_{l} / x_{1}}=\frac{x_{1}}{x_{2}} \tag{2.3}
\end{equation*}
$$

where $r_{k}$ is the stiffness ratio. The ratio between the stiffness coefficients of bearings 1 and 2 may be approximated from the ratio between the piecewise linear trend values of the bearing displacements. Thus, using the linear trend values of $x_{1}=0.006539 \mathrm{~s}$ and $x_{2}=0.00709 \mathrm{~s}$ at $t=3.368 s$ (refer to Figure 2.15 ) yields $r_{k}=0.922$. The experimental setup may therefore also be modelled as a two degree of freedom (2DOF) system where the two DOFs are translation of the centre of the shaft and rotation of the shaft ( $x$ and $\theta$ respectively), as shown in Figure $2.18 a$. The shaft mass is denoted $m_{s}$ while the shaft load weight is denoted $m_{l}$. Since the lateral mass moment of inertia of the shaft, denoted $I$, is quite small (compared to the inertia of the rest of the system)

[^11]and the shaft load weight does not rotate with the shaft, the first mode of the system is predominantly translational (i.e. movement in the $x$-direction) while the second mode is predominantly rotational (i.e. rotation in the $\theta$-direction) at a much higher frequency. The system may thus be modelled as an effective/reduced single degree of freedom (SDOF) system for the first mode, as depicted in Figure 2.18 b, with:
\[

$$
\begin{equation*}
m=m_{l}+m_{s} ; \quad k_{e f f}=k_{1}+k_{2} ; \quad c_{e f f}=c_{1}+c_{2} \tag{2.4}
\end{equation*}
$$

\]

where $m_{s}$ and $m_{l}$ are the masses of the shaft and the static shaft load respectively, $m$ is the static load (also the vibrating mass) and $k_{e f f}$ and $c_{e f f}$ are the effective stiffness and damping coefficients respectively for the first mode.

a) 2DOF Model

b) SDOF Model

Figure 2.18 - Bush Bearing Characterisation 2DOF \& SDOF Models

In order to isolate the response signals at the first natural frequency of the system and reduce noise in these signals, program BushBearingCharSP.m provides the user with an option (in the code itself) of applying either a low-pass or a band-pass filter to the bearing signals with user-specified cut-off frequencies. The program then detrends the filtered signals (with a user-defined number of break points) to remove any residual linear trends and computes the DFTs of these signals.

Figure 2.19 shows the filtered and detrended bearing displacement signals for the first recorded measurement of load case 4. A band-pass filter was used with lower and upper cut-off frequencies of 66 and 70 Hz respectively and a filter order of 100 . Figure 2.20 shows the DFTs of these signals.


Figure 2.19 - Bush Bearing Characterisation Filtered Step Response (Load Case 4, Run 1)



Figure 2.20 - Bush Bearing Characterisation Filtered Step Response DFTs (Load Case 4, Run 1)

### 2.3.1.3 Stiffness and Damping

The next step is to fit the SDOF model, depicted in Figure 2.18, on the filtered signal result. The free response of an 'underdamped' SDOF system may be described by the following equation:

$$
\begin{equation*}
x(t)=X_{0} e^{-\zeta \omega_{n} t} \cos \left(\omega_{d} t-\phi\right) \tag{2.5}
\end{equation*}
$$

where $t$ denotes time, $X_{0}$ is the (undamped) displacement amplitude, $\zeta$ is the damping coefficient, $\omega_{n}$ denotes the 'undamped' natural frequency (in $\mathrm{rad} / \mathrm{s}$ ), $\omega_{d}$ denotes the damped natural frequency (also in $\mathrm{rad} / \mathrm{s}$ ) and $\phi$ denotes the phase angle. The exponential decay of the free response is defined by the factor:

$$
\begin{equation*}
x_{\exp }(t)=X_{0} e^{-\zeta \omega_{n} t} \tag{2.6}
\end{equation*}
$$

from eqn. (2.5).
A MATLAB program db2k_1_bearing1_kc.m (included in Appendix B.1) was written to fit two exponential functions (one with positive $x$-values and one with negative $x$-values) of the form:

$$
\begin{equation*}
x_{\exp }(t)=a e^{b t} \tag{2.7}
\end{equation*}
$$

onto the filtered bearing 1 displacement signal from the first recorded measurement of load case 4 (refer to Figure 2.19). This program uses the Curve Fitting Toolbox of MATLAB to perform this task. The data points used for curve-fitting are defined in the code of db2k_1_bearing1_kc.m. The builtin MATLAB fit.m function finds the best curve fits to these data points, by applying the least-meansquare method, and returns the coefficients $a$ and $b$. From eqns. (2.6) and (2.7):

$$
\begin{equation*}
X_{0}=a \quad \text { and } \quad \omega_{n}=-\frac{b}{\zeta} \tag{2.8}
\end{equation*}
$$

But $\omega_{d}=\sqrt{1-\zeta^{2}} \omega_{n}$, thus:

$$
\begin{align*}
& \omega_{d}=-\frac{b}{\zeta} \sqrt{1-\zeta^{2}} \\
& \therefore \zeta=\frac{1}{\sqrt{\omega_{d}^{2} / b^{2}+1}} \tag{2.9}
\end{align*}
$$

Since $b$ is known from the curve fitting and $\omega_{d}=2 \pi f_{d}$ is known from the experimental results, $\zeta$ may be calculated from eqn. (2.9). The undamped natural frequency $\omega_{n}$ may then be calculated from $\omega_{d}$ and $\zeta$ :

$$
\begin{equation*}
\omega_{n}=\frac{\omega_{d}}{\sqrt{1-\zeta^{2}}} \tag{2.10}
\end{equation*}
$$

For the SDOF system, $\omega_{n}$ is also given by:

$$
\begin{gather*}
\omega_{n}=\sqrt{\frac{k_{e f f}}{m}}=\sqrt{\frac{k_{1}+k_{2}}{m}} \\
\quad \therefore m \omega_{n}^{2}=k_{1}+k_{2} \tag{2.11}
\end{gather*}
$$

Substituting eqn. (2.3) into eqn. (2.11) yields:

$$
\begin{equation*}
k_{1}=\frac{m \omega_{n}^{2}}{1+r_{k}} \tag{2.12}
\end{equation*}
$$

The damping ratio is given by :

$$
\begin{equation*}
\zeta=\frac{c_{e f f}}{c_{0}}=\frac{c_{1}+c_{2}}{2 m \omega_{n}} \tag{2.13}
\end{equation*}
$$

Assuming $c_{1}=c_{2}$ :

$$
\begin{equation*}
c_{1}=c_{2}=\zeta c_{0} / 2 \tag{2.14}
\end{equation*}
$$

Figure 2.21 shows the filtered bearing displacement signals for the first recorded measurement of load case 4 (the same as in Figure 2.19), including the data points defined in db2k_1_bearing1_kc.m and the fitted exponential curves calculated by this program.


Figure 2.21 - Bush Bearing Characterisation Filtered Step Response Curve Fitting (Load Case 4, Run 1)

Assuming that the two bearings of the experimental rotor system (the bush bearing and journal bearing) carry the entire weight of the rotor ( 2.0224 kg - refer to Table 2.1) equally, the bush bearing (bearing 1) is subject to a weight of approximately 9.920 N and out of the four static load cases considered for the characterisation of the bush bearing (refer to Table 2.5), load case 4 (with the 2 kg static load weights) provides a load of 10.582 N per bearing which is the closest matching load. The three best recorded measurements were chosen from this load case, namely the results from runs 1,3 and 4 . The damped natural frequencies $\left(f_{d}\right)$ and stiffness ratios $\left(r_{k}\right)$ were determined for each of these experimental results, as described in this section, and are given in Table 2.6.

Table 2.6 - Bush Bearing Characterisation Natural Frequencies \& Stiffness Ratios

| Load <br> Case | Run | $f_{d}[H z]$ | $\boldsymbol{t}[s]$ | $x_{1}[\mathrm{~mm}]$ | $\boldsymbol{x}_{2}[\mathrm{~mm}]$ | $r_{k}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 4 | 1 | 68 | 3.368 | 0.006539 | 0.00709 | 0.922 |
| 4 | 3 | 68 | 2.658 | 0.007455 | 0.008257 | 0.903 |
| 4 | 4 | 68 | 2.540 | 0.006299 | 0.007293 | 0.864 |

The values of $k_{1}, k_{2}, c_{1}$ and $c_{2}$ were calculated for each of these three chosen experimental results using MATLAB programs db2k_1_bearing1_kc.m, db2k_1_bearing3_kc.m and db2k_4_bearing1_kc.m - the results are given in Table 2.7. The filtered step response signals with fitted curves for runs 3 and 4 are included in Appendix F.2.

Table 2.7 - Bush Bearing Characterisation Results

| Load Case | Run | $\zeta$ | $k_{1}[N / m]$ | $k_{2}[N / m]$ | $c_{1} \& c_{2}[N . s / m]$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 4 | 1 | 0.081 | 206257.524 | 190169.437 | 74.858 |
| 4 | 3 | 0.078 | 208232.486 | 188033.935 | 72.508 |
| 4 | 4 | 0.079 | 212618.765 | 183702.613 | 73.321 |
| Average: |  |  |  |  |  |

### 2.3.2 Coupling Lateral Characterisation

The flexible element of the coupling consists of a synthetic rubber, thus its stiffness and damping characteristics (dominated by the flexible element) are nonlinear and dependent on the magnitudes and frequencies of forces transferred by the coupling. The approximate linear dynamic stiffness and damping coefficients of the coupling in its lateral directions ( $y$ - and $z$-directions in the rotor local rotating coordinate system) were therefore calculated from the step response of the coupling.

### 2.3.2.1 Experimental Setup

Figure 2.22 shows a schematic of the coupling lateral characterisation experimental setup while Figure 2.23 shows photos of the setup.


Figure 2.22 - Coupling Lateral Characterisation Experimental Setup Schematic


Figure 2.23 - Coupling Lateral Characterisation Experimental Setup

The driving half of the coupling was rigidly mounted in a shaft clamp while two wheels were attached to the driven half of the coupling. The slight offset between the sleeve, from which the static load was hanged, and the acting centre ${ }^{15}$ of the coupling resulted in a moment acting on the driven coupling half which was balanced by the top wheel pressing against a vertical flat surface facing the coupling and the bottom wheel pressing against a vertical flat surface facing away from the coupling ( $\therefore F_{l} l_{o f f}=F_{w} l_{w}$ in Figure 2.22). These wheels therefore enforce pure lateral motion of the driven coupling half when excited by the step input load. The step input load was attached to the static load via an electromagnet. The velocity step response of the driven coupling half, denoted $\dot{x}(t)$, was measured with the a laser vibrometer and stored in SVD file format.

### 2.3.2.2 Analysis and Results

All the components in this setup were assumed to be rigid compared to the flexible element of the coupling. The moving/vibrating components (excluding the step input load) are indicated in blue in Figure 2.22 - these components also comprise the static load applied to the coupling. The step input load was provided by abruptly cutting off the power supply to the electromagnet which then breaks free from the static load weight hanger. The voltage supplied to the electromagnet was also routed to the laser vibrometer system as a trigger signal that begins the vibration measurement by the laser vibrometer as soon as step input load is applied. Three different static load cases were considered, as listed in Table 2.8. The driven coupling half with the wheel attachment, the sleeve and the static load weight hanger weighed $111.5 \mathrm{~g}, 86.4 \mathrm{~g}$ and 51 g respectively ( 248.9 g in total).

Table 2.8-Coupling Lateral Characterisation Static Load Cases

| Load <br> Case | Static Load <br> Weight(s) $[\mathrm{kg}]$ | Total Vibrating <br> Mass $[\mathrm{kg}]$ | Total Static Load <br> $[\mathrm{N}]$ |
| :---: | :---: | :---: | :---: |
| 1 | 0.5 | 0.7489 | 7.347 |
| 2 | 1 | 1.2489 | 12.252 |
| 3 | 1.4 | 1.6489 | 16.176 |

The laser vibrometer settings used during the experiment are given in Table 2.9.
Table 2.9-Coupling Lateral Characterisation Laser Vibrometer Settings

| Channels | Range | Sensitivity |  |
| :--- | :---: | :---: | :---: |
| Vibrometer | 1 V | $50 \mathrm{~mm} / \mathrm{s} / \mathrm{V}$ |  |
| Electromagnet | 10 V | $1 \mathrm{~V} / \mathrm{V}$ |  |
| Frequency | 0.4 kHz |  |  |
| Bandwidth | 800 |  |  |
| FFT Lines | 1024 Hz |  |  |
| Sampling Frequency | $500 \mathrm{mHz}(\mathrm{T}=2 \mathrm{~s})$ |  |  |
| Frequency Resolution | Type |  | Shape Factor |
| Windowing | Exponential | 4 |  |
| Vibrometer |  |  |  |

A MATLAB program CouplingLatCharPolytec.m (included in Appendix B.2) was written to load the recorded data from the SVD files and store the data in comma-separated-value (*.csv) file format. As with PolytecScanFFTPIot.m (refer to section 2.2.2), this program requires an appropriate version

[^12]of PFA software to be installed and custom MATLAB function GetPointData.m (installed with PFA) to be in the working directory. The SVD files need to adhere to a naming convention used when loading the data. MATLAB program CouplingLateralCharSP.m (also included in Appendix B.2) was written to perform signal processing on data loaded or stored with CouplingLatCharPolytec.m similar to BushBearingCharSP.m (refer to section 2.3.1). This program applies an exponential window to the response signal (which also requires MATLAB function expwindow. $m$ in the working directory), determines the DFT of the windowed signal (which also requires MATLAB function dft.m in the working directory), filters the signal with a band-pass filter (to isolate the response at the natural frequency of the system), detrends the filtered signal and finally determines the DFT of the filtered signal.

Figure 2.24 shows the windowed velocity signal measured by the laser vibrometer (up to 0.5 s ) for the second recorded measurement of load case 1 and its DFT. The first peak in the DFT is at approximately 72 Hz while the second peak is at approximately 96 Hz . If the DFT results from the other load cases are also considered (refer to Appendix F. 3 for examples of DFTs from the other load cases), it seems as though the first peak indicates the damped natural frequency $\left(f_{d}\right)$ of the system since this frequency is different for each load case (i.e. a larger vibrating mass results in a lower $f_{d}$ ) while the second peak is barely visible for the other load cases (where larger step input loads were used) which may be attributed to nonlinearity of the flexible coupling element stiffness.


Figure 2.24 - Coupling Lateral Characterisation Windowed Step Response \& DFT (Load Case 1, Run 2)

Figure 2.25 shows the filtered and detrended velocity signal (using lower and upper cut-off frequencies of 67 and 77 Hz respectively and a filter order of 150) and also its DFT.


Figure 2.25-Coupling Lateral Characterisation Filtered Step Response \& DFT (Load Case 1, Run 2)

### 2.3.2.3 Stiffness and Damping

The free velocity response of a linear underdamped SDOF system may be expressed as the derivative of eqn. (2.5), however, the exponential decay of the free velocity response is still defined by the factor given as eqn. (2.6). A MATLAB program clhkFFTpnt1_2_kc.m (included in Appendix B.2), similar to db2k_1_bearing1_kc.m (refer to section 2.3.1), was written to fit two exponential functions onto the filtered velocity signal from the second measurement of load case 1. This program also calculates the damping coefficient and undamped natural frequency from eqns. (2.9) and (2.10). The approximate linear dynamic lateral stiffness and damping coefficients of the coupling are then calculated as follows:

$$
\begin{equation*}
k=m \omega_{n}^{2} \quad \text { and } \quad c=2 m \omega_{n} \zeta \tag{2.15}
\end{equation*}
$$

where $m$ is the mass of the static load (the vibrating mass).
Figure 2.26 shows the filtered velocity signal for the second recorded measurement of load case 1 (the same as in Figure 2.25), including the data points defined in clhkFFTpht1_2_kc.m and the fitted exponential curves calculated by this program.


Figure 2.26 - Coupling Lateral Characterisation Filtered Step Response Curve Fitting (Load Case 1, Run 2)

Assuming that the two bearings of the experimental rotor system (the bush bearing and journal bearing) carry most of the rotor weight, the coupling should only be lightly loaded and out of the three load cases considered, the first case (with the 0.5 kg static load weight) had the lightest static load. The three best recorded measurements, namely runs 1 to 3 , were chosen from this load case and the results from these measurements are given in Table 2.10. The filtered step response signals with fitted curves for runs 1 and 3 are included in Appendix F.3.

Table 2.10 - Coupling Lateral Characterisation Results

| Load Case | Run | $f_{d}[H z]$ | $\zeta$ | $k[N / m]$ | $c[N . s / m]$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 1 | 72 | 0.046 | 153598.253 | 31.502 |
| 1 | 2 | 72 | 0.047 | 153602.107 | 31.685 |
| 1 | 3 | 72 | 0.047 | 153603.580 | 31.755 |
|  | Average: | 72 | $\mathbf{0 . 0 4 7}$ | 153601.313 | 31.647 |

### 2.3.3 Coupling Torsional Characterisation

The approximate linear dynamic torsional stiffness and damping coefficients of the coupling (about the $x$-axis) were calculated from the angular step response of the coupling.

### 2.3.3.1 Experimental Setup

Figure 2.27 shows a schematic of the coupling torsional characterisation experimental setup while Figure 2.28 shows a photo of this setup. The driving half of the coupling was mounted in the shaft clamp while the driven half was connected to a potensiometer which measured its angular step response, denoted $\theta(t)$ (stored in TXT file format). Since the Bently Nevada rotor kit transmits no torsional load, the coupling does not experience any torsional loading while assembled in the rotor kit setup, except to overcome some bearing friction and the rotor inertia during acceleration. The linear dynamic stiffness and damping coefficients of the coupling were thus required to be determined for only a small torsional load. On the other hand, this characterisation setup required a large mass moment of inertia about the coupling spin axis (on the driven side) in order to invoke an underdamped step response from the system. These requirements were met by hanging two sets of weights from two bolts connected to opposite sides of the driven coupling half, denoted sides 1 and 2 . The bolt at side 1 was mounted in the setscrew hole of the coupling while the bolt at side 2 was mounted onto a nut placed over the driven coupling half. The effective static torque applied to the coupling therefore equalled the difference between the torsional loads applied to sides 1 and 2 . The step input torque was applied to side 1.


Figure 2.27 - Coupling Torsional Characterisation Experimental Setup Schematic


Figure 2.28-Coupling Torsional Characterisation Experimental Setup

### 2.3.3.2 Analysis and Results

All the components in this setup were assumed to be rigid compared to the flexible element of the coupling. The moving/vibrating components (excluding the step input load) are indicated in blue in Figure 2.27. The step input load was provided by severing the fishing line with a flame. Four main static load cases were considered, as listed in Table 2.11. The distances from weight hangers 1 and 2 to the coupling centreline, denoted $l_{m 1}$ and $l_{m 2}$, were 78 mm and 77 mm respectively, while the distances from the centres of gravity of bolts 1 and 2 to the coupling centreline, denoted $l_{\text {boltG1 }}$ and $l_{\text {boltG2 }}$, were 51.5 mm and 52.5 mm respectively. Bolts 1 and 2 weighed 11.3 g and 10.2 g respectively and the axial mass moments of inertia of the coupling and the nut were $7.0896 \mathrm{~kg} . \mathrm{mm}^{2}$ and $27.135 \mathrm{~kg} . \mathrm{mm}^{2}$ respectively.

Table 2.11 - Coupling Torsional Characterisation Static Load Cases

| Load <br> Coase | Weight(s) $[\mathrm{kg}]$ |  | Static Torque <br> $[\mathrm{N} / \mathrm{mm}]$ | Vibrating Inertia <br> $I\left[\right.$ t. $\left.\mathrm{mm}^{2}\right]$ |
| :---: | :---: | :---: | :---: | :---: |
| a | 0.8 | 0.6 | 220.843 | 9.125 |
| b | 0.85 | 0.65 | 221.334 | 9.726 |
| c | 0.9 | 0.6 | 297.361 | 9.733 |
| d | 1 | 0.8 | 222.805 | 11.528 |

A MATLAB program CouplingTorsChar.m (included in Appendix B.3), similar to BushBearingChar.m (refer to section 2.3.1), was written to load the raw potensiometer signal, convert the signal from millivolts $(\mathrm{mV})$ to degrees $\left({ }^{\circ}\right)$ using a sensitivity of $-4.065 \mathrm{mV} /{ }^{\circ}$ (determined during calibration of the potensiometer) and plot the resulting signals. The TXT filenames also need to adhere to a naming convention used when loading the data. MATLAB program CouplingTorsCharSP.m (also included in Appendix B.3), similar to BushBearingCharSP.m (refer to section 2.3.1), was written to perform signal processing on the loaded signal by resampling the signal (if required by the user), cropping it to extract only the step response part of the signal, detrending the signal, applying an exponential window to the signal (which requires MATLAB function expwindow.m in the working directory) and filtering the signal.

Figure 2.29 shows the angular displacement signal measured by the potensiometer (from $1.5 s$ to $4 s$ ) for the second recorded measurement of load case d. Figure 2.30 shows this signal after it has been cropped, detrended (by removing the average value of the signal) and windowed. The quick dissipation in the vibration caused by the step input load suggests a highly damped system. The damped dynamic natural frequency of the system was estimated to be 10 Hz from the duration ( 0.049 s ) between the minimum and consecutive maximum step response displacements (half an oscillation).


Figure 2.29 - Coupling Torsional Characterisation Step Response (Load Case d, Run 2)


Figure 2.30-Coupling Torsional Characterisation Isolated \& Windowed Step Response (Load Case d, Run 2)

Figure 2.31 shows the filtered and detrended angular displacement signal (using a low-pass filter with a cut-off frequency of 30 Hz and order of 10).


Figure 2.31 - Coupling Torsional Characterisation Filtered Step Response (Load Case d, Run 2)

### 2.3.3.3 Stiffness and Damping

A MATLAB program ct2m_d_2_kc.m (included in Appendix B.3) was written to fit the free response from an underdamped SDOF system onto the filtered angular displacement signal from the second measurement of load case $d$. The logarithmic decrement of the angular response of such a SDOF system for half an oscillation is given by...

$$
\begin{equation*}
\delta=2 \ln \frac{\theta_{t}}{\theta_{t+T / 2}} \tag{2.16}
\end{equation*}
$$

from which the damping coefficient may be calculated:

$$
\begin{equation*}
\zeta=\frac{\delta}{\sqrt{(2 \pi)^{2}+\delta^{2}}} \tag{2.17}
\end{equation*}
$$

The undamped natural frequency is then given by eqn. (2.10) and the approximate linear dynamic torsional stiffness and damping coefficients of the coupling are given by:

$$
\begin{equation*}
k_{\theta}=I \omega_{n}^{2} \quad \text { and } \quad c_{\theta}=2 I \omega_{n} \zeta \tag{2.18}
\end{equation*}
$$

where $I$ is the vibrating mass moment of inertia.
Figure 2.32 shows the filtered angular displacement signal for the second recorded measurement of load case $d$ (the same as in Figure 2.31), including the two data points used to calculate the logarithmic decrement and the fitted SDOF model free response, given by:

$$
\begin{equation*}
\theta(t)=\theta_{0} e^{-\zeta \omega_{n} t} \cos \left(\omega_{d} t-\phi\right) \tag{2.19}
\end{equation*}
$$



Figure 2.32 - Coupling Torsional Characterisation Filtered Step Response Curve Fitting (Load Case d, Run 2)

The best step response (with a low static torque applied to the system) was noted for static load case $d$ and thus, the three best recorded measurements, namely runs 1 to 3 , were chosen from this load case and the results from these measurements are given in Table 2.12. The filtered step response signals with fitted curves for runs 1 and 3 are included in Appendix F.4.

Table 2.12 - Coupling Torsional Characterisation Results

| Load Case | Run | $f_{d}[\mathrm{~Hz}]$ | $\zeta$ | $k[\mathrm{Nm} / \mathrm{rad}]$ | $c[$ Nm. $\mathrm{s} / \mathrm{rad}]$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| d | 1 | 11 | 0.393 | 63.576 | 0.673 |
| d | 2 | 10 | 0.391 | 55.937 | 0.628 |
| d | 3 | 11 | 0.378 | 60.087 | 0.629 |
| Average: |  | 10.667 | 0.387 | 59.867 | 0.643 |

### 2.3.4 Journal Bearing Characteristics

### 2.3.4.1 Geometrical Parameters

A telescopic gauge and micro meter (as shown in Figure 2.33) were used to accurately measure the diameter of the journal and the inner diameter of the journal bearing housing.


Figure 2.33 - Telescopic Gauge \& Micro Meter for Measuring Journal Bearing Clearance

The journal diameter was measured to be on average 24.62 mm while the bearing housing inner diameter was on average 25.04 mm . The radial clearance is thus approximately 0.2 mm .

The effective length of the journal bearing is approximately 11.5 mm , calculated from the measured dimensions of the experimental rotor system as shown in the detail drawing of the rotor system included in Appendix A.1.

### 2.3.4.2 Lubrication

Shell Tellus S2 V 46 hydraulic fluid was used as the lubricating fluid within the journal bearing. The typical physical characteristics of this fluid (extracted from a technical data sheet of the product) are included in Appendix G.1. Assuming that the journal bearing operates at approximately $40^{\circ} \mathrm{C}$, the dynamic viscosity $(\mu)$ of the fluid may be approximated from the kinematic viscosity $(v)$ of the fluid at $40^{\circ} \mathrm{C}$ and its density $(\rho)$ at $15^{\circ} \mathrm{C}$ (the only density value provided):

$$
\begin{equation*}
\mu=\rho v=\left(872 \mathrm{~kg} / \mathrm{m}^{3}\right)\left(46 \times 10^{-6} \mathrm{~m}^{2} / \mathrm{s}\right)=0.0401 \text { Pa.s } \approx 4 \times 10^{-8} \mathrm{MPa} . \mathrm{s} \tag{2.20}
\end{equation*}
$$

## 3 NUMERICAL Rotor System Models

This chapter describes the development and rotor dynamic analysis of numerical models of the experimental rotor system (described and characterised in chapter 2) including a hydrodynamic journal bearing and faults.

As mentioned in section 1.2.4, rotor systems are usually modelled with 1D elements instead of 3D elements due to the considerable computation time that may be required to solve certain rotor dynamic problems (especially nonlinear transient dynamic response analysis). 3D elements may however provide for more accurate modelling of complex rotor systems (such as turbogenerators). MSC.Nastran possesses the capability to perform lateral rotor dynamic analyses using 3D elements (a capability which currently seems to be, to a large extent, unexplored and/or undocumented). Both 1D and 3D FE models were thus developed for the experimental rotor system by employing MSC.Nastran.

The purpose of the numerical models developed is to investigate (in conjunction with the experimental rotor system) the effects of certain rotor system features and faults on the dynamic response of the system. Although these numerical models were developed specifically for the experimental (laboratory-scale) rotor system used for this dissertation, the methodologies and software used to create and analyse these models were developed to be capable of modelling and analysing flexible rotor systems in general. Future work could therefore include the modelling of other excitation forces (such as the excitation forces exerted on turbine blades by flowing steam), seal-shaft interactions, shaft cracks, etc.

A nonlinear journal bearing model was developed separately from the FE rotor system models (which were first constructed with only linear bearings) and then integrated into these rotor system models - this had to be done since the version of MSC.Nastran used does not have a built-in cylindrical hydrodynamic journal bearing model. For this and other reasons, this chapter is structured into the following main sections:

- FE rotor models (without the support structure: free-free)
- FE rotor system models with linear bearings
- Journal bearing model
- Incorporation of the journal bearing model into the FE rotor system models
- FE rotor system models with nonlinear journal bearings

The coordinate systems defined in section 2.1.2 for the rotor system are applicable here. Certain sections of this chapter require a basic knowledge of MSC.Nastran and MSC.Patran from the reader.

### 3.1 Finite Element Rotor Models

During the development of each FE rotor system model, only the rotor part of the system (as defined in section 2.1.1) was constructed at first - thus excluding the support structure or any constraints. These free-free rotor models were analysed to ensure they accurately represent the experimental rotor with regards to its dynamic properties (specifically its natural frequencies and mode shapes) and, if so required, these models could be updated to this end.

For these free-free rotor models and the rest of the FE models, a systematic model development approach was followed that allows for easy detection and elimination of possible errors in the construction and modification of the models in MSC. Patran and also to explore the Rotor Dynamics capability of MSC.Nastran and the effect of certain model features (such as damping, gravity, etc.) on the results obtained. This systematic approach entailed storing copies of the models at each prominent step ${ }^{16}$ in their construction and analysing the models at some of these steps. Tree diagrams of these systematic model construction steps and analyses performed at certain steps are included in Appendix I.

The two disks are approximately rigid, compared to bending of the shaft, which means that the influence of centrifugal stiffening is negligible - centrifugal stiffening was therefore not included in any of the FE models developed.

Since no units can be specified in MSC.Nastran, a system of consistent units (such as the SI units) needs to be used. The units used for the rotor system models are listed in Table 3.1.

Table 3.1 - System of Consistent Units used for FE Modelling

| Property | Unit |
| :--- | :---: |
| Length/Displacement | mm |
| Force | N |
| Mass | t or Mg |
| Mass Density | $\mathrm{t} / \mathrm{mm}^{3}$ |
| Mass Moment of Inertia | $\mathrm{t} . \mathrm{mm}^{2}$ |
| Elasticity Modulus | MPa |
| Stress | MPa |
| Circular Frequency | $\mathrm{rad} / \mathrm{s}$ |
| Frequency | Hz |

### 3.1.1 1D Rotor Model

The 1D rotor model consists of beam elements for the shaft and concentrated mass elements for the disks, journal and coupling located at the C.G. of each of these components. These concentrated mass elements require the calculation of the effective inertial properties of each component. The solid model referred to in section 2.2.1, the rotor mass densities determined using this model (refer to Table 2.2, p. 32) and the SolidWorks Mass Properties tool were used to determine these effective inertial properties. Figure 3.1 shows the solid models of the disks, journal and coupling (including their setscrews) used to determine their effective inertial properties.

[^13]

Figure 3.1 - Solid Models used to Determine Effective Inertial Properties for 1D Rotor Model

Table 3.2 lists the effective inertial properties (relative to the C.G. of each component) and C.G. coordinates of the disks, journal and coupling in the local coordinate system of the rotor.

Table 3.2 - Effective Inertial Properties for 1D Rotor Model

| Component | C.G. Coordinates [mm] |  |  | Mass [g] | Mass Moment of Inertia [g. $\mathrm{mm}^{2}$ ] |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $x$ | $y$ | z | m | $I_{x x}$ | $I_{y y}$ | $I_{z z}$ |
| Disk 1 | 193 | 0 | 0 | 813.64 | 577345.17 | 333836.43 | 328436.32 |
| Disk 2 | 293.5 | 0 | 0 | 810.40 | 572857.16 | 331130.49 | 325730.38 |
| Journal | 427.75 | 0 | 0 | 80.61 | 6954.02 | 7839.13 | 7839.13 |
| Coupling | -12.81 | -0.11 | 0 | 65.39 | 7089.57 | 6639.08 | 5708.88 |

The disks and coupling are not axisymmetric, hence the differences in their lateral mass moments of inertia about the $y$ - and $z$-directions. Since disks 1 and 2 have slightly different geometries (different chamfers - refer to Appendix A.1), the inertial properties of the two disks differ.

Figure 3.2 is a schematic of the 1D rotor model (see Appendix A. 2 for an enlarged version) showing its dimensions and axial ( $x$-axis) locations of its components and acting centres of the bearings and coupling connection in the global coordinate system. The symbols $m_{j}, m_{c}, m_{d 1}$ and $m_{d 2}$ denote the concentrated mass elements of the journal, the coupling, disk 1 and disk 2 respectively.


Figure 3.2-1D Rotor System Model Dimensions

The first step in the construction of the 1D model in MSC.Patran was to create the model geometry. Next, the shaft was meshed and all the rotor elements and their properties were defined. Figure 3.3 shows the 1D rotor model in MSC.Patran. The shaft consists of 45 beam elements (CBEAM elements), shown as blue lines, and 4 concentrated mass elements (CONM2), shown as orange triangles, and has 276 DOFs. The origin of the coordinate system is shown as a black cross while the node (GRID) points are shown as green circles. All shaft nodes and concentrated mass elements lie on the shaft centreline, the $x$-axis (where $y=z=0$ ). The concentrated mass elements of disks 1 and 2 , the journal and the coupling are connected to nodes 21, 32,46 and 52 respectively. Node 52 is located at the end of the shaft at $x=-12.5 \mathrm{~mm}$ while the axial location of the C.G. of the coupling is actually at $x=-12.81 \mathrm{~mm}$ (refer to Table 3.2) - the C.G. of the coupling concentrated mass element is therefore specified, in its element properties, to be offset from node 52 by 0.31 mm in the negative $x$-direction (refer to Table 3.4).


Figure 3.3-1D Rotor Model in MSC.Patran

As discussed in section 2.2.2.4, the elastic modulus of the shaft material was set equal to 210 GPa as an initial guess and then updated as described in section 3.1.1.2. The shaft material mass density listed in Table 2.2 ( p .32 ) is used in the model along with an assumed Poisson's ratio of 0.3 . The beam elements have solid circular cross-sections with radii $(r)$ as given in Table 3.3.

Table 3.3-1D Rotor Model Shaft Beam Elements Geometrical Properties

| Axial Location [mm] | $r$ [mm] |
| :---: | :---: |
| $0 \leq x \leq 443$ | 4.75 |
| $-25 \leq x<0$ | 3.9 |

Figure 3.4 shows a visual representation of the shaft beam elements geometrical properties. The smaller shaft diameter at the coupling is clearly visible.


Figure 3.4-1D Rotor Model Shaft Beam Elements Geometrical Properties

The element properties of the concentrated mass elements are given in Table 3.4.

Table 3.4-1D Rotor Model Concentrated Mass Elements Properties

| Rotor Element | Axial Location $[\mathrm{mm}]$ | $m[t]$ | $I_{x x}\left[t . \mathrm{mm}^{2}\right]$ | $I_{y y}=I_{Z Z}\left[t . \mathrm{mm}^{2}\right]$ |
| :--- | :---: | :---: | :---: | :---: |
| Disk 1 $\left(m_{d 1}\right)$ | $x=193$ | $8.1364 \times 10^{-4}$ | 0.577345 | 0.331136 |
| Disk 2 $\left(m_{d 2}\right)$ | $x=293.5$ | $8.104 \times 10^{-4}$ | 0.572857 | 0.32843 |
| Journal $\left(m_{j}\right)$ | $x=427.75$ | $8.061 \times 10^{-5}$ | 0.00695402 | 0.00783913 |
| Coupling $\left(m_{c}\right)$ | $x=-12.5$ | $6.539 \times 10^{-5}$ | 0.00708957 | 0.00617398 |

Since the disks and coupling are not axisymmetric and MSC.Nastran only supports the rotor dynamic analysis of axisymmetric rotors (refer to section 1.2.5.2), the average lateral mass moments of inertia of these components about the $y$ - and $z$-directions (from Table 3.2) have to be used in the element properties. Substantial rotor asymmetry may cause system instability at certain spin speeds, dependency of the unbalance response amplitude and phase on the angular position of the unbalance as well as double frequency vibrations (Jei \& Lee, 1992). The coupling of the rotor considered here may be asymmetrical, but it consists of a relatively low-density material and only forms a small part of the entire rotor while the disks are only slightly asymmetrical due to their setscrew holes. The effects of this slight asymmetry on the lateral rotor dynamics should thus be minimal and can be safely ignored.

### 3.1.1.1 Normal Modes

Before adding material damping to the rotor model, a normal modes (real eigenvalue) analysis was performed to determine the undamped natural frequencies of the free-free rotor model. MSC.Nastran solution sequence SOL 103 was utilised to solve this problem. The MSC.Nastran BDF (*.bdf) input file (created using MSC.Patran) for this analysis is included in Appendix D.1.1.

The first 17 resulting natural frequencies (all frequencies below 1000 Hz ) are given in Table 3.5.

Table 3.5-1D Free-Free Rotor Model Natural Frequencies

| Mode | $f_{n}[\mathrm{~Hz}]$ | Mode Type |
| :---: | :---: | :--- |
| $1-6$ | 0 | Rigid body |
| 7 \& 8 | 85.477 | Lateral |
| 9 \& 10 | 188.7 | Lateral |
| 11 | 236.17 | Torsional |
| 12 \& 13 | 458.33 | Lateral |
| 14 \& 15 | 676.81 | Lateral |
| 16 \& 17 | 878 | Lateral |

Since the rotor model is axisymmetric, it has symmetric lateral modes. The first three symmetric flexural lateral mode shape pairs are shown in Figure 3.5 to Figure 3.7. The light blue lines indicate the undeformed geometry while the darker blue lines indicate the deformed geometry. The rest of the mode shapes are included in Appendix F.5. Note that MSC.Patran can only display deformation as translation and can therefore not display the angular deformation of rotational DOFs as rotation.


Figure 3.5-1D Rotor Model $1^{\text {st }}$ Symmetric Lateral Mode Shape Pair - 85.477 Hz


Figure 3.6-1D Rotor Model $2^{\text {nd }}$ Symmetric Lateral Mode Shape Pair - 188.7 Hz


Figure 3.7-1D Rotor Model $3^{\text {rd }}$ Symmetric Lateral Mode Shape Pair - 458.33 Hz

To verify that the correct material densities and concentrated masses are specified at the correct locations, MSC.Nastran's Grid Point Weight Generator may be used. This feature calculates, amongst others, the total mass, the C.G. location and the mass moments of inertia about the C.G. relative to the specified reference point. To calculate these values relative to the origin of the global or basic coordinate system, the following MSC.Nastran entry was added (manually) to the Bulk Data Section of the BDF input file:

```
PARAM,GRDPNT, n
```

The results from this calculation are:

- C.G. position (coordinates) relative to global coordinate system: $(238.9962,0,0) \mathrm{mm}$
- Total rotor mass: 2.022729 kg

This corresponds well with the rotor mass and C.G. location determined in section 2.2.1.

### 3.1.1.2 Model Updating

Model updating was performed on the 1D rotor model by employing the MSC.Nastran Design Optimization capability (refer to section 1.2.6) to minimise the difference between the lateral mode natural frequencies obtained from the 1D rotor FE model and the experimental modal analysis (refer to section 2.2.2, and Table 2.4, p. 37). Average experimental natural frequencies in the vertical and horizontal directions (local $y$ - and $z$-directions) were used (see Table 3.6) since the FE model is required to be axisymmetric while the experimental rotor is not completely axisymmetric. Since the damped natural frequencies of the rotor model are almost identical to the undamped natural frequencies for the rotor structural damping coefficient used (refer to section 3.1.1.3), model updating was performed using normal modes. Figure 3.8 shows a comparison between the average experimental natural frequencies and the 1D rotor model natural frequencies which seem to correlate quite well.


Figure 3.8 - Free-Free Natural Frequencies: 1D Rotor Model vs. Experimental Modal Analysis

The error function used for updating the 1D rotor model, based on the Bayesian Parameter Estimation (refer to section 1.2.6) and the least-mean-square method, is:

$$
\begin{equation*}
E F=w t \sum_{k=1}^{5} w r_{k}\left(\frac{f_{n k t}-f_{n k m}}{f_{n k t}}\right)^{2}+w m\left(\frac{E-E_{i}}{E_{i}}\right)^{2} \tag{3.1}
\end{equation*}
$$

where $f_{n k t}$ and $f_{n k m}$ denote experimental/test natural frequencies and model natural frequencies respectively, while $E$ and $E_{i}$ respectively denote the updated and initial material elastic moduli of the 1D rotor model (specifically its shaft). $E$ is therefore the model parameter to be updated through optimization from an initial value of 210 GPa (refer to section 2.2.2.4). Values of 1 and 0.1 were used for wt and wm respectively to instil greater confidence in the experimental results than the modelling assumptions. The rest of the updating parameter values are given in Table 3.6.

Table 3.6-Ave. Experimental/Test Natural Frequencies with Corresponding Weighting Values

| $k$ | $w r_{k}$ | $f_{n k t}[H z]$ |
| :---: | :---: | :---: |
| $\mathbf{1}$ | 1 | 84.8 |
| 2 | 1 | 190.7 |
| 3 | 0.7 | 456.5 |
| 4 | 0.7 | 700.3 |
| 5 | 0.7 | 894.85 |

Larger $w r_{k}$ values were chosen for the lower frequency modes compared to the higher frequency modes since the dynamic response of a structure is usually dominated by its lower frequency modes.

The Design Optimization capability of MSC.Nastran was utilised by entry additions and modifications that were manually made to the BDF input file included in Appendix D.1.1 from the normal modes analysis discussed in the preceding section (3.1.1.1). These additional/modified entries are included in Appendix D.1.2.

The resulting optimized/updated value of $E$ from the model updating analysis is 213343.02 MPa which differs from the initial value of 210000 MPa by only $1.6 \%$. A decision was therefore made to keep using the value of $E=210000 M P a$ for the 1D rotor model.

### 3.1.1.3 Complex Modes

Next, material damping was included in the rotor model in the form of structural (hysteretic) damping with $\eta=2 \%$ (a typical damping coefficient for structures with bolted connections - Fredö et al., 1999). The damped natural frequencies of the free-free rotor model are solved by performing a complex modes (complex eigenvalue) analysis. The BDF input files for the direct and modal formulations (SOL 107 and SOL 110 respectively) are included in Appendix D.1.3.

The first 17 resulting damped natural frequencies are almost identical to those listed in Table 3.5 this indicates that the specified structural damping has an almost negligible effect on the natural frequencies of the rotor model. The mode shapes are thus also similar to those shown in Figure 3.5 to Figure 3.7.

### 3.1.2 3D Rotor Model

The geometry of the 3D rotor model is a simplification/approximation of the geometry of the experimental rotor system described in section 2.1.2 (and also Appendix A.1). This simplified geometry is axisymmetric, without chamfers, without setscrews, has identical disks and has a constant shaft diameter of 9.5 mm throughout the entire length of the shaft. The same material mass densities as listed in Table 2.2 (p. 32) were used along with the SolidWorks Mass Properties tool to construct this simplified model geometry (on a trial-and-error basis) such that the inertial properties of its disks and coupling are approximately the axisymmetric equivalent of the properties from the solid model referred to in section 2.2.1. Table 3.7 shows a comparison between the geometries and inertial properties (relative to the C.G.) of the detailed and simplified disks, coupling and shaft (setscrews included in the detailed components).

The inertial properties of the simplified disks were 'tuned' by adjusting the diameter of the 16 balancing holes. The inertial properties of the simplified coupling and shaft (only the portion of the shaft with a diameter of 7.8 mm in the detailed model) were tuned by adjusting the length of the coupling and the depth of the axially-concentric hole at the back of the simplified coupling. The blue dimensions in Table 3.7 indicate the positions of the C.G. of the detail and simple couplings and shafts. The material mass densities of the shaft, disks, journal and coupling as listed in Table 2.2 are therefore used in the 3D FE rotor model.

Table 3.7-3D Rotor Model Equivalent Simplified Axisymmetric Parts Properties


Figure 3.9 is a schematic of the 3D rotor model (see Appendix A. 3 for an enlarged version) showing its dimensions and axial ( $x$-axis) locations of it components and acting centres of the bearings and coupling connection in the global coordinate system.


Figure 3.9-3D Rotor System Model Dimensions

The first steps in the construction of the 3D model in MSC.Patran were to create and mesh the simplified rotor geometry. Figure 3.10 shows the meshed 3D rotor model in MSC. Patran using two different display styles (additional images of this MSC.Patran model are included in Appendix A.4). The model consists of 18224 8-node hexahedral solid elements (CHEXA) and has 68952 DOFs. The disks were constructed by first creating a surface mesh with 4-node quadrilateral shell elements and then extruding these elements to form 8-node hexahedral elements. The mesh quality/reliability was inspected within MSC.Patran. The aspect ratio, edge angle and face skew values are high only for the second radial row of disk elements from the shaft (the red-coloured elements shown in Appendix A.4). However, since these elements are located on the two disks, which remain approximately rigid during bending of the shaft, the effects of their distortion are negligible (low stress and strain).


Figure 3.10-3D Rotor Model in MSC.Patran

As discussed in section 2.2.2.4, the elastic modulus of the shaft, disk and journal material was initially set equal to $210 G P a$ while the elastic modulus of the coupling material was initially set equal to 120 GPa. These values were then updated as described in section 3.1.2.2. An assumed Poisson's ratio of 0.3 is used for the model.

### 3.1.2.1 Normal Modes

As with the 1D rotor model, a normal modes (real eigenvalue) analysis was performed (using SOL 103) on the 3 D rotor model to determine its free-free undamped natural frequencies (refer to section 3.1.1.1). The BDF input file for this analysis is included in Appendix D.2.1 (due to space constraints, most of the CHEXA entries are excluded).

The first 17 resulting natural frequencies are given in Table 3.8.
Table 3.8-3D Free-Free Rotor Model Natural Frequencies (Before Updating)

| Mode | $f_{n}[\mathrm{~Hz}]$ | Mode Type |
| :---: | :---: | :--- |
| $1-6$ | 0 | Rigid body |
| 7 \& 8 | 94.028 | Lateral |
| 9 \& 10 | 212.18 | Lateral |
| 11 | 260 | Torsional |
| 12 \& 13 | 488.71 | Lateral |
| 14 \& 15 | 822.39 | Lateral |
| 16 \& 17 | 1041.4 | Lateral |

Once again, the rotor has symmetric modes due to its axisymmetric shape. The first three symmetric flexural lateral mode shape pairs are shown in Figure 3.11, Figure 3.12 and Figure 3.14 while the first torsional mode shape is shown in Figure 3.13. For each mode, the light blue lines indicate the undeformed geometry while the filled shape with black lines indicates the deformed geometry. The fringe (fill) colours represent the relative (normalized) magnitude of translational displacement from the undeformed shape (no displacement = dark blue; maximum displacement = red). In the torsional mode (Figure 3.13) the rotor is twisted about a point between disks 1 and 2 as indicated by the dark blue fill colour between these disks. The rest of the mode shapes are included in Appendix F.6.


Figure 3.11-3D Rotor Model ${ }^{\text {st }}$ Symmetric Lateral Mode Shape Pair - 94.028 Hz


Figure 3.12-3D Rotor Model $2^{\text {nd }}$ Symmetric Lateral Mode Shape Pair - 212.18 Hz


Figure 3.13-3D Rotor Model $1^{\text {st }}$ Torsional Mode Shape (Mode 11) - 260 Hz


Figure 3.14-3D Rotor Model 3 ${ }^{\text {rd }}$ Symmetric Lateral Mode Shape Pair - 488.71 Hz

To calculate the total mass and C.G. location of the rotor, the MSC.Nastran Grid Point Weight Generator (PARAM,GRDPNT entry) was used here again. The results from this calculation are:

- C.G. position (coordinates) relative to the global coordinate system: $(239.3072,0,0) \mathrm{mm}$
- Total rotor mass: 2.022624 kg

This corresponds well with the rotor mass and C.G. location determined in section 2.2.1.

### 3.1.2.2 Model Updating

As with the 1D rotor model, the 3D rotor model was updated using the MSC.Nastran Design Optimization capability to minimise the difference between the lateral mode natural frequencies obtained from the 3 D rotor FE model and the experimental modal analysis (refer to section 3.1.1.2). The same average experimental natural frequencies and corresponding weighting factors from Table 3.6 (p. 65) were used for updating the 3D rotor model. Normal modes were, once again, used since the damped natural frequencies of the rotor model are almost identical to the undamped natural frequencies (refer to section 3.1.2.3).

Figure 3.15 a shows a comparison between the average experimental natural frequencies and the (initial) 3D rotor model natural frequencies. Some deviation is noted between the calculated and experimental natural frequencies which necessitates the use of model updating. Since the 3D rotor model has two different material properties, whereas the 1D rotor model has only one, the error function used for the 1D model, eqn. (3.1), had to be adjusted for the 3D model:

$$
\begin{equation*}
E F=w t \sum_{k=1}^{5} w r_{k}\left(\frac{f_{n k t}-f_{n k m}}{f_{n k t}}\right)^{2}+w m \sum_{k=1}^{2} w p_{k}\left(\frac{E_{k}-E_{k i}}{E_{k i}}\right)^{2} \tag{3.2}
\end{equation*}
$$

$E_{1}$ denotes the updated elastic modulus of the shaft, disks and journal material and $E_{2}$ denotes the updated elastic modulus of the coupling material. Both these values were updated from initial values ( $E_{1 i}$ and $E_{2 i}$ ) of $210 G P a$ and $120 G P a$ respectively (refer to section 2.2.2.4) with weighing factors ( $w p_{1}$ and $w p_{2}$ ) of 1 and 0.5 respectively. Values of 1 and 0.1 were once again used for $w t$ and $w m$ respectively. The entry additions and modifications that were manually made to the BDF input file included in Appendix D.2.1 from the normal modes analysis discussed in the preceding section (3.1.2.1) are included in Appendix D.2.2.

The resulting optimized values of $E_{1}$ and $E_{2}$ from the model updating analysis are 168503.66 MPa and 119 991.36 MPa respectively. The optimized value of $E_{2}$ is almost the same as its initial value and a decision was therefore made to keep the value of $E_{2}=120000 \mathrm{MPa}$ in the 3D model from this result, it also stands to reason that the elastic modulus (i.e. the stiffness) of the coupling is non-consequential to the lateral modes of the rotor (at least for the lower frequency modes that are of interest) and it was therefore adequate to model the coupling as merely a concentrated mass in the 1D rotor model. The optimized value of $E_{1}$, however, differs substantially from its initial value (by $19.8 \%$ ) and a decision was made to use a rounded value of $E_{1}=168500 \mathrm{MPa}$ in the 3D model. Table 3.9 and Figure 3.15 show comparisons between the average experimental lateral natural frequencies and the undamped natural frequencies of the initial and updated 3D rotor models.

Table 3.9 - Free-Free Natural Frequencies: Initial \& Updated 3D Models vs. Experimental Modal Analysis

| Mode | Mode Type | $f_{n}[H z]$ |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  | Experiment (ave.) | Initial Model | Updated Model |
| 1-6 | Rigid body | 0 | 0 | 0 |
| 7 \& 8 | Lateral | 84.8 | 94.028 | 84.227 |
| 9 \& 10 | Lateral | 190.7 | 212.18 | 190.06 |
| 11 | Torsional | - | 260 | 232.9 |
| 12 \& 13 | Lateral | 456.5 | 488.71 | 437.77 |
| 14 \& 15 | Lateral | 700.3 | 822.39 | 736.68 |
| 16 \& 17 | Lateral | 894.85 | 1041.4 | 932.84 |

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems


Figure 3.15 - Free- Free Natural Frequencies: Initial \& Updated 3D Models vs. Experiment

The results shown in the table and figure above indicate an improved correlation between the natural frequencies from the updated 3D model and the experimental modal analysis - especially for the first two modes, which are the most important. The deviations in the $4^{\text {th }}$ and $5^{\text {th }}$ mode shape pair frequencies may be attributed to modelling errors/approximations such as the rigidly modelled disk-shaft connections (vs. the actual contact-type connections in the experimental rotor) and the axisymmetric disks and coupling (vs. the non-axisymmetric experimental rotor). The mode shapes of the updated model are almost identical to the mode shapes of the initial model (refer to the preceding section-3.1.2.1) and are therefore not included here again.

### 3.1.2.3 Complex Modes

As with the 1D rotor model, direct and modal complex modes (complex eigenvalue) analyses (SOL 107 and SOL 110 respectively) were performed on the free-free 3D rotor model, also with a structural damping coefficient of $2 \%$ (specified in the material properties of the model), to determine its damped natural frequencies. The BDF input files for these analyses are included in Appendix D.2.3.

The first 17 resulting damped natural frequencies are almost identical to the natural frequencies of the updated model listed in Table 3.9. The specified structural damping thus has a negligible effect on the natural frequencies of the rotor model. The complex mode shapes are thus also similar to those shown in Figure 3.11 to Figure 3.14.

### 3.2 FE Rotor System Models with Linear Bearings

Most of the analyses that can be performed using the Rotor Dynamics capability of MSC.Nastran are not suited to deal with nonlinear forces from a nonlinear journal bearing model - only the nonlinear transient and nonlinear frequency response analyses have this capability. The other analyses require linear elements and forces. The FE rotor system models were therefore first constructed with only linear bearings in order to be able to perform these analyses and explore its capabilities. The next steps in the construction of the 1D and 3D models were thus to add the bearings and coupling connection ${ }^{17}$ (i.e. the support structure) as linear springs and dampers and to define the rotor, its spin axis, spin speed(s) and imbalances (i.e. its rotor dynamic properties).

The base of the rotor system, the motor (and its half of the coupling) and the bearing pedestals/housings are assumed to be rigid compared to the flexible rotor, its bearings and the coupling connection.

### 3.2.1 1D Rotor System Model with Linear Bearings

Figure 3.16 shows a schematic free body diagram of the 1D rotor system model (see Appendix A. 2 for an enlarged version) with its support structure modelled as grounded linear spring and damper elements (CELAS1 and CDAMP1 elements respectively) located at the acting centres of the bearings and the coupling.


Figure 3.16-1D Rotor System Model with Linear Bearings

[^14]The vertical and horizontal bush bearing and journal bearing stiffness coefficients are denoted $k_{b y}$, $k_{b z}, k_{j y}$ and $k_{j z}$ respectively while the vertical, horizontal and torsional (axial) coupling connection stiffness coefficients are denoted $k_{c y}, k_{c z}$ and $k_{c \theta}$ respectively. The bearing and coupling connection damping coefficients are denoted similarly as $c_{b y}, c_{b z}, c_{j y}, c_{j z}, c_{c y}, c_{c z}$ and $c_{c \theta}$. The rotor spin speed is denoted $\omega$. The imbalance masses, radii and angular positions (in the local coordinate system of the rotor) at disks 1 and 2 are denoted $m_{u 1}, m_{u 2}, r_{u 1}, r_{u 2}, \theta_{u 1}$ and $\theta_{u 2}$ respectively. The unbalance forces resulting from these imbalances at disks 1 and 2 are denoted $F_{u 1}$ and $F_{u 2}$ respectively. Important grid point/node numbers are shown in blue on Figure 3.16. Specifically, nodes 5 and 48 are the rotor nodes at the acting centres of the bush and journal bearings respectively, while node 52 is the rotor node closest to the acting centre of the coupling connection (which is located at node 102).

Linearized journal bearings are usually modelled with four stiffness and four damping coefficients to include its cross-coupled stiffness and damping properties (refer to section 1.2.4), but elements with this capability are not (currently) available in MSC.Nastran. The journal bearings are therefore only modelled with two direct linear springs and two direct linear dampers. NOLIN1 elements may be used to include cross-coupling forces but these elements may only be employed in transient response analyses.

The stiffness and damping coefficients of the spring and damper elements are given in Table 3.10. The coefficients for the bush bearing and coupling connection were deduced from the stiffness and damping coefficients determined in sections 2.3.1, 2.3.2 and 2.3.3. Since no experiment was carried out to determine linearized properties for the journal bearing, arbitrary stiffness and damping values are used.

Table 3.10-Rotor System Model Linear Support Element Properties

| Support <br> Component | Axial Location <br> $[\mathrm{mm}]$ | $k_{y}=k_{Z}$ <br> $[\mathrm{~N} / \mathrm{mm}]$ | $k_{\theta}$ <br> $[\mathrm{Nmm} / \mathrm{rad}]$ | $c_{y}=c_{z}$ <br> $[\mathrm{Ns} / \mathrm{mm}]$ | $c_{\theta}$ <br> $[\mathrm{Nmm} . \mathrm{s} / \mathrm{rad}]$ |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: |
| Bush bearing | $x=42.5$ | 209 | - | 0.074 | - |
| Journal bearing | $x=435$ | 250 | - | 0.03 | - |
| Coupling | $x=-25$ | 154 | 6000 | 0.032 | 643 |

MSC.Nastran requires that the rotor be detached from the support structure for the correct calculation of the gyroscopic terms from the stiffness and inertial properties of the rotor elements. For example, the bush bearing elements should not be directly connected to node 5 which belongs to the rotor shaft and is located at the acting centre of the bush bearing. Multi-point constraint (MPC) elements, specifically RBEi elements, should be used to connect the rotor to its support structure elements. In this case, RBE2 rigid elements are used for this purpose. Extra nodes 100, 101 and 102 were added to the rotor model at the acting centres of the bush bearing, journal bearing and coupling connection respectively (refer to Figure 3.2 for the axial locations of these nodes). These nodes are connected as dependent nodes to nodes 5, 48 and 52 (which are part of the rotor) respectively via RBE2 elements. Nodes 100 and 101 are located at the same coordinates as nodes 5 and 48 respectively - for the sake of clarity, these nodes are shown at separate locations offset from the $x$-axis for both $y$ - and $z$-directions in Figure 3.16. The grounded spring and damper elements of the support structure are, in turn, connected to nodes 100, 101 and 102. These elements are scalar/point (zero-dimensional or OD) elements, but for the sake of clarity, they
are represented as 1D elements with two separate end-points in Figure 3.16. Figure 3.17 shows the 1D rotor system model in MSC.Patran. The RBE2 elements are shown as red lines with the dependent nodes as red circles.


Figure 3.17-1D Rotor System Model with Linear Bearings in MSC.Patran

The experimental rotor system has a small plastic bush, with two setscrews, located on the coupling side of the bush bearing housing with the purpose of acting as a constraint against axial movement. This constraint is included in the 1D rotor system model as an $x$-direction displacement constraint at node 5 . All rigid DOFs of the rotor are therefore constrained.

The spin direction is defined as the positive $x$-axis. The imbalances applied to disks 1 and 2 are defined in Table 3.11 - these values were arbitrarily chosen based on the magnitude of the imbalance applied to a rotor model of similar size developed by De Castro et al. (2008).

Table 3.11 - Rotor System Model Disk Imbalance Properties

| Disk | $m_{u}[t]$ | $r_{u}[\mathrm{~mm}]$ | $\left.\theta_{u}{ }^{\circ}\right]$ |
| :---: | :---: | :---: | :---: |
| 1 | $3 \times 10^{-7}$ | 30.75 | 0 |
| 2 | $5 \times 10^{-7}$ | 30.75 | 90 |

### 3.2.1.1 Normal and Complex Modes

To test that all elements are correctly connected and defined, the model is correctly constrained and to observe the effect of damping (rotor and support damping) on the natural frequencies of the model (without rotor rotation), both normal and complex modes (direct and modal) analyses were performed. MSC. Nastran solution sequences SOL 103, SOL 107 and SOL 110 were used - the BDF input files for these analyses are included in Appendices D.3.1 and D.3.2.

The first 14 undamped and first 15 damped natural frequencies, from the direct formulation, are listed in Table 3.12. The modal formulation yields almost identical results.

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems

Table 3.12-1D Rotor System Model with Linear Bearings Natural Frequencies

| Normal Modes |  |  | Complex Modes |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Mode | $\boldsymbol{f}_{n}[\mathrm{~Hz}]$ | Mode Type | Mode | $f_{d}[\mathrm{~Hz}]$ | Mode Type |
| 1 \& 2 | 32.307 | Lateral | 1 | 0 | Torsional |
| 3 | 32.591 | Torsional | 2 \& 3 | 32.312 | Lateral |
| 4 \& 5 | 130.95 | Lateral | 4 | 60.631 | Torsional |
| 6 | 238.75 | Torsional | 5 \& 6 | 131.33 | Lateral |
| 7 \& 8 | 268.4 | Lateral | 7 | 250.32 | Torsional |
| $9 \& 10$ | 292.32 | Lateral | 8 \& 9 | 271.21 | Lateral |
| 11 \& 12 | 474.82 | Lateral | 10 \& 11 | 282.99 | Lateral |
| 13 \& 14 | 688.41 | Lateral | 12 \& 13 | 475.16 | Lateral |
|  |  |  | 14 \& 15 | 690.39 | Lateral |

The frequencies of corresponding modes are colour-matched in this table. The first complex mode (which is a torsional mode) is at 0 Hz which indicates an over-damped mode. Symmetric lateral modes are once again encountered since the rotor is axisymmetric and the support structure is isotropic. The damped lateral natural frequencies are generally slightly higher than the undamped frequencies due to effective lateral stiffening from the support damping. The symmetric damped lateral mode shape pairs are included in Appendix F.7.

### 3.2.1.2 Critical Speeds

The critical speeds of the rotor system are determined by performing a synchronous complex modes analysis. BDF input files for the direct and modal formulations (SOL 107 and SOL 110 respectively), with all the required rotor dynamic entries, were created using MSC.Patran for the rotor model with damping and are included in Appendix D.3.3. To include the rotor material damping (specified as structural damping in the material properties) in the rotor dynamic analyses, the following MSC.Nastran entry was added to the Bulk Data Sections of the input files:

PARAM, WR4,500.0
This entry specifies the "average" rotor spin speed (in $\mathrm{rad} / \mathrm{s}$ ) for the calculation of the equivalent viscous rotor damping used in the EOMs.

The first 9 resulting critical speeds (for the model with both rotor and support damping), from the direct formulation, are given in Table 3.13. The modal formulation yields almost identical results.

Table 3.13-1D Rotor System Model with Linear Bearings Critical Speeds

| Mode | $\boldsymbol{f}_{\boldsymbol{c}}[\mathbf{H z}]$ | $\boldsymbol{N}_{\boldsymbol{c}}[\mathbf{r p m}]$ | Mode Type |
| :---: | :---: | :---: | :--- |
| $\mathbf{1}$ | $\mathbf{0}$ | $\mathbf{0}$ | Torsional |
| $\mathbf{2}$ | 32.218 | 1933.08 | Backward whirling |
| $\mathbf{3}$ | 32.405 | 1944.3 | Forward whirling |
| $\mathbf{4}$ | 60.63 | 3637.8 | Torsional |
| $\mathbf{5}$ | 123.37 | 7402.2 | Backward whirling |
| $\mathbf{6}$ | 139.97 | 8398.2 | Forward whirling |
| $\mathbf{7}$ | $\mathbf{2 5 0 . 2 7}$ | $\mathbf{1 5 0 1 6 . 2}$ | Torsional |
| $\mathbf{8}$ | 245.69 | 14741.4 | Backward whirling |
| $\mathbf{9}$ | 294.58 | 17674.8 | Forward whirling |

The backward and forward whirling (BW and FW respectively) critical frequency pairs (originating from the same frequency at zero spin speed) are colour-matched in this table. The first three BW and FW critical speed mode shape pairs are shown in Figure 3.18 to Figure 3.20. The light blue lines indicate the undeformed geometry while the darker blue lines indicate the deformed geometry and the RBE2 elements are shown in green. The whirling motion of these mode shapes may be animated in MSC.Patran, but these animations are difficult to capture using static images - grey dotted lines, with arrows indicating the whirl direction, have therefore been added to single nodes on these figures to illustrate the whirling paths of these nodes.


Figure 3.18-Linear 1D Rotor System $1^{\text {st }}$ BW \& FW Lateral Critical Speed Mode Shape Pair


Figure 3.19-Linear 1D Rotor System $2^{\text {nd }}$ BW \& FW Critical Speed Mode Shape Pair


Figure 3.20-Linear 1D Rotor System $3^{\text {rd }}$ BW \& FW Critical Speed Mode Shape Pair

### 3.2.1.3 Campbell Diagram

A Campbell diagram is generated by performing an asynchronous complex modes analysis at intermittent spin speed values within the desired spin speed range. Only the direct formulation (SOL 107) with the Upper Hessenberg method (instead of the default Complex Lanczos method) seems to work for generating Campbell diagrams. Since the version of MSC.Patran used does not support the feature to generate a Campbell Diagram, a workaround had to be found to create a BDF input file for this analysis. The BDF file, along with descriptions of entries that had to be manually added or modified (and also the additional/modified entries themselves), are included in Appendix D.3.4.

Another drawback of MSC.Patran not supporting the feature to generate Campbell Diagrams is that the post-processing of the results from this type of simulation is also not supported. A decision was therefore made to utilise MSC.Nastran's feature to output results into a punch (*.pch) file ${ }^{18}$ (additional to the usual output files) and then extract the results from this text-based output file using MATLAB. The results can then be visualised as desired through MATLAB. An input file generated by MSC.Patran for this type of analysis, by default, contains a request on the VECTOR Case Control command to output the eigenvector results from all the nodes of the model. The SET Case Control command allows one to request the output of results of only a specific set of nodes, elements, etc. This is useful for generating only the required results. Since only the natural whirl frequencies at each spin speed are of interest in this type of analysis, this feature was utilised through the following additions and modifications to the Case Control Section of the input file:

```
SET 1000 = 5
VECTOR(PRINT, PUNCH, SORT1, REAL)=1000
$SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
```

The addition of these entries results in only the eigenvalue and eigenvector results for node 5 being written to the output files (including the PCH file). A MATLAB program CampbellDiag_CompPCH.m (included in Appendix C.1) was written to extract, process and plot (as a Campbell diagram) the natural whirl frequencies at each spin speed value from the PCH file. The spin speed values are

[^15]however not included in the PCH file and have to be specified by the user of the program (during runtime). For plotting a Campbell diagram, the rotor modes need to be tracked as the eigenvalues of the modes, calculated at each spin speed, may change ordering - MSC.Nastran has a built-in feature for this which does not seem to be working in this case. A mode tracking algorithm was therefore included in CampbellDiag_CompPCH.m to correct the eigenvalue ordering based on gradient changes in the modal frequency lines.

The resulting Campbell diagram from this analysis is shown in Figure 3.21 (a Campbell diagram from a model where damping is ignored is included in Appendix F.7). The coloured lines are the natural whirl and torsional frequencies as functions of spin speed. The torsional natural frequencies remain constant at all spin speeds and are therefore horizontal lines on the Campbell diagram. The FW natural frequencies tend to increase with spin speed while the BW natural frequencies tend to decrease with spin speed. The dotted grey line indicates where whirl frequency equals spin speed. The spin speeds at which this line crosses the natural whirl mode lines are the critical speeds. These critical speeds correspond with the values listed in Table 3.13.


Figure 3.21-1D Rotor System Model with Linear Bearings Campbell Diagram

### 3.2.1.4 Frequency Response

The steady state response of the rotor system due to unbalance may be determined by performing a synchronous frequency response analysis. The BDF input files for the direct and modal formulations (SOL 108 and SOL 111 respectively), along with a description of entries that had to be manually added or modified, are included in Appendix D.3.5.

Post-processing of the results from this type of analysis may be done using the Graph tool of MSC.Patran - this feature is however limited (e.g. phase data cannot be viewed). Alternatively, the results may be exported from MSC.Patran into a spread sheet, but the format in which the results are stored makes it difficult to create charts from the data. The PCH file output format is therefore utilised here again (as described in the preceding section) along with the SET Case Control command to output the displacement results at only desired rotor nodes. The following additions and modifications were made to the Case Control Sections of the input files:

```
SET 1000=5,21, 32,48,52
DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
$SPCFORCES(PRINT, PUNCH, SORT1,REAL)=ALL
```

With these entries, only the displacement results at the acting centres of the bearings and the coupling connection as well as the two disk centres are written to the output files. A MATLAB program RotorDispl_FreqRespPCH.m (included in Appendix C.1) was written to extract, plot and store the displacement frequency response results from PCH files. This program allows the user to specify (during runtime) for which frequency range and at which nodes to extract the displacement results or to extract the results of all nodes. The Cartesian results coordinate system is required to be stationary and orientated as shown in Figure 3.16. Both lateral and axial angular displacement results are extracted and the user may choose (during runtime) to plot the axial results in addition to the lateral results. Another program RotorDispl_FreqRespCSV.m (also included in Appendix C.1) was written to load and plot frequency response results stored in CSV file format by RotorDispl_FreqRespPCH.m.

The steady state rotor displacement frequency response results, from the direct formulation, at the acting centres of the bearings and the centre of the disk 1 for spin speeds $0-200 \mathrm{~Hz}(0-$ 12000 rpm ) in the rotor system global coordinate system are shown in Figure 3.22 to Figure 3.24 (results from a model where damping is ignored is included in Appendix F.7). The modal formulation yields almost identical lateral displacement results. The peaks in displacement magnitude and associated phase changes at approximately 32.5 Hz ( 1950 rpm ) and 140 Hz $(8400 \mathrm{rpm})$ indicate resonances. These are the first two FW critical speeds which correspond with the values listed in Table 3.13. These results also show that the centre of disk 1 whirls with a larger amplitude than the acting centres of the bearings at the first critical speed, while the whirl amplitudes at these three points are of the same order at the second critical speed - this corresponds with the mode shapes of these critical speeds (refer to Figure 3.18 and Figure 3.19).


Figure 3.22 - Linear 1D Rotor System Bush Bearing Displacement Frequency Response


Figure 3.23-Linear 1D Rotor System Journal Displacement Frequency Response


Figure 3.24 - Linear 1D Rotor System Disk 1 Displacement Frequency Response

Figure 3.25 shows the steady state displacement orbits of node 48 (journal bearing acting centre) at intermittent spin speeds (cascade orbits) between 25 and 40 Hz while Figure 3.26 shows the orbits between 120 and 160 Hz . The dot on each orbit represents a complete shaft revolution note that the position of these dots on the orbits change as the spin speed passes through a critical speed, indicating a phase change.


Figure 3.25 - Linear 1D Rotor System Journal Cascade Displacement Orbits ( $\mathbf{2 5 - 4 0} \mathrm{Hz}$ )


Figure 3.26 - Linear 1D Rotor System Journal Cascade Displacement Orbits (120-160 Hz)

### 3.2.1.5 Transient Response

The transient response of the rotor system due to unbalance may be determined by performing a nonlinear transient response analysis. Note that since the rotor system model has only linear bearings, so far, a linear transient response analysis would also be adequate (provided that displacements are small). However, when the rotor system is considered with a nonlinear journal bearing (as in section 3.5), a nonlinear analysis is required. So, for the sake of generalization and limiting the range of solution sequences considered that yield more or less the same results, the rotor system with linear bearings was also analysed using nonlinear solution sequences (SOL 129 and SOL 400 - refer to subsection two of section 1.2.5.3, for a brief discussion on these solution sequences). Run-up simulations were performed for the rotor running up to 10000 rpm from standstill at a constant rate ( $1000 \mathrm{rpm} / \mathrm{s}$ ) over 10 seconds. This run-up 'profile' was chosen so that at least the first two FW natural frequencies are excited within a reasonably short simulation time. The BDF input files for SOL 129 and SOL 400, along with descriptions of entries that had to be manually added or modified, are included in Appendix D.3.6.

## * Displacement Results

Post-processing of the results from this type of analysis may be performed using the Graph tool of MSC.Patran, but the limited functionality of this tool once again proves inadequate for postprocessing of these results (e.g. displacement orbits cannot be created). The PCH file output format is therefore utilised here once more along with the SET Case Control command to retrieve the displacement results at the desired rotor nodes. The following additions and modifications were made to the Case Control Sections of the input files:

```
SET 1000=5,21,32,48,52
```

DISPLACEMENT (PRINT, PUNCH, SORT1, REAL) $=1000$
\$SPCFORCES (PRINT, PUNCH, SORT1, REAL) =ALL
A MATLAB program RotorDispl_TransPCH.m (included in Appendix C.1) was written to extract the displacement response results from PCH files. The user may specify (during runtime) for which time range and nodes to extract results. As with RotorDispl_FreqResp.m, this program requires the results coordinate system to be orientated as shown in Figure 3.16. The PCH file may contain other result types (e.g. forces) since the program searches for the correct results to extract. The user is also provided with the option to plot the axial angular displacement results. The program RotorDispl_TransCSV.m (also included in Appendix C.1) may be used to load and plot the transient displacement results stored in CSV file format by RotorDispl_TransPCH.m.

The SOL 400 transient rotor displacement results (vertical and horizontal displacements and orbits) at the acting centres of the bearings and the centre of the disk 1 in the rotor system global coordinate system are shown in Figure 3.27 to Figure 3.30 (the SOL 129 results and the SOL 400 results from a model where damping is ignored are included in Appendix F.7). Figure 3.31 shows the 2D displacement of the centre of disk 1 as a function of time. The peaks in vibration magnitude at approximately $2.1 \mathrm{~s}(2100 \mathrm{rpm})$ and $8.3 \mathrm{~s}(8300 \mathrm{rpm})$ indicate the first two FW critical speeds these values, however, differ slightly from the values in Table 3.13 due to the finite run-up rate (transient effects). As with the frequency response results (from section 3.2.1.4), these results also show that the centre of disk 1 whirls with a larger amplitude than the acting centres of the bearings at the first critical speed, while the whirl amplitudes at these three points are of the same order at the second critical speed.


Figure 3.27 - Linear 1D Rotor System Run-Up Bush Bearing Displacements


Figure 3.28 - Linear 1D Rotor System Run-Up Journal Displacements


Figure 3.29-Linear 1D Rotor System Run-Up Disk 1 Displacements


Figure 3.30 - Linear 1D Rotor System Run-Up Displacement Orbits

2D Displacement of Node 21 as a Function of Time:


Figure 3.31 - Linear 1D Rotor System Run-Up Disk 1 2D Displacement

Figure 3.32 to Figure 3.34 show the transient displacement of node 48 (journal bearing acting centre) only between 1 and $3 s$ ( 1000 and 3000 rpm ).


Figure 3.32 - Linear 1D Rotor System Run-Up Journal Displacements (1-3 s)


Figure 3.33 - Linear 1D Rotor System Run-Up Journal Displacement Magnitude (1-3 s)

2D Displacement of Node 48 as a Function of Time:


Figure 3.34 - Linear 1D Rotor System Run-Up Journal 2D Displacement (1-3 s)

A MATLAB program RotorDisplSpectral.m (included in Appendix C.1) was written to calculate and plot a DFT, power spectral density (PSD) and spectrogram (a representation of the magnitude of the short-time Fourier transform or STFT of a signal) for the displacements of each node extracted by RotorDispl_TransPCH.m. The DFTs and PSDs are calculated within custom MATLAB functions dft.m and psde.m respectively (both included in Appendix B.5) - the latter by Heyns (2005). If an inconsistent sampling frequency is detected by this program, it resamples the signals with a constant sampling frequency specified by the user (during runtime). The user may also choose to resample the signals in case of a constant sampling frequency. The spectrograms are created with a user-specified window size.

Figure 3.35 shows a spectrogram of the vertical displacement ${ }^{19}$ of node 5 (bush bearing acting centre). The dark red colour on this spectrogram represents the highest vibration amplitude while dark blue represents the lowest vibration amplitude. The same frequency content (as revealed by this spectrogram of node 5) is present in the displacements of the other nodes as well. The inclined red line at frequencies equal to the spin speed of the rotor indicates synchronous whirling due to

[^16]the unbalance loads at disks 1 and 2 with the dark red portions indicating resonance (critical speeds). The horizontal red line at approximately 33 Hz ( 1980 rpm ) indicates the transient whirling (which fades over time due to damping) from the excitation of the first FW critical speed by the unbalance loads.


Figure 3.35 - Linear 1D Rotor System Run-Up Bush Bearing Vertical Displacement Spectrogram

## * Support Element Force Results

The support structure spring and damper element forces may also be selected for output to the PCH files by adding the following entries to the Case Control Sections of the input files:

```
SET 2000=60,61,62,63,64,65,66,70,71,72,73,74,75,76
FORCE(PRINT, PUNCH, SORT1, REAL, BILIN)=2000
```

A MATLAB program RotorForce1DElem_TransPCH.m (included in Appendix C.1) was written to extract the element force results from CELAS1 and CDAMP1 elements. The user may specify for which time range and elements to extract results. The program assumes the results coordinate system is orientated as shown in Figure 3.16. After extraction of the PCH file data, the user is given the option to pair the force results from individual elements into horizontal and vertical force pairs with a user-defined label for each pair to be used for plotting. The program can also individually plot the results from each element, including the torque from a torsional spring/damper element. The paired force results may be stored in CSV file format (one file per force pair).

Another MATLAB program RotorMultiPlot_TransCSV.m (also included in Appendix C.1) was written to load any number of CSV files containing either transient displacement results (e.g. from RotorDispl_TransPCH.m), transient force results (e.g. from RotorForce1DElem_TransPCH.m) or both and plot them on the same figures. These signals may have different sampling frequencies and time
limits. When both displacement and force signals are present, these signals are placed on different $y$-axes of the same figures. The user is required to specify time limits for plotting and enter a label for each signal during runtime. The user may choose to specify the line style, marker, line width and line colour for each signal or use the defaults. Clearance lines (portraying, for example, the radial clearance of a bearing housing) may optionally be added if displacements are present ${ }^{20}$.

The bush and journal bearing spring and damper element forces (between 1.5 and 2 s ; from SOL 400) are shown in Figure 3.36 and Figure 3.37 respectively (created using RotorMultiPlot_TransCSV.m). Figure 3.38 shows the 2D spring and damper forces for the journal bearing while Figure 3.39 (created using RotorForce1DElem_TransPCH.m) shows the coupling connection torsional spring torque. The forces acting on the rotor are the negatives of these (internal element) forces. Note that the damper forces are scaled by factors of 3 and 10 for better visibility. The spring and damper element forces of the bearings are out of phase by $90^{\circ}$ while the spring forces are in-phase with the acting centre displacements, as expected. The total vertical and horizontal bearing forces can be determined by adding together the spring and damper forces in these respective directions. Figure 3.39 indicates coupling between lateral and torsional vibrations.


Figure 3.36 - Linear 1D Rotor System Run-Up Bush Bearing Forces

[^17]

Figure 3.37 - Linear 1D Rotor System Run-Up Journal Bearing Forces


Figure 3.38-Linear 1D Rotor System Run-Up 2D Journal Bearing Forces


Figure 3.39 - Linear 1D Rotor System Run-Up Coupling Connection Spring Torque

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems

## * Unbalance Excitation Forces

The unbalance excitation forces may also be selected for output to the PCH files by adding the following entries to the Case Control Sections of the input files:

```
SET 4000=21,32
OLOAD(SORT1, PUNCH , REAL)=4000
```

A MATLAB program RotorForceExc_TransPCH.m (included in Appendix C.1) was written to extract applied forces, such as the forces from UNBALNC entries, from PCH files. Once again, the user may specify for which time range and nodes to extract these forces, the results coordinate system needs to be orientated as shown in Figure 3.16 and the PCH file may contain other result types. Plotting of applied axial moments (if any) may optionally be requested.

The unbalance forces for disks 1 and 2 are shown in Figure 3.40 to Figure 3.42 for $0-3 s$ $(0-3000 \mathrm{rpm})$ of the run-up simulation. The magnitudes of these forces increase exponentially with spin speed and the force at disk 2 is larger than at disk 1 due to the larger imbalance at disk 2. These forces are also seen to rotate with the shaft (i.e. at the same frequency).


Figure 3.40 - Disks 1 \& 2 Run-Up Unbalance Forces


Figure 3.41 - Disks 1 \& 2 Run-Up Unbalance Force Magnitudes


Figure 3.42 - Disks 1 \& 2 Run-Up 2D Unbalance Forces

### 3.2.1.6 Gravity Loading and Initial Conditions

The inclusion of a nonlinear journal bearing in the rotor system model (as discussed in section 3.5) necessitates the inclusion of gravity loading since the stiffness and damping properties of a nonlinear bearing is dependent on the bearing load. The capabilities of applying gravity loading in MSC.Nastran and the effects thereof were first investigated on the rotor system models with linear bearings.

## * Gravity as Inertial Loading

Gravity (and other acceleration loading) may be applied to a structure using the MSC.Nastran GRAV Bulk Data entry. This entry allows the specification of gravity loading by defining acceleration vectors (termed Inertial Loading in MSC.Patran). Gravitational acceleration of $9810 \mathrm{~mm} / \mathrm{s}^{2}$ in the negative $y$-direction was applied to the rotor model within MSC.Patran.

The static equilibrium state of the rotor system under this gravity loading does not correspond with the undeformed state of the rotor system (where the rotor spin axis coincides with the line of centres of the bush and journal bearings). Thus, before a transient run-up response analysis is performed, where the rotor starts spinning from standstill, the equilibrium state of the system due to gravity loading needs to be determined to be applied as initial conditions for the transient response analysis. Since the system analysed is linear, its equilibrium state may easily be determined by performing a linear static analysis (using MSC.Nastran solution sequence SOL 101). There are different ways of applying the results from such an analysis as initial conditions for a transient response analysis - the following steps outline a simple way of doing this for the 1D rotor system model within MSC.Patran:

1. Perform a linear static analysis on the rotor system with gravity loading (using SOL 101) to determine its equilibrium deformation
2. Attach the results from this analysis to MSC.Patran (including rotational nodal results)
3. Create a Vector plot (using the Marker tool of MSC.Patran) of the translational DOF displacement results
4. Create a Spatial Field with the following properties:

FEM Field Definition: Continuous
Field Type: Vector
Options/Extrapolation Option: Linear Extrapolation
5. Repeat steps 3 and 4 for the rotational DOF displacement results
6. Delete the results attachment from the MSC.Patran database
7. Create Initial Displacements using the Spatial FEM Fields created in step 4 and apply it to all the nodes of the rotor (excluding support nodes and dependent MPC nodes).

The last step (step 7) adds initial displacements to the model using the MSC.Nastran TIC Bulk Data entry.

A nonlinear transient response analysis (SOL 400) was performed to determine the unbalance response of the rotor system under gravity loading during run-up ( $0-10000 \mathrm{rpm}$ ), including initial conditions. The MSC.Nastran BDF input file for the linear static analysis (SOL 101) performed for step 1 is included in Appendix D.3.7. Figure 3.43 shows the static deformation of the rotor system due to the gravity loading (unbalance and the gyroscopic effect are excluded since the rotor is at standstill).


Figure 3.43-1D Rotor System Model with Linear Bearings Static Deformation due to Gravity

The maximum static lateral displacement of the rotor centreline is 0.2594 mm in the negative $y$ direction at $x=233.2 \mathrm{~mm}$ (node 26). The resulting support deflections and reaction forces (from the bearings action on the shaft) are given in Table 3.14.

Table 3.14-1D Rotor System Model Linear Support Deflections \& Reaction Forces due to Gravity

| Support Element | $y[\mathrm{~mm}]$ | $F_{y}[\mathrm{~N}]$ |
| :--- | :---: | :---: |
| Bush bearing | -0.066108 | 13.8167 |
| Journal bearing | -0.037442 | 9.36534 |
| Coupling connection | 0.005882 | -3.33421 |

The BDF input file (created using MSC.Patran) for the nonlinear transient response analysis is included in Appendix D.3.7. The entry additions/modifications from the preceding section (3.2.1.5) were also incorporated into this file. The resulting displacements at the acting centres of the bush and journal bearings are shown in Figure 3.44 and Figure 3.45 respectively. These are compared to the displacement results from the preceding section (without gravity loading). The displacements from the two simulations seem to be almost exactly the same, except that the vertical displacements from the simulation with gravity loading are offset by the static deflections due to gravity (refer to Table 3.14 above).


Figure 3.44 - Linear 1D Rotor System with Gravity: Bush Bearing Displacements


Figure 3.45 - Linear 1D Rotor System with Gravity: Journal Displacements

## * Simplified Gravity Loading

When gravity is applied as an inertial loading, it is applied to each element in the model which adds a substantial amount of extra computation time to a simulation - especially if nonlinear journal bearing forces are also present. For this reason, a simplified gravity loading was formulated consisting of only three vertical forces at the acting centres of the bearings and the coupling connection. These forces are determined by assuming that the rotor is rigid (thus, only the support structure deforms), applying gravity loading as an external point force at the C.G. of the rotor and calculating the support reaction forces. These reaction forces are then applied as the equivalent
simplified gravity loading ${ }^{21}$. Figure 3.46 shows a free body diagram of the rigid rotor with external gravity force $m g$ and the reaction forces at the acting centres of the bush bearing, journal bearing and coupling connection denoted as $F_{b y}, F_{j y}$ and $F_{c y}$ respectively. The displacements in the $y$ direction at these support acting centres are denoted $y_{b}, y_{j}$ and $y_{c}$ respectively. Dimensions $l_{b}, l_{j}$, $l_{c}$ and $l_{C G}$ are as shown in this figure.


Figure 3.46 - Rigid Rotor with Gravity Loading as an External Force

From Newton's equilibrium equations $\sum F_{y}=0$ and $\sum M_{z O}=0$ (moments about the origin):

$$
\begin{gather*}
F_{c y}+F_{b y}+F_{j y}-m g=0  \tag{3.3}\\
-F_{c y} l_{c}+F_{b y} l_{b}+F_{j y} l_{j}-m g l_{c G}=0
\end{gather*}
$$

But, the bearings and coupling connection are modelled as linear springs, thus:

$$
\begin{equation*}
F_{c y}=-k_{c} y_{c} ; \quad F_{b y}=-k_{b} y_{b} ; \quad F_{j y}=-k_{j} y_{j} \tag{3.4}
\end{equation*}
$$

Substituting these equations into eqns. (3.3) yields:

$$
\begin{gather*}
k_{c} y_{c}+k_{b} y_{b}+k_{j} y_{j}=-m g \\
-k_{c} l_{c} y_{c}+k_{b} l_{b} y_{b}+k_{j} l_{j} y_{j}=-m g l_{C G} \tag{3.5}
\end{gather*}
$$

From the assumption that the rotor is rigid:

$$
\begin{gather*}
\frac{y_{b}-y_{c}}{l_{c}+l_{b}}=\frac{y_{j}-y_{c}}{l_{c}+l_{j}} \\
\therefore\left(l_{b}-l_{j}\right) y_{c}+\left(l_{c}+l_{j}\right) y_{b}-\left(l_{c}+l_{b}\right) y_{j}=0 \tag{3.6}
\end{gather*}
$$

Eqns. (3.5) and (3.6) may be written in matrix format as:

[^18]\[

\left[$$
\begin{array}{ccc}
k_{c} & k_{b} & k_{j}  \tag{3.7}\\
-k_{c} l_{c} & k_{b} l_{b} & k_{j} l_{j} \\
l_{b}-l_{j} & l_{c}+l_{j} & -l_{c}-l_{b}
\end{array}
$$\right]\left\{$$
\begin{array}{l}
y_{c} \\
y_{b} \\
y_{j}
\end{array}
$$\right\}=-\left\{$$
\begin{array}{c}
m g \\
m g l_{C G} \\
0
\end{array}
$$\right\}
\]

and the support structure deflections may be calculated from this equation after which the reaction forces may be calculated from eqns. (3.4). The simplified gravity loading forces are equal to these reaction forces, but in opposite directions. A MATLAB program RigidRotorGravEquil.m was written to calculate the reaction forces and is included in Appendix C.2. The results from these calculations for the 1D rotor system model are given in Table 3.15.

Table 3.15-1D Rigid Rotor System Linear Support Deflections \& Reaction Forces due to Simplified Gravity

| Support Element | $\boldsymbol{y}[\mathrm{mm}]$ | $\boldsymbol{F}_{y}[\boldsymbol{N}]$ |
| :--- | :---: | :---: |
| Bush bearing | -0.02668 | 5.577 |
| Journal bearing | -0.04228 | 10.5696 |
| Coupling connection | -0.024 | 3.6964 |

These forces may be applied to the rotor using MSC.Nastran's FORCE Bulk Data entries (this can be done within MSC.Patran). A nonlinear transient response analysis (SOL 400) was performed to determine the unbalance response of the rotor system with this simplified gravity loading during run-up ( $0-10000 \mathrm{rpm}$ ), including initial conditions. The same steps that were used to determine and apply equilibrium deformations from a linear static analysis as initial conditions to a nonlinear transient response analysis for the rotor system with gravity as an inertial load (refer to the preceding subsection - Gravity as Inertial Loading) were used here again. The MSC.Nastran BDF file for the linear static analysis (SOL 101) performed for step 1 is included in Appendix D.3.8. Figure 3.47 shows the static deformation of the rotor system due to the simplified gravity loading (shown in orange). Note that the rotor remains straight (unbent) under this loading.


Figure 3.47-1D Rotor System Model with Linear Bearings Static Deformation due to Simplified Gravity

The BDF file (created using MSC.Patran) for the nonlinear transient response analysis is also included in Appendix D.3.8. The entry additions/modifications from the preceding section (3.2.1.5) were once again incorporated into this file. The resulting displacements at the acting centres of the bush and journal bearings are shown in Figure 3.48 and Figure 3.49 respectively. Once again, the DC ( 0 Hz or average) components of the vertical displacement signals are equal to the static deformations due to the simplified gravity loading (refer to Table 3.15).

Vertical Displacement Comparison:



Figure 3.48 - Linear 1D Rotor System with Simplified Gravity: Bush Bearing Displacements


Figure 3.49 - Linear 1D Rotor System with Simplified Gravity: Journal Displacements

### 3.2.1.7 Flat Surface Rubbing

Figure 3.50 illustrates a simple model for a circular shaft rubbing with a horizontal flat surface located above the shaft.


Figure 3.50 - Simple Flat Surface Rubbing Model
$\boldsymbol{F}_{n}$ and $\boldsymbol{F}_{t}$ are the normal (impact or contact) and tangential (frictional) rubbing forces, respectively, acting on the shaft with radius $r$ spinning at angular velocity $\omega$ in the direction as shown on in Figure 3.50. The vertical clearance between the undeformed shaft and the rubbing surface is denoted $c$. A simple way of modelling the rotor-stator contact (normal) force is by defining this force as a function of the amount by which the rotor centreline displacement exceeds (or penetrates) the vertical clearance. The penetration $p$ is given by:

$$
p(y)=\left\{\begin{array}{cc}
0 & y \leq c  \tag{3.8}\\
y-c & y>c
\end{array}\right.
$$

and the contact force as a function of this penetration ${ }^{22}$ :

$$
\begin{equation*}
\left|\boldsymbol{F}_{n}\right|=F_{n}=f(p(y)) \tag{3.9}
\end{equation*}
$$

The tangential friction force, modelled as a Coulomb friction force, is given by:

$$
\begin{equation*}
\left|\boldsymbol{F}_{t}\right|=F_{t}=\mu_{f} F_{n} \tag{3.10}
\end{equation*}
$$

where $\mu_{f}$ is the dynamic friction coefficient. The resultant rubbing forces and rubbing moment acting at the shaft centreline are:

$$
\begin{gather*}
F_{r y}=-F_{n} \\
F_{r z}=-F_{t}=-\mu_{f} F_{n}  \tag{3.11}\\
M_{r x}=-r F_{t}=-r \mu_{f} F_{n}
\end{gather*}
$$

[^19]Figure 3.51 shows an arbitrary contact force function (as a function of rotor-stator penetration) used to demonstrate this rubbing model.


| $\boldsymbol{p}[\mathrm{mm}]$ | $F_{n}[\mathrm{~N}]$ |
| :---: | :---: |
| 0 | 0 |
| 0.01 | 50 |
| 0.02 | 120 |
| 0.03 | 210 |
| 0.04 | 320 |
| 0.05 | 450 |
| 0.06 | 600 |
| 0.07 | 770 |
| 0.08 | 960 |
| 0.09 | 1170 |
| 0.1 | 1400 |

Figure 3.51 - Rotor-Stator Contact Force as a Function of Penetration

MSC.Nastran's NOLIN1 Bulk Data entries may be used to define the rubbing forces as functions of displacement (in tabular form with TABLED1 entries) for a transient response analysis. This rubbing model is applied halfway between disks 1 and 2 (node 27 - where the shaft radius is 4.75 mm ), with a clearance of 0.2 mm and a friction coefficient of 0.3 , by incorporating the extra entries included in Appendix D.3.9 into the BDF input files included in Appendix D.3.6 from the nonlinear transient response analyses discussed in section 3.2.1.5. Due to the nature of the rubbing forces (highly nonlinear short pulse/impact loading) and the method by which they are included in the model (as external forces), modifications had to be made to the TSTEPNL entry in the SOL 400 BDF file. These modifications are also discussed and included in Appendix D.3.9.

The SOL 400 transient rotor displacement results at node 27 (halfway between disks 1 and 2) are shown in Figure 3.52 to Figure 3.55. A MATLAB program RotorDisplClearance.m (included in Appendix C.1) was written to add visual clearance representations to transient displacement figures created by RotorDispl_TransPCH.m or RotorDispl_TransCSV.m. Two types of clearances are supported: radial and top flat surface. The former refers to the clearance between the rotor shaft and a stationary housing (such as a bearing housing) while the latter refers to the clearance between the rotor shaft and a flat horizontal surface located above the shaft. The user is required to specify (during runtime) at which nodes to add clearance information and what type of clearance information to add. Clearance lines are added to the vertical displacement, horizontal displacement and orbit figures while clearance surfaces are added to the 2D displacement figures (along with legends). Figure 3.52, Figure 3.53 and Figure 3.55 include clearance information. These results show that the rotating shaft comes into contact with the stationary flat surface as the rotor passes through its first critical speed. Figure 3.53 shows the distorted orbits caused by this rubbing. Clearance penetration occurs where the displacement breaches the clearance lines/plane, as clearly shown in Figure 3.55. The angular displacement response, mainly due to frictional forces, is shown in Figure 3.54.


Figure 3.52 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Displacements


Figure 3.53-Linear 1D Rotor System Flat Surface Rubbing Node 27 Displacement Orbit


Figure 3.54 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Angular Displacement

2D Displacement of Node 27 as a Function of Time:


Figure 3.55 - Linear 1D Rotor System Flat Surface Rubbing Node 27 2D Displacement

Figure 3.56 shows a spectrogram of the vertical displacement of node 27. Harmonic terms (indicating distortion of the usual harmonic unbalance response) and a DC offset are noted where rubbing occurs.


Figure 3.56 - Linear 1D Rotor System Flat Surface Rubbing Node 27 Vertical Displacement Spectrogram

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems

The following entries were also added to the Case Control Section of the input file in order to select the nonlinear contact forces for output to the PCH file:

```
SET 4000=27
NLLOAD (PUNCH)=4000
```

A MATLAB program RotorForceNL_TransPCH.m (included in Appendix C.1) was written to extract nonlinear forces from PCH files. As with the other transient response results processing MATLAB programs, the user may specify for which time range and nodes to extract these forces, axial moments figures may be requested and the PCH file may contain other result types.

Figure 3.57 and Figure 3.58 show both the displacement and rubbing force (acting on the shaft) results on the same figures (with the displacement $y$-axis on the left and the force $y$-axis on the right; created using RotorMultiPlot_TransCSV.m), clearly indicating that the rubbing forces act on the shaft when it breaches the clearance gap once every whirl revolution. Figure 3.59 shows the axial rubbing moment acting on the shaft due to the frictional rubbing force.


Figure 3.57 - Linear 1D Rotor System Flat Surface Rubbing Forces \& Node 27 Displacements


Figure 3.58 - Linear 1D Rotor System Flat Surface Rubbing Force \& Node 27 Displacement Magnitudes


Figure 3.59-Linear 1D Rotor System Flat Surface Rubbing Node 27 Axial Moment

### 3.2.1.8 Radial Rubbing

MSC.Nastran has a built-in radial rubbing model to simulate contact between any circular shaft enclosed by a cylindrical housing. This model is invoked with the NLRGAP entry and follows the same idealized approach as the flat surface rubbing model described in the preceding section (3.2.1.7). Figure 3.60 illustrates this model.


Figure 3.60 - Simple Radial Rubbing Model
$F_{n}$ and $F_{t}$ are once again the normal (contact) and tangential (frictional) rubbing forces acting on the shaft with radius $r$ at spin speed $\omega$. The radial clearance between the shaft and housing is denoted $c$. The shaft centreline eccentricity from the housing centre is given by:

$$
\begin{equation*}
e=\sqrt{y^{2}+z^{2}} \tag{3.12}
\end{equation*}
$$

The angle between the positive $y$-axis and the line of centres of the housing and shaft centres, angle $\theta_{r}$, may be calculated from:

$$
\begin{equation*}
\cos \theta_{r}=\frac{z}{e} \quad \text { and } \quad \sin \theta_{r}=\frac{y}{e} \tag{3.13}
\end{equation*}
$$

The radial penetration is given by:

$$
p(e)=\left\{\begin{array}{cc}
0 & e \leq c  \tag{3.14}\\
e-c & e>c
\end{array}\right.
$$

The contact force is once again a function of the penetration. This contact force and the Coulomb friction force are given by:

$$
\begin{equation*}
\left|\boldsymbol{F}_{n}\right|=F_{n}=f(p(e)) \quad \text { and } \quad\left|\boldsymbol{F}_{t}\right|=F_{t}=\mu F_{n} \tag{3.15}
\end{equation*}
$$

The resulting rubbing forces and moment acting at the shaft centreline are:

$$
\begin{equation*}
F_{r y}=F_{n y}+F_{t y} \quad \text { and } \quad F_{r z}=F_{n z}+F_{t z} \quad \text { and } \quad M_{r x}=-r F_{t} \tag{3.16}
\end{equation*}
$$

where

$$
\begin{align*}
& F_{n y}=F_{n} \cos \theta_{r}=F_{n} \frac{z}{e} \quad \text { and } \quad F_{n z}=F_{n} \sin \theta_{r}=F_{n} \frac{y}{e} \\
& F_{t y}=F_{t} \sin \theta_{r}=F_{t} \frac{y}{e} \quad \text { and } \quad F_{t z}=F_{t} \cos \theta_{r}=F_{t} \frac{z}{e} \tag{3.17}
\end{align*}
$$

The NLRGAP entry applies these forces and the moment at the centre of the shaft to a node belonging to the shaft (grid point A). These same forces and the same moment are also applied at the centre of the housing to a node belonging to the housing (grid point B ), but in the opposite
directions ( $F_{r h y}, F_{r h z}$ and $M_{r h x}$ ). Both these nodes are required to be unconstrained (w.r.t. the DOFs effected by the rubbing forces and moment) and coincident in the undeformed shape of the rotor system. Incorporating this radial rubbing model into the 1D rotor system model at node 27 (as grid point A) therefore requires an additional node that represents the housing. Node 104, shown in Figure 3.61, was added as the housing centre node (grid point B) - coincident with node 27. To aid in maintaining numerical stability, it is recommended that spring and/or damper elements with very small stiffness and/or damping coefficients (so as not to influence the NLRGAP forces) are connected between grid points $A$ and $B$. Since these elements may not be directly connected to the rotor, an extra node (node 103) was added (coincident with node 27) and connected to node 27 with an RBE2 rigid element. The spring and damper elements, with stiffness and damping coefficients $1 \times 10^{-10} \mathrm{~N} / \mathrm{mm}$ and $1 \times 10^{-5} \mathrm{~N} . \mathrm{s} / \mathrm{mm}$ in the $y$ - and $z$-directions respectively, were then placed between nodes 103 and 104 (not shown in Figure 3.61 due to space constraints).


Figure 3.61-1D Rotor System Model with Linear Bearings \& Radial Rubbing

Since the contact stiffness is already defined on the NLRGAP entry, the housing node (node 104) should actually be rigidly constrained (fixed) - but since the NLRGAP entry does not allow this, highly-stiff grounded spring elements (with coefficients denoted $k_{r h y}, k_{r h z}$ and $k_{r h \theta}$ ) are connected to this node to provide (almost rigid) stiffness $\left(1 \times 10^{10} \mathrm{~N} / \mathrm{mm}\right)$ for the DOFs that are effected by the rubbing forces and moment while the rest of the DOFs of node 104 are constrained. Grounded damper elements were also connected to node 104, but are not shown in Figure 3.61 (due to space constraints).

The same contact force as shown in Figure 3.51 (from the preceding section) is also used to demonstrate this rubbing model. A radial clearance of 0.2 and a coefficient of friction of 0.3 are used. SOL 129 and SOL 400 nonlinear transient response BDF input files were created (using MSC.Patran) to determine the unbalance response of the rotor system during run-up. The entry additions/modifications from section 3.2.1.5 were incorporated into these files. The version of MSC.Patran used does not seem to support the use of the NLRGAP entry - the entries that were therefore added manually are included in Appendix D.3.10 along with the BDF input files. The same TSTEPNL entry modifications applied to the SOL 400 BDF file from the preceding section (3.2.1.7 and Appendix D.3.9) were applied to the SOL 400 BDF file from this section.

The SOL 400 transient rotor displacement results (including radial clearance information) are shown in Figure 3.62 to Figure 3.65. These results show that the rotating shaft comes into contact with the stationary cylindrical housing as the rotor passes through its first critical speed. Figure 3.63 shows distorted and elliptical orbits caused by the rubbing. Clearance penetration, where the displacement breaches the clearance cylinder, is clearly visible in Figure 3.65. The angular displacement response, mainly due to frictional forces, is shown in Figure 3.64.


Figure 3.62 - Linear 1D Rotor System Radial Rubbing Node 27 Displacements


Figure 3.63 - Linear 1D Rotor System Radial Rubbing Node 27 Displacement Orbit


Figure 3.64 - Linear 1D Rotor System Radial Rubbing Node 27 Angular Displacement

2D Displacement of Node 27 as a Function of Time:


Figure 3.65 - Linear 1D Rotor System Radial Rubbing Node 27 2D Displacement

Figure 3.66 shows a spectrogram of the vertical displacement of node 27 . The rubbing seems to cause some super-synchronous vibrations.


Figure 3.66 - Linear 1D Rotor System Radial Rubbing Node 27 Vertical Displacement Spectrogram

Figure 3.67 and Figure 3.68 show the radial rubbing forces acting on the shaft. Figure 3.69 shows both the displacement and rubbing force magnitudes on the same figure, clearly indicating that the rubbing forces act on the shaft when it breaches the clearance gap. The contact frequency (and duration) decreases with spin speed while the contact force magnitude increases with spin speed until the displacement magnitude starts to subside after passing through the critical speed.


Figure 3.67 - Linear 1D Rotor System Radial Rubbing Forces


Figure 3.68 - Linear 1D Rotor System Radial Rubbing 2D Force


Figure 3.69-Linear 1D Rotor System Radial Rubbing Force \& Node 27 Displacement Magnitudes

### 3.2.1.9 Bearing Anisotropy

Bearing anisotropy may be incorporated by simply changing the translational stiffness of one of the bearing spring elements. The horizontal stiffness of the bush bearing ( $k_{b z}$ ) was changed from $209 \mathrm{~N} / \mathrm{mm}$ to $220 \mathrm{~N} / \mathrm{mm}$ and a direct synchronous frequency response analysis (SOL 108) performed to reveal the effect of this anisotropy. Figure 3.70 shows the resulting steady state cascade orbits of node 48 (journal bearing acting centre) between 30 and 35 Hz . As the spin speed increases, the elliptical orbits change their shape (the major-to-minor axis ratios become larger closer to the critical speed) and orientation (the major axes are more or less vertical before the critical speed and horizontal after the critical speed). As mentioned in section 1.2.1.6, misalignment may cause (amongst other effects) bearing anisotropy, which is easily incorporated into this rotor system model ${ }^{23}$.

2D Displacement Orbits of Node 48 as a Function of Frequency:


Figure 3.70 - Linear 1D Rotor System with Bearing Anisotropy: Journal Cascade Displacement Orbits

### 3.2.2 3D Rotor System Model with Linear Bearings

Figure 3.71 shows a schematic free body diagram of the 3D rotor system model (see Appendix A. 3 for an enlarged version) with its support structure modelled as grounded linear bushing elements and damper elements (CBUSH and CDAMP1 elements respectively) located at the acting centres of the bearings and the coupling.

[^20]

Figure 3.71 - 3D Rotor System Model with Linear Bearings

The parameters/symbols on this schematic diagram are the same as those of the 1D rotor system model with linear bearings (refer to Figure 3.16, p. 74). The linear stiffness and damping coefficients of the bearings and coupling connection of the 3D rotor system model are also the same as those of the 1D model (refer to Table 3.10, p. 75). Important node/grid point numbers are shown in blue on the schematic diagram. Specifically, nodes 239, 2563 and 40000 are the rotor nodes on the spin axis of the rotor at the acting centres of the bush bearing, journal bearing and coupling connection respectively. The grounded bushing and damper elements of the support structure are point (OD) elements, but for the sake of clarity, they are represented as 1D elements with two separate end-points in the schematic diagram.

In order to use MSC.Nastran's Rotor Dynamic capability to perform rotor dynamic analyses on a 3D rotor model, the 3 D rotor must be placed within a superelement with its boundary on the (collinear) spin axis of the rotor (refer to section 1.2.5.2 and subsection three of section 1.2.5.3). The superelement must be defined using SESET Bulk Data entries and not by using Partitioned Bulk Data (PARTs). The ROTORSE Bulk Data entry may then be used to specify the nodes of the rotor line model instead of the ROTORG entry. Use of the ROTORSE entry allows direct connections between the rotor and support structure (i.e. no MPCs are required to connect the rotor to its support structure elements as with the 1D rotor system model - refer to section 3.2.1). The bush bearing and journal bearing elements are therefore directly connected to nodes 239 and 2563 (which belong to the rotor) respectively. Since the acting centre of the coupling does not fall on/within the solid geometry or 'body' of the 3D rotor model, an extra node (node 40000) was created at the coupling connection acting centre and connected to nearby hexahedral elements, forming part of the body of the rotor, via an MPC element (which is discussed in more detail in the next paragraph)
as shown in Figure 3.72 c. The MPC element is shown as red lines with the dependent nodes as red circles. The coupling connection elements are connected to this node (node 40000) as shown in Figure 3.71.

Treating the bearing forces as concentrated/point forces acting on the shaft only via the nodes at the acting centres of the bearings (nodes 239 and 2563 located on the spin axis of the rotor) is unrealistic, since this will cause concentrated stresses (high strain) at these locations. RBE2 (MPC) elements were therefore added to the rotor model to connect all the cross-sectional nodes at the axial locations (planes) of the bearing acting centres to the nodes in the middle of these crosssections, as shown in Figure $3.72 a$ and $b$. These rigidly connected cross-sections are therefore assumed to have negligible deformation (compared to shaft bending) and the bearing forces are (more realistically) transferred to the shaft via the entire cross-section at the acting centre of each bearing. For this same reason, and also as a means to connect the acting centre of the coupling (node 40000) to the body of the rotor, an RBE2 element was also created at the coupling as shown in Figure 3.72 c. Nodes 239, 2563 and 40000 are defined as the independent nodes of these MPCs while the rest of the nodes at each cross-section are dependent upon these nodes - hence the 'spider web'-like structure formed by the MPCs.


Figure 3.72-3D Rotor System Model Bearings \& Coupling Acting Centre MPCs

Before adding the support structure elements in MSC.Patran, the rotor (including the MPCs described above) was placed inside a superelement with all nodes located on the centreline defined as its boundary. The residual structure (superelement 0 , used during the solution procedure) thus consists of these boundary nodes, along with the support elements and nodes added after creation of the rotor superelement (superelement 1 ).

As with the 1D rotor system model (refer to section 3.2.1), an axial ( $x$-direction) displacement constraint is included in the 3D rotor system model at the bush bearing. Specifically, node 225, the node on the rotor spin axis right before node 239 in the positive $x$-direction, is axially constrained. The spin direction of the rotor is also defined as the positive $x$-axis and the imbalances applied to disks 1 and 2 are defined in Table 3.11 (p. 76).

### 3.2.2.1 Normal and Complex Modes

As with the 1D rotor system model, both normal and complex modes analyses were performed on the 3D rotor system (refer to section 3.2.1.1). It seems that the inclusion of internal rotor material structural damping (or any other rotor damping) is not supported in the analysis of 3D rotors placed in superelements with the MSC.Nastran Rotor Dynamics capability. Therefore, unlike the 1D rotor system model, the 3D model was analysed without considering rotor material damping (this applies to all the rotor dynamic analyses performed), considering only support structure damping. The effects of placing the rotor in a superelement were also investigated. Both static (Guyan) reduction and CMS are supported when using the ROTORSE entry (refer to subsection three of section 1.2.5.3). However, CMS can only be used in modal formulations and it seems that, even then, some difficulty is experienced with CMS in complex modes analyses of the 3D rotor system model (specifically during superelement data recovery).

Normal modes analyses (using SOL 103) were performed for three different approaches: a rotor system with no superelements, a rotor superelement statically reduced and a rotor superelement with component mode reduction. The BDF input file for the second approach is included in Appendix D.4.1 (the BDF file for the third approach has only one extra entry in the Case Control Section: PARAM,AUTOQSET,YES). The first 14 undamped natural frequencies from these three approaches are listed in Table 3.16. The frequencies of corresponding modes are colour-matched. A good correlation is noted between the lateral mode frequencies from the first and second approaches, while discrepancies are noted between the torsional mode frequencies from these approaches. The results from the third approach correspond well with the first approach, indicating that component mode reduction is more accurate for a dynamic problem than static reduction. Nevertheless, the static reduction approach appears to be sufficiently accurate for lateral dynamic analyses.

Table 3.16-3D Rotor System Model with Linear Bearings Undamped Natural Frequencies

| No Superelement |  |  | Superelement - Static Red. |  |  | Superelement - CMS |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Mode | $f_{n}[\mathrm{~Hz}]$ | Type | Mode | $f_{n}[\mathrm{~Hz}]$ | Type | Mode | $f_{n}[\mathrm{~Hz}]$ | Type |
| 1 \& 2 | 31.905 | Lateral | 1 \& 2 | 31.905 | Lateral | 1 \& 2 | 31.905 | Lateral |
| 3 | 32.113 | Torsional | 3 | 33.107 | Torsional | 3 | 32.113 | Torsional |
| 4 \& 5 | 131.11 | Lateral | 4 \& 5 | 131.12 | Lateral | 4 \& 5 | 131.11 | Lateral |
| 6 | 235.47 | Torsional | 6 \& 7 | 272.2 | Lateral | 6 | 235.57 | Torsional |
| 7 \& 8 | 272.16 | Lateral | 8 \& | 298.96 | Lateral | 7 \& 8 | 272.16 | Lateral |
| 9 \& 10 | 298.93 | Lateral | 10 | 458.7 | Torsional | 9 \& 10 | 298.94 | Lateral |
| 11 \& 12 | 458.29 | Lateral | 11 \& 12 | 459.44 | Lateral | 11 \& 12 | 458.3 | Lateral |
| 13 \& 14 | 763.31 | Lateral | 13 \& 14 | 767.08 | Lateral | 13 \& 14 | 763.36 | Lateral |

Since difficulty is experienced with CMS during complex modes analyses of the 3D rotor system, complex modes analyses were only performed on the rotor system without any superelements and for the rotor placed in a superelement and statically reduced to a line model (both including support structure damping). SOL 110 (modal formulation) was used and the BDF input file for the second approach is included in Appendix D.4.2. The first 14 resulting damped natural frequencies from these two approaches are listed in Table 3.17 with corresponding modes colour-matched. Once again, the static reduction approach appears to be sufficiently accurate for lateral dynamic analyses - for this reason and the fact that static reduction can also be used in direct formulations,
static reduction has been used from here on forward for all rotor dynamic analyses of the 3D rotor system model.

Table 3.17-3D Rotor System Model with Linear Bearings Damped Natural Frequencies

| No Superelement |  |  | Superelement |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Mode | $f_{d}[\mathrm{~Hz}]$ | Mode Type | Mode | $f_{d}[\mathrm{~Hz}]$ | Mode Type |
| 1 | 0 | Torsional | 1 | 0 | Torsional |
| 2 \& 3 | 31.91 | Lateral | 2 \& 3 | 31.91 | Lateral |
| 4 | 58.108 | Torsional | 4 | 65.617 | Torsional |
| 5 \& 6 | 131.48 | Lateral | 5 \& 6 | 131.49 | Lateral |
| 7 | 245.93 | Torsional | 7 \& 8 | 276.88 | Lateral |
| 8 \& 9 | 276.89 | Lateral | $9 \& 10$ | 287.53 | Lateral |
| 10 \& 11 | 287.47 | Lateral | 11 \& 12 | 459.64 | Lateral |
| 12 \& 13 | 458.47 | Lateral | 13 \& 14 | 773.27 | Lateral |
| 14 \& 15 | 769.64 | Lateral |  |  |  |

As with the 1D rotor system, symmetric lateral modes are present due to the axisymmetric rotor and isotropic bearings. The first complex mode (torsional mode) is once again at 0 Hz indicating an over-damped mode. The first 9 complex mode shapes from the rotor system model with no superelements are included in Appendix F.8.

### 3.2.2.2 Critical Speeds

As with the 1D rotor system model, the critical speeds of the 3D rotor system were determined by performing synchronous complex modes analyses (refer to section 3.2.1.2). The BDF input files for the direct and modal formulations (SOL 107 and SOL 110 respectively), along with descriptions of entries that had to be manually added or modified, are included in Appendix D.4.3.

The first 11 resulting critical speeds, from the direct formulation, are given in Table 3.18. The modal formulation yields almost identical results.

Table 3.18-3D Rotor System Model with Linear Bearings Critical Speeds

| Mode | $\boldsymbol{f}_{\boldsymbol{c}}[\mathbf{H z}]$ | $\boldsymbol{N}_{c}[\mathbf{r p m}]$ | Mode Type |
| :---: | :---: | :---: | :--- |
| $\mathbf{1}$ | $\mathbf{0}$ | $\mathbf{0}$ | Torsional |
| $\mathbf{2}$ | 31.415 | 1884.9 | Backward whirling |
| $\mathbf{3}$ | 32.414 | 1944.84 | Forward whirling |
| $\mathbf{4}$ | 65.617 | 3937.02 | Torsional |
| $\mathbf{5}$ | 126.84 | 7610.4 | Backward whirling |
| $\mathbf{6}$ | 134.46 | 8067.6 | Forward whirling |
| $\mathbf{7}$ | 207.04 | 12422.4 | Backward whirling |
| $\mathbf{8}$ | 239.02 | 14341.2 | Backward whirling |
| $\mathbf{9}$ | 290.81 | 17448.6 | Forward whirling |
| $\mathbf{1 0}$ | 319.97 | $\mathbf{1 9 1 9 8 . 2}$ | Backward whirling |
| $\mathbf{1 1}$ | 342.22 | 20533.2 | Forward whirling |

The BW and FW critical speed pairs are colour-matched in this table. The first three BW and FW critical speed mode shape pairs are shown in Figure 3.74 and Figure 3.76 to Figure 3.78 while the first two torsional mode shapes are shown in Figure 3.73 and Figure 3.75 . For each mode, the light
blue lines indicate the undeformed geometry while the filled shape with black lines indicates the deformed geometry (as in section 3.1.2.1). The grey dotted lines with arrows indicate the whirling paths and directions of single points on the centreline of the rotor. RBE2 elements are shown in red. The frequencies of the torsional modes are, of course, not accurate as discussed in the preceding section. In the first torsional mode, the coupling torsional spring (not visible) is deformed while the rotor rotates rigidly (hence, the uniform fill colour distribution in the axial direction). In the second torsional mode, the rotor deforms (twists) along with the coupling torsional spring with larger deformation towards the journal-end of the rotor (as indicated by the fill colours).


Figure 3.73 - Linear 3D Rotor System $1^{\text {st }}$ Torsional Mode Shape (Mode 1)-0 Hz


Figure 3.74 - Linear 3D Rotor System $1^{\text {st }}$ BW \& FW Lateral Critical Speed Mode Shape Pair


Figure 3.75 - Linear 3D Rotor System $2^{\text {nd }}$ Torsional Mode Shape (Mode 2) - 65.617 Hz


Figure 3.76-Linear 3D Rotor System $2^{\text {nd }}$ BW \& FW Lateral Critical Speed Mode Shape Pair


Figure 3.77 - Linear 3D Rotor System $3^{\text {rd }}$ BW \& FW Lateral Critical Speed Mode Shape Pair


Figure 3.78 - Linear 3D Rotor System $4^{\text {th }}$ BW \& FW Lateral Critical Speed Mode Shape Pair

Table 3.19 shows a comparison between the first three critical speed pairs of the 1D and 3D rotor system models. The first critical speeds seem to correlation well while this correlation seems to fade with increasing mode order. This is also observed when comparing the Campbell diagrams of the 1D and 3D rotor system models (Figure 3.21 and Figure 3.79 respectively) - the gyroscopic effect seems to have a larger influence in the 3D model in comparison to the 1D model.

Table 3.19-Comparison between 1D \& 3D Rotor System Models with Linear Bearings Critical Speeds

| Mode | 1D Model $-\boldsymbol{f}_{\boldsymbol{c}}[\mathbf{H z}]$ | 3D Model $-\boldsymbol{f}_{\boldsymbol{c}}[\mathbf{H z}]$ |
| :--- | :---: | :---: |
| $\mathbf{1}^{\text {st }} \mathbf{B W}$ | 32.218 | 31.415 |
| $\mathbf{1}^{\text {st }} \mathbf{~}$ W | 32.405 | 32.414 |
| $\mathbf{2}^{\text {dd }} \mathbf{~ B W}$ | 123.37 | 126.84 |
| $\mathbf{2}^{\text {nd }}$ FW | 139.97 | 134.46 |
| $\mathbf{3}^{\text {rd }}$ BW | 245.69 | 207.04 |
| $\mathbf{3}^{\text {rd }} \mathbf{~ F W}$ | 294.58 | 342.22 |

### 3.2.2.3 Campbell Diagram

Campbel diagrams of the 3D rotor system are generated by following the same workaround method discussed in section 3.2.1.3 and Appendix D.3.4 with the same CAMPBELL, CAMPBLL and DDVAL entries that need to be manually added to the BDF input file. The BDF file for the 3D rotor system model is included in Appendix D.4.4. The Case Control commands that were added/modified to output only the results at the centre of disk 1 (node 1107) are:

```
SET 1000 = 1107
VECTOR (PRINT, PUNCH , SORT1, REAL)=1000
$SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
```

The resulting Campbell diagram, shown in Figure 3.79, was created using the MATLAB program CampbellDiag_CompPCH.m (described briefly in section 3.2.1.3). This program (and all the rotor analysis results processing programs discussed in section 3.2.1) has additional functionality for models with superelements: the user can choose which superelement's results to be extracted -a
user-modifiable parameter SEIDspec in the code itself specifies the superelement ID ${ }^{24}$. The coloured lines in Figure 3.79 once again indicate the natural whirl and torsional frequencies as functions of rotor spin speed and the dotted grey line is the line on which whirl frequency equals spin speed. The critical speeds identified from this diagram correspond well with the values listed in Table 3.18, although the torsional modes (which are in any case incorrect due to the static reduction) differ. A Campbell diagram from a model where damping is ignored is included in Appendix F.8.


Figure 3.79-3D Rotor System Model with Linear Bearings Campbell Diagram

### 3.2.2.4 Frequency Response

As with the 1D rotor system model, the steady state frequency response of the 3D rotor system was determined by performing a synchronous frequency response analysis (refer to section 3.2.1.4). Only the direct formulation (SOL 108) was considered and the BDF input files, along with entries that had to be manually added or modified, are included in Appendix D.4.5.

The steady state rotor displacement frequency response results at the acting centres of the bearings and the centre of disk 1, for spin speeds $0-200 \mathrm{~Hz}(0-12000 \mathrm{rpm})$, are shown in Figure 3.80 to Figure 3.82. These figures were also created using RotorDispl_FreqRespPCH.m (described briefly in section 3.2.1.4). The resonant peaks at approximately 32.5 Hz ( 1950 rpm )

[^21]and $134.5 \mathrm{~Hz}(8070 \mathrm{rpm})$ correspond well with the values listed in Table 3.18. As with the 1D model results, these results show that the centre of disk 1 whirls with a larger amplitude at the first critical speed than the acting centres of the bearings, while the whirl amplitudes at these three points are more or less of the same order at the second critical speed - this corresponds with the mode shapes of these critical speeds (refer to Figure 3.74 and Figure 3.76).


Figure 3.80 - Linear 3D Rotor System Bush Bearing Displacement Frequency Response


Figure 3.81 - Linear 3D Rotor System Journal Displacement Frequency Response


Figure 3.82 - Linear 3D Rotor System Disk 1 Displacement Frequency Response

Figure 3.83 shows the steady state displacement orbits of node 2563 (journal bearing acting centre) at intermittent spin speeds (cascade orbits) between 25 and 40 Hz while Figure 3.84 shows the orbits between 115 and 155 Hz . Note that, as with the 1D model results (Figure 3.25 and Figure 3.26 ), the position of these dots on the orbits change as the spin speed passes through a critical speed, indicating a phase change.

2D Displacement Orbits of Node 2563 as a Function of Frequency:


Figure 3.83 - Linear 3D Rotor System Journal Cascade Displacement Orbits (25-40 Hz)

2D Displacement Orbits of Node 2563 as a Function of Frequency:


Figure 3.84 - Linear 3D Rotor System Journal Cascade Displacement Orbits (115-155 Hz)

### 3.2.2.5 Transient Response

As with the 1D rotor system model, the unbalance response of the 3D rotor system during run-up from standstill to 10000 rpm within 10 s was also determined by performing a nonlinear transient response analysis using SOL 400 (refer to section 3.2.1.5) and the BDF input file, along with entries that had to be manually added or modified, are included in Appendix D.4.6.

The transient rotor displacement results (vertical and horizontal displacements at the acting centres of the bearings and the centre of disk 1) are shown in Figure 3.85 to Figure 3.87. These figures were also created using RotorDispl_TransPCH.m (described briefly in subsection one of section 3.2.1.5). As with the 1D model, these results show larger whirl amplitudes at the centre of disk 1 than at the acting centres of the bearings at the first critical speed (corresponding with its mode shape), while the whirl amplitudes are of the same order for the second critical speed (also corresponding with its mode shape).


Figure 3.85 - Linear 3D Rotor System Run-Up Bush Bearing Displacements


Figure 3.86 - Linear 3D Rotor System Run-Up Journal Displacements


Figure 3.87 - Linear 3D Rotor System Run-Up Disk 1 Displacements

Figure 3.88 shows the transient displacement of node 2563 (journal bearing acting centre) only between 1 and $3 s$ (1000 and 3000 rpm$)$.

2D Displacement of Node 2563 as a Function of Time:


Figure 3.88 - Linear 3D Rotor System Run-Up Journal 2D Displacement (1-3 s)

MATLAB program RotorDisp/Spectral.m (discussed in subsection one of section 3.2.1.5) was used to create a spectrogram of the vertical displacement of node 239 (bush bearing acting centre), as shown in Figure 3.89. As with the 1D model, the inclined red line indicates synchronous whirling due to unbalance (with the dark red portions indicating resonance) while the horizontal line (in this case at approximately 32.5 Hz or 1950 rpm ) indicates the transient whirling at the first FW critical speed.


Figure 3.89 - Linear 3D Rotor System Run-Up Bush Bearing Vertical Displacement Spectrogram

### 3.2.2.6 Gravity Loading and Initial Conditions

## * Gravity as Inertial Loading

As with the 1D rotor system model, gravity as inertial loading may be applied to the 3D rotor using the MSC.Nastran GRAV Bulk Data entry (refer to section 3.2.1.6). Gravitational acceleration of $9810 \mathrm{~mm} / \mathrm{s}^{2}$ in the negative $y$-direction was also applied to the 3D model. The method used to apply initial conditions to the 1D rotor system model (using TIC entries) is however not compatible with the superelement used in the 3D rotor system model. A different approach thus had to be followed: as mentioned in the second subsection of section 1.2.5.3, SOL 400 offers the ability to use the results from one analysis as the initial conditions for another analysis (via the STEP Case Control command). A static analysis may therefore be performed in the first step, to determine the equilibrium state of the rotor system due to gravity loading, followed by a transient analysis step using the solution from the static analysis step as initial conditions. The version of MSC.Patran used does not seem to support the use of the STEP Case Control command. A SOL 400 BDF input file was therefore created (using MSC.Patran) for a nonlinear transient response analysis, to determine the unbalance response of the rotor system under gravity loading during run-up ( $0-10000 \mathrm{rpm}$ ), and modified manually to apply the results from a static analysis step as initial conditions. This BDF file, along with descriptions of entries that had to be manually added or modified, is included in Appendix D.4.7.

The statically deformed rotor system under gravity loading (ignoring unbalance and gyroscopic effects) is shown in Figure 3.90.


Figure 3.90-3D Rotor System Model with Linear Bearings Static Deformation due to Gravity

The maximum static lateral displacement of the rotor centreline is 0.2696 mm in the negative $y$ direction at $x=238.217 \mathrm{~mm}$ (node 1387). The resulting support deflections and reaction forces (from the bearings acting on the shaft) are given in Table 3.20.

Table 3.20-3D Rotor System Linear Support Deflections \& Reaction Forces due to Gravity

| Support Element | $\boldsymbol{y}[\mathrm{mm}]$ | $F_{y}[\mathrm{~N}]$ |
| :--- | :---: | :---: |
| Bush bearing | -0.067473 | 14.1019 |
| Journal bearing | -0.037326 | 9.33152 |
| Coupling connection | 0.002332 | -3.59141 |

The transient displacement response results for the journal bearing acting centre are shown in Figure 3.91. These are compared to the displacement results from the preceding section (without gravity loading). As with the 1D rotor model, the displacements from the two simulations seem to be almost exactly the same, except for the DC offset between the vertical displacements (equal to the static deflection due to gravity - refer to Table 3.20).


Figure 3.91 - Linear 3D Rotor System with Simplified Gravity: Journal Displacements

## * Simplified Gravity

The simplified gravity loading formulation developed in the second subsection of section 3.2.1.6 is utilised here again. The results calculated with MATLAB program RigidRotorGravEquil.m (included in Appendix C.2) for the 3D rotor system are given in Table 3.21.

Table 3.21-3D Rigid Rotor System Linear Support Deflections \& Reaction Forces due to Simplified Gravity

| Support Element | $y[\mathrm{~mm}]$ | $F_{y}[\mathrm{~N}]$ |
| :--- | :---: | :---: |
| Bush bearing | -0.02665 | 5.5698 |
| Journal bearing | -0.042334 | 10.5835 |
| Coupling connection | -0.023952 | 3.6887 |

The same approach for applying the results of a static analysis step as initial conditions for a transient analysis step that follows was used for the simplified gravity loading, but with a few adjustments. A description of the adjusted approach, including the added/modified entries, is provided in Appendix D.4.8, along with the modified SOL 400 BDF input file that was created (using MSC. Patran) for a nonlinear transient response analysis to determine the unbalance response of the rotor system under the simplified gravity loading during run-up (0-10000 rpm $)$.

The statically deformed rotor system under this simplified gravity loading (ignoring unbalance and gyroscopic effects) is shown in Figure 3.92.


Figure 3.92-3D Rotor System Model with Linear Bearings Static Deformation due to Simplified Gravity

The transient displacement response results for the journal bearing acting centre are shown in Figure 3.93. As with the 1D rotor system model, the DC component of the vertical displacement is equal to the static deformation due to the simplified gravity loading (refer to Table 3.21).


Figure 3.93 - Linear 3D Rotor System with Simplified Gravity: Journal Displacements

### 3.2.2.7 Flat Surface Rubbing

To demonstrate the inclusion of flat surface rubbing in the 3D model, the same flat surface rubbing model described in section 3.2.1.7 is applied halfway between disks 1 and 2 (node 1415 on the centreline of the rotor) of the 3D model. As with the bearing cross-sections, an MPC needs to be added to the cross-section at this node for realistic force transfer (refer to section 3.2.2 and Figure 3.72). A SOL 400 BDF input file was created (using MSC. Patran) for a nonlinear transient response analysis to determine the unbalance response of the rotor system during run-up from standstill to 10000 rpm within 10 s . The entry additions/modifications from section 3.2.2.5 were incorporated into this file. This BDF input file, along with the entries that were added/modified to include the flat surface rubbing model (similar to the 1D model entries), are included in Appendix D.4.9.

The transient rotor displacement results at node 1415 (halfway between disks 1 and 2) are shown in Figure 3.94 to Figure 3.96. These figures were also created using RotorDispl_TransPCH.m and RotorDisp/Clearance.m (the latter is briefly described in section 3.2.1.7). As with the 1D model, the shaft comes into contact with the flat surface as the rotor passes through its first critical speed. Figure 3.96 clearly shows where the displacement breaches (penetrates) the clearance plane.


Figure 3.94 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 Displacements


Figure 3.95 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 Displacement Orbit


Figure 3.96 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 2D Displacement

A spectrogram of the vertical displacement of node 1415 is shown in Figure 3.97. As with the 1D model, harmonic terms and a DC offset are noted where rubbing occurs.


Figure 3.97 - Linear 3D Rotor System Flat Surface Rubbing Node 1415 Vertical Displacement Spectrogram

Figure 3.98 shows both the displacement and rubbing force (acting on the shaft) results on the same figure (created using RotorMultiPlot_TransCSV.m - briefly discussed in subsection two of
section 3.2.1.5). As with the 1D model, this indicates that the rubbing forces act on the shaft when it breaches the clearance once every whirl revolution.


Figure 3.98 - Linear 3D Rotor System Flat Surface Rubbing Forces \& Node 1415 Displacements

### 3.2.2.8 Radial Rubbing

To demonstrate the inclusion of radial rubbing in the 3 D model, the same radial rubbing model described in section 3.2.1.8 is applied at node 1415 (as grid point A) of the 3D model using the NLRGAP entry. Once again, an MPC needs to be added to the cross-section at this node for realistic force transfer. As with the 1D model, an additional node that represents the cylindrical housing needs to be added to the 3D model. Node 41000, shown in Figure 3.99, was added as the housing centre node (grid point B). Bushing elements, damper elements and constraints, similar to that of the 1D model, were added to the model for the cylindrical housing and to aid in maintaining numerical stability, as (partially) demonstrated in Figure 3.99.

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems


Figure 3.99-3D Rotor System Model with Linear Bearings \& Radial Rubbing

A SOL 400 BDF input file was created (using MSC.Patran) for a nonlinear transient response analysis to determine the unbalance response of the rotor system during run-up. The entry additions/modifications from section 3.2.2.5 were incorporated into this file. This BDF input file, along with the entries that were added/modified to include the radial rubbing model (similar to the 1D model entries), are included in Appendix D.4.10.

The transient rotor displacement results at node 1415 (halfway between disks 1 and 2) are shown in Figure 3.100 to Figure 3.102. As with the 1D model, the shaft comes into contact with the cylindrical housing as the rotor passes through its first critical speed. Figure 3.102 clearly shows where the displacement breaches (penetrates) the clearance cylinder. Figure 3.101 shows a significantly different angular displacement compared to Figure 3.64 - it seems as though the rubbing moment (due to friction) is not adequately transferred to the shaft of the 3D model via the MPC unlike the correct transfer of the rubbing moment to the shaft of the 1D model. This may once again be due to the shortcomings of the statically reduced superelement in the 3D model. The effect of this shortcoming is also evident in the lateral displacement results.


Figure 3.100-Linear 3D Rotor System Radial Rubbing Node 1415 Displacements

Angular Displacement of Node 1415:


Figure 3.101 - Linear 3D Rotor System Radial Rubbing Node 1415 Angular Displacement


Figure 3.102 - Linear 3D Rotor System Radial Rubbing Node 1415 2D Displacement

Figure 3.103 shows a spectrogram of the vertical displacement of node 1415 . As with the 1D model, the rubbing seems to cause some super-synchronous vibrations.


Figure 3.103-Linear 3D Rotor System Radial Rubbing Node 1415 Vertical Displacement Spectrogram

Figure 3.104 shows the radial rubbing forces acting on the shaft. Figure 3.105 shows both the displacement and rubbing force magnitudes on the same figure. As with the 1D model, this clearly indicates that the rubbing forces act on the shaft when it breaches the clearance gap and that more or less the same rubbing force behaviour is noted as the rotor passes through the critical speed. Discrepancies are however noted between the 1D and 3D model displacements and rubbing forces possibly due to the shortcomings of the statically reduced superelement in the 3D model, as previously discussed in this section.


Figure 3.104 - Linear 3D Rotor System Radial Rubbing Forces


Figure 3.105 - Linear 3D Rotor System Radial Rubbing Force \& Node 1415 Displacement Magnitudes

### 3.3 Journal Bearing Model

For a rotor system suffering from large unbalance, rubbing, instability due to oil whip, internal damping or any other effect that causes large amplitude vibrations (as may be the case for aging turbomachines), the conventional (linearized) journal bearing models are inadequate in accurately describing the dynamic behaviour of the rotor system. Therefore, a nonlinear model for a general finite-length cylindrical hydrodynamic journal bearing model has been developed (based on the theory discussed in section 1.2.2) to provide accurate nonlinear journal bearing fluid film forces to a rotor system. To solve the Reynolds equation for a finite bearing, the finite difference method is employed (as discussed in section 1.2.2.3). To avoid complexities that are outside the scope of the current work, a few simplifying assumptions (in addition to the inherent assumptions of the Reynolds equation) were made regarding the characteristics of the bearings being modelled. These assumptions are:

- The journal and housing are perfectly cylindrical
- The journal and housing are rigid
- The journal is perfectly aligned with the housing and therefore has negligible inclination in the axial direction

This model may however be easily modified to accommodate different types of bearings (such as wedge-shape, 'lemon'-shape or hydrostatic bearings) and additional fluid effects (such as turbulent flow). Before the finite difference method can be employed to solve the pressure distribution within the fluid film, the boundary value problem (BVP) to be solved needs to be defined.

### 3.3.1 Boundary Value Problem

Multiplying the dynamic Reynolds equation on p.9, eqn. (1.9), with $6 \mu r^{2}$ results in the following:

$$
\begin{equation*}
\frac{\partial}{\partial \theta}\left(h^{3} \frac{\partial p}{\partial \theta}\right)+r^{2} \frac{\partial}{\partial z}\left(h^{3} \frac{\partial p}{\partial z}\right)=6 \mu r^{2} \omega \frac{\partial h}{\partial \theta}+12 \mu r^{2} \frac{\partial h}{\partial t} \tag{3.18}
\end{equation*}
$$

The assumption that the journal has no inclination in the axial direction dictates that $h$ is only a function of $\theta$ and $t$ and that $\partial h / \partial z=0$. Applying this assumption to eqn. (3.18) yields:

$$
\begin{equation*}
\frac{\partial}{\partial \theta}\left(h^{3} \frac{\partial p}{\partial \theta}\right)+r^{2} h^{3} \frac{\partial^{2} p}{\partial z^{2}}=6 \mu r^{2} \omega \frac{\partial h}{\partial \theta}+12 \mu r^{2} \frac{\partial h}{\partial t} \tag{3.19}
\end{equation*}
$$

Applying the product and chain rules to this equation yields:

$$
\begin{equation*}
h^{3} \frac{\partial^{2} p}{\partial \theta^{2}}+3 h^{2} \frac{\partial p}{\partial \theta}+r^{2} h^{3} \frac{\partial^{2} p}{\partial z^{2}}=6 \mu r^{2} \omega \frac{\partial h}{\partial \theta}+12 \mu r^{2} \frac{\partial h}{\partial t} \tag{3.20}
\end{equation*}
$$

Also, since the journal is assumed to have no inclination in the axial direction, the film thickness may be expressed as:

$$
\begin{equation*}
h=c\left(1+\varepsilon \theta^{\prime}\right) \tag{3.21}
\end{equation*}
$$

where $\varepsilon=e / c$ is the eccentricity ratio of the journal bearing. The film thickness may also be expressed in the fixed coordinate system as:

$$
\begin{equation*}
h=c-x \cos \theta-y \sin \theta \tag{3.22}
\end{equation*}
$$

The partial derivative of $h$ with respect to $\theta$ is therefore:

$$
\begin{equation*}
\frac{\partial h}{\partial \theta}=x \sin \theta-y \cos \theta \tag{3.23}
\end{equation*}
$$

and the partial derivative of $h$ with respect to time is:

$$
\begin{equation*}
\frac{\partial h}{\partial t}=-\dot{x} \cos \theta-\dot{y} \sin \theta \tag{3.24}
\end{equation*}
$$

where $\dot{x}=d x / d t$ and $\dot{y}=d y / d t$.
Atmospheric pressures are assumed at the edges of the bearing and the BCs from eqn. (1.13) therefore apply. Since the Reynolds or Swift-Stieber BCs are said to be more realistic than the Gümbel BCs (refer to section 1.2.2.2), a decision was made to apply the Reynolds BCs (which necessitates the use of an iterative solution method). Furthermore, to refrain from falling back to the short bearing approximation to determine the location of the starting boundary of the positivepressure region, this boundary is also treated as a free boundary. Due to the incompatible methods used to apply either Gümbel or Reynolds BCs in an iterative method, as described in section 1.2.2.3, both these boundaries must conform to the same $B C s^{25}$ (i.e. either Reynolds $B C s$ or Gümbel $B C s$ ). The BCs used in this case are, therefore:

$$
\begin{align*}
& p\left(\theta_{1}(z), z\right)=\frac{\partial p}{\partial \theta}\left(\theta_{1}(z), z\right)=0 \\
& p\left(\theta_{2}(z), z\right)=\frac{\partial p}{\partial \theta}\left(\theta_{2}(z), z\right)=0 \tag{3.25}
\end{align*}
$$

To apply these BCs without limiting the range of $\theta$-values within which these boundaries fall, the solution domain should cover the entire surface area of the journal (i.e. $0 \leq \theta \leq 2 \pi$ and $-l / 2 \leq z \leq l / 2)$. Furthermore, to ensure continuity of the pressure distribution around the journal, continuity BCs [eqns. (1.16)] should apply. Now that the BVP is defined, the finite difference method can be employed to solve the pressure distribution in the fluid film.

### 3.3.2 Finite Difference Method

Figure 3.106 shows the grid on the outer cylindrical surface of the journal (flattened in this figure) where the surface is divided into $n$ equal sections along the $\theta$-coordinate (with spacing $\Delta \theta$ ) and $m$ equal sections along the $z$-coordinate (with spacing $\Delta z$ ) with $\theta_{0}=0, \theta_{n}=2 \pi, z_{0}=-l / 2$ and $z_{n}=l / 2$. The coordinates of grid point $(i, j)$ are given by:

$$
\begin{gather*}
\theta_{i}=\theta_{0}+i \Delta \theta=i \Delta \theta \quad \text { for } \quad i=0,1,2, \ldots, n \\
z_{j}=z_{0}+j \Delta z=-\frac{l}{2}+j \Delta z \quad \text { for } \quad j=0,1,2, \ldots, m \tag{3.26}
\end{gather*}
$$

[^22]with
\[

$$
\begin{equation*}
\Delta \theta=\frac{\theta_{n}-\theta_{0}}{n}=\frac{2 \pi}{n} \quad \text { and } \quad \Delta \mathrm{z}=\frac{\mathrm{z}_{\mathrm{n}}-z_{0}}{m}=\frac{l}{m} \tag{3.27}
\end{equation*}
$$

\]



Figure 3.106 - Grid on the Flattened Outer Surface of the Bearing Journal

By replacing the partial derivatives of eqn. (3.20) with the central difference equations given below (refer to Appendix H.2),

$$
\begin{gather*}
\frac{\partial p}{\partial \theta}=\frac{1}{2 \Delta \theta}\left(p_{i+1, j}-p_{i-1, j}\right) \\
\frac{\partial^{2} p}{\partial \theta^{2}}=\frac{1}{\Delta \theta^{2}}\left(p_{i+1, j}-2 p_{i, j}+p_{i-1, j}\right)  \tag{3.28}\\
\frac{\partial^{2} p}{\partial z^{2}}=\frac{1}{z^{2}}\left(p_{i, j+1}-2 p_{i, j}+p_{i, j-1}\right)
\end{gather*}
$$

the following finite difference equation is obtained:

$$
\begin{equation*}
P_{01 i} p_{i-1, j}+P_{11 i} p_{i, j}+P_{21 i} p_{i+1, j}+P_{10 i} p_{i, j-1}+P_{12 i} p_{i, j+1}=f_{i} \tag{3.29}
\end{equation*}
$$

with

$$
\begin{gather*}
P_{01 i}=\frac{h_{i}^{3}}{\Delta \theta^{2}}-\frac{3 h_{i}^{2}}{2 \Delta \theta} \frac{\partial h_{i}}{\partial \theta_{i}} \quad P_{11 i}=-2\left(\frac{h_{i}^{3}}{\Delta \theta^{2}}+\frac{r^{2} h_{i}^{3}}{\Delta z^{2}}\right) \\
P_{21 i}=\frac{h_{i}^{3}}{\Delta \theta^{2}}+\frac{3 h_{i}^{2}}{2 \Delta \theta} \frac{\partial h_{i}}{\partial \theta_{i}} \quad P_{10 i}=P_{12 i}=\frac{r^{2} h_{i}^{3}}{\Delta z^{2}}  \tag{3.30}\\
f_{i}=6 \mu r^{2} \omega \frac{\partial h_{i}}{\partial \theta_{i}}+12 \mu r^{2} \frac{\partial h_{i}}{\partial t}
\end{gather*}
$$

where, from eqns. (3.22) to (3.24) and (3.26),

$$
\begin{gather*}
h_{i}=c-x \cos \theta_{i}-y \sin \theta_{i} \\
\frac{\partial h_{i}}{\partial \theta_{i}}=x \sin \theta_{i}-y \cos \theta_{i}  \tag{3.31}\\
\frac{\partial h_{i}}{\partial t}=-\dot{x} \cos \theta_{i}-\dot{y} \sin \theta_{i}
\end{gather*}
$$

The BCs defined by eqn. (1.13) may be written in finite difference form by utilising eqns. (3.26):

$$
\begin{equation*}
p_{i, 0}=p_{i, m}=0 \tag{3.32}
\end{equation*}
$$

In an iterative solution method, the above BCs are dealt with by setting the initial pressures at $z_{0}$ and $z_{m}$ equal to zero and leaving these pressures unchanged by not calculating new pressures at these locations.

The first part of the pressure continuity BCs [eqns. (1.16)] may be written in finite difference form by utilising eqns. (3.26):

$$
\begin{align*}
p\left(\theta_{i}, z_{j}\right) & =p\left(\theta_{i}+\theta_{n}, z_{j}\right) \\
\therefore p_{i, j} & =p_{i+n, j} \tag{3.33}
\end{align*}
$$

The continuity and periodicity of the grid on the journal surface is demonstrated in Figure 3.107. Specifically, this shows that $\theta_{0}=0$ and $\theta_{n}=2 \pi$ define the same line on the journal surface.


Figure 3.107 - Continuous Grid on the Outer Surface of the Bearing Journal

The BC defined by eqn. (3.33) has the following implications:

- When the pressure is calculated at $\theta_{n}$ (i.e. $p_{n, j}$ is calculated), the pressure at $\theta_{n+1}$ is required (i.e. $p_{n=1, j}$ is required, which falls outside the boundaries defined for this problem). Due to this pressure continuity $\mathrm{BC}: p_{n+1, j}=p_{1, j}$ - the pressures at $\theta_{1}$ can and should therefore be used to calculate $p_{n, j}$.
- The pressures at the boundary limits in the $\theta$-direction should be equal (i.e. $p_{0, n}=p_{n, j}$ ). As per the SOR method, the pressures at $\theta_{0}$ are set equal to zero before the first iteration (as an initial guess) and are not re-calculated during successive iterations, while the pressures at $\theta_{n}$ are calculated at each iteration. To ensure that this pressure continuity BC holds, the pressures at $\theta_{0}$ should therefore be equated to the pressures calculated at $\theta_{n}$ before each successive iteration.

Incorporating these two points into the solution process satisfies the pressure continuity BCs defined by eqns. (1.16).

Finally, the method used to apply the BCs from eqn. (3.25) is discussed in section 1.2.2.2.
In order to solve this finite difference problem using the SOR method, eqn. (3.29) may be written in the form of eqn. (1.17), on p. 13, with coefficients:

$$
\begin{array}{rlr}
A_{1 i}=\frac{P_{01 i}}{2 P_{11 i}} & A_{2 i}=\frac{P_{21 i}}{2 P_{11 i}} \\
A_{3 i}=\frac{P_{10 i}}{2 P_{11 i}} & A_{4 i}=-\frac{f_{i}}{2 P_{11 i}} \tag{3.34}
\end{array}
$$

Eqn. (1.20), on p. 14, may then be used (along with the BCs discussed above) to iteratively solve the pressure value at each grid point.

### 3.3.3 Bearing Forces

The fluid film forces acting on the journal of a fluid film bearing can be calculated by integrating the fluid film pressure distribution over the journal surface area (Gunter et al., 1975):

$$
\left\{\begin{array}{c}
F_{x}  \tag{3.35}\\
F_{y}
\end{array}\right\}=-\int_{0}^{2 \pi} \int_{-l / 2}^{l / 2} p(\theta, z) r\left\{\begin{array}{c}
\cos \theta \\
\sin \theta
\end{array}\right\} d z d \theta
$$

The forces on the housing are equal in magnitude to the journal forces, but in the opposite direction.

Since the finite difference method provides a pressure distribution as discrete pressure values, a numerical integration scheme may be used to evaluate these integrals. The trapezoidal rule is a simple numerical integration scheme and applying it (in both the $\theta$ - and $z$-directions) to integrals (3.35) yields:

$$
\begin{gather*}
F_{x}=-\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{4}\left(p_{i, j} \cos \theta_{i}+p_{i, j-1} \cos \theta_{i}+p_{i-1, j} \cos \theta_{i-1}\right. \\
\left.+p_{i-1, j-1} \cos \theta_{i-1}\right) r \Delta z \Delta \theta \\
F_{y}=-\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{4}\left(p_{i, j} \sin \theta_{i}+p_{i, j-1} \sin \theta_{i}+p_{i-1, j} \sin \theta_{i-1}\right.  \tag{3.36}\\
\left.+p_{i-1, j-1} \sin \theta_{i-1}\right) r \Delta z \Delta \theta
\end{gather*}
$$

which may be simplified to:

$$
\begin{array}{r}
\left\{\begin{array}{l}
F_{x} \\
F_{y}
\end{array}\right\}=-\frac{r \Delta z \Delta \theta}{4} \sum_{i=1}^{n} \sum_{j=1}^{m}\left[\left(p_{i, j}+p_{i, j-1}\right)\left\{\begin{array}{c}
\cos \theta_{i} \\
\sin \theta_{i}
\end{array}\right\}\right.  \tag{3.37}\\
\left.+\left(p_{i-1, j}+p_{i-1, j-1}\right)\left\{\begin{array}{c}
\cos \theta_{i-1} \\
\sin \theta_{i-1}
\end{array}\right\}\right]
\end{array}
$$

### 3.3.4 Journal Bearing Model in MATLAB

A MATLAB function FinHDJB_Reynolds.m (included in Appendix C.3) was written to perform the iterative finite difference method and bearing force calculations as described in sections 3.3.2 and 3.3.3. The journal centre coordinates and velocities, the journal spin speed and radius, the bearing clearance and length, the lubricant dynamic viscosity, the number of grid divisions in the $\theta$ - and $z$ directions, the cavitation pressure and the pressure norm tolerance (for convergence of the SOR method) are received as input for this function. Bearing housing offsets from the coordinate system may be specified in the $x$ - and $y$-directions (if the centre of the housing is not located at the origin of the coordinate system). The function also provides the options to choose whether cavitation should be included or excluded from the solution and whether information on the iteration process (such as the pressure norm at each iteration) should be displayed in the Command Window. As output, the function returns the bearing forces ( $F_{x}$ and $F_{y}$ ) and the fluid film pressure distribution matrix $\left[p_{i, j}\right]$, along with the corresponding discrete $\theta$ - and $z$-vectors ( $\left\{\theta_{i}\right\}$ and $\left\{z_{j}\right\}$ respectively), for the purpose of creating a 3D surface diagram of the pressure distribution. The pressure norm (which is compared to the pressure norm tolerance to establish convergence of the SOR method) is defined as the largest singular value of $p_{i, j}$. Since MATLAB does not support the use of matrix or vector indices smaller than 1 , values of $i+1$ and $j+1$ were used as indices for $\left[p_{i, j}\right],\left\{\theta_{i}\right\}$ and $\left\{z_{j}\right\}$ (for example: $\theta_{0}$ is located in MATLAB vector theta_v at index 1 and $\theta_{n}$ is located at index $n+1$ ).

A MATLAB program FinHDJB_ReynoldsRun.m (included in Appendix C.3) was written to provide input parameters to the FinHDJB_Reynolds function and to plot the results from this function. The program extracts input parameters from an input (*.inp) file which is a (custom) text-based file with input parameters in a specific format (an example INP file, InputParameters.inp, is included in Appendix C.3).

To demonstrate the functionality of the finite journal bearing model and these MATLAB programs, the geometrical parameters and lubricating fluid properties of the experimental rotor system journal bearing (refer to section 2.3.4) are used - a rounded value of 24.5 mm is used for the journal diameter. The input parameters are listed in Table 3.22.

Table 3.22 - Journal Bearing Demonstration Input Parameters

| Parameter | Symbol | Value | Unit |
| :---: | :---: | :---: | :--- |
| $x$ | $x$ | 0.09079 | mm |
| $y$ | $y$ | -0.091336 | mm |
| xdot $\&$ ydot | $\dot{x} \& \dot{y}$ | 0 | $\mathrm{~mm} / \mathrm{s}$ |
| xoff $\&$ yoff | $x_{\text {off }} \& y_{\text {off }}$ | 0 | mm |
| $N$ | $N$ | 4000 | rpm |
| $c$ | $c$ | 0.2 | mm |
| $\boldsymbol{l}$ | $l$ | 11.5 | mm |
| $d$ | $d$ | 24.5 | mm |
| $n u$ | $\mu$ | $4 \times 10^{-8}$ | $\mathrm{MPa.s}$ |
| $n$ | $n$ | 120 | - |
| $m$ | $m$ | 30 | - |
| pcav | $p_{\text {cav }}$ | 0 | MPa |
| tol | - | $1 \times 10^{-6}$ | - |

Furthermore, two cases are considered: where cavitation is ignored ( $c a v=0$ ) and where cavitation is included $(c a v=1)$. Figure 3.108 to Figure 3.112 were created with FinHDJB_ReynoldsRun.m and are discussed below.

Figure 3.108 and Figure 3.109 show the different 3D pressure distributions in the fluid film (as functions of $\theta$ and $z$, on a flat surface) for the cases where cavitation is ignored and included respectively.


Figure 3.108 - Journal Bearing Pressure Distribution (Cavitation Ignored)


Figure 3.109 - Journal Bearing Pressure Distribution (Cavitation Included)

Part $a$ in both Figure 3.110 and Figure 3.111 show the same bearing geometry with the same line of centres, while these figures show the different pressure distributions in the middle of the bearing (at $Z=0$ ) and the directions of the bearing forces for the cases where cavitation is ignored and included respectively. The journal eccentricity and housing diameter are exaggerated (scaled) and the pressure magnitudes are scaled relative to the journal diameter - FinHDJB_ReynoldsRun.m allows the user to choose these scaling values (in the code itself). Part $b$ in both these figures show polar plots of the pressure distributions in the middle of the bearing (scaled relative to the journal diameter) for the cases where cavitation is ignored and cavitation is included respectively.


Figure 3.110 - Journal Bearing Geometry and Pressure Distribution (Cavitation Ignored)


Figure 3.111 - Journal Bearing Geometry and Pressure Distribution (Cavitation Included)

Figure 3.112 shows the pressure distribution, for the case where cavitation is included, with contour lines at intermittent pressure values. The bottom contour line (at $0.5 \%$ of the maximum pressure) clearly shows the curved cavitation boundaries subject to the Reynolds BCs [eqns. (3.25)]. FinHDJB_ReynoldsRun.m allows the user to choose the number of contour levels to be included.


Figure 3.112 - Journal Bearing Pressure Distribution Contours (Cavitation Boundaries)

Figure $3.113 a$ and $b$ show the (scaled, cylindrical) 3D pressure distributions around the journal for the cases where cavitation is ignored and included respectively. A MATLAB program HDJB_CyIPressDist.m (included in Appendix C.3) was written to create these figures.


Figure 3.113-3D Pressure Distribution around Journal

The forces acting on the journal, for the case where cavitation is included, are calculated as $F_{x}=0$ and $F_{y}=10 \mathrm{~N}$. The pressure distribution for this case was solved in 206 iterations of the SOR method.

A MATLAB program HDJB_StatEquil.m (included in Appendix C.3) was also written to use function FinHDJB_Reynolds.m to determine the static equilibrium position/coordinates of the journal for a given bearing load (specified by the user in the code itself). This program utilises the NewtonRaphson numerical method with load stepping to determine the equilibrium position. The load steps are user-modifiable (in the code itself) and are defined as percentages of the complete load. The equations and methodology used are included in Appendix H.3. For a vertical load of 10 N (in the negative $y$-direction), this program calculates the journal equilibrium coordinates as $x=$ 0.09079 mm and $y=-0.091336 \mathrm{~mm}$ (the same as in Table 3.22) and also plots (amongst others) the equilibrium path that the journal follows as the load is increased step-by-step, as shown in Figure 3.114 ( 10 steps were used).


Figure 3.114 - Journal Static Equilibrium Path/Locus as Bearing Load Increases

### 3.3.5 Other Journal Bearing Models

Before arriving at the model described in sections 3.3.1 and 3.3.2, a number of other models (with accompanying MATLAB functions and programs) were developed, where the main discerning features of these different models are the solution techniques and BCs used:

- As a starting point for developing a finite journal bearing model, the first model developed was based on the short bearing approximation (as discussed in section 1.2.2.1). A derivation of the fluid film pressure distribution from the Reynolds equation is included in Appendix H.4. Cavitation is accounted for in this model by equating negative pressures to zero (Gümbel BCs). Two different MATLAB functions ShortHDJB.m and ShortHDJB_Num.m (included in Appendix C.3) were written to calculate and return the fluid film pressure distribution and bearing forces, similar to FinHDJB_Reynolds.m. The former function utilises the built-in MATLAB numerical integration function quadl while the latter function utilises the trapezoidal rule to calculate the bearing forces.
- The next model was developed to apply the finite difference method in solving the pressure distribution for a finite bearing. The finite difference equations (assembled into sparse matrices) are solved via a direct solution method (MATLAB's built-in function mldivide or $\backslash$ is used which performs Gaussian elimination with partial pivoting). The locations of the cavitation boundaries, subject to Gümbel BCs, are estimated from the short bearing approximation.
- The third model also utilises the finite difference method, but the equations are solved iteratively via the SOR method. The cavitation boundary locations are once again estimated from the short bearing approximation.
- The fourth model was developed to remove the reliance on the short bearing approximation to estimate the cavitation boundary locations (subject to Gümbel BCs) by treating both the starting and break-down cavitation boundaries as free boundaries and by applying the continuity BCs, as described in section 3.3.1. Gümbel BCs were, therefore, applied by equating negative pressures to zero after the iterative solution of the pressure distribution.

Figure 3.115 shows a comparison between the pressure distributions (at $z=0$ ) calculated by these different models for the input values of Table 3.22. The maximum pressure from the short bearing approximation differs significantly from the maximum pressures of the finite-length bearing models. The break-down boundary location of the model with Reynolds BCs is noted to be significantly different from the break-down boundary locations of all the other models (which are all effectively subject to Gümbel BCs). This figure was created with the combination of FinHDJB_ReynoldsRun.m, ShortHDJBRun.m (included in Appendix C.3) and similar programs written for the other bearing models.


Figure 3.115 - Comparison of Different Journal Bearing Models

### 3.4 Incorporation of the Journal Bearing Model into the Rotor System Models

The UDS capability of MSC.Nastran (briefly described in section 1.2.7) is utilised to incorporate the journal bearing model, developed in the preceding section, into the 1D and 3D rotor system models - specifically, a user-defined NLRSFD service is utilised. The templates for the FORTRAN user subroutines, provided with the installation of MSC.Nastran, were (partially) employed to this end since this is the simpler approach (compared to generating skeleton files with the genskeleton utility). For the MSC.Nastran 2012 installation used, the template files were located in the directory C:\MSC.Software\MSC_Nastran\20121\ msc20121\nast\services. The templates for a user-defined NLRSFD service were located in the directory $C: \mid M S C$.Software $\mid M S C$ _Nastran $\mid$ 20121\msc20121\nast\services\/mplementations\Elements\src\NIrsfd.

### 3.4.1 Build Environment

Figure 3.116 shows the UDS source tree and all the files included in the source tree.


Figure 3.116 - UDS Build Environment Source Tree

The build configuration files provided with the UDS template files were mostly not used. The SConstruct file was copied from C:\MSC.Software\SDK\20121\Tools\RunTime\lib\python\SCASCons (named SConstruct.example in this directory). This file is included in Appendix E. 3 along with the other build configuration files used. The SConopts file was copied from the templates directory. The SConscript files were copied from the same directory as the SConstruct.example file (named SConscript.example in this directory) into the UDS, src and fortran directories of the source tree and modified by including the code highlighted in blue in Appendix E.3. The SDL, CDL, CPP, H (*.h) and F files were copied from the UDS template files (specifically from the N/rsfd folder) .The modifications made to the CPP and F files are discussed in the next section.

### 3.4.2 Journal Bearing Model in FORTRAN

FORTRAN cannot be used directly with SCA and a 'wrapper' is therefore provided in the template CPP file (N/rsfd.cpp) which allows FORTRAN subroutines to be called through the C++ language. The C++ wrapper calls the pseudo FORTRAN file ext_nlrsfd.F, which may contain pre-processor directives (developed by MSC.Software) that are not supported by FORTRAN. A user-developed FORTRAN subroutine(s) may then, in turn, be called by ext_nlrsfd.F.

FiniteHDJB.f90 (included in Appendix E.1) is the user-defined FORTRAN subroutine created to incorporate the journal bearing model developed for this dissertation (refer to section 3.3) into any MSC.Nastran FE rotor system model. This subroutine is based on the custom MATLAB function FinHDJB_Reynolds.m (refer to section 3.3.4 and Appendix C.3). Table 3.23 provides a mapping between the input and output parameters of ext_nlrsfd.F and FiniteHDJB.f90, as well as a description of each parameter. The names of the NLRSFD Bulk Data entry parameters that correspond with the parameters of ext_nlrsfd.F are indicated in blue in this table.

Table 3.23 - FiniteHDJB.f90-ext_nIrsfd.F Input \& Output Parameter Mapping

| ext_nlrsfd.F | FiniteHDJB.f90 | Description | Input/Output |
| :---: | :---: | :---: | :---: |
| bdia | d | Journal diameter | Input |
| blen | 1 | Length | Input |
| balr | c | clearance | Input |
| visco | nu | Dynamic viscosity | Input |
| pvapco | pcav | Lubricating fluid cavitation pressure | Input |
| offset1 | xoff | Housing centre $x$-direction offset | Input |
| offset2 | yoff | Housing centre $y$-direction offset | Input |
| time | $t$ | Time | Input |
| $x x$ | $x A$ | Journal centre $x$-direction displacement | Input |
| $y y$ | $y A$ | Journal centre $y$-direction displacement | Input |
| $x d t$ | xAdt | Journal centre $x$-direction velocity | Input |
| ydt | yAdt | Journal centre $y$-direction velocity | Input |
| $x b$ | $x B$ | Housing centre $x$-direction displacement | Input |
| $y b$ | $y B$ | Housing centre $y$-direction displacement | Input |
| $x b t$ | xBdt | Housing centre $x$-direction velocity | Input |
| ybt | yBdt | Housing centre $y$-direction velocity | Input |
| $f x$ | FX | Journal fluid film $x$-direction force | Output |
| fy | Fy | Journal fluid film $y$-direction force | Output |
| parm1 | $n r$ | Number of grid divisions in the $\theta$-direction | Input |
| parm2 | $m r$ | Number of grid divisions in the $z$-direction | Input |
| parm3 | tol | Pressure norm tolerance (for convergence of the SOR method) | Input |
| parm4 | eps | Eccentricity Ratio | Output |
| parm5 | kr | Total SOR iterations | Output |
| parm7 | NA | Parameter that indicates whether a new analysis is being started (value=0) or whether an analysis is already in progress (value=1) | Input \& output |

The coordinates in this table refer to the local coordinate system of the journal bearing model where directions $x$ and $y$ correspond with directions 1 and 2 of the NLRSFD entry respectively. Since the journal spin speed is not supplied to FiniteHDJB.f90 through the NLRSFD entry, another subroutine SpinSpeed.f90 (included in Appendix E.1) was written which receives the simulation
time from FiniteHDJB.f90 and returns the spin speed. The user therefore needs to define the spin speed as a function of time ${ }^{26}$ within SpinSpeed.f90. The relative displacements and velocities between the journal and housing are calculated by subtracting the housing displacements and velocities from the journal displacements and velocities respectively. The pressure norm is calculated slightly different from FinHDJB_Reynolds.m - it is defined as the square root of the sum of the squares of each pressure value in matrix $p$ (the Frobenius-norm).

FiniteHDJB.f90 includes an error reporting function that checks whether matrices $p$ and $p \_p r e v$ have been successfully initialized (w.r.t. memory allocation) - in the case of unsuccessful initialization, an error message is written to an error TXT file (of which the location and filename may be modified by the user). An error message is also written to this file if divergence is detected. Furthermore, FiniteHDJB.f90 includes a diagnostic-parameter-output function (via subroutine FiniteHDJBOut.f90 - included in Appendix E.1) which writes (user-specified) parameter values to a TXT file (of which the location and filename may also be modified by the user) each time the subroutine is called by MSC.Nastran. When a new MSC.Nastran analysis is started, the NA parameter receives an initial default value of zero from the NLRSFD entry, indicating a new analysis has been started. This prompts FiniteHDJB.f90 and FiniteHDJBOut.f90 to delete any existing error or diagnostic output TXT files and create new files. After the first call to FiniteHDJB.f90, the value of $N A$ is changed to 1 for the rest of the analysis, which allows FiniteHDJB.f90 and FiniteHDJBOut.f90 to append text to these TXT files every time these subroutines are called.

In addition to the main finite journal bearing model subroutines, similar subroutines for the short bearing approximation, ShortHDJB.f90 and ShortHDJBOut.f90 (included in Appendix E.1), were also written based on the custom MATLAB function ShortHDJB_Num.m (refer to section 3.3.5 and Appendix C.3). The extra code written in ext_nlrsfd.F (included in Appendix E.2) calls either FiniteHDJB.f90 or ShortHDJB.f90 based on the value of parm6 (FiniteHDJB.f90 is called if parm6 = 0 and ShortHDJB.f90 is called if parm6 >0). The finite journal bearing model is therefore used as the default if PARM6 is left blank on the NLRSFD entry.

Extra C++ code was also added to the user implementation sections (for both 32 bit and 64 bit operating systems) of NIrsfd.cpp to print the eccentricity ratio and iteration information (if applicable) for each call to FiniteHDJB.f90 or ShortHDJB.f90 into the MSC.Nastran LOG (.log) output file. This extra code is also included in Appendix E.2.

### 3.4.3 Modifications to the FE Rotor System Models

As with the NLRGAP Bulk Data entry, used to incorporate radial rubbing into the rotor system models (refer to sections 3.2.1.8 and 3.2.2.8 respectively), the NLRSFD Bulk Data entry, used to incorporate the journal bearing model ${ }^{27}$ into the rotor system models, applies forces to both the journal and the bearing housing. This entry thus also requires two coincident nodes, one belonging to the journal (grid point A) and one belonging to the housing (grid point B), with unconstrained DOFs in the directions of the forces being applied. The rotor system models described in section 3.2 (excluding the modified models for rubbing) therefore had to be modified in order to incorporate the journal bearing model.

[^23]
### 3.4.3.1 1D Rotor System Model Modifications

Figure 3.117 shows a schematic free body diagram of the 1D rotor system model, modified to incorporate the journal bearing model. Node 103 was added to the model to represent the bearing housing centre (grid point B), located at the acting centre of the journal bearing (coincident with node 48 , which is grid point $A$, and node 101). Grounded spring and damper elements, with very high stiffness and damping coefficients ( $1 \times 10^{10} \mathrm{~N} / \mathrm{mm}$ and $1 \mathrm{~N} . \mathrm{s} / \mathrm{mm}$ respectively) in the $y$ and $z$-directions, were connected to node 103 to provide effectively rigid support (compared to the rotor stiffness) for these DOFs (since the housing is assumed to be rigid compared to the rotor) without constraining them. The rest of the DOFs of node 103 were constrained. The housing stiffness coefficients $k_{j h y}$ and $k_{j h z}$ are shown in Figure 3.117 (the damper elements are left out for clarity). The original grounded linear journal bearing spring and damper elements (connected to node 101) were replaced with 1D linear spring and damper elements connecting nodes 101 and 103 (not shown in Figure 3.117). As with the radial rubbing model, the purpose of these elements are to aid in maintaining numerical stability and should therefore have very small stiffness and damping coefficients so as not to influence the NLRSFD forces. However, before including the nonlinear journal bearing forces via a UDS, this modified model was analysed with the stiffness and damping coefficients of these elements set equal to the original linear journal bearing stiffness and damping coefficients ( $250 \mathrm{~N} / \mathrm{mm}$ and $0.3 \mathrm{~N} . \mathrm{s} / \mathrm{mm}$ respectively - refer to Table $3.10, \mathrm{p} .75$ ) and then compared to the original model with linear bearings to check that the added bearing housing elements are stiff enough (effectively rigid compared to the rotor) to not alter the results - this was the case.


Figure 3.117-1D Rotor System Model with a Nonlinear Journal Bearing

### 3.4.3.2 3D Rotor System Model Modifications

Figure 3.118 shows a schematic free body diagram of the 3D rotor model, modified to incorporate the journal bearing model. Node 41000 was added at the acting centre of the journal bearing (coincident with node 2563, which is grid point A) to represent the bearing housing centre (grid point B). Similar to the modified 1D rotor system model, grounded bushing elements and damper elements were connected to node 41000 to provide effectively rigid support in the $y$ - and $z$ directions while the rest of the DOFs were constrained (the damper elements are not shown in Figure 3.118). Also similar to the modified 1D model, the original grounded linear journal bearing bushing and damper elements (connected to node 2563) were replaced with 1D bushing and damper elements connecting nodes 2563 and 41000 (not shown in Figure 3.118). This modified 3D model was also verified against the original 3D model with linear bearings.


Figure 3.118-3D Rotor System Model with a Nonlinear Journal Bearing

The RBE2 element at the acting centre cross-section of the original model seems to cause erratic results when the nonlinear journal bearing forces are applied to it through the NLRSFD entry. This element was therefore replaced with an RBE2 element connecting all the outer nodes of the crosssection to the top outer node (node 11212), as shown in Figure 3.119 (the red circles in part $a$ show the dependent nodes all connected to the top independent node while part $b$ shows a virtual 'rigid ring' on which these nodes lie). To provide an effectively rigid connection between the acting centre node (node 2563) and this 'rigid ring', four bushing elements were used, as shown in Figure 3.119 (these elements are shown as green lines in part $a$ ). Stiffness coefficients of magnitude $1 \times 10^{5} \mathrm{~N} / \mathrm{mm}$ were defined for the $y$-and $z$-directions, $1 \times 10^{10} \mathrm{~N} / \mathrm{mm}$ for the $x$-direction and
$1 \times 10^{10} \mathrm{~N} . \mathrm{s} / \mathrm{mm}$ for the $\theta_{\chi}$-direction (rotation about the centreline) of each of these bushing elements. The torsional stiffness is required to "constrain" the axial rotational DOF of node 2563 since no torsional stiffness is provided by the 1 D support elements ${ }^{28}$ or the 3D hexahedral elements (which the rotor consists of) connected to this node.


Figure 3.119-3D Rotor System Model Modified Journal Bearing Force Transfer Structure

[^24]
### 3.5 FE Rotor System Models with Nonlinear Journal Bearings

Before the UDS developed as described in the preceding section (3.4) could be used in MSC.Nastran simulations, the following environment variables (refer to 1.2.7.3) were created (for the MSC.Nastran 2012 installation used):

- SCA_LIBRARY_PATH: $C: \ U D S \backslash A p p s \backslash W I N 8664 \backslash / i b$
- SCA_SERVICE_CATALOG: C:\UDS\Apps\res\SCAServiceCatalog.xml
- SCA_RESOURCE_DIR: C:\UDS\Apps\res

The NLRSFD entry (used to incorporate the journal bearing model into the rotor system models refer to the preceding section: section 3.4) has a parameter PLANE that specifies the orientation of the bearing model in a FE model. In the stationary local coordinate system of the journal bearing, the $z$-axis is the spin axis while in the stationary global coordinate system of the rotor system, the $x$-axis is the spin axis. In order to align these spin axes, the PLANE parameter needs to be defined as $Y Z$. This means that the positive $x$ - and $y$-directions in the bearing local coordinate system are the same as the positive $y$ - and $z$-directions, respectively, in the rotor global coordinate system, as shown in Figure 3.120. Furthermore, the positive journal spin direction is about the positive $x$-axis in the global coordinate system, as shown in this figure.


Figure 3.120-Global \& Local Coordinate Systems of the Rotor System \& Journal Bearing Respectively

### 3.5.1 1D Rotor System Model with a Nonlinear Journal Bearing

### 3.5.1.1 Transient Response without Gravity Loading

The transient response of the 1D rotor system with a nonlinear journal bearing (included via a UDS) can only be solved with a SOL 400 nonlinear transient response analysis. A SOL 400 BDF input file was created (using MSC.Patran) for a nonlinear transient response analysis to determine the unbalance response (with the imbalances given in Table 3.11, p. 76) of the rotor system during runup from standstill to 5000 rpm within 5 s . The entry additions/modifications from section 3.2.1.5 were once again incorporated into this file. The BDF input file, along with a description of the entries that had to be manually added to include the nonlinear journal bearing model via the developed UDS, are included in Appendix D.5.1. Due to the highly nonlinear external journal bearing forces, similar modifications were made to the TSTEPNL entry as discussed in section 3.2.1.7 and Appendix D.3.9. The specific TSTEPNL modifications are also discussed and included in Appendix D.5.1. The journal bearing parameters discussed in section 2.3.4 were used in the analysis with a $121 \times 31$ node mesh/grid.

The stiffness and damping coefficients of the 1D spring and damper elements, connecting nodes 101 and 103, were changed from the arbitrary values used in the linear model ( $250 \mathrm{~N} / \mathrm{mm}$ and $0.3 \mathrm{~N} . \mathrm{s} / \mathrm{mm}$ respectively - refer to Table $3.10, \mathrm{p} .75$ ) to very small values that do not influence the NLRSFD forces (but still aid in maintaining numerical stability) by the following entry modifications:

| PELAS | 8 | $1 .-10$ |
| :--- | :--- | :--- |
| PELAS | 9 | $1 .-10$ |
| PDAMP | 14 | $1 .-5$ |
| PDAMP | 15 | $1 .-5$ |

The transient displacement results at the journal bearing acting centre (node 48) and the centre of disk 1 (node 21) are shown in Figure 3.121 to Figure 3.124 (created with RotorDispl_TransPCH.m, briefly discussed in section 3.2.1.5) while Figure 3.125 shows a spectrogram of the vertical displacement of node 21. This spectrogram was once again created using RotorDisplSpectral.m (also discussed briefly in section 3.2.1.5) and the time-values were replaced by spin speed values by using MATLAB program RotorDisplSpinSpeed.m (included in Appendix C. 1 and discussed later in this section). Black lines at $0.5 \times$ and $1 \times$ shaft spin speed were also added to the spectrogram by this program. The inclined red line, which corresponds with the $1 \times$ spin speed line, indicates synchronous whirling due to unbalance. From approximately $2700 \mathrm{rpm}(2.7 \mathrm{~s})$, a sub-synchronous whirl is noted just below the $0.5 \times$ spin speed line which increases its frequency with increasing spin speed - this indicates the presence of oil whirl. From approximately $3000 \mathrm{rmp}(3 \mathrm{~s})$, this oil whirl transforms into a whirling motion that remains at a constant frequency of approximately 33 Hz (1980 rmp) even though the spin speed increases. Furthermore, the time-domain displacement signals of node 21 show that the whirl amplitude at this frequency continues to increase with increasing spin speed while the displacement signals of node 48 show that the whirl amplitude has increased from $3 s$ onwards to almost the full radial clearance of the journal bearing - this indicates the presence of oil whip at the first FW critical speed of the rotor system. These results therefore confirm that the nonlinear journal bearing model is capable of modelling oil whirl and oil whip.


Figure 3.121-1D Rotor with Nonlinear Journal Bearing: Journal Displacements


Figure 3.122-1D Rotor with Nonlinear Journal Bearing: Journal Displacement Orbit

2D Displacement of Node 48 as a Function of Time:


Figure 3.123-1D Rotor with Nonlinear Journal Bearing: Journal 2D Displacement

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems


Figure 3.124-1D Rotor with Nonlinear Journal Bearing: Disk 1 Displacements


Figure 3.125-1D Rotor with Nonlinear Journal Bearing: Disk 1 Vertical Displacement Spectrogram

MATLAB program RotorDisp/SpinSpeed.m was written to convert the horizontal axes of the figures created by either RotorDispl_TransPCH/CSV.m, RotorDisplSpectral.m or both programs from time to rotor spin speed based on a spin speed function entered by the user (during runtime). This program also provides the option to add lines at any factor of the spin speed to spectrograms.

### 3.5.1.2 Transient Response with Gravity Loading and Initial Conditions

Due to the highly nonlinear nature of a journal bearing, gravity has a significant influence on its stiffness and damping properties. As discussed in subsection two of section 3.2.1.6, use of the GRAV Bulk Data entry to include gravity as an inertial load leads to a substantial increase in computation time required for a transient response simulation - the simplified gravity formulation (from subsection two of section 3.2.1.6) was therefore used here instead.

Before a transient response analysis is performed to determine the unbalance response of the rotor system with this simplified gravity loading during run-up, its state at the beginning of the simulation (which does not correspond with its undeformed state, primarily due to the gravity loading) needs to be determined and applied as initial conditions. The journal bearing cannot support any load at zero spin speed (without contact between the journal and housing) and thus, in order to avoid complexities in modelling journal-to-housing contact for the transient run-up response analysis performed, the analysis was started at 500 rpm and run up to 5000 rpm in 5 s . As an approximation ${ }^{29}$ of the state of the rotor system at the beginning of the simulation, the static equilibrium state of the rotor system under gravity loading at 500 rpm was used as the initial conditions. Since this equilibrium state depends upon the effective stiffness of the journal bearing which, in turn, depends upon the system's equilibrium state (and also the spin speed), an iterative procedure is required to determine the equilibrium state. The simplified gravity loading also depends on the stiffness of the journal bearing and thus also needs to be determined iteratively for a specific spin speed. A spin speed 2500 rpm (an approximately average speed) was used for calculating the simplified gravity loading. Figure $3.126 a$ depicts the basic iterative procedure followed to determine the simplified gravity loading (using a rigid rotor model) while Figure 3.126 b depicts the basic iterative procedure followed to determine the static equilibrium state under this gravity loading. Part $c$ of this figure depicts a mapping between the journal bearing model local coordinate system and the rotor system global coordinate system. Specific descriptions of how these two iterative procedures are implemented are provided below this figure.

[^25]
a) Determine simple gravity loading at average spin speed

b) Determine static equilibrium state of rotor under simple gravity loading at starting spin speed

c) journal bearing local - rotor system global mapping

Figure 3.126 - Basic Iterative Procedures to Determine Equilibrium State of Rotor with Nonlinear Bearing


#### Abstract

Iterative procedure to determine the simplified gravity loading (part a) at $\mathbf{2 5 0 0} \mathbf{r p m}$ : The journal bearing equilibrium coordinates ( $y_{j}$ and $z_{j}$ in the global coordinate system of the rotor system) are calculated from initial values of the vertical and horizontal journal bearing loads ( $F_{j y}$ and $F_{j z}$ respectively) using MATLAB program HDJB_StatEquil.m (briefly discussed in section 3.3.4). A new $F_{j y}$ value is calculated using RigidRotorGravEquil.m (with $k_{j y} \cong\left|F_{j y} / y_{j}\right|$ ) and a new $F_{j z}$ value is calculated (as a reaction force in a static analysis) by applying $z_{j}$ as an enforced displacement in the rotor system FE model (to node 48). These steps are repeated until $F_{j y}$ and $F_{j z}$ converge. The initial value of $F_{j y}$ was calculated using RigidRotorGravEquil.m with an assumed value of $250 \mathrm{~N} / \mathrm{mm}$ for $k_{j y}$ while an initial value of zero was used for $F_{j z}$. Iterative procedure to determine the equilibrium state under simplified gravity loading (part b) at 500 rpm: Coordinates $y_{j}$ and $z_{j}$ are calculated from initial values of $F_{j y}$ and $F_{j z}$ (the simplified gravity loading values were used as initial values). New $F_{j y}$ and $F_{j z}$ values are calculated (as reaction forces) by applying $y_{j}$ and $z_{j}$ as enforced displacements in the rotor system FE model (to node 48). These steps are repeated until $y_{j}$ and $z_{j}$ converge.


The resulting simplified gravity loading and corresponding bearing displacements (at 2500 rpm) from implementing these iterative procedures are given in Table 3.24.

Table 3.24-1D Rotor System with a Nonlinear Journal Bearing Simplified Gravity Loading @ 2500 rpm

| Support Element | $y[\mathrm{~mm}]$ | $F_{y}[\mathrm{~N}]$ |
| :--- | :---: | :---: |
| Bush bearing | -0.032072 | 6.7031 |
| Journal bearing | -0.11528 | 10.4044 |
| Coupling connection | -0.017763 | 2.7355 |

The equilibrium coordinates calculated for the journal bearing under this gravity loading at 500 rpm is $y_{j}=-0.1654 \mathrm{~mm}$ and $z_{j}=-0.07009 \mathrm{~mm}$. To verify the accuracy of the simplified gravity loading, the iterative procedure depicted in Figure $3.126 b$ was performed for the rotor system with full gravity loading applied as an inertial load (using the GRAV entry) to determine the journal bearing equilibrium coordinates at 500 rpm . This yielded the coordinates $y_{j}=$ -0.1629 mm and $z_{j}=-0.07178 \mathrm{~mm}$ which are in good agreement with the equilibrium coordinates calculated for the simplified gravity loading.

The method for applying initial conditions to the 1D model using MSC.Patran described in subsection one of section 3.2.1.6 was utilised here again, with a few changes:

1. Apply the journal bearing equilibrium coordinates as enforced displacements to nodes 48 and 103
2. Perform a linear static analysis on the rotor system with the simplified gravity loading (using SOL 101) to determine its equilibrium deformation
3. Delete the enforced displacements applied in step 1
4. Attach the results from this analysis to MSC.Patran (including rotational nodal results)
5. Create a Vector plot (using the Marker tool of MSC.Patran) of the translational DOF displacement results
6. Create a Spatial Field with the following properties:

FEM Field Definition: Continuous
Field Type: Vector
Options/Extrapolation Option: Linear Extrapolation
7. Repeat steps 3 and 4 for the rotational DOF displacement results
8. Delete the results attachment from the MSC.Patran database
9. Create Initial Displacements using the Spatial FEM Fields created in step 6 and apply it to all the nodes of the rotor (excluding support nodes and dependent MPC nodes).

The BDF input file for the SOL 400 nonlinear transient response analysis is included in Appendix D.5.2. The same entry additions/modifications from the preceding section (3.5.1.1) were applied to this analysis.

Figure 3.127 to Figure 3.131 show the displacement results at the acting centres of the bush and journal bearings (nodes 5 and 48 respectively). Figure 3.129 and Figure 3.130 also include radial clearance lines representing the journal bearing clearance - the high eccentricity of the journal bearing (eccentricity ratio $\varepsilon$ close to 1 ) indicates that it is relatively heavily loaded. The journal bearing vibration amplitude peaks at approximately $1.56 s$ ( 1900 rpm ), indicating a critical speed (the first FW critical speed). Figure 3.130 and Figure 3.131 show non-elliptical asymmetric orbits, especially near the critical speed - this phenomenon was also observed in the experimental results discussed in section 4.3.3 and may be a symptom of heavy loading, i.e. a nonlinear effect caused by the high eccentricity ratio and bearing load (Bently \& Hatch, 2002). It seems as though the journal reaches a position, at the top of its whirl path, where the fluid film forces no longer counter-act gravity; gravity then pulls the journal downward (its motion also influenced by the imbalance inertia/unbalance forces of the rotor) until the fluid film forces become significant once more further down the whirl path. At approximately 1.7 s (2030 rpm), the shape and orientation of the whirl orbit changes, indicating a phase change as the critical speed is traversed. The DC
components of the journal bearing displacement signals change as the spin speed increases - as expected, the journal centre moves closer to the housing centre as the spin speed increases.


Figure 3.127-1D Rotor with Nonlinear Journal Bearing \& Gravity: Bush Bearing Displacements


Figure 3.128-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements


Figure 3.129-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements \& Clearance


Figure 3.130-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacement Orbit

2D Displacement of Node 48 as a Function of Time:


Figure 3.131-1D Rotor with Nonlinear Journal Bearing \& Gravity: Journal 2D Displacement

Figure 3.132 and Figure 3.133 show spectrograms of the vertical and horizontal displacements of the journal bearing acting centre. As usual, the inclined red lines at frequencies equal to the spin speed of the rotor indicate synchronous whirling due to unbalance. Since the highly nonlinear journal bearing causes non-elliptical asymmetric orbits (especially near the critical speed), which may be seen as distortions of a normal harmonic response, these distortions cause additional integer harmonic lines to appear on the spectrograms. The horizontal line at approximately 32 Hz (1920 rpm) indicates transient whirling from the excitation of the first FW critical speed.


Figure 3.132-1D Rotor with Journal Bearing \& Gravity: Journal Vertical Displacement Spectrogram


Figure 3.133-1D Rotor with Journal Bearing \& Gravity: Journal Horizontal Displacement Spectrogram

### 3.5.2 3D Rotor System Model with a Nonlinear Journal Bearing

### 3.5.2.1 Transient Response without Gravity Loading

As with the 1D rotor system model, a BDF input file was created (using MSC.Patran) for a nonlinear transient response analysis to determine the unbalance response of the 3D rotor system during run-up $(0-5000 \mathrm{rpm})$. The entry additions/modifications from section 3.2.2.5 were once again incorporated into this file. This BDF input file, along with other entries that were manually added or modified to include the journal bearing model (similar to the 1D model), are included in Appendix D.6.1.

The transient displacement results at the journal bearing acting centre (node 2563) and the centre of disk 1 (node 1107) are shown in Figure 3.134 to Figure 3.136 while Figure 3.137 shows a spectrogram of the vertical displacement of node 1107 with black lines at $0.5 \times$ and $1 \times$ shaft spin speed. As with the 1D model, these results indicate the presence of oil whirl and oil whip, although the onset of these phenomena occur slightly earlier during the run-up (from approximately 2.6 s and 2.9 s respectively). The oil whip frequency is however still indicated as approximately 33 Hz (1080 rpm).

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems


Figure 3.134-3D Rotor with Nonlinear Journal Bearing: Journal Displacements


Figure 3.135-3D Rotor with Nonlinear Journal Bearing: Journal 2D Displacement


Figure 3.136-3D Rotor with Nonlinear Journal Bearing: Disk 1 Displacements


Figure 3.137-3D Rotor with Nonlinear Journal Bearing: Disk 1 Displacement Spectrogram

### 3.5.2.2 Transient Response with Gravity Loading and Initial Conditions

As with the 1D model, the static equilibrium state of the 3D rotor system model at 500 rpm under simplified gravity loading (calculated at 2500 rpm ) was used as the initial conditions for the nonlinear transient response analysis performed to determine the unbalance response of the 3D rotor system during run-up from 500 rpm to 5000 rpm within 5 s . This equilibrium state was determined by following the same iterative procedures discussed for the 1D model (refer to section
3.5.1.2 and Figure 3.126). The resulting simplified gravity loading and corresponding bearing acting centre displacements are given in Table 3.25.

Table 3.25-3D Rotor System with a Nonlinear Journal Bearing Simplified Gravity Loading @ 2500 rpm

| Support Element | $y[\mathrm{~mm}]$ | $F_{y}[\mathrm{~N}]$ |
| :--- | :---: | :---: |
| Bush bearing | -0.032032 | 6.6948 |
| Journal bearing | -0.11526 | 10.4184 |
| Coupling connection | -0.017719 | 2.7288 |

The equilibrium coordinates calculated for the journal bearing under this gravity loading at 500 rpm is $y_{j}=-0.16541 \mathrm{~mm}$ and $z_{j}=-0.070173 \mathrm{~mm}$. As with the 1D model, the accuracy of the simplified gravity loading applied to the 3D model was verified by performing the iterative procedure depicted in Figure $3.126 b$ for the rotor system with full gravity loading applied as an inertial load (using the GRAV entry) to determine the journal bearing equilibrium coordinates at 500 rpm . This yielded the coordinates $y_{j}=-0.16281 \mathrm{~mm}$ and $z_{j}=-0.071939 \mathrm{~mm}$ which are in good agreement with the equilibrium coordinates calculated for the simplified gravity loading.

In order to apply the equilibrium state as initial conditions, a similar approach as described in the second subsection of section 3.2.2.6 and Appendix D.4.8 was followed. However, this approach had to be slightly adapted in this case since the nonlinear journal bearing model (included via a UDS) cannot be used in a static analysis (the version of MSC.Nastran used only allows the use of this type of nonlinear forces in frequency response and transient response analyses) and different MPCs cannot be applied in different analysis steps. In the adapted approach, the journal bearing equilibrium forces (in this case: $F_{j y}=10.3543 N$ and $F_{j z}=-0.08967 \mathrm{~N}$ ) are applied to the journal bearing acting centre using FORCE Bulk Data entries during the static analysis step (where the rotor system equilibrium state due to gravity is determined), after which these static forces are replaced by nonlinear journal bearing forces (from the NLRSFD entry) during the transient analysis step (where the results from the static step are used as initial conditions). The SOL 400 nonlinear transient response analysis BDF input file, along with the entry additions/modifications from the adapted approach, are included in Appendix D.6.2.

Figure 3.138 to Figure 3.142 show the displacement results at the acting centres of the bush and journal bearings (nodes 239 and 2563 respectively). Figure 3.140 and Figure 3.141 also include radial clearance lines representing the journal bearing clearance. As with the 1D rotor system model, the journal bearing vibration amplitude peaks at approximately 1.56 s ( 1900 rpm ), indicating a critical speed (the first FW critical speed). The orbits are also non-elliptical and asymmetric near this critical speed. A phase change is also noted at approximately 1.7 s ( 2030 rpm ) and the average position of the journal centre moves closer towards the centre of the housing due to stiffening of the bearing with an increase in spin speed.


Figure 3.138-3D Rotor with Nonlinear Journal Bearing \& Gravity: Bush Bearing Displacements


Figure 3.139-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements


Figure 3.140-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacements \& Clearance


Figure 3.141-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal Displacement Orbit

2D Displacement of Node 2563 as a Function of Time:


Figure 3.142-3D Rotor with Nonlinear Journal Bearing \& Gravity: Journal 2D Displacement

Figure 3.143 and Figure 3.144 show spectrograms of the vertical and horizontal displacements of the journal bearing centre. Harmonic lines are present near the critical speed due to the nonelliptical asymmetric orbits from the nonlinear journal bearing and the horizontal line at approximately 32 Hz ( 1920 rpm ) indicates whirling from the excitation of the first FW critical speed.


Figure 3.143-3D Rotor with Journal Bearing \& Gravity: Journal Vertical Displacement Spectrogram


Figure 3.144-3D Rotor with Journal Bearing \& Gravity: Journal Horizontal Displacement Spectrogram

## 4 Rotor System Experimental Analysis

In this chapter, the experimental analyses performed on the complete experimental rotor system are discussed. Specifically, two analyses (with slightly different setups) were performed: an unbalance response analysis (during run-up) and a rotor-stator rubbing analysis (also including imbalances). As discussed in section 1.1, the signals from proximity probes include undesirable runout components that "interfere" with the desired vibration components. Runout thus needs to be compensated for (usually done by subtracting it from the total measured signal) in order to eliminate or reduce its effect on the vibration signals. For both the analyses performed on the rotor system, shaft runout was measured beforehand and used for runout compensation on the vibration signals. Before any measurements were recorded, the rotor was balanced to effectively eliminate (most of) its inherent imbalances.

### 4.1 Rotor Balancing

Two-plane balancing was performed on the rotor to counteract its inherent imbalances (mainly contributed to uneven fastening of the disks onto the shaft and a slightly bent shaft). The centres of the two disks were chosen as the two balancing planes, as show in Figure 4.1.


Figure 4.1 - Two-Plane Balancing: Balancing Planes, Bearings \& Phase Reference

An SKF Microlog CMVA10, an accelerometer and optical tachometer, shown in Figure 4.2, were used to perform the balancing - the accelerometer measured vibration amplitude while the tachometer measured vibration phase. The $0^{\circ}$ mark on disk 1 was used as the reference/phase mark from which all angles were measured in the direction opposite to the spin direction. Two strips of reflective tape were adhered to the shaft near the two bearings (labelled bearings A and B as shown in Figure 4.1) in line with the $0^{\circ}$ reference mark. When the light from the tachometer passes over one of these reflective strips, it causes a pulse in the tachometer signal. The following two-plane balancing procedure was followed...

- Reference run: Measure and record the vibration magnitude and phase (using the Microlog) at bearings $A$ and $B$ due to the inherent/initial unbalance.
- Trial run 1: Add a known trial weight to disk 1 (plane 1) and record its size and position. Measure and record the vibration magnitude and phase at bearings $A$ and $B$. Remove the trial weight.
- Trial run 2: Add a known trial weight to disk 2 (plane 2) and record its size and position. Measure and record the vibration magnitude and phase at bearings $A$ and $B$. Remove the trial weight.
- Calculate the correction weights using the Microlog and mount it onto the rotor.


Figure 4.2 - Two-Plane Balancing: Instrumentation

Balancing was performed at approximately 30 Hz ( 1800 rpm - near the first critical speed of the rotor). Table 4.1 lists the trial weights used, the balancing measurement results, as well as the calculated correction weights.

Table 4.1 - Two-Plane Balancing Measurements \& Results

| Trial Weights |  |  | Trial Run 1 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $r=30.75 \mathrm{~mm}$ | Plane 1 | Plane 2 | Vibration | Bearing A | Bearing B |
| $m[g]$ | 2.4 | 2.4 | Amplitude | 1.8782 | 1.6799 |
| $\boldsymbol{\theta}$ [deg] | 180 | 180 | Phase [deg] | 254.7 | 256.1 |
| Reference Run |  |  | Trial Run 2 |  |  |
| Vibration | Bearing A | Bearing B | Vibration | Bearing A | Bearing B |
| Amplitude | 2.3531 | 2.3109 | Amplitude | 1.9672 | 1.8530 |
| Phase [deg] | 322.7 | 327.3 | Phase [deg] | 259.2 | 265.5 |
| Calculated Counter Weights |  |  |  |  |  |
| $r=30.75 \mathrm{~mm}$ | Plane 1 | Plane 2 |  |  |  |
| $m[g]$ | 1.3914 | 2.7544 |  |  |  |
| $\theta$ [deg] | 218.88 | 98.24 |  |  |  |

After this procedure, only one correction weight was applied at disk 2: a 2.4 g weight at an angle of $112.5^{\circ}$ which seemed to eliminate most of the vibration caused by unbalance, although a small residual unbalance still remained. Attempts were made to determine this residual unbalance by following the same balancing procedure described above, but a consistent result could not be obtained. This may be attributed to the fact that the balancing procedure is intended for rigid rotors with linear bearings, which is inaccurate in this case where the rotor is flexible and a nonlinear journal bearing is present.

### 4.2 Runout Compensation

Two different runout compensation methods were developed: the first method is applicable where the DC component of the vibration signal is not important (which is usually the case if the DC component remains constant at all spin speeds - for example: near a bearing where the stiffness of the bearing is not influenced by the spin speed) and the second method is applicable where the position of the shaft (or journal) relative to a bearing housing is important (for example: in the case of a hydrodynamic journal bearing where the DC component of the signal measured near this bearing changes as the spin speed changes).

### 4.2.1 Method 1: DC Component Unimportant

Figure $4.3 a$ shows a shaft in its equilibrium position during slow-roll (when runout is measured) while part $b$ shows the shaft displaced from this equilibrium position due to vibration during operation. Exaggerated (fictitious) runout is depicted by the blue line superimposed on the ideal (perfectly round) shaft.


Figure 4.3 - Runout \& Vibration Measurement when DC Component Unimportant

The ideal shaft radius is denoted $r$ while the measured gaps between the shaft and the $y$ - and $z$ probes (including runout) are denoted $g_{R O y}$ and $g_{R O Z}$ respectively during runout measurement (part a) and $g_{y}$ and $g_{z}$ respectively during vibration measurement (part $b$ ). The shaft radii, including runout, in the $y$ and $z$-directions are denoted $r_{R O y}$ and $r_{R O z}$ respectively. The deviations from the ideal radius due to runout in the $y$ - and $z$-directions are denoted $v_{R O}$ and $w_{R O}$ respectively while the centreline displacements from the equilibrium position during vibration measurement are denoted $v$ and $w$ in the $y$ - and $z$-directions respectively.

From Figure 4.3:

$$
\begin{align*}
r+v_{R O}+g_{R O y} & =v+r+v_{R O}+g_{y} \\
\therefore v & =g_{R O y}-g_{y} \tag{4.1}
\end{align*}
$$

Similarly:

$$
\begin{equation*}
w=g_{R O_{z}}-g_{z} \tag{4.2}
\end{equation*}
$$

### 4.2.2 Method 2: DC Component Important

Figure 4.4 a shows a journal in its bearing housing during slow-roll (when runout is measured) while part $b$ shows this same setup during operation when the journal position, relative to the housing, is measured.


Figure 4.4 - Runout \& Vibration Measurement when DC Component Important

Additional parameters from this figure, not defined in the preceding section, are: $R$, which denotes the bearing housing inner radius; $g_{h y}$ and $g_{h z}$, which denote the gaps between the housing and the $y$-and $z$-probes respectively; $y_{R O}$ and $z_{R O}$, which denote the coordinates of the journal centre relative to the housing centre during slow-roll (runout measurement); and $y$ and $z$, which denote the journal centre coordinates during operation (vibration measurement).

An assumption is made that the average gap (probe-to-shaft gap) measured during slow-roll (including runout), by either the $y$ - or $z$-probe over one shaft revolution, is (approximately) equal to the gap between the probe and the ideal shaft (denoted $g_{y}$ or $g_{z}$ ), i.e.:

$$
\begin{equation*}
g \cong g_{R o_{a v e}}=\frac{1}{2 \pi} \int_{0}^{2 \pi} g_{R O} d \theta \quad \Leftrightarrow \quad r \cong r_{R o_{a v e}}=\frac{1}{2 \pi} \int_{0}^{2 \pi} r_{R O} d \theta \tag{4.3}
\end{equation*}
$$

where angle $\theta$ is measured from the tachometer pulse (assuming a single pulse-per-rotation) in the spin direction. From this assumption, the deviations from the ideal shaft due to runout may be determined:

$$
\begin{equation*}
v_{R O}=g_{R O y_{\text {ave }}}-g_{R O y} \quad \text { and } \quad w_{R O}=g_{R O z_{\text {ave }}}-g_{R O z} \tag{4.4}
\end{equation*}
$$

and thus:

$$
\begin{equation*}
r_{R O y}=r+v_{R O} \quad \text { and } \quad r_{R O z}=r+w_{R O} \tag{4.5}
\end{equation*}
$$

From Figure 4.4:

$$
\begin{aligned}
g_{h y}+R & =y+r+v_{R O}+g_{y} \\
\therefore y & =R-r-v_{R O}+g_{h y}-g_{y}
\end{aligned}
$$

but, $R-r=D / 2-r=c[$ refer to eqn. (1.12)],

$$
\begin{equation*}
\therefore y=c-v_{R O}+g_{h y}-g_{y} \tag{4.6}
\end{equation*}
$$

Similarly:

$$
\begin{equation*}
z=c-w_{R O}+g_{h z}-g_{z} \tag{4.7}
\end{equation*}
$$

### 4.3 Transient Unbalance Response Analysis

This experiment was conducted to determine the transient dynamic response of the rotor system (specifically at its bearings) due to unbalance forces as the main excitation source.

### 4.3.1 Experimental Setup

Figure 4.5 is a schematic of the experimental setup used to measure the unbalance response of the rotor system while Figure 4.6 and Figure 4.7 show photos of this setup.


Figure 4.5 - Rotor System Transient Unbalance Response Experimental Setup Schematic

A pair of eddy current proximity probes were installed in the journal bearing housing in the $y$ - and $z$-directions - see Figure 4.7 a. Proximity probes could not be directly installed in the bush bearing pedestal since the horizontal probe hole does not line up with bearing housing centre which causes cross-interference between the magnetic fields of the $y$ - and $z$-probes. A sensor pedestal with a pair of proximity probes was therefore mounted right next to the bush bearing pedestal - see Figure 4.7 d . These four proximity probes were connected to a transducer unit (which is part of the rotor kit with four proximity probe drivers) powered by a $24 V$ DC power supply. A thermocouple was installed to measure the lubricant temperature in the journal bearing - see Figure 4.7 e. A proximity probe tachometer (which is part of the Bently Nevada rotor kit) is located next to the coupling to measure the motor spin speed - see Figure 4.7 c. This probe sends its signals (as feedback) to the motor controller (which controls the maximum speed, slow-roll speed and runup/down rate of the motor). The same optical tachometer used during balancing (refer to section
4.1) was installed at the journal bearing. The signals from all these sensors were digitised and recorded using an HBM SOMAT eDAQ lite data acquisition device and then uploaded to a computer (in *.sie file format) using TCE 3.16 .0 software and converted to text file format (tab delimited) using Infield 2.50 software. Figure $4.7 b$ shows the load-carrying fluid film and cavitation regions within the journal bearing during operation (a blue dye was added to the lubricating fluid to allow for visual inspection of the fluid film).


Figure 4.6 - Rotor System Transient Unbalance Response Experimental Setup: Overall

In order to perform runout compensation according to the second method (refer to section 4.2.2) on the vibration signals measured at the journal bearing, the gaps between the housing and the $y$ and $z$-probes ( $g_{h y}$ and $g_{h z}$ respectively) had to be determined. While recording journal displacements with these probes, the journal was manually moved (by hand) until it made contact with the bearing housing at the top, bottom, left-hand and right-hand sides of the bearing (this was repeated a number of times) as demonstrated in Figure 4.8. The average minimum gap values measured by the $y$ - and $z$-probes at the top and left positions $(0.27 \mathrm{~mm}$ and 0.42 mm respectively) are regarded to be $g_{h y}$ and $g_{h z}$ respectively. The bearing diametrical clearances in the vertical and horizontal directions were also calculated from these gap values to be approximately 0.4 mm in both directions, which confirms a radial clearance of 0.2 mm (also measured as described in section 2.3.4.1).


Figure 4.7 - Rotor System Transient Unbalance Response Experimental Setup: Sensors \& Fluid Film


Figure 4.8 - Journal Bearing Housing-Proximity Probes Gaps \& Bearing Clearance Measurement

### 4.3.2 Runout Measurement

Runout at the bush and journal bearings were measured at a slow-roll spin speed of approximately 105 rpm over a duration of 20 s . A MATLAB program Runout.m (included in Appendix B.4) was written to load and plot the raw recorded signals from a TXT file. Before running this program, the user is required to specify the types of channels included in the TXT file by specifying names/labels for these channels under the correct channel type in the code itself - only runout and tachometer signals are supported and both are required. Runout signals may be specified in pairs ( $y$ - and $z-$ probe pairs) or as single channels. During runtime, the program requests the user to enter the filename of the TXT file as well as the time range for which to extract the signals. This program assumes that the lateral runout measurements are in the $y z$-plane (where the $y$-axis is the vertical axis and the $z$-axis is the horizontal axis). Furthermore, the program calculates and displays the sampling frequencies of the signals and applies the time limits specified by the user. The measured runout gap signals are also converted from $V$ to $m m$ using the sensitivity of the eddy current proximity probes, which is $-8 \mathrm{~V} / \mathrm{mm}$ (the calibration certificate of one of the proximity probes used is included in Appendix G.2).

Figure 4.9 and Figure 4.10 show the $y$-probe runout gap signals (only between 10 and $15 s$ ) at the bush and journal bearings respectively (from the first recorded measurement) along with the corresponding non-driving end tachometer signal, while Figure 4.11 shows the driving end tachometer signal (between 10 and 12 s ). The runout signals are periodic where each period corresponds with a single rotation of the shaft - as expected.


Figure 4.9 - Rotor System Bush Bearing y-Probe Runout Gap Signal


Figure 4.10 - Rotor System Journal Bearing y-Probe Runout Gap Signal


Figure 4.11 - Rotor System Slow-Roll Driving End Tachometer Signal

Another MATLAB program RunoutSP.m (also included in Appendix B.4) was written to process these runout gap signals and plot the processed results. When executed, the slow-roll speed is determined as a function of time. A MATLAB function TachoSpinSpeed.m (included in Appendix B.5), called within RunoutSP.m, was written to do this. This function (based on function getrpm.m from the VibraTools MATLAB toolbox suite) finds the tachometer pulse locations (time-values in a tachometer signal) by determining where the tachometer signal magnitude passes a user-defined trigger level (in Volts) with a positive slope. The spin speed (in rad/s) for each segment in-between consecutive tachometer pulses is determined by dividing the shaft angle between each pulse $\left(360^{\circ}\right.$ or $2 \pi \mathrm{rad}$ divided by the number of pulses-per-revolution) by the duration of the segment ${ }^{30}$.

To determine an average runout signal per shaft revolution for each probe, synchronous averaging is performed. A MATLAB function SyncAve.m (included in Appendix B.5), called within RunoutSP.m, was written to perform the synchronous averaging by breaking the run-up signals up into periodic segments for each rotation of the shaft (according to the pulse locations returned by TachoSpinSpeed.m from a single pulse-per-revolution tachometer signal, such as the non-driving end tachometer in this case) and then calculating the average signal from these periodic segments.

[^26]Due to slight variations in the slow-roll speed, these periodic segments are generally not the exact same length and are thus interpolated to all have the same length as either the shortest (fastest) or the longest (slowest) segment, depending on the option selected in RunoutSP.m (the shortest segment in this case).

RunoutSP.m also calculates the approximate shaft/journal runout radii and displacement deviations from the ideal radius due to runout. Finally, the synchronous averages of the runout gap signals may be stored in CSV file format, if so-chosen by the user.

Figure 4.12 shows the slow-roll spin speed as a function of time calculated from both the driving end and non-driving end tachometer signals. Figure 4.13 and Figure 4.14 show the periodic segments, as functions of shaft angular displacement (per rotation), as well as the synchronous averaged runout gap signals (calculated from these periodic segments) from the $y$ - and $z$-probes at the bush and journal bearings respectively. The journal bearing signals seem to contain far less noise in comparison to the bush bearing signals - the driving motor and coupling may be sources of noise at the bush bearing. Since the $y$-and $z$-probes (at the same axial location) are separated by $90^{\circ}$, the runout signals recorded by these probes should be out-of-phase by $90^{\circ}$ with the $y$-probe signal lagging the z-probe signal due to the spin direction of the rotor (which is in the negative global $x$-direction ${ }^{31}$ ). There should also be a DC offset between the two signals since the probes are installed at different distances from the shaft. Figure 4.15 and Figure 4.16 show comparisons between the $y$ - and $z$-probe signals at the bush and journal bearings respectively where the $y$ probe signals have been shifted by $90^{\circ}$ and the average value (or DC offset) of each signal has been removed. The results show good correlations between the runout signals measured by the $y$ - and $z$-probes at each of these bearings (especially at the journal bearing), confirming that runout was measured accurately. Figure 4.17 illustrates the measured shaft and journal radii (including runout) from the $y$-probes at the bush and journal bearings respectively, superimposed on the ideal shaft and journal radii at these locations - the deviations from the ideal radii due to runout are, naturally, exaggerated (so that the maximum deviations are shown as $10 \%$ of the ideal radii) for better visibility.


Figure 4.12 - Rotor System Slow-Roll Spin Speed

[^27]

Figure 4.13 - Rotor System Bush Bearing Runout Gap Signal Synchronous Ave.


Figure 4.14 - Rotor System Journal Bearing Runout Gap Signal Synchronous Ave.


Figure 4.15 - Rotor System Bush Bearing y- \& z-Probe Runout Phase Correlation


Figure 4.16 - Rotor System Journal Bearing y- \& z-Probe Runout Phase Correlation


Figure 4.17 - Rotor System Bush Bearing \& Journal Bearing y-Probe Runout Radii

### 4.3.3 Vibration Measurement

The numerical rotor system models, including a nonlinear journal bearing, described in sections 3.5.1.2 and 3.5.2.2 predict that the first critical speed of the rotor system is approximately 1900 rpm . Lateral vibration signals were recorded for the rotor running up to approximately 2800 rpm (so that at least the first FW natural frequency is excited) from a starting spin speed of approximately 500 rpm (to coincide with the starting spin speed of the simulations discussed in sections 3.5.1.2 and 3.5.2.2) over approximately 10 s . Two different unbalance load cases were investigated: for case 1 , a 1 g weight was applied to disk 2 at an angle of $90^{\circ}$ from the reference mark (refer to Figure 4.1) and for case 2, a $2 g$ weight was applied at the same location. Since a residual unbalance remained after balancing was performed (refer to section 4.1), the total rotor imbalance in each case was a combination of the residual and applied imbalances. Control tests (not included here) were also conducted with only the residual unbalance.

A MATLAB program RotorVib.m (included in Appendix B.4), similar to Runout.m, was written to load and plot the raw recorded vibration signals from a TXT file. In addition to the required vibration and tachometer channels that need to be defined by the user, one or more thermocouple channels may also be specified. When executed, the program also calculates the sampling
frequencies, applies the time limits (specified by the user) and converts the vibration signals from $V$ to mm .

Another MATLAB program RotorVibSP.m (also included in Appendix B.4) was written to process the vibration signals loaded with RotorVib.m. When executed, this program determines the rotor spin speed as a function of time from the tachometer signals by utilising function TachoSpinSpeed.m the same function used in RunoutSP.m. The user may then choose whether to perform runout compensation. Before discussing this program further, the two runout compensation functions written and used by this program need to be described:

MATLAB functions RunoutComp1.m and RunoutComp2.m (included in Appendix B.5) were written to perform runout compensation on a single vibration signal according to methods 1 and 2 respectively (refer to section 4.2). Both these functions loop through the time segments in-between consecutive pulse locations ${ }^{32}$ from a single pulse-per-revolution tachometer (such as the non-driving end tachometer in this case) - each such segment is thus for one revolution of the rotor. The single-revolution runout gap signal (a function of shaft angular displacement provided as a function input) is interpolated to fit into each of these segments with the same number of data points as the vibration signal has in each corresponding segment. This then allows runout compensation to be performed segment-by-segment according to eqns. (4) and (4.2) for RunoutComp1.m or eqns. (4.6) and (4.7) for RunoutComp2.m. Within both these functions, the runout compensated segments are then assembled into one complete/composite signal returned by the function.

Returning to the description of RotorVibSP.m: In case the user selects to perform runout compensation, runout signals (corresponding to the vibration channels) are loaded from CSV files (indicated by the user during runtime) stored with RunoutSP.m (i.e. synchronous averaged runout signals). For each vibration channel, either function RunoutComp.m or RunoutCompRel.m is called, depending on the runout compensation method/type specified by the user in the code of RotorVibSP.m for each channel/pair of channels. These functions make use of the loaded runout signals to perform runout compensation. Function RunoutCompRel.m requires that the bearing clearance and gap size between the probe and bearing housing also be specified by the user in the code of RotorVibSP.m before running this program. Furthermore, RotorVibSP.m has an option to filter the signals with a band-pass filter for the purpose of isolating a specific component of the signal (e.g. the $1 \times$ spin speed component). This filter is thus not adequate for run-up response signals, as were analysed in this case. Finally, figures showing the processed signals are created and the user is provided with an option to store these signals in CSV file format.

### 4.3.3.1 Unbalance Load Case 1: $1 g$ weight at $90^{\circ}$ applied to disk 2

Figure 4.18 and Figure 4.19 show the measured $y$ - and $z$-probe vibration gap signals (before runout compensation) during run-up (from the second recorded measurement) at the bush and journal bearings respectively. Figure 4.20 shows the non-driving end tachometer signal. Note that the tachometer pulses come closer together as the rotor spin speed increases. Figure 4.21 shows the rotor spin speed during the run-up as a function of time, calculated from the tachometer signals. Figure 4.22 shows the journal bearing outlet lubricant temperature measured by the thermocouple which indicates an average temperature of $25.6^{\circ} \mathrm{C}$.

[^28]

Figure 4.18 - Rotor System Run-Up Bush Bearing Gap Signals - 1 g Unbalance


Figure 4.19 - Rotor System Run-Up Journal Bearing Gap Signals - 1 g Unbalance


Figure 4.20 - Rotor System Run-Up Non-Driving End Tachometer Signal


Figure 4.21 - Rotor System Run-Up Spin Speed - 1 g Unbalance Test


Figure 4.22 - Rotor System Run-Up Thermocouple Signal - 1 g Unbalance

Figure 4.23 to Figure 4.25 show the runout-compensated displacements at the bush and journal bearings. Figure 4.26 shows the 2D displacement of the journal bearing (from both the $y$ - and $z$ probes) as a function of time. Lines that represent the journal bearing clearance are also included in Figure 4.24 and Figure 4.25 (these lines were added using MATLAB program RotorVibClearance.m, included in Appendix B.4, which is based on another MATLAB program RotorDisplClearance.m briefly described in section 3.2.1.7 and included in Appendix C.1) - the high eccentricity of the journal bearing (eccentricity ratio $\varepsilon$ close to 1 ) indicates that it is relatively heavily loaded. The journal bearing vibration amplitude peaks at approximately 6 s ( 1880 rpm ), indicating a critical speed. Figure 4.25 and Figure 4.26 portray the same type of non-elliptical asymmetric orbits (near the critical speed) as discussed in section 3.5.1.2, but for the opposite spin direction. At approximately 6.32 s (1954 rpm), the shape and orientation of the whirl orbit changes, indicating a phase change as the critical speed is traversed. The DC components of the journal bearing displacements change as the spin speed increases - the journal moves closer towards the centre of the housing due to the effective bearing stiffness increasing with spin speed.


Figure 4.23 - Rotor System Run-Up Bush Bearing Displacements - 1 g Unbalance


Figure 4.24 - Rotor System Run-Up Journal Bearing Displacements \& Clearance - 1 g Unbalance


Figure 4.25 - Rotor System Run-Up Journal Bearing Displacement Orbits - 1 g Unbalance

Journal Bearing 2D Displacement as a Function of Time:


Figure 4.26 - Rotor System Run-Up Journal Bearing 2D Displacement - 1 g Unbalance

A MATLAB program RotorVibSpectral.m (included in Appendix B.4) was written to calculate and plot, amongst others, a spectrogram for the runout-compensated displacements from RotorVibSP.m. This program is based on MATLAB program RotorDisplSpectral.m, briefly described in section 3.2.1.5 and included in Appendix C.1). Figure 4.27 and Figure 4.28 show spectrograms of the vertical and horizontal displacements of the journal bearing. The inclined red lines at frequencies equal to the spin speed of the rotor (most visible in Figure 4.27) indicate synchronous whirling due to unbalance. The non-elliptical orbits near the critical speed cause harmonic lines to appear on the spectrograms, as discussed in section 3.5.1.2 (much more prominent for the second unbalance case - refer to Figure 4.35 and Figure 4.36). Other excitation sources, such as the dynamic reaction forces from the coupling may also cause harmonic lines (and other lines) on these spectrograms.


Figure 4.27 - Rotor System Run-Up Journal Bearing Vertical Displacement Spectrogram - 1 g Unbalance


Figure 4.28 - Rotor System Run-Up Journal Bearing Horizontal Displacement Spectrogram - 1 g Unbalance

### 4.3.3.2 Unbalance Load Case 1: 2 g weight at $90^{\circ}$ applied to disk 2

Figure 4.29 shows the rotor spin speed during the run-up (from the first recorded measurement), calculated from the tachometer signals. Figure 4.30 shows journal bearing outlet lubricant temperature measured by the thermocouple which indicates an average temperature of $25^{\circ} \mathrm{C}$. Figure 4.31 to Figure 4.34 show the runout-compensated displacements at the bush and journal bearings (including lines that represent the journal bearing clearance in Figure 4.32 and Figure 4.33). As with unbalance load case 1, the journal bearing vibration amplitude peaks at approximately $6 s(1880 \mathrm{rpm})$ which indicates a critical speed. The orbits of the journal are also non-elliptic near this critical speed. Due to the larger unbalance of load case 2, the non-elliptical asymmetric orbits near the critical speed are larger than- and have a different (flatter) shape compared to the orbits in unbalance load case 1. The effect that gravity and the fluid film forces seem to have on the journal motion, described in section 3.5.1.2, are more pronounced in this case. A phase change is also noted at approximately $6.3 \mathrm{~s}(1950 \mathrm{rpm})$ and the average position of the journal moves closer towards the centre of the housing due to stiffening of the bearing with an increase in spin speed.


Figure 4.29 - Rotor System Run-Up Spin Speed - $\mathbf{2 g}$ Unbalance Test


Figure 4.30 - Rotor System Run-Up Thermocouple Signal - 2 g Unbalance


Figure 4.31 - Rotor System Run-Up Bush Bearing Displacements - 2 g Unbalance


Figure 4.32 - Rotor System Run-Up Journal Bearing Displacements \& Clearance - 2 g Unbalance


Figure 4.33 - Rotor System Run-Up Journal Bearing Displacement Orbits - $\mathbf{2}$ g Unbalance


Figure 4.34 - Rotor System Run-Up Journal Bearing 2D Displacement - 2 g Unbalance

Figure 4.35 and Figure 4.36 show spectrograms of the vertical and horizontal displacements of the journal bearing (also created using RotorVibSpectral.m). Synchronous whirling due to unbalance is once again indicated by inclined red lines that track the spin speed. Integer harmonic lines are clearly visible near the critical speed due to the non-elliptical orbits caused by the nonlinear journal bearing.


Figure 4.35 - Rotor System Run-Up Journal Bearing Vertical Displacement Spectrogram - 2 g Unbalance


Figure 4.36 - Rotor System Run-Up Journal Bearing Horizontal Displacement Spectrogram - 2 g Unbalance

### 4.4 Transient Flat Surface Rubbing Analysis

This experiment was conducted to determine the dynamic response of the rotor system where the shaft is subject to rubbing with a horizontal flat surface (from above). The flat surface was located almost halfway between disks 1 and 2 and the shaft was brought into contact with this flat surface (intermittently) near the first critical speed of the rotor due to vibrations caused by unbalance forces.

### 4.4.1 Experimental Setup

The same experimental setup used for the transient unbalance response measurement was also used for this experiment, but with a few alterations. A pedestal with a vertically installed bronze bolt (which acted as the flat surface with which the shaft made contact), secured by a locking nut, was mounted between disks 1 and 2 (as shown in Figure 4.37). The sensor pedestal with two proximity probes, which had been mounted next to the bush bearing, was moved to a position right next to the pedestal with the bronze bolt (as also shown in Figure 4.37) and the horizontal probe was removed from the sensor pedestal.


Figure 4.37 - Rotor System Flat Surface Rubbing Response Experimental Setup

### 4.4.2 Runout Measurement

Runout from the proximity probes at the journal bearing and the vertical probe at the bronze bolt was measured at a slow-roll speed of approximately 101 rpm over a duration of 20 s . MATLAB programs Runout.m and RunoutSP.m (refer to section 4.3.2) were utilised here again to extract, plot and process the recorded runout signals, then store the calculated synchronous averages of the these signals.

Figure 4.38 and Figure 4.39 show the periodic segments of the runout gap signals at the rubbing location and journal bearing respectively as functions of shaft angular displacement (per rotation), as well as the synchronous averaged runout signals from the probes at these locations.


Figure 4.38 - Rotor System Rubbing Location Runout Gap Signal Synchronous Ave.


Figure 4.39 - Rotor System Journal Bearing Runout Gap Signal Synchronous Ave. - Rubbing Experiment

### 4.4.3 Vibration Measurement

The $1 g$ imbalance weight was once again applied to disk 2 at an angle of $90^{\circ}$ from the reference mark (refer to Figure 4.1). The bronze bolt was adjusted until the shaft would only come into contact with it (due to unbalance vibration) near the first critical speed. Lateral vibration signals were recorded for the rotor running up, approximately, from 500 rpm to 2800 rpm over 10 s (as for the transient unbalance response measurements).

MATLAB programs RotorVib.m and RotorVibSP.m (refer to section 4.3.3) were used here again to extract and process the recorded vibration signals (including performing runout compensation using the signals stored as CSV files with RunoutSP.m) and then plot the processed signals.

Figure 4.40 shows the vibration gap signal measured by the $y$-probe at the rubbing location during run-up (from the second recorded measurement) while Figure 4.41 shows the spin speed during the run-up, calculated from the tachometer signals. Figure 4.42 shows the runout-compensated vertical displacement at the rubbing location. The flattened section in the displacement (where the peaks in the positive $y$-direction which have more or less the same peak value) from approximately 5.5 to $7.7 \mathrm{~s}(1765$ to 2270 s$)$ is a result of the restriction imposed on the movement of the shaft by the flat surface (of the bronze bolt), i.e. a result of the rubbing forces acting on the shaft.


Figure 4.40 - Rotor System Run-Up Rubbing Location Gap Signal


Figure 4.41 - Rotor System Run-Up Spin Speed - Rubbing Experiment


Figure 4.42 - Rotor System Run-Up Rubbing Location Displacement

Figure 4.43 shows a spectrogram of the vertical displacement at the rubbing location which indicates the presence of some super-synchronous or harmonic vibrations caused by the rubbing.

This figure was created with MATLAB program RotorVibSpectral1Ch.m, a slightly modified version of RotorVibSpectral.m capable of handling a single channel signal such as the signal from the probe at the rubbing location.


Figure 4.43 - Rotor System Flat Surface Rubbing Vertical Displacement Spectrogram at Rubbing Location

## 5 EXPERIMENTAL VS. NUMERICAL RESULTS

In this chapter, simulated transient dynamic responses from the numerical rotor system models (detailed in chapter 3) are compared to the recorded transient dynamic responses from the experimental rotor system analysis (chapter 4) as a measure of the capabilities and accuracy of the numerical models.

### 5.1 Transient Unbalance Response

In order to compare unbalance responses from the numerical rotor system models to the unbalance responses of the experimental rotor system from section 4.3, simulations had to be performed with the same/similar excitations and operating conditions as were experienced during the experimental analysis of the rotor system.

To conform to the spin direction of the experimental rotor system (which is in the negative $x$ direction), the spin direction of the FE rotor system models from sections 3.5.1.2 and 3.5.2.2 would have to be reversed (e.g. by specifying the spin direction from the centre of disk 2 to the centre of disk 1 on the RSPINT entry) along with the spin direction of the journal bearing model (by specifying negative spin speeds in the FORTRAN subroutine SpinSpeed.f90 - refer to section 3.4.2). Furthermore, the initial conditions would have to be recalculated and re-applied based on the journal bearing equilibrium coordinates for the reversed spin direction. The only difference resulting from these changes would be an inversion of the displacements in the (global) $z$-direction. Thus, a simple way of dealing with this reversed spin direction is to perform simulations for the rotor spinning in the positive $x$-direction and then simply change the signs of resulting displacements in the z-direction. Naturally, the run-up speed profile (from 500 rpm to 2800 rpm in 10 s ) from the experimental analysis also had to be used for the simulations (which needed to be specified in the BDF input files as well as the FORTRAN subroutine SpinSpeed.f90).

Both unbalance load cases from the experimental analysis were modelled - however, since the experimental rotor had an unknown residual unbalance in addition to the applied imbalance weights (refer to section 4.1), this residual unbalance had to be accounted for in the rotor system models. The residual unbalance was approximated on a trial-and-error basis whereby the simulated unbalance response (specifically the displacement magnitude near the critical speed) was roughly matched with the experimental result for each unbalance load case. During the experiment rotor system analysis, the imbalance weights were both applied at angles of $90^{\circ}$ from the reference mark (measured opposite to the spin direction). These weights were therefore approximately located on the $y$-axis of the rotor local rotating coordinate system (as defined in section 2.1.2). To imitate this, the imbalances given in Table 5.1 were applied to disk 2 of the rotor system models, in addition to the approximated residual unbalance (where $\theta_{u 2}$ is the angle from the rotor local coordinate system $y$-axis to the imbalance in the spin direction - refer to Figure 3.16 and Figure 3.71 on pp. 74 and 112 respectively).

Table 5.1 - Applied Disk 2 Imbalances from Experimental Analysis used in Rotor System Models

| Unbalance Case | $m_{u 2}[t]$ | $r_{u 2}[\mathrm{~mm}]$ | $\boldsymbol{\theta}_{u 2}\left[^{\circ}\right]$ |
| :---: | :---: | :---: | :---: |
| 1 | $1 \times 10^{-6}$ | 30.75 | 0 |
| 2 | $2 \times 10^{-6}$ | 30.75 | 0 |

The residual unbalance, roughly approximated by trial-and-error, is given in Table 5.2. These approximate residual imbalance weights seem to be of a reasonable/realistic magnitude.

Table 5.2-Approximate Residual Unbalance of the Experimental Rotor System

| Disk | $m_{u}[t]$ | $r_{u}[\mathrm{~mm}]$ | $\boldsymbol{\theta}_{u}{ }^{\left[{ }^{\circ}\right]}$ |
| :---: | :---: | :---: | :---: |
| 1 | $5 \times 10^{-7}$ | 30.75 | 290 |
| 2 | $3 \times 10^{-7}$ | 30.75 | 180 |

Figure 5.1 and Figure 5.2 show the displacement results from the 1D rotor system model compared to the experimental results for unbalance load cases 1 and 2 respectively (created with RotorMultiPlot_TransCSV.m - briefly discussed in subsection two of section 3.2.1.5 and included in Appendix C.1). The enlarged part of Figure 5.2 shows similarly shaped displacements for the numerical and experimental results (although these displacements are out-of-phase). Figure 5.3 and Figure 5.4 show comparisons between the numerical and experimental 2D displacement results for load cases 1 and 2 respectively. Figure 5.5 shows a comparison between the numerical and experimental displacement orbits for load case 2. Figure 5.6 to Figure 5.10 show the same result comparisons for the 3D rotor system model. The numerical model results seem to be in good agreement with the experimental results (especially for the $2 g$ applied imbalance mass unbalance load case 2 ):

- Both results (numerical and experimental) indicate that the critical speed is at approximately 6 s (1880 rpm).
- Both results show a phase change as the critical speed is traversed at approximately 6.3 s (1950 rpm), although the phase change seems to feature more prominently in the numerical results.
- Both results show similar non-elliptical asymmetric orbits near the critical speed.
- The DC components of the numerical journal displacements correspond well with the experimental results, although the numerical results predict a slightly steeper rise in elevation (increase in the average $y$-coordinate) of the journal as its spin speed increases.

Discrepancies between the numerical and experimental results may be attributed, amongst others, to the simplifying assumptions of linear bush bearings and couplings in the numerical rotor system models - in reality, however, these support elements have pronounced nonlinear characteristics due to their synthetic rubber components. The rigid base and bearing pedestals modelling assumption may lead to the calculation of slightly higher natural frequencies and smaller vibration amplitudes than observed experimentally (Nicholas \& Barrett, 1985). Other factors that may also cause discrepancies are inaccurate approximation of the residual unbalance of the rotor, uncertainties regarding vibration excitation sources, rotor material properties, journal bearing lubricating fluid viscosity and other fluid properties, the difference between the axial locations where vibrations were observed during the experimental analyses and simulations, the simplified axisymmetric rotor model geometries, the simplified disk-shaft connections in the models,
numerical errors from discretisation (element types and meshes), the fluid film bearing model assumptions (Newtonian fluid with constant temperature and viscosity throughout the fluid film, etc.), and so forth.


Figure 5.1 - Experimental vs. 1D Model Run-Up Unbalance Response Displacements - 1g Unbalance


Figure 5.2 - Experimental vs. 1D Model Run-Up Unbalance Response Displacements - 2g Unbalance


Figure 5.3 - Experimental vs. 1D Model Run-Up Unbalance Response 2D Displacements - 1g Unbalance


Figure 5.4 - Experimental vs. 1D Model Run-Up Unbalance Response 2D Displacements - 2g Unbalance


Figure 5.5 - Experimental vs. 1D Model Run-Up Unbalance Response Orbits - 2g Unbalance

Vertical Displacement Comparison


Figure 5.6 - Experimental vs. 3D Model Run-Up Unbalance Response Displacements - 1g Unbalance


Figure 5.7 - Experimental vs. 3D Model Run-Up Unbalance Response Displacements - 2g Unbalance


Figure 5.8 - Experimental vs. 3D Model Run-Up Unbalance Response 2D Displacements - 1g Unbalance


Figure 5.9 - Experimental vs. 3D Model Run-Up Unbalance Response 2D Displacements - 2g Unbalance


Figure 5.10 - Experimental vs. 3D Model Run-Up Unbalance Response Orbits - 2g Unbalance

### 5.2 Transient Flat Surface Rubbing

To simulate the transient response of the experimental rotor system subject to flat surface rubbing (refer to section 4.4), the simple flat surface rubbing model described in section 3.2.1.7 was applied at node 29 (in between disks 1 and 2; closer to disk 2 - approximately where the bronze bolt made contact with the shaft during the experiment) of the 1 D model used for the $1 g$ unbalance load case simulation from the preceding section (5.1). A clearance of 0.09 mm from the bearing centreline was used (based on the experimental displacements).

Figure 5.11 shows the simulated vertical displacement result at node 25 (in between disks 1 and 2; closer to disk 1) compared to the experimental displacement recorded at approximately the same position along the shaft. The simulation predicts that the rubbing is initiated at approximately 6 s while rubbing was initiated at approximately 5.7 s during the experiment. The simulation also predicts that the rubbing ends a while later than the experimental result. The larger vibration amplitude observed from the experimental result before and after rubbing occurs (compared to the numerical result) may be attributed to added vibration from one or more additional excitation sources not considered in the model (such as coupling vibration). This may account for the earlier initiation of rubbing during the experiment compared to the rubbing initiation time predicted by the simulation. Other discrepancies may be attributed to uncertainty regarding contact stiffness as well as uncertainties mentioned in the preceding section (5.1). Both the numerical and experimental results show an abrupt decrease in vibration magnitude directly after the shaft loses contact with the flat surface.


Figure 5.11 - Experimental vs. 1D Model Run-Up Rubbing Location Displacements

## 6 Conclusions

## * Journal Bearing Model

A numerical model was developed (as described in section 3.3) for a general cylindrical finite-length hydrodynamic journal bearing whereby the Reynolds equation is solved through the finite difference numerical method and the resulting fluid film pressure distribution is numerically integrated over the journal surface area to obtain the fluid film forces. Transition of the pressure distribution from the load-carrying region of the bearing to the cavitation region is governed by Reynolds BCs which are in good agreement with experimental observations. The numerical model was programmed in both MATLAB and FORTRAN. Although the model was developed for a cylindrical bearing operating in the laminar region, it may easily be modified to accommodate different shapes (such as wedge- or 'lemon'-shapes) and additional fluid effects such as turbulent flow and fluid inertia effects. The model does however assume that the journal and housing are rigid and there is no misalignment/inclination in the axial direction. The capability of this bearing model to capture fluid-induced instabilities (oil whirl and oil whip) was demonstrated in sections 3.5.1.1 and 3.5.2.1.

## * Rotor System Characterisation and FE Rotor System Models

Before the FE rotor system models were developed, comprehensive characterisation of the experimental rotor system (Bently Nevada rotor kit) was conducted (as described in chapter 2). The densities of the rotor materials were approximated from the weights of the rotor components by constructing a solid model of the rotor and utilising a mass properties calculation tool within the solid modelling software. The elastic moduli of the rotor components were approximated by performing an experimental modal analysis and using its results to perform model updating of the 1D and 3D FE rotor models. Custom experiments were conducted to determine approximate linear dynamic stiffness and damping coefficients for the bush bearing and coupling connection.

Two FE models (1D and 3D models) of the experimental rotor system were developed based on the characterisation results. Linear direct stiffness and damping coefficients were initially used to model the journal bearing (as described in section 3.2). The rotor dynamic analysis capabilities of MSC.Nastran were explored during the systematic development of these models, as well as the effects of certain model features and rotor faults on the analysis results. Rotor dynamic analyses that were considered are synchronous and asynchronous complex modes analysis (to calculate critical speeds and generate Campbell diagrams), synchronous frequency response analysis (to determine steady state response due to unbalance) and nonlinear transient response analysis (to determine transient response due to unbalance and rubbing). In many cases, manual entry additions/modifications had to be made to the input files generated by MSC.Patran before submitting these files to MSC.Nastran to be solved. The capability of MSC.Nastran to perform rotor dynamic analyses using 3D element rotor models seems to be, to a large extent, unexplored and/or undocumented - this dissertation however explores this capability. Although 3D elements may be used to model the rotors, these 3D models must be reduced to line models in order to be analysed - dynamic reduction may only be used in modal formulations while static reduction is required for direct formulations. The shortcomings of static reduction in dynamic analyses seem to have a significant effect on torsional dynamic results, but a negligible effect on lateral dynamic results. Custom MATLAB programs were written to perform post-processing of the analysis results as well as signal processing for investigating the spectral content of transient results.

The software and model modifications required for incorporating the developed nonlinear journal bearing model into the FE rotor system models as a user-defined FORTRAN subroutine were discussed and nonlinear transient response (run-up) analyses were performed on the rotor system models with nonlinear journal bearing forces (as described in sections 3.4 and 3.5). Although these models were developed specifically for the experimental rotor system, the methodologies and software used to create and analyse these models were developed to be generic in nature for any flexible rotor system. The use of a similar finite-length hydrodynamic journal bearing nonlinear force model in MSC. Nastran has not been documented before. This added capability allows for the transient dynamic analysis of flexible rotor systems with nonlinear finite-length hydrodynamic journal bearings (such as turbogenerators) using the commercial FE software MSC.Nastran integrated with many of its powerful multidisciplinary modelling capabilities, such as its Rotor Dynamics capability, inertial (gravitational) load modelling, rubbing/contact modelling, damping modelling, differential stiffness modelling, capabilities included in the SOL 400 nonlinear transient response solution sequence, blade flutter analysis, fluid-structure interaction (FSI) modelling, etc. Furthermore, models may be constructed with scalar, 1D, 2D, 3D and many other elements available in MSC.Nastran.

## * Rotor System Experimental Analysis

The transient unbalance response of the experimental rotor system during run-up was investigated (as described in chapter 4). The transient response of the rotor during rotor-stator rubbing with a flat vertical surface above the shaft was also investigated. MATLAB programs were written to perform runout compensation (i.e. minimise unwanted runout distortion) on the measured vibration response signals by using runout signals measured during slow-roll. The first critical speed of the rotor system could be identified from the unbalance response and non-elliptical asymmetric orbits near the critical speed, which seem to be caused by the heavily loaded journal bearing (similar to simulated orbits), were noted.

## * Experimental Validation of Numerical Models

Simulations were performed (with the numerical rotor system models) to emulate the experimental analyses performed on the rotor system (refer to chapter 5). Overall, the numerical results are in good agreement with the experimental results which indicate that the models are able to sufficiently capture important dynamics involved in the rotor system, especially the highly nonlinear behaviour of a journal bearing with large eccentricities (heavily loaded). Discrepancies between the numerical and experimental results may be attributed to a number of simplifying assumptions made during modelling, inherent errors of the numerical methods used and uncertainties regarding the true physical properties of the experimental rotor system. The developed numerical rotor system models may therefore serve as benchmarks for data measured on the experimental rotor system which may be used to develop diagnostic data processing techniques for the research project discussed in section 1.1.

## * Recommendations for Future/Additional Work

Future work with regards to FE rotor system modelling in MSC.Nastran could include the following:

- Nonlinear frequency response analysis.
- Nonlinear complex modes analysis - a feature that may become available in future releases of MSC.Nastran.
- Rotor casing modelling - including rotor-casing interactions and foundation stiffness.
- Turbine blade modelling - including differential stiffness from centrifugal loading (which may be included in MSC.Nastran via the RFORCE entry).
- The MSC.Nastran Rotor Dynamics capability is supported in aero solution sequences (refer to MSC.Nastran Release Guide 2006) - this could possibly be used to model dynamic effects such as blade flutter.
- Coupling misalignment static and dynamic forces and moments (reference may be made to the model developed by Xu and Marangoni, 1994).
- MSC.Nastran has the capability to model FSI with external or third-party computational fluid dynamics (CFD) software which could be used to incorporate more complex journal bearing models - certain commercial CFD solvers are supported, but a user may also write CFD software in C/C++ (refer to any MSC.Nastran User Defined Services manual from 2012 onwards).
- Non-axisymmetric rotor modelling in rotating coordinate systems - a feature that may become available in future releases of MSC.Nastran; this feature would allow the modelling of shaft cracks.

Newer models of the Bently Nevada rotor kit (similar to the setup used for this dissertation) include a device that permits the adjustment of the journal bearing journal position within its bearing housing - this allows for easy experimental investigation of oil whirl and oil whip which could also be beneficial to the Eskom project and other future rotor dynamic research.

## References

Abdul Azeez, M. F. \& Vakakis, A.F. (1999), Numerical and Experimental Anslysis of a Continuous Overhung Rotor Undergoing Vibro-Impacts, International Journal of Non-Linear Mechanics, vol. 34, pp. 415-435

Ahuja, S. \& Sharan, A.M. (1984), Balancing of Finite Element Modelled Rotor-Disks using Dynamic Matrix Reduction Technique in Conjunction with Modal Analysis and Least Square Analysis, Accepted for presentation at the Eighth Machinery Dynamics Seminar to be held at Halifax, Nova Scotia, Canada

Bently, D.E. \& Hatch, C.T. (2002), Fundamentals of Rotating Machinery Diagnostics, Bently Pressurised Bearing Press

Berry, J.E. (2005), Oil Whirl and Whip Instabilities - Within Journal Bearings, Machinery Lubrication, Available at: http://www.machinerylubrication.com/Read/754/oil-whirl-whip [2012/01/17]

Blakely, K. (1991), Updating MSC/Nastran Models to Match Test Data, The MSC 1991 World Users' Conference Proceedings, vol. II, paper 50

Budynas, R.G. \& Nisbett, J.K. (2008), Shigley's Mechanical Engineering Design, $8^{\text {th }}$ ed., McGrawHill: Singapore

Cao, J., Ma, C., Jiang, Z. \& Liu, S. (2011), Nonlinear Dynamic Analysis of Fractional Order RubImpact Rotor System, Communications in Nonlinear Science and Numerical Simulation, vol. 16, pp. 1443-1463

Cheng, M., Meng, G. \& Jing, J. (2007), Numerical and Experimental Study of a Rotor-Bearing-Seal System, Mechanism and Machine Theory, vol. 42, pp. 1043-1057

Chouksey, M., Dutt, J.K. \& Modak, S.V. (2012), Modal Analysis of Rotor-Shaft System under the Influence of Rotor-Shaft Material Damping and Fluid Film Forces, Mechanism and Machine Theory, vol. 48, pp. 81-93

Choy, F.K. \& Padovan, J. (1987), Non-Linear Transient Analysis of Rotor-Casing Rub Events, Journal of Sound and Vibration, vol. 113, pp. 529-545

Curzon, A.E. (1986), A Simple Sixth Order Numerical Method for Solving the Reynolds Equation, Tribology International, vol. 18, pp. 321-325

Dai, X., Jin, Z. \& Zhang, X. (2002), Dynamic Behavior of the Full Rotor/Stop Rubbing: Numerical Simulation and Experimental Verification, Journal of Sound and Vibration, vol. 251, pp. 807-822

De Castro, H.F., Cavalca, K.L. \& Nordmann, R. (2008), Whirl and Whip Instabilities in Rotor-Bearing System Considering a Nonlinear Force Model, Journal of Sound and Vibration, vol. 317, pp. 273293

Dostal, M., Roberts, J.B. \& Holmes, R. (1974), Stability Control of Flexible Shafts Supported on OilFilm Bearings, Journal of Sound and Vibration, vol. 35, pp. 361-377

Dostal, M., Roberts, J.B. \& Holmes, R. (1977), The Effect of External Damping on the Vibration of Flexible Shafts Supported on Oil-Film Bearings, Journal of Sound and Vibration, vol. 51, pp. 69-87

DuBois, G.B. \& Ocvirk, F.W. (1952), Analytical Derivation and Experimental Evaluation of ShortBearing Approximation for Full Journal Bearings, Report 1157 - National Advisory Committee for Aeronautics

Fan, C., Syu, J., Pan, M. \& Tsao, W. (2011), Study of Start-Up Vibration Response for Oil Whirl, Oil Whip and Dry Whip, Mechanical Systems and Signal Processing, vol. 25, pp. 3102-3115

Fredö, C.R., Andrén, P. \& Falk, T. (1999), Mechanical Damping Simulation in MSC.Nastran, Nordic MSC User's Conference, Gothenburg

Frew, D.A. \& Scheffer, C. (2008), Numerical Modelling of a High-Speed Rigid Rotor in a SingleAerostatic Bearing using Modified Euler Equations of Motion, Mechanical Systems and Signal Processing, vol. 22, pp. 133-154

Friswell, M.I., Penny, J.E.T., Garvey, S.D. \& Lees, A.W. (2001), Damping Ratio and Natural Frequency Bifurcations in Rotating Systems, Journal of Sound and Vibration, vol. 245, pp. 960-967

Genta, G. \& Delprete, C. (1995), Acceleration Through Critical Speeds of an Anisotropic, NonLinear, Torsionally Stiff Rotor with Many Degrees of Freedom, Journal of Sound and Vibration, vol. 180, pp. 369-386

Genta, G. (1988), Whirling of Unsymmetrical Rotors: a Finite Element Approach based on Complex Co-ordinates, Journal of Sound and Vibration, vol. 124, pp. 27-53

Gunter, E.J., Barrett, L.E. \& Allaire, P.E. (1975), Design and Application of Squeeze Film Dampers for Turbomachinery Stabilization, Proceedings of the Fourth Turbomachinery Symposium, pp. 127-141

Heyns, P.S. (2005), psde.m [MATLAB code], Vibration-Based Condition Monitoring, University of Pretoria, South Africa

Heyns, P.S. (2011), Model Updating and Modification [lecture notes], Experimental Structural Dynamics, University of Pretoria, South Africa

Holmes, A.G. (1975), A Study of Iterative Solution Techniques for Elliptic Partial Differential Equations with Particular Reference to the Reynolds Equation, Computer Methods in Applied Mechanics and Engineering, vol. 5, pp. 309-328

Holmes, R. (1980), The Role of Oil-Film Bearings in Promoting Shaft Instability and the Remedial Effect of Damping, Tribology International, October 1980, pp. 243-248

Hsieh, S.-C., Chen, J.-H. \& Lee, A.-C. (2006), A Modified Transfer Matrix Method for the Coupling Lateral and Torsional Vibrations of Symmetric Rotor-Bearing Systems, Journal of Sound and Vibration, vol. 289, pp. 294-333

Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N., Tung, C.C. \& Liu, H.H. (1998), The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and NonStationary Time Series Analysis, Proceedings of the Royal Society A: Mathematical, Physical \& Engineering Sciences, vol. 454, pp. 903-995

Jei, Y.-G. \& Lee, C.-W. (1992), Modal Analysis of Continuous Asymmetrical Rotor-Bearing Systems, Journal of Sound and Vibration, vol. 152, pp. 245-262

Kr. Jalan, A. \& Mohanty, A.R. (2009), Model Based Fault Diagnosis of a Rotor-Bearing System for

Misalignment and Unbalance under Steady-State Condition, Journal of Sound and Vibration, vol. 327, pp. 604-622

Lawrence, C. (2004), An MSC.Nastran Primer for Rotordynamics, Virtual Product Development Conference: Huntington Beach, California

Lee, C.-W. \& Jei, Y.-G. (1988), Modal Analysis of Continuous Rotor-Bearing Systems, Journal of Sound and Vibration, vol. 126, pp. 345-361

Lee, Y.-S. \& Lee, C.-W. (1999), Modelling and Vibration Analysis of Misaligned Rotor-Ball Bearing Systems, Journal of Sound and Vibration, vol. 224, pp. 17-32

Li, W., Yang, Y., Sheng, D. \& Jianhong, C. (2011), A Novel Nonlinear Model of Rotor/Bearing/Seal System and Numerical Analysis, Mechanism and Machine Theory, vol. 46, pp. 618-631

Lin, J.-F. \& Wang, L.-Y. (1990), Thermohydrodynamic Analysis of Finite-Width, Partial-Arc Journal Bearings with Non-Newtonian Lubricants: Part II, Tribology International, vol. 23, pp. 211-216

Lin, Y.-H. \& Lin, S.-C. (2001), Optimal Weight Design of Rotor Systems with Oil-Film Bearings subject to Frequency Constraints, Finite Elements in Analysis and Design, vol. 37, pp. 777-798

Littrell, N. (2005), Understanding and Mitigating Shaft Runout, Orbit, vol. 25, pp. 5-17

Mohiuddin, M.A., Bettayeb, M. \& Khulief, Y.A. (1998), Dynamic Analysis and Reduced Order Modelling of Flexible Rotor-Bearing Systems, Computers and Structures, vol. 69, pp. 349-359

Movahhedy, M.R. \& Mosaddegh, P. (2006), Prediction of Chatter in High Speed Milling including Gyroscopic Effects, International Journal of Machine Tools and Manufacture, vol. 46, pp. 9961001

MSC.Software (2003), MSC.Nastran 2004 Reference Manual [user documentation]
MSC.Software (2004), MSC.Nastran 2001 Superelement User's Guide [user documentation]
MSC.Software (2006), MD Nastran 2006 Release Guide [user documentation]
MSC.Software (2011), MSC Software Development Kit 2012 User's Guide [user documentation]
MSC.Software (2011), MSC.Nastran 2012 Demonstration Problems [user documentation]
MSC.Software (2011), MSC.Nastran 2012 Design Sensitivity and Optimization User's Guide [user documentation]

MSC.Software (2011), MSC.Nastran 2012 Quick Reference Guide [user documentation]
MSC.Software (2011), MSC.Nastran 2012 User Defined Services [user documentation]
Nelson, F.C. (2007), Rotor Dynamics without Equations, International Journal of Comadem, vol. 10, pp. 2-10

Nicholas, J.C. \& Barrett, L.E. (1985), The Effect of Bearing Support Flexibility on Critical Speed Prediction, Rotor Bearing Dynamics Inc. $40^{\text {th }}$ Annual Meeting, Las Vegas, Nevada

Oak, M. (2008), What is Gyroscopic Effect?, Buzzle.com, Available at:
http://www.buzzle.com/articles/what-is-gyroscopic-effect.html [2011/03/07]
Padovan, J., Adams, M., Fertis, D., Zeid, I. \& Lam, P. (1984), Nonlinear Transient Finite Element Analysis of Rotor-Bearing-Stator Systems, Computers and Structures, vol. 18, pp. 629-639

Prata, A.T.; Ferriera, R.T.S.S.; Mile, D.E.B. \& Bortoli, M.G.D. (1988), Dynamically Loaded Journal Bearings: Finite Volume Method Analysis, International Compressor Engineering Conference, paper 599

Qiu, Z.L. \& Tieu, A.K. (1997), Identification of Sixteen Force Coefficients of Two Journal Bearings form Impulse Responses, Wear, vol. 212, pp. 206-212

Randall, R.B. (2011), Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications, John Wiley and Sons: United Kingdom

Rao, J.S. \& Sharan, A.M. (1994), The Calculation of the Natural Frequencies of Multi-Disk-Rotor Systems using the Influence Coefficient Method including the Gyroscopic Effects, Mechanical Machine Theory, vol. 29, pp. 739-748

Rao, J.S. (1982), Conditions for Backward Synchronous Whirl of a Flexible Rotor in Hydrodynamic Bearings, Mechanism and Machine Theory, vol. 17, pp. 143-152

Rao, J.S. (1985), Instability of Rotors Mounted in Fluid-Film Bearings with a Negative CrossCoupled Stiffness Coefficient, Mechanism and Machine Theory, vol. 20, pp. 181-187

Rao, J.S. (1996), Rotor Dynamics, $3^{\text {rd }}$ ed., New Age International: New Delhi
Rao, S.S. (2004), Mechanical Vibrations, $4^{\text {th }}$ ed., Pearson Prentice Hall: New Jersey
Reynolds, O. (1886), On the Theory of Lubrication and its Application to Mr Tower's Experiments, Philosophical Transactions of the Royal Society A: Mathematical, Physical \& Engineering Sciences, vol. 177, pp. 159-209

Roques, S., Legrand, M., Cartraud, P., Stoisser, C. \& Pierre, C. (2010), Modelling of a Rotor Speed Transient Response with Radial Rubbing, Journal of Sound and Vibration, vol. 329, pp. 527-546

RPM Turbo (2005), Flutter Analysis, RPM Turbo, Available at: http://www.rpmturbo.com/flutter/index.html [2011/02/15]

Ruhl, R.I. \& Booker, J.F. (1972), A Finite Element Model for Distributed Parameter Turbo-Rotor Systems, ASME Journal of Engineering for Industry, vol. 94, pp. 126-132

Saad, Y. (2003), Iterative Methods for Sparse Linear Systems, $2^{\text {nd }}$ ed., Society for Industrial and Applied Mathematics

San Andrés, L. (2006), Hydrodynamic Fluid Film Bearings and Their Effect on the Stability of Rotating Machinery [lecture notes], In the Design and Analysis of High Speed Pumps (pp. 10-1 to 10-36), Texas A\&M University, United States of America

Sharan, A.M. \& Rao, J.S. (1985), Unbalance Response of Rotor Disks Supported by Fluid Film Bearings with a Negative Cross Coupled Stiffness using the Influence Coefficient Method, Mechanism and Machine Theory, vol. 20, pp. 415-426

Singhal, G.C. (1981), Computational Methods for Hydrodynamic Problems (Reynold's Equation),

Computer-Aided Design, vol. 13, pp. 151-154
Sommerfeld, A. (1904), The Hydrodynamic Theory of Lubrication Friction, Z. Math. und Physik, vol. 50, pp. 97-155

Surial, A. \& Kaushal, A. (1999), Dynamic Analysis of Flexible Turbo-Rotor System using SuperElements, The MSC 1999 Aerospace User's Conference Proceedings, paper 1799

Szeri, A.Z. (2011), Fluid Film Lubrication, $2^{\text {nd }}$ ed., Cambridge University Press: New York
Tiwari, R. \& Chakravarthy, V. (2006), Simultaneous Identification of Residual Unbalances and Bearing Dynamic Parameters from Impulse Response of Rotor-Bearing Systems, Mechanical Systems and Signal Processing, vol. 20, pp. 1590-1614

Tiwari, R. \& Chakravarthy, V. (2009), Simultaneous Estimation of the Residual Unbalance and Bearing Dynamic Parameters from the Experimental Data in a Rotor-Bearing System, Mechanism and Machine Theory, vol. 44, pp. 792-812

Vignolo, G.G., Barilá, D.O. \& Quinzani, L.M. (2011), Approximate Analytical Solution to Reynolds Equation for Finite Length Journal Bearings, Tribology International, vol. 44, pp. 1089-1099

Xiang, J., Chen, D., Chen, X. \& He, Z. (2009), A Novel Wavelet-Based Finite Element Method for the Analysis of Rotor-Bearing Systems, Finite Elements in Analysis \& Design, vol. 45, pp. 908-916

Xu, M. \& Marangoni, R.D. (1994), Vibration Analysis of a Motor-Flexible Coupling-Rotor System Subject to Misalignment \& Unbalance, Part 1: Theoretical Model and Analysis, Journal of Sound and Vibration, vol. 176, pp. 663-679

Yungong, L., Jinping, Z., Liqiang, W. \& Yang, C. (2011), A Fault Feature Extraction Method for Rotor Rubbing Based on Load Identification and Measured Impact Response, Procedia Engineering, vol. 24, pp. 793-797
 Python ${ }^{\text {M }}$, SolidWorks ${ }^{\circledR}$, Polytec, VibraTools ${ }^{\text {™ }}$ and all other brand or product names are (registered) trademarks of their respective owners.

## Appendix A: Rotor System Drawings, Diagrams and Images

## A. 1 Experimental Rotor System

Experimental Rotor Assembly Drawing


Experimental Rotor System Assembly Drawing


## A. 2 1D Rotor System Model Schematic Diagrams

1D Rotor System Model Dimensions



## A. 3 3D Rotor System Model Schematic Diagrams

```
3D Rotor System Model Dimensions
```




## A. 4 3D Rotor Model in MSC.Patran

Hexahedral Element Mesh


Rear View (Coupling End)
Front View (Journal End)



## Mesh Quality

## Aspect Ratio:



## Edge Angle:



## Face Skew:



## Appendix B: MATLAB Programs - Experimental Analyses

## B. 1 Bush Bearing Characterisation

## BushBearingChar.m

```
% Program that loads, processes & plots two LVDT displacement signals from
% a specific bush bearing characterisation experiment (bush bearings: from
% the Bently Nevada Rotor Kit in the Sasol Laboratory at the University of
% Pretoria). The signals are step response measurements measured on a shaft
% supported by two bush bearings.
。
% Note: User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: July }201
% Last revised: February 2013
clear all; clc; close all
```



```
%Load Raw Data:
run = 1; %::Measurement number
wl = '2k'; %::Static load weight(s) (& measurement type)
filename = ['db' wl '_' num2str(run) '.txt'];
data = load(filename);
disp(['Data read from file: ',filename])
t = data(:,1); displA_raw = data(:,2); displB_raw = data(:,3);
%Determine Sampling Frequency:
dt = t(2) - t(1);
fs = round(1/dt); dt = 1/fs;
disp(['Sampling frequency = ',num2str(fs),' Hz'])
%Correctly Reconstruct Time Vector:
t = 0:dt:dt*(length(t)-1);
%Plot Raw Data:
figure(1)
handle = plot(t,displA_raw,t,displB_raw);
set(handle(1),'color', [0 0.7 0.9])
set(handle(2),'color',[0.5 0.8 0.2])
title('Raw Bush Bearing Step Response Signals:')
xlabel('t [s]'),ylabel('LVDT Signal [mV]')
legend('LVDT A','LVDT B'),grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONVERSION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Convert Data:
sensA = 40.098; %::LVDT A sensitivity [mV/mm]
sensB = 40.589; %::LVDT B sensitivity [mV/mm]
offstA = -displA_raw(1)/sensA; displA = -displA_raw/sensA - offstA;
offstB = -displB_raw(1)/sensB; displB = -displB_raw/sensB - offstB;
%Plot Converted Data:
figure(2)
handle = plot(t,displA,t,displB);
set(handle(1),'color',[0 0.7 0.9])
set(handle(2),'color',[0.5 0.8 0.2])
title('Bush Bearing Step Response Signals:')
xlabel('t [s]'),ylabel('x [mm]')
legend('LVDT A','LVDT B'),grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INTERPOLATION % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % 
%Interpolate Data:
lA1 = 21 + 5; %::Dimension [mm]
112 = 17 + 25*2 - 5*2; %::Dimension [mm]
lB2 = 23 + 5; %::Dimension [mm]
l = lA1 + l12 + lB2;
displ1 = lA1/l*(displB - displA) + displA;
```

```
displ2 = lB2/l*(displA - displB) + displB;
%Plot Interpolated Data:
figure(3)
handle = plot(t,displ1,t,displ2);
set(handle(1),'color',[0 0.7 0.9])
set(handle(2),'color',[0.5 0.8 0.2])
title('Bush Bearing Step Response Signals:')
xlabel('t [s]'),ylabel('x [mm]')
legend('Bearing 1','Bearing 2'),grid on
```


## BushBearingCharSP.m

```
Program that applies signal processing to signals loaded by
BushBearingChar.m in order to determine the step response frequency (or
frequencies) and filter-out other frequencies.
Note: *First run BushBearingChar.m.
    *Custom functions dft.m & expwindow.m are required in the working
        directory to run this program.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
Created: July }201
% Last revised: February }201
%%%%%%%%%%%%%%%%%%%%%% SIGNAL RESAMPLING (if required) % % % % % % % % % % % % % % % % % % % % %
if fs > 1600
    STF = round(fs/1600); %Sampling time scale factor
else
    STF = 1; %Sampling time scale factor
end
if STF ~= 1
    dt = STF*dt;
    fs = 1/dt;
    disp(['New sampling frequency = ',num2str(fs),' Hz'])
    t_old = t; displ1_old = displ1; displ2_old = displ2;
    t}\mp@subsup{}{}{-}=t(1):dt:t(end)
    displ1 = interp1(t_old,displ1_old,t)';
    displ2 = interp1(t_old,displ2_old,t)';
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIGNAL CROPPING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Crop Signal:
switch wl
    case 'hk'
            tm = [2.198 2.588 3.862 2.399 1.793 2.394 %::Signal cropping times
            2.45 2.9 4.2 2.61 2 2.65];
    case '1k'
            tm}=[\begin{array}{lllllllll}{1.42}&{6.375}&{2.931}&{2.91}&{2.761}&{1.789}
            1.8 6.7 3.5 3.16 3.08 2.16];
        case '1hk'
            tm=[llllllllll
            3 2.76 3.65 3.5];
        case '2k'
            tm}=[\begin{array}{llllllllll}{3.367}&{3.301}&{2.611}&{2.5215}&{1.567}&{1.426}
            4 3.621 2.95 2.8 1.9 1.66];
        case '1ktest'
            tm}=[2.93
                        3.2];
end
tstart = tm(1,run); tend = tm(2,run);
indx = find(t>=tstart & t<=tend)';
t_c = t(indx); displ1_c = displ1(indx); displ2_c = displ2(indx);
%Spectral Analysis:
N = 2^(nextpow2(length(t c)));
[f,Fn1] = dft(displ1_c,N,fs); [f,Fn2] = dft(displ2_c,N,fs);
```

```
%Plot Results:
figure(4)
handle = plot(t_c,displ1_c,t c,displ2 c);
set(handle(1),'Color', [0-0.7-0.9])
set(handle(2),'color',[0.5 0.8 0.2])
title('Cropped Bush Bearing Step Response Signals:')
xlabel('t [s]'),ylabel('Displacement [mm]')
legend('Bearing 1','Bearing 2'),grid on
figure(5)
subplot(2,1,1)
handle = plot(f,abs(Fn1)); set(handle,'color',[0 0.7 0.9])
title('DFT of Cropped Bush Bearing 1 Step Response Signal:')
xlabel('f [Hz]'),ylabel('Displacement Amplitude [mm]'),grid on
subplot(2,1,2)
handle = plot(f,abs(Fn2)); set(handle,'color',[0 0.7 0.9])
title('DFT of Cropped Bush Bearing 2 Step Response Signal:')
xlabel('f [Hz]'),ylabel('Displacement Amplitude [mm]'),grid on
%%%%%%%%%%%%%%%%%%%%%% SIGNAL DETRENDING & WINDOWING %%%%%%%%%%%%%%%%%%%%%%%%%
%Detrend Signal:
nbp = 15; %::Number of break points
l = length(t c); indx = 1:l; dbp = round((l-1)/nbp);
n = 1; bp = zeros(nbp-1,1);
for k = dbp:dbp:l-dbp
    bp(n) = indx(k);
    n = n + 1;
end
displ1_old = displ1_c; displ2_old = displ2_c;
```



```
detr1 = displ1_old - displ1_c; detr2 = displ2_ol\overline{d}- displ2_c;
%Apply Exponential Window to Signal:
sf = 70; %::Percentage of t at which w = 0.01 [%]
w = expwindow(l,sf/100)';
displ1_w = w.*displ1_c; displ2_w = w.*displ2_c;
%Spectral Analysis:
[f,Fn1] = dft(displ1_w,N,fs); [f,Fn2] = dft(displ2_w,N,fs);
%Plot Results:
figure(6)
handle = plot(t_c,displ1 old,t_c,displ2_old,t_c,detr1,t_c,detr2);
set(handle(1),'Color', [0 0.7 0.9])
set(handle(2),'color',[0.5 0.8 0.2])
set(handle(3),'color','r')
set(handle(4),'color',[1 0.6 0])
title('Bush Bearing Step Response Signals with Trends:')
xlabel('t [s]'),ylabel('Displacement [mm]')
legend('Bearing 1','Bearing 2','Piecewise Linear Trend 1','Piecewise Linear Trend
2'),grid on
figure(7)
handle = plot(t c,displ1 c,t c,displ2 c);
set(handle(1),'Color', [0`0.7-0.9])
set(handle(2),'color',[0.5 0.8 0.2])
title('Detrended Bush Bearing Step Response Signals:')
xlabel('t [s]'),ylabel('Displacement [mm]')
legend('Bearing 1','Bearing 2'),grid on
figure(8)
handle = plot(t_c,displ1_w,t_c,displ2_w);
set(handle(1),'color', [0 0.7 0.9])
set(handle(2),'color',[0.5 0.8 0.2])
title('Detrended & Windowed Bush Bearing Step Response Signals:')
xlabel('t [s]'),ylabel('Displacement [mm]')
legend('Bearing 1','Bearing 2'),grid on
figure(9)
subplot(2,1,1)
```

```
handle = plot(f,abs(Fn1)); set(handle,'color',[0 0.7 0.9])
title('DFT of Detrended & Windowed Bush Bearing 1 Step Response Signal:')
xlabel('f [Hz]'),ylabel('Displacement Amplitude [mm]'),grid on
subplot(2,1,2)
handle = plot(f,abs(Fn2)); set(handle,'color',[0 0.7 0.9])
title('DFT of Detrended & Windowed Bush Bearing 2 Step Response Signal:')
xlabel('f [Hz]'),ylabel('Displacement Amplitude [mm]'),grid on
figure(10)
handle = plot(f,abs(Fn1),f,abs(Fn2));
set(handle(1),'color',[0 0.7 0.9])
set(handle(2),'color',[0.5 0.8 0.2])
title('DFT of Detrended & Windowed Bush Bearing Step Response Signals:')
xlabel('f [Hz]'),ylabel('Displacement Amplitude [mm]')
legend('Bearing 1','Bearing 2'),grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTERING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Low-/Band-Pass Filtering:
ftype = 'band'; %::Filter type (low/band)
switch ftype
    case 'low'
    if strcmp(wl,'hk')
        if fs < 1250
            fc = 130; %::Cut-off frequency [Hz]
            else
            fc = 110;
            end
            ord = 15;
        elseif strcmp(wl,'1k')
            fc = 95; %::Cut-off frequency [Hz]
            ord = 20;
        elseif strcmp(wl,'1hk')
            fc = 70;
            ord = 15; %::Filter order
        else %wl = '2k'
            fc = 28; %::Cut-off frequency [Hz]
            ord = 15; %::Filter order
        end
        [b,a] = butter(ord,fc/(fs/2));
        h = dfilt.df2t(b,a);
        fvtool(h)
        displ1_filt = filter(h,displ1_c); displ2_filt = filter(h,displ2_c);
    case 'band'
        if strcmp(wl,'hk')
            flc = 144.5;
            fhc = 148.5;
                    %::Lower cut-off frequency [Hz]
                                    %::Upper cut-off frequency [Hz]
            if fs < 1250
                    ord = 100;
                    %::Filter order
            else
                    ord = 130; %::Filter order
                end
            elseif strcmp(wl,'1k')
                fhc = 86; %::Upper cut-off frequency [Hz]
                if fs < 1250
                    ord = 100;
                    %::Filter order
                else
                    ord = 130; %::Filter order
                end
        elseif strcmp(wl,'1hk')
                flc = 73; %::Lower cut-off frequency [Hz]
                fhc = 79; %::Upper cut-off frequency [Hz]
                if fs < 1250
                    ord = 100; %::Filter order
            else
                ord = 120; %::Filter order
            end
        else %wp = '2k'
            flc = 66; %::Lower cut-off frequency [Hz]
            fhc = 70; %::Upper cut-off frequency [Hz]
```

```
    if fs < 1250
    ord = 100; %::Filter order
else
    ord = 130; %::Filter order
    end
end
d = fdesign.bandpass('n,fc1,fc2',ord,flc,fhc,fs);
h = design(d);
fvtool(h)
displ1_filt = filtfilt(h.Numerator,1,displ1_c);
displ2_filt = filtfilt(h.Numerator,1,displ2_c);
end
%Detrend Signal:
nbp = 20; %::Number of break points
dbp = round((l-1)/nbp);
n = 1;
for k = dbp:dbp:l-dbp
    bp(n) = indx(k);
    n = n + 1;
end
displ1_filt = detrend(displ1_filt,'linear',bp); displ2_filt =
detrend(displ2_filt,'linear',bp);
%Spectral Analysis:
[f,Fn1_filt] = dft(displ1_filt,N,fs); [f,Fn2_filt] = dft(displ2_filt,N,fs);
%Plot Results:
filtfig = 12;
figure(filtfig)
handle = plot(t_c,displ1_filt,t_c,displ2_filt);
set(handle(1),'\overline{color', [0-0.7 0. '9])}
set(handle(2),'color',[0.5 0.8 0.2])
title('Detrended & Filtered Bush Bearing Step Response Signals:')
xlabel('t [s]'),ylabel('Displacement [mm]')
legend('Bearing 1','Bearing 2'),grid on
figure(13)
subplot (2,1,1)
handle = plot(f,abs(Fn1_filt)); set(handle,'color',[0 0.7 0.9])
title('DFT of Filtered Bush Bearing 1 Step Response Signal:')
xlabel('f [Hz]'),ylabel('Displacement Amplitude [mm]'),grid on
subplot (2,1,2)
handle = plot(f,abs(Fn2_filt)); set(handle,'color',[0 0.7 0.9])
title('DFT of Filtered Bush Bearing 2 Step Response Signal:')
xlabel('f [Hz]'),ylabel('Displacement Amplitude [mm]'),grid on
```


## db2k_1_bearing1_kc.m

```
Program that determines approximate linear dynamic stiffness- & damping
coefficients from the isolated (cropped & filtered) step response signals
from file db2k_1.txt (processed by BushBearingCharSP.m).
Note: *First run BushBearingCharSP.m.
    *User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
Created: September 2012
Last revised: February 2013
%Data File- & Signal Processing Information & -Settings:
    File: db2k_1.txt
    Bearing: bearing 1
    BushBearingCharSP.m settings:
        tstart = 3.367
        tend = 4
        n.bp (1) = 15
        ftype = 'band'
        flc = 66
        fhc = 70
```

```
% ord = 100
% nbp (2) = 20
%Input Parameters:
fd = 68; %::Damped natural frequency [Hz]
rk = 0.922; %::k2/k1 ratio
m = 0.0672 + 0.051 + 2 + 0.0392; %::Total vibrating mass [kg]
zeta_0 = 0.05; 表:Damping ratio starting guess value
scale = le-4; %::Data points scale value
% ::Positive- & negative data point t-values:
t_pos = [3.4 3.415 3.429 3.444 3.458 3.473 3.487]';
t_neg = [ll.378 3.3925 3.4075 3.422 3.4365 3.451 3.466 3.48]';
%Data Points x-Values:
x_pos = zeros(length(t_pos),1);
for k = 1:length(t_pos)
    loc = find(t_c}>= t_pos(k))
    x_pos(k) = displ1_filt(loc(1));
end
x_neg = zeros(length(t_neg),1);
for k = 1:length(t_neg)
    loc = find(t_c >= t_neg(k));
    x_neg(k) = displ1_filt(loc(1));
end
%Curve Fitting:
t0 = t_pos(1);
t_pos = t_pos - t0; t_neg = t_neg - t0;
omega d = 2*pi*fd;
a0b0_pos = [x_pos(1)/scale -zeta_0*omega_d];
a0b0 neg = [x neg(1)/scale -zeta 0*omega d];
opts_pos = fitoptions('exp1','start',a0b\overline{0}_pos);
ab_pos = fit(t_pos,x_pos/scale,'expl',opts_pos);
opts_neg = fitoptions('exp1','start',a0b0_neg);
ab_neg = fit(t_neg,x_neg/scale,'expl',opts_neg);
```



```
disp('Negative Curve:'),disp(ab_neg)
%Fitted Curves:
expt = linspace(t c(1)-t0,t c(end)-t0,300);
exp_pos = scale*ab_pos.a*exp}(ab_pos.b*expt)
exp_neg = scale*ab_neg.a*exp(ab_neg.b*expt);
%Plot Fitted Curves:
figure(filtfig)
hold on
plot(t_pos+t0,x_pos,'ko')
plot(expt+t0,exp_pos,'--','Color',[0.5 0 0.8])
plot(t neg+t0,x neg,'ko')
plot(expt+t0,exp_neg,'--','Color',[0.5 0 0.8])
legend('Bearing \overline{1}','Bearing 2','Data points','Exponential curve fit')
hold off
%Average b-value:
b_ave = mean([ab_pos.b ab_neg.b]);
disp(['Average b-value: ','num2str(b_ave)])
%Calculate k1, k2, c1 & c2 from SDOF Approximation:
zeta = 1/sqrt(omega_d^2/b_ave^2 + 1);
omega_n = omega d/sqrt(1 - zeta^2);
k1 = m*omega_n^2/(1 + rk);
k2 = rk*k1;
ccrit = 2*m*omega n,
c1 = zeta*ccrit/2;
c2 = c1;
%Display Results:
disp(' '),disp(['zeta = ',num2str(zeta)])
disp(['k_1 = ',num2str(k1),' N/m'])
```

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems

```
disp(['k 2 = ',num2str(k2),' N/m'])
disp(['c_1 = ',num2str(c1),' N.s/m']
disp(['c 2 = ',num2str(c2),' N.s/m'])
```


## B. 2 Coupling Lateral Characterisation

## CouplingLatCharPolytec.m

```
Program that extracts, plots and (optionally) stores a velocity signal
recorded by a Polytec Scanning Laser Vibrometer for a specific coupling
characterisation experiment. The signal is a lateral step response
measurement.
Note: *The appropriate version of Polytec File Access (PFA) needs to be
    installed (depending on the version of PSV or Vibsoft used to
    acquire the measured data).
    *PFA function GetPointData.m is required in the working directory
        to run this program.
    *User-modifiable parameters/code indicated by :: in comments.
```

\% Author: Armand Krüger
Created: October 2012
\% Last revised: January 2013
clear all; clc; close all

\%Extract Time- \& Frequency Domain Response Data:
pnt = 1; \%::Measurement point
run $=1 ; \quad$ \%: :Measurement number
wl = 'hk'; \%::Static load weight(s) (\& measurement type)
filename $=[' c l '$ wl 'EFTpnt' num2str(pnt) '_' num2str(run) '.pvd'];
domainname = 'Time';
channelname = 'Vib';
signalname = 'Velocity';
displayname = 'Samples';
point = 0;
frame $=0$;
[t,V,usd] =
GetPointData(filename, domainname, channelname, signalname, displayname, point,frame);
t = t'; V = V';
domainname = 'FFT';
channelname = 'Vib';
signalname = 'Velocity';
displayname = 'Real \& Imag.';
[f,Fn,usd] =
GetPointData(filename, domainname, channelname,signalname, displayname, point,frame);
$\mathrm{f}=\mathrm{f}^{\prime} ; \mathrm{Fn}=\mathrm{Fn}{ }^{\prime}$;
\%Plot Response:
figure(1)
plot(t,V,'color', [0 0.7 0.9]),title('Vibration Response:')
xlabel('t [s]'),ylabel('Velocity [m/s]'),grid on
figure (2)
plot(f,abs(Fn),'color',[0 0.7 0.9]), title('DFT of Vibration Response:')
xlabel('f [Hz]'),ylabel('Velocity Magnitude [m/s]'), grid on

store = input('Do you want to store this data (y/n)? ','s')
if store == 'y'
storename = input(' Enter a filename (*.CSv): ','s');
storename = [storename '.csv'];
csvwrite (storename, [t V])
disp([' Data stored in ',storename])
else
disp(' No data stored')
end

## CouplingLatCharSP.m

```
Program that applies signal processing to a signal loaded or stored by
CouplingLatCharPolytec.m in order to determine the step response
frequency (or frequencies) and filter-out other frequencies.
Note: *Custom functions dft.m & expwindow.m are required in the working
    directory to run this program.
    *User-modifiable parameters/code indicated by : : in comments.
% Author: Armand Krüger
% Created: November 2012
% Last revised: January 2013
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DATA SOURCE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
source = 'CSV'; %::'CSV' or 'memory'
%To use the 'memory' option: first run CouplingLatCharPolytec.m
if strcmpi(source,'CSV')
    clear all; clc
    wl = 'hk'; %::Static load weight(s) (& measurement type)
    filename = input('Enter filename (*.CSV): ','s');
    filename = [filename '.csv'];
    data = csvread(filename);
    t = data(:,1); V = data(:,2);
end
%Determine Sampling Frequency:
dt = t(2) - t(1);
fs = round(1/dt); dt = 1/fs;
disp(['Sampling frequency = ',num2str(fs),' Hz'])
%Correctly Reconstruct Time Vector:
t = 0:dt:dt*(length(t)-1);
%Plot Raw Data:
close all
figure(1)
handle = plot(t,V); set(handle,'color',[0 0.7 0.9])
title('Coupling Lateral Step Response Signal:')
xlabel('t [s]'), ylabel('V [m/s]'),grid on
%%%%%%%%%%%%%%%%%%%%%%%% SIGNAL CROPPING & WINDOWING %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Crop Signal:
pretrig = 0.02148; %::Pre-trigger duration [s]
indx = find(t >= pretrig);
t = t(indx); V = V(indx);
%Apply Exponential Window to Signal:
l = length(t);
sf = 30; %::Percentage of t at which w = 0.01 [%]
w = expwindow(l,sf/100)';
V_w = w.*V;
```

\%Spectral Analysis:
$\mathrm{N}=2^{\wedge}$ (nextpow2(l));
$[f, \mathrm{Fn}]=\mathrm{dft}\left(\mathrm{V} \_\mathrm{w}, \mathrm{N}, \mathrm{fs}\right)$;
\%Plot Results:
figure (2)
handle $=$ plot(t, V); set(handle,'color', [0 0.7 0.9])
title('Cropped Coupling Lateral Step Response Signal:')
xlabel('t [s]'), ylabel('Velocity [m/s]'), grid on
figure (3)
subplot $(2,1,1)$
handle $=$ plot(t,V_w); set(handle,'color', [0 0.7 0.9])
title('Windowed Cōupling Lateral Step Response Signal:')
xlabel('t [s]'),ylabel('Velocity [m/s]'),grid on
subplot $(2,1,2)$
handle $=$ plot(f,abs(Fn)); set(handle,'color', 0 0.7 0.9])
title('DFT of Coupling Lateral Step Response Signal:')

```
xlabel('f [Hz]'),ylabel('Velocity Amplitude [m/s]'),grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTERING % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % 
%Band-Pass Filtering
if strcmp(wl,'hk')
    flc = 67; %::Lower cut-off frequency [Hz]
    fhc = 77; %::Upper cut-off frequency [Hz]
    ord = 150; %::Filter order
elseif strcmp(wl,'1k')
    flc = 64; %::Lower cut-off frequency [Hz]
    fhc = 74; %::Upper cut-off frequency [Hz]
    ord = 150; %::Filter order
else %wl = 1k4
    flc = 56; %::Lower cut-off frequency [Hz
    fhc = 65; %::Upper cut-off frequency [Hz]
    ord = 150; %::Filter order
end
d = fdesign.bandpass('n,fc1,fc2',ord,flc,fhc,fs)
h = design(d);
fvtool(h)
V filt = filtfilt(h.Numerator,1,V);
%Detrend Signal:
nbp = 10; %::Number of break points
dbp = round((l-1)/nbp);
n = 1;
for k = dbp:dbp:l-dbp
    bp(n) = indx(k);
    n = n + 1;
end
V_filt = detrend(V_filt,'linear',bp);
%Spectral Analysis:
N = 2^(nextpow2(l));
[f,Fn_filt] = dft(V_filt,N,fs);
%Plot Results:
filtfig = 5;
figure(filtfig)
subplot(2,1,1)
handle = plot(t,V filt); set(handle,'color',[0 0.7 0.9])
title('Filtered Coupling Lateral Step Response Signal:')
xlabel('t [s]'),ylabel('Velocity [m/s]'),grid on
subplot(2,1,2)
handle = plot(f,abs(Fn_filt)); set(handle,'color',[0 0.7 0.9])
title('DFT of Filtered Coupling Lateral Step Response Signal:')
xlabel('f [Hz]'),ylabel('Velocity Amplitude [m/s]'),grid on
```


## clhkFFTpnt1_2_kc.m

```
Program that determines approximate linear dynamic stiffness- & damping
```

Program that determines approximate linear dynamic stiffness- \& damping
coefficients from the filtered step response signals from file
coefficients from the filtered step response signals from file
clhkFFTpnt1_2.pvd (processed by CouplingLatCharSP.m).
clhkFFTpnt1_2.pvd (processed by CouplingLatCharSP.m).
Note: *First run CouplingLatCharSP.m.
Note: *First run CouplingLatCharSP.m.
*User-modifiable parameters/code indicated by :: in comments.
*User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
Author: Armand Krüger
Created: November 2012
Created: November 2012
Last revised: January 2013
Last revised: January 2013
%Data File- \& Signal Processing Information \& -Settings:
%Data File- \& Signal Processing Information \& -Settings:
File: clhkFFTpnt1 2.pvd/csv
File: clhkFFTpnt1 2.pvd/csv
CouplingLatCharSP.m settings:
CouplingLatCharSP.m settings:
pretrig = 0.02148
pretrig = 0.02148
flc = 67
flc = 67
fhc = 77
fhc = 77
ord = 150
ord = 150
n.bp = 10

```
        n.bp = 10
```

```
%Input Parameters:
fd = 72; %::Damped natural frequency [Hz]
m = 0.1115 + 0.0864 + 0.051 + 0.5; %::Total vibrating mass [kg]
zeta_0 = 0.05; %::Damping ratio starting guess value
scale = le-4; %::Data points scale value
% ::Positive- & negative data point t-values:
t_pos = [0.03906 0.05273 0.06738 0.08105 0.0957 0.1094 0.124 0.1367 0.1504]';
t_neg =[[l0.04585 0.060545 0.07422 0.08886 0.1025 0.1162 0.1299 0.1436 0.1572]';
%Curve Fit Points x-Values:
V_pos = zeros(length(t_pos),1);
for k = 1:length(t_pos)
    loc = find(t >= t_pos(k));
    V_pos(k) = V_filt(loc(1));
end
V neg = zeros(length(t neg),1);
for k = 1:length(t_neg)
    loc = find(t >= t neg(k));
    V_neg(k) = V_filt(loc(1));
end
%Curve Fitting:
t0 = t pos(1);
t_pos = t_pos - t0; t_neg = t_neg - t0;
omega_d = 2*pi*fd;
a0b0 pos = [V_pos(1)/scale -zeta_0*omega_d];
a0b0_neg = [V_neg(1)/scale -zeta_0*omega_d];
opts_pos = fitoptions('exp1','start',a0b0_pos);
ab_pos = fit(t pos,V pos/scale,'expl',opts_pos);
opts_neg = fitoptions('exp1','start',a0b0_neg);
ab_neg = fit(t_neg,V_neg/scale,'expl',opts_neg);
disp(' '),disp('Positive Curve:'),disp(ab_pos)
disp('Negative Curve:'),disp(ab_neg)
%Fitted Curves:
expt_pos = linspace(t(1)-t0,t(end)-t0,300);
exp_pos = scale*ab_pos.a*exp(ab_pos.b*expt_pos);
expt_neg = linspace(t(1)-t0,t(end)-t0,300);
exp_neg = scale*ab_neg.a*exp(ab_neg.b*expt_neg);
%Plot Fitted Curves:
figure(filtfig)
subplot(2,1,1)
hold on
plot(t_pos+t0,V_pos,'ko')
handle = plot(expt_pos+t0,exp_pos,'--'); set(handle,'color',[0.5 0 0.8])
plot(t_neg+t0,V ne\overline{g},'ko')
handle = plot(expt_neg+t0,exp_neg,'--'); set(handle,'color',[0.5 0 0.8])
legend('Filtered Sīgnal','Dat\overline{a points','Exponential curve fit')}
hold off
%Average b-value:
b ave = mean([ab pos.b ab neg.b]);
disp(['Average b-value: ',
%Calculate k & C from SDOF Approximation:
zeta = 1/sqrt(omega d^2/b ave^2 + 1);
omega_n = omega d/squrt(1 - zeta^2);
k = m*omega_n^2;
ccrit = 2*m*omega_n;
c = zeta*ccrit;
%Display Results:
disp(' '),disp(['zeta = ',num2str(zeta)])
disp(['k = ',num2str(k),' N/m'])
disp(['c = ',num2str(c),' N.s/m'])
```


## B. 3 Coupling Torsional Characterisation

## CouplingTorsChar.m

```
% Program that loads, processes & plots a displacement signal from a
% specific coupling bearing characterisation experiment. The signal is an
% angular step response measurement.
%
% Note: User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: August 2012
% Last revised: February 2013
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RAW DATA % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % 
%Load Raw Data:
run = 1; %::Measurement number
wtl = '2m d'; %::Static torsional load weight(s) (& measurement
type)
filename = ['ct' wtl ' ' num2str(run) '.txt'];
data = load(filename);
disp(['Data read from file: ',filename])
t = data(:,1); theta_raw = data(:,2);
%Determine Sampling Frequency:
dt = t(2) - t(1);
fs = round(1/dt); dt = 1/fs;
disp(['Sampling frequency = ',num2str(fs),' Hz'])
%Correctly Reconstruct Time Vector:
t = 0:dt:dt*(length(t)-1);
%Plot Raw Data:
figure(1)
handle = plot(t,theta_raw); set(handle,'color',[0 0.7 0.9])
title('Raw Coupling Torsional Step Response Signal:')
xlabel('t [s]'),ylabel('0 [mV]'),grid on
\circ}%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONVERSION % % % % % % % % % % % % % % 5% % % % % % % % % % % % % % % %
%Convert Data:
sens = -4.065; %::Potensiometer sensitivity [mV/deg]
offst = theta_raw(1)/sens;
theta = theta_raw/sens - offst;
%Plot Converted Data:
figure(2)
handle = plot(t,theta); set(handle,'color',[0 0.7 0.9])
title('Coupling Torsional Step Response Signal:')
xlabel('t [s]'), ylabel('0 [deg]'),grid on
CouplingTorsCharSP.m
```

```
Program that applies signal processing to signals loaded by
CouplingTorsChar.m in order to determine the step response frequency (or
frequencies) and filter-out other frequencies.
Note: *First run CouplingTorsChar.m.
    *Custom function expwindow.m is required in the working directory
    to run this program.
    *User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
```

```
% Created: August 2012
% Last revised: February 2013
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIGNAL RESAMPLING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
STF = 1; %::Sampling time scale factor
if STF ~= 1
    dt = STF*dt;
    fs = 1/dt;
    disp(['New sampling frequency = ',num2str(fs),' Hz'])
    t_old = t; theta_old = theta;
    t = t(1):dt:t(end);
    theta = interp1(t_old,theta_old,t)';
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIGNAL CROPPING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Crop Signal:
switch wtl
    case 'hk'
        tm = [0 %:Signal cropping times
            1];
    case '1k'
```



```
            2.59 4.673 4];
    case '1hk'
        tm=[[2.525 2.525 1.985 2.516 1.86 1.661 2.488
            3.7 3.7 2.758 3.5 2.86 2.45 3.652];
    case '2m a'
        tm =- [lll.821 1.961 1.982
            2.3 3 3];
    case '2m_b'
        tm = [lllllll
            3 3.8 2.98];
    case '2m_c'
```



```
            3.5 4.308 3.9];
    case '2m d'
        tm =-[[l2.44 1.961 1.982
            3.41 3 3];
    case '2me'
        tm = [2.602
            3.76];
    case '2m_f'
        tm =- [1.791
            2.8];
end
tstart = tm(1,run); tend = tm(2,run);
indx = find(t>=tstart & t<=tend)';
t_c = t(indx); theta_c = theta(indx);
%Plot Results:
figure(3)
handle = plot(t_c,theta_c); set(handle,'color',[0 0.7 0.9])
```



```
xlabel('t [s]'),ylabel('0 [deg]'),grid on
%%%%%%%%%%%%%%%%%%%%% SIGNAL DETRENDING & WINDOWING %%%%%%%%%%%%%%%%%%%%%%%%%
%Detrend Signal:
l = length(t c);
theta old = theta c;
theta_c = detrend(theta_c,'constant');
detr = theta_old - theta_c;
%Apply Exponential Window to Signal:
sf = 70; %::Percentage of t at which w = 0.01 [%]
w = expwindow(l,sf/100)';
theta_w = w.*theta_c;
%Plot Results:
figure(4)
handle = plot(t_c,theta_old,t_c,detr);
```

```
set(handle(1),'color',[0 0.7 0.9])
set(handle(2),'color',[0.5 0.8 0])
title('Cropped Coupling Torsional Step Response Signal with Trends:')
xlabel('t [s]'),ylabel('0 [deg]')
legend('Data Before Detrend','Constant Trend'),grid on
figure(5)
handle = plot(t_c,theta_c); set(handle,'color',[0 0.7 0.9])
title('Detrended Coupling Torsional Step Response Signal:')
xlabel('t [s]'),ylabel('0 [deg]'),grid on
figure(6)
handle = plot(t_c,theta_w); set(handle,'color',[0 0.7 0.9])
title('Detrended & Windowed Coupling Torsional Step Response Signal:')
xlabel('t [s]'),ylabel('0 [deg]'),grid on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTERING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Low-/Band-Pass Filtering:
ftype = 'low'; %::Filter type (low/band)
switch ftype
    case 'low'
        if strcmp(wtl,'1k')
            flc = 18;
            ord = 7;
        elseif strcmp(wtl,'1hk')
                flc = 17;
                ord = 7;
        elseif strcmp(wtl,'2m_a')
                flc = 80; %::Low-pass cut-off frequency [Hz]
                ord = 15; %::Filter order
            elseif strcmp(wtl,'2m_b')
                flc = 80; %::Low-pass cut-off frequency [Hz]
                ord = 15; %::Filter order
            elseif strcmp(wtl,'2m_c')
                flc = 80; %::Low-pass cut-off frequency [Hz]
                ord = 15; %::Filter order
            elseif strcmp(wtl,'2m_d')
                flc = 30; %::Low-pass cut-off frequency [Hz]
                ord = 10; %::Filter order
            else
                flc = 80;
                ord = 15;
                disp('Warning: default filtering values were used')
            end
            [b,a] = butter(ord,flc/(fs/2));
            h = dfilt.df2t(b,a);
            fvtool(h)
            theta_filt = filter(h,theta_w);
    case 'band'
        if strcmp(wtl,'1k')
            flc = 14;
            fhc = 18;
                ord = 50;
            elseif strcmp(wtl,'1hk')
                flc = 13;
                fhc = 17;
                ord = 50;
            elseif strcmp(wtl,'2m_a')
                flc = 10;
                fhc = 20;
                ord = 50; %::Filter order
            elseif strcmp(wtl,'2m_b')
                flc = 10;
                fhc = 20;
                ord = 50; %::Filter order
            elseif strcmp(wtl,'2m_c')
                flc = 10;
                fhc = 20;
                ord = 50; %::Filter order
            elseif strcmp(wtl,'2m_d')
```

```
    flc = 10;
    fhc = 20;
    ord = 50; %::Filter order
    else
        flc = 10;
        fhc = 20
        ord = 50;
        disp('Warning: default filtering values were used')
    end
    d = fdesign.bandpass('n,fc1,fc2',ord,flc,fhc,fs);
    h = design(d);
    fvtool(h)
    theta_filt = filtfilt(h.Numerator,1,theta_w);
end
%Detrend Signal:
theta_filt = detrend(theta_filt,'constant');
%Plot Results:
filtfig = 8;
figure(filtfig)
handle = plot(t_c,theta_filt); set(handle,'color',[0 0.7 0.9])
title('Detrended & Filtered Coupling Torsional Step Response Signal:')
xlabel('t [s]'),ylabel('0 [deg]'),grid on
ct2m_d_2_kc.m
```

```
Program that determines approximate linear dynamic stiffness- & damping
```

Program that determines approximate linear dynamic stiffness- \& damping
coefficients from the filtered step response signals from file
coefficients from the filtered step response signals from file
ct2m d 2.pvd (processed by CouplingTorsCharSP.m).
ct2m d 2.pvd (processed by CouplingTorsCharSP.m).
Note: *First run CouplingTorsCharSP.m.
Note: *First run CouplingTorsCharSP.m.
*User-modifiable parameters/code indicated by :: in comments.
*User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
Author: Armand Krüger
Created: November 2012
Created: November 2012
Last revised: February 2013
Last revised: February 2013
%Data File- \& Signal Processing Information \& -Settings:
%Data File- \& Signal Processing Information \& -Settings:
% File: ct2m_d_2.txt
% File: ct2m_d_2.txt
% CouplingTorsCharSP.m settings:
% CouplingTorsCharSP.m settings:
% STF = 1
% STF = 1
% tstart = 1.961
% tstart = 1.961
% tend = 3
% tend = 3
% flc = 30
% flc = 30
% ord = 10
% ord = 10
%Input Parameters:
%Input Parameters:
w1 = 1; %::Static load weight(s) 1 [kg]
w1 = 1; %::Static load weight(s) 1 [kg]
w2 = 0.8; %::Static load weight(s) 2 [kg]
w2 = 0.8; %::Static load weight(s) 2 [kg]
lm1 = 78; %::Static load weight 1 torque arm [mm]
lm1 = 78; %::Static load weight 1 torque arm [mm]
lm2 = 77; %::Static load weight 2 torque arm [mm]
lm2 = 77; %::Static load weight 2 torque arm [mm]
mbolt1 = 0.0113; %::Bolt 1 mass [kg]
mbolt1 = 0.0113; %::Bolt 1 mass [kg]
mbolt2 = 0.0102; %::Bolt 2 mass [kg]
mbolt2 = 0.0102; %::Bolt 2 mass [kg]
lboltG1 = 51.5; %::Bolt 1 CG distance [mm]
lboltG1 = 51.5; %::Bolt 1 CG distance [mm]
lboltG2 = 52.5; %::Bolt 2 CG distance [mm]
lboltG2 = 52.5; %::Bolt 2 CG distance [mm]
Ic = 7.0896; %::Coupling torsional inertia [kg.mm^2]
Ic = 7.0896; %::Coupling torsional inertia [kg.mm^2]
Inut = 27.135; %::Nut torsional inertia [kg.mm^2]
Inut = 27.135; %::Nut torsional inertia [kg.mm^2]
% ::Positive- \& negative data point t-values:
% ::Positive- \& negative data point t-values:
t1_2 = [2.019 2.068];
t1_2 = [2.019 2.068];
thèta1_2 = [-0.2393 0.063];
thèta1_2 = [-0.2393 0.063];
%Inertia Calculations:
%Inertia Calculations:
m1 = w1 + 0.051 + 0.0392;
m1 = w1 + 0.051 + 0.0392;
m2 = w2 + 0.01;
m2 = w2 + 0.01;
Iw = m1*lm1^2 + m2*lm2^2 ... %Torsional inertia of bolts \& weights
Iw = m1*lm1^2 + m2*lm2^2 ... %Torsional inertia of bolts \& weights
+ mbolt1*lboltG1^2 + mbolt2*lboltG2^2;

```
    + mbolt1*lboltG1^2 + mbolt2*lboltG2^2;
```

```
I = Ic + Inut + Iw; %Total torsional inertia about pnt O
I = I/1000^2;
%Logarithmic Decrement:
delta = 2*log(abs(theta1 2(1)/theta1 2(2)));
zeta = delta/sqrt((2*pi)^2
fd = 1/(2*(t1 2(2) - t1 2(1)));
wd = 2*pi*fd;
wn = wd/sqrt(1 - zeta^2);
k = I*Wn^2;
c0 = 2*I*Wn;
c = zeta*c0;
%Curve Fitting:
t_fit = linspace(2.003,t_c(end),1000);
phase = -atan(-zeta/sqrt(1-zeta^2));
theta0 = thetal_2(1)/cos(-phase);
theta_fit = theta0*exp(-zeta*wn*(t_fit - t1_2(1))).*cos(wd*(t_fit - t1_2(1)) -
phase);
%Plot Fitted Curve:
figure(filtfig)
hold on
handle = plot(t1_2,theta1_2,'ko',t_fit,theta_fit,'--');
set(handle(2),'color',[0.5 0 0.8])
hold off
legend('Filtered Signal','Data Points','SDOF Model Fit')
%Display Results:
disp(' '),disp(['zeta = ',num2str(zeta)])
disp(['f_d = ',num2str(fd),' Hz'])
disp(['k = ',num2str(k),' N.m/rad'])
disp(['c = ',num2str(c),' N.m.s/rad'])
```


## B. 4 Rotor System Experimental Analysis

Runout.m

```
Program that loads and plots rotor slow-roll runout signals (from
% horizontal & vertical proximity probes) and tachometer signals. The
% runout signals are converted from electrical to displacement signals
% using the sensitivity of the proximity probes.
% Note: *The lateral runout measurements are assumed to be in the yz-plane.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: March 2013
% Last revised: June 2014
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% : : CHANELS: : % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%RUNOUT & TACHOMETER DATA FILE:
% Double (y & z) Runout Channel Location(s):
ROLoc = {'Bush Bearing'
    'Journal Bearing'};
% Single Runout Channel Location(s) & Direction(s) [optional: leave blank if none]:
ROLoc1Ch = {}; %Location(s)
RODir1Ch = {}; %Direction(s)
% Tachometer Channel Location(s):
TachoLoc = {'Driving End'
    'Non-Driving End'};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LOAD SIGNALS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET FILENAME:
flag = 1;
while flag
    filename = input('Enter runout & tachometer data filename (*.txt): ','s');
    filename = [filename '.txt'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or '''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data loading cancelled')
            return
        end
    else
        flag = 0;
    end
end
disp(' ')
t start = input('Enter starting time of signals to be loaded: ');
t_end = input('Enter ending time of signals to be loaded: ');
%LOAD RUNOUT & TACHOMETER DATA:
disp(' '),disp('Loading runout & tachometer data...')
if fscanf(fid,'%c',[1,1]) == '"'
    skipline1 = 1;
else
    skipline1 = 0;
end
fclose(fid);
if skipline1
    signals = importdata(filename);
    signals = signals.data;
else
    signals = load(filename);
end
```

```
t = signals(:,1); l = length(t);
gyRO = zeros(l,length(ROLoc)); gzRO = gyRO;
for k = 1:length(ROLoc)*2
    if mod(k,2) ~= 0
        gyRO(:,k/2+0.5) = signals(:,k+1)/1000; %convert from mV to V
    else
        gzRO(:,k/2) = signals(:,k+1)/1000; %convert from mV to V
    end
end
if not(isempty(ROLoc1Ch))
    gRO = zeros(l,length(ROLoc1Ch));
    ROEnd = 1 + length(ROLOc)*2;
    for k = 1:length(ROLoc1Ch)
        gRO(:,k) = signals(:,k+ROEnd)/1000; %convert from mV to V
    end
end
Ta = zeros(l,length(TachoLoc));
ROEnd = 1 + length(ROLOc)*2 + length(ROLoc1Ch);
for k = 1:length(TachoLoc)
    Ta(:,k) = signals(:,k+ROEnd)/1000; %convert from mV to V
end
disp('Done loading data'), clear signals
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROCESS SIGNALS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DETERMINE SAMPLING FREQUENCY & CORRECTLY RECONSTRUCT TIME VECTOR:
% Sampling Frequency:
dt = t(2) - t(1);
fs = round(1/dt); dt = 1/fs;
disp(' ')
disp(['Slow roll data sampling frequency: ',num2str(fs),' Hz'])
% Correctly Reconstruct Time Vector (to eliminate rounding errors):
t = 0:dt:dt*(length(t)-1); t = t';
OAPPLY TIME LIMITS:
index = find(t >= t_start & t <= t_end);
t = t(index);
gyRO = gyRO(index,:); gzRO = gzRO(index,:);
if not(isempty(ROLoc1Ch))
    gRO = gRO(index,:);
end
Ta = Ta(index,:);
%CONVERT SIGNALS:
sens = -8; %::Eddy current probe sensitivity [V/mm]
gyRO = gyRO/sens; gzRO = gzRO/sens;
if not(isempty(ROLoc1Ch))
    gRO = gRO/sens;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT SIGNALS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tach = 2; %::Tachometer channel to use in plots
fig = 0;
nRO = length(ROLoc);
% Double Runout Signals:
for k = 1:nRO
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = plot(t,gzRO(:,k));
    title([ROLoc{k},' Horizontal Probe Runout Gap Signal:'])
    xlabel('t [s]'),ylabel('g_{zRO} [mm]'), set(handle,'color', [0 0.7 0.9])
    A = max(gzRO(:,k)) - min(gzRO(:,k));
    if A > 0
        axis([t(1) t(end) min(gzRO(:,k))-0.05*A max(gzRO(:,k)) +0.05*A])
    end
    subplot(2,1,2),handle = plot(t,Ta(:,tach));
    title([TachoLoc{tach} ' Tachometer Signal:'])
```

```
    xlabel('t [s]'),ylabel('Tachometer Signal [V]')
    set(handle,'color',[0 0.7 0.9])
    A = max(Ta(:,tach)) - min(Ta(:,tach));
    if A > 0
    axis([t(1) t(end) min(Ta(:,tach))-0.05*A max(Ta(:,tach))+0.05*A])
    end
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = plot(t,gyRO(:,k));
    title([ROLoc{k},' Vertical Probe Runout Gap Signal:'])
    xlabel('t [s]'),ylabel('g_{yRO} [mm]'), set(handle,'color',[0 0.7 0.9])
    A = max(gyRO(:,k)) - min(gyRO(:,k));
    if A > 0
    axis([t(1) t(end) min(gyRO(:,k))-0.05*A max(gyRO(:,k))+0.05*A])
    end
    subplot(2,1,2),handle = plot(t,Ta(:,tach));
    title([TachoLoc{tach} ' Tachometer Signal:'])
    xlabel('t [s]'),ylabel('Tachometer Signal [V]')
    set(handle,'color', [0 0.7 0.9])
    A = max(Ta(:,tach)) - min(Ta(:,tach));
    if A > 0
        axis([t(1) t(end) min(Ta(:,tach))-0.05*A max(Ta(:,tach)) +0.05*A])
    end
end
% Single Runout Signals (if required):
if not(isempty(ROLoc1Ch))
    nRO1Ch = length(ROLoc1Ch);
    for k = 1:nRO1Ch
        fig = fig + 1; figure(fig)
        subplot(2,1,1),handle = plot(t,gRO(:,k));
        title([ROLoc1Ch{k} ' Probe Gap Signal:'])
        xlabel('t [s]'),ylabel(['g_{',RODir1Ch{k},'RO} [mm]'])
        set(handle,'color',[0 0.7 0.9])
        A = max(gRO(:,k)) - min(gRO(:,k));
        if A > 0
                axis([t(1) t(end) min(gRO(:,k)) -0.05*A max(gRO(:,k))+0.05*A])
    end
    subplot(2,1,2),handle = plot(t,Ta(:,tach));
    title([TachoLoc{tach} ' Tachometer Signal:'])
    xlabel('t [s]'),ylabel('Tachometer Signal [V]')
    set(handle,'color',[0 0.7 0.9])
    A = max(Ta(:,tach)) - min(Ta(:,tach));
    if A > 0
                axis([t(1) t(end) min(Ta(:,tach))-0.05*A max(Ta(:,tach))+0.05*A])
            end
    end
end
% Tachometer Signals:
for k = 1:length(TachoLoc)
    fig = fig + 1; figure(fig)
    handle = plot(t,Ta(:,k)); set(handle,'color',[0 0.7 0.9])
    title([TachoLoc{k} ' Tachometer Signal:'])
    xlabel('t [s]'),ylabel('Tachometer Signal [V]')
end
```


## RunoutSP.m

```
A program that determines and plots the rotor slow-roll speed (as a
function of time) from tachometer signal(s), then calculates and saves
(optionally) the synchronous averages of the loaded runout signals
(loaded by Runout.m).
Note: *First run Runout.m.
    *Custom functions TachoSpinSpeed.m and SyncAve.m are required in
    the working directory to run this program.
    *User-modifiable parameters/code indicated by :: in comments.
```

Author: Armand Krüger

```
% Created: March 2013
% Last revised: June 2014
%%%%%%%%%%%%%%%%%% CHECK IF REQUIRED DATA IS AVAILABLE %%%%%%%%%%%%%%%%%%%
if not(exist('t','var') && exist('gyRO','var') && exist('gzRO','var'))
    disp('Error: Required data not detected in memory')
    disp(' First run Runout.m')
    disp(' Program terminated')
    return
end
if exist('Ta_detr','var')
    disp(' ')
    disp('Error: First re-run Runout.m before re-running RunoutSP.m')
    disp(' Program terminated')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SLOW ROLL SPEED %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PROCESS TACHOMETER SIGNALS (optional):
% Detrend Tachometer Signal(s):
Ta_detr = 1; %::Tachometer signals to detrend [vector if > 1]
for k = Ta_detr
    Ta(:,k) = detrend(Ta(:,k));
end
% Plot Detrended Tachometer Signal(s) (if any):
for k = Ta detr
    fig = \overline{fig + 1; figure(fig)}
    handle = plot(t,Ta(:,k)); set(handle,'color',[0 0.7 0.9])
    title(['Detrended ',TachoLoc{k},' Signal:'])
    xlabel('t [s]'),ylabel('Tachometer Signal [V]')
end
%DETERMINE & PLOT SPIN SPEED FROM TACHOMETER SIGNALS:
% Input Parameters (columns correspond with TachoLoc rows):
triglvl = [0.2 3]; %::Tachometer triger level(s) [V]
ppr = [12 1]; %::Tachometer pulses per revolution / no. of teeth
% Determine Slow Roll Spin Speed:
disp(' '),lseg = zeros(length(TachoLoc),1);
for k = 1:length(TachoLoc)
    [tseg_vec,omega_vec] = TachoSpinSpeed(Ta(:,k),t,ppr(k),triglvl(k));
    lseg(\overline{k}) = length}(tseg_vec)
    tseg(1:lseg(k),k) = tseg_vec; omega(1:lseg(k) -1,k) = omega_vec;
    N_ave = sum(omega_vec)/length(omega_vec)*60/2/pi;
    disp(['Average spin speed at ',TachoLoc{k},' Tachometer = ',num2str(N_ave),...
        (rmp'])
end
% Plot Spin Speed Results:
omegacolours = [0 0.7 0.9; 0.5 0 0.8; 0.5 0.8 0.2];
lgnd = cell(length(TachoLoc),1);
fig = fig + 1; figure(fig)
for k = 1:length(TachoLoc)
    hold all
    handle = plot(tseg(1:lseg(k) -1,k),omega(1:lseg(k)-1,k)*60/2/pi);
    if length(TachoLoc) <= 2
            set(handle,'color',omegacolours(k,:))
    end
    hold off
    TachoLoc_str = TachoLoc{k};
    lgnd{k} = [TachoLoc_str(1) lower(TachoLoc_str(2:end))];
end
title('Tachometer Spin Speeds:')
xlabel('t [s]'),ylabel('N [rpm]'),grid on
legend(lgnd)
```


\%DETERMINE RUNOUT SIGNAL SYNCHRONOUS AVERAGES:
\% Determine Synchronous Averages of Runout Signals for Shortest (Fastest) Segment:

```
Ta_sync = 2; %::Tachometer signal to use for synchronous averaging...
    (required: ppr = 1)
for k = 1:nRO
    [tsync,gyROsync(:,k),segs,gyROsegs(:,:,k),nLS_loc] = ..
        SyncAve(t,gyRO(:,k),tseg(1:lseg(Ta_sync),Ta_sync));
    [~,gzROsync(:,k),~,gzROsegs(:,:,k),~] =...
        SyncAve(t,gzRO(:,k),tseg(1:lseg(Ta_sync),Ta_sync));
end
if not(isempty(ROLoc1Ch))
    for k = 1:nRO1Ch
                [~,gRO1Chsync(:,k),~,gRO1Chsegs(:, :,k),~] = ...
        SyncAve(t,gRO(:,k),tseg(1:lseg(Ta_sync),Ta_sync));
    end
end
% Construct Angular Position Vector for Shortest (Fastest) Segment:
dtheta = omega(nLS_loc,Ta_sync)*dt;
theta = 0:dtheta:dtheta*(\overline{length(tsync)-1); theta = theta';}
%PLOT SYNCHRONOUS AVERAGING RESULTS:
% Initialize Legend:
nlgndmax = 4; %::Maximum number of segments to include in legends
if segs > nlgndmax
    lgnd = cell(nlgndmax+2,1);
else
    lgnd = cell(segs+1,1);
end
lgnd{1} = 'Synchronous Ave.';
% Specify Line Colours:
cmapsize = segs+200; %::Jet colormap size (refer to 'Colormap' in Matlab help)
crange = 0.5; %::fraction of colormap used (range)
coffst = 0.3; %::fraction of colormap offset (crange + coffst <= 1)
temp = 1; %::1 = warm; 0 = cold
darkness = 1; %::smaller = darker (must be < 1)
cmap = darkness*colormap(jet(cmapsize+1));
if temp
    cmap = flipud(cmap);
end
if segs < 2
    ndiv = 2;
else
    ndiv = segs;
end
kfact = floor(crange*cmapsize/(ndiv-1));
koffst = floor(coffst*cmapsize);
% Plot Results:
for k = 1:nRO
    % Runout Signal Segments & -Synchronous Averages:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),plot(theta*180/pi,gyROsync(:,k),'k','linewidth',1.5)
    subplot(2,1,2),plot(theta*180/pi,gzROsync(:,k),'k','linewidth',1.5)
    klgnd = 2;
    for s = 1:segs
        kcmap = (s-1)*kfact + koffst + 1;
        subplot(2,1,1)
            hold all
            handle = plot(theta*180/pi,squeeze(gyROsegs(:,s,k)));
            set(handle,'Color',cmap(kcmap,:))
            hold off
        subplot(2,1,2)
            hold all
            handle = plot(theta*180/pi,squeeze(gzROsegs(:,s,k)));
            set(handle,'Color', cmap (kcmap,:))
            hold off
        if klgnd <= nlgndmax + 1
            lgnd{klgnd} = ['Segment ' num2str(s)];
            klgnd = klgnd + 1;
```

```
    elseif klgnd == nlgndmax + 2
        lgnd{klgnd} = 'etc.';
    end
end
subplot(2,1,1)
    hold on
    plot(theta*180/pi,gyROsync(:,k),'k','linewidth',1.5)
    hold off
    title([ROLoc{k},' Vertical Probe Runout Signal Segments:'])
    xlabel('0 [deg]'),ylabel('g {yRO} [mm]')
    legend(lgnd),grid on
subplot(2,1,2)
    hold on
    plot(theta*180/pi,gzROsync(:,k),'k','linewidth',1.5)
    hold off
    title([ROLoc{k},' Horizontal Probe Runout Signal Segments:'])
    xlabel('0 [deg]'),ylabel('g_{zRO} [mm]')
    legend(lgnd),grid on
    % Runout Signal Synchronous Averages Only:
    fig = fig + 1; figure(fig)
    subplot(2,1,1)
    handle = plot(theta*180/pi,gyROsync(:,k));
    set(handle,'color', [0 0.7 0.9])
    title([ROLoc{k},' Vertical Probe Runout Signal Synchronous Average:'])
    xlabel('0 [deg]'),ylabel('g_{yROsync} [mm]'),grid on
    subplot (2,1,2)
    handle = plot(theta*180/pi,gzROsync(:,k));
    set(handle,'color',[0 0.7 0.9])
    title([ROLoc{k},' Horizontal Probe Runout Signal Synchronous Average:'])
    xlabel('0 [deg]'),ylabel('g_{zROsync} [mm]'),grid on
    % Phase Correlation between Horizontal & Vertical Runout Signals:
    fig = fig + 1; figure(fig)
    thetapos = find(theta >= pi/2);
    gyROsync90 = [gyROsync(thetapos,k);gyROsync(1:thetapos(1)-1,k)];
    gyROsync90 = detrend(gyROsync90,'constant');
    gzROsync90 = detrend(gzROsync(:,k),'constant');
    handle = plot(theta*180/pi,gyROsync90,theta*180/pi,gzROsync90);
    set(handle(1),'color',[0 0.7 0.9])
    set(handle(2),'color',[0.5 0.8 0.2])
    title([ROLoc{k},' Horizontal & Vertical Probes Runout Phase Correlation:'])
    xlabel('0 [deg]'),ylabel('Runout Signal [mm]')
    legend('g_{yROsync}(0 + 90^{o}) - mean(g_{yROsync})',...
    'g_{zROsync}(0) - mean(g_{zROsync})')
end
if not(isempty(ROLoc1Ch))
    for k = 1:nRO1Ch
    % Runout Signal Segments & -Synchronous Averages:
    fig = fig + 1; figure(fig)
    plot(theta*180/pi,gRO1Chsync(:,k),'k','linewidth',1.5)
    for s = 1:segs
            kcmap = (s-1)*kfact + koffst + 1;
            hold all
            plot(theta*180/pi,squeeze(gRO1Chsegs(:,s,k)),'Color',cmap(kcmap,:));
        end
    plot(theta*180/pi,gRO1Chsync(:,k),'k','linewidth',1.5)
    hold off
    title([ROLoc1Ch{k},' Probe Runout Signal Segments:'])
    xlabel('0 [deg]'),ylabel(['g {',RODir1Ch{k},'RO} [mm]'])
    legend(lgnd),grid on
    % Runout Signal Synchronous Averages Only:
    fig = fig + 1; figure(fig)
    plot(theta*180/pi,gRO1Chsync(:,k),'color',[0 0.7 0.9]);
    title([ROLoc1Ch{k},' Probe Runout Signal Synchronous Average:'])
    xlabel('0 [deg]'),ylabel(['g_{',RODirlCh{k},'ROsync} [mm]'])
    grid on
    end
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%% SHAFT/JOURNAL RUNOUT % % % % % % % % % % % % % % % % % % % % % % % % % % %
%INPUT PARAMETERS (columns correspond with ROLoc rows):
r = [9.5/2 24.5/2]; %::Shaft- or journal radii [mm]
rscale = [0.1 0.1]; %::Runout scale factors
%DETERMINE & PLOT APPROX. SHAFT/JOURNAL RUNOUT RADII & CENTRELINE DISPLACEMENT ...
%...RUNOUT ERRORS (DUE TO RUNOUT):
vRO = zeros(length(theta), nRO); wRO = vRO;
ryRO = vRO; rzRO = wRO;
for k = 1:nRO
    vRO = -detrend(gyROsync(:,k),'constant');
    wRO = -detrend(gzROsync(:,k),'constant');
    ryRO = vRO/max (vRO)*r(k)*rscale(k) +r(k);
    rzRO = wRO/max(wRO)*r(k)*rscale(k) + r(k);
    r_vec = r(k)*ones(length(theta),1);
    vROlines = zeros(length(vRO)*2,1); wROlines = vROlines;
    for k2 = 1:length(vRO)
        vROlines(2*k2) = vRO(k2); wROlines(2*k2) = wRO(k2);
    end
    % Runout Radii:
    fig = fig +1; figure(fig)
    handle = polar(theta,ryRO);
    set(handle,'color',[0 0.7 0.9])
    hold all
    handle = polar(theta,r_vec)
    set(handle,'color',0.6`[[1 1 1])
    hold off
    title([ROLoc{k},' Vertical Probe Runout Radius:'])
    xlabel('0 [deg]'),ylabel('r_{yRO} [mm]'),legend('Runout',...
            'Ideal shaft/journal')
    fig = fig +1; figure(fig)
    handle = polar(theta,rzRO);
    set(handle,'color',[0 0.7 0.9])
    hold all
    handle = polar(theta,r_vec);
    set(handle,'color',0.6``[1 1 1])
    hold off
    title([ROLoc{k},' Horizontal Probe Runout Radius:'])
    xlabel('0 [deg]'),ylabel('r_{zRO} [mm]'),legend('Runout',...
            'Ideal shaft/journal')
    % Centreline Displacement Errors Due to Runout:
    fig = fig + 1; figure(fig)
    handle = plot(wRO,vRO,wROlines,vROlines); axis equal
    set(handle,'color', [0 0.7 0.9])
    title(['Centreline Displacement Orbit Error due to Runout at...
            the ',ROLoc{k},':'])
    xlabel('w_{RO} [mm]'),ylabel('v_{RO} [mm]'),grid on
end
%%%%%%%%%%%%%%%%%%%%%%%%%%% SAVE RUNOUT SIGNALS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' '),store = lower(input('Save runout signals (y/n)? ','s'));
if store == 'y'
    filename = input(' Enter a filename (*.CSV): ','s');
    filename = [filename,'.csv'];
    signals = theta;
    for k = 1:nRO
        signals(:,2*k:2*k+1) = [gyROsync(:,k) gzROsync(:,k)];
    end
    if not(isempty(ROLoc1Ch))
            for k = 1:nRO1Ch
                signals(:,k+2*nRO+1) = gRO1Chsync(:,k);
            end
    end
    csvwrite(filename,signals)
```

```
    disp([' Runout signals saved as ',filename])
else
    disp(' No data saved')
end
```


## RotorVib.m

```
A program that loads and plots rotor vibration signals (from horizontal &
vertical proximity probes), tachometer signals and thermocouple signals.
The vibration signals are converted from electrical to displacement
signals using the sensitivity of the proximity probes.
% Note: *The lateral vibrations are assumed to be in the yz-plane.
    *User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
% Created: March 2013
% Last revised: June 2014
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% : CHANELS: : % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%VIBRATION & TACHOMETER DATA FILE:
% Double (y & z) Vibration Channel Location(s):
VibLoc = {'Bush Bearing'
    'Journal Bearing'};
% Single Vibration Channel Location(s) & Direction(s) [optional: leave blank if
none]:
VibLoclCh = {}; %Location(s)
VibDir1Ch = {}; %Direction(s)
% Tachometer Channel Location(s):
TachoLoc = {'Driving End'
    'Non-Driving End'};
%THERMOCOUPLE DATA FILE (optional):
% Thermocouple Channel Location(s) [leave blank if none]:
ThermoLoc = {'Journal Bearing'};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET FILENAME(S):
flag = 1;
while flag
    filenameVib = input('Enter vibration & tachometer data filename...
        (*.txt): ','s');
    filenameVib = [filenameVib '.txt'];
    fid = fopen(filenameVib,'r');
    if fid == -1
        disp([' Error: file ''',filenameVib,''' not found'])
        disp(' Enter filename without extension or ''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data loading cancelled')
            return
        end
    else
        flag = 0;
    end
end
if not(isempty(ThermoLoc))
    flag = 1;
    while flag
        filenameTh = input('Enter thermometer data filename (*.txt): ','s');
        filenameTh = [filenameTh '.txt'];
        fid2 = fopen(filenameTh,'r');
        if fid2 == -1
                disp([' Error: file ''',filenameTh,''' not found'])
                disp(' Enter filename without extension or ''''')
```

```
            proc = lower(input(' Proceed (y/n)? ','s'));
            if proc == 'n'
                disp(' Data loading cancelled')
                return
            end
        else
            flag = 0;
        end
    end
end
disp(' ')
t_start = input('Enter starting time of signals to be loaded: ');
t_end = input('Enter ending time of signals to be loaded: ');
%LOAD VIBRATION & TACHOMETER DATA:
disp(' '),disp('Loading vibration & tachometer data...')
if fscanf(fid,'%c',[1,1]) == '"'
    skipline1 = 1;
else
    skipline1 = 0;
end
fclose(fid);
if skipline1
    signals = importdata(filenameVib);
    signals = signals.data;
else
    signals = load(filenameVib);
end
t = signals(:,1); l = length(t);
gy = zeros(l,length(VibLoc)); gz = gy;
for k = 1:length(VibLoc)*2
    if mod}(k,2) ~= 
            gy(:,k/2+0.5) = signals(:,k+1)/1000; %convert from mV to V
        else
            gz(:,k/2) = signals(:,k+1)/1000; %convert from mV to V
        end
end
if not(isempty(VibLoc1Ch))
    g = zeros(l,length(VibLoc1Ch));
    VibEnd = 1 + length(VibLoc)*2;
    for k = 1:length(VibLoc1Ch)
            g(:,k) = signals(:,k+VibEnd)/1000; %convert from mV to V
        end
end
Ta = zeros(l,length(TachoLoc));
VibEnd = 1 + length(VibLoc)*2 + length(VibLoc1Ch);
for k = 1:length(TachoLoc)
    Ta(:,k) = signals(:,k+VibEnd)/1000; %convert from mV to V
end
%LOAD THERMOCOUPLE DATA (if required):
if not(isempty(ThermoLoc))
    disp('Loading thermocouple data...')
    if fscanf(fid2,'%c',[1,1]) == '"'
    skipline1 = 1;
        else
            skipline1 = 0;
        end
        fclose(fid2);
        if skipline1
            signals = importdata(filenameTh);
            signals = signals.data;
        else
            signals = load(filenameTh);
        end
        tTh = signals(:,1);
```

```
    Th = zeros(length(tTh),length(ThermoLoc));
    for k = 1:length(ThermoLoc)
        Th(:,k) = signals(:,k+1);
    end
end
disp('Done loading data'), clear signals
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROCESS SIGNALS % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % 
%DETERMINE SAMPLING FREQUENCIES & CORRECTLY RECONSTRUCT TIME VECTORS:
% Sampling Frequencies:
disp(' ')
dt = t(2) - t(1);
fs = round(1/dt); dt = 1/fs;
disp(['Vibration & tachometer data sampling frequency: ',num2str(fs),' Hz'])
if not(isempty(ThermoLoc))
    dtTh = tTh(2) - tTh(1);
    fsTh = round(1/dtTh); dtTh = 1/fsTh;
    disp(['Thermocouple signal(s) sampling frequency: ',num2str(fsTh),' Hz'])
end
% Correctly Reconstruct Time Vectors (to eliminate rounding errors):
t = 0:dt:dt*(length(t)-1); t = t';
if not(isempty(ThermoLoc))
    tTh = 0:dtTh:dtTh*(length(tTh)-1); tTh = tTh';
end
%APPLY TIME LIMITS:
index = find(t >= t_start & t <= t_end);
t = t(index);
gy = gy(index,:); gz = gz(index,:);
if not(isempty(VibLoc1Ch))
    g = g(index,:);
end
Ta = Ta(index,:);
if not(isempty(ThermoLoc))
    index = find(tTh >= t_start & tTh <= t_end);
    tTh = tTh(index);
    Th = Th(index,:);
end
%CONVERT SIGNALS:
sens = -8; %::Eddy current probe sensitivity [V/mm]
gy = gy/sens; gz = gz/sens;
if not(isempty(VibLoc1Ch))
    g = g/sens;
end
Ta(:,1) = -Ta(:,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT SIGNALS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig = 0;
nVib = length(VibLoc);
% Double Vibration Signals:
for k = 1:nVib
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = plot(t,gy(:,k));
    title([VibLoc{k},' Vertical Probe Gap Signal:'])
    xlabel('t [s]'),ylabel('g_{y} [mm]'), set(handle,'color',[0 0.7 0.9])
    A = max(gy(:,k)) - min(gy(:,k));
    if A > 0
            axis([t(1) t(end) min(gy(:,k))-0.05*A max(gy(:,k))+0.05*A])
    end
    subplot(2,1,2),handle = plot(t,gz(:,k));
    title([VibLoc{k},' Horizontal Probe Gap Signal:'])
    xlabel('t [s]'),ylabel('g_{z} [mm]'),set(handle,'color',[0 0.7 0.9])
    A = max(gz(:,k)) - min(gz(:,k));
    if A > 0
            axis([t(1) t(end) min(gz(:,k)) -0.05*A max(gz(:,k))+0.05*A])
    end
end
```

```
% Single Vibration Signals (if required):
if not(isempty(VibLoc1Ch))
    nVib1Ch = length(VibLoc1Ch);
    for k = 1:nViblCh
        fig = fig + 1; figure(fig)
        handle = plot(t,g(:,k)); set(handle,'color',[0 0.7 0.9])
        title([VibLoc1Ch{k} ' Probe Gap Signal:'])
        xlabel('t [s]'),ylabel(['g_{',VibDir1Ch{k},'} [mm]'])
        A = max(g(:,k)) - min(g(:,k));
        if A > 0
            axis([t(1) t(end) min(g(:,k))-0.05*A max (g(:,k)) +0.05*A])
        end
    end
end
% Tachometer Signals:
for k = 1:length(TachoLoc)
    fig = fig + 1; figure(fig)
    handle = plot(t,Ta(:,k)); set(handle,'color',[0 0.7 0.9])
    title([TachoLoc{k} ' Tachometer Signal:'])
    xlabel('t [s]'),ylabel('Tachometer Signal [V]')
end
% Thermocouple Signals (if required):
if not(isempty(ThermoLoc))
    for k = 1:length(ThermoLoc)
        fig = fig + 1; figure(fig)
        handle = plot(tTh,Th(:,k)); set(handle,'color',[0 0.7 0.9])
        title([ThermoLoc{k} ' Thermometer Signal:']),grid on
        xlabel('t [s]'),ylabel('Temperature [^{o}C]')
    end
end
```


## RotorVibSP.m

```
A program that determines and plots the rotor spin speed (as a function
of time) from tacohmeter signal(s), then (optionally) performs runout
compensation on vibration signals (loaded by RotorVib.m) and plots the
resulting displacements relative to the (approximate) equilibrium
position of the shaft or relative to the sensor-/bearing housing. These
displacement signals may also be saved.
Note: *First run RotorVib.m.
    *The runout signals required for runout compensation (in *.csv
        format) are stored by RunoutSP.m.
    *Custom functions TachoSpinSpeed.m, RunoutComp1.m, RunoutComp2.m
        and rms.m are required in the working directory to run this
        program.
    *User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
% Created: March 2013
% Last revised: July 2014
%%%%%%%%%%%%%%%%%%% CHECK IF REQUIRED DATA IS AVAILABLE %%%%%%%%%%%%%%%%%%%%%%
if not(exist('t','var') && exist('gy','var') && exist('gz','var'))
    disp('Error: Required data not detected in memory')
    disp(' First run RotorVib.m')
    disp(' Program terminated')
    return
end
if exist('figSP','var')
    disp(' ')
    disp('Error: First re-run RotorVib.m before re-running RotorVibSP.m')
    disp(' Program terminated')
    return
end
```

```
%PROCESS TACHOMETER SIGNALS (optional):
% Detrend Tachometer Signal(s):
Ta detr = 1; %::Tachometer signals to detrend [vector if > 1]
for k = Ta_detr
    Ta(:,k) = detrend(Ta(:,k));
end
% Plot Detrended Tachometer Signal(s) (if any):
for k = Ta detr
    fig = \overline{fig + 1; figure(fig)}
    handle = plot(t,Ta(:,k)); set(handle,'color',[0 0.7 0.9])
    title(['Detrended ',TachoLoc{k},' Signal:'])
    xlabel('t [s]'),ylabel('Tachometer Signal [V]')
end
%DETERMINE & PLOT SPIN SPEED FROM TACHOMETER SIGNALS:
% Input Parameters (columns correspond with TachoLoc rows):
triglvl = [0.2 3]; %::Tachometer triger level(s) [V]
ppr = [12 1]; %::Tachometer pulses per revolution / no. of teeth
% Determine the Spin Speed as a Function of Time:
disp(' '),lseg = zeros(length(TachoLoc),1);
for k = 1:length(TachoLoc)
    [tseg_vec,omega_vec] = TachoSpinSpeed(Ta(:,k),t,ppr(k),triglvl(k));
    lseg(k) = length(tseg_vec);
    tseg(1:lseg(k),k) = tseg_vec; omega(1:lseg(k) - 1,k) = omega_vec;
    N ave(k) = mean(omega vec)*60/2/pi;
    disp(['Average spin speed at ',TachoLoc{k},' = ',num2str(N_ave(k)),' rmp'])
end
% Plot Spin Speed Results:
omegacolours = [0 0.7 0.9; 0.5 0 0.8; 0.5 0.8 0.2];
lgnd = cell(length(TachoLoc),1);
fig = fig + 1; figure(fig)
for k = 1:length(TachoLoc)
    hold all
    handle = plot(tseg(1:lseg(k) -1,k),omega(1:lseg(k) -1,k)*60/2/pi);
    if length(TachoLoc) <= 3
        set(handle,'color',omegacolours(k,:))
    end
    hold off
    TachoLoc_str = TachoLoc{k};
    lgnd{k} = [TachoLoc_str(1) lower(TachoLoc_str(2:end))];
end
title('Tachometer Spin Speeds:')
xlabel('t [s]'), ylabel('N [rpm]'),grid on
legend(lgnd)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ROTOR DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' '),ROcomp = lower(input('Perform runout compensation (y/n): ','s'));
if isempty(ROcomp)
    ROcomp = 'y';
end
%SENSOR-/BEARING HOUSING PARAMETERS (columns correspond with VibLoc rows):
cy = [0 0.2]; %::Housing clearance in y-direction [mm]
Cz = [0 0.2]; %::Housing clearance in z-direction [mm]
ghy = [0 0.28]; %::Sensor-housing gap in y-direction [mm]
ghz = [0 0.43]; %:Sensor-housing gap in z-direction [mm]
y = zeros(length(t),nVib); z = y;
if ROcomp ~= 'n'
%RUNOUT COMPENSATION:
    % Compensation Type & Tachometer Channel Selection:
    CompType = [1 2]; %::Compensation type:
                                    relative to: equilibrium-> 1, housing-> 2
                            % Note: columns correspond with VibLoc rows
        Ta_comp = 2; %::Tachometer signal to use for runout compensation...
                            (required: ppr = 1)
```

```
% Load Runout Signals from CSV Files:
flag = 1;
while flag
    filenameRO = input(' Enter runout signals filename (*.CSV): ', 'S');
    filenameRO = [filenameRO '.CSV'];
    fid = fopen(filenameRO,'r');
    if fid == -1
        disp([' Error: file ''',filenameRO,''' not found'])
        disp(' Enter filename without extension or '''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data loading cancelled')
            return
        end
    else
        flag = 0;
    end
end
fclose(fid);
disp(' '),disp('Loading runout data...')
data = csvread(filenameRO);
if isempty(VibLoc1Ch)
    nchannels = nVib*2;
else
    nchannels = nVib*2 + nVib1Ch;
end
if length(data(1,:)) ~= nchannels + 1
    disp(' ')
    disp('Error: The loaded runout data does not correspond with the...
        channels specified in RotorVib.m')
    disp('Program terminated')
    return
end
theta = data(:,1);
for k = 1:nVib*2
    if mod}(k,2) ~= 
        gyRO(:,k/2+0.5) = data(:,k+1);
    else
        gzRO(:,k/2) = data(:,k+1);
    end
end
if not(isempty(VibLoc1Ch))
    for k = 1:nVib1Ch
        gRO(:,k) = data(:,k+2*nVib+1);
    end
end
disp('Done loading data'), clear data
for k = 1:nVib
    fig = fig + 1; figure(fig)
    subplot(2,1,1)
    handle = plot(theta*180/pi,gyRO(:,k));
    set(handle,'color',[0 0.7 0.9])
    title([VibLoc{k},' Vertical Probe Runout Signal:'])
    xlabel('0 [deg]'),ylabel('g_{yRO} [mm]'),grid on
    subplot(2,1,2)
    handle = plot(theta*180/pi,gzRO(:,k));
    set(handle,'color', [0 0.7 0.9])
    title([VibLoc{k},' Horizontal Probe Runout Signal:'])
    xlabel('0 [deg]'),ylabel('g_{zRO} [mm]'),grid on
end
if not(isempty(VibLoc1Ch))
    for k = 1:nVib1Ch
        fig = fig + 1; figure(fig)
        plot(theta*180/pi,gRO(:,k),'color',[0 0.7 0.9]);
        title([VibLoclCh{k},' Probe Runout Signal:']),grid on
        xlabel('0 [deg]'),ylabel(['g_{',VibDir1Ch{k},'RO} [mm]'])
    end
end
```

```
    % Perform Runout Compensation:
    for k = 1:nVib
        if CompType(k) == 2
            y(:,k) = RunoutComp2(t,gy(:,k),theta,gyRO(:,k),...
                    tseg(1:lseg(Ta_comp),Ta_comp),cy(k),ghy(k));
                z(:,k) = RunoutComp2(t,gz(:,k),theta,gzRO(:,k),\ldots
                    tseg(1:lseg(Ta_comp),Ta_comp),cz(k),ghz(k));
        else
            y(:,k) = RunoutComp1(t,gy(:,k),theta,gyRO(:,k),...
                    tseg(1:lseg(Ta_comp),Ta_comp));
            z(:,k) = RunoutComp1(t,gz(:,k),theta,gzRO(:,k),\ldots
                tseg(1:lseg(Ta_comp),Ta_comp));
        end
    end
    if not(isempty(VibLoc1Ch))
        x = zeros(length(t),nVib1Ch);
        for k = 1:nVib1Ch
            x(:,k) = RunoutComp1(t,g(:,k),theta,gRO(:,k),...
                tseg(1:lseg(Ta_comp),Ta_comp));
        end
    end
else
%NO RUNOUT COMPENSATION:
    BearingType = [0 1]; %::Bearing type: bush-> 0, journal-> 1
                            % Note: columns correspond with VibLoc rows
    % Calculate Displacements Relative to Equlibrium Position or Housing Centre:
    for k = 1:nVib
        if BearingType(k)
            y(:,k) = cy(k) + ghy(k) - gy(:,k);
            z(:,k) = cz(k) + ghz(k) - gz(:,k);
        else
            y(:,k) = -detrend(gy(:,k),'constant');
            z(:,k) = -detrend(gz(:,k),'constant');
        end
    end
    if not(isempty(VibLoc1Ch))
        x = zeros(length(t),nVib1Ch);
        for k = 1:nViblCh
            x(:,k) = -detrend(g(:,k),'constant');
        end
    end
end
```



```
fltr = 0; %::0 = do not filter; 1 = filter
spin comp = 1; %::Spin speed component to filter to
if fltrr
    flc = N_ave(Ta_comp)*spin_comp/60*0.98;
    fhc = N_}\mp@subsup{N}{-}{-}\mathrm{ ave(Ta_comp)*spin_comp/60*1.02;
    ord = 300; %::Filter order
    d = fdesign.bandpass('n,fc1,fc2',ord,flc,fhc,fs);
    h = design(d);
    %fvtool(h)
    for k = 1:nVib
        y_det = detrend(y(:,k),'constant'); z_det = detrend(z(:,k),'constant');
        y_trend = y(:,k) - y_det; z_trend = z(:,k) - z_det;
        y(:,k) = filtfilt(h.Numerator,1,y_det) + y_trend;
        z(:,k) = filtfilt(h.Numerator,1,z_det) + z_trend;
    end
    if not(isempty(VibLoc1Ch))
        for k = 1:nVib1Ch
            x det = detrend(x(:,k),'constant'); x trend = x(:,k) - x det;
            x(:,k) = filtfilt(h.Numerator,1,x_det) + x_trend;
        end
    end
    disp(' ')
    t_filtstart = input('Enter starting time of filtered signals to plot: ');
    t_filtend = input('Enter ending time of filtered signals to plot: ');
```

```
    index = find(t >= t_filtstart & t <= t_filtend);
    t = t(index); y = y(index,:); z = z(index,:);
    if not(isempty(VibLoc1Ch))
        x = x(index,:);
    end
end
%%%%%%%%%%%%%%%%%%% DISPLACEMENT MAGNITUDES & RMS VALUES %%%%%%%%%%%%%%%%%%%
%CALCULATE DISPLACEMENT MAGNITUDES:
mag = sqrt(y.^2 + z.^2);
displ_max = zeros(nVib,1); t_max = displ_max; disp(' ')
for k = 1:nVib;
    [displ_max(k),mpos] = max(mag(:,k)); t_max(k) = t(mpos);
    disp(['Maximum displacement at ',lower(VibLoc{k}),...
            ' = ',num2str(displ_max(k)),...
            ' mm at t = ',num2str(t_max(k)),' s'])
end
%CALCULATE RMS VALUES:
y_rms = zeros(nVib,1); z_rms = y_rms; disp(' ')
for k = 1:nVib
    y_rms(k) = rms(y(:,k));
    z rms(k) = rms(z(:,k));
    loc = VibLoc{k};
    disp([loc(1),lower(loc(2:end)),...
            ' vertical displacement RMS value = ',num2str(y_rms(k)),'mm'])
    disp([loc(1),lower(loc(2:end)),...
            ' horizontal displacement RMS value = ',num2str(z_rms(k)),'mm'])
end
%%%%%%%%%%%%%%%%%%%%%%%%% PLOT & SAVE DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%%
%PLOT DISPLACEMENTS:
figSP = fig;
for k = 1:nVib
    % Vertical- & Horizontal Displacements:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = plot(t,y(:,k));
    title([VibLoc{k},' Vertical Displacement:'])
    xlabel('t [s]'), ylabel('y [mm]'), set(handle,'color',[0 0.7 0.9])
    A = max (y(:,k)) - min(y(:,k));
    if A > 0
            axis([t(1) t(end) min(y(:,k))-0.05*A max (y(:,k))+0.05*A])
    end
    subplot(2,1,2),handle = plot(t,z(:,k));
    title([VibLoc{k},' Horizontal Displacement:'])
    xlabel('t [s]'),ylabel('z [mm]'),set(handle,'color',[0 0.7 0.9])
    A = max(z(:,k)) - min(z(:,k));
    if A>0
            axis([t(1) t(end) min(z(:,k))-0.05*A max(z(:,k))+0.05*A])
    end
    % Displacement Orbit:
    fig = fig + 1; figure(fig)
    handle = plot(z(:,k),y(:,k));
    title([VibLoc{k},' Displacement Orbit:'])
    xlabel('z [mm]'),ylabel('y [mm]')
    axis equal,set(handle,'color', [0 0.7 0.9])
    % Displacement Magnitude:
    fig = fig + 1; figure(fig)
    handle = plot(t,mag(:,k));
    title([VibLoc{k},' Displacement Magnitude:'])
    xlabel('t [s]'),ylabel('r [mm]'), set(handle,'color',[0 0.7 0.9])
    A = max(mag(:,k)) - min(mag(:,k));
    if A > 0
        axis([t(1) t(end) min(mag(:,k)) -0.05*A max(mag(:,k)) +0.05*A])
    end
    % 2D Displacement:
    fig = fig + 1; figure(fig)
```

```
    t_scale = 5; %::Time scale factor (t : displacement range)
    handle = plot3(t,z(:,k),y(:,k));
    title([VibLoc{k},' 2D Displacement as a Function of Time:'])
    xlabel('t [s]'),ylabel('z [mm]'),zlabel('y [mm]')
    set(handle,'color',[0 0.7 0.9])
    A = max(max(y(:,k)) - min(y(:,k)),max(z(:,k)) - min(z(:,k)));
    scale = t_scale*A/(t(end)-t(1));
    set(gca,'D}ataAspectRatio',[1 scale scale]
end
if not(isempty(VibLoc1Ch))
    for k = 1:nVib1Ch
        fig = fig + 1; figure(fig)
        plot(t,x(:,k),'color',[0 0.7 0.9]);
        title([VibLoc1Ch{k},' Vertical Displacement:'])
        xlabel('t [s]'),ylabel([VibDir1Ch{k},' [mm]'])
        A = max(x(:,k)) - min(x(:,k));
        if A > 0
                axis([t(1) t(end) min(x(:,k)) -0.05*A max(x(:,k)) +0.05*A])
            end
    end
end
%SAVE DISPLACEMENTS:
disp(' '),store = lower(input('Save vibration signals (y/n)? ','s'));
if store == 'y'
    filename = input(' Enter a basic filename (*.CSV): ','s');
    if filename(end) ~= ' '
        filename = [filenāme ' '];
    end
    for k = 1:nVib
        suffix = VibLoc{k};
        if length(suffix) > 4
                filenamek = [filename,suffix(1:4),'.csv'];
            else
                filenamek = [filename,suffix,'.csv'];
            end
            csvwrite(filenamek,[t y(:,k) z(:,k)])
            if nVib > 1
                disp([' Data saved as ',filenamek,'...'])
            else
                disp([' Data saved as ',filenamek])
            end
    end
    if nVib > 1
            disp(' Done saving data')
    end
    if not(isempty(VibLoc1Ch))
        for k = 1:nVib1Ch
            suffix = VibLoc1Ch{k};
            if length(suffix) > 4
                    filenamek = [filename,suffix(1:4),'.csv'];
            else
                filenamek = [filename,suffix,'.cSv'];
            end
                csvwrite(filenamek, [t x(:,k)])
                if nVib > 1
                    disp([' Single Channel Data saved as ',filenamek,'...'])
                    else
                        disp([' Single Channel Data saved as ',filenamek])
            end
            end
            if nVib1Ch > 1
                        disp(' Done saving single channel data')
        end
    end
else
    disp(' No data saved')
end
```


## RotorVibClearance.m

```
A program that adds clearance lines to figures with vibration signals
processed by RotorVibSP.m.
Note: *First run RotorVib.m.
    *Lateral displacements are assumed to be in the yz-plane.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: March 2013 (based on RotorDisplClearance.m by Armand Krüger)
% Last revised: July 2014
%CHECK IF REQUIRED DATA IS AVAILABLE:
if not(exist('t','var') && exist('y','var') && exist('z','var') &&
exist('figSP','var'))
    disp('Error: Required data not detected in memory')
    disp(' First run RotorVibSP.m')
    disp(' Program terminated')
    return
end
%DETERMINE AVERAGE RADIAL CLEARANCE SIZE:
pos = find(CompType == 2);
if isempty(pos)
    disp('Error: Runout compensation method 2 was not performed on any signal')
    disp(' Program terminated')
    return
end
rc = (cy(pos) + cz(pos))/2;
%ADD CLEARANCE LINES TO FIGURES:
thetac = linspace(0,2*pi,100);
kc = 1;
prevfigs = 8 + length(ThermoLoc);
for k = pos
    if not(exist('figspec','var'))
        fgr = (k-1)*(fig-prevfigs)/nVib + prevfigs + 1;
    else
        fgr = (k-1)*(figspec-prevfigs)/nVib + prevfigs + 1;
    end
    % Vertical- & Horizontal Displacements:
    tc = [t(1) t(end)];
    xc = rc(kc)*[ll 1];
    figure(fgr),subplot(2,1,1)
    hold on
    plot(tc,xc,tc,-xc,'Color',0.6*[1 1 1])
    ymin = min(min(y(:,k)),-rc(kc));
    legend('Vertical Displacement','Radial Clearance Lines')
    ymax = max(max(y(:,k)),rc(kc));
    A = ymax - ymin; axis([t(1) t(end) ymin-0.05*A ymax+0.05*A])
    hold off
    subplot(2,1,2)
    hold on
    plot(tc,xc,tc,-xc,'Color',0.6*[[1 1 1])
    zmin = min(min(z(:,k)),-rc(kc)); zmax = max(max(z(:,k)),rc(kc));
    A = zmax - zmin; axis([t(1) t(end) zmin-0.05*A zmax+0.05*A])
    legend('Horizontal Displacement','Radial Clearance Lines')
    hold off
    % Displacement Orbit:
    figure(fgr+1)
    hold on
    zco = rc(kc)*sin(thetac); yco = rc(kc)*cos(thetac);
    lgnd = {'Displacement Orbit','Radial Clearance Circle'};
    if displ_max(k) > rc(kc)
        lwidth = 1.5;
```

```
    else
    lwidth = 0.5;
    end
    plot(zco,yco,'Color',0.6*[1 1 1],'LineWidth',lwidth)
    legend(lgnd)
    hold off
    % Displacement Magnitude:
    figure(fgr+2)
    hold on
    plot(tc,xc,'Color',0.6*[1 1 1])
    rmin = min(min(mag(:,k)),rc(kc)); rmax = max(max(mag(:,k)),rc(kc));
    A = rmax - rmin; axis([t(1) t(end) rmin-0.05*A rmax+0.05*A])
    legend('Displacement','Radial Clearance Line')
    hold off
    % 2D Displacement:
    figure(fgr+3)
    hold on
    posr = find(mag(:,k) >= rc(kc));
    if not(isempty(posr))
    tc1 = t(posr(1))-(t(posr(end))-t(posr(1)))*0.3;
    tc2 = t(posr(end)) +(t(posr(end)) -t(posr(1)))*0.3;
    else
    tc1 = t(1); tc2 = t(end);
    end
    tc_v = [tc1 tc2];
    zc_v = rc(kc)*sin(thetac); yc_v = rc(kc)*cos(thetac);
    yc_m = zeros(100,2);
    for kj = 1:2
    yc_m(:,kj) = yc_v';
    end
    lgnd = {'2D Displacement','Radial Clearance Cylinder'};
    surf(tc_v,zc_v,yc_m,'EdgeColor','none','FaceColor',0.9*[1 1 1])
    alpha(0.85),\overline{legen\overline{d}}(l\mathrm{ lgnd)}
    hold off
    kc = kc + 1;
end
```


## RotorVibSpectral.m

```
A program that performs a spectral analysis on vibration signals
processed by RotorVibSP.m and then plots the results.
Note: *First run RotorVibSP.m.
    *Custom functions dft.m & psde.m are required in the working
        directory to run this program.
    *User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
% Created: March 2013 (based on RotorDisplSpectral.m by Armand Krüger)
% Last revised: July 2014
%%%%%%%%%%%%%%%%%%%% CHECK IF REQUIRED DATA IS AVAILABLE %%%%%%%%%%%%%%%%%%%%
if not(exist('t','var') && exist('y','var') && exist('z','var')) &&
exist('figSP','var')
    disp('Error: Required data not detected in memory')
    disp(' First run RotorVibSP.m')
    disp(' Program terminated')
    return
end
%%%%%%%%%%%%%%%%%%%%%% RESAMPLE SIGNALS (if required) % % % % % % % % % % % % % % % % % % % % %
%SPECIFY SAMPLING TIME SCALE FACTOR:
STF = 5; %::Sampling time scale factor (default)
disp(' ')
STFinp = input(['Enter a sampling time scale factor (default=',...
    num2str(STF),'): ']);
if not(isempty(STFinp))
```

```
    STF = STFinp;
end
%RESAMPLE SIGNALS (if required):
if STF ~= 1
    dt_new = STF*dt; fs_new = 1/dt_new;
    disp([' New sampling frequency = ',num2str(fs_new),' Hz'])
    t_new = (t(1):dt_new:t(end))';
    y_new = interpl(t,y,t_new);
    z_new = interp1(t,z,t_new);
else
    fs new = fs;
    t_\overline{n}ew = t; y_new = y; z_new = z;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DETREND SIGNALS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fc = 5; %::Detrend lower cutoff frequency [Hz]
l = length(t new);
dbp = round(fs_new/fc); nbp = round((l - 1)/dbp - 1);
n = 1; indx = 1:l; bp = zeros(nbp,1);
for k = dbp:dbp:l-dbp
    bp(n) = indx(k+1);
    n = n + 1;
end
y_det = detrend(y_new,'linear',bp);
z_det = detrend(z_new,'linear',bp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SPECTRAL ANALYSIS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SPECIFY PARAMETERS FOR SPECTRAL ANALYSES & PLOTTING:
N = 2^(nextpow2(l));
windowl = 512; %::Window length for spectrogram (default)
disp(' ')
window_inp = input(['Enter a window length for the Spectrograms...
    (default=',num2str(windowl),'): ']);
if not(isempty(window_inp))
    windowl = window_inp;
end
noverlap = floor(windowl/10)*10;
nfft = windowl;
%CALCULATE & PLOT SPECTRAL CONTENT:
if not(exist('figspec','var'))
    figspec = fig;
else
    close(figspec+1:fig)
    fig = figspec;
end
for k = 1:nVib
    % Calculate DFTs & PSDs:
    [fy1,Fny] = dft(y_det(:,k),N,fs_new);
    [fz1,Fnz] = dft(z_det(:,k),N,fs_new);
    [Py,fy2] = psde(y_det(:,k),N/2,N/4,fs_new);
    [Pz,fz2] = psde(z_det(:,k),N/2,N/4,fs_new);
    % Vertical & Horizontal Displacement DFTs:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = semilogy(fy1,abs(Fny));
    title(['DFT of the ',VibLoc{k},' Vertical Displacement:'])
    xlabel('f [Hz]'),ylabel('|F_{ny}| [mm]'),grid on
    set(handle,'color',[0 0.7 0.9])
    subplot(2,1,2),handle = semilogy(fz1,abs(Fnz));
    title(['DFT of the ',VibLoc{k},' Horizontal Displacement:'])
    xlabel('f [Hz]'),ylabel('|F {nz}| [mm]'),grid on
    set(handle,'color',[0 0.7 0.9])
    % Vertical & Horizontal Displacement PSDS:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = semilogy(fy2,Py);
    title(['PSD of the ',VibLoc{k},' Vertical Displacement:'])
    xlabel('f [Hz]'),ylabel('PSD_{y} [mm^{2}/Hz]'),grid on
```

```
set(handle,'color',[0 0.7 0.9])
subplot(2,1,2),handle = semilogy(fz2,Pz);
title(['PSD of the ',VibLoc{k},' Horizontal Displacement:'])
xlabel('f [Hz]'),ylabel('PSD_{z} [mm^{2}/Hz]'),grid on
set(handle,'color',[0 0.7 0.\overline{9}])
%-------------------------------------------------------------------------
% Calculate Spectrograms:
[Sy,Fy,Ty,Py] = spectrogram(y_new(:,k),windowl,noverlap,nfft,fs_new);
[Sz,Fz,Tz,Pz] = spectrogram(z_new(:,k),windowl,noverlap,nfft,fs_new);
Ty = Ty + t_new(1); Tz = Tz + t_new(1);
% Vertical Displacement Spectrogram:
fig = fig + 1; figure(fig)
subplot(2,1,1),surf(Ty,Fy,10*log10(Py),'edgecolor','none')
axis tight,view(0,90)
title(['Spectrogram of the ',VibLoc{k},' Vertical Displacement:'])
xlabel('t [s]'),ylabel('f [Hz]')
subplot(2,1,2),handle = plot(t,y(:,k));
title([VibLoc{k},' Vertical Displacement:'])
xlabel('t [s]'),ylabel('y [mm]'), set(handle,'color',[0 0.7 0.9])
A = max (y(:,k)) - min(y(:,k));
if A > 0
    axis([Ty(1) Ty(end) min(y(:,k)) -0.05*A max (y(:,k))+0.05*A])
end
grid on
% Horizontal Displacement Spectrogram:
fig = fig + 1; figure(fig)
subplot(2,1,1),surf(Tz,Fz,10*log10(Pz),'edgecolor','none')
axis tight,view (0,90)
title(['Spectrogram of the ',VibLoc{k},' Horizontal Displacement:'])
xlabel('t [s]'),ylabel('f [Hz]')
subplot(2,1,2),handle = plot(t,z(:,k));
title([VibLoc{k},' Horizontal Displacement:'])
xlabel('t [s]'),ylabel('z [mm]'),set(handle,'color',[0 0.7 0.9])
A = max (z(:,k)) - min(z(:,k));
if A > 0
    axis([Tz(1) Tz(end) min(z(:,k)) -0.05*A max(z(:,k))+0.05*A])
end
grid on
```

end

## B. 5 Generic Programs/Functions

## PolytecScanFFTPlot.m

```
Program that extracts & plots FRFs (mobility) from scanning measurements
performed with a Polytec Scanning Laser Vibrometer. The extracted data
may optionally be stored in a format (ASCII) which may be used in OROS
Modal software.
Note: *The appropriate version of Polytec File Access (PFA) needs to be
    installed (depending on the version of PSV or Vibsoft used to
    acquire the measured data).
    *PFA functions GetPointData.m, GetFFTAcqProps.m,
        GetXYZCoordinates.m and GetVideoImage.m are required in the
        working directory to run this program.
Author: Armand Krüger
% Created: June 2012
% Last revised: February 2013
clear all; close all; clc
```

\%Retrieve Filename:
flag = 1;
while flag
filename = input('Enter filename (*.svd): ','s');
filename $=$ [filename '.svd'];
fid $=$ fopen(filename,'r');
if fid $==-1$
disp([' Error: file "',filename,'" not found'])
disp(' (Enter filename without extension or '' '')')
proc = input(' Proceed (y/n)? ','s');
if proc == 'n'
disp(' Data extraction cancelled')
return
end
else
flag $=0$;
end
end
fclose(fid);
\%Extract \& Plot FRF Data:
domainname = 'FFT';
channelname = 'Vib \& Ref1';
signalname = 'H1 Velocity / Force';
displayname $=$ 'Real \& Imag.';
frame $=0$;
point $=0$;
[f,Fn, usd] =
GetPointData(filename, domainname, channelname, signalname, displayname, point,frame);
Fn_ave $=\operatorname{mean}(\operatorname{abs}($ Fn) );
fftProps $=$ GetFFTAcqProps(filename);
disp(' '),disp('FFT Properties:'),disp(fftProps)
npnts $=$ length (Fn(: 1$)$ );
for pnt = 1:npnts
figure (pnt)
subplot (2,1,1)
semilogy (f, abs(Fn(pnt, :)), 'Color', 0 0.7 0.9])
title(['Point ', num2str(pnt),' - Mobility:'])
xlabel('f [Hz]'),ylabel('Magnitude [m/s/N]')
subplot $(2,1,2)$
plot (f, angle(Fn(pnt,:))*180/pi,'Color', [0 0.7 0.9])
xlabel('f [Hz]'),ylabel('Phase [deg]')
end
figure (pnt+1)
semilogy(f,Fn_ave, 'Color', [0 0.7 0.9]), title('Average Mobility of All Points:')

```
xlabel('f [Hz]'),ylabel('Magnitude [m/s/N]')
%Extract & Display Point Coordinates:
PntCoords = GetXYZCoordinates(filename,point);
pnts = linspace(1,npnts,npnts)';
disp('Point Coordinates:')
disp(' Pnt: X-Coord: Y-Coord: Z-Coord:')
disp([pnts PntCoords])
%Display the Image From the Video Camera of the Laser Vibrometer:
imageData = GetVideoImage(filename);
figure(pnt+2)
image(imageData)
%Store Data in ASCII format:
disp(' ')
store = input('Store this data (y/n)? ','s');
if store == 'y'
    storename = input(' Enter a filename (*.asc): ','s');
    FDstorename = [storename 'FD.asc'];
    fVstorename = [storename 'Fvec.asc'];
    fVector = f';
    FnRe = real(Fn); FnIm = imag(Fn);
    fDomainData = [FnRe(1,:)' FnIm(1,:)'];
    for pnt = 2:npnts
        fDomainData = [fDomainData FnRe(pnt,:)' FnIm(pnt,:)'];
    end
    save(FDstorename','fDomainData','-ascii')
    save(fVstorename,'fVector','-ascii')
    disp([' Frequency-domain data stored in ',FDstorename])
    disp([' Frequency vector data stored in ',fVstorename])
else
    disp(' No data stored')
end
```


## dft.m

```
function [f,F] = dft(x,N,fs)
% [f,Fn] = dft(x,N,fs) returns the complex Fourier coefficients and
% associated frequency range from data vector x.
% Input: x - Data vector; N - No. of DFT points; fs - Sampling frequency
% Output: f - Frequency vector; Fn - Complex Fourier coefficients
% Author: Armand Krüger
% Created: February }201
% Last revised: August 2012
F = fft (x,N); %Do unscaled FFT of x
F = F(1:N/2+1)/N; %Remove second (non-unique) half of FFT
    %(after Nyquest freq) & scale by N
F(2:N/2+1) = 2*F(2:N/2+1); %Compensate for removed half
f = (N/2/(N-1)*fs)*(0:N/2)/(N/2); %Frequency vector from 0 Hz (DC) to just
    %after the Nyquest frequency
```


## expwindow.m

```
function w = expwindow(N,alpha)
% w = expwindow(N,alpha) returns an N-point exponential window in row
vector w.
    Input: -N: number of data points in signal to be windowed
    -alpha: fraction of N at which w = 0.01
    Output: -w: exponential window coefficients row vector
% Author: Armand Krüger
% Created: October 2012
```

```
beta = log(0.01)/(alpha*(N-1));
k = linspace(0,N-1,N);
w = exp(beta*k);
```


## psde.m

```
function [P,f] = psde(x,m,noverlap,fs)
    PSDe One-sided Power Spectral Density Estimate of a data sequence.
    P = psde(x,m,noverlap,fs,tl) performs an FFT analysis of the sequence x.
    The sequence of }n\mathrm{ points are then divided into k sections of m points each
    (m must be a power of two).
    These sections should overlap noverlap points.
    Using an m-point FFT, all sections are Hanning windowed, FFT'd and then
    averaged.
    To compensate for the window the signal amplitude is corrected
    with a correction factor 8/3 (see Encyclopedia of Vibration p 1591)
    PSDe returns the m/2 by 1 array P as function of f where
    P is the Power Spectral Density expressed in [Units^2/Hz]
    and f the corresponding frequencies in [Hz]
    PS Heyns 20 April 2005 (based on original concept by H v Niekerk)
x = x(:); % Ensure x is a column vector
n = max(size(x)); % Number of data points
k = fix((n-noverlap)/(m-noverlap)); % Number of analysis windows
index = 1:m;
w = hanning(m); % Hanning window specification
cf = sqrt(8/3); % correction factor
T = m/fs; % Measurement time (1/df)
Pxx = zeros(m,1); % Initialise PSD
for i=1:k
    xw = (w.*detrend(x(index))).*Cf;
    index = index+(m-noverlap);
    Xx = fft(xw)./m;
    Xx = Xx.*conj(Xx).*T*2; % One-sided correction
    Pxx = Pxx + Xx;
end
Pxx = Pxx(1:m/2); % Consider up to Nyquist
P = Pxx./k;
z = m/2;
f = (0:z-1)/ z*fs/2; % Scale frequency
```


## TachoSpinSpeed.m

```
function [tseg,omega] = TachoSpinSpeed(tacho,t,ppr,triglvl)
% [tseg,omega] = TachoSpinSpeed(tacho,t,ppr,triglvl) calculates spin speed
% from a tachometer signal.
% Input: -tacho: tachometer signal (vector)
            -t: time (vector)
            -ppr: pulses per revolution
            -triglvl: trigger level (at which a tachometer pulse is detected)
Output: -tseg: vector with tachometer pulse locations
                    -omega: spin speed (vector)
% Author: Armand Krüger
% Part of this function is based on code (getrpm.m) by Prof PS Heyns
% Created: March 2013
% Last revised: June 2014
stacho = sign(tacho-triglvl);
dtacho = diff(stacho);
td = t(1:end-1);
```

```
dtheta = 2*pi/ppr;
tseg = td(find(dtacho>0));
dtseg = diff(tseg);
omega = dtheta./dtseg;
```


## SyncAve.m

```
function [tsync,xsync,segs,xsegs,nLS loc] = SyncAve(t,x,tseg,LS)
% [tsync,xsync,segs,xsegs,nLS_loc] = -SyncAve(t,x,tseg,LS) calculates the
synchronous average of a periodic signal from the positions
[tseg,omega] = TachoSpinSpeed(tacho,t,ppr,triglvl) calculates spin speed
from a tachometer signal.
Input: -t: time (vector)
-x: periodic signal (vector)
    -tseg: vector with tachometer pulse locations (single ppr)
    -LS: periodic segments are interpolated to have the same length
        as either the longest segment ('long') or the shortes segment
        ('short') - this parameter may be omitted in which case its
        default value 'short' is used
    Output: -tsync: time vector for synchronised signal
        -xsync: synchronised signal (vector)
        -segs: number of periodic segments in x
        -xsegs: periodic segments matrix
        -nLS_loc: section number of longest/shortes (dependent on LS)
        periodic section
% Author: Armand Krüger
% Part of this function is based on code (getrpm.m) by Prof PS Heyns
Created: March 2013
% Last revised: June 2014
%Determine Positions of Segment Breaks:
segs = length(tseg)-1; %No of middle/full-rotation segements
pos = zeros(segs+1,1); %Positions of segment starts/ends in t
for k = 1:segs+1
    posvec = find(t >= tseg(k));
    pos(k) = posvec(1);
end
%Determine Longest/Shortest (Slowest/Fastest) Segment & tsync:
n = diff(pos);
if nargin < 4 %(Default)
    [nLS,nLS_loc] = min(n);
elseif strcmpi(LS,'long')
    [nLS,nLS loc] = max(n);
elseif strcmpi(LS,'short')
    [nLS,nLS_loc] = min(n);
else
    error(['''',LS,''' is not a valid value for LS (''long'' or ''short'')'])
end
dt = t(2) - t(1);
tsync = 0:dt:dt*nLS; tsync = tsync';
%Determine Synchronous Average:
xsegs = zeros(nLS+1,segs);
xtot = zeros(nLS+1,1);
for s = 1:segs
    x seg = x(pos(s):pos(s+1));
    i\overline{f}}\mathrm{ length(x_seg) ~= length(tsync)
        t_seg = linspace(tsync(1),tsync(end),length(x_seg));
        x_seg = interp1(t_seg,x_seg,tsync);
    end
    xsegs(:,s) = x_seg;
    xtot = xtot + \overline{x_seg;}
end
xsync = xtot/segs;
```


## RunoutComp1.m

```
function u = RunoutCompl(t,g,theta,gRO,tseg)
    u = RunoutComp1(t,g,theta,gRO,tseg) performs runout compensation on
    proximity probe vibration gap signal g using runout gap signal gRO. The
    compensation method employed by this function assumes that the 0 Hz (DC)
    component of the vibration signal remains constant. The function returns
    the compensated displacement (relative to the DC component).
    Input: -t: time (vector)
        -g: vibration gap signal (vector) g(t)
        -theta: angular position (vector)
        -gRO: runout gap signal (vector) gRO(theta)
        -tseg: vector with tachometer pulse locations (single ppr)
    Output: u: compensated displacement signal (vector)
% Author: Armand Krüger
Created: March 2013
Last revised: June 2014
%DETERMINE POSITIONS OF SEGMENT BREAKS:
segs = length(tseg)-1; %No of middle/full-rotation segements
pos = zeros(segs+1,1); %Positions of segment breaks in t
for k = 1:segs+1
    posvec = find(t >= tseg(k));
    pos(k) = posvec(1);
end
%CALCULATE CENTRELINE DISPLACEMENT FOR EACH SEGMENT:
u = zeros(length(g),1);
% Segment 0 (1st semi-segment - if any):
if pos(1) ~= 1
    n_seg0 = pos(1) - 1; n_seg1 = pos(2) - pos(1);
    g_seg0 = g(1:pos(1));
    if n_seg0 < n_seg1
            % Interpolate Runout for Segment 1:
            if n_seg1+1 ~= length(theta)
                    theta seg1 = linspace(theta(1),theta(end),n_seg1+1)';
                    gRO_seg1 = interp1(theta,gRO,theta_seg1);
            else
                    gRO_seg1 = gRO;
            end
            % Partial Runout for Segment 0:
            gRO_seg0 = gRO_seg1 (end-n_seg0: end);
        else
            % Interpolate Runout for Segment 0:
            if n seg0+1 ~= length(theta)
                    theta_seg0 = linspace(theta(1), theta(end),n_seg0+1)';
                    gRO seg0 = interp1(theta,gRO,theta seg0);
            else
                    gRO_seg0 = gRO;
            end
        end
        % Subtract Runout Signal from Segment 0 Signal:
    u_seg0 = gRO_seg0 - g_seg0;
    u(1: pos (1)) = u_seg0;
end
% Middle Segments:
for s = 1:segs
    g_seg = g(pos(s):pos(s+1));
    if length(g_seg) ~= length(theta)
            theta_seg = linspace(theta(1),theta(end),length(g_seg))';
            gRO_seg = interp1(theta,gRO,theta_seg);
    else
            gRO seg = gRO;
    end
    u_seg = gRO_seg - g_seg;
```

```
    u(pos(s):pos(s+1)) = u_seg;
end
% Segment segs+1 (last semi-segment - if any):
if pos(end) ~= length(g)
    g_segsp1 = g(pos(end):end);
    n_segs = pos(segs+1) - pos(segs); n_segsp1 = length(g_segsp1)-1;
    if n_segsp1 < n_segs
        % Interpolate Runout for Segment segs:
        if n segs+1 ~= length(theta)
            theta_segs = linspace(theta(1),theta(end),n_segs+1)';
            gRO_segs = interp1(theta,gRO,theta_segs);
        else
            gRO segs = gRO;
        end
        % Partial Runout for Segment segs+1:
        gRO_segsp1 = gRO_segs(1:n_segsp1+1);
    else
        % Interpolate Runout for Segment segs+1:
        if n_segsp1+1 ~= length(theta)
                theta segsp1 = linspace(theta(1),theta(end),n segsp1+1)';
                gRO_segsp1 = interp1(theta,gRO,theta_segsp1);
            else
            gRO_segsp1 = gRO;
            end
    end
    % Subtract Runout Signal from Segment segs+1 Signal:
    u_segsp1 = gRO_segsp1 - g_segsp1;
    u(pos(end):length(g)) = u_segsp1;
end
```


## RunoutComp2.m

```
function x = RunoutComp2(t,g,theta,gRO,tseg,c,gh)
% x = RunoutComp2(t,g,theta,gRO,tseg,c,gh) performs runout compensation on
    proximity probe vibration gap signal g using runout gap signal gRO. The
    compensated shaft/journal centre coordinate (in the probe measurement
    direction) is calculated relative to the bearing housing centre. This
    compensation method employed by this function assumes that the average of
    the runout gap signal is equal to the actual gap between the
    shaft/journal and the probe.
    Input: -t: time (vector)
        -g: vibration gap signal (vector) g(t)
        -theta: angular position (vector)
        -gRO: runout gap signal (vector) gRO(theta)
        -tseg: vector with tachometer peak locations (single ppr)
        -c: clearance
        -gh: gap between the housing and the proximity probe
    Output: x: compensated shaft/journal centre coordinate (vector)
% Author: Armand Krüger
% Created: March 2013
% Last revised: June 2014
%DETERMINE POSITIONS OF SEGMENT BREAKS:
segs = length(tseg)-1; %No of middle/full-rotation segements
pos = zeros(segs+1,1); %Positions of segment breaks in t
for k = 1:segs+1
    posvec = find(t >= tseg(k));
    pos(k) = posvec(1);
end
%CALCULATE CENTRELINE POSITION FOR EACH SEGMENT:
uRO = -detrend(gRO,'constant');
x = zeros(length(g),1);
% Segment 0 (1st semi-segment - if any):
if pos(1) ~= 1
```

```
    n_seg0 = pos(1) - 1; n_seg1 = pos(2) - pos(1);
    g_seg0 = g(1:pos(1));
    if n seg0 < n seg1
        % Interpolate Runout for Segment 1:
        n_seg1 = pos(2) - pos(1);
        if n_seg1+1 ~= length(theta)
            theta_seg1 = linspace(theta(1),theta(end),n_seg1+1)';
            uRO_seg1 = interp1(theta,uRO,theta_seg1)
        else
            uRO_seg1 = uRO;
        end
        % Partial Runout for Segment 0:
        uRO_seg0 = uRO_seg1 (end-n_seg0:end);
    else
        % Interpolate Runout for Segment 0:
        if n_seg0+1 ~= length(theta)
            theta_seg0 = linspace(theta(1),theta(end),n_seg0+1)';
            uRO_seg0 = interp1(theta,uRO,theta_seg0);
        else
            uRO_seg0 = uRO;
        end
    end
    % Subtract Runout Signal from Segment 0 Signal:
    x_seg0 = c - uRO_seg0 + gh - g_seg0;
    x(1:pos(1)) = x_seg0;
end
% Middle Segments:
for s = 1:segs
    g_seg = g(pos(s):pos(s+1));
    if length(g_seg) ~= length(theta)
        theta_seg = linspace(theta(1),theta(end),length(g_seg))';
        uRO seg = interp1(theta,uRO,theta seg);
    else
        uRO_seg = uRO;
    end
    x_seg = c - uRO_seg + gh - g_seg;
    x(pos(s):pos(s+\overline{1})) = x_seg;
end
% Segment segs+1 (last semi-segment - if any):
if pos(end) ~= length(g)
    g_segsp1 = g(pos(end):end);
    n_segs = pos(segs+1) - pos(segs); n_segsp1 = length(g_segsp1)-1;
    if n_segsp1 < n_segs
        % Interpolate Runout for Segment segs:
        if n_segs+1 ~= length(theta)
            theta_segs = linspace(theta(1),theta(end),n_segs+1)';
            uRO_seggs = interp1(theta,uRO,theta_segs);
        else
            uRO segs = uRO;
        end
        % Partial Runout for Segment segs+1:
        uRO_segsp1 = uRO_segs(1:n_segsp1+1);
    else
        % Interpolate Runout for Segment segs+1
        if n segsp1+1 ~= length(theta)
            theta_segsp1 = linspace(theta(1),theta(end),n_segsp1+1)';
            uRO_segsp1 = interp1(theta,uRO,theta_segsp1);
        else
            uRO_segsp1 = uRO;
        end
    end
    % Subtract Runout Signal from Segment segs+1 Signal:
    x segsp1 = c - uRO segsp1 + gh - g_segsp1;
    x(pos(end):length(\overline{g})) = x_segsp1;
end
```

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems

## Appendix C: MATLAB Programs - Numerical Modelling

## C. 1 MSC.Nastran Results Processing

## CampbellDiag_CompPCH.m

```
% A program that extracts and plots rotor Campbell diagram results from an
% MSC.Nastran(TM) PCH output file.
Note: *Results from multiple subcases are not supported.
% *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: August 2013
% Last revised: June 2014
clear all; clc; close all
%EXTRACT EIGENVALUES:
% Get PCH File Info:
flag = 1;
while flag
    filename = input('Enter the PCH filename (*.pch): ','s');
    filename = [filename '.pch'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or ''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
                disp(' Data extraction cancelled')
                return
        end
    else
        flag = 0;
    end
end
% Extract Data from PCH File:
SEIDspec = 0; %::ID of superelement data to extract (if applicable)
n = 1; k = 0; eof = 0; disp(' ')
while eof == 0 %loop through lines
    line = fscanf(fid,'%c',[15,1])';
    if strcmp(line,'$REAL-IMAGINARY')
        dummy = fscanf(fid,'%c',[10,1])';
        SE = fscanf(fid,'%c',[5,1])';
        SEID = fscanf(fid,'%d',[1,1]);
        if SEID == SEIDspec || not(strcmp(SE,'SEID='))
            dummy = fgetl(fid); dummy = fgetl(fid);
            dummy = fscanf(fid,'%c',[17,1])';
            sigma = fscanf(fid,'%f \n',[1,1]);
            dummy = fscanf(fid,'%c',[2,1])';
            wd = fscanf(fid,'%f \n',[1,1]);
            dummy = fscanf(fid,'%c',[12,1])';
            mode = fscanf(fid,'%d',[1,1]);
            if mode == 1
                k = k + 1;
                    disp(['Done extracting data for spin speed # ',num2str(k)])
            end
            sr(k,mode) = sigma + 1i*wd;
        end
    end
    dummy = fgetl(fid);
    eof = feof(fid);
    n = n + 1;
end
fclose(fid);
```

```
% Remove Complex Conjugates (optional):
remdup = 1; %::=1: remove duplicates; =0: keep duplicates
if remdup == 1
    wd = imag(sr);
    index = find(wd(floor(k/2),:) >= 0);
    sr = sr(:,index);
end
% Extract Specific Natural (Whirl) Frequencies from the PCH File Data:
sr = flipud(sr);
sigma = real(sr); wd = imag(sr);
wr = sqrt(wd.^2 + sigma.^2); %Natural frequencies [rad/s]
fr = wr/(2*pi); %Natural frequencies [Hz]
zr = -sigma./wr; %Damping ratios
%GET SPIN SPEED- & PLOTTING INFO:
disp(' ')
flag = 1;
while flag
    N start = input('Enter starting rotor spin speed [rpm]: ');
    N_inc = input('Enter spin speed increment [rpm]: ');
    N_max = input('Enter maximum spin speed to plot [rpm]: ');
    if N_start + N_inc > N_max
            disp(' Error: N_start + N_increment must be <= N_max')
    else
            flag = 0;
    end
end
disp(' ')
modes_inp = input('Enter the maximum number of modes to plot (default:20): ');
if isempty(modes inp)
    nmodesmax = \overline{20;}
else
    nmodesmax = modes_inp;
end
%CONSTRUCT SPIN SPEED VECTOR &...
%...EXTRACT WHIRL FREQUENCIES FOR SPECIFIED SPIN SPEED RANGE:
N_end = N_start + (k-1)*N_inc;
N
Nindex = find( }\textrm{N}<=\overline{\textrm{N}}\mathrm{ _max);
N = N(Nindex); fr = \́r(Nindex,:);
f = N/60;
%SORT MODES IN ASCENDING ORDER ACCORDING TO FIRST FREQUENCY:
[~,sorder] = sort(fr(1,:));
fr = fr(:,sorder);
%TRACK MODES (optional):
[l,nmodes] = size(fr);
trackm = 1; %::=1: track modes; =0: do not track modes
if trackm == 1
    disp(' ')
    gradtoll = le-1; %::tolerance for change in line gradient
    frdf = diff(fr); frd2f = diff(frdf);
    for fstep = 2:l-2
        for mode = 1:nmodes-1
            if abs(frd2f(fstep,mode)) > gradtoll
                % Find Mode to Swap with:
                nswap = 1;
                while abs(frd2f(fstep,mode+nswap)) <= gradtoll
                    nswap = nswap + 1;
                if mode + nswap > nmodes
                    break
                    end
                end
                % Swap Modes:
                if mode + nswap <= nmodes
                    temp = fr(fstep+2:l,mode);
```

```
                    fr(fstep+2:l,mode) = fr(fstep+2:l,mode+nswap);
                    fr(fstep+2:l,mode+nswap) = temp;
                    swaped = 1;
                    frdf = diff(fr); frd2f = diff(frdf);
                    disp([' Mode ', num2str(mode),' swapped with mode ',...
                        num2str(mode+nswap),' from frequency no. ',...
                        num2str(fstep+2)])
                end %if
            end %if
        end %for
    end %for
end %if
%CAMPBELL DIAGRAM:
% Specify Line Colours:
nmodesplot = min(nmodes,nmodesmax);
cmapsize = nmodesplot+100; %::Jet colormap size (refer to 'Colormap' in Matlab
help)
crange = 1; %::fraction of colormap used (range)
coffst = 0; %::fraction of colormap offset (crange + coffst <= 1)
temp = 1; %::1 = warm; 0 = cold
darkness = 1; %::smaller = darker (must be < 1)
cmap = darkness*colormap(jet(cmapsize+1));
if temp
    cmap = flipud(cmap);
end
if nmodesplot < 2
    ndiv = 2;
else
    ndiv = nmodesplot;
end
kfact = floor(crange*cmapsize/(ndiv-1));
koffst = floor(coffst*cmapsize);
% Plot Campbell Diagram:
figure(1)
plot([0 f(end)],[0 f(end)],'k--','Color',0.6*[lllll
lgnd{1} = 'Whirl Freq. = Spin Speed';
hold all
handle = zeros(1,nmodes);
for mode = 1:nmodesplot
    handle(mode) = plot(f,fr(:,mode));
    kcmap = (mode-1)*kfact + koffst + 1;
    set (handle(mode),'Color',cmap (kcmap, :))
    lgnd{mode+1} = ['Mode ' num2str(mode)];
end
hold off
axis([f(1) f(end) 0 max(max(fr(:,1:nmodesplot)))*1.05])
title('Campbell Diagram:')
xlabel('Spin Speed [Hz]'),ylabel('Natural Whirl Frequency [Hz]')
legend(lgnd)
```


## RotorDispl_FreqRespPCH.m

```
A program that extracts, plots and saves (optional) rotor frequency
```

A program that extracts, plots and saves (optional) rotor frequency
response displacment results from an MSC.Nastran(TM) PCH output file.
response displacment results from an MSC.Nastran(TM) PCH output file.
Note: *Lateral displacements must be in the yz-plane.
Note: *Lateral displacements must be in the yz-plane.
*The PCH file may contain the displacements of multiple grid
*The PCH file may contain the displacements of multiple grid
points/nodes.
points/nodes.
*Multiple result types not supported (only displacements).
*Multiple result types not supported (only displacements).
*Results from multiple subcases are not supported.
*Results from multiple subcases are not supported.
*User-modifiable parameters/code indicated by :: in comments.
*User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Author: Armand Krüger
% Created: March 2013
% Created: March 2013
% Last revised: June 2014
% Last revised: June 2014
clear all; clc; close all

```
clear all; clc; close all
```

```
%%%%%%%%%%%%%%%%%%%%%%% EXTRACT DISPLACEMENT DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET PCH FILE- & EXTRACTION INFO:
flag = 1;
while flag
    filename = input('Enter the PCH filename (*.pch): ','s');
    filename = [filename '.pch'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or '''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
                disp(' Data extraction cancelled')
                return
        end
    else
        flag = 0;
    end
end
disp(' ')
f start = input('Enter starting frequency of response(s) to be extracted: ');
f_end = input('Enter ending frequency of response(s) to be extracted: '); disp(' ')
nspec = lower(input('Only extract response of a specific node(s) (y/n)? ','s'));
if nspec == 'y'
    nodes = input(' Enter node number(s) [in vector format if > 1]: ');
else
    nspec = 'n';
    disp(' Automatic node response extraction enabled')
    disp(' Press any key to continue... '),pause
end
disp(' ')
%EXTRACT DATA FROM PCH FILE:
dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid);
datatype = fscanf(fid,'%c',[14,1])'; dummy = fgetl(fid);
if not(strcmp(datatype,'$DISPLACEMENTS'))
    disp(['Error: Incorrect data type in file ',filename])
    disp(' Program terminated'),fclose(fid);
    return
end
SEIDspec = 0; %::ID of superelement data to extract (if applicable)
SEfound = 0;
dummy = fscanf(fid,'%c',[25,1])'; SE = fscanf(fid,'%c',[5,1])';
SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
if strcmp(SE,'SEID=') %if superelements detected
    SEfound = 1; found = 0;
    while found == 0
        if feof(fid) %if end of file reached
                disp(['Error: No displacement data found for SE ',...
                    num2str(SEIDspec),' in file ',filename])
                disp(' Program terminated'),fclose(fid);
                return
            end
            if strcmp(SE,'SEID=') && SEID == SEIDspec
                found = 1;
            else
                dummy = fscanf(fid,'%c',[25,1])'; SE = fscanf(fid,'%c',[5,1])';
                SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
            end
    end
end
% Loop through frequency steps:
n = 1; eof = 0; f n = 0; nextdata = 0;
while f n < f end && nextdata == 0
```



```
    dummy = fscanf(fid,'%c',[12,1])';
    if n == 1
        if not(strcmp(dummy,'$FREQUENCY ='))
        disp(['Error: Incorrect result type in file ',filename])
```

```
            disp(' Program terminated'),fclose(fid);
            return
        end
    end
    f n = fscanf(fid,'%f \n',[1,1]);
    f(n) = f_n; dummy = fgetl(fid);
    % Loop through nodes:
    k = 1; kspec = 0; %k: nodes in file; kspec: specified nodes found
    next = ' ';
    while eof == 0 && not(strcmp(next,'$'))
    node = fscanf(fid,'%d',[1,1]);
    if n == 1 && nspec ~= 'y' %if auto node extraction enabled & @ f1
        nodes(k) = node; %store node found
    end
    dummy = fscanf(fid,'%c',[14,1])';
    displtemp = fscanf(fid,'%f \n',[3,1])'; dummy = fgetl(fid);
    dummy = fscanf(fid,'%c',[20,1])';
    displtemp(1,4:6) = fscanf(fid,'%f \n',[3,1])'; dummy = fgetl(fid);
    dummy = fscanf(fid,'%c',[20,1])';
    displtemp(1,7:9) = fscanf(fid,'%f \n',[3,1])'; dummy = fgetl(fid);
    dummy = fscanf(fid,'%c',[20,1])';
    displtemp(1,10:12) = fscanf(fid,'%f \n',[3,1])';
    if nspec == 'y' %if nodes specified by user
        compare = find(nodes == node);
        if not(isempty(compare)) %if node found
                kspec = kspec + 1;
                displ(n,:,kspec) = displtemp;
                pos(kspec) = compare; %store node position
        end
    else %if auto. node extraction enabled
        displ(n,:,k) = displtemp;
    end
    dummy = fgetl(fid);
    next = fscanf(fid,'%c',[1,1]);
    eof = feof(fid); %check for end of file
    k = k + 1;
    end %nodes loop end
    disp(['Displacement response read for f ', num2str(n),' = ',num2str(f_n),' Hz'])
    if eof %if end of file reached in nodes loop
    break
    end
    if nspec == 'y' && kspec == 0 %if none of specified node(s) found
    disp([' Error: Response data not found for specified node(s) in...
        file ',filename])
    disp(' Program terminated'),fclose(fid);
    return
    end
    dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid);
    if SEfound
    dummy = fscanf(fid,'%c',[25,1])'; SE = fscanf(fid,'%c',[5,1])';
    SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
    if SEID ~= SEIDspec %if different superelement found
                nextdata = 1;
    end
else
    dummy = fgetl(fid);
    end
    n = n + 1;
end %frequency steps loop end
if nspec == 'y' %if nodes spec. by user
    nodespos = 1:length(nodes);
    nodesmiss = setxor(nodespos,pos);
    if not(isempty(nodesmiss)) %if certain spec. node(s) not found
        disp(' ')
        disp('Note: Response data not found for the following specified nodes:')
        disp(nodes(nodesmiss))
    end
    nodes = nodes(pos);
end
```

```
fclose(fid);
%EXTRACT SPECIFIC DISPLACEMENT RESPONSES:
f = f';
index = find(f >= f_start);
f = f(index);
y = squeeze(displ(index,2,:)) + 1i*squeeze(displ(index,8,:));
z = squeeze(displ(index,3,:)) + 1i*squeeze(displ(index,9,:));
theta = squeeze(displ(index,4,:)) + 1i*squeeze(displ(index,10,:));
clear displ
%%%%%%%%%%%%%%%% CALCULATE DISPLACEMENT MAGNITUDES & PHASES %%%%%%%%%%%%%%%
ymag = abs(y); yphase = angle(y);
zmag = abs(z); zphase = angle(z);
thetamag = abs(theta); thetaphase = angle(theta);
%%%%%%%%%%%%%%%%%%%%%%%%% PLOT & SAVE DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PLOT DISPLACEMENTS:
disp(' ')
incl_ang = lower(input('Include angular displacement figures (y/n)? ','s'));
fig = 0; nnodes = length(nodes);
for k = 1:nnodes
    node = num2str(nodes(k));
    % Vertical- & Horizontal Displacements:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = plot(f,ymag(:,k));
    title(['Vertical Displacement Response of Node ',node,':'])
    xlabel('f [Hz]'),ylabel('|y| [mm]'), set(handle,'color',[0 0.7 0.9])
    A = max (ymag(:,k)) - min(ymag(:,k));
    if A > 0
        axis([f(1) f(end) min(ymag(:,k))-0.05*A max(ymag(:,k)) +0.05*A])
    end
    subplot(2,1,2),handle = plot(f,yphase(:,k)*180/pi);
    xlabel('f [Hz]'),ylabel('\angley [deg]'), set(handle,'color',[0 0.7 0.9])
    yphasemax = max(yphase(:,k))*180/pi; yphasemin = min(yphase(:,k))*180/pi;
    A = yphasemax - yphasemin;
    if A > 0
    axis([f(1) f(end) yphasemin-0.05*A yphasemax+0.05*A])
    end
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = plot(f,zmag(:,k));
    title(['Horizontal Displacement Response of Node ',node,':'])
    xlabel('f [Hz]'),ylabel('|z| [mm]'), set(handle,'color',[0 0.7 0.9])
    A = max(zmag(:,k)) - min(zmag(:,k));
    if A > 0
    axis([f(1) f(end) min(zmag(:,k)) -0.05*A max(zmag(:,k)) +0.05*A])
    end
    subplot(2,1,2),handle = plot(f,zphase(:,k)*180/pi);
    xlabel('f [Hz]'),ylabel('\anglez [deg]'),set(handle,'color',[0 0.7 0.9])
    zphasemax = max(zphase(:,k))*180/pi; zphasemin = min(zphase(:,k))*180/pi;
    A = zphasemax - zphasemin;
    if A > 0
        axis([f(1) f(end) zphasemin-0.05*A zphasemax+0.05*A])
    end
    % 2D Displacement Orbits:
    theta_orb = linspace(0,2*pi*0.95,100);
    fig = fig + 1; figure(fig)
    f_int = 1; %::Frequency value interval
    hold on
    for kf = 1:f_int:length(f)
        y_orb = ymag(kf,k)*cos(theta_orb - yphase(kf,k));
        z orb = zmag(kf,k)*cos(theta orb - zphase(kf,k));
        f_k = ones(length(theta_orb),1);
        f_k = f_k*f(kf);
        hāndle = plot3(f_k,z_orb,y_orb,f_k(1),z_orb(end),y_orb(end),'.');
        set(handle,'color',[0 0.7 0.9])
    end
```

```
    hold off
    f_scale = 10; %::Frequency scale factor (f : displacement)
    title(['2D Displacement Orbits of Node ',node,' as a Function of Frequency:'])
    xlabel('f [Hz]'),ylabel('z [mm]'), zlabel('y [mm]')
    scale = f_scale*max(max(ymag(:,k)),max(zmag(:,k)))/(f(end) -f(1));
    if scale > 0
        set(gca,'DataAspectRatio',[1 scale scale])
    end
    % Angular Displacement:
    if incl_ang == 'Y'
        fig = fig + 1; figure(fig)
        subplot(2,1,1),handle = plot(f,thetamag(:,k)*180/pi);
        title(['Angular Displacement Response of Node ',node,':'])
        xlabel('f [Hz]'),ylabel('|0_{x}| [deg]'),set(handle,'color',...
            [0 0.7 0.9])
        thetamax = max(thetamag(:,k))*180/pi;
        thetamin = min(thetamag(:,k))*180/pi;
        A = thetamax - thetamin;
        if A > 0
            axis([f(1) f(end) thetamin-0.05*A thetamax+0.05*A])
        end
        subplot(2,1,2),handle = plot(f,thetaphase(:,k)*180/pi);
        xlabel('f [Hz]'),ylabel('\angle0_{x} [deg]')
        set(handle,'color',[0 0.7 0.9])
        thetamax = max(thetaphase(:,k))*180/pi;
        thetamin = min(thetaphase(:,k))*180/pi;
        A = thetamax - thetamin;
        if A > 0
            axis([f(1) f(end) thetamin-0.05*A thetamax+0.05*A])
        end
    end
end
%SAVE DISPLACEMENTS:
disp(' '),store = lower(input('Save this data (y/n)? ','s'));
if store == 'y'
    filename = input(' Enter a basic filename (*.cSv): ','s');
    for k = 1:nnodes
        node = num2str(nodes(k));
        filenamek = [filename,' n',node,'.cSV'];
        csvwrite(filenamek,[f y(:,k) z(:,k) theta(:,k)])
        if nnodes > 1
                disp([' Data saved as ',filenamek,'...'])
            else
                disp([' Data saved as ',filenamek])
        end
    end
    if nnodes > 1
            disp(' Done saving data')
    end
else
    disp(' No data saved')
end
```


## RotorDispl_FreqRespCSV.m

```
% A program that loads and plots rotor frequency response displacements
% from CSV files (created by RotorDispl_FreqRespPCH.m).
Note: *Lateral displacements are assumed to be in the yz-plane.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: March 2013
% Last revised: June 2014
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%% LOAD DISPLACEMENT DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%GET CSV FILE INFO:
flag = 1;
while flaq
    filename = input('Enter the basic CSV filename...
        [without suffix '' n<node>''] (*.csv): ','s');
    if strcmp(filename(end),'''') || strcmp(filename(end-3:end),'.csv')
                disp(' Enter filename without extension or ''''')
                proc = lower(input(' Proceed (y/n)? ','s'));
                if proc == 'n'
                    disp(' Data loading cancelled')
                    return
        end
    else
            flag = 0;
    end
end
nodes = input('Enter node number(s) [in vector format if > 1]: ');
f start = input('Enter starting frequency of response to be loaded: ');
f_end = input('Enter ending frequency of response to be loaded: ');
%LOAD CSV DATA:
nnodes = length(nodes);
kfound = 1;
for k = 1:nnodes
    node = num2str(nodes(k));
    if filename(end) == '_'
        filenamek = [filename,'n',node,'.csv'];
    else
        filenamek = [filename,'_n',node,'.csv'];
    end
    fid = fopen(filenamek,'r');
    if fid == -1
        disp([' File ''',filenamek,''' not found'])
    else
        data = csvread(filenamek);
        f = data(:,1);
        Y(:,kfound) = data(:,2); z(:,kfound) = data(:,3); ...
            theta(:,kfound) = data(:,4);
        pos(kfound) = k;
        kfound = kfound + 1;
        disp([' Displacement response loaded for node ',node,' (',filenamek,')'])
        fclose(fid);
    end
end
clear data
if not(exist('pos','var'))
    disp(' Error: None of the specified CSV files found')
    disp(' Program terminated')
    return
end
nodes = nodes(pos);
%APPLY FREQUENCY LIMITS
index = find(f >= f start & f <= f end);
f = f(index); y = y(index,:); z = \overline{z}(index,:); theta = theta(index,:);
%%%%%%%%%%%%%%%% CALCULATE DISPLACEMENT MAGNITUDES & PHASES %%%%%%%%%%%%%%%
ymag = abs(y); yphase = angle(y);
zmag = abs(z); zphase = angle(z);
thetamag = abs(theta); thetaphase = angle(theta);
%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


## RotorDispl_TransPCH.m

```
A program that extracts, plots and saves (optional) transient rotor
displacment results from an MSC.Nastran(TM) PCH output file.
Note: *Lateral displacements must be in the yz-plane.
    *The PCH file may contain the displacements of multiple grid
    points/nodes and may also contain other transient result types.
    *Results from multiple subcases are not supported.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: February }201
% Last revised: June 2014
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%% EXTRACT DISPLACEMENT DATA % % % % % % % % % % % % % % % % % % % % % % % % % 
%GET PCH FILE- & EXTRACTION INFO:
flag = 1;
while flag
    filename = input('Enter the PCH filename (*.pch): ','s');
    filename = [filename '.pch'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or ''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data extraction cancelled')
            return
        end
    else
        flag = 0;
    end
end
disp(' ')
t_start = input('Enter starting time of signal(s) to be extracted: ');
t end = input('Enter ending time of signal(s) to be extracted: '); disp(' ')
nspec = lower(input('Only extract displacements of specific node(s) (y/n)? ','s'));
if nspec == 'y'
    nodes = input(' Enter node number(s) [in vector format if > 1]: ');
else
    nspec = 'n';
    disp(' Automatic node displacement extraction enabled')
    disp(' Press any key to continue... '),pause
end
```

\%SEARCH FOR DISPLACEMENT DATA IN PCH FILE:
SEIDspec $=0$; : ID of superelement data to extract (if applicable)
disp(' '),disp('Searching for displacement data...')
eof $=0$; found $=0$; SEfound $=0$;
while found $==0$
line $=$ fscanf(fid,'\%c', [14,1])'; dummy = fgetl(fid);
if strcmp(line,'\$DISPLACEMENTS') \%if displacement data found
dummy = fscanf(fid,'\%c', $[25,1])^{\prime} ; ~ S E=f s c a n f(f i d, ' \% c ',[5,1]) ' ;$
SEID = fscanf(fid,' $\%$ d', [1,1]); dummy = fgetl(fid);
if strcmp(SE,'SEID=') \%if superelements detected
SEfound = 1;
if SEID == SEIDspec
found $=2$;
end
else
found $=1$;
end
end
if feof(fid) \%if end of file reached
if SEfound
disp([' Error: No displacement data found for SE ',...
num2str(SEIDspec),' in file ',filename])

```
        else
        disp([' Error: No displacement data found in file ',filename])
        end
        disp(' Program terminated'),fclose(fid);
        return
    end
end
%EXTRACT DATA FROM PCH FILE:
% Loop through time steps:
n = 1; t_n = 0; datanext = 0;
while t_n < t_end && datanext == 0
    dummy = fgetl(fid);
    dummy = fscanf(fid,'%c',[7,1])';
    if n == 1
        if not(strcmp(dummy,'$TIME ='))
            disp(' Error: Displacement data type not transient')
            disp(' Program terminated'),fclose(fid);
            return
        end
    end
    t_n = fscanf(fid,'%f \n',[1,1]);
    t(n) = t_n; dummy = fgetl(fid);
    % Loop through nodes:
    k = 1; kspec = 0; %k: nodes in file; kspec: specified nodes found
    next = ' ';
    while eof == 0 && not(strcmp(next,'$'))
    node = fscanf(fid,'%d',[1,1]);
    if n == 1 && nspec ~= 'y' %if auto node extraction enabled & @ t1
            nodes(k) = node; %store node found
        end
        dummy = fscanf(fid,'%c',[10,1])';
        displtemp = fscanf(fid,'%f \n',[3,1])'; dummy = fgetl(fid);
        dummy = fscanf(fid,'%c',[20,1])';
        displtemp(1,4:6) = fscanf(fid,'%f \n',[3,1])';
        if nspec == 'y' %if nodes specified by user
            compare = find(nodes == node);
            if not(isempty(compare)) %if node found
                kspec = kspec + 1;
                    displ(n,:,kspec) = displtemp;
                    pos(kspec) = compare; %store node position
            end
        else %if auto node extraction enabled
            displ(n,:,k) = displtemp;
        end
        dummy = fgetl(fid);
        next = fscanf(fid,'%c',[1,1]);
        eof = feof(fid); %check for end of file
        k = k + 1;
    end %nodes loop end
    disp(['Displacements read for t ',num2str(n),' = ',num2str(t_n),' s'])
    if eof %if end of file reached in nodes loop
            break
    end
    if nspec == 'y' && kspec == 0 %if none of specified node(s) found
            disp([' Error: Displacement data not found for specified node(s) in...
                file ',filename])
            disp(' Program terminated'),fclose(fid);
            return
    end
    dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid);
    datatype = fscanf(fid,'%c',[14,1])'; dummy = fgetl(fid);
    if not(strcmp(datatype,'$DISPLACEMENTS')) %if different data type found
        datanext = 1;
    else
        if SEfound
            dummy = fscanf(fid,'%c',[30,1])';
            SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
            if SEID ~= SEIDspec %if different superelement found
```

```
                datanext = 1;
            end
        else
        dummy = fgetl(fid);
        end
    end
    n = n + 1;
end %time steps loop end
if nspec == 'y' %if nodes specified by user
    nodespos = 1:length(nodes);
    nodesmiss = setxor(nodespos,pos);
    if not(isempty(nodesmiss)) %if certain spec. node(s) not found
            disp(' ')
            disp('Note: Displacement data not found for the following...
                specified nodes:')
            disp(nodes(nodesmiss))
    end
    nodes = nodes(pos);
end
fclose(fid);
%EXTRACT SPECIFIC DISPLACEMENT SIGNALS:
t = t';
index = find(t >= t_start);
t = t(index);
y = squeeze(displ(index,2,:));
z = squeeze(displ(index,3,:));
theta = squeeze(displ(index,4,:));
clear displ
```



```
mag = sqrt(y.^2 + z.^2);
nnodes = length(nodes);
displ_max = zeros(nnodes,1); t_max = displ_max; disp(' ')
for k-= 1:nnodes;
    [displ_max(k),mpos] = max(mag(:,k)); t_max(k) = t(mpos);
    disp(['Maximum displacement of node ',\overline{num2str(nodes(k)),' = ',...}
        num2str(displ_max(k)),' mm at t = ',num2str(t_max(k)),' s'])
end
%%%%%%%%%%%%%%%%%%%%%%%% PLOT & SAVE DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%
%PLOT DISPLACEMENTS:
disp(' ')
incl_ang = lower(input('Include angular displacement figures (y/n)? ','s'));
fig = 0;
for k = 1:nnodes
    node = num2str(nodes(k));
    % Vertical- & Horizontal Displacements:
    fig = fig + 1; figure(fig)
    subplot(2,1,1), handle = plot(t,y(:,k));
    title(['Vertical Displacement of Node ',node,':'])
    xlabel('t [s]'),ylabel('y [mm]'),set(handle,'color',[0 0.7 0.9])
    A = max(y(:,k)) - min(y(:,k));
    if A>0
        axis([t(1) t(end) min(y(:,k))-0.05*A max(y(:,k))+0.05*A])
    end
    subplot(2,1,2),handle = plot(t,z(:,k));
    title(['Horizontal Displacement of Node ',node,':'])
    xlabel('t [s]'),ylabel('z [mm]'),set(handle,'color',[0 0.7 0.9])
    A = max(z(:,k)) - min(z(:,k));
    if A > 0
        axis([t(1) t(end) min(z(:,k))-0.05*A max(z(:,k))+0.05*A])
    end
    % Displacement Orbit:
    fig = fig + 1; figure(fig)
    handle = plot(z(:,k),y(:,k));
    title(['Displacement Orbit of Node ',node,':'])
    xlabel('z [mm]'),ylabel('y [mm]')
```

```
    axis equal,set(handle,'color',[0 0.7 0.9])
    % Displacement Magnitude:
    fig = fig + 1; figure(fig)
    handle = plot(t,mag(:,k));
    title(['Displacement Magnitude of Node ',node,':'])
    xlabel('t [s]'),ylabel('r [mm]'),set(handle,'color',[0 0.7 0.9])
    A = max (mag(:,k)) - min(mag(:,k));
    if A > 0
    axis([t(1) t(end) min(mag(:,k)) -0.05*A max(mag(:,k)) +0.05*A])
    end
    % 2D Displacement:
    fig = fig + 1; figure(fig)
    t_scale = 3; %::Time scale factor (t : displacement)
    handle = plot3(t,z(:,k),y(:,k));
    title(['2D Displacement of Node ',node,' as a Function of Time:'])
    xlabel('t [s]'),ylabel('z [mm]'), zlabel('y [mm]')
    set(handle,'color', [0 0.7 0.9])
    A = max(max(y(:,k)) - min(y(:,k)),max(z(:,k)) - min(z(:,k)));
    scale = t_scale*A/(t(end)-t(1));
    if scale > 0
        set(gca,'DataAspectRatio',[1 scale scale])
    end
    % Angular Displacement:
    if incl_ang == 'y'
        fig= fig + 1; figure(fig)
        handle = plot(t,theta(:,k)*180/pi);
        title(['Angular Displacement of Node ', node,':'])
        xlabel('t [s]'),ylabel('0_{x} [deg]')
        set(handle,'color',[0 0.7 0.9])
        A = max(theta(:,k)*180/pi) - min(theta(:,k)*180/pi);
        if A > 0
            axis([t(1) t(end) min(theta(:,k)*180/pi)-0.05*A...
                max(theta(:,k)*180/pi)+0.05*A])
        end
    end
end
%SAVE DISPLACEMENTS:
disp(' '),store = lower(input('Save this data (y/n)? ','s'));
if store == 'Y'
    filename = input(' Enter a basic filename (*.CSV): ','s');
    for k = 1:nnodes
        node = num2str(nodes(k));
        filenamek = [filename,' n',node,'.csv'];
        csvwrite(filenamek,[t y(:),k) z(:,k) theta(:,k)])
        if nnodes > 1
            disp([' Data saved as ',filenamek,'...'])
        else
            disp([' Data saved as ',filenamek])
        end
    end
    if nnodes > 1
        disp(' Done saving data')
    end
else
    disp(' No data saved')
end
```


## RotorDispl_TransCSV.m

```
A program that loads and plots transient rotor displacement signals from
CSV files (created by RotorDispl_TransPCH.m).
Note: *Lateral displacements are assumed to be in the yz-plane.
    *User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
```

```
% Created: April }201
% Last revised: June 2014
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%% LOAD DISPLACEMENT DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET CSV FILE INFO:
flag = 1;
while flag
    filename = input('Enter the basic CSV filename...
        [without suffix ''_n<node>''] (*.csv): ','s');
    if strcmp(filename(end),'''') || strcmp(filename(end-3:end),'.csv')
        disp(' Enter filename without extension or ''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data loading cancelled')
            return
        end
    else
        flag = 0;
    end
end
nodes = input('Enter node number(s) [in vector format if > 1]: ');
t_start = input('Enter starting time of signal(s) to be loaded: ');
t_end = input('Enter ending time of signal(s) to be loaded: ');
%LOAD CSV DATA:
nnodes = length(nodes);
kfound = 1;
for k = 1:nnodes
    node = num2str(nodes(k));
    if filename(end) == ''
        filenamek = [filename,'n',node,'.csv'];
    else
        filenamek = [filename,'_n',node,'.cSv'];
    end
    fid = fopen(filenamek,'r');
    if fid == -1
        disp([' File ''',filenamek,''' not found'])
    else
        data = csvread(filenamek);
        t = data(:,1);
        y(:,kfound) = data(:,2); z(:,kfound) = data(:,3); theta(:,kfound)...
            = data(:,4);
            pos(kfound) = k;
            kfound = kfound + 1;
            disp([' Displacement data loaded for node ',node,' (',filenamek,')'])
            fclose(fid)
    end
end
clear data
if not(exist('pos','var'))
    disp(' Error: None of the specified CSV files found')
    disp(' Program terminated')
    return
end
nodes = nodes(pos);
%APPLY TIME LIMITS
index = find(t >= t_start & t <= t_end);
t = t(index); y = y(index,:); z = \overline{z}(index,:); theta = theta(index,:);
%%%%%%%%%%%%%%%%%%%% CALCULATE DISPLACEMENT MAGNITUDES %%%%%%%%%%%%%%%%%%%%
mag = sqrt(y.^2 + z.^2);
displ_max = zeros(nnodes,1); t_max = displ_max; disp(' ')
for k-= 1:nnodes;
    [displ max(k),mpos] = max(mag(:,k)); t_max(k) = t(mpos);
    disp(['Maximum displacement of node ',\overline{num2str(nodes(k)),' = ',...}
        num2str(displ_max(k)),' mm at t = ',num2str(t_max(k)),' s'])
end
```


## RotorDisplSpectral.m

```
A program that performs a spectral analysis on transient rotor
% displacement signals (extracted/loaded by RotorDispl_TransPCH.m or
RotorDispl_TransCSV.m) and plots the results.
Note: *First run RotorDispl_TransPCH.m or RotorDispl_TransCSV.m.
    *Custom functions dft.m & psde.m are required in the working
        directory to run this program.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: February }201
% Last revised: June 2014
%%%%%%%%%%%%%%%%%%%% CHECK IF REQUIRED DATA IS AVAILABLE %%%%%%%%%%%%%%%%%%%%%%
if not(exist('t','var') && exist('y','var') && exist('z','var') &&
exist('theta','var'))
    disp('Error: Required data not detected in memory')
    disp(' First run RotorDispl_TransPCH.m or RotorDispl_TransCSV.m')
    disp(' Program terminated')
    return
end
%%%%%% RESAMPLE SIGNALS AT CONSTANT SAMPLING FREQUENCY (if required) %%%%%%
%CHECK CONSISTENCY OF SAMPLING FREQUENCY:
dt_tol = -5; %::Sampling time consistency check decimal tolerance
dt v = diff(t);
dt_min = min(dt_v);
if dt_min < dt_tol
    disp(' ')
    disp(['Warning: dt_min = ',num2str(dt_min),' < dt_tol = ',num2str(dt_tol)])
end
dt_v = roundn(dt_v,dt_tol);
dt\overline{2}_v= diff(dt_\overline{v});
disp(' ')
if max(abs(dt2 v)) == 0
    dt = dt_v(\overline{1});
    disp(['Sampling frequency = ',num2str(1/dt),' Hz'])
else
    dt_min = min(dt_v);
    di\overline{sp}('Inconsistēnt sampling frequency detected:')
    disp([' Maximum sampling frequency = ',num2str(1/dt_min),' Hz'])
    dt = dt_min;
end
%SPECIFY SAMPLING TIME SCALE FACTOR:
STF = 2; %::Sampling time scale factor (default)
STFinp = input(['Enter a sampling time scale factor (default=',num2str(STF),'):
']);
if not(isempty(STFinp))
    STF = STFinp;
end
%RESAMPLE SIGNALS (if required):
if max(abs(dt2_v)) ~= 0 || STF ~= 1
    dt = STF*d\overline{t};
    disp([' New sampling frequency = ',num2str(1/dt),' Hz'])
    t new = (t(1):dt:t(end))';
    y_new = interp1(t,y,t_new);
    z_new = interp1(t,z,t_new);
    theta_new = interp1(t,theta,t_new);
else
```

```
    t_new = t; y_new = y; z_new = z; theta_new = theta;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DETREND SIGNALS % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % 
fc = 5; %::Detrend lower cutoff frequency [Hz]
fs = 1/dt;
l = length(t new);
dbp = round(白S/fc); nbp = round((1 - 1)/dbp - 1);
n = 1; indx = 1:l; bp = zeros(nbp,1);
for k = dbp:dbp:l-dbp
    bp(n) = indx(k+1);
    n = n + 1;
end
y det = detrend(y new,'linear',bp);
z_det = detrend(z_new,'linear',bp);
theta_det = detrend(theta_new,'linear',bp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SPECTRAL ANALYSIS % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % 
%SPECIFY PARAMETERS FOR SPECTRAL ANALYSES & PLOTTING:
N = 2^(nextpow2(l));
windowl = 512; %::Window length for spectrogram (default)
disp(' ')
window inp = input(['Enter a window length for the Spectrograms
(default=',num2str(windowl),'): ']);
if not(isempty(window inp))
    windowl = window_inp;
end
noverlap = floor(windowl/10)*10;
nfft = windowl;
%SPECIFY FOR WHICH NODE(S) TO CREATE SPECTRAL FIGURES:
if length(nodes) > 1
    disp(' ')
    nspec = lower(input('Only perform spectral analysis on specific node(s)...
                (y/n)? ','s'));
else
    nspec = 'n';
end
if nspec == 'y'
    nodesspec = input(' Enter node number(s) [in vector format if > 1]: ');
    k1 = 1; k2 = 1; clear nodesspecmiss specpos
    for kspec = 1:length(nodesspec)
            compare = find(nodes == nodesspec(kspec));
            if not(isempty(compare))
                specpos(k1) = compare;
                    k1 = k1 + 1;
            else
                    nodesspecmiss(k2) = kspec;
                    k2 = k2 + 1;
            end
        end
        if not(exist('specpos','var'))
            disp(' Error: No displcement data in memory for any of the...
                specified nodes')
            disp(' Program terminated')
            return
        end
        if exist('nodesspecmiss','var')
            disp(' ')
                disp('No displacement data in memory for the following specified nodes:')
            disp(nodesspec(nodesspecmiss))
        end
else
    specpos = 1:length(nodes);
end
%CALCULATE & PLOT SPECTRAL CONTENT:
if not(exist('figstart','var'))
    figstart = fig;
else
```

```
    close(figstart+1:fig)
    fig = figstart;
end
for k = specpos
    node = num2str(nodes(k));
    % Calculate DFTs & PSDs:
    [fy1,Fny] = dft(y_det(:,k),N,fs);
    [fz1,Fnz] = dft(z_det(:,k),N,fs);
    [Py,fy2] = psde(y_det(:,k),N/2,N/4,fs);
    [Pz,fz2] = psde(z_det(:,k),N/2,N/4,fs);
    if incl ang == 'y'
        [ftheta1,Fntheta] = dft(theta_det(:,k),N,fs);
        [Ptheta,ftheta2] = psde(theta_det(:,k),N/2,N/4,fs);
    end
    % Vertical & Horizontal Displacement DFTs:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = semilogy(fy1,abs(Fny));
    title(['DFT of the Vertical Displacement of Node ',node,':'])
    xlabel('f [Hz]'),ylabel('|F_{ny}| [mm]'),grid on
    set(handle,'color',[0 0.7 0.9])
    subplot(2,1,2),handle = semilogy(fz1,abs(Fnz));
    title(['DFT of the Horizontal Displacement of Node ',node,':'])
    xlabel('f [Hz]'),ylabel('|F {nz}| [mm]'),grid on
    set(handle,'color',[0 0.7 0.9])
    % Vertical & Horizontal Displacement PSDs:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = semilogy(fy2,Py);
    title(['PSD of the Vertical Displacement of Node ',node,':'])
    xlabel('f [Hz]'),ylabel('PSD_{y} [mm^{2}/Hz]'),grid on
    set(handle,'color', [0 0.7 0.9])
    subplot(2,1,2),handle = semilogy(fz2,Pz);
    title(['PSD of the Horizontal Displacement of Node ',node,':'])
    xlabel('f [Hz]'),ylabel('PSD {z} [mm^{2}/Hz]'),grid on
    set(handle,'color',[[0 0.7 0.\overline{9}}]
    % Angular Displacement DFT & PSD:
    if incl_ang == 'Y'
        fig = fig + 1; figure(fig)
        subplot(2,1,1),handle = semilogy(ftheta1,abs(Fntheta));
        title(['DFT of the Angular Displacement of Node ',node,':'])
        xlabel('f [Hz]'),ylabel('|F_{n0_{x}}| [deg]'),grid on
        set(handle,'color',[0 0.7 0.9])
        subplot(2,1,2),handle = semilogy(ftheta2,Ptheta);
        title(['PSD of the Angular Displacement of Node ',node,':'])
        xlabel('f [Hz]'),ylabel('PSD_{0_{x}} [deg^{2}/Hz]'),grid on
        set(handle,'color',[0 0.7 0.9])
    end
    %----------------------------------------------------------------------------------
    % Calculate Spectrograms:
    [Sy,Fy,Ty,Py] = spectrogram(y new(:,k),windowl,noverlap,nfft,fs);
    [Sz,Fz,Tz,Pz] = spectrogram(z_new(:,k),windowl,noverlap,nfft,fs);
    Ty = Ty + t_new(1); Tz = Tz + t_new(1);
    if incl_ang-== 'Y'
        [Stheta,Ftheta,Ttheta,Ptheta] =...
            spectrogram(theta_new(:,k),windowl, noverlap,nfft,fs);
        Ttheta = Ttheta + t_new(1);
    end
    % Vertical Displacement Spectrogram:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),surf(Ty,Fy,10*log10(Py),'edgecolor','none')
    axis tight,view (0,90)
    title(['Spectrogram of the Vertical Displacement of Node ',node,':'])
    xlabel('t [s]'),ylabel('f [Hz]')
    subplot(2,1,2),handle = plot(t,y(:,k));
    title(['Vertical Displacement of Node ',node,':'])
```

```
    xlabel('t [s]'),ylabel('y [mm]'),set(handle,'color',[0 0.7 0.9])
    A = max (y(:,k)) - min(y(:,k));
    if A > 0
        axis([Ty(1) Ty(end) min(y(:,k))-0.05*A max(y(:,k))+0.05*A])
    end
    grid on
    % Horizontal Displacement Spectrogram:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),surf(Tz,Fz,10*log10(Pz),'edgecolor','none')
    axis tight,view (0,90)
    title(['Spectrogram of the Horizontal Displacement of Node ',node,':'])
    xlabel('t [s]'),ylabel('f [Hz]')
    subplot(2,1,2),handle = plot(t,z(:,k));
    title(['Horizontal Displacement of Node ',node,':'])
    xlabel('t [s]'),ylabel('z [mm]'),set(handle,'color',[0 0.7 0.9])
    A = max(z(:,k)) - min(z(:,k));
    if A > 0
        axis([Tz(1) Tz(end) min(z(:,k)) -0.05*A max(z(:,k))+0.05*A])
    end
    grid on
    % Angular Displacement Spectrogram:
    if incl_ang == 'Y'
    fig = fig + 1; figure(fig)
    subplot(2,1,1),surf(Ttheta,Ftheta,10*log10(Ptheta),'edgecolor','none')
    axis tight,view (0,90)
    title(['Spectrogram of the Angular Displacement of Node ',node,':'])
    xlabel('t [s]'),ylabel('f [Hz]')
    subplot(2,1,2),handle = plot(t,theta(:,k)*180/pi);
    title(['Angular Displacement of Node ',node,':'])
    xlabel('t [s]'),ylabel('0_{x} [deg]')
    set(handle,'color',[0 0.7 0.9])
    A = max(theta(:,k)*180/pi) - min(theta(:,k)*180/pi);
    if A > 0
        axis([Ttheta(1) Ttheta(end) min(theta(:,k)*180/pi)-0.05*A...
            max(theta(:,k)*180/pi)+0.05*A])
    end
    grid on
end
end
```


## RotorDisplClearance.m

```
% A program that adds clearance lines to transient rotor displacement
figures (created by RotorDispl_TransPCH.m or RotorDispl_TransCSV.m).
Note: *First run RotorDispl_TransPCH.m or RotorDispl_TransCSV.m.
    *Lateral displacements are assumed to be in the yz-plane.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: March 2013
% Last revised: June 2014
%CHECK IF REQUIRED DATA IS AVAILABLE:
if not(exist('t','var') && exist('nodes','var') && exist('fig','var'))
    disp('Error: Required data not detected in memory')
    disp(' First run RotorDispl TransPCH.m or RotorDispl_TransCSV.m')
    disp(' Program terminated')
    return
end
%SPECIFY FOR WHICH NODE(S) TO ADD CLEARANCE LINES:
if length(nodes) > 1
    disp(' '),disp('Specify at which node(s) to add clearance(s):')
    nodesspec = input(' Enter node number(s) [in vector format if > 1]: ');
    k1 = 1; k2 = 1; clear nodesspecmiss pos
    for kspec = 1:length(nodesspec)
        compare = find(nodes == nodesspec(kspec));
```

```
    if not(isempty(compare))
        pos(k1) = compare;
        k1 = k1 + 1;
        else
            nodesspecmiss(k2) = kspec;
            k2 = k2 + 1;
        end
    end
    if not(exist('pos','var'))
        disp(' Error: No displcement data in memory for any of the...
        specified nodes')
        disp(' Program terminated')
        return
    end
    if exist('nodesspecmiss','var')
    disp(' ')
    disp('No displacement data in memory for the following specified nodes:')
    disp(nodesspec(nodesspecmiss))
    end
else
    pos = 1;
end
%SPECIFY CLEARANCE TYPE(S) AND SIZE(S):
rc = zeros(length(pos)); kc = 1;
for k = pos
    node = num2str(nodes(k));
    flag = 1; disp(' ')
    while flag
        ctype{kc} = lower(input(['Specify type of clearance at node ', node,'...
                (radial/flat[top]): '],'s'));
        if strcmp(ctype{kc},'radial') || strcmp(ctype{kc},'flat')
                flag = 0;
            else
                disp([' Error: ''',ctype{kc},''' is not a valid clearance type'])
                disp(' Type: ''radial'' for radial clearance')
                disp(' ''flat'' for flat clearance above rotor (without '''')')
            end
    end
    rc(kc) = input(['Enter the clearance [mm] at node ',node,': ']);
    kc = kc + 1;
end
%ADD CLEARANCE LINES TO FIGURES:
thetac = linspace(0,2*pi,100);
kc = 1;
for k = pos
    node = num2str(nodes(k));
    if not(exist('figstart','var'))
        fgr = (k-1)*fig/length(nodes) + 1;
    else
        fgr = (k-1)*figstart/length(nodes) + 1;
    end
    % Vertical- & Horizontal Displacements:
    tc = [t(1) t(end)];
    xc = rc(kc)*[1 1];
    figure(fgr),subplot(2,1,1)
    hold on
    if strcmp(ctype{kc},'radial')
        plot(tc,xc,tc,-xc,'Color',0.6*[[1 1 1 1])
        ymin = min(min(y(:,k)),-rc(kc));
        legend('Vertical Displacement','Radial Clearance Lines')
    else %if flat
        plot(tc,xc,'Color',0.6*[1 1 1])
        ymin = min(min(y(:,k)),rc(kc));
        legend('Vertical Displacement','Top Flat Clearance Line')
    end
```

```
ymax = max(max(y(:,k)),rc(kc));
A = ymax - ymin; axis([t(1) t(end) ymin-0.05*A ymax+0.05*A])
hold off
if strcmp(ctype{kc},'radial')
    subplot(2,1,2)
    hold on
    plot(tc,xc,tc,-xc,'Color',0.6*[[1 1 1])
    zmin = min(min(z(:,k)),-rc(kc)); zmax = max(max(z(:,k)),rc(kc));
    A = zmax - zmin; axis([t(1) t(end) zmin-0.05*A zmax+0.05*A])
    legend('Horizontal Displacement','Radial Clearance Lines')
    hold off
end
% Displacement Orbit:
figure(fgr+1)
hold on
if strcmp(ctype{kc},'radial')
    zco = rc(kc)*sin(thetac); yco = rc(kc)*cos(thetac);
    lgnd = {'Displacement Orbit','Radial Clearance Circle'};
    if displ max(k) > rc(kc)
        lwidth = 1.5;
    else
        lwidth = 0.5;
    end
else %if flat
    Az = max(z(:,k)) - min(z(:,k));
    zco = [min(z(:,k))-0.05*Az max(z(:,k))+0.05*Az]; yco = xc;
    lgnd = {'Displacement Orbit','Top Flat Clearance Line'};
    lwidth = 0.5
end
plot(zco,yco,'Color',0.6*[1 1 1],'LineWidth',lwidth)
legend(lgnd)
hold off
% Displacement Magnitude:
if strcmp(ctype{kc},'radial')
    figure(fgr+2)
    hold on
    plot(tc,xc,'Color',0.6*[1 1 1])
    rmin = min(min(mag(:,k)),rc(kc)); rmax = max(max(mag(:,k)),rc(kc))
    A = rmax - rmin; axis([t(1) t(end) rmin-0.05*A rmax+0.05*A])
    legend('Displacement','Radial Clearance Line')
    hold off
end
% 2D Displacement:
figure(fgr+3)
hold on
posr = find(mag(:,k) >= rc(kc));
if not(isempty(posr))
    tc1 = t(posr(1))-(t(posr(end))-t(posr(1)))*0.3;
    tc2 = t(posr(end)) +(t(posr(end)) -t(posr(1)))*0.3;
else
    tc1 = t(1); tc2 = t(end);
end
tc_v = [tc1 tc2];
if strcmp(ctype{kc},'radial')
    zc_v = rc(kc)*sin(thetac); yc_v = rc(kc)*cos(thetac);
    yc_m = zeros(100,2);
    for kj = 1:2
        yc_m(:,kj) = yc_v';
    end
    lgnd = {'2D Displacement','Radial Clearance Cylinder'};
else
    zC_v = zco;
    yc_m = rc(kc)*ones(2);
    lgnd = {'2D Displacement','Flat Clearance Surface'};
end
surf(tc_v,zc_v,yc_m,'EdgeColor','none','FaceColor',0.9*[1 1 1])
```

```
    alpha(0.85),legend(lgnd)
    hold off
    kc = kc + 1;
end
```


## RotorDisplSpinSpeed.m

```
A program that converts the time-values on transient displacement figures
(created by RotorDispl TransPCH.m, RotorDispl TransCSV.m and/or
% RotorDisplSpectral.m) to spin speed values. Optionally, whirl frequency
vs. spin speed lines may be added to spectrogram figures.
Note: *First run RotorDispl_TransPCH.m, RotorDispl_TransCSV.m and/or
    RotorDisplSpectral.m.
    *The spin speed should, preferably, either be increasing or
    decreasing with time.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: March 2013
% Last revised: June 2014
%%%%%%%%%%%%%%%%%% CHECK IF REQUIRED DATA IS AVAILABLE %%%%%%%%%%%%%%%%%%%%%%%
if not(exist('t','var') && exist('y','var') && exist('z','var') &&
exist('nodes','var'))
    disp('Error: Required data not detected in memory')
    disp(' First run RotorDispl_TransPCH.m, RotorDispl_TransCSV.m or
RotorDisplSpectral.m')
    disp(' Program terminated')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%% SPECIFY FIGURES TO CONVERT %%%%%%%%%%%%%%%%%%%%%%%%%
if exist('figstart','var')
    flag = 1; disp(' ')
    while flag
            conv = lower(input('Enter type of figures to convert...
                (displ/spec/default:both): ','s'));
            if isempty(conv)
                conv = 'both'; flag = 0;
            elseif strcmp(conv,'displ') || strcmp(conv,'spec') || strcmp(conv,'both')
                flag = 0;
            else
                disp([' Error: ''',conv,''' is not a valid conversion option'])
                disp(' Type: ''displ'' for converting only displacement figures')
                disp(' ''spec'' for converting only spectral figures')
                disp(' ''both'' for converting both displacement &...
                        spectral figures (without '''')')
            end
    end
else
    conv = 'displ';
end
%%%%%%%%%%%%%%%%% DEFINE SPIN SPEED AS A FUNCTION OF TIME %%%%%%%%%%%%%%%%%%%
disp(' ')
SSstr = input('Enter the spin speed as a function of time (t): ','s');
SSstr = ['@(t)' SSstr];
SpinSpeed = str2func(SSstr);
flag = 1;
while flag
    Ntype = lower(input('Specify the spin speed unit (default:rpm/Hz): ','s'));
    if strcmp(Ntype,'rpm') || isempty(Ntype)
        Ntype = 'rpm'; Nlbl = 'N';
        flag = 0;
    elseif strcmp(Ntype,'hz')
        Ntype = 'Hz'; Nlbl = 'f_{spin}';
        flag = 0;
```

```
    else
        disp([' Error: ''',Ntype,''' is not a valid spin speed unit'])
        disp(' Type: ''rpm'' for revolutions per minute')
        disp(' ''Hz'' for Hertz (without '''')')
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONVERT FIGURES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DISPLACEMENT FIGURES:
fignum = 0;
N = SpinSpeed(t);
if strcmp(conv,'displ') || strcmp(conv,'both')
    for k = 1:nnodes
        node = num2str(nodes(k));
        % Vertical- & Horizontal Displacements:
        fignum = fignum + 1; figure(fignum)
        subplot(2,1,1),handle = plot(N,y(:,k));
        title(['Vertical Displacement of Node ',node,':'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('y [mm]'), set(handle,...
            'color',[0 0.7 0.9])
        A = max (y(:,k)) - min(y(:,k));
        if A > 0
            axis([N(1) N(end) min(y(:,k))-0.05*A max(y(:,k))+0.05*A])
        end
        subplot(2,1,2),handle = plot(N,z(:,k));
        title(['Horizontal Displacement of Node ',node,':'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('z [mm]'),set(handle,...
            'color',[0 0.7 0.9])
        A = max(z(:,k)) - min(z(:,k));
        if A > 0
                axis([N(1) N(end) min(z(:,k))-0.05*A max(z(:,k))+0.05*A])
            end
        % Displacement Magnitude:
        fignum = fignum + 2; figure(fignum)
        handle = plot(N,mag(:,k));
        title(['Displacement Magnitude of Node ',node,':'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('r [mm]'), set(handle,...
                'color',[0 0.7 0.9])
        A = max(mag(:,k)) - min(mag(:,k));
        if A > 0
                axis([N(1) N(end) min(mag(:,k)) -0.05*A max(mag(:,k)) +0.05*A])
            end
        % 2D Displacement:
        fignum = fignum + 1; figure(fignum)
        N_scale = 3; %::Spin speed scale factor (N : max displacement)
        handle = plot3(N,z(:,k),y(:,k));
        title(['2D Displacement of Node ',node,' as a Function of Spin Speed:'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('z [mm]'), zlabel('y [mm]')
        set(handle,'color',[0 0.7 0.9])
        A = max(max (y(:,k)) - min(y(:,k)),max(z(:,k)) - min(z(:,k)));
        scale = N_scale*A/(N (end)-N(1));
        if scale > 0
        set(gca,'DataAspectRatio',[1 scale scale])
            end
        % Angular Displacement:
        if incl_ang == 'y'
            fignum = fignum + 1; figure(fignum)
            handle = plot(N,theta(:,k)*180/pi);
            title(['Angular Displacement of Node ',node,':'])
            xlabel([Nlbl,' [',Ntype,']']),ylabel('0_{x} [deg]')
            set(handle,'color',[0 0.7 0.9])
            A = max(theta(:,k)*180/pi) - min(theta(:,k)*180/pi);
            if A > 0
                axis([N(1) N(end) min(theta(:,k)*180/pi)-0.05*A...
                        max(theta(:,k)*180/pi)+0.05*A])
            end
```

```
        end
    end
end
%SPECTRAL FIGURES:
if (strcmp(conv,'spec') || strcmp(conv,'both')) && exist('figstart','var')
    fignum = figstart;
    for k = specpos
        node = num2str(nodes(k));
        % Calculate Spectrograms:
        [Sy,Fy,Ty,Py] = spectrogram(y_new(:,k),windowl,noverlap,nfft,fs);
        [Sz,Fz,Tz,Pz] = spectrogram(z_new(:,k),windowl,noverlap,nfft,fs);
        Ny = SpinSpeed(Ty + t_new(1)); Nz = SpinSpeed(Tz + t_new(1));
        if incl_ang == 'y'
            [Stheta,Ftheta,Ttheta,Ptheta] =...
            spectrogram(theta_new(:,k),windowl,noverlap,nfft,fs);
        Ntheta = SpinSpeed(Ttheta + t_new(1));
        end
        % Vertical Displacement Spectrogram:
        if strcmp(incl_ang,'y')
            fignum = fignum + 4;
        else
        fignum = fignum + 3;
        end
        figure(fignum)
        subplot(2,1,1), surf(Ny, Fy,10*log10(Py),'edgecolor','none')
        axis tight,view(0,90)
        title(['Spectrogram of the Vertical Displacement of Node ',node,':'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('f [Hz]')
        subplot(2,1,2),handle = plot(N,y(:,k));
        title(['Vertical Displacement of Node ',node,':'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('y [mm]')
        set(handle,'color',[0 0.7 0.9])
        A = max(y(:,k)) - min(y(:,k));
        if A > 0
        axis([Ny(1) Ny(end) min(y(:,k))-0.05*A max(y(:,k))+0.05*A])
    end
    grid on
    % Horizontal Displacement Spectrogram:
    fignum = fignum + 1; figure(fignum)
    subplot(2,1,1),surf(Nz,Fz,10*log10(Pz),'edgecolor','none')
    axis tight,view(0,90)
    title(['Spectrogram of the Horizontal Displacement of Node ',node,':'])
    xlabel([Nlbl,' [',Ntype,']']),ylabel('f [Hz]')
    subplot(2,1,2), handle = plot(N,z(:,k));
    title(['Horizontal Displacement of Node ',node,':'])
    xlabel([Nlbl,' [',Ntype,']']),ylabel('z [mm]')
    set(handle,'color',[0 0.7 0.9])
    A = max(z(:,k)) - min(z(:,k));
    if A > 0
        axis([Nz(1) Nz(end) min(z(:,k)) -0.05*A max(z(:,k))+0.05*A])
    end
    grid on
    % Angular Displacement Spectrogram:
    if incl_ang == 'y'
        fignum = fignum + 1; figure(fignum)
        subplot (2,1,1),surf(Ntheta,Ftheta,10*log10(Ptheta),'edgecolor','none')
        axis tight,view(0,90)
        title(['Spectrogram of the Angular Displacement of Node ',node,':'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('f [Hz]')
        subplot(2,1,2),handle = plot(N,theta(:,k)*180/pi);
        title(['Angular Displacement of Node ',node,':'])
        xlabel([Nlbl,' [',Ntype,']']),ylabel('0_{x} [deg]')
        set(handle,'color',[0 0.7 0.9])
        A = max(theta(:,k)*180/pi) - min(theta(:,k)*180/pi);
        if A > 0
```

```
                axis([Ntheta(1) Ntheta(end) min(theta(:,k)*180/pi)-0.05*A...
                        max(theta(:,k)*180/pi)+0.05*A])
            end
        grid on
    end
end
%DEFINE & PLOT WHIRL FREQUENCY VS. SPIN SPEED LINES (optional):
% Define Lines:
disp(' ')
WFvsSS = lower(input('Add whirl frequency (WF) vs. spin speed (SS) lines...
    to figures (y/n)? ','s'));
if WFvsSS == 'y'
    nWFSS = input(' Number of lines to add: ');
    SS = [Ny(1) Ny(end)]';
    WFSSfact = zeros(nWFSS,1); WF = zeros(2,nWFSS);
    for WFSSk = 1:nWFSS
        WFSSfact(WFSSk) = input([' Line ',num2str(WFSSk),' WF:SS factor: ']);
        if strcmp(Ntype,'rpm')
            WF(:,WFSSk) = WFSSfact(WFSSk)*SS/60;
        else
            WF(:,WFSSk) = WFSSfact (WFSSk)*SS;
        end
    end
    %Add Lines to Spectrogram Figures:
    fignum = figstart;
    for k = specpos
        if incl_ang == 'y'
            fign̄um = fignum + 4;
        else
            fignum = fignum + 3;
        end
        figure(fignum)
        subplot(2,1,1), hold on
        for WFSSk = 1:nWFSS
            plot3(SS,WF(:,WFSSk),displ_max(k)*[llll','k')
        end
        hold off
        fignum = fignum + 1; figure(fignum)
        subplot (2,1,1), hold on
        for WFSSk = 1:nWFSS
            plot3(SS,WF(:,WFSSk),displ_max(k)*[1 1]','k')
        end
        hold off
        if incl_ang == 'Y'
            fignum = fignum + 1; figure(fignum)
            subplot(2,1,1),hold on
            for WFSSk = 1:nWFSS
                    plot3(SS,WF(:,WFSSk),displ_max(k)*[1 1]','k')
            end
            hold off
        end %if
    end %for
    disp(' Note: Place a Data Cursor on each figure to see these lines')
end %if
```

end \%if

## RotorForce1DElem_TransPCH.m

```
A program that extracts, plots and saves (optional) transient spring-/
damper element force results from an MSC.Nastran(TM) PCH output file.
Note: *Lateral forces must be in the yz-plane.
    *The PCH file may contain the force results from multiple
        elements and may also contain other transient result types.
    *Results from multiple subcases are not supported.
    *User-modifiable parameters/code indicated by :: in comments.
```

```
% Author: Armand Krüger
% Created: August 2013
% Last revised: November 2013
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EXTRACT FORCE DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET PCH FILE- & EXTRACTION INFO:
flag = 1;
while flag
    filename = input('Enter the PCH filename (*.pch): ','s');
    filename = [filename '.pch'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or ''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data extraction cancelled')
            return
        end
    else
        flag = 0;
    end
end
disp(' ')
t_start = input('Enter starting time of signal(s) to be extracted: ');
t_end = input('Enter ending time of signal(s) to be extracted: '); disp(' ')
espec = lower(input('Only extract forces for a specific element(s) (y/n)? ','s'));
if espec == 'y'
    elems = input(' Enter element ID(s) [in vector format if > 1]: ');
else
    espec = 'n';
    disp(' Automatic element force extraction enabled')
    disp(' Press any key to continue... '),pause
end
%SEARCH FOR ELEMENT FORCE DATA IN PCH FILE:
SEIDspec = 0; %:ID of superelement data to extract (if applicable)
disp(' '),disp('Searching for element force data...')
eof = 0; found = 0; SEfound = 0;
while found == 0
    line = fscanf(fid,'%c',[15,1])'; dummy = fgetl(fid);
    if strcmp(line,'$ELEMENT FORCES') %if element force data found
        dummy = fscanf(fid,'%c',[25,1])'; SE = fscanf(fid,'%c',[5,1])';
        SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
        if strcmp(SE,'SEID=') %if superelements detected
            SEfound = 1;
            if SEID == SEIDspec
                found = 2;
            end
        else
            found = 1;
        end
    end
    if feof(fid) %if end of file reached
        if SEfound
                        disp([' Error: No element force data found for SE ',...
                    num2str(SEIDspec),' in file ',filename])
        else
            disp([' Error: No element force data found in file ',filename])
                end
                disp(' Program terminated'),fclose(fid);
                return
    end
end
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[29,1])';
elemtype = fscanf(fid,'%c',[5,1])'; dummy = fgetl(fid);
elemtype1 = elemtype;
```

```
%EXTRACT DATA FROM PCH FILE:
% Loop throught time steps:
n = 1; t_n = 0; datanext = 0;
while t n < t end && datanext == 0
    dum\overline{y}y= fs=scanf(fid,'%c',[7,1])';
    t n = fscanf(fid,'%f \n',[1,1]);
    t(n) = t_n; dummy = fgetl(fid);
    % Loop through element types:
    ktype = 1; tnext = 0;
    k = 1; kspec = 0; kunsupp = 0; %k: elements in file; kspec: spec....
    elements found
    while tnext == 0
        % Loop through elements:
        next = ' ';
        while eof == 0 && not(strcmp(next,'$'))
            if strcmp(elemtype,'ELAS1') || strcmp(elemtype,'DAMP1')
                elem = fscanf(fid,'%d',[1,1]);
                if n == 1 && espec ~= 'y' %if auto element extraction enabled...
                                    & @ t1
                elems(k) = elem; %store element ID
                elemtypes{k} = elemtype(1:end); %store element type
            end
            dummy = fscanf(fid,'%c',[13,1])';
            forcetemp = fscanf(fid,'%f \n',[1,1])';
            if espec == 'y' %if elements specified by user
                compare = find(elems == elem);
                if not(isempty(compare)) %if element found
                        kspec = kspec + 1;
                        force(n,kspec) = forcetemp;
                        pos(kspec) = compare; %store element position
                            elemtypes{kspec} = elemtype(1:end); %store element type
                end
            else %if auto element extraction enabled
                force(n,k) = forcetemp;
            end
                dummy = fgetl(fid);
                next = fscanf(fid,'%c',[1,1]);
                eof = feof(fid); %check for end of file
                k = k + 1;
            else %if not ELAS1 or DAMP1 element
                dummy_unsupp = fgetl(fid);
                next = fscanf(fid,'%c',[1,1]);
                eof = feof(fid); %check for end of file
            end
        end %elements loop end
        if exist('dummy_unsupp','var') && n == 1 %if unsupported element detected
            kunsupp = kunsupp + 1;
            elems_unsupp{kunsupp} = elemtype;
        end
        if eof %if end of file reached in elements loop
            break
        end
        dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid);
        datatype = fscanf(fid,'%c',[15,1])'; dummy = fgetl(fid);
        if not(strcmp(datatype,'$ELEMENT FORCES')) %if different data type found
            datanext = 1; break
        else
            if SEfound
                dummy = fscanf(fid,'%c',[30,1])';
                SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
                        if SEID ~= SEIDspec %if different superelement found
                datanext = 1; break
                    end
            else
                        dummy = fgetl(fid);
            end
            dummy = fgetl(fid);
```

```
            dummy = fscanf(fid,'%c',[29,1])';
            elemtype = fscanf(fid,'%c',[5,1])'; dummy = fgetl(fid);
            if strcmp(elemtype,elemtype1) %if 1st element type is repeated
                tnext = 1;
            else
            dummy = fgetl(fid);
            end
        end
        ktype = ktype + 1;
    end %element types loop end
    disp(['Forces read for t_',num2str(n),' = ', num2str(t_n),' s'])
    if eof %if end of file reached in elements loop
        break
    end
    if espec == 'y' && kspec == 0 %if no specified element(s) found
        disp([' Error: Element force data not found/supported for...
            specified element(s) in file ',filename])
        disp(' Program terminated'),fclose(fid);
        return
    end
    n = n + 1;
end %time steps loop end
if exist('elems_unsupp','var')
    disp(' '),disp('Note: Unsupported element type(s) detected:')
    disp(elems_unsupp')
end
if not(exist('elems','var'))
    disp(['Error: No supported element force data found in file ',filename])
    disp(' Program terminated'),fclose(fid);
    return
end
elemspos = 1:length(elems);
if espec == 'y' %if elements specified by user
    elemsmiss = setxor(elemspos,pos);
    if not(isempty(elemsmiss)) %if certain spec. element(s) not found
            disp(' ')
            disp('Note: Element force data not found for the following...
                    specified elements:')
            disp(elems(elemsmiss))
    end
    elems = elems(pos);
end
fclose(fid);
%EXTRACT SPECIFIC FORCE SIGNALS:
t = t';
index = find(t >= t_start);
t = t(index);
F = force(index,:);
clear force
%%%%%%%%%%%%%%%%%%%%%% PAIR FORCE SIGNALS (optional) % % % % % % % % % % % % % % % % % % % % 
%GET SIGNAL PAIRING INFO:
disp(' '),disp('Signals extracted (element ID & type):')
for k = 1:length(elems)
    disp([' ',num2str(elems(k)),': ',elemtypes{k}])
end
if length(elems) > 1
    even = 0;
    while even == 0
    disp('Enter IDs of element signals to be paired (optional)')
    disp(' (in vector format: [Fy1 Fz1 Fy2 Fz2...]): ')
    pairs = input(' ');
    if isempty(pairs)
                even = 1;
            elseif not(mod(length(pairs),2))
                even = 1;
            else
                disp(' Error: enter an even number of vector elements or an...
```

```
                    empty vector')
                disp(' Press any key to continue...'),pause
        end
    end
    if isempty(pairs)
        singles = elems; singlespos = elemspos;
    else
        [singles,singlespos,~] = setxor(elems,pairs);
        [pairs,pairspos,~] = setxor(elems,singles);
    end
else
    pairs = []; singles = elems; singlespos = 1;
end
%SORT SIGNALS ACCORDING TO PAIRING INFO:
if not(isempty(pairs))
    Fp = F(:,pairspos);
    npairs = length(pairs)/2;
    Fy = Fp(:,1:2:2*npairs-1); Fz = Fp(:,2:2:2*npairs);
end
if not(isempty(singles))
    Fs = F(:,singlespos);
    nsingles = length(singles);
end
%%%%%%%%%%%%%%%%%%%%%%%%% GET ELEMENT LABEL INFO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' '),lspec = lower(input('Specify label(s) for element(s) (y/n)? ','s'));
if not(isempty(pairs))
    for k = 1:npairs
        if lspec == 'y'
            lblp{k} = input([' Enter label for element pair ',...
                    num2str(pairs(2*k-1)),...
                    '& ',num2str(pairs(2*k)),': '],'s');
            else
                lblp{k} = ['Element Pair ',num2str(pairs(2*k-1)),...
                    ' & ',num2str(pairs(2*k))];
            end
    end
end
if not(isempty(singles))
    for k = 1:nsingles
        if lspec == 'y'
                lbls{k} = input([' Enter label for element ',num2str(singles(k)),...
                    ': '],'s');
                untspec = lower(input([' Element ',num2str(singles(k)),...
                    ': lateral or torsional (l/t)? '],'s'));
            else
                lbls{k} = ['Element ',num2str(singles(k))];
                untspec = 'l';
            end
            if untspec == 't'
                        unts{k} = 'M [N.mm]'; lbls{k} = [lbls{k} ' Torque'];
            else
                        unts{k} = 'F [N]'; lbls{k} = [lbls{k} ' Force'];
            end
        end
end
%%%%%%%%%%%%%% CALCULATE PAIRED FORCE MAGNITUDES (if needed) % % % % % % % % % % % %
if not(isempty(pairs))
    Fmag = sqrt(Fy.^2 + Fz.^2);
    F_max = zeros(npairs,1); t_max = F_max; disp(' ')
    for k = 1:npairs
            [F_max (k),mpos] = max(Fmag(:,k)); t_max(k) = t(mpos);
            di\overline{sp}(['Maximum ',lower(lblp{k}),' fōrce = ',num2str(F_max(k)),...
            ' N at t = ',num2str(t_max(k)),' s'])
    end
end
```

```
%%%%
%PLOT PAIRED FORCES:
fig = 0;
if not(isempty(pairs))
    for k = 1:npairs
        % Vertical- & Horizontal Forces:
        fig = fig + 1; figure(fig)
        subplot (2,1,1),handle = plot(t,Fy(:,k));
        title(['Vertical ',lblp{k},' Force:'])
        xlabel('t [s]'),ylabel('F_{y} [N]'), set(handle,'color', [0 0.7 0.9])
        A = max (Fy(:,k)) - min(Fy(:,k));
        if A > 0
                axis([t(1) t(end) min(Fy(:,k)) -0.05*A max(Fy(:,k))+0.05*A])
            end
            subplot (2,1,2),handle = plot(t,Fz(:,k));
            title(['Horizontal ',lblp{k},' Force:'])
            xlabel('t [s]'),ylabel('F_{z} [N]'), set(handle,'color', [0 0.7 0.9])
            A = max (Fz(:,k)) - min(Fz(:,k));
            if A > 0
                axis([t(1) t(end) min(Fz(:,k))-0.05*A max(Fz(:,k))+0.05*A])
            end
            % Force Magnitude:
            fig = fig + 1; figure(fig)
            handle = plot(t,Fmag(:,k));
            title([lblp{k},' Force Magnitude:'])
            xlabel('t [s]'),ylabel('F [N]'), set(handle,'color', [0 0.7 0.9])
            A = max (Fmag(:,k)) - min(Fmag(:,k));
            if A > 0
                axis([t(1) t(end) min(Emag(:,k))-0.05*A max(Fmag(:,k))+0.05*A])
            end
            % 2D Force:
            fig = fig + 1; figure(fig)
            t_scale = 3; %::Time scale factor (t : force)
            handle = plot3(t,Fz(:,k),Fy(:,k));
            title([lblp{k},' 2D Force as a Function of Time:'])
            xlabel('t [s]'),ylabel('F_{z} [N]'),zlabel('F_{y} [N]')
            set(handle,'color', [0 0.7 0.9])
            A = max (max(Fy(:,k)) - min(Fy(:,k)),max(Fz(:,k)) - min(Fz(:,k)));
            scale = t_scale*A/(t (end)-t(1));
            if scale > 0
                set(gca,'DataAspectRatio',[1 scale scale])
            end
    end
end
%PLOT SINGLE FORCES:
if not(isempty(singles))
    for k = 1:nsingles
            fig = fig + 1; figure(fig)
            handle = plot(t,Fs(:,k)); title(lbls{k})
            xlabel('t [s]'),ylabel(unts{k}), set(handle,'color', [0 0.7 0.9])
            A = max(Fs(:,k)) - min(Fs(:,k));
            if A > 0
                axis([t(1) t(end) min(Fs(:,k)) -0.05*A max(Fs(:,k))+0.05*A])
            end
        end
end
%SAVE PAIRED FORCES:
if not(isempty(pairs))
    disp(' '),store = lower(input('Save the force pairs data (y/n)? ','s'));
    if store == 'y'
            filename = input(' Enter a basic filename (*.CSV): ','s');
            for k = 1:npairs
                    pair = [num2str(pairs(2*k-1)),' ',num2str(pairs(2*k))];
                    filenamek = [filename,' e',pair,'.CSv'];
                    csvwrite(filenamek,[t Fy(:,k) Fz(:,k)])
                    if npairs > 1
```

```
                disp([' Data saved as ',filenamek,'...'])
                else
                    disp([' Data saved as ',filenamek])
            end
        end
        if npairs > 1
            disp(' Done saving data')
        end
    else
    disp(' No data saved')
    end
end
```


## RotorForceExc_TransPCH.m

```
% A program that extracts, plots and saves (optional) transient rotor
excitation forces from an MSC.Nastran(TM) PCH output file.
Note: *Lateral forces must be in the yz-plane.
    *The PCH file may contain the excitation forces at multiple grid
    points/nodes and may also contain other transient result types.
    *Results from multiple subcases are not supported.
    *User-modifiable parameters/code indicated by :: in comments.
Author: Armand Krüger
% Created: August 2013
% Last revised: July }201
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%% EXTRACT FORCE DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET PCH FILE- & EXTRACTION INFO:
flag = 1;
while flag
    filename = input('Enter the PCH filename (*.pch): ','s');
    filename = [filename '.pch'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or ''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data extraction cancelled')
            return
        end
    else
        flag = 0;
    end
end
disp(' ')
t_start = input('Enter starting time of signal(s) to be extracted: ');
t_end = input('Enter ending time of signal(s) to be extracted: '); disp(' ')
nspec = lower(input('Only extract excitation force(s) at a specific node(s)...
    (y/n)? ','s'));
if nspec == 'Y'
    nodes = input(' Enter node number(s) [in vector format if > 1]: ');
else
    nspec = 'n';
    disp(' Automatic node force extraction enabled')
    disp(' Press any key to continue... '), pause
end
%SEARCH FOR EXCITATION FORCE DATA IN PCH FILE:
SEIDspec = 1; %::ID Of superelement data to extract (if applicable)
disp(' '),disp('Searching for excitation force data...')
eof = 0; found = 0; SEfound = 0;
while found == 0
    line = fscanf(fid,'%c',[7,1])'; dummy = fgetl(fid);
    if strcmp(line,'$OLOADS') %if force data found
        dummy = fscanf(fid,'%c',[25,1])'; SE = fscanf(fid,'%c',[5,1])';
```

```
        SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
        if strcmp(SE,'SEID=') %if superelements detected
            SEfound = 1;
            if SEID == SEIDspec
                found = 2;
            end
        else
        found = 1;
        end
    end
    if feof(fid) %if end of file reached
        if SEfound
                disp([' Error: No excitation force data found for SE ',...
                num2str(SEIDspec),' in file ',filename])
        else
            disp([' Error: No excitation force data found in file ',filename])
        end
        disp(' Program terminated'),fclose(fid);
        return
    end
end
%EXTRACT DATA FROM PCH FILE:
% Loop throught time steps:
n = 1; t n = 0; datanext = 0;
while t_\overline{n}< t_end && datanext == 0
    dummy = fgetl(fid);
    dummy = fscanf(fid,'%c',[7,1])';
    t_n = fscanf(fid,'%f \n',[1,1]);
    t(n) = t_n; dummy = fgetl(fid);
    % Loop through nodes:
    k = 1; kspec = 0; %k: nodes in file; kspec: specified nodes found
    next = ' ';
    while eof == 0 && not(strcmp(next,'$'))
        node = fscanf(fid,'%d',[1,1]);
        if n == 1 && nspec ~= 'y' %if auto node extraction enabled & @ t1
            nodes(k) = node; %store node found
        end
        dummy = fscanf(fid,'%c',[10,1])';
        forcetemp = fscanf(fid,'%f \n',[3,1])'; dummy = fgetl(fid);
        dummy = fscanf(fid,'%c',[20,1])';
        forcetemp(1,4:6) = fscanf(fid,'%f \n',[3,1])';
        if nspec == 'y' %if nodes specified by user
            compare = find(nodes == node);
            if not(isempty(compare)) %if node found
                kspec = kspec + 1;
                force(n,:,kspec) = forcetemp;
                pos(kspec) = compare; %store node position
            end
        else %if auto node extraction enabled
            force(n,:,k) = forcetemp;
        end
        dummy = fgetl(fid);
        next = fscanf(fid,'%c',[1,1]);
        eof = feof(fid); %check for end of file
        k = k + 1;
    end %nodes loop end
    disp(['Forces read for t_',num2str(n),' = ',num2str(t_n),' s'])
    if eof %if end of file reached in nodes loop
            break
    end
    if nspec == 'Y' && kspec == 0 %if no specified node(s) found
            disp([' Error: Excitation force data not found for specified node(s)...
                in file ',filename])
            disp(' Program terminated'),fclose(fid);
            return
    end
    dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid);
    datatype = fscanf(fid,'%c',[7,1])'; dummy = fgetl(fid);
```

```
    if not(strcmp(datatype,'$OLOADS')) %if different data type found
        datanext = 1;
    else
        if SEfound
                dummy = fscanf(fid,'%c',[30,1])';
                SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
                if SEID ~= SEIDspec %if different superelement found
                    datanext = 1;
            end
        else
            dummy = fgetl(fid);
        end
    end
    n = n + 1;
end %time steps loop end
if nspec == 'y' %if nodes specified by user
    nodespos = 1:length(nodes);
    nodesmiss = setxor(nodespos,pos);
    if not(isempty(nodesmiss)) %if certain spec. node(s) not found
        disp(' ')
        disp('Note: Excitation force data not found for the following...
            specified nodes:')
        disp(nodes(nodesmiss))
    end
    nodes = nodes(pos);
end
fclose(fid);
%EXTRACT SPECIFIC FORCE SIGNALS:
t = t';
index = find(t >= t_start);
t = t(index);
Fy = squeeze(force(index,2,:));
Fz = squeeze(force(index,3,:));
Mx = squeeze(force(index,4,:));
clear force
%%%%%%%%%%%%%%%%%%%%%%%%% CALCULATE FORCE MAGNITUDES %%%%%%%%%%%%%%%%%%%%%%%%%%
Fmag = sqrt(Fy.^2 + Fz.^2);
nnodes = length(nodes);
F_max = zeros(nnodes,1); t_max = F_max; disp(' ')
for k = 1:nnodes;
    [F max(k),mpos] = max(Fmag(:,k)); t max(k) = t(mpos);
    disp(['Maximum excitation force at node ',num2str(nodes(k)),' = ',...
        num2str(F_max(k)),' N at t = ',num2str(t_max(k)),' s'])
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT & SAVE FORCES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PLOT FORCES:
disp(' ')
incl_ang = lower(input('Include moment figures (y/n)? ','s'));
fig = 0;
for k = 1:nnodes
    node = num2str(nodes(k));
    % Vertical- & Horizontal Forces:
    fig = fig + 1; figure(fig)
    subplot(2,1,1),handle = plot(t,Fy(:,k));
    title(['Vertical Excitation Force at Node ',node,':'])
    xlabel('t [s]'),ylabel('F_{y} [N]'),set(handle,'color',[0 0.7 0.9])
    A = max(Fy(:,k)) - min(Fy(:,k));
    if A > 0
        axis([t(1) t(end) min(Fy(:,k))-0.05*A max(Fy(:,k))+0.05*A])
    end
    subplot(2,1,2),handle = plot(t,Fz(:,k));
    title(['Horizontal Excitation Force at Node ',node,':'])
    xlabel('t [s]'),ylabel('F_{z} [N]'),set(handle,'color',[0 0.7 0.9])
    A = max(Fz(:,k)) - min(Fz(:,k));
    if A > 0
        axis([t(1) t(end) min(Fz(:,k)) -0.05*A max(Fz(:,k))+0.05*A])
```

```
    end
    % Force Magnitude:
    fig = fig + 1; figure(fig)
    handle = plot(t,Fmag(:,k));
    title(['Excitation Force Magnitude at Node ',node,':'])
    xlabel('t [s]'),ylabel('F [N]'), set(handle,'color',[0 0.7 0.9])
    A = max(Fmag(:,k)) - min(Fmag(:,k));
    if A > 0
        axis([t(1) t(end) min(Fmag(:,k)) -0.05*A max(Fmag(:,k)) +0.05*A])
    end
    % 2D Force:
    fig = fig + 1; figure(fig)
    t_scale = 5; %::Time scale factor (t : force)
    handle = plot3(t,Fz(:,k),Fy(:,k));
    title(['2D Excitation Force at Node ',node,' as a Function of Time:'])
    xlabel('t [s]'),ylabel('F_{z} [N]'), zlabel('F_{y} [N]')
    set(handle,'color', [0 0.7-0.9])
    A = max(max(Fy(:,k)) - min(Fy(:,k)),max(Fz(:,k)) - min(Fz(:,k)));
    scale = t_scale*A/(t(end)-t(1));
    if scale > 0
        set(gca,'DataAspectRatio',[1 scale scale])
    end
    % Axial Moment:
    if incl_ang == 'Y'
        fig}\mp@subsup{}{}{-}=\textrm{fig}+1; figure(fig
        handle = plot(t,Mx(:,k));
        title(['Axial Excitation Moment at Node ',node,':'])
        xlabel('t [s]'),ylabel('M_{x} [N.mm]')
        set(handle,'color',[0 0.7 0.9])
        A = max (Mx (:,k)) - min (Mx(:,k));
        if A > 0
            axis([t(1) t(end) min(Mx(:,k)) -0.05*A max(Mx(:,k))+0.05*A])
        end
    end
end
%SAVE FORCES:
disp(' '),store = lower(input('Save this data (y/n)? ','s'));
if store == 'y'
    filename = input(' Enter a basic filename (*.csv): ','s');
    for k = 1:nnodes
        node = num2str(nodes(k));
        filenamek = [filename,'_n',node,'.csv'];
        csvwrite(filenamek,[t Fy(:,k) Fz(:,k) Mx(:,k)])
        if nnodes > 1
            disp([' Data saved as ',filenamek,'...'])
        else
            disp([' Data saved as ',filenamek])
        end
    end
    if nnodes > 1
            disp(' Done saving data')
    end
else
    disp(' No data saved')
end
```


## RotorForceNL_TransPCH.m

```
A program that extracts, plots and saves (optional) nonlinear transient
rotor force results from an MSC.Nastran(TM) PCH output file.
Note: *Lateral forces must be in the yz-plane.
    *The PCH file may contain the nonlinear forces at multiple grid
        points/nodes and may also contain other transient result types.
    *Results from multiple subcases are not supported.
    *User-modifiable parameters/code indicated by :: in comments.
```

```
% Author: Armand Krüger
% Created: August 2013
% Last revised: July 2014
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EXTRACT FORCE DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET PCH FILE- & EXTRACTION INFO:
flag = 1;
while flag
    filename = input('Enter the PCH filename (*.pch): ','s');
    filename = [filename '.pch'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or '''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Data extraction cancelled')
            return
        end
    else
        flag = 0;
    end
end
disp(' ')
t_start = input('Enter starting time of signal(s) to be extracted: ');
t_end = input('Enter ending time of signal(s) to be extracted: '); disp(' ')
nspec = lower(input('Only extract nonlinear force(s) at a specific node(s)...
        (y/n)? ','s'));
if nspec == 'y'
    nodes = input(' Enter node number(s) [in vector format if > 1]: ');
else
    nspec = 'n';
    disp(' Automatic node force extraction enabled')
    disp(' Press any key to continue... '),pause
end
%SEARCH FOR NONLINEAR FORCE DATA IN PCH FILE:
SEIDspec = 0; %::ID of superelement data to extract (if applicable)
disp(' '),disp('Searching for nonlinear force data...')
eof = 0; found = 0; SEfound = 0;
while eof == 0 && found == 0
    line = fscanf(fid,'%c',[18,1])'; dummy = fgetl(fid);
    if strcmp(line,'$NON-LINEAR-FORCES') %if force data found
        dummy = fscanf(fid,'%c',[25,1])'; SE = fscanf(fid,'%c',[5,1])';
        SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
        if strcmp(SE,'SEID=') %if superelements detected
            SEfound = 1;
            if SEID == SEIDspec
                        found = 2;
            end
        else
            found = 1;
        end
    end
    if feof(fid) %if end of file reached
        if SEfound
            disp([' Error: No nonlinear force data found for SE ',...
                num2str(SEIDspec),' in file ',filename])
        else
            disp([' Error: No nonlinear force data found in file ',filename])
        end
        disp(' Program terminated'),fclose(fid);
        return
    end
end
%EXTRACT DATA FROM PCH FILE:
```

```
% Loop through nodes:
k = 1; kspec = 0; %k: nodes in file; kspec: specified nodes found
datanext = 0;
while datanext == 0
    dummy = fgetl(fid);
    dummy = fscanf(fid,'%c',[15,1])';
    node = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
    if nspec == 'y' %if node(s) specified by user: search next specified node
        nodefound = 0; ksearch = 1;
        while eof == 0 && nodefound == 0
            if ksearch == 1
                disp(' '),disp('Searching for next specified node (if any)...')
                disp(' ')
        end
        compare = find(nodes == node);
        if not(isempty(compare)) %if node found
                kspec = kspec + 1;
                pos(kspec) = compare;
                nodefound = 1;
        else
                line = fscanf(fid,'%c',[18,1])'; dummy = fgetl(fid);
                if strcmp(line,'$NON-LINEAR-FORCES') %if force data found
                        if SEfound
                        dummy = fscanf(fid,'%c',[30,1])';
                        SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
                        if SEID ~= SEIDspec %if different superelement found
                        datanext = 1; eof = 1;
                        break
                                end
                else
                        dummy = fgetl(fid);
                        end
                        dummy = fgetl(fid);
                        dummy = fscanf(fid,'%c',[20,1])';
                        node = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
                end
                eof = feof(fid); %check for end of file
            end %if node found end
            ksearch = ksearch + 1;
        end %while node not found end
    else %if auto node extraction enabled
        nodes(k) = node; %store node found
    end
    % Loop through time steps:
    n = 1; next = ' '; t n = 0;
    while eof == 0 && not}(strcmp(next,'$')) && t_n < t_end
        t_n = fscanf(fid,'%f \n',[1,1]);
        t(n) = t n;
        dummy = fscanf(fid,'%c',[7,1])';
        forcetemp = fscanf(fid,'%f \n',[3,1])'; dummy = fgetl(fid);
        dummy = fscanf(fid,'%c',[20,1])';
        forcetemp(1,4:6) = fscanf(fid,'%f \n',[3,1])';
        if nspec == 'y' %if nodes specified by user
            force(n,:,kspec) = forcetemp;
        else %if auto node extraction enabled
            force(n,:,k) = forcetemp;
        end
        disp(['Forces read for node ',num2str(node),' - t_',num2str(n),...
                ' = ',num2str(t n),' s'])
        dummy = fgetl(fid);
        next = fscanf(fid,'%c',[1,1]);
        eof = feof(fid); %check for end of file
        n = n + 1;
    end %time steps loop end
    if eof %if end of file reached in time steps loop or during node search
        break
    end
    if kspec >= length(nodes)
```

```
        eof = 1; break
    end
    if t n >= t end
        disp(' '),disp('t_end reached - searching for next node (if any)...')
        while eof == 0 && not(strcmp(next,'$'))
            dummy = fgetl(fid);
            next = fscanf(fid,'%c',[1,1]);
            eof = feof(fid); %check for end of file
        end
    end
    dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid);
    datatype = fscanf(fid,'%c',[18,1])'; dummy = fgetl(fid);
        if not(strcmp(datatype,'$NON-LINEAR-FORCES')) %if different data type found
        datanext = 1;
    else
        if SEfound
            dummy = fscanf(fid,'%c',[30,1])';
            SEID = fscanf(fid,'%d',[1,1]); dummy = fgetl(fid);
            if SEID ~= SEIDspec %if different superelement found
                datanext = 1;
            end
        else
            dummy = fgetl(fid);
        end
    end
    k = k + 1;
end %nodes loop end
if nspec == 'y' && kspec == 0 %if no specified node(s) found
    disp(['Error: Nonlinear force data not found for specified node(s) in...
        file ',filename])
    disp(' Program terminated'),fclose(fid);
    return
end
if nspec == 'y' %if nodes specified by user
    nodespos = 1:length(nodes);
    nodesmiss = setxor(nodespos,pos);
    if not(isempty(nodesmiss)) %if certain spec. node(s) not found
        disp('Note: Nonlinear force data not found for the following...
            specified nodes:')
        disp(nodes(nodesmiss))
    end
    nodes = nodes(pos);
end
fclose(fid);
%EXTRACT SPECIFIC FORCE SIGNALS:
t = t';
index = find(t >= t_start);
t = t(index);
Fy = squeeze(force(index,2,:));
Fz = squeeze(force(index,3,:));
Mx = squeeze(force(index,4,:));
clear force
%%%%%%%%%%%%%%%%%%%%%%%% CALCULATE FORCE MAGNITUDES %%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fmag = sqrt(Fy.^2 + Fz.^2);
nnodes = length(nodes);
F_max = zeros(nnodes,1); t_max = F_max; disp(' ')
for k = 1:nnodes;
    [F_max(k),mpos] = max(Fmag(:,k)); t_max(k) = t(mpos);
    di\overline{sp}(['Maximum excitation force at node ',num2str(nodes(k)),' = ',...
        num2str(F_max(k)),' N at t = ', num2str(t_max(k)),' s'])
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT & SAVE FORCES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PLOT FORCES:
disp(' ')
incl_ang = lower(input('Include moment figures (y/n)? ','s'));
fig = 0;
for k = 1:nnodes
```

```
    node = num2str(nodes(k));
    % Vertical- & Horizontal Forces:
fig = fig + 1; figure(fig)
subplot(2,1,1),handle = plot(t,Fy(:,k));
title(['Vertical Nonlinear Force at Node ',node,':'])
xlabel('t [s]'),ylabel('F_{y} [N]'), set(handle,'color',[0 0.7 0.9])
A = max (Fy(:,k)) - min(Fy(:,k));
if A > 0
    axis([t(1) t(end) min(Fy(:,k)) -0.05*A max(Fy(:,k))+0.05*A])
end
subplot(2,1,2),handle = plot(t,Fz(:,k));
title(['Horizontal Nonlinear Force at Node ',node,':'])
xlabel('t [s]'),ylabel('F {z} [N]'), set(handle,'color',[0 0.7 0.9])
A = max(Fz(:,k)) - min(Fz(:, k));
if A > 0
    axis([t(1) t(end) min(Fz(:,k))-0.05*A max(Fz(:,k))+0.05*A])
end
% Force Magnitude:
fig = fig + 1; figure(fig)
handle = plot(t,Fmag(:,k));
title(['Nonlinear Force Magnitude at Node ',node,':'])
xlabel('t [s]'),ylabel('F [N]'),set(handle,'color',[0 0.7 0.9])
A = max(Fmag(:,k)) - min(Fmag(:,k));
if A > 0
    axis([t(1) t(end) min(Fmag(:,k))-0.05*A max(Fmag(:,k)) +0.05*A])
end
% 2D Force:
fig = fig + 1; figure(fig)
t scale = 5; %::Time scale factor (t : force)
handle = plot3(t,Fz(:,k),Fy(:,k));
title(['2D Nonlinear Force at Node ',node,' as a Function of Time:'])
xlabel('t [s]'),ylabel('F_{z} [N]'), zlabel('F_{y} [N]')
set(handle,'color',[0 0.7 0.9])
A = max(max(Fy(:,k)) - min(Fy(:,k)),max(Fz(:,k)) - min(Fz(:,k)));
scale = t scale*A/(t(end)-t(1));
if scale > 0
    set(gca,'DataAspectRatio',[1 scale scale])
end
% Axial Moment:
if incl_ang == 'Y'
    fig = fig + 1; figure(fig)
    handle = plot(t,Mx(:,k));
    title(['Axial Nonlinear Moment at Node ',node,':'])
    xlabel('t [s]'),ylabel('M_{x} [N.mm]')
    set(handle,'color',[0 0.7 0.9])
    A = max(Mx(:,k)) - min(Mx(:,k));
    if A > 0
            axis([t(1) t(end) min(Mx(:,k)) -0.05*A max(Mx(:,k)) +0.05*A])
    end
end
end
%SAVE FORCES:
disp(' '),store = lower(input('Save this data (y/n)? ','s'));
if store == 'y'
    filename = input(' Enter a basic filename (*.cSv): ','s');
    for k = 1:nnodes
        node = num2str(nodes(k));
        filenamek = [filename,' n',node,'.csv'];
        csvwrite(filenamek,[t Fy(:,k) Fz(:,k) Mx(:,k)])
        if nnodes > 1
            disp([' Data saved as ',filenamek,'...'])
        else
            disp([' Data saved as ',filenamek])
        end
    end
```

```
    if nnodes > 1
        disp(' Done saving data')
    end
else
    disp(' No data saved')
end
```


## RotorMultiPlot TransCSV.m

```
A program that creates figures to compare transient rotor displacement
and/or force signals loaded from CSV files (created by
RotorDispl TransPCH.m, RotorForceElem TransPCH.m,
RotorForceExc_TransPCH.m and/or RotorForceNL_TransPCH.m).
Note: *Lateral displacements and forces are assumed to be in the yz-plane.
    *Clearance lines (stored by Radialclearance.m) can also be added to
    these figures.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: Aug 2013
% Last revised: July 2014
clear all; clc; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LOAD DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GET CSV FILE- & PLOTTING INFO:
linespec = lower(input('Specify line properties or use defaults...
    (spec/Enter)? ','s'));
disp(' ')
k = 0;
while k <= 20
    k = k + 1;
    % Get File Name:
    flag = 1;
    while flag
    disp(['Enter the CSV filename for file ',num2str(k),' (*.CSv)'])
    cmd = input('or enter ''stop'' to stop loading new data: ','s');
    if strcmpi(cmd,'stop')
                break
            end
            filename{k} = [cmd '.csv'];
            fid = fopen(filename{k},'r');
            if fid == -1
                disp([' Error: file ''',filename{k},''' not found'])
                disp(' Enter filename/stop without extension or ''''')
                proc = lower(input(' Proceed (y/n)? ','s'));
                if proc == 'n'
                    disp(' Data loading cancelled')
                    return
                end
            else
                flag = 0;
                fclose(fid);
            end
        end
        if strcmpi(cmd,'stop')
            break
        end
        % Get Data Type:
        typestr = lower(input(' Enter data type (displ,force): ','s'));
        cortype = 0;
        while cortype == 0
            if strcmp(typestr,'displ')
                type(k) = 1; cortype = 1;
            elseif strcmp(typestr,'force')
                type(k) = 2; cortype = 1;
            else
```

```
            disp([' Error: ''',typestr,''' is not a valid data type'])
            disp(' Type: ''displ'' for displacement signals')
                disp(' ''force'' for force signals (without '''')')
                typestr = input(' Enter data type: ','s');
            end
    end
    % Get Line Specifications:
    if strcmp(linespec,'spec')
        style{k} = lower(input(' Enter a line style specifier...
            (default-,--,:,-.,none): ','s'));
        if isempty(style{k})
            style{k} = '-';
        end
        marker{k} = lower(input(' Enter a line marker specifier...
            (default:none,o,*,etc.): ','s'));
        if isempty(marker{k})
            marker{k} = 'none';
        end
        width_inp = input(' Enter a line width (default=0.5): ');
        if isempty(width_inp)
            width(k) = 0.5;
        else
            width(k) = width_inp;
        end
    else
    style{k} = '-'; marker{k} = 'none'; width(k) = 0.5;
    end
    % Get Scale Factor:
    scale_inp = input(' Enter a scale factor (default=1): ');
    if isempty(scale_inp)
        scale(k) = 1;
    else
        scale(k) = scale_inp;
    end
    disp(' ')
%LOAD DATA:
    data = csvread(filename{k});
    tvec = data(:,1); l(k) = length(tvec);
    yvec = scale(k)*data(:,2); zvec = scale(k)*data(:,3);
    t(1:l(k),k) = tvec; y(1:l(k),k) = yvec; z(1:l(k),k) = zvec;
end
n = k - 1; disp(' ')
if n == 0
    disp('Error: No files loaded'),disp(' Program terminated')
    return
end
disp('Files loaded:')
for k = 1:n
    disp([' ',num2str(k),'.',filename{k}])
end
proc = lower(input('Proceed(Enter) or exit(exit)? ','s'));
if strcmp(proc,'exit')
    disp(' Program terminated')
    return
end
%%%%%%%%%%%%%%%%%%%%%%%%%%% SPECIFY DATA LEGENDS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' ')
lgnd = cell(n,1);
for k = 1:n
    lgnd{k} = input(['Enter legend label for file ''',filename{k},''': '],'s');
    if scale(k) ~= 1
        lgnd{k} = [lgnd{k},' (x',num2str(scale(k)),')'];
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%% GET & APPLY TIME LIMITS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' ')
t_start = input('Enter starting time to use for plotting: ');
t_end = input('Enter ending time to use for plotting: ');
for k = 1:n
    index = find(t(1:l(k),k) >= t_start & t(1:l(k),k) <= t_end);
    l(k) = length(index);
    t_new(1:l(k),k) = t(index,k);
    Y_new(1:l(k),k) = y(index,k); z_new(1:l(k),k) = z(index,k);
    tlast(k) = t_new(l(k),k);
end
t = t_new; y = y_new; z = z_new;
%%%%%%%%%%%%%%%%%%%%%%%%%%%% SPECIFY LINE COLOURS % % % % % % % % % % % % % % % % % % % % % % % % % %
%COLORMAP PARAMETERS OR LINE COLOURS:
% Default Jet Colormap Parameters:
cmapsize = 100; %::Jet colormap size (refer to 'Colormap' in Matlab help)
crange = 0.21; %::fraction of colormap used (range)
coffst = 0.33; %::fraction of colormap offset (crange + coffst <= 1)
temp = 0; %::1 = warm; 0 = cold
darkness = 0.85; %::smaller = darker (must be < 1)
% User-Specified Line Colours:
disp(' ')
cspec = lower(input('Specify line colours or use defaults (spec/Enter)? ','s'));
plotfigs = 1;
while plotfigs
    if strcmp(cspec,'spec')
        cspectype = lower(input('Specify Jet Colormap parameters or...
            LineColor (cmap/cline)? ','s'));
        if strcmp(cspectype,'cmap')
            flag = 1;
            disp(' MATLAB Jet Colormap parameters:')
            while flag
                cmapsize = input(' Enter number of colours in map: ');
                crange = input(' Enter % of map colours to be used: ')/100;
                temp = input(' Enter 1 for starting at warm side of map or 0...
                            for cold side: ');
                coffst = input(' Enter % offset from a specific side of the...
                        map: ')/100;
                darkness = input(' Enter brightness %: ')/100;
                if cmapsize < n
                            disp(' Error: number of map colours must be >= number...
                                    of lines to be plotted')
                elseif coffst+crange > 1
                        disp(' Error: % map colours used + % offset must be...
                        <= 100%')
                else
                        flag = 0;
                end
            end
        else
            clines = cell(n,1);
            for k = 1:n
                clines{k} = input([' Enter line colour [RBG] for file ''',...
                        filename{k},''': ']);
            end
        end
    else
        cspectype = 'cmap';
    end
    %DETERMINE LINE COLORMAP POSITION FACTORS (if required):
    if strcmp(cspectype,'cmap')
        cmap = darkness*colormap(jet(cmapsize+1));
        if temp
            cmap = flipud(cmap);
        end
```

```
    if n < 2
    nmap = 2;
    else
        nmap = n;
    end
    kfact = floor(crange*cmapsize/(nmap-1));
    koffst = floor(coffst*cmapsize);
end
%%%%%%%%%%%%%%%%%%% CALCULATE SIGNAL MAGNITUDES & LIMITS %%%%%%%%%%%%%%%%%%%%
    mag = sqrt(y.^2 + z.^2);
    ymax = zeros(n,1); ymin = ymax; zmax = ymax; zmin = ymax;
    magmax = ymax; magmin = ymax;
    for k = 1:n
    ymax(k) = max(y(1:l(k),k)); ymin(k) = min(y(1:l(k),k));
    zmax(k) = max(z(1:l(k),k)); zmin(k) = min(z(1:l(k),k));
    magmax(k) = max(mag(1:l(k),k)); magmin(k) = min(mag(1:l(k),k));
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %IF NO FORCE DATA LOADED:
    if isempty(find(type == 2,1))
        A2D = zeros(n,1);
        for k = 1:n
            if strcmp(cspectype,'cmap')
                kcmap = (k-1)*kfact + koffst + 1;
                cline = cmap(kcmap,:);
            else
            cline = clines{k};
            end
                % Vertical- & Horizontal Displacements Comparison:
                figure(1)
                subplot(2,1,1)
                hold on
                    handle11 = plot(t(1:l(k),k),y(1:l(k),k),'Color',cline);
            hold off
            set(handle11,'LineStyle',style{k},'Marker',marker{k},...
                    'LineWidth',width(k))
            subplot(2,1,2)
            hold on
                    handle12 = plot(t(1:l(k),k), z(1:l(k),k),'Color',cline);
            hold off
            set(handle12,'LineStyle',style{k},'Marker',marker{k},...
                    'LineWidth',width(k))
                % Displacement Magnitude Comparison:
                figure(2)
            hold on
                    handle2 = plot(t(1:l(k),k),mag(1:l(k),k),'Color',cline);
            hold off
            set(handle2,'LineStyle',style{k},'Marker',marker{k},...
                    'LineWidth',width(k))
            % 2D Displacement Comparison:
            figure(3)
            hold on
                    handle3 = plot3(t(1:l(k),k),z(1:l(k),k),y(1:l(k),k),'Color',cline);
            hold off
            set(handle3,'LineStyle',style{k},'Marker',marker{k},...
                    'LineWidth',width(k))
                A2D(k) = max(max(y (1:l (k),k)) -min(y(1:l (k),k)),max(z(1:l(k),k)) ...
                    -min(z(1:l(k),k)));
        end
        figure(1)
        subplot(2,1,1),title('Vertical Displacement Comparison:')
        xlabel('t [s]'),ylabel('y [mm]'),legend(lgnd)
        A = max(ymax) - min(ymin);
        if A > 0
            axis([min(t(1,:)) max(tlast) min(ymin) - 0.05*A max(ymax) +0.05*A])
```

```
    end
    subplot(2,1,2),title('Horizontal Displacement Comparison:')
    xlabel('t [s]'),ylabel('z [mm]'),legend(lgnd)
    A = max(zmax) - min(zmin);
    if A > 0
        axis([min(t(1,:)) max(tlast) min(zmin)-0.05*A max(zmax) +0.05*A])
    end
    figure(2)
    title('Displacement Magnitude Comparison:')
    xlabel('t [s]'), ylabel('r [mm]'),legend(lgnd)
    A = max(magmax) - min(magmin);
    if A > 0
        axis([min(t(1,:)) max(tlast) min(magmin) -0.05*A max(magmax) +0.05*A])
    end
    figure(3)
    title('2D Displacement Comparison:')
    xlabel('t [s]'),ylabel('z [mm]'),zlabel('y [mm]'),legend(lgnd)
    t scale = 5; %::Time scale factor (t : displacement)
    scale = t scale*max(A2D)/(max(tlast)-min(t(1,:)));
    if scale > 0
        set(gca,'DataAspectRatio',[1 scale scale])
    end
%IF ONLY FORCE DATA LOADED
elseif isempty(find(type == 1,1))
    A2D = zeros(n,1);
    for k = 1:n
        if strcmp(cspectype,'cmap')
            kcmap = (k-1)*kfact + koffst + 1;
            cline = cmap(kcmap,:);
        else
            cline = clines{k};
        end
        % Vertical- & Horizontal Forces Comparison:
        figure(1)
        subplot(2,1,1)
        hold on
            handle11 = plot(t(1:l(k),k),y(1:l(k),k),'Color',cline);
        hold off
        set(handle11,'LineStyle',style{k},'Marker',marker{k},...
            'LineWidth',width(k))
        subplot(2,1,2)
        hold on
            handle12 = plot(t(1:l(k),k),z(1:l(k),k),'Color',cline);
        hold off
        set(handle12,'LineStyle',style{k},'Marker',marker{k},...
            'LineWidth',width(k))
        % Force Magnitude Comparison:
        figure(2)
        hold on
            handle2 = plot(t(1:l(k),k),mag(1:l(k),k),'Color',cline);
        hold off
        set(handle2,'LineStyle',style{k},'Marker',marker{k},...
            'LineWidth',width(k))
        % 2D Displacement Comparison:
        figure(3)
        hold on
            handle3 = plot3(t(1:l(k),k),z(1:l(k),k),y(1:l(k),k),'Color',cline);
    hold off
    set(handle3,'LineStyle',style{k},'Marker',marker{k},...
            'LineWidth',width(k))
        A2D(k) = max(max(y(1:l(k),k))-min(y(1:l(k),k)),max(z(1:l(k),k)) ...
            -min(z(1:l(k),k)));
end
figure(1)
subplot(2,1,1),title('Vertical Force Comparison:')
xlabel('t [s]'),ylabel('F_{y} [N]'), legend(lgnd)
```

```
    A = max(ymax) - min(ymin);
if A > 0
    axis([min(t(1,:)) max(tlast) min(ymin) -0.05*A max(ymax) +0.05*A])
end
subplot(2,1,2),title('Horizontal Force Comparison:')
xlabel('t [s]'),ylabel('F_{z} [N]'), legend(lgnd)
A = max(zmax) - min(zmin);
if A > 0
    axis([min(t(1,:)) max(tlast) min(zmin)-0.05*A max(zmax) +0.05*A])
end
figure(2)
title('Force Magnitude Comparison:')
xlabel('t [s]'),ylabel('F [N]'), legend(lgnd)
A = max(magmax) - min(magmin);
if A > 0
        axis([min(t(1,:)) max(tlast) min(magmin) -0.05*A max(magmax) +0.05*A])
end
figure(3)
title('2D Force Comparison:')
xlabel('t [s]'),ylabel('F_{z} [N]'),zlabel('F_{y} [N]'),legend(lgnd)
t_scale = 3; %-:Time scale factor (t : force)
scale = t_scale*max(A2D)/(max(tlast) -min(t(1,:)));
if scale > 0
        set(gca,'DataAspectRatio',[1 scale scale])
end
%IF MIXED DISPLACEMENT & FORCE DATA LOADED:
else
    % Differentiate Data Types:
    type1 = find(type == 1); a = type1(1);
    type2 = find(type == 2); b = type2(1);
    type12 = [type1 type2];
    % Vertical- & Horizontal Signals Comparison:
    figure(1)
    subplot(2,1,1)
    [ax11,h111(a),h111(b)] = ...
        plotyy(t(1:l(a),a),y(1:l(a),a),t(1:l(b),b),y(1:l(b),b));
    if length(type1) > 1
        for k = type1(2:end)
            hl11(k) = line(t(1:l(k),k),y(1:l(k),k),'Parent',ax11(1));
        end
end
if length(type2) > 1
        for k = type2(2:end)
            hl11(k) = line(t(1:l(k),k),y(1:l(k),k),'Parent',ax11(2));
        end
end
k1 = 1;
for k = type12
    if strcmp(cspectype,'cmap')
            kcmap = (kl-1)*kfact + koffst + 1;
            k1 = k1 + 1;
            cline = cmap(kcmap,:);
        else
            cline = clines{k};
        end
        set(hl11(k),'Color',cline)
        set(hlll(k),'LineStyle',style{k},'Marker',marker{k},...
            'LineWidth',width(k))
end
title('Vertical Signal Comparison:'),xlabel('t [s]')
set(get(ax11(1),'Ylabel'),'String','y [mm]')
set(get(ax11(2),'Ylabel'),'String','F_{y} [N]'),legend(lgnd(type12))
set(ax11(1),'YColor','k'), set(ax11(2),'YColor','k')
linkaxes(ax11,'x')
A1 = max(ymax(type1)) - min(ymin(type1)); A2 = max(ymax(type2))...
    - min(ymin(type2));
set(ax11(1),'XLim',[min(t(1,:)) max(tlast)]),...
    set(ax11(2),'XLim',[min(t(1,:)) max(tlast)])
```

```
set(ax11(1),'YLim',[min(ymin(type1))-0.05*A1 max(ymax(type1))+0.05*A1])
set(ax11(2),'YLim', [min(ymin(type2))-0.05*A2 max(ymax(type2)) +0.05*A2])
subplot(2,1,2)
[ax12,hl12(a),hl12(b)] = ...
    plotyy(t(1:l(a),a),z(1:l(a),a),t(1:l(b),b), z(1:l (b),b));
if length(type1) > 1
        for k = type1(2:end)
            hl12(k) = line(t(1:l(k),k),z(1:l(k),k),'Parent',ax12(1));
        end
end
if length(type2) > 1
        for k = type2(2:end)
            hl12(k) = line(t(1:l(k),k),z(1:l(k),k),'Parent',ax12(2));
        end
end
k1 = 1;
for k = type12
    if strcmp(cspectype,'cmap')
        kcmap = (k1-1)*kfact + koffst + 1;
        k1 = k1 + 1;
        cline = cmap(kcmap,:);
        else
            cline = clines{k};
        end
        set(hl12(k),'Color',cline)
        set(hl12(k),'LineStyle',style{k},'Marker',marker{k},...
            'LineWidth',width(k))
end
title('Horizontal Signal Comparison:'),xlabel('t [s]')
set(get(ax12(1),'Ylabel'),'String','z [mm]')
set(get(ax12(2),'Ylabel'),'String','F {z} [N]'),legend(lgnd(type12))
set(ax12(1),'YColor','k'), set(ax12(2),'YColor','k')
linkaxes(ax12,'x')
A1 = max(zmax(type1)) - min(zmin(type1));...
    A2 = max(zmax(type2)) - min(zmin(type2));
set(ax12(1),'XLim',[min(t(1,:)) max(tlast)]),...
    set(ax12(2),'XLim',[min(t(1,:)) max(tlast)])
set(ax12(1),'YLim', [min(zmin(type1)) -0.05*A1 max(zmax(type1))+0.05*A1])
set(ax12(2),'YLim',[min(zmin(type2))-0.05*A2 max(zmax(type2))+0.05*A2])
% Signal Magnitude Comparison:
figure(2)
[ax2,hl2(a),hl2(b)] = ...
    plotyy(t(1:l(a),a),mag(1:l(a),a),t(1:l(b),b),mag(1:l(b),b));
if length(type1) > 1
    for k = type1(2:end)
            hl2(k) = line(t(1:l(k),k),mag(1:l(k),k),'Parent',ax2(1));
        end
end
if length(type2) > 1
        for k = type2(2:end)
            hl2(k) = line(t(1:l(k),k),mag(1:l(k),k),'Parent',ax2(2));
        end
end
k1 = 1;
for k = type12
    if strcmp(cspectype,'cmap')
        kcmap = (k1-1)*kfact + koffst + 1;
        k1 = k1 + 1;
        cline = cmap(kcmap,:);
        else
            cline = clines{k};
        end
        set(hl2(k),'Color',cline)
        set(hl2(k),'LineStyle',style{k},'Marker',marker{k},...
            'LineWidth',width(k))
end
title('Signal Magnitude Comparison:'),xlabel('t [s]')
set(get(ax2(1),'Ylabel'),'String','r [mm]')
```

```
    set(get(ax2(2),'Ylabel'),'String','F [N]'),legend(lgnd(type12))
    set(ax2(1),'YColor','k'),set(ax2(2),'YColor','k')
    linkaxes(ax2,'x')
    A1 = max(magmax(type1)) - min(magmin(type1)); .. 
        A2 = max(magmax(type2)) - min(magmin(type2));
    set(ax2(1),'XLim',[min(t(1,:)) max(tlast)]),...
        set(ax2(2),'XLim',[min(t(1,:)) max(tlast)])
    set(ax2(1),'YLim', [min(magmin(type1))-0.05*A1 max(magmax(type1)) +0.05*A1])
    set(ax2(2),'YLim', [min(magmin(type2))-0.05*A2 max(magmax(type2)) +0.05*A2])
end
if strcmp(cspec,'spec')
    disp(' '),recspec = lower(input('Re-specify line colours (y/n)? ','s'));
    if recspec == 'y'
            close all
        else
            plotfigs = 0;
    end
else
    plotfigs = 0,
end
end
%CLEARANCE LINES (OPTIONAL; IF DISPLACEMENT DATA PRESENT):
if not(isempty(find(type == 1,1)))
    disp(' '),addclear = lower(input('Add clearance lines (y/n)? ','s'));
    if addclear == 'y'
        % Specify Clearance Type and Size:
        flag = 1;
        while flag
        ctype = input(' Specify type of clearance (radial/flat[top]): ','s');
        if strcmp(ctype,'radial') || strcmp(ctype,'flat')
            flag = 0;
            else
                disp([' Error: ''',ctype,''' is not a valid clearance type'])
                disp(' Type: ''radial'' for radial clearance')
                disp(' ''flat'' for flat clearance above rotor..
                    (without '''')')
            end
        end
        rc = input(' Enter the clearance [mm]: ');
            % Contruct Clearance Lines:
            tc = [min(t(1,:)) max(tlast)];
            xc = rc*[1 1];
            % Add to Vertical Signals Comparison:
            lgndc = lgnd;
            if strcmp(ctype,'radial')
            lgndc{end+1} = 'Radial Clearance Lines';
            if isempty(find(type == 2,1)) %displ only
                    figure(1),subplot(2,1,1)
                        hold on
                    plot(tc,xc,tc,-xc,'Color',0.6*[1 1 1])
                        hold off
                    ymax1 = max(max(ymax),rc); ymin1 = min(min(ymin),-rc);
                        A = ymax1 - ymin1; axis([min(t(1,:)) max(tlast) ...
                    ymin1-0.05*A ymax1+0.05*A])
                    legend(lgndc)
            else %mixed displ & force
                        hc = line(tc,xc,'Color',0.6*[1 1 1],'Parent',ax11(1));
                line(tc,-xc,'Color',0.6*[1 1 1],'Parent',ax11(1))
                ymax1 = max(max(ymax(type1)),rc); ymin1 = min(min(ymin(type1)),...
                    -rc)
                        A1 = ymax1 - ymin1;
                set(ax11(1),'YLim',[ymin1-0.05*A1 ymax1+0.05*A1])
                    legend([hl11 hc],lgndc')
            end
        else %if flat
            lgndc{end+1} = 'Top Flat Clearance Line';
```

```
        if isempty(find(type == 2,1)) %displ only
        figure(1),subplot(2,1,1)
        hold on
        plot(tc,xc,'Color',0.6*[1 1 1])
        hold off
        ymax1 = max(max(ymax),rc); ymin1 = min(min(ymin),rc);
        A = ymaxl - ymin1; axis([min(t(1,:)) max(tlast) ...
            ymin1-0.05*A ymax1+0.05*A])
        legend(lgndc)
        else %mixed displ & force
        hc = line(tc,xc,'Color',0.6*[1 1 1],'Parent',ax11(1));
        ymax1 = max(max(ymax(type1)),rc); ymin1 = min(min(ymin(type1)),rc);
        A1 = ymax1 - ymin1;
        set(ax11(1),'YLim',[ymin1-0.05*A1 ymax1+0.05*A1])
        legend([hl11 hc],lgndc')
        end
    end
    % Add to Horizontal Signals Comparison:
    if strcmp(ctype,'radial')
        if isempty(find(type == 2,1)) %displ only
        figure(1),subplot(2,1,2)
            hold on
            plot(tc,xc,tc,-xc,'Color',0.6*[[1 1 1])
            hold off
            zmax1 = max(max(zmax),rc); zmin1 = min(min(zmin),-rc);
            A = zmax1 - zmin1; axis([min(t(1,:)) max(tlast) ...
                zmin1-0.05*A zmax1+0.05*A])
            legend(lgndc)
        else %mixed displ & force
            hc = line(tc,xc,'Color',0.6*[1 1 1],'Parent',ax12(1));
            line(tc,-xc,'Color',0.6*[1 1 1],'Parent',ax12(1))
            zmax1 = max(max(zmax(type1)),rc); zmin1 = min(min(zmin(type1)),...
                    -rc);
            A1 = zmax1 - zmin1;
            set(ax12(1),'YLim',[zmin1-0.05*A1 zmax1+0.05*A1])
            legend([hl12 hc],lgndc')
        end
        % Add to Signal Magnitude Comparison:
        lgndc{end} = 'Radial Clearance Line';
        if isempty(find(type == 2,1)) %displ only
            figure(2)
            hold on
            plot(tc,xc,'Color',0.6*[1 1 1])
            hold off
            rmax = max(max(magmax),rc); rmin = min(min(magmin),rc);
            A = rmax - rmin; axis([min(t(1,:)) max(tlast)...
                rmin-0.05*A rmax+0.05*A])
            legend(lgndc)
    else %mixed displ & force
        hc = line(tc,xc,'Color',0.6*[1 1 1],'Parent',ax2(1));
        rmax = max (max (magmax(type1)),rc); rmin =...
            min(min(magmin(type1)),rc);
        A1 = rmax - rmin;
        set(ax2(1),'YLim',[rmin-0.05*A1 rmax+0.05*A1])
        legend([hl2 hc],lgndc')
        end
    end
    end
end
%ADJUST FORCE Y-AXIS LIMITS (OPTIONAL; IF MIXED DISPLACEMENT & FORCE DATA):
if not(isempty(find(type == 1,1)) || isempty(find(type == 2,1)))
    disp(' '),fadj = lower(input('Adjust force y-axis limits (y/n)? ','s'));
    if fadj == 'Y'
        flimits11 = input(' Enter y-axis limits for Figure 1 top...
            [in vector form]: ');
        flimits12 = input(' Enter y-axis limits for Figure 1 bottom...
            [in vector form]: ');
```

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems
flimits2 = input(' Enter y-axis limits for Figure 2 [in vector form]: '); set (ax11 (2), 'YLim',flimits11), set(ax12(2),'YLim',flimits12) set (ax2(2),'YLim',flimits2)
end
end

## C. 2 Rigid Rotor Model

## RigRotorGravEquil.m

```
A program that calculates the equilibrium deformation (and support
reaction forces) of a rigid rotor with two bearings (bush- & journal
bearing) and a coupling (with a specific configuration).
%
% Note: *Vertical displacement is assumed to be in the y-direction
            (gravitational acceleration: negative y-direction)
            *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: October 2013
% Last revised: July 2014
clear all; close all; clc
%Input Parameters:
m = 2.022729; %::Rotor mass [kg]
kb = 209; %::Bush bearing stiffness coefficient [N/mm]
kj = 250; %::Journal bearing stiffness coefficient [N/mm]
kc = 154; %::Coupling connection stiffness coefficient [N/mm]
lCG = 238.9962; %::Rotor centre of gravity distance [mm]
lb = 42.5; %::Bush bearing acting centre distance [mm]
lj = 435; %::Journal bearing acting centre distance [mm]
lc = 25; %::Coupling connection acting centre distance[mm]
%Calculations:
g = 9.81; %Gravitational acceleration [m/s^2]
m = m/1000; %Convert m from kg to t
g = g*1000; %Convert g from m/s^2 to mm/s^2
A = [kc kb kj
    -kc*lc kb*lb kj*lj
    lb-lj lc+lj -lc-lb];
B = -[m*g m*g*lCG 0]';
y = A\B;
F = -[kc kb kj]'.*y;
%Results:
disp(['Y_b = ',num2str (y(2)),' mm & F_b = ',num2str(F(2)),' N'])
disp(['y j = ',num2str(y(3)),' mm & F j = ',num2str(F(3)),' N'])
disp(['Y_c = ',num2str(y(1)),' mm & F_C = ',num2str(F(1)),' N'])
```


## C. 3 Journal Bearing Models

## FinHDJB_Reynolds.m

```
function [theta_v,z_v,p,Fx,Fy] = .. 
FinHDJB_Reynolds(x,y,xdot,ydot,xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,iterout)
% [theta_v,z_v,p,Fx,Fy] =
% FinHDJB_Reynolds(x,y,xdot,ydot,xoff,yoff,omega,r,c,l,nu,n,m,tol,
cav,pcav,iterout)
Function that calculates the fluid-film pressure distribution within a
hydrodynamic journal bearing using an iterative finite difference method.
This function can include the effects of cavitation by applying Reynolds
BCs.
Function that calculates the fluid-film pressure distribution within a
hydrodynamic journal bearing using an iterative finite difference method.
Cavitation can be accounted for with Reynolds BCs by equating negative
pressures to zero at after each iteration during the solution of the
pressure distribution.The bearing forces are calculated by integrating
the pressure distribution over the journal surface.
Input: -x & y: journal centre coordinates (horizontal & vertical
    respectively
    -xdot & ydot: journal centre velocities (horizontal & vertical
    respectively)
    -xoff & yoff: bearing housing offsets (horizontal & vertical
    respectively)
    -omega: journal spin speed
    -r: journal radius
    -c: clearance
    -l: length
    -nu: lubricant dynamic viscosity
    -n & m: number of grid divisions in theta- & z-directions
        respectively
    -tol: pressure tolerance for the SOR method
    -cav: 1 = include cavitation; 0 = exclude cavitation
    -pcav: cavitation pressure
    -iterout: 1 = display iteration information; 0 = do not display
        iteration information
Output: -theta v: vector of theta-coordinate values
    -z_v: vector of z-coordinate values
    -p: fluid film pressure distribution matrix
    -Fx & Fy: journal fluid film forces in the x- & y-directions
        respectively
Author: Armand Krüger
% Created: May 2012
% Last revised: June 2014
%FINITE DIFFERENCE METHOD:
tic
x = x - xoff; y = y - yoff;
% Discrete Spatial Variables:
theta0 = 0; thetan = 2*pi;
dtheta = (thetan - theta0)/n;
dz = l/m;
theta_v = theta0:dtheta:thetan;
z_v = -l/2:dz:l/2;
% Optimal Over-Relaxation Factor:
D = 2* (r+c);
B = 1-pi^2/2*(4 + (pi*D/l)^2)/(m^2 + (pi*D*n/l)^2);
alpha = 2*(1-sqrt(1-B^2))/B^2;
if iterout
    disp(['Optimal OR Factor: ', num2str(alpha)])
end
% Initial Guess for p:
p = zeros(n+1,m+1);
```

```
% Solve p (SOR Method):
k = 1; dpnorm = tol + 1;
while k <= 3000 && dpnorm > tol
    p_prev = p;
    for i = 2:n+1
        thetai = theta0 + (i-1)*dtheta;
        hi = c - x*cos(thetai) - y*sin(thetai);
        dhidthetai = x*sin(thetai) - y*cos(thetai);
        dhidt = -xdot*cos(thetai) - ydot*sin(thetai);
        P01i = hi^3/dtheta^2 - 3*hi^2/(2*dtheta)*dhidthetai;
        P11i = hi^3/dtheta^2 + r^2*hi^3/dz^2;
        P21i = hi^3/dtheta^2 + 3*hi^2/(2*dtheta)*dhidthetai;
        P10i = r^2*hi^3/dz^2;
        fi = 6*nu*r^2*omega*dhidthetai + 12*nu*r^2*dhidt;
        A1i = 0.5*P01i/P11i;
        A2i = 0.5*P21i/P11i;
        A3i = 0.5*P10i/P11i;
        A4i = -0.5*fi/P11i;
        for j = 2:m
            if i == n+1
            i_plus1 = 2;
            else
            i_plus1 = i + 1;
            end
            p(i,j) = p(i,j) + alpha*(A1i*p(i-1,j) + A2i*p(i_plus1,j) +...
                    A3i*(p(i,j-1)+p(i,j+1)) + A4i - p(i,j));
        end
    end
    p(1,:) = p(n+1,:); %Pressure continuity BC
    if cav == 1 %Reynolds BCs
        p_new = pcav*ones(n+1,m+1);
        index = find(p > pcav);
        p_new(index) = p(index);
        p = p_new;
    end
    dp = p - p_prev;
    dpnorm = norm(dp);
    if iterout
        disp(['SOR iteration: ',num2str(k),'; Pressure difference...
            norm: ',num2str(dpnorm)])
    end
    k = k + 1;
end
finish = toc;
if iterout
    disp(['Done with finite bearing pressure calculation: ',num2str(finish),' s'])
end
%BEARING FORCES:
Fx = 0; Fy = 0;
for i = 2:n+1
    for j = 2:m+1
        Fx = Fx + (p(i,j) +p(i,j-1))*cos(theta v(i)) +...
            (p(i-1,j)+p(i-1,j-1))*cos(theta_v(\overline{i}-1));
        Fy = Fy + (p(i,j)+p(i,j-1))*sin(thēta_v(i)) + ...
            (p(i-1,j) +p(i-1,j-1))*sin(theta_v(i-1));
    end
end
Fx = -r*dz*dtheta/4*Fx; Fy = -r*dz*dtheta/4*Fy;
```


## FinHDJB_ReynoldsRun.m

```
A program to provide input parameters to function FinHDJB Reynolds.m and
then display and plot the results from this function. The program also
creates or adds to a pressure distribution figure for comparison to other
bearing models (figure 1).
Note: *Custom function FinHDJB_Reynolds.m is required in the working
    directory.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: May }201
% Last revised: July }201
```


if not (exist('comp','var'))
clear all; clc; close all
else
disp(' ')
end
flag = 1;
while flag
filename = input('Enter the input file name (*.inp): ','s');
filename = [filename '.inp'];
fid $=$ fopen(filename,'r');
if fid == -1
disp([' Error: file ''',filename,''' not found'])
disp(' Enter filename without extension or ''''')
proc = lower(input(' Proceed (y/n)? ','s'));
if proc == 'n'
disp(' Program terminated')
return
end
else
flag = 0;
end
end
disp(' ')
dummy $=$ fgetl(fid); dummy $=$ fscanf(fid,'\%c',[3,1])';
$\mathrm{x}=\mathrm{fscanf}\left(\mathrm{fid}, ' \% f \mathrm{n}^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid); dummy = fscanf(fid,'\%c',[3,1])';
$y=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ; ~ d u m m y=$ fgetl(fid); dummy = fscanf(fid,' $\% C^{\prime},[6,1]$ )';
xdot $=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ; ~ d u m m y=$ fgetl(fid); dummy =...
fscanf(fid, '\%c', [6,1])';
ydot $=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid); dummy $=\ldots$
fscanf(fid, ' \% ',$[6,1]$ )';
xoff $=\mathrm{fscanf}\left(f i d, ' \% f \backslash n^{\prime},[1,1]\right) ;$ dummy $=f g e t l(f i d) ; ~ d u m m y=.$.
fscanf(fid,'\%c', [6,1])';
yoff $=$ fscanf(fid,' $\left.\% f \backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid);
dummy $=$ fgetl(fid); dummy = fscanf(fid,' \%c', [3,1])';
$\mathrm{N}=\mathrm{fscanf}\left(\mathrm{fid}, ' \% \mathrm{f} \backslash \mathrm{n}^{\prime},[1,1]\right) ;$ dummy = fgetl(fid);
dummy $=$ fgetl(fid); dummy = fscanf(fid,' \%c', [3,1])';
$c=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ; ~ d u m m y=f g e t l(f i d) ; ~ d u m m y=f s c a n f\left(f i d, ' \% c^{\prime},[3,1]\right) ' ;$
$1=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ; ~ d u m m y=$ fgetl(fid); dummy = fscanf(fid,' $\left.\% \mathrm{f}^{\prime},[3,1]\right)$ ';
$d=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid);
dummy $=$ fgetl(fid); dummy = fscanf(fid,' $\%$ ', $[4,1]$ )';
$n u=f s c a n f\left(f i d, 1 \% f \backslash n^{\prime},[1,1]\right) ;$ dummy = fgetl(fid);
dummy $=$ fgetl(fid); dummy $=$ fscanf(fid,'\%c', [3,1])';
$\left.\mathrm{n}=\mathrm{fscanf(fid,'} \mathrm{\% f} \mathrm{\ n}^{\prime},[1,1]\right) ; ~ d u m m y=$ fscanf(fid,'\%c',[3,1])';
$m=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right)$;
dummy $=$ fgetl(fid); dummy = fscanf(fid,'\%c',[5,1])';
cav $=\mathrm{fscanf}\left(f i d, ' \% f \backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid); dummy $=\ldots$
fscanf(fid,' \%c', [6,1])';
pcav $=$ fscanf(fid,' $\left.\% f \backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid);
dummy $=$ fgetl(fid); dummy = fscanf(fid,' $\%$ ', [5,1])';
tol $=\mathrm{fscanf}\left(f i d, ' \% f \backslash n^{\prime},[1,1]\right)$;
fclose(fid);

\%BEARING PARAMETERS:

```
omega = 2*pi*N/60;
r = d/2;
D = d + 2*C;
e = sqrt( }\mp@subsup{x}{}{\wedge}2+\mp@subsup{y}{}{\wedge}2)
eps = e/c;
l_d = l/d;
%BEARING PRESSURE DISTRIBUTION & FORCES:
[theta,z,p,Fx,Fy] = ...
    FinHDJB_Reynolds(x,y,xdot,ydot,xoff,yoff,omega,r,c,l,nu,n,m,tol, cav,pcav,1);
theta = theta/pi*180;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%BEARING PARAMETERS:
disp(' '),disp(['Housing diameter: D = ',num2str(D),' mm'])
disp(['Bearing eccentricity: e = ',num2str(e),' mm'])
disp(['Excentricity ratio: epsilon = ',num2str(eps)])
disp(['Length/diameter ratio: l/d = ',num2str(l_d)])
%BEARING FORCES:
disp(['Horizontal force: Fx = ',num2str(Fx),' N'])
disp(['Vertical force: Fy = ',num2str(Fy),' N'])
%3D PRESSURE DISTRIBUTION:
if exist('fig','var')
    fig = fig + 1;
else
    fig = 2;
end
figure(fig)
surf(z,theta,p),title('Journal Bearing Pressure Distribution:')
xlabel('z [mm]'),ylabel('0 [deg]'), zlabel('p [MPa]')
colorbar
%3D PRESSURE DISTRIBUTION WITH CONTOURS:
fig = fig + 1;
figure(fig)
cn = 30; %::Number of contour levels
p_max = max (max (p)); p_min = min(min(p));
v = linspace(0.005*p_max+p_min,0.99*p_max,cn);
contour3(z,theta,p,v)
surface(z,theta,p,'EdgeColor',0.8*[1 1 1],'FaceColor','none')
title('Journal Bearing Pressure Distribution:')
xlabel('z [mm]'),ylabel('0 [deg]'), zlabel('p [MPa]')
colorbar
grid off
%MID-JOURNAL (z=0) CYLINDRICAL PRESSURE DISTRIBUTION:
% Plot Appearance/Adjustment Parameters:
p_scale = 0.4; %::Pressure scale factor relative to d
J_scale = 5; %::Journal eccentricity scale factor
H_scale = 1.1; %::Housing radius scale factor
% Housing & Journal:
journal_x = d/2*cosd(theta) + x*J_scale;
journal_y = d/2*sind(theta) + y*J_scale;
housing_x = D/2*cosd(theta)*H scale;
housing_y = D/2*sind(theta)*H_scale;
% Pressure Distribution:
p_mid = p(:,m/2)';
p_mid_m = max(p_mid);
p_mid_x = (p_mid/p_mid_m*d*p_scale + d/2).*cosd(theta) + x*J_scale;
p_mid_y = (p_mid/p_mid_m*d*p_scale + d/2).*sind(theta) + y*J_scale;
% Line of Ceñtres:
lc}x=[-0.6*D/sqrt(1+(y/x)^2) 0.6*D/sqrt(1+(y/x)^2)]
lc_-y = y/x*lc_x;
% Velocity Vector:
Vmag = sqrt(xdot^2 + ydot^2);
if Vmag > 0
    xdot_v = [0 xdot/Vmag*0.3*D];
```

```
    ydot_v = [0 ydot/Vmag*0.3*D];
end
% Force Vector:
Fmag = sqrt(Fx^2 + FY^2);
if Fmag > 0
    Fx_v = [0 Fx/Fmag*0.2*D];
    Fy_v = [0 Fy/Fmag*0.2*D];
end
% Plot:
fig = fig + 1;
figure(fig)
handle = plot(journal_x,journal_y,x*J_scale,y*J_scale,'+');
set(handle,'color', [0 0.7 0.9])
hold on
handle = plot(housing_x,housing_y,0,0,'+',lc_x,lc_y,'--');
set(handle,'color',0.\overline{6}*[[1 1 1])
handle = plot(p_mid_x,p_mid_y);
set(handle,'colōr', [0.5-0.8-0.2])
if Vmag > 0
    plot(xdot_v,ydot_v,'r')
end
if Fmag > 0
    handle = plot(Fx_v,Fy_v);
    set(handle,'color',[0.5 0.0 0.8])
end
hold off
title('Journal Bearing Geometry & Pressure Distribution:')
xlabel('x [mm]'),ylabel('y [mm]')
if Vmag > 0 && Fmag > 0
    legend('Journal','Journal Centre','Housing','Housing Centre',...
            'Line of Centres',...
                            'Pressure [p(0,0)]','Velocity Direction','Force Direction')
elseif Vmag > 0
    legend('Journal','Journal Centre','Housing','Housing Centre',...
            'Line of Centres',...
            'Pressure [p(0,0)]','Velocity Direction')
elseif Fmag > 0
    legend('Journal','Journal Centre','Housing','Housing Centre',...
        'Line of Centres',...
        'Pressure [p(0,0)]','Force Direction')
else
    legend('Journal','Journal Centre','Housing','Housing Centre',...
            'Line of Centres',...
            'Pressure [p(0,0)]')
end
axis equal
%MID-JOURNAL CYLINDRICAL PRESSURE DISTRIBUTION (Polar Plot):
fig = fig + 1;
figure(fig)
lines theta = zeros(2*length(theta),1); lines_p = lines theta;
for k = 1:length(theta)
    lines theta(2*k-1) = theta(k)*pi/180;
    lines_theta(2*k) = theta(k)*pi/180;
    if mod(k,2) == 0
        k1 = 2*k; k2 = 2*k - 1;
    else
        k1 = 2*k - 1; k2 = 2*k;
    end
    lines_p(k1) = d/2;
    lines_p(k2) = p_mid(k)/p_mid_m*d*p_scale + d/2;
end
journal_r = d/2*ones(1,length(theta));
handle = polar(lines_theta,lines_p);
```



```
hold on
handle = polar(theta*pi/180,journal_r);
set(handle,'color',0.6*[1 1 1],'LinēWidth',1.5)
```

```
handle = polar(theta*pi/180,p_mid/p_mid_m*d*p_scale + d/2);
set(handle,'color',[0 0.7 0.9],'LineWidth',1.5)
hold off
title('Journal Geometry & Pressure Distribution:')
legend('Pressure Distribution [p(0,0)]','Journal')
%MID-JOURNAL PRESSURE DISTRIBUTION COMPARISON FIGURE:
if exist('comp','var')
    comp = comp + 1;
else
    comp = 1;
end
lgnd{comp} = 'Iterative Finite Difference Method (Reynolds BCs)'; %::Method label
figure(1)
hold on
handle = plot(theta,p mid,'r');
set(handle,'color',[0`0.7 0.9],'LineWidth',1.5) %::Comparison plot line colour
title('Journal Bearing Centre (z=0) Pressure Distribution:')
xlabel('0 [deg]'),ylabel('p(0,0) [MPa]'),grid on
hold off
if comp > 1
    legend(char(lgnd))
end
```


## InputParameters.inp

```
JOURNAL CENTRE POSITION, VELOCITY & OFFSET:
x = 0.09079 [mm]
y = -0.091336 [mm]
xdot = 0 [mm/s]
ydot = 0 [mm/s]
xoff = 0 [mm]
yoff = 0 [mm]
JOURNAL SPIN SPEED:
N = 4000 [rpm]
BEARING DIMENSIONS:
C = 0.2 [mm]
l = 11.5 [mm]
d = 24.5 [mm]
OIL VISCOSITY:
nu = 4e-8 [MPa.s]
NUMBER OF PRESSURE GRID POINTS IN X- & Y-DIRECTIONS:
n = 120
m = 30
CAVITATION [MPa]:
cav = 1 (1 = include; 0 = exclude)
pcav = 0 [MPa]
PRESSURE TOLERANCE FOR SOR METHOD:
tol = 1e-6
```


## HDJB_CylPressDist.m

```
% A program that creates a 3-D figure of the bearing journal outer surface
and the cylindrical fluid-film pressure distribution around the journal
using the pressure distribution calculated by one of the programs
listed above.
% Note: *First run ShortHDJBRun.m, FinHDJB_DirRun.m, FinHDJB_EstRun.m,
    FinHDJB_GumbelRun.m or FinHDJB_ReynoldsRun.m
% Author: Armand Krüger
% Created: May }201
% Last revised: July 2014
p_scale = 0.5; %::Pressure scale factor relative to d
fig = fig + 1;
figure(fig)
```

```
% Journal Outer Surface:
xn = length(theta); zn = length(z);
j theta = linspace (0,360,xn)';
j_xv = 0.99*d/2*cosd(j_theta);
j_yv = 0.99*d/2*sind(j_theta);
j_zv = linspace(-1/2,1//2,zn);
j_y = zeros(xn,zn);
for k = 1:zn
    j_y(:,k) = j_yv;
end
surf(j_zv,j_xv,j_y,'EdgeColor',0.6*[llll
% Cylindrical Pressure Distribution:
p_max = max(max(p));
p_theta x = zeros(length(theta),length(z));
p_theta_y = p_theta_x; p_theta_z = p_theta_x;
for k = 1:length(z)
    p_theta = p(:,k);
    p_theta_x(:,k) = (p_theta/p_max*d*p_scale + d/2).*cosd(theta');
    p_theta_y(:,k) = (p_theta/p_max*d*p_scale + d/2).*sind(theta');
    p_theta_z(:,k) = linspace(z(k),z(k),length(theta))';
end
hold on
surface(p_theta_z,p_theta_x,p_theta_y,p/p_max*d*p_scale*50);
hold off
title('Journal Bearing Pressure Distribution:')
xlabel('z [mm]'),ylabel('x [mm]'), zlabel('y [mm]')
%camlight ('left','infinite'); lighting phong
%alpha(0.8)
axis equal
```


## HDJB_StatEquil.m

```
% A program that determines the static equilibrium position/coordintes of a
journal bearing subject to a given static load.
%
% Note: *Custom function FinHDJB_Reynolds.m is required in the working
    directory.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: September 2013
% Last revised: July 2014
clear all; close all; clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%% INPUT PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Bearing Input Parameters:
flag = 1;
while flag
    filename = input('Enter the input file name (*.inp): ','s');
    filename = [filename '.inp'];
    fid = fopen(filename,'r');
    if fid == -1
        disp([' Error: file ''',filename,''' not found'])
        disp(' Enter filename without extension or ''''')
        proc = lower(input(' Proceed (y/n)? ','s'));
        if proc == 'n'
            disp(' Program terminated')
            return
            end
    else
            flag = 0;
    end
end
disp(' ')
dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid); dummy = fgetl(fid);
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[6,1])';
```

```
xoff = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid); dummy = ...
    fscanf(fid,'%c',[6,1])';
yoff = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid)
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[3,1])';
N = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid);
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[3,1])';
c = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid); dummy = fscanf(fid,'%c',[3,1])';
l = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid); dummy = fscanf(fid,'%c',[3,1])';
d = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid);
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[4,1])';
nu = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid);
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[3,1])';
n = fscanf(fid,'%f \n',[1,1]); dummy = fscanf(fid,'%c',[3,1])';
m = fscanf(fid,'%f \n',[1,1]);
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[5,1])';
cav = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid); dummy = ...
    fscanf(fid,'%c',[6,1])';
pcav = fscanf(fid,'%f \n',[1,1]); dummy = fgetl(fid);
dummy = fgetl(fid); dummy = fscanf(fid,'%c',[5,1])';
tol = fscanf(fid,'%f \n',[1,1]);
fclose(fid);
%Equilibrium Position Input Parameters:
Lx = 0; %::Bearing x-direction loading [N]
Ly = -10; %::Bearing y-direction loading [N]
% Load stepping factors:
Lsteps =[[0.005 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0. 8 1];
tol_equil = 1e-7; %::Newton-Raphson method tolerance
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculate Certain Bearing Parameters:
omega = 2*pi*N/60;
r = d/2;
%Guess/Starting Values:
dx = 1e-5; dy = 1e-5;
x0 = 0; y0 = 0;
x_v = zeros(length(Lsteps)+1,1); y_v = x_v; Lx_v = x_v; Ly_v = x_v; e_v = x_v;
x_v(1) = x0; y v(1) = y0;
for Lk = 1:length(Lsteps)
    Lxstep = Lsteps(Lk)*Lx;
    Lystep = Lsteps(Lk)*Ly;
    disp(['========= Load Step ',num2str(Lk),' - ',num2str(Lsteps(Lk)*100), ...
            '% ========='])
        %Evaluate f1 & f2 @ Guess/Starting Values:
        [~,~,~,Fx0,Fy0] = .. 
            FinHDJB Reynolds(x0,y0,0,0,xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,0);
        [~,~,~,Fx0dx,Fy0dx] = FinHDJB_Reynolds (x0-dx,y0,0,0,\ldots
            xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,0);
        [~, ~, ~,Fx0dy,Fy0dy] = FinHDJB_Reynolds (x0,y0-dy,0,0,\ldots
            xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,0);
        R1_0 = Fx0 + Lxstep; R2_0 = Fy0 + Lystep;
        R1_0dx = Fx0dx + Lxstep; R2_0dx = Fy0dx + Lystep;
        R1_0dy = Fx0dy + Lxstep; R2_0dy = Fy0dy + Lystep;
        uk = [x0;y0];
        Rk = [R1_0;R2_0];
        Rkdxdy =- [R1_0}dx R1_0dy
                        R2_0dx R2_0dy];
        %Solve Displacements:
        k = 0; Enorm = 1;
        while k <= 1000 && Enorm > tol_equil
            %Construct Jacobian Matrix:
            Jk11 = (Rk(1) - Rkdxdy (1,1))/dx;
            Jk12 = (Rk(1) - Rkdxdy (1,2))/dy;
            Jk21 = (Rk(2) - Rkdxdy (2,1))/dx;
```

```
    Jk22 = (Rk(2) - Rkdxdy (2,2))/dy;
    Jk = [Jk11 Jk12
        Jk21 Jk22];
    %Calculate New Displacements:
    Ek = -Jk\Rk; Enorm = norm(Ek);
    ukpls1 = Ek + uk;
    disp(['Iteration: ',num2str(k+1),'; Displ. norm: ',num2str(Enorm)])
    %Calculate New Forces:
    [~,~,~,Fxkpls1,Fykpls1] = FinHDJB_Reynolds(ukpls1(1),ukpls1(2),0,0, ...
        xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,0);
    [~,~,~,Fxkplsldx,Fykpls1dx] = FinHDJB_Reynolds(ukpls1(1) -dx,ukpls1(2), ...
    0,0,xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,0);
[~,~,~,Fxkplsldy,Fykpls1dy] = FinHDJB_Reynolds(ukpls1(1),...
    ukpls1(2)-dy,0,0,xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,0);
R1kpls1 = Fxkpls1 + Lxstep; R2kpls1 = Fykpls1 + Lystep;
R1kpls1dx = Fxkpls1dx + Lxstep; R2kpls1dx = Fykpls1dx + Lystep;
R1kpls1dy = Fxkpls1dy + Lxstep; R2kpls1dy = Fykpls1dy + Lystep;
Rkpls1 = [R1kpls1;R2kpls1];
Rkpls1dxdy = [R1kpls1dx R1kpls1dy
                                    R2kpls1dx R2kpls1dy];
%For Next Iteration:
k = k + 1;
Rk = Rkpls1; Rkdxdy = Rkpls1dxdy;
uk = ukpls1;
end
x = ukpls1(1); y = ukplsl(2);
Lx_v(Lk+1) = Lxstep; Ly_v(Lk+1) = Lystep;
L= sqrt(Lxstep^2 + Lystep^2); L_v(Lk+1) = L;
x_v(Lk+1) = x; y_v(Lk+1) = y;
e = sqrt (x^2 + y^2); e_v(Lk+1) = e;
[~,~,~,Fx,Fy] = ...
FinHDJB_Reynolds(x,y,0,0,xoff,yoff,omega,r,c,l,nu,n,m,tol,cav,pcav,0);
disp(' '),disp(['Fx = ',num2str(Fx),' N & Fy = ',num2str(Fy),' N'])
disp(['at x = ', num2str(x),' mm & y = ', num2str(y),' mm']),disp(' ')
%For Next Load Step:
x0 = x; y0 = y;
end
e = sqrt( (x^2 + y^2);
eps = e/c;
disp(['Bearing eccentricity: e = ',num2str(e),' mm'])
disp(['Excentricity ratio: epsilon = ',num2str(eps)])
figure(1)
subplot (2,1,1)
plot(Lx_v,x_v,'color', [0 0.7 0.9])
title('\overline{x}-Direction Static Journal Displacement vs. -Load:')
xlabel('L_{x} [N]'),ylabel('x [mm]')
subplot (2,1,2)
plot(Ly v,y v,'color',[0 0.7 0.9])
```



```
xlabel('L_{y} [N]'),ylabel('y [mm]')
figure(2)
x_clear = linspace(0,sign(x)*c,100);
y_clear = sign(y)*sqrt(c^2-x_clear.^2);
hāndle = plot(x_v,y_v,'o-',x_clear,y_clear,'--');
set(handle(1),'\overline{color', [0 0.7-0.9]), sēt(handle(2),'color',0.6*[11 1 1])}
title('Journal Static Displacement due to Loading:')
xlabel('x [mm]'),ylabel('y [mm]'), axis equal
legend('Journal Path','Clearance Circle')
figure(3)
plot(L_v,e_v,'color',[[0 0.7 0.9])
title('Static Eccentricity vs. Load:')
xlabel('L [N]'),ylabel('e [mm]')
```


## ShortHDJB.m

```
function [theta v, z v, P,Fx,Fy] = ..
ShortHDJB(x,y,x\overline{dot, प}\textrm{ydot},\textrm{xoff,yoff,omega,r, c,l, nu, n, m, cav,durout)}
[theta_v, z_v,P,Fx,Fy] =
ShortH\overline{D}JB(\overline{x},y,xdot,ydot, xoff,yoff, omega,r, c, l, nu, n,m, cav, durout)
Function that calculates the fluid-film pressure distribution within a
hydrodynamic journal bearing using the short bearing approximation.
Cavitation can be accounted for by applying Gümbel or Half-Sommerfeld BCs
(equating negative pressures to zero). The bearing forces are calculated
by integrating the pressure distribution over the journal surface.
Input: -x & y: journal centre coordinates (horizontal & vertical
        respectively
    -xdot & ydot: journal centre velocities (horizontal & vertical
    respectively)
    -xoff & yoff: bearing housing offsets (horizontal & vertical
    respectively)
    -omega: journal spin speed
    -r: journal radius
    -c: clearance
    -l: length
    -nu: dynamic viscosity
    -n & m: number of grid divisions in theta- & z-directions
        respectively
    -cav: 1 = include cavitation; 0 = exclude cavitation
    -durout: 1 = display calcs duration; 0 = do not display calcs
    duration
Output: -theta_v: vector of theta-coordinate values
    -z_v: vector of z-coordinate values
    -p: fluid film pressure distribution matrix
    -Fx & Fy: journal fluid film forces in the x- & y-directions
        respectively
% Author: Armand Krüger
% Created: March 2012
% Last revised: June 2014
%PRESSURE DISTRIBUTION:
tic;
x = x - xoff; y = y - yoff;
theta_v = linspace(0,2*pi,n);
z_v = linspace(-l/2,l/2,m);
theta_vl = length(theta_v);
z_vl = length(z_v);
p = zeros(theta_vl,z_vl); p_nocav = p;
for i = 1:n
    for j = 1:m
        theta = theta_v(i); z = z_v(j);
        h = c - x*cos(theta) - y*sin(theta);
        dh_dtheta = x*sin(theta) - y*cos(theta);
        dh_dt = -xdot*cos(theta) - ydot*sin(theta);
        p(\overline{i},j) = 3*nu/h^3* (z^2 - l^2/4)*(omega*dh_dtheta + 2*dh_dt);
        p_nocav(i,j) = p(i,j);
        if cav == 1
                if p(i,j) < 0
                    p(i,j) = 0;
                end
            end
    end
end
finish = toc;
if durout
    disp(['Done with short bearing pressure approximation: ',num2str(finish),' s'])
end
```

```
%BEARING FORCES:
% Cavitation Boundaries:
if cav == 1
    % Determine theta1 & theta2 if Cavitation Considered:
    theta2 = atan2(omega*y + 2*xdot,omega*x - 2*ydot);
    if theta2 < 0
            theta2 = theta2 + 2*pi;
    end
    theta1 = theta2 - pi;
else
    % Integration Limits if Cavitation Ignored:
    theta1 = 0;
    theta2 = 2*pi;
end
% Functions to be Integrated:
clear theta
function F_int_x = FIntegralX(theta)
    num = omega*(x*sin(theta) - y* cos(theta)) - 2*(xdot*cos(theta) +
ydot*sin(theta));
    den = (c - x* cos(theta) - y*sin(theta)).^3;
    F_int_x = num./den.*}\operatorname{cos(theta);
end
function F_int_y = FIntegralY(theta)
    num = omega*(x*sin(theta) - y*cos(theta)) - 2*(xdot*cos(theta) +
ydot*sin(theta));
    den = (c - x*cos(theta) - y*sin(theta)).^3;
    F_int_y = num./den.*sin(theta);
end
% Bearing Forces:
Fx = nu*r*l^3/2*quadl(@FIntegralX,theta1,theta2);
Fy = nu*r*l^3/2*quadl(@FIntegralY,theta1,theta2);
End
```


## ShortHDJB_Num.m

```
function [theta_v,z_v,p,Fx,Fy] = .. 
ShortHDJB Num(x,y,xdot,ydot,xoff,yoff,omega,r, c, l, nu,n,m,cav,durout)
    [theta_v,z_v,p,Fx,Fy] =
    ShortHDJB Num(x,y,xdot,ydot,xoff,yoff,omega,r, c, l, nu, n,m,cav,durout)
    Function that calculates the fluid-film pressure distribution within a
    hydrodynamic journal bearing using the short bearing approximation.
    Cavitation can be accounted for by applying Gümbel or Half-Sommerfeld BCs
    (equating negative pressures to zero). The bearing forces are calculated
    by numerically integrating the pressure distribution over the journal
    surface.
    Input: -x & y: journal centre coordinates (horizontal & vertical
        respectively
        -xdot & ydot: journal centre velocities (horizontal & vertical
            respectively)
            -xoff & yoff: bearing housing offsets (horizontal & vertical
            respectively)
            -omega: journal spin speed
            -r: journal radius
            -c: clearance
            -l: length
            -nu: dynamic viscosity
            -n & m: number of grid divisions in theta- & z-directions
            respectively
            -cav: 1 = include cavitation; 0 = exclude cavitation
            -durout: 1 = display calcs duration; 0 = do not display calcs
            duration
    Output: -theta_v: vector of theta-coordinate values
            -z_v: vector of z-coordinate values
            -p: fluid film pressure distribution matrix
```

```
%
%
% Author: Armand Krüger
% Created: April 2014
%PRESSURE DISTRIBUTION:
tic;
x = x - xoff; y = y - yoff;
theta_v = linspace(0,2*pi,n);
z_v = linspace(-l/2,l/2,m);
theta vl = length(theta v);
z_vl = length(z_v);
p = zeros(theta_vl,z_vl); p_nocav = p;
for i = 1:n
    for j = 1:m
        theta = theta_v(i); z = z_v(j);
        h = c - x*cos(theta) - y*sin(theta);
        dh_dtheta = x*sin(theta) - y*cos(theta);
        dh dt = -xdot*cos(theta) - ydot*sin(theta);
        p(i},j)=3*nu/h^3* (z^2 - l^2/4)*(omega*dh_dtheta + 2*dh_dt)
        p_nocav(i,j) = p(i,j);
        if cav == 1
            if p(i,j) < 0
                        p(i,j) = 0;
            end
        end
    end
end
finish = toc;
if durout
    disp(['Done with short bearing pressure approximation: ',num2str(finish),' s'])
end
%BEARING FORCES:
% Cavitation Boundaries:
if cav == 1
    % Determine thetal & theta2 if Cavitation Considered:
    theta2 = atan2(omega*y + 2*xdot,omega*x - 2*ydot);
    if theta2 < 0
        theta2 = theta2 + 2*pi;
    end
    theta1 = theta2 - pi;
else
    % Integration Limits if Cavitation Ignored:
    thetal = 0;
    theta2 = 2*pi;
end
% Bearing Forces:
dtheta = (theta2 - thetal)/n;
Fx = 0; FY = 0;
for i = 1:n
    thetai = thetal + i*dtheta; thetaim1 = thetai - dtheta;
    dhdthetai = x*sin(thetai) - y* cos(thetai); dhdthetaim1 = x*sin(thetaim1)...
        - y*}\operatorname{cos(thetaim1);
    dhdti = -xdot*cos(thetai) - ydot*sin(thetai); dhdtim1 = -xdot*cos(thetaim1).. 
        - ydot*sin(thetaim1);
    h3i = (c - x*cos(thetai) - y*Sin(thetai))^3; h3im1 = (c - x*cos(thetaim1)..
        - y*}\operatorname{sin}(thetaim1))^3
    Fi = (omega*dhdthetai + 2*dhdti)/h3i; Fim1 = (omega*dhdthetaim1..
        + 2*dhdtim1)/h3im1;
    Fx = Fx + Fi*cos(thetai) + Fiml*cos(thetaim1);
    Fy = Fy + Fi*sin(thetai) + Fim1*sin(thetaim1);
end
```

Fx $=n u^{*} r^{*} l^{\wedge} 3^{*}$ dtheta/4*Fx; $F y=n u * r^{*} l^{\wedge} 3^{*} d t h e t a / 4 * F y ;$

## ShortHDJBRun.m

```
A program to provide input parameters to function ShortHDJB.m and then
display and plot the results from this function. The program also creates
or adds to a pressure distribution figure for comparison to other bearing
models (figure 1).
Note: *Custom functions ShortHDJB.m and ShortHDJB_Num are required in the
    working directory.
    *User-modifiable parameters/code indicated by :: in comments.
% Author: Armand Krüger
% Created: March 2012
% Last revised: July }201
```


## 

if not (exist('comp','var'))
clear all; clc; close all
else
disp(' ')
end
flag = 1;
while flag
filename = input('Enter the input file name (*.inp): ','s');
filename $=$ [filename '.inp'];
fid $=$ fopen(filename, 'r');
if fid == -1
disp([' Error: file ''',filename,''' not found'])
disp(' Enter filename without extension or ''''')
proc $=$ lower (input(' Proceed (y/n)? ','s'));
if proc == 'n'
disp(' Program terminated')
return
end
else
flag $=0 ;$
end
end
disp(' ')
dummy $=$ fgetl(fid); dummy $=$ fscanf(fid,' $\%$ ', $[3,1])^{\prime}$ ';
$x=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ; ~ d u m m y=$ fgetl(fid); dummy = fscanf(fid,' $\left.\% C^{\prime},[3,1]\right)$ ';
$y=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ; ~ d u m m y=$ fgetl(fid); dummy = fscanf(fid,'\%c',[6,1])';
$x d o t=$ fscanf(fid, '\%f $\left.\backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid); dummy $=\ldots$
fscanf(fid,' \%c', [6,1])';
ydot $=$ fscanf(fid,' $\% f$ $\left.\backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid); dummy $=\ldots$
fscanf(fid, '\%c', [6,1])';
xoff = fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid); dummy $=\ldots$
fscanf(fid,' $\%$ c', $[6,1]$ )';
yoff $=$ fscanf(fid, '\%f $\left.\backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid);
dummy $=$ fgetl(fid); dummy = fscanf(fid,'\%c', [3,1])';
$\mathrm{N}=\mathrm{fscanf}\left(\mathrm{fid}, \mathrm{f} \% \mathrm{f} \backslash \mathrm{n}^{\prime},[1,1]\right) ;$ dummy = fgetl(fid);
dummy $=$ fgetl(fid); dummy $=$ fscanf(fid,' $\%$ ',$[3,1]$ )';
$c=$ fscanf(fid,' $\left.\% f \backslash n^{\prime},[1,1]\right) ; ~ d u m m y=$ fgetl(fid); dummy = fscanf(fid,' $\left.\% c^{\prime},[3,1]\right)$ ';

$d=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ; ~ d u m m y=f g e t l(f i d) ;$
dummy $=$ fgetl(fid); dummy $=$ fscanf(fid,'\%c', $[4,1]$ )';
$n u=f s c a n f\left(f i d, 1 \% f \quad \backslash n^{\prime},[1,1]\right) ;$ dummy $=$ fgetl(fid);
dummy $=$ fgetl(fid); dummy $=$ fscanf(fid,' $\%{ }^{\prime}$ ', 3,1$]$ )';
$n=$ fscanf(fid, ' $\left.\% f \backslash n^{\prime},[1,1]\right) ; ~ d u m m y=f s c a n f\left(f i d, ' \% c^{\prime},[3,1]\right) ' ;$
$m=$ fscanf(fid,'\%f \n', [1,1]);
dummy $=$ fgetl(fid); dummy $=$ fscanf(fid,' $\%$ ', $[5,1]$ )';
cav $=$ fscanf(fid,'\%f $\left.\backslash n^{\prime},[1,1]\right) ;$
fclose(fid);
$\% \frac{2}{\circ} \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \%$ CALCULATIONS $\% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \% \%$
\%BEARING PARAMETERS:
omega $=2 *$ pi*N/60;

```
r = d/2;
D = d + 2*C;
e = sqrt((x-xoff)^2 + (y-yoff)^2);
eps = e/c;
l_d = l/d;
%BEARING PRESSURE DISTRIBUTION & FORCES:
F_int = 1; %O = MATLAB quadl integration; 1 = numerical integration
if F int == 0
    [theta,z,p,Fx,Fy] = ...
        ShortHDJB(x,y,xdot,ydot,xoff,yoff,omega, r, c, l, nu,n,m,cav,1);
else
    [theta,z,p,Fx,Fy] = .. 
        ShortHDJB Num(x,y,xdot,ydot,xoff,yoff,omega,r,c,l,nu,n,m,cav,1);
end
theta = theta/pi*180;
x = x - xoff; y = y - yoff;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%BEARING PARAMETERS:
disp(' '),disp(['Housing diameter: D = ',num2str(D),' mm'])
disp(['Bearing eccentricity: e = ',num2str(e),' mm'])
disp(['Excentricity ratio: epsilon = ',num2str(eps)])
disp(['Length/diameter ratio: l/d = ',num2str(l_d)])
%BEARING FORCES:
disp(['Horizontal force: Fx = ',num2str(Fx),' N'])
disp(['Vertical force: Fy = ',num2str(Fy),' N'])
%3D PRESSURE DISTRIBUTION:
if exist('fig','var')
    fig = fig + 1;
else
    fig = 2;
end
figure(fig)
surf(z,theta,p),title('Journal Bearing Pressure Distribution:')
xlabel('z [mm]'),ylabel('0 [deg]'), zlabel('p [MPa]')
colorbar
%MID-JOURNAL (z=0) CYLINDRICAL PRESSURE DISTRIBUTION:
% Plot Appearance/Adjustment Parameters:
p_scale = 0.4; %::Pressure scale factor relative to d
J_scale = 5; %::Journal eccentricity scale factor
H scale = 1.1; %::Housing radius scale factor
% Housing & Journal:
journal_x = d/2*cosd(theta) + x*J_scale;
journal_y = d/2*sind(theta) + y*J_scale;
housing_x = D/2*cosd(theta)*H scale;
housing_y = D/2*sind(theta)*H_scale;
% Pressure Distribution:
p_mid = p(:,m/2)';
p mid m = max(p mid);
p_mid_x = (p_mi\overline{d}/p_mid_m*d*p_scale + d/2).*cosd(theta) + x*J_scale;
p_mid_y = (p_mid/p_mid_m*d*p_scale + d/2).*sind(theta) + y*J_scale;
% Line of Centres:
lc_x = [-0.6*D/sqrt(1+(y/x)^2) 0.6*D/sqrt(1+(y/x)^2)];
lc_y = y/x*lc_x;
% Velocity Vector:
Vmag = sqrt(xdot^2 + ydot^2);
if Vmag > 0
    xdot v = [0 xdot/Vmag*0.3*D];
    ydot_v = [0 ydot/Vmag*0.3*D];
end
% Force Vector:
Fmag = sqrt(Fx^2 + Fy^2);
if Fmag > 0
    Fx_v = [0 Fx/Fmag*0.2*D];
    Fy_v = [0 Fy/Fmag*0.2*D];
```

```
end
% Plot:
fig = fig + 1;
figure(fig)
handle = plot(journal_x,journal_y,x*J_scale,y*J_scale,'+');
set(handle,'color',[0 0.7 0.9])
hold on
handle = plot(housing_x,housing_y,0,0,'+',lc_x,lc_y,'--');
set(handle,'color',0.\overline{6}*[[1 1 1])
handle = plot(p_mid_x,p_mid_y);
set(handle,'color',[0.5 0.8 0.2])
if Vmag > 0
    plot(xdot v,ydot v,'r')
end
if Fmag > 0
    handle = plot(Fx_v,Fy_v);
    set(handle,'color',[0.5 0.0 0.8])
end
hold off
title('Journal Bearing Geometry & Pressure Distribution:')
xlabel('x [mm]'),ylabel('y [mm]')
if Vmag > 0 && Fmag > 0
    legend('Journal','Journal Centre','Housing','Housing Centre',...
            'Line of Centres',...
            'Pressure [p(0,0)]','Velocity Direction','Force Direction')
elseif Vmag > 0
    legend('Journal','Journal Centre','Housing','Housing Centre',...
            'Line of Centres',...
            'Pressure [p(0,0)]','Velocity Direction')
elseif Fmag > 0
    legend('Journal','Journal Centre','Housing','Housing Centre',...
            'Line of Centres',...
            'Pressure [p(0,0)]','Force Direction')
else
    legend('Journal','Journal Centre','Housing','Housing Centre',...
            'Line of Centres',...
            'Pressure [p(0,0)]')
end
axis equal
%MID-JOURNAL CYLINDRICAL PRESSURE DISTRIBUTION (Polar Plot):
fig = fig + 1;
figure(fig)
lines_theta = zeros(2*length(theta),1); lines_p = lines_theta;
for k = 1:length(theta)
    lines theta(2*k-1) = theta(k)*pi/180;
    lines_theta(2*k) = theta(k)*pi/180;
    if mod}(\textrm{k},2)==
        k1 = 2*k; k2 = 2*k - 1;
    else
            k1 = 2*k - 1; k2 = 2*k;
    end
    lines p(k1) = d/2;
    lines_p(k2) = p_mid(k)/p_mid_m*d*p_scale + d/2;
end
journal_r = d/2*ones(1,length(theta));
handle = polar(lines theta,lines_p);
set(handle,'color',[0 0.7 0.9])
hold on
handle = polar(theta*pi/180,journal_r);
set(handle,'color',0.6*[1 1 1],'LineWidth',1.5)
handle = polar(theta*pi/180,p_mid/p_mid_m*d*p_scale + d/2);
set(handle,'color',[0 0.7 0.9],'LineWidth',1.5)
hold off
title('Journal Geometry & Pressure Distribution:')
legend('Pressure Distribution [p(0,0)]','Journal')
%MID-JOURNAL PRESSURE DISTRIBUTION COMPARISON FIGURE:
```

```
if exist('comp','var')
    comp = comp + 1;
else
    comp = 1;
end
lgnd{comp} = 'Short Bearing Approximation'; %::Method label
figure(1)
hold on
handle = plot(theta,p_mid,'k')
set(handle,'color',[0.5 0.8 0.2]) %::Comparison plot line colour
title('Journal Bearing Centre (z=0) Pressure Distribution:')
xlabel('0 [deg]'),ylabel('p(0,0) [MPa]'),grid on
hold off
if comp > 1
    legend(char(lgnd))
end
```


## Appendix D: MSC.NASTRAN InPuT Files and Entry Additions/Modifications

## D. 1 1D (Free-Free) Rotor Model

## D.1.1 Norm Modes

```
Input File - SOL 103:
$ Normal Modes Analysis, Database
SOL 103
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
    TITLE=This is a default subcase.
    METHOD = 1
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM,GRDPNT,0
EIGRL 1 17 0 MASS
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4.75
$ Pset: "shaft_beam1" will be imported as: "pbeam1.1"
CBEAM 
CBEAM 2 2 1 % 2 % 3 % 0. 1. 
CBEAM 3 1 1 % 3 % 4 % 0. 1. 
CBEAM 4 4 1 % 4 % 5 % 0. 1. 
CBEAM 5 1.llllll
CBEAM [10
CBEAM 
lllllll
```



```
CBEAM 11 
CBEAM 
Clllllll
CBEAM 14 14 1 %
Cllllll
```



```
CBEAM 18 1% 1 
CBEAM 19 19% 1 20 21 
lllllll
CBBEAM 22 1-1 24 25 25 0. 0. 1. 
CBEAM 23 1.1 
CBEAM 24 24 1 % 26 %
CBEAM 
CBEAM 26 1-1 28 29 29 0. 0. 1. 
CBEAM 
CBEAM 
lllllll
```



```
Cllllll
CBEAM 33 1rrrrll
CBEAM 
lllllll
```

| CBEAM | 37 | 1 | 40 | 41 | 0. | 1. | 0. |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| CBEAM | 38 | 1 | 41 | 42 | 0. | 1. | 0. |
| CBEAM | 39 | 1 | 42 | 43 | 0. | 1. | 0. |
| CBEAM | 40 | 1 | 43 | 44 | 0. | 1. | 0. |
| Cbeam | 41 | 1 | 44 | 45 | 0. | 1. | 0. |
| CBEAM | 42 | 1 | 45 | 46 | 0. | 1. | 0. |
| CBEAM | 43 | 1 | 46 | 48 | 0. | 1. | 0. |
| CBEAM | 44 | 1 | 48 | 50 | 0. | 1. | 0. |
| \$ Elements and Element Properties for region : shaft_beam2 |  |  |  |  |  |  |  |
| PBEAML | 2 | 1 |  | ROD |  |  |  |
| 3.9 |  |  |  |  |  |  |  |
| \$ Pset: "shaft_beam2" will be imported as: "pbeam1.2" |  |  |  |  |  |  |  |
| CbEAM | 45 | 2 | 1 | 52 |  | 1. | 0. |
| \$ Elements and Element Properties for region : disk1_mass |  |  |  |  |  |  |  |
| CONM2 | 50 | 21 | 0 | 8.13 |  |  |  |
|  | . 577 |  | . 331136 |  |  | . 331 |  |
| \$ Elements and Element Properties for region : disk2_mass |  |  |  |  |  |  |  |
| CONM2 | 51 | 32 | 0 | 8.10 |  |  |  |
|  | . 572 |  | . 32843 |  |  | . 328 |  |
| \$ Elements and Element Properties for region : journal_mass |  |  |  |  |  |  |  |
| CONM2 | 52 |  | 0 | 8.06 |  |  |  |
| * | . 006 |  |  |  | . 007 |  |  |
| * |  |  | . 0078391 |  |  |  |  |
| \$ Elements and Element Properties for region : coupling_mass |  |  |  |  |  |  |  |
| CONM2 | 53 |  | 0 | 6.53 | $5-.31$ | 0. | 0. |
| * | . 007 |  |  |  | . 006 |  |  |
| * |  |  | . 0061739 |  |  |  |  |
| \$ Referenced Material Records |  |  |  |  |  |  |  |
| \$ Material Record : rotor_steel |  |  |  |  |  |  |  |
| MAT1 | 1 | 210 |  | . 3 | 7.89 |  |  |
| \$ Nodes of the Entire Mode1 |  |  |  |  |  |  |  |
| GRID | 1 |  | 0. | 0. | 0. |  |  |
| GRID | 2 |  | 10.625 | 0. | 0. |  |  |
| GRID | 3 |  | 21.25 | 0. | 0. |  |  |
| GRID | 4 |  | 31.875 | 0. | 0. |  |  |
| GRID | 5 |  | 42.5 | 0. | 0. |  |  |
| GRID | 7 |  | 52.5333 | 0. | 0. |  |  |
| GRID | 8 |  | 62.5667 | 0. | 0. |  |  |
| GRID | 9 |  | 72.6 | 0. | 0. |  |  |
| GRID | 10 |  | 82.6333 | 0. | 0. |  |  |
| GRID | 11 |  | 92.6667 | 0. | 0. |  |  |
| GRID | 12 |  | 102.7 | 0. | 0. |  |  |
| GRID | 13 |  | 112.733 | 0. | 0. |  |  |
| GRID | 14 |  | 122.767 | 0. | 0. |  |  |
| GRID | 15 |  | 132.8 | 0. | 0. |  |  |
| GRID | 16 |  | 142.833 | 0. | 0. |  |  |
| GRID | 17 |  | 152.867 | 0. | 0. |  |  |
| GRID | 18 |  | 162.9 | 0. | 0. |  |  |
| GRID | 19 |  | 172.933 | 0. | 0. |  |  |
| GRID | 20 |  | 182.967 | 0. | 0. |  |  |
| GRID | 21 |  | 193. | 0. | 0. |  |  |
| GRID | 23 |  | 203.05 | 0. | 0. |  |  |
| GRID | 24 |  | 213.1 | 0. | 0. |  |  |
| GRID | 25 |  | 223.15 | 0. | 0. |  |  |
| GRID | 26 |  | 233.2 | 0. | 0. |  |  |
| GRID | 27 |  | 243.25 | 0. | 0. |  |  |
| GRID | 28 |  | 253.3 | 0. | 0. |  |  |
| GRID | 29 |  | 263.35 | 0. | 0. |  |  |
| GRID | 30 |  | 273.4 | 0. | 0. |  |  |
| GRID | 31 |  | 283.45 | 0. | 0. |  |  |
| GRID | 32 |  | 293.5 | 0. | 0. |  |  |
| GRID | 34 |  | 303.827 | 0. | 0. |  |  |
| GRID | 35 |  | 314.154 | 0. | 0. |  |  |
| GRID | 36 |  | 324.481 | 0. | 0. |  |  |
| GRID | 37 |  | 334.808 | 0. | 0. |  |  |
| GRID | 38 |  | 345.135 | 0. | 0. |  |  |
| GRID | 39 |  | 355.462 | 0. | 0. |  |  |
| GRID | 40 |  | 365.789 | 0. | 0. |  |  |
| GRID | 41 |  | 376.115 | 0. | 0. |  |  |


| GRID | 42 | 386.442 | 0. | 0. |
| :--- | :--- | :--- | :--- | :--- |
| GRID | 43 | 396.769 | 0. | 0. |
| GRID | 44 | 407.096 | 0. | 0. |
| GRID | 45 | 417.423 | 0. | 0. |
| GRID | 46 | 427.75 | 0. | 0. |
| GRID | 48 | 435. | 0. | 0. |
| GRID | 50 | 443. | 0. | 0. |
| GRID 52 | -12.5 | 0. | 0. |  |
| \$ Loads for Load Case <br> \$ Referenced Coordinate <br> Default |  |  |  |  |
| ENDDATA |  |  |  |  |

## D.1.2 Model Updating

## Extra Entries/Modifications Only - SOL 200:

The following entry additions and modifications were manually made to the BDF input file included in Appendix D.1.1 from the normal modes analysis discussed in section 3.1.1.1:

File Management Section:
DIAG 8,19

## Executive Control Section:

SOL 200

## Case Control Section:

```
DESOBJ(MIN) = 500
ANALYSIS = MODES
$RESVEC = YES
$VECTOR(SORT1,REAL)=ALL
$SPCFORCES(SORT1,REAL)=ALL
```


## Bulk Data Section:

```
$ PARAMETER OVERRIDES (Output Control):
$
$1-----12------23------34------45------56------67------78---------------99--
$ PARAM1 VAL1 PARAM2 VAL2 PARAM3 VAL3 -etc.-
DOPTPRM P2
$
$ DESIGN VARIABLE(S):
$
$1-----12------23------34------45------56------67-------78----------------9
$ ID LABEL XINIT XLB XUB DELXV DDVAL
DESVAR 1 ELAS1 210000. 100000. 350000.
$1-----12------23------34------45------56------67------78-------89-------
$ ID TYPE MID MFNAME MPMIN CMPMAX C0
$ DVID1 COEF1 DVID2 COEF2 -etc.-
DVMREL1 1 MAT1 1 E
    1.
$
$ DESIGN RESPONSES:
$
$1-----12------23------34------45------56------67------78-------89-------
$ ID LABEL RTYPE PTYPE REGION ATTA ATTB ATT1
DRESP1 101 NFREQ1 FREQ 7
DRESP1 102 NFREQ2 FREQ 9
DRESP1 103 NFREQ3 FREQ 12
DRESP1 104 NFREQ4 FREQ 14
DRESP1 105 NFREQ5 FREQ 16
$
$ ERROR FUNCTION:
$
```



The parameters that correspond between different MSC.Nastran entries are colourcoded/matched. The orange parameter relates to the MAT1 entry.

## D.1.3 Complex Modes

```
Input File - Direct Formulation SOL 107:
```

```
$ Complex Eigenvalue Analysis, Direct Formulation, Database
SOL 107
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
$ Subcase name : Default
        SUBTITLE=Defau7t
        CMETHOD = 1
        VECTOR(SORT1,REAL)=ALL
        SPCFORCES(SORT1,REAL)=ALL
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
EIGC 1 CLAN MAX
    0. 0. 17
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4.75
$ Pset: "shaft_beam1" will be imported as: "pbeam1.1"
CBEAM 1 1 1 1 <lllllll
CBEAM 44 1 4 48 50 0. 1. 
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 
```

```
    . 577345 . 331136 . 331136
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    .572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0
* . 00695402 . 00783913
* .00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* . 00708957 . 00617398
.00617398
$ Referenced Material Records
$ Materia1 Record : rotor_stee1
MAT1 1 210000. .3 7.897-9 . 002
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
GRID 52 0. -12.5 0. 0.
$ Loads for Load Case : Defau7t
$ Referenced Coordinate Frames
ENDDATA
Input File - Modal Formulation SOL 110:
```

```
$ Complex Eigenvalue Analysis, Modal Formulation, Database
SOL }11
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
    TITLE=This is a default subcase.
    METHOD = 1
    CMETHOD = 2
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
EIGRL 1 17 0
EIGC 2 ClAN MAX 17
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4.75
$ Pset: "shaft_beam1" wil1 be imported as: "pbeam1.1"
CBEAM 1 1 <llllllll
CBEAM 44 1 48 50 0. 1. 1. 0.
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 % 21 0
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    . 572857 . }32843\mathrm{ . . }3284
$ Elements and Element Properties for region : journal_mass
```



## D. 2 3D (Free-Free) Rotor Model

## D.2.1 Norm Modes (Before Updating)

```
Input File - SOL 103:
```

\$ Normal Modes Analysis, Database
SOL 103
CEND
\$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
TITLE=This is a default subcase.
METHOD $=1$
VECTOR (SORT1, REAL)=ALL
SPCFORCES (SORT1,REAL)=ALL
\$ Direct Text Input for this Subcase
BEGIN BULK
\$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, GRDPNT, 0
PARAM, PRGPST, NO
EIGRL 1 MASS
\$ Elements and Element Properties for region : rotor_solid
$\begin{array}{llll}\text { PSOLID } & 1 & 1 & 0\end{array}$
$\begin{array}{llrrrrrrl}\text { \$ Pset: } & \text { "rotor_solid" will be imported } & \text { as: } & \text { "psolid.1" } & & \\ \text { CHEXA } & 1 & 1 & 1 & 2 & 6 & 5 & 15 & 16\end{array}$
$\begin{array}{lllllllll}\text { CHEXA } & 1 & 1 & 1 & 2 & 6 & 5 & 15 & 16 \\ & 20 & 19 & & & & & & \end{array}$
$\begin{array}{llllllllll}\text { CHEXA } & 18822 & 1 & 33032 & 32134 & 32157 & 33059 & 33992 & 33094\end{array}$
3311734019
\$ Elements and Element Properties for region : coupling_solid
PSOLID 2 2 $\quad 0$
\$ Pset: "coupling_solid" will be imported as: "psolid.2"
$\begin{array}{llllllllll}\text { CHEXA } & 6337 & 2 & 2664 & 11994 & 11998 & 2666 & 2692 & 12014\end{array}$
120182694

| CHEXA | 6704 | 2 | 12856 | 12733 | 12738 | 12861 | 12866 | 12743 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

    1274812871
    \$ Referenced Material Records
\$ Material Record : rotor_stee1
$\begin{array}{llllll}\text { MAT1 } 1 & 210000 . & \text { 7.897-9 }\end{array}$
\$ Material Record : rotor_iron .3 7.897-9
MAT1 2 120000. 3 6.206-9
\$ Nodes of the Entire Mode1
$\begin{array}{llll}\text { GRID } 1 & 0 . & 0 .\end{array}$
GRID $34019 \quad 306 . \quad 20.1355$-2.64751
\$ Loads for Load Case : Default
\$ Referenced Coordinate Frames
ENDDATA

## D.2.2 Model Updating

## Extra Entries/Modifications Only - SOL 200:

The following entry additions and modifications were manually made to the BDF input file included in Appendix D.2.1 from the normal modes analysis discussed in section 3.1.2.1:

File Management Section:
DIAG 8,19

## Executive Control Section:

SOL 200

## Case Control Section:

```
DESOBJ(MIN) = 500
ANALYSIS = MODES
$RESVEC = YES
$VECTOR(SORT1,REAL)=ALL
$SPCFORCES(SORT1,REAL)=ALL
```


## Bulk Data Section:



| \$ |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$1-----12------23------34------45------56------67------78------89------9 |  |  |  |  |  |  |  |  |
| \$ | ID | LABEL | EQID | REGION | METHOD | C1 | C2 | C3 |
| \$ | DTABLE | LABL1 | LABL2 | LABL3 | LABL4 | LABL5 | LABL6 | -etc.- |
| \$ | DRESP1 | NR1 | NR2 | NR3 | NR4 | NR5 | NR6 | -etc.- |
| \$ | DESVAR | DVID1 | DVID2 | DVID3 | DVID4 | DVID5 | DVID6 | -etc.- |
| DRESP2 | 500 | EFUNC | 100 |  |  |  |  |  |
|  | DESVA | 1 | 2 |  |  |  |  |  |
|  | DTABLE | WT | WM | WR1 | WR2 | WR3 | WR4 | WR5 |
|  |  | WP1 | WP2 | FN1T | FN2T | FN3T | FN4T | FN5T |
|  |  | E1I | E2I |  |  |  |  |  |
|  | DRESP1 | 101 | 102 | 103 | 104 | 105 |  |  |

The parameters that correspond between different MSC.Nastran entries are colourcoded/matched. The orange parameters relate to the MAT1 entries.

## D.2.3 Complex Modes

Input File - Direct Formulation SOL 107:

```
$ Complex Eigenvalue Analysis, Direct Formulation, Database
SOL 107
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
$ Subcase name : Default
        SUBTITLE=Default
        CMETHOD = 1
        VECTOR(SORT1,REAL)=ALL
        SPCFORCES(SORT1,REAL)=ALL
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
EIGC 1 CLAN MAX
    0. 0. 17
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 1 0
$ Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 
    20 19
\begin{tabular}{lllllllll} 
CHEXA & 6704 & 2 & 12856 & 12733 & 12738 & 12861 & 12866 & 12743
\end{tabular}
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 168500. 7.897-9 . 3 002
$ Material Record : rotor_iron
MAT1 2 120000. . % 6 206-9 002
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
GRID 34019 306. 20.1355 -2.64751
$ Loads for Load Case : Default
$ Referenced Coordinate Frames
ENDDATA
```

Input File - Modal Formulation SOL 110:
\$ Complex Eigenvalue Analysis, Modal Formulation, Database
SOL 110

```
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
        TITLE=This is a default subcase.
        METHOD = 1
        CMETHOD = 2
        VECTOR(SORT1,REAL)=ALL
        SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
EIGRL 1 17 0
EIGC 2 ClAN MAX Cl
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 0
```



```
    20 19
CHEXA 
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 168500. 7.897-9 . 3 002
$ Material Record : rotor_iron
MAT1 2 120000. .3 6.206-9 . 002
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
GRID 34019 306. 20.1355 -2.64751
$ Loads for Load Case : Defau7t
$ Referenced Coordinate Frames
ENDDATA
```


## D. 3 1D Rotor System Model with Linear Bearings

## D.3.1 Normal Modes

```
Input File - SOL 103:
$ Norma1 Modes Analysis, Database
SOL 103
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = NO
SUBCASE 1
    TITLE=This is a default subcase.
    METHOD = 1
    SPC = 2
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
EIGRL 1 14 0 MASS
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4.75
$ Pset: "shaft_beam1" wil1 be imported as: "pbeam1.1"
CBEAM 1 1 1 < 1 0
\begin{tabular}{llllllll} 
CBEAM 44 & 1 & 48 & 50 & 0. & 1. & 0.
\end{tabular}
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    . 577345 . }331136 .33113
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    .572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journa1_mass
CONM2 52 46 0
* . 00695402 . 00783913
    .00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* . 00708957 . 00617398
* . 00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" wil1 be imported as: "pe1as.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : bush_spring_z
PELAS 4 209
$ Pset: "bush_spring_z" will be imported as: "pelas.4"
CELAS1 61 4 4
$ Elements and Element Properties for region : journal_spring_y
PELAS 5 250.
$ Pset: "journal_spring_y" will be imported as: "pelas.5"
CELAS1 62 5 101 2
$ Elements and Element Properties for region : journal_spring_z
PELAS 6 250.
$ Pset: "journal_spring_z" wil1 be imported as: "pelas.6"
```

```
CELAS1 63 6 101 3
$ Elements and Element Properties for region : coupling_spring_y
PELAS 7 154.
$ Pset: "coupling_spring_y" wil1 be imported as: "pelas.7"
CELAS1 64 7 102 2
$ Elements and Element Properties for region : coupling_spring_z
PELAS 8 154.
$ Pset: "coupling_spring_z" wil1 be imported as: "pelas.8"
CELAS1 65 8 102 3
$ Elements and Element Properties for region : coupling_spring_theta
PELAS 9 60000.
$ Pset: "coupling_spring_theta" wil1 be imported as: "pelas.9"
CELAS1 66 9
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 67 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 68 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 69 52 123456 102
$ Nodes of the Entire Model
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 5
$ Referenced Coordinate Frames
ENDDATA
```


## D.3.2 Complex Modes

## Input File - Direct Formulation SOL 107:

```
$ Complex Eigenvalue Analysis, Direct Formulation, Database
SOL }10
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
        TITLE=This is a default subcase.
        CMETHOD = 1
        SPC = 2
        VECTOR(SORT1,REAL)=ALL
        SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
EIGC 1 CLAN MAX
    0. 0. 29
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4.75
$ Pset: "shaft_beam1" wil1 be imported as: "pbeam1.1"
```

$\begin{array}{llllllll}\text { CBEAM } & 1 & 1 & 1 & 2 & 0 . & 1 . & 0 .\end{array}$

```
CBEAM 44 1 4 48 50 0. 1. 0.
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    .577345 . }331136\mathrm{ . }33113
$ Elements and Element Properties for region : disk2_mass
CONM2 51 
    . 572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0
* .00695402 .00783913
* . 00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5-.31 0. 0.
    .00708957 . 00617398
    . }0061739
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" wil1 be imported as: "pelas.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : bush_spring_z
PELAS 4 209.
$ Pset: "bush_spring_z" will be imported as: "pelas.4"
CELAS1 61 4 100 3
$ Elements and Element Properties for region : journal_spring_y
PELAS 5 250.
$ Pset: "journal_spring_y" will be imported as: "pelas.5"
CELAS1 62 5
$ Elements and Element Properties for region : journal_spring_z
PELAS 6 250.
$ Pset: "journal_spring_z" will be imported as: "pelas.6"
CELAS1 63 6 101 3
$ Elements and Element Properties for region : coupling_spring_y
PELAS 7 154.
$ Pset: "coupling_spring_y" wil1 be imported as: "pelas.7"
CELAS1 64 7 102 2
$ Elements and Element Properties for region : coupling_spring_z
PELAS 8 154.
$ Pset: "coupling_spring_z" will be imported as: "pelas.8"
CELAS1 65 8 102 3
$ Elements and Element Properties for region : coupling_spring_theta
PELAS 9 60000.
$ Pset: "coupling_spring_theta" will be imported as: "pelas.9"
CELAS1 66 9 4 102 4
$ Elements and Element Properties for region : bush_damp_y
PDAMP 10 . }07
$ Pset: "bush_damp_y" will be imported as: "pdamp.10"
CDAMP1 70 10 100 2
$ Elements and Element Properties for region : bush_damp_z
PDAMP 11 . }07
$ Pset: "bush_damp_z" will be imported as: "pdamp.11"
CDAMP1 71 11 100 3
$ Elements and Element Properties for region : journal_damp_y
PDAMP 12 . }0
$ Pset: "journal_damp_y" will be imported as: "pdamp.12"
CDAMP1 72 12 101 2
$ Elements and Element Properties for region : journal_damp_z
PDAMP 13 . }0
$ Pset: "journal_damp_z" will be imported as: "pdamp.13"
CDAMP1 73 13 101 3
$ Elements and Element Properties for region : coupling_damp_y
PDAMP 14 . }03
$ Pset: "coupling_damp_y" will be imported as: "pdamp.14"
```

```
CDAMP1 74 14 102 2
$ Elements and Element Properties for region : coupling_damp_z
PDAMP 15 . }03
$ Pset: "coupling_damp_z" will be imported as: "pdamp.15"
CDAMP1 75 15 102 3
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_stee1
MAT1 1 210000. .3 7.897-9 . 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 5
$ Referenced Coordinate Frames
ENDDATA
Input File - Modal Formulation SOL 110:
$ Complex Eigenvalue Analysis, Modal Formulation, Database
SOL }11
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
    TITLE=This is a default subcase.
    METHOD = 1
    CMETHOD = 2
    SPC = 2
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
EIGRL 1 29 0
EIGC 2 CLAN MAX 
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4 . 7 5
$ Pset: "shaft_beam1" will be imported as: "pbeam1.1"
CBEAM 1 1 1 1 1 0 0 0.
\begin{tabular}{llllllll} 
CBEAM 44 & 1 & 48 & 50 & 0. & 1. & 0.
\end{tabular}
$ Elements and Element Properties for region : shaft_beam2
```

```
PBEAML 2 1 ROD
            3.9
$ Pset: "shaft_beam2" wil1 be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    .577345 . }331136 .33113
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0 8.104-4
    . 572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journa1_mass
CONM2 52 46 0
* . 00695402 . 00783913
* .00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* .00708957 .00617398
* . 00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" will be imported as: "pelas.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 . 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axia1_constr
SPC1 1 1 1 5
$ Referenced Coordinate Frames
ENDDATA
```


## D.3.3 Critical Speeds

## Input File - Direct Formulation SOL 107:

```
$ Complex Eigenvalue Analysis, Direct Formulation, Database
SOL 107
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
    TITLE=This is a default subcase.
    CMETHOD = 1
    SPC = 2
```

```
    RGYRO = 1
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM,WR4,500.0
EIGC 1. CLAN MAX 
$ Rotor Dynamics data for RGYRO 1
ROTORG 7 1
ROTORG 7 2
ROTORG 7 3
ROTORG 7 4
ROTORG 7 5
ROTORG 7 7
ROTORG 7 8
ROTORG 7 9
ROTORG 7 10
ROTORG 7 11
ROTORG 7 12
ROTORG 7 13
ROTORG 7 14
ROTORG 7 15
ROTORG 7 16
ROTORG 7 17
ROTORG 7 18
ROTORG 7 19
ROTORG 7 20
ROTORG 7 21
ROTORG 7 23
ROTORG 7 24
ROTORG 7 25
ROTORG 7 26
ROTORG 7 27
ROTORG 7 28
ROTORG 7 29
ROTORG 7 30
ROTORG 7 7 31
ROTORG 7 32
ROTORG 7 34
ROTORG 7 35
ROTORG 7 36
ROTORG 7 37
ROTORG 7 38
ROTORG 7 39
ROTORG 7 40
ROTORG 7 41
ROTORG 7 42
ROTORG 7 43
ROTORG 7 44
ROTORG 7 45
ROTORG 7 46
ROTORG 7 48
ROTORG 7 50
ROTORG 7 52
RSPINR 7 21 32 RPM 3000.
RGYRO 1 SYNC 7 RPM 0. 99999.
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4 . 7 5
$ Pset: "shaft_beam1" wil1 be imported as: "pbeam1.1"
CBEAM 1 1 1 1 1 0
```

UNIVERSITEIT YAN PRETORIA
UNIVERSITY OF PRETOR

```
CBEAM 44 1 4 48 50 0. 1. 0.
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 5 < 0 0. 1. 
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    .577345 . }331136 .33113
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0 8.061-5
* . 00695402 . 00783913
* . 00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* . 00708957 .00617398
    .00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" wil1 be imported as: "pe1as.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. 7.897-9 . 3 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 5
$ Referenced Coordinate Frames
ENDDATA
Input File - Modal Formulation SOL 110:
$ Complex Eigenvalue Analysis, Modal Formulation, Database
SOL }11
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
    TITLE=This is a default subcase.
    METHOD = 1
```

```
    CMETHOD = 2
    SPC = 2
RGYRO = 1
VECTOR(SORT1,REAL)=ALL
SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM,WR4,500.0
EIGRL \(1 \quad 170\)
EIGC 2 CLAN MAX 
$ Rotor Dynamics data for RGYRO 1
ROTORG 7 1
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline ROTORG & 7 & 52 & & & & \\
\hline \multirow[t]{2}{*}{RSPINR} & 7 & \multirow[t]{2}{*}{21} & \multirow[t]{2}{*}{32} & \multirow[t]{2}{*}{RPM} & \multirow[t]{2}{*}{3000} & \\
\hline & 0. & & & & & \\
\hline RGYRO & 1 & SYNC & 7 & RPM & 0. & 99999. \\
\hline \multicolumn{2}{|l|}{\$ Elements} & Element & \multicolumn{2}{|l|}{Properties for} & region & shaft_beam1 \\
\hline PBEAML & 1 & 1 & \multicolumn{2}{|r|}{\multirow[t]{2}{*}{ROD}} & & \\
\hline \multicolumn{5}{|c|}{4.75} & & \\
\hline \$ Pset: & & beam1" & wi1 b & port & as: & am.1" \\
\hline CBEAM & 1 & 1 & 1 & 2 & 0. & 1. \\
\hline
\end{tabular}
```



```
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 .002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
```

```
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Model
GRID 1 0. 0. 0.
lllll
GRID 101 435. 0. 0.
GRID 102 -25. 0. 0.
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 5
$ Referenced Coordinate Frames
ENDDATA
```


## D.3.4 Campbell Diagram

## Extra Entry Additions/Modifications (not included in section 3.2.1.3):

The feature to generate a Campbell diagram, it seems, was only introduced in the MSC.Nastran 2005 R3 version. The version of MSC.Patran used, however, does not support this feature. A workaround is to generate the BDF input file for an asynchronous complex modes analysis (at any spin speed) using MSC.Patran and to then add the following entries to the input file manually:

## Case Control Section:

```
CAMPBELL = 1
```


## Bulk Data Section:

| CAMPBLL 1 | SPEED | 10 | RPM |  |
| :---: | :---: | :---: | :---: | :---: |
| DDVAL 10 | 180. | THRU | 30060. BY | 180. |

The CAMPBLL entry is briefly discussed in subsection four of section 1.2.5.3. The DDVAL entry specifies the discrete spin speed values for the generation of the Campbell diagram. To include the rotor material damping in the analysis, the PARAM,WR4,500.0 entry was added to the Bulk Data Section.

```
Input File - Direct Formulation SOL 107:
```

```
$ Complex Eigenvalue Analysis, Direct Formulation, Database
SOL 107
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
    TITLE=This is a default subcase.
    CMETHOD = 1
    SPC = 2
    RGYRO = 1
    CAMPBELL = 1
    SET 1000 = 5
    VECTOR(PRINT, PUNCH, SORT1, REAL)=1000
    $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
```

```
PARAM PRTMAXIM YES
PARAM,WR4,500.0
EIGC 1 HESS MAX
$ Rotor Dynamics data for RGYRO 1
ROTORG 7 1
```



| CBEAM 44 | 1 | 48 | 50 | 0 | 1. | 0 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

\$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 ROD
3.9
\$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
$\begin{array}{llllllll}\text { CBEAM } 45 & 2 & 1 & 52 & 0 . & 1 . & 0 .\end{array}$
\$ Elements and Element Properties for region : disk1_mass
$\begin{array}{lllll}\text { CONM2 } & 50 & 21 & 0 & 8.1364-4\end{array}$
.577345 . 331136 . 331136
\$ Elements and Element Properties for region : disk2_mass
$\begin{array}{llllll}\text { CONM2 } & 51 & 32 & 0 & 8.104-4 & .32843\end{array}$
\$ Elements and Element Properties for region : journal_mass
$\begin{array}{lllll}\text { CONM2 } & 52 & 46 & 0 & 8.061-5\end{array}$

* . 00695402 . 00783913
* . 00783913
\$ Elements and Element Properties for region : coupling_mass
$\begin{array}{lllllll}\text { CONM2 } 53 & 52 & 0 & 6.539-5 & -.31 & 0 . & 0 .\end{array}$
* . 00708957 . 00617398
* . 00617398
\$ Elements and Element Properties for region : bush_spring_y
PELAS 309.
\$ Pset: "bush_spring_y" will be imported as: "pelas.3"
CELAS1 60 3 100 2
\$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16643.
\$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
$\begin{array}{lllll}\text { CDAMP1 } & 76 & 16 & 102 & 4\end{array}$
\$ Referenced Material Records
\$ Material Record : rotor_stee1
MAT1 1 210000. 3 7.897-9 . 002
\$ Multipoint Constraints of the Entire Model
\$ ID conflict : the PATRAN MPC ID was 1
RBE2 $77 \quad 5 \quad 123456100$
\$ ID conflict : the PATRAN MPC ID was 2
RBE2 $78 \quad 48 \quad 123456101$
\$ ID conflict : the PATRAN MPC ID was 3
$\begin{array}{lllll}\text { RBE2 } & 79 & 52 & 123456 & 102\end{array}$
\$ Nodes of the Entire Mode1
$\begin{array}{lllll}\text { GRID } & 1 & 0 & 0 & 0\end{array}$

```
GRID 52 -12.5 0. 0
GRID 100 42.5 0. 0
GRID 101 435. 0. 0.
GRID 102 -25. 0. 0.
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 5
$ Referenced Coordinate Frames
ENDDATA
```


## D.3.5 Frequency Response

## Extra Entry Additions/Modifications (not included in section 3.2.1.4):

The version of MSC.Patran used does not support the use of the UNBALNC entry (briefly discussed in subsection four of section 1.2.5.3) for frequency response analysis (which was only introduced with the release of MD R2 Nastran in 2007). The following additional entries were therefore added to the input files generated by MSC.Patran:

## Case Control Section:

```
DLOAD = 100
```


## Bulk Data Section:

|  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| UNBALNC | 100 | 3.-7 | 21 | 0. | 1. | 0 . |  |
| * | 30.75 |  | 0. |  | 0. |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |
| UNBALNC | 100 | 5.-7 | 32 | 0. | 1. | 0. |  |
| * | 30.75 |  | 90. |  | 0. |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |

Input File - Direct Formulation SOL 108:

```
$ Frequency Response Analysis, Direct Formulation, Database
SOL 108
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
    TITLE=This is a default subcase.
    FREQUENCY = 1
    SPC = 2
    DLOAD = 100
    RGYRO = 1
    SET 1000=5,21, 32,48,52
    DISPLACEMENT (PRINT, PUNCH, SORT1, REAL)=1000
    $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
FREQ1 1 0. 1. 300
$ Rotor Dynamics data for RGYRO 1
ROTORG 7 1
ROTORG 7 52
\begin{tabular}{llllll} 
RSPINR & 7 & 21 & 32 & RPM 3000 .
\end{tabular}
```



```
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 5
$ Referenced Coordinate Frames
$
$
ENDDATA
```

Input File - Modal Formulation SOL 111:

```
$ Frequency Response Analysis, Moda1 Formulation, Database
SOL }11
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
        TITLE=This is a default subcase.
        METHOD = 1
        FREQUENCY = 1
        DLOAD = 100
        SPC = 2
        RGYRO = 1
        SET 1000=5,21,32,48,52
        DISPLACEMENT(PRINT, PUNCH , SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH, SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
FREQ1 1 0. 1. 300
EIGRL 1 0
$ Rotor Dynamics data for RGYRO 1
ROTORG 7 1
```



```
CBEAM 44 1 4 48 50 0. 1. 0
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    . 577345 . }331136 .33113
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    . 572857 . }32843\mathrm{ . . }3284
```

```
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0 8.061-5
* .00695402 . 00783913
* . 00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* . 00708957 .00617398
* .00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" will be imported as: "pelas.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. . 3 7.897-9 . 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 5
$ Referenced Coordinate Frames
$
$
ENDDATA
```


## D.3.6 Nonlinear Transient Response

## Extra Entry Additions/Modifications (not included in section 3.2.1.5):

Before the release of MSC.Nastran 2005 R3, only structural damping (defined on the RSPINR and RSPINT entries) could be specified for the rotor - in this release, additional rotor damping options were added, including Rayleigh damping. The RSPINR and RSPINT entries (refer to subsection four of section 1.2.5.3) were modified and are not compatible with the previous versions. The version of MSC.Patran used in this dissertation, however, still generates the RSPINT entry in its older format. The RSPINT Bulk Data entries in the input files, generated by MSC.Patran, were therefore modified to:

| RSPINT | 8 | 21 | 32 | RPM | 1 |
| :--- | :--- | :--- | :--- | :--- | :--- |

Along with the release of SOL 400 in MSC.Nastran, the Lagrange multiplier rigid elements processing method was implemented that makes use of large rotation theory in geometric
nonlinear analysis, instead of small rotation theory used by the linear method. The Lagrange multiplier method was only set as the default method (instead of the linear method) for SOL 400 from MD.Nastran 2010. The Rotor Dynamics capability of MSC.Nastran only supports the use of linear rigid elements. The version of MSC.Patran used assumes that the linear method is still the default for SOL 400 and therefore the following Case control command has to be added to the MSC.Nastran input files (above SUBCASE level):

```
RIGID = LINEAR
```

The nonlinear transient analysis time stepping (marching) procedure is defined on the TSTEPNL Bulk Data entry. For SOL 129, the default nonlinear iteration method ADAPT is used with a (starting) time step size of 0.001 s . For SOL 400, the full Newton iteration method FNT is used (which is one of the recommended methods for highly nonlinear problems) with a (starting) time step size of $0.0005 s$. Since this method is not available in the version of MSC.Patran used, the TSTEP Bulk Data entry had to be modified as follows:

| \$ | ID | NDT | DT | NO | METHOD | KSTEP | MAXITER | CONV |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |
| TSTEPNL | 1 | 20000 | . 0005 |  | FNT |  |  |  |

To include the rotor material damping in the analyses, the PARAM,WR4,500.0 entry was added to the Bulk Data Sections of the input files.

```
Input File - SOL 129:
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL }12
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
    TITLE=This is a default subcase.
    TSTEPNL = 1
    SPC = 2
    RGYRO = 1
    SET 1000=5,21,32,48,52
    DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
    $SPCFORCES (PRINT, PUNCH , SORT1, REAL)=ALL
    SET 2000=60,61,62,63,64,65,66,70,71,72,73,74,75,76
    FORCE(PRINT, PUNCH, SORT1, REAL,BILIN)=2000
    PARAM TSTATIC -1
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,WR4,500.0
TSTEPNL 1 10000 .001 1 ADAPT 2 10
$ Transient Rotor Dynamics data
ROTORG 8 1
\begin{tabular}{llcccc} 
ROTORG & 8 & 52 & & \\
RSPINT & 8 & 21 & 32 & RPM & 1 \\
UNBALNC & 1 & \(3-7\) & 21 & 0 & 1
\end{tabular}
* 30.75 0. 0. 0.
```

```
* 999999. NONE
\begin{tabular}{llcccccc} 
UNBALNC & 1 & \(5 .-7\) & 32 & 0. & 1. & 0. & \\
\(*\) & 30.75 & & 90. & & 0. & & 0. \\
\(*\) & 999999. & & NONE & & & &
\end{tabular}
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4 . 7 5
$ Pset: "shaft_beam1" will be imported as: "pbeam1.1"
CBEAM 1 1 1 <llllll
\begin{tabular}{llllllll} 
CBEAM 44 & 1 & 48 & 50 & 0. & 1. & 0.
\end{tabular}
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    . 577345 . 331136 . 331136
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    .572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0 8.061-5
* . 00695402 . 00783913
* .00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5-.31 0. 0.
* . 00708957 . 00617398
    .00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" wil1 be imported as: "pelas.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 . 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Mode\
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 5
$ Referenced Transient Rotor (Speed Vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
```

```
-1. 0. 0. 0. 10. 10000. 11. 10000.
    ENDT
$ Referenced Coordinate Frames
ENDDATA
Input File - SOL 400:
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        RGYRO = 1
        SET 1000=5,21,32,48,52
        DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
        $SPCFORCES (PRINT, PUNCH, SORT1, REAL)=ALL
        SET 2000=60,61,62,63,64,65,66,70,71,72,73,74,75,76
        FORCE(PRINT, PUNCH, SORT1, REAL, BILIN)=2000
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,WR4,500.0
$1-----12------23------34------45------56------67------78---------------9
$ ID NDT DT NO METHOD KSTEP MAXITER CONV
$ EPSU EPSP EPSW MAXDIV MAXQN MAXLS FSTRESS
$ MAXBIS ADJUST MSTEP RB MAXR UTOL RTOLB MINITER
TSTEPNL 1 20000 . 0005 FNT
$ Transient Rotor Dynamics data
ROTORG 8 1
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline ROTORG & 8 & 52 & & & & & \\
\hline RSPINT & 8 & 21 & 32 & RPM & 1 & & \\
\hline UNBALNC & 1 & 3.-7 & 21 & & 1. & 0. & \\
\hline * 30 & 30.75 & & 0. & & 0. & & 0. \\
\hline * & 999999. & & NONE & & & & \\
\hline UNBALNC & 1 & 5.-7 & 32 & 0. & 1. & 0. & \\
\hline * & 30.75 & & 90. & & 0. & & 0. \\
\hline * 9 & 999999. & & NONE & & & & \\
\hline \multicolumn{8}{|l|}{\$ Elements and Element Properties for region : shaft_beam1} \\
\hline PBEAML & 1 & 1 & & ROD & & & \\
\hline \multicolumn{8}{|c|}{4.75} \\
\hline \multicolumn{8}{|l|}{\$ Pset: "shaft_beam1" will be imported as: "pbeam1.1"} \\
\hline CBEAM & 1 & 1 & & & & & 0. \\
\hline \multicolumn{8}{|l|}{. 1} \\
\hline \multicolumn{8}{|l|}{} \\
\hline CBEAM & 44 & 1 & 48 & 50 & 0. & 1. & 0. \\
\hline \multicolumn{8}{|l|}{\$ Elements and Element Properties for region : shaft_beam2} \\
\hline \multirow[t]{2}{*}{PBEAML} & 2 & 1 & & ROD & & & \\
\hline & 3.9 & & & & & & \\
\hline \multicolumn{8}{|l|}{\$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"} \\
\hline CbEAM & 45 & 2 & 1 & 52 & 0. & 1. & 0. \\
\hline \multicolumn{8}{|l|}{\$ Elements and Element Properties for region : disk1_mass} \\
\hline \multirow[t]{2}{*}{\$0NM2} & 50 & 21 & 0 & . 136 & & & \\
\hline & . 577345 & & .33113 & & & 33 & \\
\hline
\end{tabular}
```

```
$ Elements and Element Properties for region : disk2_mass
```



```
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0 8.061-5
* . 00695402 . 00783913
* .00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* .00708957 . 00617398
* . 00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" wil1 be imported as: "pelas.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 . 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Model
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axia1_constr
SPC1 1 1 5
$ Referenced Transient Rotor (Speed Vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
                -1. 0. 0. 0. 10. 10000. 11. 10000.
                        ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D.3.7 Gravity as Inertial Loading

```
Input File - Linear Static SOL 101:
$ Linear Static Analysis, Database
SOL 101
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
$ Subcase name : Default
    SUBTITLE=Default
    SPC = 2
    LOAD = 2
    SET 1000=5,48,52
```

```
    DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
    SPCFORCES(SORT1,REAL)=ALL
    STRESS(SORT1,REAL,VONMISES,BILIN)=ALL
    SET 2000=60,61,62,63,64,65,66
    FORCE(PRINT, PUNCH, SORT1, REAL, BILIN)=2000
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
                4.75
$ Pset: "shaft_beam1" will be imported as: "pbeam1.1"
CBEAM 1 1 1 1 1 0 0. 1. 
\begin{tabular}{llllllll} 
CBEAM 44 & 1 & 48 & 50 & 0. & 1. & 0.
\end{tabular}
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
                3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    .577345 . }331136 .33113
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0 % 8.104-4
    .572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0
* . 00695402 . 00783913
* . 00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* .00708957 .00617398
* .00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" wil1 be imported as: "pelas.3"
CELAS1 60 3 100 2
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Materia1 Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 .002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 67 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 68 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 69 52 123456 102
$ Nodes of the Entire Model
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1
```

```
LOAD 2 1. 1. 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 5
$ Gravity Loading of Load Set : grav
GRAV 1 0 9810. 0. -1. 0.
$ Referenced Coordinate Frames
ENDDATA
```

Input File - Nonlinear Transient Response with Initial Conditions SOL 400:

```
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
LOADSET = 1
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        IC = 2
        DLOAD = 3
        RGYRO = 1
        SET 1000=5,21, 32,48,52
        DISPLACEMENT (PRINT, PUNCH, SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
        SET 2000=60,61,62,63,64,65,66,70,71,72,73,74,75,76
        FORCE(PRINT, PUNCH, SORT1, REAL,BILIN)=2000
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,WR4,500.0
$1-----12------23------34------45------56------67------78---------------9
$ ID NDT DT NO METHOD KSTEP MAXITER CONV
$ EPSU EPSP EPSW MAXDIV MAXQN MAXLS FSTRESS
$ MAXBIS ADJUST MSTEP RB MAXR UTOL RTOLB MINITER
TSTEPNL 1 20000 . 0005 FNT
$ Transient Rotor Dynamics data
ROTORG 8 1
```

| ROTORG | 8 | 52 |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| RSPINT | 8 | 21 | 32 | RPM | 1 | 0 |  |
| UNBALNC | 1 | 3.-7 | 21 | 0. | 1. |  |  |
| * | 30.75 |  | 0. |  | 0. |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |
| UNBALNC | 1 | 5.-7 | 32 | 0. | 1. | 0. |  |
| * | 30.75 |  | 90. |  | 0. |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |
| \$ Elements and Element Properties for region : shaft_beam1 |  |  |  |  |  |  |  |
| PBEAML | 1 | 1 |  | ROD |  |  |  |
|  | 4.75 |  |  |  |  |  |  |
| \$ Pset: "shaft_beam1" will be imported as: "pbeam1.1" | "shaft_beam1" will be imported as: "pbeam1.1" |  |  |  |  |  |  |
| $\begin{array}{llllllll}\text { CBEAM } & 1 & 1 & 1 & 2 & 0 . & 1 . & 0 .\end{array}$ |  |  |  |  |  |  |  |
| - |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| CBEAM | 44 | 1 | 48 | 50 | 0. |  | 0. |
| \$ Elements and Element Properties for region : shaft_beam2 |  |  |  |  |  |  |  |
| PBEAML | 2 | 1 |  | ROD |  |  |  |

```
                3.9
$ Pset: "shaft_beam2" wil1 be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 50 % 21 0
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    . 572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0
* . 00695402 . 00783913
* . 00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* . 00708957 . 00617398
    .00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209
$ Pset: "bush_spring_y" wil1 be imported as: "pelas.3"
CELAS1 60 3 100 2
```

```
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_stee1
MAT1 1 210000. .3 7.897-9 .002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
```

| GRID | 52 | -12.5 | 0. | 0. |
| :--- | :--- | :--- | :--- | :--- |
| GRID | 100 | 42.5 | 0. | 0. |
| GRID | 101 | 435. | 0. | 0. |
| GRID | 102 | -25. | 0. | 0. |

GRID $102 \quad-25.00$
\$ Loads for Load Case : Default
SPCADD 21
\$ Initial Displacements of Load Set : initial_disp1



```
$ Gravity Loading of Load Set : grav
GRAV 4 0 9810. 0. -1. 0.
$ Referenced Transient Rotor (Speed vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
    -1. 0. 0. 0. 10. 10000. 11. 10000.
    ENDT
$ Referenced Dynamic Load Tables
$ Constant Load Table
TABLED1 2
    0. 1. 10. 1.
ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D.3.8 Simplified Gravity Loading

```
Input File - Linear Static SOL 101:
```

```
$ Linear Static Analysis, Database
SOL }10
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
$ Subcase name : Default
    SUBTITLE=Default
        SPC = 2
        LOAD = 2
        DISPLACEMENT(SORT1,REAL)=ALL
        SPCFORCES(SORT1,REAL)=ALL
        STRESS(SORT1,REAL,VONMISES,BILIN)=ALL
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4 . 7 5
$ Pset: "shaft_beam1" will be imported as: "pbeam1.1"
CBEAM 1 1 1 1 <llllll
CBEAM 44 1 4 48 50 0. 1. 
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" wil1 be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 0.
$ Elements and Element Properties for region : disk1_mass
CONM2 
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0
    . 572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journa1_mass
CONM2 52 46 0
* . 00695402 .00783913
    00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* . 00708957 . 00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" will be imported as: "pelas.3"
CELAS1 60 3 100 2
```

```
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Materia1 Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 . 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 67 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 68 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 69 52 123456 102
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 5
$ Nodal Forces of Load Set : coupling_grav_force
FORCE 1 102 0 3.6964 0. 0.1. 0.
$ Noda1 Forces of Load Set : bush_grav_force
FORCE 3 100 0 5.577 0. 0.1. 0.
$ Noda1 Forces of Load Set : journal_grav_force
FORCE 4 101 0 10.5696 0. -1. 0.
$ Referenced Coordinate Frames
ENDDATA
Input File - Nonlinear Transient Response with Initial Conditions SOL 400:
```

```
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
```

\$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
SOL 400
CEND
CEND
\$ Direct Text Input for Global Case Control Data
\$ Direct Text Input for Global Case Control Data
ECHO = NONE
ECHO = NONE
LOADSET = 1
LOADSET = 1
RIGID=LINEAR
RIGID=LINEAR
SUBCASE 1
SUBCASE 1
STEP 1
STEP 1
TITLE=This is a default subcase.
TITLE=This is a default subcase.
ANALYSIS = NLTRAN
ANALYSIS = NLTRAN
TSTEPNL = 1
TSTEPNL = 1
SPC = 2
SPC = 2
IC = 2
IC = 2
DLOAD = 3
DLOAD = 3
RGYRO = 1
RGYRO = 1
SET 1000=5,21,32,48,52
SET 1000=5,21,32,48,52
DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
\$SPCFORCES (PRINT, PUNCH, SORT1, REAL)=ALL
\$SPCFORCES (PRINT, PUNCH, SORT1, REAL)=ALL
SET 2000=60,61,62,63,64,65,66,70,71,72,73,74,75,76
SET 2000=60,61,62,63,64,65,66,70,71,72,73,74,75,76
FORCE(PRINT, PUNCH, SORT1, REAL, BILIN)=2000
FORCE(PRINT, PUNCH, SORT1, REAL, BILIN)=2000
\$ Direct Text Input for this Subcase
\$ Direct Text Input for this Subcase
BEGIN BULK
BEGIN BULK
\$ Direct Text Input for Bulk Data
\$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM LGDISP 1
PARAM PRTMAXIM YES

```
PARAM PRTMAXIM YES
```



| ROTORG | 8 | 52 |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| RSPINT | 8 | 21 | 32 | RPM | 1 |  |  |
| UNBALNC | 1 | 3.-7 | 21 | 0. | 1. | 0. |  |
| * | 30.75 |  | 0. |  | 0 . |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |
| UNBALNC | 1 | 5.-7 | 32 | 0. | 1. | 0. |  |
| * | 30.75 |  | 90. |  | 0. |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |
| \$ Elements and Element Properties for region : shaft_beam1 |  |  |  |  |  |  |  |
| PBEAML | 1 | 1 |  | ROD |  |  |  |
|  | $4.75$ |  |  |  |  |  |  |
| \$ Pset: | "shaft_b | beam1' | wil be | orte | as: | 1. |  |
| CBEAM | 1 | 1 | 1 |  | 0. |  |  |



```
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 .002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 77 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 78 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 79 52 123456 102
$ Nodes of the Entire Mode1
```

| GRID | 1 | 0. | 0. | 0. |
| :--- | :--- | :--- | :--- | :--- |
| $\vdots$ |  |  |  |  |
|  |  |  |  |  |
| GRID | 52 | -12.5 | 0. | 0. |
| GRID | 100 | 42.5 | 0. | 0. |
| GRID | 101 | 435. | 0. | 0. |
| GRID | 102 | -25. | 0. | 0. |

\$ Loads for Load Case : Default
SPCADD 2 1
\$ Initial Displacements of Load Set : initial_disp1
TIC* 2 1 2 -. 0249958
*

| TIC* | 2 | 52 | 2 | -.0244992 |
| :--- | :--- | :--- | :--- | :--- |
| $*$ |  |  |  |  |
| TIC* | 2 | 1 | 6 | $-3.97292-5$ |



## D.3.9 Flat Surface Rubbing

## Extra Entries /Modifications Only:

The following entries were manually added to the BDF input files included in Appendix D.3.6, from the nonlinear transient response analyses discussed in section 3.2.1.5, for including flat surface rubbing as discussed in section 3.2.1.7:

## Case Control Section:

```
NONLINEAR = 1
```


## Bulk Data Section:


\$Contact force vs. vertical displacement:

TABLED1 3

| 0. | 0. | .2 | 0. | .21 | 50. | .22 | 120. |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| .23 | 210. | .24 | 320. | .25 | 450. | .26 | 600. |
| .27 | 770. | .28 | 960. | .29 | 1170. | .3 | 1400. |

The nonlinear iteration methods used in SOL 129 and SOL 400, by default, include automatic time step size adjustment based on the dominant response frequency of the structure analysed. This algorithm does not consider the forcing frequency and should be suppressed for short pulse/impact loading (such as rubbing) since the short pulses may not be accurately tracked. In contrast to the SOL 400 FNT iteration method used, the default SOL 129 iteration method ADAPT does not seem to have problems tracking the rubbing forces. Furthermore, for SOL 400, the default error factors used to test for convergence are displacement (U), load (P) and work (W). In this case, the load and work error factors sometimes seem to cause difficulty for convergence, which may be attributed to problematic load error calculation for the "external" forces applied by NOLIN1 entries (which, for a current time step, are calculated with the displacements from the previous time step). The TSTEPNL entry was therefore modified as follows for the SOL 400 simulation:

|  |  |  |  |  |  |  |  | CONV |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | ID | NDT | DT | NO | METHOD | KSTEP | MAXITER |  |  |
| \$ | EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |  |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |  |
| TSTEPNL | 1 | 100000 | 5.-5 | 10 | FNT |  |  | U | +TS101 |
| +TS101 |  |  |  |  |  |  |  |  | +TS102 |
| +TS102 |  | 0 |  |  |  |  |  |  |  |

## D.3.10 Radial Rubbing

Extra Entry Additions/Modifications (not included in section 3.2.1.8):
Extra entries for including radial rubbing as discussed in section 3.2.1.8:

## Case Control Section:

```
NONLINEAR = 1
```

SET 3000=27
NLLOAD (PUNCH) $=3000$

## Bulk Data Section:


\$Gap clearance vs. time:


```
TABLED1 4
    0. .2 1. .2 ENDT
$Coefficient of friction vs. time:
$1-----12------23------34------45------56------67------------------------
TABLED1 5
    0. . 3 1. . 3 ENDT
```

Input File - Nonlinear Transient Response SOL 129:
\$ Nonlinear Transient Response Analysis, Direct Formulation, Database SOL 129
CEND
\$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
TITLE=This is a default subcase.
TSTEPNL = 1
$\mathrm{SPC}=2$
NONLINEAR $=1$
RGYRO = 1
SET $1000=5,21,32,48,52,27$
DISPLACEMENT (PRINT, PUNCH, SORT1, REAL) $=1000$
\$SPCFORCES (PRINT, PUNCH, SORT1, REAL)=ALL
SET 3000=27
NLLOAD (PUNCH) $=3000$
PARAM TSTATIC -1
\$ Direct Text Input for this Subcase
BEGIN BULK
\$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM, WR4,500.0
$\begin{array}{lllllll}\text { TSTEPNL } 1 & 5000 & .001 & 1 & \text { ADAPT } 2 & 10\end{array}$
\$ Transient Rotor Dynamics data
ROTORG 81

. 577345
. 331136
. 331136
\$ Elements and Element Properties for region : disk2_mass $\begin{array}{lllll}\text { CONM2 } & 51 & 32 & 0 & 8.104-4\end{array}$

$$
.572857 \quad .32843 \quad .32843
$$

\$ Elements and Element Properties for region : journal_mass $\begin{array}{lllll}\text { CONM2 } & 52 & 46 & 0 & 8.061-5\end{array}$

* . 00695402 . 00783913
* . 00783913
\$ Elements and Element Properties for region : coupling_mass

| CONM2 53 | 52 | 0 | $6.539-5$ | -.31 | 0. | 0. |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |

* . 00708957 . 00617398
.00617398
\$ Elements and Element Properties for region : bush_spring_y PELAS 3209.
\$ Pset: "bush_spring_y" will be imported as: "pe1as.3"
CELAS1 $60 \quad 3 \quad 100 \quad 2$
\$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16643.
\$ Pset: "coupling_damp_theta" will be imported as: "pdamp.16"
CDAMP1 $76 \quad 16 \quad 1024$
\$ Elements and Element Properties for region : rub_spring_y
PELAS 17 1.-10
\$ Pset: "rub_spring-y" will be imported as: "pelas.17"
$\begin{array}{lllllll}\text { CELAS1 } & 80 & 17 & 103 & 2 & 104 & 2\end{array}$
\$ Elements and Element Properties for region : rub_spring_z
PELAS 18 1.-10
\$ Pset: "rub_spring_z" will be imported as: "pelas.18"
$\begin{array}{lllllll}\text { CELAS1 } & 81 & 18 & 103 & 3 & 104 & 3\end{array}$
\$ Elements and Element Properties for region : rubhousing_spring_y
PELAS 19 1.+10
\$ Pset: "rubhousing_spring_y" will be imported as: "pelas.19"
CELAS1 $82 \quad 19 \quad 104 \quad 2$
\$ Elements and Element Properties for region : rubhousing_spring_z
PELAS 20 1.+10
\$ Pset: "rubhousing_spring_z" will be imported as: "pelas.20"
$\begin{array}{lllll}\text { CELAS1 } & 83 & 20 & 104 & 3\end{array}$
\$ Elements and Element Properties for region : rub_damp_y
PDAMP 21 1.-5
\$ Pset: "rub_damp_y" will be imported as: "pdamp.21"
$\begin{array}{lllllll}\text { CDAMP1 } & 84 & 21 & 103 & 2 & 104 & 2\end{array}$
\$ Elements and Element Properties for region : rub_damp_z
PDAMP 22 1.-5
\$ Pset: "rub_damp_z" wil1 be imported as: "pdamp.22"
$\begin{array}{lllllll}\text { CDAMP1 } & 85 & 22 & 103 & 3 & 104 & 3\end{array}$
\$ Elements and Element Properties for region : rubhousing_damp_y
PDAMP 231.
\$ Pset: "rubhousing_damp_y" will be imported as: "pdamp.23"
CDAMP1 $86 \quad 23 \quad 104$ 2
\$ Elements and Element Properties for region : rubhousing_damp_z
PDAMP 241.
\$ Pset: "rubhousing_damp_z" will be imported as: "pdamp.24"
$\begin{array}{lllll}\text { CDAMP1 } & 87 & 24 & 104 & 3\end{array}$
\$ Referenced Material Records
\$ Material Record : rotor_steel
MAT1 1 210000. 3 7.897-9 . 002
\$ Multipoint Constraints of the Entire Mode
\$ ID conflict : the PATRAN MPC ID was 1
$\begin{array}{lllll}\text { RBE2 } & 88 & 5 & 123456 & 100\end{array}$
\$ ID conflict : the PATRAN MPC ID was 2
$\begin{array}{lllll}\text { RBE2 } & 89 & 48 & 123456 & 101\end{array}$
\$ ID conflict : the PATRAN MPC ID was 3
$\begin{array}{lllll}\text { RBE2 } 90 & 52 & 123456 & 102\end{array}$
\$ ID conflict : the PATRAN MPC ID was 4
$\begin{array}{lllll}\text { RBE2 } & 91 & 27 & 123456 & 103\end{array}$
\$ Nodes of the Entire Model
$\begin{array}{llll}\text { GRID } 1 & 0 & 0\end{array}$

| GRID | 52 | -12.5 | 0. | 0. |
| :--- | :--- | :--- | :--- | :--- |
| GRID | 100 | 42.5 | 0. | 0. |
| GRID | 101 | 435. | 0. | 0. |
| GRID | 102 | -25. | 0. | 0. |
| GRID | 103 | 243.25 | 0. | 0. |
| GRID | 104 | 243.25 | 0. | 0. |

\$ Loads for Load Case : Default
SPCADD 2 1 3
\$ Displacement Constraints of Load Set : axial_constr
SPC1 $1 \quad 1 \quad 1$
\$ Displacement Constraints of Load Set : rubhousing_constr
SPC1 3156104
\$ Referenced Transient Rotor (Speed Vs. Time) Tables
\$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
-1.0 .0 .0 10. 10000. 11. 10000.
ENDT
\$Contact force vs. penetration:

TABLED1 3

| 0. | 0. | .01 | 50. | .02 | 120. | .03 | 320. |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| .05 | 450. | .06 | 600. | .07 | 770. | .08 | 960. |

\$Gap clearance vs. time:

TABLED1 4
$0 . \quad .2$ 1. . 2 ENDT
\$Coefficient of friction vs. time:

TABLED1 5
$0 . \quad .3$ 1. . 3 ENDT
\$ Referenced Coordinate Frames
ENDDATA
Input File - Nonlinear Transient Response SOL 400:

```
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a defau7t subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        NONLINEAR = 1
        RGYRO = 1
        SET 1000=5,21,32,48,52,27
        DISPLACEMENT (PRINT, PUNCH , SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
        SET 3000=27
        NLLOAD(PUNCH)=3000
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,WR4,500.0
$1-----12------23------34------45------56------67------78------89------910-----10
```

| $\$$ | EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |  |
| TSTEPNL | 1 | 100000 | $5 .-5$ | 10 | FNT |  |  | U | +TS101 |
| +TS101 |  |  |  |  |  |  |  |  | +TS102 |

+TS102 0
\$ Transient Rotor Dynamics data ROTORG 81

| ROTORG | 8 | 52 |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| RSPINT | 8 | 21 | 32 | RPM | 1 |  |  |  |
| UNBALNC | 1 | $3 .-7$ | 21 | 0. | 1. | 0. |  |  |
| * | 30.75 |  | 0. |  | 0. |  | 0. |  |
| * | 999999. |  | NONE |  |  |  |  |  |
| UNBALNC | 1 | 5.-7 | 32 | 0. | 1. | 0. |  |  |
| * | 30.75 |  | 90. |  | 0. |  | 0. |  |
| * | 999999. |  | NONE |  |  |  |  |  |
| \$1-----12------23------34------45------56------67------78------89------9 |  |  |  |  |  |  |  |  |
| \$ | SID | GA | GB | PLANE | TABK | TABG | TABU | RADIUS |
| NLRGAP | 1 | 27 | 104 | YZ | -3 | 4 | 5 | 4.75 |
| \$ Elements and Element Properties for region : shaft_beam1 |  |  |  |  |  |  |  |  |
| PBEAML | 1 | 1 |  | ROD |  |  |  |  |
|  | 4.75 |  |  |  |  |  |  |  |
| \$ Pset: | "shaft_beam1" will be imported as: "pbeam1.1" |  |  |  |  |  |  |  |
| CBEAM |  |  |  |  |  |  |  |  |

```
CBEAM 44 1 1 48 50 0. 1. 1. 
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 
$ Elements and Element Properties for region : disk1_mass
CONM2 50 21 0
    .577345 . }331136 .33113
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0 8.104-4
    . 572857 . }32843\mathrm{ . }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0
* . 00695402 . 00783913
    . }0078391
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* .00708957 6.539-5 -.31
* . 00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" will be imported as: "pe1as.3"
CELAS1 60 3 100 2
```

```
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 16 643.
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.16"
CDAMP1 76 16 102 4
$ Elements and Element Properties for region : rub_spring_y
PELAS 17 1.-10
$ Pset: "rub_spring_y" wil1 be imported as: "pelas.17"
CELAS1 80 17 103 2 104 2
$ Elements and Element Properties for region : rub_spring_z
PELAS 18 1.-10
$ Pset: "rub_spring_z" will be imported as: "pelas.18"
CELAS1 81 18 103 3 104 3
$ Elements and Element Properties for region : rubhousing_spring_y
```

```
PELAS 19 1.+10
$ Pset: "rubhousing_spring_y" will be imported as: "pelas.19"
CELAS1 82 19 104 2
$ Elements and Element Properties for region : rubhousing_spring_z
PELAS 20 1.+10
$ Pset: "rubhousing_spring_z" will be imported as: "pelas.20"
CELAS1 83 20 104 3
$ Elements and Element Properties for region : rub_damp_y
PDAMP 21 1.-5
$ Pset: "rub_damp_y" will be imported as: "pdamp.21"
CDAMP1 84 21 103 2 104 2
$ Elements and Element Properties for region : rub_damp_z
PDAMP 22 1.-5
$ Pset: "rub_damp_z" wi11 be imported as: "pdamp.22"
CDAMP1 85 1.22 103 3
$ Elements and Element Properties for region : rubhousing_damp_y
PDAMP 23 1.
$ Pset: "rubhousing_damp_y" will be imported as: "pdamp.23"
CDAMP1 86 23 104 2
$ Elements and Element Properties for region : rubhousing_damp_z
PDAMP 24 1.
$ Pset: "rubhousing_damp_z" will be imported as: "pdamp.24"
CDAMP1 87 24 104 3
$ Referenced Material Records
$ Material Record : rotor_stee1
MAT1 1 210000. .3 7.897-9 . 002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 88 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 89 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 90 52 123456 102
$ ID conflict : the PATRAN MPC ID was 4
RBE2 91 27 123456 103
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
```

| GRID | 52 | -12.5 | 0. | 0. |
| :--- | :--- | :--- | :--- | :--- |
| GRID | 100 | 42.5 | 0. | 0. |
| GRID | 101 | 435. | 0. | 0. |
| GRID | 102 | -25. | 0. | 0. |
| GRID | 103 | 243.25 | 0. | 0. |
| GRID | 104 | 243.25 | 0. | 0. |

\$ Loads for Load Case : Default
$\begin{array}{lcc}\text { SPCADD } & 2 & 1 \\ \$ ~ D i s p l a c e m e n t ~ C o n s t r a i n t s ~ o f ~ L o a d ~ S e t ~: ~ a x i a l \_c o n s t r ~\end{array}$

\$ Displacement Constraints of Load Set : rubhousing_constr
SPC1 3156104
\$ Referenced Transient Rotor (Speed Vs. Time) Tables
\$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
-1.0 .0 .0 10. 10000. 11. 10000.
ENDT
\$Contact force vs. penetration:

TABLED1 3
0. 0. . 01 50. . 02 120. . 03320.

| .05 | 450. | .06 | 600 | .07 | 770. | .08 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |

    . 09 1170. . 1 1400. ENDT
    \$Gap clearance vs. time:

TABLED1 4
0. . 2 1. . 2 ENDT

```
$Coefficient of friction vs. time:
$1-----12------23------34------45------56------67------78----------------9
TABLED1 5
    0. .3 1. . 3 ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D. 4 3D Rotor System Model with Linear Bearings

## D.4.1 Normal Modes

```
Input File - Static Reduction - SOL 103:
$ Normal Modes Analysis, Database
SOL }10
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = NO
SUBCASE 1
    TITLE=This is a default subcase.
    METHOD = 1
    SPC = 2
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST, NO
EIGRL 1 14 0
    MASS
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 0
$ Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 
    20 19
\begin{tabular}{lllllllll} 
CHEXA & 6704 & 2 & 12856 & 12733 & 12738 & 12861 & 12866 & 12743
\end{tabular}
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209. 209.
$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH 50000 3 239 0
$ Elements and Element Properties for region : journal_bush
PBUSH 4 K 250. 250.
$ Pset: "journal_bush" wil1 be imported as: "pbush.4"
CBUSH 51000 4 2563 0
$ Elements and Element Properties for region : coupling_bush
PBUSH 5 K 154. 154. 60000.
$ Pset: "coupling_bush" will be imported as: "pbush.5"
CBUSH 52000 5 40000
    0
$ Referenced Material Records
$ Material Record : rotor_stee1
MAT1 1 168500. .3 7.897-9
$ Material Record : rotor_iron
MAT1 2 120000. 6 6.206-9
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
\begin{tabular}{lllllllll} 
RBE2 & 52001 & 239 & 123 & 240 & 241 & 242 & 243 & 244 \\
& 245 & 246 & 247 & 248 & 249 & 250 & 251 & 252 \\
& 3001 & 3002 & 3003 & 3004 & 3005 & 3006 & 3007 & 3008 \\
& 3009 & 3010 & 5759 & 5760 & 5761 & 5762 & 5763 & 5764 \\
& 5765 & 5766 & 5767 & 5768 & 8518 & 8519 & 8521 & 8522
\end{tabular}
        8524 8525 8526
```

Transient Dynamic Finite Element Modelling of Flexible Rotor Systems


## D.4.2 Complex Modes

## Input File - Modal Formulation SOL 110:

\$ Complex Eigenvalue Analysis, Modal Formulation, Database
SOL 110
CEND
\$ Direct Text Input for Global Case Control Data
ECHO = NONE
RESVEC = YES
SUBCASE 1
TITLE=This is a default subcase.
METHOD $=1$
CMETHOD $=2$
$\mathrm{SPC}=2$
VECTOR (SORT1,REAL)=ALL
SPCFORCES (SORT1,REAL)=ALL
\$ Direct Text Input for this Subcase
BEGIN BULK
\$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST, NO
EIGRL 1 27 0
EIGC 2 CLAN MA
$0 . \quad 0$. 27
\$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 0
\$ Pset: "rotor_solid" will be imported as: "psolid.1"

| universiteitUNIVERSITY OF PRETORIA <br> YUNIBESITHI YA PRETORIA |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Transient Dynamic Finite Element Modelling of Flexible Rotor Systems |  |  |  |  |  |  |  |  |
| CHEXA | $\begin{aligned} & 1 \\ & 20 \end{aligned}$ | $\begin{aligned} & 1 \\ & 19 \end{aligned}$ | 1 | 2 | 6 | 5 | 15 | 16 |
|  |  |  |  |  |  |  |  |  |
| CHEXA | 6704 | 2 | 12856 | 12733 | 12738 | 12861 | 12866 | 12743 |
|  | 12748 | 12871 |  |  |  |  |  |  |
| \$ Elements and Element Properties for region : bush_bush |  |  |  |  |  |  |  |  |
| PBUSH 3 K 209. 209 |  |  |  |  |  |  |  |  |
| \$ Pset: "bush_bush" will be imported as: "pbush.3" |  |  |  |  |  |  |  |  |
| CBUSH | 50000 |  |  |  |  |  |  | 0 |
| \$ Elements and Element Properties for region : journal_bush |  |  |  |  |  |  |  |  |
| PBUSH 4 K 250. 250. |  |  |  |  |  |  |  |  |
| \$ Pset: "journal_bush" will be imported as: "pbush.4" |  |  |  |  |  |  |  |  |
| CBUSH | 51000 | 4 | 2563 |  |  |  |  | 0 |
| \$ Elements and Element Properties for region : coupling_bush |  |  |  |  |  |  |  |  |
| PBUSH |  |  |  | 154. | 154. | 60000. |  |  |
| \$ Pset: "coupling_bush" will be imported as: "pbush.5" |  |  |  |  |  |  |  |  |
| CBUSH | 52000 | 5 | 40000 |  |  |  |  | 0 |
| \$ Elements and Element Properties for region : bush_damp_y |  |  |  |  |  |  |  |  |
| PDAMP | 6 |  |  |  |  |  |  |  |
| \$ Pset: "bush_damp_y" will be imported as: "pdamp.6" |  |  |  |  |  |  |  |  |
| CDAMP1 | 60000 | 6 | 239 | 2 |  |  |  |  |
| \$ Elements and Element Properties for region : bush_damp_z |  |  |  |  |  |  |  |  |
| PDAMP | 7 |  |  |  |  |  |  |  |
| \$ Pset: "bush_damp_z" will be imported as: "pdamp.7" |  |  |  |  |  |  |  |  |
| CDAMP1 | 61000 | 7 | 239 |  |  |  |  |  |
| \$ Elements and Element Properties for region : journal_damp_y |  |  |  |  |  |  |  |  |
| PDAMP | 8 |  |  |  |  |  |  |  |
| \$ Pset: "journal_damp_y" will be imported as: "pdamp.8" |  |  |  |  |  |  |  |  |
| CDAMP1 | 62000 |  | 2563 |  |  |  |  |  |
| \$ Elements and Element Properties for region : journal_damp_z |  |  |  |  |  |  |  |  |
| PDAMP 9 . 03 |  |  |  |  |  |  |  |  |
| \$ Pset: "journal_damp_z" will be imported as: "pdamp.9" |  |  |  |  |  |  |  |  |
| CDAMP1 | 63000 |  | 2563 | 3 |  |  |  |  |
| \$ Elements and Element Properties for region : coupling_damp_y |  |  |  |  |  |  |  |  |
| PDAMP |  | 032 |  |  |  |  |  |  |
| \$ Pset: "coupling_damp_y" will be imported as: "pdamp.10" |  |  |  |  |  |  |  |  |
| CDAMP1 | 64000 | 10 | 40000 | 2 |  |  |  |  |
| \$ Elements and Element Properties for region : coupling_damp_z |  |  |  |  |  |  |  |  |
| PDAMP | 11 | 032 |  |  |  |  |  |  |
| \$ Pset: "coupling_damp_z" will be imported as: "pdamp.11" |  |  |  |  |  |  |  |  |
| CDAMP1 6500011400003 |  |  |  |  |  |  |  |  |
| \$ Elements and Element Properties for region : coupling_damp_theta |  |  |  |  |  |  |  |  |
| PDAMP | 12 | 643. |  |  |  |  |  |  |
| \$ Pset: "coupling_damp_theta" will be imported as: "pdamp.12" |  |  |  |  |  |  |  |  |
| CDAMP1 660001240000 |  |  |  |  |  |  |  |  |
| \$ Referenced Material Records |  |  |  |  |  |  |  |  |
| \$ Material Record : rotor_stee1 |  |  |  |  |  |  |  |  |
| MAT1 | 1 | 68500. | - | . 3 | . 897-9 |  |  |  |
| \$ Material Record : rotor_iron |  |  |  |  |  |  |  |  |
| MAT1 | 2 | 20000. |  | . 3 | 6.206-9 |  |  |  |
| \$ Mu7tipoint Constraints of the Entire Model |  |  |  |  |  |  |  |  |
| \$ ID conflict : the PATRAN MPC ID was 1 |  |  |  |  |  |  |  |  |
| RBE2 | 66001 | 239 | 123 | 240 | 241 | 242 | 243 | 244 |
|  | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 |
|  | 3001 | 3002 | 3003 | 3004 | 3005 | 3006 | 3007 | 3008 |
|  | 3009 | 3010 | 5759 | 5760 | 5761 | 5762 | 5763 | 5764 |
|  | 5765 | 5766 | 5767 | 5768 | 8518 | 8519 | 8521 | 8522 |
|  | 8524 | 8525 | 8526 |  |  |  |  |  |
| \$ ID conflict : the PATRAN MPC ID was 2 |  |  |  |  |  |  |  |  |
| RBE2 | 66002 | 2563 | 123 | 2564 | 2565 | 2566 | 2567 | 2568 |
|  | 2569 | 2570 | 2571 | 2572 | 2573 | 2574 | 2575 | 2576 |
|  | 5325 | 5326 | 5327 | 5328 | 5329 | 5330 | 5331 | 5332 |
|  | 5333 | 5334 | 8083 | 8084 | 8085 | 8086 | 8087 | 8088 |
|  | 8089 | 8090 | 8091 | 8092 | 10842 | 10843 | 10845 | 10846 |
|  | 10848 | 10849 | 10850 | 11198 | 11199 | 11200 | 11201 | 11202 |
|  | 11203 | 11204 | 11205 | 11206 | 11207 | 11208 | 11209 | 11210 |
|  | 11211 | 11212 | 11378 | 11379 | 11380 | 11381 | 11383 | 11384 |



## D.4.3 Critical Speeds

## Extra Entry Additions/Modifications:

The version of MSC.Patran used does not support the use of the ROTORSE entry (only the ROTORG entry) - all the ROTORG Bulk Data entries in the BDF files created by MSC.Patran were therefore (manually) replaced by the following entry (which points to the rotor superelement):

```
ROTORSE 2 1
```

Since hexahedral (CHEXA) elements provide stiffness to only the translational DOFs of its nodes, the AUTOSPC feature of MSC.Nastran (active by default) automatically constrains the rotational DOFs of these nodes (via single point constraints - SPCs) that are not already constrained or connected to elements which provide stiffness to these DOFs. An extra Bulk Data entry PARAM,PRGPST,NO may be added to the BDF file to suppress the output from these automatic constraints to the F06 (*.f06) output file.

## Input File - Direct Formulation SOL 107:

```
$ Complex Eigenvalue Analysis, Direct Formulation, Database
SOL 107
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
    TITLE=This is a default subcase.
    CMETHOD = 1
    SPC = 2
    RGYRO = 1
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
```

```
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
```



```
\begin{tabular}{lllllllll} 
CHEXA & 6704 & 2 & 12856 & 12733 & 12738 & 12861 & 12866 & 12743
\end{tabular}
        12748 12871
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209 209
$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH 50000 3 239 0
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 12 643
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.12"
CDAMP1 66000 12 40000 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 168500. .3 7.897-9
$ Material Record : rotor_iron
MAT1 2 120000. 6.206-9
$ Multipoint Constraints of the Entire Model
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
lllcc
$ rotor_SE
SESET 1 2 THRU 14
SESET 11 34014 34015 34018 34019
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 225
$ Referenced Coordinate Frames
ENDDATA
Input File - Modal Formulation SOL 110:
$ Complex Eigenvalue Analysis, Modal Formulation, Database
SOL }11
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
```

```
RESVEC = YES
SUBCASE 1
    TITLE=This is a default subcase.
    METHOD = 1
    CMETHOD = 2
    SPC = 2
    RGYRO = 1
    VECTOR(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
EIGRL 1 23 0
EIGC 2 ClAN MAX 
$ Rotor Dynamics data for RGYRO 1
ROTORSE 2 1
RSPINR 2 1107 1723 RPM 1.
RGYRO 1 SYNC 2 RPM 0. 99999.
$ Elements and Element Properties for region : rotor_solid
```



```
CHEXA 
    20 19
CHEXA 
    12748 12871
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209. 209.
$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH 50000 3 239
    0
```



```
$ Nodes of the Entire Model
GRID 1 0. 0. 0.
\begin{tabular}{lllccc} 
GRID & 34019 & & 306. & 20.1355 & -2.64751 \\
GRID & 40000 & & -25. & 0. & 0. \\
\$ rotor_SE & & & & \\
SESET & 1 & 2 & THRU & 14
\end{tabular}
SESET 1 1 34014 34015 34018 34019
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axia1_constr
```

```
SPC1 1 1 225
$ Referenced Coordinate Frames
ENDDATA
```


## D.4.4 Campbell Diagram

```
Input File - Direct Formulation SOL 107:
$ Complex Eigenvalue Analysis, Direct Formulation, Database
SOL 107
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
    TITLE=This is a default subcase.
    CMETHOD = 1
    SPC = 2
    RGYRO = 1
    CAMPBELL = 1
    SET 1000 = 1107
    VECTOR(PRINT, PUNCH, SORT1,REAL)=1000
    $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
EIGC 1 HESS MAX
$ Rotor Dynamics data for RGYRO 1
ROTORSE 2 1
RSPINR 2 1107 1723 RPM 1.
RGYRO 1 ASYNC 2 RPM 1.
$1-----12------23------34------45------56------67------78----------------9
CAMPBLL 1 SPEED 10 RPM
DDVAL 10 180. THRU 30060. BY 180.
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 0
$ Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 1 1 1 
    20 19
CHEXA 
    12748 12871
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209. 209.
$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH 50000 3 239 0
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 12 643.
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.12"
CDAMP1 66000 12 40000 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 168500. .3 7.897-9
$ Material Record : rotor_iron
MAT1 2 120000. . 3 6.206-9
$ Multipoint Constraints of the Entire Model
```

```
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
GRID 34019 306. 20.1355 -2.64751
GRID 40000 -25. 0. 0.
$ rotor_SE
SESET 1 2 THRU 14
SESET 1 34014 34015 34018 34019
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axia1_constr
SPC1 1 1 1 225
$ Referenced Coordinate Frames
ENDDATA
```


## D.4.5 Frequency Response

## Extra Entry Additions/Modifications:

As with the 1D rotor system model (refer to section 3.2.1.4 and Appendix D.3.5), the following entries were added/modified:

## Case Control Section:

```
DLOAD = 100
SET 1000=239,1107,1723,2563,40000
DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
$SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
```


## Bulk Data Section:

| \$1-----1 | 12 | 23--- |  |  | 6 | 7 | 8 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| UNBALNC | 100 | 3.-7 | 1107 | 0. | 1. | 0 . |  |
| * | 30.75 |  | 0. |  | 0. |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |
| UNBALNC | 100 | 5.-7 | 1723 | 0. | 1. | 0. |  |
| * | 30.75 |  | 90. |  | 0. |  | 0. |
| * | 999999. |  | NONE |  |  |  |  |

All ROTORG Bulk Data entries were also replaced by:
ROTORSE 2 1
and the Case Control command PARAM,PRGPST,NO was added to the input files to suppress the output from MSC.Nastran's AUTOSPC feature.

Input File - Direct Formulation SOL 108:

```
$ Frequency Response Analysis, Direct Formulation, Database
SOL }10
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
SUBCASE 1
    TITLE=This is a default subcase.
    FREQUENCY = 1
    SPC = 2
    DLOAD = 100
```

```
            RGYRO = 1
            SET 1000=239,1107,1723,2563,40000
            DISPLACEMENT (PRINT, PUNCH , SORT1, REAL)=1000
            $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM POST 0
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
FREQ1 1 0. 1. 300
$ Rotor Dynamics data for RGYRO 1
ROTORSE 2 % 1 1107 1723 RPM 1.
```



```
UNBALNC 100 3.-7 1107 0. 1. 0
* 30.75 0. 0. 0
* 999999. NONE
UNBALNC 100 5.-7 1723 0. 1. 0.
# 30.75 90. 0. 0.
* 999999. NONE
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 1 0 0 % 
$ Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 
    20 19
CHEXA 
    12748 12871
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209. 209.
$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH 50000 3 239
    0
```

```
$ Elements and Element Properties for region : coupling_damp_theta
```

\$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 12 643.
PDAMP 12 643.
\$ Pset: "coupling_damp_theta" will be imported as: "pdamp.12"
\$ Pset: "coupling_damp_theta" will be imported as: "pdamp.12"
CDAMP1 66000 12 40000 4
CDAMP1 66000 12 40000 4
\$ Referenced Material Records
\$ Referenced Material Records
\$ Material Record : rotor_stee1
\$ Material Record : rotor_stee1
MAT1 1 168500. .3 7.897-9
MAT1 1 168500. .3 7.897-9
\$ Material Record : rotor_iron
\$ Material Record : rotor_iron
MAT1 2 120000. . 2 6.206-9
MAT1 2 120000. . 2 6.206-9
\$ Multipoint Constraints of the Entire Model
\$ Multipoint Constraints of the Entire Model
\$ Nodes of the Entire Mode1
\$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
GRID 1 0. 0. 0.
lllcc
lllcc
1 2 THRU
1 2 THRU
1 4
1 4
SESET 1 1 34014 34015 34018 34019
SESET 1 1 34014 34015 34018 34019
\$ Loads for Load Case : Default
\$ Loads for Load Case : Default
SPCADD 2 1
SPCADD 2 1
\$ Displacement Constraints of Load Set : axial_constr

```
$ Displacement Constraints of Load Set : axial_constr
```

```
SPC1 1 1 1 
$ Referenced Coordinate Frames
$
$
ENDDATA
```


## D.4.6 Nonlinear Transient Response

## Extra Entry Additions/Modifications:

As with the 1D rotor system model (refer to section 3.2.1.5 and Appendix D.3.6), the following entries were added/modified:

## Case Control Section:

```
RIGID = LINEAR
SET 1000=239,1107,1723,2563,40000
DISPLACEMENT (PRINT, PUNCH, SORT1, REAL) =1000
$SPCFORCES(PRINT, PUNCH, SORT1,REAL)=ALL
```


## Bulk Data Section:

| RSPINT | 3 | 1107 | 1723 | RPM | 1 |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$1-----12------23------34------45------56------67------78------89------9 |  |  |  |  |  |  |  |  |
| \$ | ID | NDT | DT | NO | METHOD | KSTEP | MAXITER | CONV |
| \$ | EPSU | EPSP | EPSW | MAXDIV | MAXQ | MAXLS | FSTRESS |  |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |
| TSTEPNL | 1 | 20000 | . 0005 |  | FNT |  |  |  |

All ROTORG Bulk Data entries were also replaced by:

```
ROTORSE 3 1
```

and the Case Control command PARAM, PRGPST,NO was added to the input files to suppress the output from MSC.Nastran's AUTOSPC feature.

```
Input File - SOL 400:
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        RGYRO = 1
        SET 1000=239,1107,1723,2563,40000
        DISPLACEMENT (PRINT, PUNCH, SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,PRGPST,NO
```

| \$ | ID | NDT | DT | NO | METHOD | KSTEP | MAXITER | CONV |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |
| TSTEPNL | 1 | 20000 | . 0005 |  | FNT |  |  |  |
| \$ Transient Rotor Dynamics data |  |  |  |  |  |  |  |  |
| ROTORSE 31 |  |  |  |  |  |  |  |  |
| RSPINT | 3 | 1107 | 1723 | RPM | 1 |  |  |  |
| UNBALNC | 100 | 3.-7 | 1107 | 0. | 1. | 0. |  |  |
| * | 30.75 |  | 0. |  | 0. |  | 0. |  |
| * | 999999. |  | NONE |  |  |  |  |  |
| UNBALNC | 100 | 5.-7 | 1723 | 0. | 1. | 0. |  |  |
| * | 30.75 |  | 90. |  | 0. |  | 0. |  |
| * | 999999. |  | NONE |  |  |  |  |  |
| \$ Elements and Element Properties for region : rotor_solid PSOLID $1 \quad 1 \quad 0$ |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| \$ Pset:CHEXA | "rotor_ | solid" w | i11 be | mported | as: "pso | 1id.1" |  |  |
|  | 1 | 1 | 1 | 2 | 6 | 5 | 15 | 16 |
|  | 20 | 19 |  |  |  |  |  |  |
| . |  |  |  |  |  |  |  |  |
| - |  |  |  |  |  |  |  |  |
| CHEXA | 6704 | 2 | 12856 | 12733 | 12738 | 12861 | 12866 | 12743 |
|  | 12748 | 12871 |  |  |  |  |  |  |
| \$ Elements and Element Properties for region : bush_bush |  |  |  |  |  |  |  |  |
| PBUSH | 3 | K |  | 209. | 209. |  |  |  |
| \$ Pset: "bush_bush" will be imported as: "pbush.3" |  |  |  |  |  |  |  |  |
| CBUSH | 50000 | 3 | 239 |  |  |  |  | 0 |

```
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 12 643.
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.12"
CDAMP1 66000 12 40000 4
$ Referenced Material Records
$ Material Record : rotor_stee1
MAT1 1 168500. . 3 7.897-9
$ Material Record : rotor_iron
MAT1 2 120000. .3 6.206-9
$ Multipoint Constraints of the Entire Model
$ Nodes of the Entire Model
GRID 1 0. 0. 0.
\begin{tabular}{lllcc} 
GRID & 34019 & 306. & 20.1355 & -2.64751 \\
GRID & 40000 & -25. & 0. & 0.
\end{tabular}
$ rotor_SE
SESET 1 1 2 THRU 14
SESET 1
$ Loads for Load Case : Default
SPCADD 2 1
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 225
$ Referenced Transient Rotor (Speed Vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
                        -1. 0. 0. 0. 10. 10000. 11. 10000.
                        ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D.4.7 Gravity as Inertial Loading

## Extra Entry Additions/Modifications:

MSC.Patran applies gravity as inertial loading in a transient response analysis by using a combination of DLOAD, TLOAD1, LSEQ, GRAV and TABLED1 Bulk Data entries (accompanied by DLOAD and LOADSET Case Control commands). The TLOAD1 entry defines a time-dependent load in the following form:

$$
\{P(t)\}=\{A\} \cdot F(t)
$$

where $\{P(t)\}$ is the time dependent load vector, $\{A\}$ is the constant-magnitude factor (including the load direction) and $F(t)$ is the scalar time-dependent factor. The LSEQ entry specifies a static load for use in dynamic analysis - in this case, the static load specified by the GRAV entry is used as factor $\{A\}$ through the LSEQ entry. A TABLED1 entry is used to specify $F(t)$ in tabular form - in this case a constant value of unity is used over the duration of the simulation. In newer versions of MSC.Nastran, the LSEQ entry is no longer required to apply static loads in dynamic analyses - the TLOAD1 entry may directly point to the load defined by the GRAV entry to be used as factor $\{A\}$.

The SOL 400 BDF input file created (using MSC.Patran) for the nonlinear transient unbalance run-up response analysis was modified as follows to apply the results from a static analysis step as initial conditions for the transient analysis step that followed:

1. The entry additions/modifications for the nonlinear transient response analyses, discussed in section 3.2.2.5, were incorporated into this BDF file.
2. To include the nonlinear static analysis step before the nonlinear transient analysis step, the following was done:
a. The following Case Control commands were added right above the existing (transient) STEP command:
```
STEP 1
    ANALYSIS = NLSTATIC
    NLPARM = 1
    SPC = 2
    LOAD = 2
    DISPLACEMENT(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
```

b. The number of the existing transient step was changed to 2 (i.e. STEP 2).
c. The following Bulk Data entries were added:

3. Use of the LSEQ and LOADSET entries (as discussed in the preceding paragraph) is not permitted in this multi-step approach involving a static analysis step. The following modifications were therefore made for the nonlinear transient analysis step:
a. The LOADSET and LSEQ entries were removed.
b. The TLOAD1 Bulk Data entry was modified as follows (to directly point to the GRAV entry):

```
$1-----12------23------34------45------56------67-------78----------------9
TLOAD1 4 3 2
```

Input File - Nonlinear Transient Response with Initial Conditions SOL 400:

```
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
$LOADSET = 1
RIGID=LINEAR
SUBCASE 1
    STEP 1
        ANALYSIS = NLSTATIC
        NLPARM = 1
        SPC = 2
        LOAD = 2
        DISPLACEMENT(SORT1,REAL)=ALL
        SPCFORCES(SORT1,REAL)=ALL
    STEP 2
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        DLOAD = 2
        RGYRO = 1
        SET 1000=239,1107,1723,2563,40000
        DISPLACEMENT (PRINT, PUNCH, SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
$1-----12------23------34------45------56------67-------78----------------9
NLPARM 1 10 AUTO
$1-----12------23------34------45------56------67-------78--------89--------9
\begin{tabular}{lllllllll}
\(\$\) & ID & NDT & DT & NO & METHOD & KSTEP & MAXITER CONV \\
\(\$\) & EPSU & EPSP & EPSW & MAXDIV & MAXQN & MAXLS & FSTRESS \\
\$ & MAXBIS & ADJUST & MSTEP & RB & MAXR & UTOL & RTOLB & MINITER
\end{tabular}
```



```
$ Transient Rotor Dynamics data
ROTORSE 3 1
RSPINT 3 1107 1723 RPM 1
UNBALNC 100 3.-7 1107 0. 1. 0.
* 30.75 0. 0. 0.
* UNBALNC 100 59999. 5.-7 NONE N 1723 0. 1. 0. 0. 
* 30.75 90. 0. 0.
* 999999. NONE
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 N Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 1 1 1 
    20 19
CHEXA 6704 2 1 12856 12733 12738
    12748 12871
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209. 209.
$ Pset: "bush_bush" wil1 be imported as: "pbush.3"
CBUSH 50000 3 239 0
```

```
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 12 643.
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.12"
CDAMP1 66000 12 40000 4
$ Referenced Material Records
$ Material Record : rotor_stee1
MAT1 1 168500. .3 7.897-9
$ Material Record : rotor_iron
MAT1 2 120000. 6 6.206-9
$ Multipoint Constraints of the Entire Model
$ Nodes of the Entire Model
GRID 1 0.
0. 0. 0.
\begin{tabular}{lllcc} 
GRID & 34019 & 306. & 20.1355 & -2.64751 \\
GRID & 40000 & -25. & 0. & 0.
\end{tabular}
$ rotor_SE
SESET 1 1 2 THRU 14
```



```
ENDT
$ Referenced Dynamic Load Tables
$ Constant Load Table
TABLED1 2
    0. 1. 10. 1. ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D.4.8 Simplified Gravity Loading

## Extra Entry Additions/Modifications:

Similar to applying gravity as inertial loading (discussed in Appendix D.4.7), MSC. Patran applies the three simplified gravity forces in a transient response analysis by using a combination of one DLOAD, three TLOAD1, three LSEQ, three FORCE and one TABLED1 Bulk Data entries (accompanied by DLOAD and LOADSET Case Control commands). The same approach for applying the results of a static analysis step as initial conditions of a transient analysis step was used for the simplified gravity loading, but with a few adjustments - a SOL 400 BDF input file for a nonlinear transient unbalance run-up response analysis was once again created using MSC.Patran (including the simplified gravity forces) and modified as follows:

The SOL 400 BDF input file created (using MSC.Patran) for the nonlinear transient unbalance run-up response analysis was modified as follows to apply the results from a static analysis step as initial conditions for the transient analysis step that followed:

1. The entry additions/modifications for nonlinear transient response analyses, discussed in section 3.2.2.5, were incorporated into this BDF file.
2. To include the nonlinear static analysis step before the nonlinear transient analysis step, the following was done:
a. The following Case Control commands were added right above the existing (transient) STEP command:
```
STEP 1
    ANALYSIS = NLSTATIC
    NLPARM = 1
    SPC = 2
    LOAD = 2
    DISPLACEMENT(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
```

b. The number of the existing transient step was changed to 2 (i.e. STEP 2).
c. The following Bulk Data entries were added:

3. The following modifications were made for the nonlinear transient analysis step:
a. The LOADSET a three LSEQ entries were removed.
b. The DLOAD Bulk Data entry was modified as follows:

c. The three TLOAD1 Bulk Data entries were modified as follows (to directly point to the three FORCE entries):

| TLOAD1 | 6 | 3 | 2 |
| :---: | :---: | :---: | :---: |
| TLOAD1 | 7 | 4 | 2 |
| TLOAD1 | 8 | 5 | 2 |

Input File - Nonlinear Transient Response with Initial Conditions SOL 400:

```
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
$LOADSET = 1
RIGID=LINEAR
SUBCASE 1
    STEP 1
        ANALYSIS = NLSTATIC
        NLPARM = 1
        SPC = 2
        LOAD = 2
        DISPLACEMENT(SORT1,REAL)=ALL
        SPCFORCES(SORT1,REAL)=ALL
    STEP 2
```

```
    TITLE=This is a default subcase.
    ANALYSIS = NLTRAN
    TSTEPNL = 1
    SPC = 2
    DLOAD = 2
    RGYRO = 1
    SET 1000=239,1107,1723,2563,40000
    DISPLACEMENT (PRINT, PUNCH , SORT1, REAL)=1000
    $SPCFORCES(PRINT, PUNCH,SORT1,REAL)=ALL
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
$1-----12------23------34------45------56------67--------------------------
NLPARM 1 10 AUTO
$1-----12------23------34------45------56------67-------78------89--------
$ EPSU EPSP EPSW MAXDIV MAXQN MAXLS FSTRESS
$ MAXBIS ADJUST MSTEP RB MAXR UTOL RTOLB MINITER
TSTEPNL 1 20000 . 0005 FNT
$ Transient Rotor Dynamics data
ROTORSE 3 1
RClloccle
* 30.75 0. 0. 0.
* 999999. NONE 0 0 0 % 0
```



```
* 999999. NONE 
```



```
$ Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 
    20 19
\begin{tabular}{lllllllll} 
CHEXA & 6704 & 2 & 12856 & 12733 & 12738 & 12861 & 12866 & 12743
\end{tabular}
        12748 12871
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209. 209.
$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH 50000 3 239 0
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 12 643
$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.12"
CDAMP1 66000 12 40000 4
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 168500. .3 7.897-9
$ Material Record : rotor_iron
MAT1 2 120000. 6.206-9
$ Multipoint Constraints of the Entire Model
$ Nodes of the Entire Model
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 34019 & 306. & 20.1355 & -2.64751 \\
GRID & 40000 & -25 & 0. & 0.
\end{tabular}
GRID 40000 -25. 0. 0.
```

```
$ rotor_SE
SESET 1 1 2 THRU 14
SESET 1 34014 34015 34018 34019
$ Loads for Load Case : Default
```



```
LOAD 2 1. 1. 1. 3 1. 1. 1. % 4
DLOAD 1-1. 2 1. 1. 
TLOAD1 6 3 2
TLOAD1 7 7 4 4
TLOAD1 8 5 5 2
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 225
$ Noda1 Forces of Load Set : bush_grav_force
FORCE 3 239 0 5.5698 0. -1. 0.
$ Noda1 Forces of Load Set : journa1_grav_force
FORCE 4 2563 0 10.5835 0. -1. 0.
$ Nodal Forces of Load Set : coupling_grav_force
FORCE 5 40000 0 3.6887 0. -1. 0.
$ Referenced Transient Rotor (Speed Vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
    -1. 0. 0. 0. 10. 10000. 11. 10000.
    ENDT
$ Referenced Dynamic Load Tables
$ Constant Load Table
TABLED1 2
    0. 1. 10. 1. ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D.4.9 Flat Surface Rubbing

## Extra Entry Additions/Modifications (not included in section 3.2.2.7):

As with the 1D rotor system model (refer to section 3.2.1.7 and Appendix D.3.9), the following entries were added/modified to include flat surface rubbing:

## Case Control Section:

```
NONLINEAR = 1
```

SET $1000=239,1107,1723,2563,40000,1415$
SET 4000=1415
NLLOAD (PUNCH) $=4000$

## Bulk Data Section:

| \$ | ID | NDT | DT | NO | METHOD | KSTEP | MAXITER | CONV |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |  |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |  |
| TSTEPNL | 1 | 100000 | 5.-5 | 10 | FNT |  |  | U | +TS101 |
| +TS101 |  |  |  |  |  |  |  |  | +TS102 |
| +TS102 |  | 0 |  |  |  |  |  |  |  |
| \$1----- | 2 | 3--- | 4- | 5 | 56 | 67- | 8------8 | 9------9 |  |
| \$ | SID | GI | CI | S | GJ | CJ | TID |  |  |
| NOLIN1 | 1 | 1415 | 2 | -1. | 1415 | 2 | 3 |  |  |
| NOLIN1 | 1 | 1415 | 3 | -. 3 | 1415 | 2 | 3 |  |  |
| NOLIN1 | 1 | 1415 | 4 | -1.425 | 1415 | 2 | 3 |  |  |

```
$Contact force vs. vertical displacement:
$1-----12------23------34------45------56------67-------78----------------9
TABLED1 }
\begin{tabular}{llllllll}
0. & 0. & .2 & 0. & .21 & 50. & .22 & 120. \\
.23 & 210. & .24 & 320. & .25 & 450. & .26 & 600. \\
.27 & 770. & .28 & 960. & .29 & 1170. & .3 & 1400.
\end{tabular}
```

Input File - Nonlinear Transient Response SOL 400:

```
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        NONLINEAR = 1
        RGYRO = 1
        SET 1000=239,1107,1723,2563,40000,1415
        DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH,SORT1,REAL)=ALL
        SET 4000=1415
        NLLOAD(PUNCH)=4000
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,PRGPST,NO
$1-----12------23------34------45------56------67------78---------------
$ ID NDT DT NO METHOD KSTEP MAXITER CONV
$ EPSU EPSP EPSW MAXDIV MAXQN MAXLS FSTRESS
$ MAXBIS ADJUST MSTEP RB MAXR UTOL RTOLB MINITER
TSTEPNL 1 100000 5.-5 10 FNT U +TS101
+TS101 
+TS102 0
ROTORSE 3 1
llllllll
* 999999. NONE N N N 0. 
#NBALNC 100 %llllllll
* 999999. NONE 
\begin{tabular}{llllllll}
\(\$\) & SID & GI & CI & S & GJ & CJ & TID \\
NOLIN1 & 1 & 1415 & 2 & -1. & 1415 & 2 & 3 \\
NOLIN1 & 1 & 1415 & 3 & -.3 & 1415 & 2 & 3
\end{tabular}
\begin{tabular}{llllllll} 
NOLIN1 & 1 & 1415 & 3 & -.3 & 1415 & 2 & 3 \\
NOL & 1 & 1415 & 4 & -1.425 & 1415 & 2 & 3
\end{tabular}
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 0
$ Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 
        20 19
\begin{tabular}{lllllllll} 
CHEXA & 6704 & 2 & 12856 & 12733 & 12738 & 12861 & 12866 & 12743
\end{tabular}
```

\$ Elements and Element Properties for region : bush_bush PBUSH 3 K 209. 209. \$ Pset: "bush_bush" will be imported as: "pbush.3" CBUSH 500003239
\$ Elements and Element Properties for region : coupling_damp_theta PDAMP 12643.
\$ Pset: "coupling_damp_theta" will be imported as: "pdamp.12"
CDAMP1 $6600012 \quad 400004$
\$ Referenced Material Records
\$ Material Record : rotor_stee1

| MAT1 | 1 | 168500. | .3 | $7.897-9$ |
| :--- | :--- | :---: | :--- | :--- |
| \$ Material | Record $:$ rotor_iron |  |  |  |
| MAT1 | 2 | 120000. | .3 | $6.206-9$ |

\$ ID conflict : the PATRAN MPC ID was 1

| RBE2 | 66001 | 239 | 123 | 240 | 241 | 242 | 243 | 244 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 |
|  | 3001 | 3002 | 3003 | 3004 | 3005 | 3006 | 3007 | 3008 |
|  | 3009 | 3010 | 5759 | 5760 | 5761 | 5762 | 5763 | 5764 |
|  | 5765 | 5766 | 5767 | 5768 | 8518 | 8519 | 8521 | 8522 |

\$ ID conflict : the PATRAN MPC ID was 2

| RBE2 | 66002 | 2563 | 123 | 2564 | 2565 | 2566 | 2567 | 2568 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |


| 2569 | 2570 | 2571 | 2572 | 2573 | 2574 | 2575 | 2576 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 5325 | 5326 | 5327 | 5328 | 5329 | 5330 | 5331 | 5332 |
| 5333 | 5334 | 8083 | 8084 | 8085 | 8086 | 8087 | 8088 |
| 8089 | 8090 | 8091 | 8092 | 10842 | 10843 | 10845 | 10846 |
| 10848 | 10849 | 10850 | 11198 | 11199 | 11200 | 11201 | 11202 |
| 11203 | 11204 | 11205 | 11206 | 11207 | 11208 | 11209 | 11210 |
| 11211 | 11212 | 11378 | 11379 | 11380 | 11381 | 11383 | 11384 |
| 11385 | 11386 | 11388 | 11389 | 11390 | 11391 | 11558 | 11559 |
| 11560 | 11561 | 11563 | 11564 | 11565 | 11566 | 11568 | 11569 |
| 11570 | 11571 | 11739 | 11740 | 11741 | 11744 | 11745 | 11746 |

\$ ID conflict : the PATRAN MPC ID was 3
$\begin{array}{llllllllll}\text { RBE2 } & 66003 & 40000 & 123 & 12743 & 12744 & 12745 & 12746 & 12747\end{array}$

| 12748 | 12749 | 12750 | 12751 | 12752 | 12784 | 12785 | 12786 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 12787 | 12789 | 12790 | 12791 | 12792 | 12824 | 12825 | 12826 |


| 12787 | 12789 | 12890 | 12791 | 12792 | 12824 | 12825 | 12826 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 12869 | 12870 | 12830 | 12831 | 12832 | 12864 | 12865 | 12866 |

\$ ID conflict : the PATRAN MPC ID was 4

| RBE2 | 66004 | 1415 | 123 | 1416 | 1417 | 1418 | 1419 | 1420 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | 1421 | 1422 | 1423 | 1424 | 1425 | 1426 | 1427 | 1428 |
|  | 4177 | 4178 | 4179 | 4180 | 4181 | 4182 | 4183 | 4184 |
|  | 4185 | 4186 | 6935 | 6936 | 6937 | 6938 | 6939 | 6940 |
|  | 6941 | 6942 | 6943 | 6944 | 9694 | 9695 | 9697 | 9698 |

\$ Nodes of the Entire Mode1
GRID 10 .
0.0 .

$\begin{array}{llllll}\text { SESET } & 1 & 34014 & 34015 & 34018 & 34019\end{array}$
\$ Loads for Load Case : Default
SPCADD 21
\$ Displacement Constraints of Load Set : axial_constr
SPC1 $1 \quad 1 \quad 225$
\$ Referenced Transient Rotor (Speed Vs. Time) Tables
\$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1

| $-1$ <br> ENDT | 0. | 0. | 0. | 10. | 10000. 11. |  | 10000. |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$Contact force vs. vertical displacement: |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| TABLED1 3 |  |  |  |  |  |  |  |
| 0. | 0. | . 2 | 0. | . 21 | 50. | . 22 | 120. |
| . 23 | 210. | . 24 | 320. | . 25 | 450. | . 26 | 600. |
| . 27 | 770. | . 28 | 960. | . 29 | 1170. | . 3 | 1400. |
| ENDT |  |  |  |  |  |  |  |
| \$ Referenced | ordin | Fra |  |  |  |  |  |
| ENDDATA |  |  |  |  |  |  |  |

## D.4.10 Radial Rubbing

Extra Entry Additions/Modifications (not included in section 3.2.2.8):
As with the 1D rotor system model (refer to section 3.2.1.8 and Appendix D.3.10), the following entries were added/modified:

## Case Control Section:

```
NONLINEAR = 1
SET 3000=1415
NLLOAD (PUNCH)=3000
```


## Bulk Data Section:

| \$ ID | NDT | DT | NO | METHOD | KSTEP | MAXITER | CONV |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |  |
| \$ MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |  |
| TSTEPNL 1 | 100000 | 5.-5 | 10 | FNT |  |  | U | +TS101 |
| +TS101 |  |  |  |  |  |  |  | +TS102 |
| +TS102 | 0 |  |  |  |  |  |  |  |
| \$1-----12------23------34------45------56------67------78------89-----9 |  |  |  |  |  |  |  |  |
| \$ SID | GA | GB | PLANE | TABK | TABG | TABU | RADIUS |  |
| NLRGAP 1 | 1415 | 41000 | YZ | -3 | 4 | 5 | 4.75 |  |
| \$Contact force vs. penetration: |  |  |  |  |  |  |  |  |
| \$1-----12------ |  |  |  |  |  |  |  |  |
| TABLED1 3 |  |  |  |  |  |  |  |  |
| 0. | 0. | . 01 | 50. | . 02 | 120. | . 03 | 320. |  |
| . 05 | 450. | . 06 | 600. | . 07 | 770. | . 08 | 960. |  |
| . 09 | 1170. | . 1 | 1400. | ENDT |  |  |  |  |
| \$Gap clearance vs. time: |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| TABLED1 4 |  |  |  |  |  |  |  |  |
| 0 . | . 2 | 1. | . 2 | ENDT |  |  |  |  |
| \$Coefficient of friction vs. time: |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| TABLED1 5 |  |  |  |  |  |  |  |  |
| 0 . | . 3 | 1. | . 3 | ENDT |  |  |  |  |

Input File - Nonlinear Transient Response SOL 400:
NASTRAN SYSTEM (316)=19
\$ Nonlinear Transient Response Analysis, Direct Formulation, Database
SOL 400
CEND
\$ Direct Text Input for Global Case Control Data
ECHO = NONE
RIGID=LINEAR

```
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        NONLINEAR = 1
        RGYRO = 1
        SET 1000=239,1107,1723,2563,40000,1415
        DISPLACEMENT (PRINT, PUNCH, SORT1, REAL) =1000
        $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
        SET 3000=1415
        NLLOAD(PUNCH)=3000
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM, PRGPST,NO
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline \$ & ID & NDT & DT & NO & METHOD & KSTEP & MAXITER & CONV \\
\hline \$ & EPSU & EPSP & EPSW & MAXDIV & MAXQN & MAXLS & FSTRESS & \\
\hline \$ & MAXBIS & ADJUST & MSTEP & RB & MAXR & UTOL & RTOLB & MINITER \\
\hline TSTEPNL & 1 & 100000 & 5.-5 & 10 & FNT & & & U \\
\hline +TS101 & & & & & & & & \\
\hline
\end{tabular}
+TS101
+TS102 0
ROTORSE 3 1
RSPINT 3 1107 1723 RPM 1
UNBALNC 100 
* 999999. 
* 30.75 90. 0. 0.
* 999999. NONE N
```



```
NLRGAP 1 
$ Elements and Element Properties for region : rotor_solid
PSOLID 1 1 1 0
$ Pset: "rotor_solid" will be imported as: "psolid.1"
CHEXA 
    20 19
\begin{tabular}{lllllllll} 
CHEXA & 6704 & 2 & 12856 & 12733 & 12738 & 12861 & 12866 & 12743
\end{tabular}
        12748 12871
$ Elements and Element Properties for region : bush_bush
PBUSH 3 K 209. 209.
$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH 50000 3 239 0
```

```
$ Elements and Element Properties for region : coupling_damp_theta
```

\$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 12 643.
PDAMP 12 643.
\$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.12"
\$ Pset: "coupling_damp_theta" wil1 be imported as: "pdamp.12"
CDAMP1 66000 12 40000 4
CDAMP1 66000 12 40000 4
\$ Elements and Element Properties for region : rub_bush
\$ Elements and Element Properties for region : rub_bush
PBUSH 13 K 1.-10 1.-10
PBUSH 13 K 1.-10 1.-10
\$ Pset: "rub_bush" wil1 be imported as: "pbush.13"
\$ Pset: "rub_bush" wil1 be imported as: "pbush.13"
CBUSH 53000 13 1415 41000 0
CBUSH 53000 13 1415 41000 0
\$ Elements and Element Properties for region : rubhousing_bush
\$ Elements and Element Properties for region : rubhousing_bush
PBUSH 14 K 1.+10 1.+10
PBUSH 14 K 1.+10 1.+10
\$ Pset: "rubhousing_bush" will be imported as: "pbush.14"
\$ Pset: "rubhousing_bush" will be imported as: "pbush.14"
CBUSH 54000 14 41000
CBUSH 54000 14 41000
0
0
\$ Elements and Element Properties for region : rub_damp_y

```
$ Elements and Element Properties for region : rub_damp_y
```

| PDAMP 15 1.-5 |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ Pset: "rub_damp_y" will be imported as: "pdamp.15" |  |  |  |  |  |  |  |
| CDAMP1 67000 | 15 | 1415 | 2 | 41000 | 2 |  |  |
| \$ Elements and Element Properties for region : rub_damp_z |  |  |  |  |  |  |  |
| PDAMP 16 | 1.-5 |  |  |  |  |  |  |
| \$ Pset: "rub_damp_z" will be imported as: "pdamp.16" |  |  |  |  |  |  |  |
| CDAMP1 68000 | 16 | 1415 | 3 | 41000 | 3 |  |  |
| \$ Elements and Element Properties for region : rubhousing_damp_y |  |  |  |  |  |  |  |
| PDAMP 17 |  |  |  |  |  |  |  |
| \$ Pset: "rubhousing_damp_y" will be imported as: "pdamp.17" |  |  |  |  |  |  |  |
| CDAMP1 69000 | 17 | 41000 |  |  |  |  |  |
| \$ Elements and Element Properties for region : rubhousing_damp_z |  |  |  |  |  |  |  |
| PDAMP 18 |  |  |  |  |  |  |  |
| \$ Pset: "rubhousing_damp_z" will be imported as: "pdamp.18" |  |  |  |  |  |  |  |
| CDAMP1 70000 | 18 | 41000 |  |  |  |  |  |
| \$ Referenced Material Records |  |  |  |  |  |  |  |
| \$ Material Record : rotor_steel |  |  |  |  |  |  |  |
| MAT1 1 | 168500. |  | . 3 | 7.897-9 |  |  |  |
| \$ Material Record : rotor_iron |  |  |  |  |  |  |  |
| MAT1 2 | 120000. |  | . 3 | 6.206-9 |  |  |  |
| \$ ID conflict : the PATRAN MPC ID was 1 |  |  |  |  |  |  |  |
| RBE2 | 239 | 123 | 240 | 241 | 242 | 243 | 244 |
|  | 246 | 247 | 248 | 249 | 250 | 251 | 252 |
|  | 3002 | 3003 | 3004 | 3005 | 3006 | 3007 | 3008 |
|  | 3010 | 5759 | 5760 | 5761 | 5762 | 5763 | 5764 |
|  | 5766 | 5767 | 5768 | 8518 | 8519 | 8521 | 8522 |
|  | 8525 | 8526 |  |  |  |  |  |
| \$ ID conflict : the PATRAN MPC ID was 2 |  |  |  |  |  |  |  |
| RBE2 70002 | 2563 | 123 | 2564 | 2565 | 2566 | 2567 | 2568 |
| 2569 | 2570 | 2571 | 2572 | 2573 | 2574 | 2575 | 2576 |
| 5325 | 5326 | 5327 | 5328 | 5329 | 5330 | 5331 | 5332 |
| 5333 | 5334 | 8083 | 8084 | 8085 | 8086 | 8087 | 8088 |
| 8089 | 8090 | 8091 | 8092 | 10842 | 10843 | 10845 | 10846 |
| 10848 | 10849 | 10850 | 11198 | 11199 | 11200 | 11201 | 11202 |
| 11203 | 11204 | 11205 | 11206 | 11207 | 11208 | 11209 | 11210 |
| 11211 | 11212 | 11378 | 11379 | 11380 | 11381 | 11383 | 11384 |
| 11385 | 11386 | 11388 | 11389 | 11390 | 11391 | 11558 | 11559 |
| 11560 | 11561 | 11563 | 11564 | 11565 | 11566 | 11568 | 11569 |
| 11570 | 11571 | 11739 | 11740 | 11741 | 11744 | 11745 | 11746 |
| 11749 | 11750 | 11751 |  |  |  |  |  |
| \$ ID conflict : the PATRAN MPC ID was 3 |  |  |  |  |  |  |  |
| RBE2 | 40000 | 123 | 12743 | 12744 | 12745 | 12746 | 12747 |
|  | 12749 | 12750 | 12751 | 12752 | 12784 | 12785 | 12786 |
|  | 12789 | 12790 | 12791 | 12792 | 12824 | 12825 | 12826 |
|  | 12829 | 12830 | 12831 | 12832 | 12864 | 12865 | 12866 |
|  | 12870 | 12871 |  |  |  |  |  |
| \$ ID conflict : the PATRAN MPC ID was 4 |  |  |  |  |  |  |  |
| RBE2 | 1415 | 123 | 1416 | 1417 | 1418 | 1419 | 1420 |
|  | 1422 | 1423 | 1424 | 1425 | 1426 | 1427 | 1428 |
|  | 4178 | 4179 | 4180 | 4181 | 4182 | 4183 | 4184 |
|  | 4186 | 6935 | 6936 | 6937 | 6938 | 6939 | 6940 |
|  | 6942 | 6943 | 6944 | 9694 | 9695 | 9697 | 9698 |
|  |  |  |  |  |  |  |  |
| \$ Nodes of the Entire Mode1 |  |  |  |  |  |  |  |
| GRID 1 |  | 0. | 0. | 0. |  |  |  |
| . |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  | 306. | 20.1355 | -2.64751 |  |  |  |
| $\begin{array}{ll}\text { GRID } & 34019 \\ \text { GRID } & 40000\end{array}$ |  | -25. | 0. | 0. |  |  |  |
| GRID 41000 |  | 243.25 | 0 . | 0 . |  |  |  |
| \$ rotor_SE |  |  |  |  |  |  |  |
| SESET | 1 | 2 THRU | U 14 |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| - ${ }^{\text {c }}$ |  |  |  |  |  |  |  |
| SESET 1 | 13401 | 43401 | 534018 | 3401 |  |  |  |
| \$ Loads for LoadSPCADD2 | d Case : | Default |  |  |  |  |  |
|  | 1 | 3 |  |  |  |  |  |

```
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 225
$ Displacement Constraints of Load Set : rubhousing_constr
SPC1 3 156 41000
$ Referenced Transient Rotor (Speed Vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
    -1. 0. 0. 0. 10. 10000. 11. 10000.
    ENDT
$Contact force vs. penetration:
$1-----12------23------34------45------56------67-------78-------89-------9
TABLED1 }
    0. 0. . 01 50. . 02 120. . 03 320.
    05 450. . 06 600. . 07 770. . 08 960.
    .09 1170. .1 1400. ENDT
$Gap clearance vs. time:
$1-----12------23------34------45------56------67-------------------------
TABLED1 4
    0. .2 1. .2 ENDT
$Coefficient of friction vs. time:
$1-----12------23------34------45------56------67------78----------------9--9
TABLED1 5
    0. . 3 1. . 3 ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D. 5 1D Rotor System Model with Nonlinear Journal Bearing

## D.5.1 Nonlinear Transient Response without Gravity Loading

Extra Entry Additions/Modifications (not included in section 3.5.1.1):
The following entries were added to include the nonlinear journal bearing model via the UDS developed as discussed in section 3.4:

## FMS:

CONNECT SERVICE MyUDS 'SCA.MDSolver.Obj.Uds.Elements.N7rsfd'

## Case Control Section:

```
NONLINEAR = 1
```

SET 3000=48
NLLOAD (PUNCH) $=3000$

## Bulk Data Section:

| $\$ 12-----12-----23------34------45------56------67------78------89-----9$ |  |  |  |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| $\$$ | SID | GA | GB | PLANE | BDIA | BLEN | BCLR | SOLN |
| $\$$ | VISCO | PVAPCO | NPORT | PRES1 | THETA1 | PRES2 | THETA2 | NPNT |
| $\$$ | OFFSET1 | OFFSET2 | GRPNAME | NAME2 |  |  |  |  |
| $\$$ | PARM1 | PARM2 | PARM3 | PARM4 | PARM5 | PARM6 | PARM7 | PARM8 |
| NLRSFD | 1 | 48 | 103 | YZ | 24.5 | 11.5 | 0.2 |  |
|  | $4 .-8$ | 0. | 1 | 0. | 0. |  |  |  |
|  | 0. | 0. | MyUDS |  |  |  |  |  |
|  | 120. | 30. | $1 .-6$ |  |  |  |  |  |

The journal bearing parameters discussed in section 2.3.4 with a pressure norm tolerance of $1 \times 10^{-6} M P a, 120$ mesh divisions in the local bearing model $\theta$-direction ( $n$ ) and 30 mesh divisions in the local bearing model $z$-direction $(m)$ were used for the analysis - a rounded value of 24.5 mm was used for the journal diameter.

As with the flat rubbing simulation, discussed in section 3.2.1.7, the automatic time step size adjustment was disabled (for better tracking of the highly nonlinear journal bearing forces) and the error factors used to test for convergence was changed to the displacement factor ( $U$ ) only to avoid convergence problems due to problematic load error calculations for the external forces supplied through the NLRSFD entry. The TSTEPNL Bulk Data entry was therefore modified as follows:

| \$ | ID | NDT | DT | NO | METHOD | KSTEP | MAXITER | CONV |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |  |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |  |
| TSTEPNL | 1 | 500000 | 1.-5 | 10 | FNT |  |  | U | +TS101 |
| +TS101 |  |  |  |  |  |  |  |  | +TS102 |
| +TS102 |  | 0 |  |  |  |  |  |  |  |

A small time step size of $1 \times 10^{-5} s$ is required for convergence and to accurately track the highly nonlinear journal bearing forces (calculated at each time step based on the previous time step) and resulting displacements.

Input File - SOL 400:
NASTRAN SYSTEM(316)=19
\$ Nonlinear Transient Response Analysis, Direct Formulation, Database

```
CONNECT SERVICE MyUDS 'SCA.MDSolver.Obj.Uds.Elements.N7rsfd'
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        NONLINEAR = 1
        RGYRO = 1
        SET 1000=5,21, 32,48,52
        DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH, SORT1, REAL)=ALL
        SET 3000=48
        NLLOAD(PUNCH)=3000
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,WR4,500.0
$1-----12------23------34------45------56------67------78-------89-------910----10
\begin{tabular}{llllllll}
\(\$\) & ID & NDT & DT & NO & METHOD & KSTEP & MAXITER CONV \\
\(\$\) & EPSU & EPSP & EPSW & MAXDIV & MAXQN & MAXLS & FSTRESS
\end{tabular}
$ MAXBIS ADJUST MSTEP RB MANR MANOL
TSTEPNL 1 500000 1.-5 10 FNT U +TS101
+TS101 
$ Transient Rotor Dynamics data
ROTORG 8 1
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline ROTORG & 8 & 52 & & & & & & \\
\hline RSPINT & 8 & 21 & 32 & RPM & 1 & & & \\
\hline UNBALNC & 1 & 3.-7 & 21 & 0 . & 1. & 0. & & \\
\hline * & 30.75 & & 0. & & 0. & & 0. & \\
\hline * & 999999. & & NONE & & & & & \\
\hline UNBALNC & 1 & 5.-7 & 32 & 0. & 1. & 0. & & \\
\hline * & 30.75 & & 90. & & 0. & & 0. & \\
\hline * & 999999. & & NONE & & & & & \\
\hline \multicolumn{9}{|l|}{} \\
\hline \$ & SID & GA & GB & PLANE & BDIA & BLEN & BCLR & SOLN \\
\hline \$ & VISCO & PVAPCO & NPORT & PRES1 & THETA1 & PRES2 & THETA2 & NPNT \\
\hline \$ & OFFSET1 & OFFSET2 & GRPNAME & NAME2 & & & & \\
\hline \$ & PARM1 & PARM2 & PARM3 & PARM4 & PARM5 & PARM6 & PARM7 & PARM8 \\
\hline \multirow[t]{4}{*}{NLRSFD} & 1 & 48 & 103 & YZ & 24.5 & 11.5 & \multicolumn{2}{|l|}{\multirow[t]{4}{*}{0.2}} \\
\hline & 4.-8 & 0. & 1 & 0. & 0 . & & & \\
\hline & 0. & 0. & MyUDS & & & & & \\
\hline & 120. & 30. & 1.-6 & & & & & \\
\hline
\end{tabular}
$ Elements and Element Properties for region : shaft_beam1
PBEAML 1 1 ROD
    4.75
$ Pset: "shaft_beam1" wil1 be imported as: "pbeam1.1"
CBEAM 1 1 1 1 1 % 2 0. lllllll
CBEAM 44 1 4 48 50 0. 1. 0.
$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 1 ROD
    3.9
$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
CBEAM 45 2 1 1 52 0. 1. 
```

```
$ Elements and Element Properties for region : disk1_mass
CONM2 50 % 21 0
$ Elements and Element Properties for region : disk2_mass
CONM2 51 32 0 8.104-4
    . 572857 . }3284
. }3284
$ Elements and Element Properties for region : journal_mass
CONM2 52 46 0
* .00695402 .00783913
.00783913
$ Elements and Element Properties for region : coupling_mass
CONM2 53 52 0 6.539-5 -.31 0. 0.
* . 00708957 .00617398
* .00617398
$ Elements and Element Properties for region : bush_spring_y
PELAS 3 209.
$ Pset: "bush_spring_y" will be imported as: "pelas.3"
CELAS1 60 % 3 100 2 % for region : bush spring z
PELAS 4 209.
$ Pset: "bush_spring_z" will be imported as: "pelas.4"
CELAS1 61 4 4 100 3
$ Elements and Element Properties for region : coupling_spring_y
PELAS 5 154.
$ Pset: "coupling_spring_y" will be imported as: "pelas.5"
CELAS1 64 5 102 2
$ Elements and Element Properties for region : coupling_spring_z
PELAS 6 154.
$ Pset: "coupling_spring_z" will be imported as: "pelas.6"
CELAS1 65 6 102 3
$ Elements and Element Properties for region : coupling_spring_theta
PELAS 7 60000.
$ Pset: "coupling_spring_theta" will be imported as: "pelas.7"
CELAS1 66 7 7 102 4
$ Elements and Element Properties for region : journal_spring_y
PELAS 8 1.-10
$ Pset: "journa1_spring_y" wil1 be imported as: "pelas.8"
CELAS1 62 % 8 101 2 103 2
$ Elements and Element Properties for region : journal_spring_z
PELAS 9 1.-10
$ Pset: "journal_spring_z" will be imported as: "pelas.9"
CELAS1 63 9 101 3 103 3
$ Elements and Element Properties for region : journalhousing_spring_y
PELAS 10 1.+10
$ Pset: "journalhousing_spring_y" will be imported as: "pelas.10"
CELAS1 67 10 103 2
$ Elements and Element Properties for region : journalhousing_spring_z
PELAS 11 1.+10
$ Pset: "journalhousing_spring_z" will be imported as: "pelas.11"
CELAS1 68 11 103 3
$ Elements and Element Properties for region : bush_damp_y
PDAMP 12 . }07
$ Pset: "bush_damp_y" will be imported as: "pdamp.12"
CDAMP1 70 12 100 2
$ Elements and Element Properties for region : bush_damp_z
PDAMP 13 . }07
$ Pset: "bush_damp_z" wil1 be imported as: "pdamp.13"
CDAMP1 71 13 100 3
$ Elements and Element Properties for region : journal_damp_y
PDAMP 14 1.-5
$ Pset: "journa1_damp_y" wil1 be imported as: "pdamp.14"
CDAMP1 72 14 101 2 103 2
$ Elements and Element Properties for region : journal_damp_z
PDAMP 15 1.-5
$ Pset: "journal_damp_z" wil1 be imported as: "pdamp.15"
CDAMP1 73 15 101 3 103 3
$ Elements and Element Properties for region : coupling_damp_y
PDAMP 16 . }03
$ Pset: "coupling_damp_y" will be imported as: "pdamp.16"
```

```
CDAMP1 74 16 102 2
$ Elements and Element Properties for region : coupling_damp_z
PDAMP 17 . }03
$ Pset: "coupling_damp_z" will be imported as: "pdamp.17"
CDAMP1 75 17 102 3
$ Elements and Element Properties for region : coupling_damp_theta
PDAMP 18 643
$ Pset: "coupling_damp_theta" will be imported as: "pdamp.18"
CDAMP1 76 18 102 4
$ Elements and Element Properties for region : journalhousing_damp_y
PDAMP 19 1.
$ Pset: "journalhousing_damp_y" wil1 be imported as: "pdamp.19"
CDAMP1 77 19 103 2
$ Elements and Element Properties for region : journalhousing_damp_z
PDAMP 20 1.
$ Pset: "journalhousing_damp_z" will be imported as: "pdamp.20"
CDAMP1 78 20 103 3
$ Referenced Material Records
$ Material Record : rotor_steel
MAT1 1 210000. .3 7.897-9 .002
$ Multipoint Constraints of the Entire Model
$ ID conflict : the PATRAN MPC ID was 1
RBE2 79 5 123456 100
$ ID conflict : the PATRAN MPC ID was 2
RBE2 80 48 123456 101
$ ID conflict : the PATRAN MPC ID was 3
RBE2 81 52 123456 102
$ Nodes of the Entire Mode1
GRID 1 0. 0. 0.
\begin{tabular}{lllll} 
GRID & 52 & -12.5 & 0. & 0. \\
GRID & 100 & 42.5 & 0. & 0. \\
GRID & 101 & 435. & 0. & 0. \\
GRID & 102 & -25. & 0. & 0. \\
GRID & 103 & 435. & 0. & 0.
\end{tabular}
$ Loads for Load Case : Default
SPCADD 2 1 3
$ Displacement Constraints of Load Set : axial_constr
SPC1 1 1 1 5
$ Displacement Constraints of Load Set : journalhousing_constr
SPC1 3 1456 103
$ Referenced Transient Rotor (Speed vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
    -1. 0. 0. 0. 10. 10000. 11. 10000.
    ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D.5.2 Nonlinear Transient Response with Gravity Loading and Initial Conditions

## Input File - SOL 400:

```
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
CONNECT SERVICE MyUDS 'SCA.MDSolver.Obj.Uds.Elements.NTrsfd'
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
LOADSET = 1
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
```

```
    TSTEPNL = 1
    SPC = 2
    \(I C=2\)
    NONLINEAR = 1
    DLOAD \(=3\)
    RGYRO \(=1\)
    SET \(1000=5,21,32,48,52\)
    DISPLACEMENT (PRINT, PUNCH, SORT1, REAL) =1000
    \$SPCFORCES (PRINT, PUNCH, SORT1, REAL) =ALL
    SET 3000=48
    NLLOAD (PUNCH) =3000
\$ Direct Text Input for this Subcase
BEGIN BULK
\$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM,WR4, 500.0
\begin{tabular}{llllllllll}
\(\$ \$ 1-----12------23------34------45------56------67------78------89-----910----10\) \\
\(\$ ~\) & ID & NDT & DT & NO & METHOD & KSTEP & MAXITER CONV & & \\
\(\$\) & EPSU & EPSP & EPSW & MAXDIV & MAXQN & MAXLS & FSTRESS & & \\
\(\$\) & MAXBIS & ADJUST & MSTEP & RB & MAXR & UTOL & RTOLB & MINITER & \\
TSTEPNL & 1 & 500000 & \(1 .-5\) & 10 & FNT & & & U & +TS101 \\
+TS101 & & & & & & & & & \\
+TS102
\end{tabular}
S101
+TS102 0
\$ Transient Rotor Dynamics data
ROTORG 81
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline ROTORG & 8 & 52 & & & & & & \\
\hline RSPINT & 8 & 21 & 32 & RPM & 1 & & & \\
\hline UNBALNC & 1 & 3.-7 & 21 & 0 . & 1. & 0. & & \\
\hline * & 30.75 & & 0. & & 0 . & & 0. & \\
\hline * & 999999. & & NONE & & & & & \\
\hline UNBALNC & 1 & 5.-7 & 32 & 0. & 1. & 0. & & \\
\hline * & 30.75 & & 90. & & 0. & & 0. & \\
\hline * & 999999. & & NONE & & & & & \\
\hline \multicolumn{9}{|l|}{} \\
\hline \$ & SID & GA & GB & PLANE & BDIA & BLEN & BCLR & SOLN \\
\hline \$ & VISCO & PVAPCO & NPORT & PRES1 & THETA1 & PRES2 & THETA2 & NPNT \\
\hline \$ & OFFSET1 & OFFSET2 & GRPNAME & NAME2 & & & & \\
\hline \$ & PARM1 & PARM2 & PARM3 & PARM4 & PARM5 & PARM6 & PARM7 & PARM8 \\
\hline \multirow[t]{4}{*}{NLRSFD} & 1 & 48 & 103 & YZ & 24.5 & 11.5 & 0.2 & \\
\hline & 4.-8 & 0. & 1 & 0. & \multicolumn{4}{|l|}{0 .} \\
\hline & 0. & 0. & \multicolumn{6}{|l|}{MyUdS} \\
\hline & 120. & 30. & \multicolumn{6}{|l|}{1.-6} \\
\hline \multicolumn{9}{|l|}{\$ Elements and Element Properties for region : shaft_beam1} \\
\hline PBEAML & 1 & 1 & & \multicolumn{5}{|l|}{ROD} \\
\hline & \multicolumn{8}{|l|}{4.75} \\
\hline \$ Pset: & "shaft_b & beam1" w & i11 be i & ported & as: "pbe & m.1" & & \\
\hline
\end{tabular}
\begin{tabular}{llllllll} 
CBEAM & 44 & 1 & 48 & 50 & 0. & 1. & 0
\end{tabular}
\$ Elements and Element Properties for region : shaft_beam2
PBEAML 2 ROD
    3.9
\$ Pset: "shaft_beam2" will be imported as: "pbeam1.2"
\(\begin{array}{lllllll}\text { CBEAM } 45 & 2 & 1 & 52 & 0 . & 1 . & 0 .\end{array}\)
\$ Elements and Element Properties for region : disk1_mass
\(\begin{array}{lllll}\text { CONM2 } & 50 & 21 & 0 & 8.1364-4\end{array}\)
    .577345 . 331136 . 331136
\$ Elements and Element Properties for region : disk2_mass
\(\begin{array}{lllll}\text { CONM2 } & 51 & 32 & 0 & 8.104-4\end{array}\)
    572857 . 32843 . 32843
\$ Elements and Element Properties for region : journal_mass
\(\begin{array}{lllll}\text { CONM2 } & 52 & 46 & 0 & 8.061-5\end{array}\)
```



Transient Dynamic Finite Element Modelling of Flexible Rotor Systems

```
$ Referenced Dynamic Load Tables
$ Constant Load Table
TABLED1 2
0. 1. 1. 1.
1.
ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D. 6 3D Rotor System Model with Nonlinear Journal Bearing

## D.6.1 Nonlinear Transient Response without Gravity Loading

Extra Entry Additions/Modifications (not included in section 3.5.2.1):
As with the 1D rotor system model (refer to section 3.5.1.1 and Appendix D.5.1), the following entries were added/modified:

## FMS:

```
CONNECT SERVICE MyUDS 'SCA.MDSolver.Obj.Uds.Elements.N1rsfd'
```


## Case Control Section:

```
NONLINEAR = 1
```

SET 3000=2563
NLLOAD (PUNCH) $=3000$

## Bulk Data Section:



Input File - SOL 400:

```
NASTRAN SYSTEM(316)=19
$ Nonlinear Transient Response Analysis, Direct Formulation, Database
CONNECT SERVICE MyUDS 'SCA.MDSolver.Obj.Uds.Elements.Nlrsfd'
SOL 400
CEND
$ Direct Text Input for Global Case Control Data
ECHO = NONE
RIGID=LINEAR
SUBCASE 1
    STEP 1
        TITLE=This is a default subcase.
        ANALYSIS = NLTRAN
        TSTEPNL = 1
        SPC = 2
        NONLINEAR = 1
        RGYRO = 1
        SET 1000=239,1107,1723,2563,40000
        DISPLACEMENT(PRINT, PUNCH, SORT1, REAL)=1000
        $SPCFORCES(PRINT, PUNCH, SORT1,REAL)=ALL
```

SET 3000=2563
NLLOAD (PUNCH)=3000
\$ Direct Text Input for this Subcase
BEGIN BULK
\$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM, PRGPST, NO

|  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | ID | NDT | DT | NO | METHOD | KSTEP | MAXITER | CONV |  |
| \$ | EPSU | EPSP | EPSW | MAXDIV | MAXQN | MAXLS | FSTRESS |  |  |
| \$ | MAXBIS | ADJUST | MSTEP | RB | MAXR | UTOL | RTOLB | MINITER |  |
| TSTEPNL | 1 | 500000 | 1.-5 | 10 | FNT |  |  | U | +TS101 |
| +TS101 |  |  |  |  |  |  |  |  | +TS102 |

+TS102
\$ Transient
Rotor
Dynamics data
ROTORSE 31

| RSPINT | 3 | 1107 | 1723 | RPM | 1 |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| UNBALNC | 100 | 3.-7 | 1107 | 0 . | 1. | 0. |  |  |
| * | 30.75 |  | 0. |  | 0 . |  | 0. |  |
| * | 999999. |  | NONE |  |  |  |  |  |
| UNBALNC | 100 | 5. -7 | 1723 | 0. | 1. | 0. |  |  |
| * | 30.75 |  | 90. |  | 0. |  | 0. |  |
| * | 999999. |  | NONE |  |  |  |  |  |
| \$1-----12------23------34------45------56------67------78------89------9 |  |  |  |  |  |  |  |  |
| \$ | SID | GA | GB | PLANE | BDIA | BLEN | BCLR | SOLN |
| \$ | VISCO | PVAPCO | NPORT | PRES1 | THETA1 | PRES2 | THETA2 | NPNT |
| \$ | OFFSET1 | OFFSET2 | GRPNAME | NAME2 |  |  |  |  |
| \$ | PARM1 | PARM2 | PARM3 | PARM4 | PARM5 | PARM6 | PARM7 | PARM8 |
| NLRSFD | 1 | 2563 | 41000 | YZ | 24.5 | 11.5 | 0.2 |  |
|  | 4.-8 | 0. | 1 | 0. | 0 . |  |  |  | $0 . \quad 0 . \quad$ MyUDS 120. 30. 1.-6

\$ Elements and Element Properties for region : rotor_solid PSOLID $1 \quad 1 \quad 0$
\$ Pset: "rotor_solid" will be imported as: "psolid.1"
$\begin{array}{llllllllll}\text { CHEXA } & 1 & 1 & 1 & 2 & 6 & 5 & 15 & 16\end{array}$ 2019

| CHEXA | 6704 | 2 | 12856 | 12733 | 12738 | 12861 | 12866 | 12743 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | 1274812871

\$ Elements and Element Properties for region : bush_bush PBUSH 3 K 209. 209.
\$ Pset: "bush_bush" will be imported as: "pbush.3"
CBUSH $500003239 \quad 0$
\$ Elements and Element Properties for region : coupling_bush
PBUSH 4 K 154. 154. 60000.
\$ Pset: "coupling_bush" will be imported as: "pbush.4"
CBUSH 52000440000
\$ Elements and Element Properties for region : journal_bush
PBUSH 5 K 1.-10 1.-10
\$ Pset: "journal_bush" will be imported as: "pbush.5"
CBUSH $5100050256341000 \quad 0$
\$ Elements and Element Properties for region : journalhousing_bush
PBUSH 6 K $1 .+101 .+10$
\$ Pset: "journalhousing_bush" wi11 be imported as: "pbush.6"
CBUSH 53000641000
0
\$ Elements and Element Properties for region : bush_damp_y
PDAMP 7 . 074
\$ Pset: "bush_damp_y" will be imported as: "pdamp.7"
CDAMP1 6000072392
\$ Elements and Element Properties for region : bush_damp_z
PDAMP 8 . 074
\$ Pset: "bush_damp_z" will be imported as: "pdamp.8"
CDAMP1 61000832393
\$ Elements and Element Properties for region : journal_damp_y


```
SESET 1 34014 34015 34018 34019
$ Loads for Load Case : Default
SPCADD 2 1 3
$ Displacement Constraints of Load Set : axia1_constr
SPC1 1 1 225
$ Displacement Constraints of Load Set : journalhousing_constr
SPC1 3 1456 41000
$ Referenced Transient Rotor (Speed Vs. Time) Tables
$ Dynamic Load Table : spinspeed_field (scale 1.)
TABLED1 1
    -1. 0. 0. 0. 10. 10000. 11. 10000.
        ENDT
$ Referenced Coordinate Frames
ENDDATA
```


## D.6.2 Nonlinear Transient Response with Gravity Loading and Initial Conditions

## Extra Entry Additions/Modifications:

The SOL 400 nonlinear transient response analysis BDF input file (created using MSC.Patran) was modified as follows :

1. The entry additions/modifications for nonlinear transient response analyses, discussed in section 3.2.2.5, were incorporated into this BDF file.
2. The entry additions/modifications from section 3.5.2.1 were also incorporated into this BDF file.
3. To include the nonlinear static analysis step with the journal bearing equilibrium forces before the nonlinear transient analysis step, the following was done:
a. The following Case Control commands were added right above the existing (transient) STEP command:
```
STEP 1
    ANALYSIS = NLSTATIC
    NLPARM = 1
    SPC = 2
    LOAD = 2
    DISPLACEMENT(SORT1,REAL)=ALL
    SPCFORCES(SORT1,REAL)=ALL
```

b. The number of the existing transient step was changed to 2 (i.e. STEP 2).
c. The following Bulk Data entries were added:

4. The following modifications were made for the nonlinear transient analysis step:
a. The LOADSET and three LSEQ entries were removed.
b. The DLOAD Bulk Data entry was modified as follows:

c. The three TLOAD1 Bulk Data entries were modified as follows (to directly point to the three FORCE entries):


Input File - SOL 400:
NASTRAN SYSTEM(316)=19
\$ Nonlinear Transient Response Analysis, Direct Formulation, Database
CONNECT SERVICE MyUDS 'SCA.MDSolver.Obj.Uds.Elements.N1rsfd'
SOL 400
CEND
\$ Direct Text Input for Global Case Control Data
ECHO = NONE
\$LOADSET = 1
RIGID=LINEAR
SUBCASE 1
STEP 1
ANALYSIS = NLSTATIC
NLPARM $=1$
SPC = 2
LOAD = 2
DISPLACEMENT (SORT1,REAL)=ALL
SPCFORCES(SORT1,REAL)=ALL
STEP 2
TITLE=This is a default subcase.
ANALYSIS $=$ NLTRAN
TSTEPNL = 1
$\mathrm{SPC}=2$
NONLINEAR $=1$
DLOAD $=2$
RGYRO $=1$
SET 1000=239,1107,1723,2563,40000
DISPLACEMENT (PRINT, PUNCH, SORT1, REAL) $=1000$
\$SPCFORCES (PRINT, PUNCH, SORT1, REAL)=ALL
SET 3000=2563
NLLOAD (PUNCH)=3000
\$ Direct Text Input for this Subcase
BEGIN BULK
\$ Direct Text Input for Bulk Data
PARAM AUTOSPC NO
PARAM LGDISP 1
PARAM PRTMAXIM YES
PARAM, PRGPST, NO


$\$$ ID NDT DT NO METHOD KSTEP MAXITER CONV
\$ EPSU EPSP EPSW MAXDIV MAXQN MAXLS FSTRESS
\$ MAXBIS ADJUST MSTEP RB MAXR UTOL RTOLB MINITER
$\begin{array}{lllllll}\text { TSTEPNL } 1 & 500000 & 1 .-5 & 10 & \text { FNT } & \text { U TS101 }\end{array}$
+TS101
+TS102
+TS102 $\quad 0$
$\$$ Transient Rotor Dynamics data
ROTORSE 31
$\begin{array}{llllll}\text { RSPINT } & 3 & 1107 & 1723 & \text { RPM }\end{array}$
$\begin{array}{llllll}\text { UNBALNC } & 100 & 3 .-7 & 1107 & 0 . & 1 .\end{array}$

* 30.750
* 999999. 

UNBALNC 100 5.-7 1723 0. 1.

| $*$ | 30.75 | 90 | 0. | 0. |
| :--- | :--- | :--- | :--- | :--- |



| \$ | VISCO | PVAPCO | NPORT | PRES1 | THETA1 | PRES2 | THETA2 | NPNT |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| \$ | OFFSET1 | OFFSET2 | GRPNAME | NAME2 |  |  |  |  |
| \$ | PARM1 | PARM2 | PARM3 | PARM4 | PARM5 | PARM6 | PARM7 | PARM8 |
| NLRSFD | 1 | 2563 | 41000 | YZ | 24.5 | 11.5 | 0.2 |  |
|  | 4.-8 | 0. | 1 | 0. | 0 . |  |  |  |
|  | 0. | 0. | MyUDS |  |  |  |  |  |
|  | 120. | 30. | 1.-6 |  |  |  |  |  |
| \$ Elements and Element Properties for region : rotor_solid |  |  |  |  |  |  |  |  |
| PSOLID 1 1 10 |  |  |  |  |  |  |  |  |
| \$ Pset: "rotor_solid" will be imported as: "psolid.1 |  |  |  |  |  |  |  |  |
| CHEXA | $\begin{aligned} & 1 \\ & 20 \end{aligned}$ | $\begin{aligned} & 1 \\ & 19 \end{aligned}$ | 1 | 2 | 6 | 5 | 15 | 16 |
| . |  |  |  |  |  |  |  |  |
| . |  |  |  |  |  |  |  |  |
| CHEXA | 6704 | 2 | 12856 | 12733 | 12738 | 12861 | 12866 | 12743 |
|  | 12748 | 12871 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| \$ Elements and Element Properties for region : bush_bush |  |  |  |  |  |  |  |  |
| \$ Pset: "bush_bush" will be imported as: "pbush.3" |  |  |  |  |  |  |  |  |
| CBUSH | 50000 | 3 | 239 |  |  |  |  | 0 |





## Appendix E: MSC.NASTRAN UDS PROGRAMS

## E. 1 FORTRAN Journal Bearing Model Subroutines

## FiniteHDJB.f90

```
subroutine...
    FiniteHDJB(xA, yA, xB, yB, xAdt,yAdt, xBdt, yBdt, xoff, yoff,t,d, c, l, nu, pcav, nr,mr,tol,...
    eps,kr,Fx,Fy,NA)
    ! Subroutine that calculates the fluid-film pressure distribution
    ! within a hydrodynamic journal bearing using an iterative finite
    ! difference method. The effects of cavitation are included by
    ! applying Reynolds BCs.
    Input: - xA, yA, xB & yB: journal(A)- & housing(B) centre
        displacements in the local x- & y-directions
            - xAdt, yAdt, xBdt & yBdt: journal- & housing centre
                velocities in the local x- & y-directions
            - xoff & yoff: journal centre offsets (in the x- & y-
                directions respectively) relative to the housing
                centre
            - t: time
            - d: journal diameter
            - c: clearance
            - l: length
            - nu: dynamic viscosity
            - pcav: lubricant cavitation pressure
            - nr & mr: number of grid divisions in the theta- & z-
                directions respectively
            - tol: pressure tolerance for the SOR method
            Output: - eps: eccentricity ratio
            - kr: total SOR iterations
            - Fx & Fy: journal fluid film force in the x- & y-
                directions respectively
    Input/Output: - NA: parameter that indicates whether a
                new analysis is started (= 0.0: new analysis;
                > 0.0: analysis in progress)
                    User-modifiable parameters/code indicated by :: in comments
                    Author: Armand Krüger
                    ! Last revised: Aug 2012
                    implicit none
    !Input Variables:
    real(8), intent(in) :: xA, yA, xB, yB, xAdt, yAdt, xBdt, yBdt
    real, intent(in) :: xoff, yoff, t, d, c, l, nu, pcav
    real, intent(inout) :: nr, mr, tol, NA
    !Calculation Variables:
    real, parameter :: pi = 3.14159265359
    integer :: n, m, i, j, k, i_plus1
    real(8) :: x, y, xdot, ydot
    real(8) :: RPM, omega, e, r, theta0, thetan, Dh, B, alpha, dp, dp_norm
    real(8) :: dtheta, dz, thetai, zi, hi, dhi_dthetai, dhi_dt
    real(8) :: P01i, P11i, P21i, P10i, fi, A1i, A2i, A3i, A4i
    real(8), allocatable, dimension(:,:) :: p, p_prev
    !Error Reporting Variables:
    character(25), parameter :: filename = "c:\\FiniteHDJB_errors.txt" !::Error file...
        location
    integer :: AM
    !Output Variables:
    real, intent(inout) :: eps, kr
```

```
real(8), intent(inout) :: Fx, Fy
!Allocate Memory to p, p_prev & dp:
n = nr
m}=m
allocate(p(0:n,0:m),p_prev(0:n,0:m),stat=AM)
if (NA == 0.0) then
    open(unit=1,file=filename)
    close(unit=1,status="delete")
end if
if (AM /= 0) then
    open(unit=1,file=filename,position="append")
    write(1,*) "t = ",t," Allocation of memory to matrices p and/or p_prev failed"
    close(unit=1)
end if
!JOURNAL SPIN SPEED AT TIME t:
call SpinSpeed(t,RPM)
omega = 2*pi*RPM/60
!JOURNAL DISPLACEMENT & VELOCITY RELATIVE TO HOUSING:
x = xA - xB - xoff
y = yA - yB - yoff
xdot = xAdt - xBdt
ydot = yAdt - yBdt
e = sqrt(x**2 + y**2)
eps = e/c
!FINITE DIFFERENCE METHOD:
r = d/2
! Discrete Spatial Variables:
theta0 = 0.0
thetan = 2*pi
dtheta = (thetan - theta0)/n
dz = l/m
! Optimal Over-Relaxation Factor:
Dh = 2*(r + c)
B = 1-pi**2/2*(4 + (pi*Dh/l)**2)/(m**2 + (pi*Dh*n/l)**2)
alpha = 2*(1-sqrt(1-B**2))/B**2
! Initial Guess for p:
do i = 0,n
    do j = 0,m
        p(i,j) = 0
    end do
end do
! Solve p (SOR Method):
k = 1
dp_norm = tol + 1.0
do
    p_prev = p
    do i = 1,n
        thetai = theta0 + i*dtheta
        hi = c - x*cos(thetai) - y*sin(thetai)
        dhi_dthetai = x*sin(thetai) - y*}\operatorname{cos}(thetai
        dhi_dt = -xdot*cos(thetai) - ydot*sin(thetai)
        P01i = hi**3/dtheta**2 - 3*hi**2/(2*dtheta)*dhi_dthetai
        P11i = hi**3/dtheta**2 + r**2*hi**3/dz**2
        P21i = hi**3/dtheta**2 + 3*hi**2/(2*dtheta)*dhi_dthetai
        P10i = r**2*hi**3/dz**2
        fi = 6*nu*r**2*omega*dhi_dthetai + 12*nu*r**2*dhi_dt
```

```
    A1i = 0.5*P01i/P11i
    A2i = 0.5*P21i/P11i
    A3i = 0.5*P10i/P11i
    A4i = -0.5*fi/P11i
    do j = 1,m-1
        if (i == n) then
            i_plus1 = 1
        else
            i_plus1 = i + 1
        end if
        p(i,j) = p(i,j) + alpha*(A1i*p(i-1,j) + A2i*p(i_plus1,j)...
            + A3i*(p(i,j-1)+p(i,j+1)) + A4i - p(i,j))
        end do
    end do
    ! Pressure Continuity BCs:
    do j = 1,m-1
        p(0,j) = p(n,j)
    end do
    ! Reynolds BCs:
    do i = 0,n
        do j = 1,m-1
            if (p(i,j) < pcav) p(i,j) = pcav
        end do
    end do
    ! Norm of Difference in p:
    dp_norm = 0
    do i = 0,n
        do j = 0,m
            dp = p(i,j) - p_prev(i,j)
            dp_norm = dp_norm + dp**2
        end do
end do
dp_norm = sqrt(dp_norm)
k = k + 1
if (k > 3000) then
        open(unit=1,file=filename, position="append")
        write(1,*) "t = ",t," Maximum number of iterations reached: k = ",(k-1)
        close(unit=1)
        exit
    end if
    if (dp_norm <= tol) exit
end do
kr = k - 1
!BEARING FORCES:
Fx = 0
Fy = 0
do i = 1,n
    do j = 1,m
        thetai = theta0 + i*dtheta;
        Fx = Fx + (p(i,j) + p(i,j-1))*cos(thetai) + (p(i-1,j) ...
            + p(i-1,j-1))*cos(thetai-dtheta);
        Fy = Fy + (p(i,j) + p(i,j-1))*sin(thetai) + (p(i-1,j)...
            + p(i-1,j-1))*sin(thetai-dtheta);
    end do
end do
Fx = -r*dz*dtheta/4*Fx
Fy = -r*dz*dtheta/4*Fy
```

!::Uncomment line below for diagnostic output to a text file:
call ...
FiniteHDJBOut(t, RPM, $x, y$, xdot, ydot, e, $\left.d, c, l, n u, n, m, p c a v, t o l, a l p h a, k, d p \_n o r m, e p s, F x, F y, N A\right)$

```
    deallocate(p,p_prev)
    NA = 1.0
end subroutine FiniteHDJB
```


## SpinSpeed.f90

```
subroutine SpinSpeed(t,N)
    Subroutine that defines the spin speed of a rotor or journal
    ! of a journal bearing as a function of time
    !
    Input: t: time
    Output: N: spin speed
    ! User-modifiable parameters/code indicated by :: in comments
    !
    ! Author: Armand Krüger
    ! Created: August 2012
    implicit none
    !Input- & Output Variables:
    real, intent(in) :: t
    real(8), intent(out) :: N
    !::User-modifiable function - N(t):
    N = 1000.0*t
end subroutine SpinSpeed
```


## FiniteHDJBOut.f90

```
subroutine
FiniteHDJBOut(t, RPM, x, y, xdot,ydot,e,d, c,l, nu, n,m, pcav, tol, alpha, k,dp_norm, eps, Fx, Fy,NA)
    Subroutine that writes diagnostic information from subroutine
    FiniteHDJB to a text file
    User-modifiable parameters/code indicated by :: in comments
    Author: Armand Krüger
    Created: August 2012
    implicit none
    !Output File:
    character(22), parameter :: filename = "c:\\FiniteHDJB_out.txt" !::Output file location
    !Input Variables:
    real, intent(in) :: t, d, c, l, nu, pcav
    real, intent(inout) :: tol, NA
    !Calculation Variables:
    integer, intent(in) :: n, m, k
    real(8), intent(in) :: x, y, xdot, ydot
    real(8), intent(in) :: RPM, e, alpha, dp_norm
    !Output Variables:
    real, intent(in) :: eps
    real(8), intent(in) :: Fx, Fy
    !Write to Output File:
    if (NA == 0.0) then
        open(unit=1,file=filename)
        close(unit=1,status="delete")
    end if
    open(unit=1,file=filename,position="append")
    write(1,*)
```

```
    write(1,*) " Time step: t = ",t
write(1,*) "-----------------------------"
! ::Uncomment code below according to desired output to text file:
write(1,*) "N = ",RPM
write(1,*) "x = ",x
write(1,*) "y = ",y
!write(1,*) "xdot = ",xdot
!write(1,*) "ydot = ",ydot
!write(1,*) "e = ",e
!write(1,*) "d = ",d
!write(1,*) "c = ",c
!write(1,*) "l = ",l
!write(1,*) "nu = ",nu
!write(1,*) "n = ",n
!write(1,*) "m = ",m
!write(1,*) "pcav = ",pcav
!write(1,*) "tol = ",tol
write(1,*) "alpha = ",alpha
!write(1,*) "dp_norm = ",dp_norm
write(1,*) "k = ",(k-1)
write(1,*) "epsilon = ",eps
write(1,*) "Fx = ",Fx
write(1,*) "Fy = ",Fy
write(1,*)
close(unit=1)
end subroutine FiniteHDJBOut
```


## ShortHDJB.f90

```
subroutine ShortHDJB(xA, yA, xB, yB, xAdt, yAdt, xBdt, yBdt, xoff, yoff, t, d, c, l, nu, nr, eps, Fx, Fy,NA)
    ! Subroutine that calculates the fluid-film pressure distribution
    within a hydrodynamic journal bearing using the short bearing
    approximation. The effects of cavitation are included by
    applying Gümbel BCs.
    Input: - xA, yA, xB & yB: journal(A)- & housing(B) centre
        displacements in the local x- & y-directions
    - xAdt, yAdt, xBdt & yBdt: journal- & housing centre
        velocities in the local x- & y-directions
    - xoff & yoff: journal centre offsets (in the x- & y-
        directions respectively) relative to the housing
        centre
    - t: time
    - d: journal diameter
    - c: clearance
    - l: length
    - nu: dynamic viscosity
    - pcav: lubricant cavitation pressure
    - nr: number of grid divisions in the theta-direction
    Output: - eps: eccentricity ratio
    - Fx & Fy: journal fluid film force in the x- & y-
        directions respectively
    Input/Output: - NA: parameter that indicates whether a
        new analysis is started (= 0.0: new analysis;
        > 0.0: analysis in progress)
    User-modifiable parameters/code indicated by :: in comments
    Author: Armand Krüger
    ! Created: August }201
    ! Last revised: June 2014
    implicit none
    !Input Variables:
    real(8), intent(in) :: xA, yA, xB, yB, xAdt, yAdt, xBdt, yBdt
```

```
real, intent(in) :: xoff, yoff, t, d, c, l, nu
real, intent(inout) :: nr, NA
!Calculation Variables:
real, parameter :: pi = 3.1415926
integer :: n, i
real(8) :: x, y, xdot, ydot
real(8) :: RPM, omega, e, r, theta1, theta2, dtheta, thetai, thetaim1
real(8) :: dh_dthetai, dh_dthetaim1, dh_dti, dh_dtim1, h3i, h3im1, Fi, Fim1
!Output Variables:
real, intent(inout) :: eps
real(8), intent(inout) :: Fx, Fy
!JOURNAL SPIN SPEED AT TIME t:
call SpinSpeed(t,RPM)
omega = 2*pi*RPM/60
!JOURNAL DISPLACEMENT & VELOCITY RELATIVE TO HOUSING:
x = xA - xB - xoff
y = yA - yB - yoff
xdot = xAdt - xBdt
ydot = yAdt - yBdt
e = sqrt(x**2 + y**2)
eps = e/c
!SHORT BEARING APPROXIMATION:
n = nr
r=d/2
! Cavitation Boundary Limits:
theta2 = atan2(omega*y + 2*xdot,omega*x - 2*ydot)
if (theta2 < 0) then
    theta2 = theta2 + 2*pi
end if
theta1 = theta2 - pi
! Bearing Forces:
dtheta = (theta2 - theta1)/n
Fx = 0
Fy = 0
do i = 1,n
    thetai = theta1 + i*dtheta
    thetaim1 = thetai - dtheta
    dh_dthetai = x*sin(thetai) - y*}\operatorname{cos(thetai)
    dh_dthetaim1 = x*sin(thetaim1) - y*cos(thetaim1)
    dh__dti = -xdot*cos(thetai) - ydot*sin(thetai)
    dh_dtim1 = -xdot*cos(thetaim1) - ydot*sin(thetaim1)
    h3\overline{i}=(c - x*cos(thetai) - y*sin(thetai))**3
    h3im1 = (c - x*cos(thetaim1) - y*sin(thetaim1))**3
    Fi = (omega*dh_dthetai + 2*dh_dti)/h3i
    Fim1 = (omega*dh_dthetaim1 + \2*dh_dtim1)/h3im1
    Fx = Fx + Fi*cos(thetai) + Fim1* cos(thetaim1)
    Fy = Fy + Fi*sin(thetai) + Fim1*sin(thetaim1)
end do
Fx = nu*r*l**3*dtheta/4*Fx
Fy = nu*r*l**3*dtheta/4*Fy
!::Uncomment code below for diagnostic output to a text file:
call ShortHDJBOut(t,RPM, x,y,xdot,ydot,e,d, c,l,nu,n,eps,Fx, Fy,NA)
NA = 1.0
end subroutine ShortHDJB
```


## ShortHDJBOut.f90

```
subroutine ShortHDJBOut(t,RPM, x,y,xdot,ydot,e,d,c,l, nu,n,eps,Fx,Fy,NA)
```

```
Subroutine that writes diagnostic information from subroutine
ShortHDJB to a text file
User-modifiable parameters/code indicated by :: in comments
Author: Armand Krüger
Created: August }201
implicit none
```

```
!Output File:
character(22), parameter :: filename = "c:\\ShortHDJB_out.txt" !::Output file location
!Input Variables:
real, intent(in) :: t, d, c, l, nu
real, intent(inout) :: NA
!Calculation Variables:
integer, intent(in) :: n
real(8), intent(in) :: x, y, xdot, ydot
real(8), intent(in) :: RPM, e
!Output Variables:
real, intent(in) :: eps
real(8), intent(in) :: Fx, Fy
!Write to Output File
if (NA == 0.0) then
    open(unit=1,file=filename)
    close(unit=1, status="delete")
end if
open(unit=1,file=filename,position="append")
write(1,*) "--------------------------------"
write(1,*) " Time step: t = ",t
write(1,*) "--------------------------------"
! ::Uncomment code below according to desired output to text file:
write(1,*) "N = ",RPM
write(1,*) "x = ",x
write(1,*) "y = ",y
write(1,*) "xdot = ",xdot
write(1,*) "ydot = ",ydot
write(1,*) "e = ",e
write(1,*) "d = ",d
write(1,*) "c = ",c
write(1,*) "l = ",l
write(1,*) "nu = ",nu
write(1,*) "n = ",n
write(1,*) "epsilon = ",eps
write(1,*) "Fx = ",Fx
write(1,*) "Fy = ",Fy
write(1,*)
close(unit=1)
```

end subroutine ShortHDJBOut

## E. 2 C++ and F Source Code

Nlrsfd.cpp (Extra Code Only)

```
SCA::SCAInt32 k;
if (parm6 > 0.0) {
        fprintf(stdout,"ShortHDJB - Time step: t = %f; Eccentricity ratio:
        %f\n",time,parm4);
}
else {
        k = (int) parm5;
        fprintf(stdout,"FiniteHDJB - Time step: t = %f; Total SOR iterations:
        %i; Eccentricity ratio: %f\n",time,k,parm4);
}
```


## ext_nlrsfd.F

```
subroutine ext nlrsfd(sid, ga, gb, plane, bdia, blen, bclr, soln,
                    visco, pvapco, nport, pres1, theta1, pres2, theta2,
                        npnt, offset1, offset2, evalname, time, xx, yy, xdt,
    ydt, xb, yb, xbt, ybt, fx, fy, fuseit, bisect,
    parm1, parm2, parm3, parm4, parm5, parm6, parm7,
    parm8, omega)
implicit none
integer, intent(in) :: ga, gb, npnt, nport, sid
integer, intent(inout) :: bisect, fuseit
real, intent(in) :: bclr, bdia, blen, offset1, offset2
real, intent(in) :: omega, pres1, pres2, pvapco, theta1
real, intent(in) :: theta2, time, visco
real, intent(inout) :: parm1, parm2, parm3, parm4
real, intent(inout) :: parm5, parm6, parm7, parm8
real(8), intent(in) :: xb, xbt, xdt, xx, yb, ybt, ydt, yy
real(8), intent(inout) :: fx, fy
character(len=*), intent(in) :: evalname, plane, soln
if (parm6 > 0.0) then
call ShortHDJB(xx,yy,xb,yb,xdt,ydt,xbt,ybt,offset1,offset2,time,
&
    bdia,bclr,blen,visco,parm1, parm4, fx,fy,parm7)
else
call FiniteHDJB(xx,yy,xb,yb,xdt,ydt,xbt,ybt,offset1,offset2,time,
    bdia,bclr,blen,visco, pvapco,parm1, parm2,parm3,
    parm4,parm5,fx,fy,parm7)
end if
end subroutine ext_nlrsfd
```


## E. 3 Build Configuration Files

## SConstruct

\#
\# Main SCons configuration file for the SCA build system.
\#
from SCASCons.Configure import *
\#====================1Perform local customization here $===================$
\# This file allows you do special build configuration.
\# ThirdPartyPackage('name','description',initsub=initpackage)
\#

\#
\# Execute build
\#
import SCASCons.Setup
SCASCons.Setup. Setup ()
SConopts
\#
\# Source tree specific build options file
\#
\# Set build type
\#
BUILDTYPE = 'opt'
\#
\# Directory selections
\#
APPS DIR = 'SCAKernel'
TOOLS_DIR $=$ 'Tools'
PRODUCT LINE = 'nastran'
APPS2_SYSTEM = 'C:/MSC.Software/MSC_Nastran/20121/msc20121/sdk'
SConscript (root directory - UDS)

```
Import("env base")
env = env_base.Copy()
#=================== Perform local customization here =====================
env_base.EnableMSCFPP(extlist=[".F",".F90",".cftn",".cppftn"],d2fpp=0,mscfpp=1)
env }\mp@subsup{}{}{-}\mathrm{ base.Append(CPPDEFINES=[' DOUBLE', 'NO FTN CVF'])
```



```
retval = env.ProcessDir(env_base)
Return('retval')
```

SConscript (src directory)

```
import os
Import("env base")
env = env_bāse.Copy()
#================== Perform local customization here
```

```
srcPath=env.AbsDirPathInSou(".") + os.sep
```



```
retval = env.ProcessDir(env base)
Return('retval')
```

SConscript (fortran directory)

```
Import("env_base")
env = env_base.Copy()
#=================== Perform local customization here ====================
#============================================================================
```

retval = env.ProcessDir()
Return('retval')

## Appendix F: Additional ReSUlts

## F. 1 Experimental Modal Analysis

*Continuation of results from section 2.2.2.3.

## Horizontal FRF Point 1:




Horizontal \& Vertical FRFs Point 2:


Horizontal \& Vertical FRFs Point 3:


Horizontal \& Vertical FRFs Point 4:


Horizontal \& Vertical FRFs Point 5:





Horizontal \& Vertical FRFs Point 6:





Horizontal \& Vertical FRFs Point 7:





Horizontal \& Vertical FRFs Point 8:





Horizontal \& Vertical FRFs Point 9:





Horizontal \& Vertical FRFs Point 10:


Horizontal \& Vertical FRFs Point 12:





Horizontal \& Vertical FRFs Point 13:


Horizontal \& Vertical FRFs Point 14:


Horizontal \& Vertical FRFs Point 15:





Horizontal \& Vertical FRFs Point 16:


Horizontal \& Vertical FRFs Point 17:





Horizontal \& Vertical FRFs Point 18:





Horizontal \& Vertical FRFs Point 19:





## F. 2 Bush Bearing Characterisation

*Continuation of results from section 2.3.1.
Filtered Step Response Curve Fitting - Load Case 4, Run 3:


Filtered Step Response Curve Fitting - Load Case 4, Run 4:


## F. 3 Coupling Lateral Characterisation

*Continuation of results from section 2.3.2.

Windowed Step Response \& DFT - Load Case 2, Run 1:


Windowed Step Response \& DFT - Load Case 3, Run 5:



Filtered Step Response Curve Fitting - Load Case 1, Run 1:


Filtered Step Response Curve Fitting - Load Case 1, Run 3:


## F. 4 Coupling Torsional Characterisation

*Continuation of results from section 2.3.3.
Filtered Step Response Curve Fitting - Load Case d, Run 1:


Filtered Step Response Curve Fitting - Load Case d, Run 3:


## F. 5 1D (Free-Free) Rotor Model - Normal Mode Shapes

*Continuation of results from section 3.1.1.1.
$4^{\text {th }}$ Symmetric Lateral Mode Shape Pair - 767.81 Hz:

$5^{\text {th }}$ Symmetric Lateral Mode Shape Pair -878 Hz :

a) Mode 16 - Vertical
b) Mode 17 - Horizontal

## F. 6 3D (Free-Free) Rotor Model - Normal Mode Shapes (Before Updating)

*Continuation of results from section 3.1.2.1.
$4^{\text {th }}$ Symmetric Lateral Mode Shape Pair $-\mathbf{8 2 2 . 3 9 ~ H z : ~}$

$5^{\text {th }}$ Symmetric Lateral Mode Shape Pair - 1041.4 Hz:


## F. 7 1D Rotor System Model with Linear Bearings

Complex Mode Shapes
*Results from section 3.2.1.1.
Direct Formulation - SOL 107:

## $1^{\text {st }}$ Symmetric Lateral Mode Shape Pair - $\mathbf{3 2 . 3 1 2} \mathbf{~ H z}$ :


$2^{\text {nd }}$ Symmetric Lateral Mode Shape Pair - $\mathbf{1 3 1 . 3 3 \mathrm { Hz } \text { : }}$


## $3^{\text {rd }}$ Symmetric Lateral Mode Shape Pair - 271.21 Hz:



## $4^{\text {th }}$ Symmetric Lateral Mode Shape Pair $\mathbf{- 2 8 2 . 9 9 ~ H z : ~}$



## $5^{\text {th }}$ Symmetric Lateral Mode Shape Pair $\mathbf{- 4 7 5 . 1 6 ~ H z : ~}$



## $6^{\text {th }}$ Symmetric Lateral Mode Shape Pair - 690.39 Hz:



Campbell Diagram
*Results from section 3.2.1.3.

Model with no Damping - SOL 107:

*Results from section 3.2.1.4.

Model with no Damping - SOL 108:

Bush Bearing Displacement Frequency Response:


Journal Bearing Displacement Frequency Response:


## Disk 1 Displacement Frequency Response:



Nonlinear Transient Response
*Results from section 3.2.1.5.

Model with Damping - SOL 129:

## Bush Bearing Displacements:



Journal Bearing Displacements:


Disk 1 Displacements:


Horizontal Displacement of Node 21:


Model with no Damping - SOL 400:

## Bush Bearing Displacements:



Journal Bearing Displacements:


Horizontal Displacement of Node 48:


Disk 1 Displacements:

Vertical Displacement of Node 21


Horizontal Displacement of Node 21:


## F. 8 3D Rotor System Model with Linear Bearings

Complex Mode Shapes (No Superelement)
*Results from section 3.2.2.1

Direct Formulation - SOL 107:
$1^{\text {st }}$ Torsional Mode Shape (Mode 1) - 0 Hz:

$1^{\text {st }}$ Symmetric Lateral Mode Shape Pair - 31.91 Hz :

a) Mode 2
b) Mode 3
$2^{\text {nd }}$ Torsional Mode Shape (Mode 4) - 58.108 Hz :

$2^{\text {nd }}$ Symmetric Lateral Mode Shape Pair $\mathbf{- 1 3 1 . 4 8 ~ H z : ~}$

$3^{\text {rd }}$ Torsional Mode Shape (Mode 7) - 245.93 Hz :

$3^{\text {rd }}$ Symmetric Lateral Mode Shape Pair $\mathbf{- 2 7 6 . 8 9 ~ H z : ~}$


## Campbell Diagram

*Results from section 3.2.2.3.

Model with no Damping - SOL 107:


Transient Dynamic Finite Element Modelling of Flexible Rotor Systems

## Appendix G: Calibration Certificates and Technical Data Sheets

G. 1 Journal Bearing Lubricant - Shell Tellus S2 V 46 Physical Properties

| Properties |  |  | Method | Shell Tellus S2 V |
| :---: | :---: | :---: | :---: | :---: |
| ISO Viscosity Grade |  |  | ISO 3448 | 46 |
| ISO Fluid Type |  |  |  | HV |
| Kinematic Viscosity | @-20 ${ }^{\circ} \mathrm{C}$ | cSt | ASTM D445 | 2350 |
| Kinematic Viscosity | @ $40^{\circ} \mathrm{C}$ | cSt | ASTM D445 | 46 |
| Kinematic Viscosity | @100 ${ }^{\circ} \mathrm{C}$ | cSt | ASTM D445 | 7.9 |
| Viscosity Index |  |  | ISO 2909 | 143 |
| Density | @15 ${ }^{\circ} \mathrm{C}$ | kg/l | ISO 12185 | 0.872 |
| Flash Point (COC) |  | ${ }^{\circ} \mathrm{C}$ | ISO 2592 | 225 |
| Pour Point |  | ${ }^{\circ} \mathrm{C}$ | ISO 3016 | -36 |
| Dielectric Strength* |  | kV | ASTM D877 | >30 |

## G. 2 Proximity Probe Calibration Certificate



## Appendix H: Equations and DERIVATIONS

## H. 1 The atan2 Function

An angle (e.g. $\theta$ ) computed with the arctangent function can only lie within the first and fourth quadrants (range: $-\pi / 2<\theta, \pi / 2$ ). Contrary to the arctangent function, which accepts only one argument, the atan 2 function accepts two arguments which are used to determine the correct quadrant of the computed angle. The atan2 function is given by (not defined for $x=y=0$ ):

$$
\operatorname{atan} 2(x, y)=\left\{\begin{array}{cc}
\arctan y / x & x>0 \\
\arctan y / x+\pi & x<0 ; y \geq 0 \\
\arctan y / x-\pi & x<0 ; y<0 \\
\pi / 2 & x=0 ; y>0 \\
-\pi / 2 & x=0 ; y<0
\end{array}\right.
$$

## H. 2 2D Central Difference Equations for $1^{\text {st }}$ and $2^{\text {nd }}$ Order Partial Derivatives

The following are central difference equations for the first and second order derivatives of function $z=f(x, y)$ on a uniformly-spaced grid (refer to Figure 1.8 on p. 12):

$$
\begin{array}{cc}
\frac{\partial z}{\partial x}=\frac{1}{2 \Delta x}\left(z_{i+1, j}-z_{i-1, j}\right) & \frac{\partial z}{\partial y}=\frac{1}{2 \Delta y}\left(z_{i, j+1}-z_{i, j-1}\right) \\
\frac{\partial^{2} z}{\partial x^{2}}=\frac{1}{\Delta x^{2}}\left(z_{i+1, j}-2 z_{i, j}+z_{i-1, j}\right) & \frac{\partial^{2} z}{\partial y^{2}}=\frac{1}{\Delta y^{2}}\left(z_{i, j+1}-2 z_{i, j}+z_{i, j-1}\right) \\
\frac{\partial^{2} z}{\partial x \partial y}=\frac{1}{4 \Delta x \Delta y}\left(z_{i+1, j+1}-z_{i+1, j-1}-z_{i-1, j+1}+z_{i-1, j-1}\right)
\end{array}
$$

where $i$ and $j$ are integers that define the discrete grid points along independent variables $x$ and $y$ respectively. $\Delta x$ and $\Delta y$ are the constant spacing between grid points along $x$ and $y$ respectively.

## H. 3 Newton-Raphson Method for Determining the Static Equilibrium Position of a Finite Hydrodynamic Journal Bearing

## Journal Bearing Static Equilibrium:

Suppose that loads $L_{x}$ and $L_{y}$ are applied to a hydrodynamic journal bearing, as shown in the figure on the next page.


At the static equilibrium position of the journal, these loads are balanced by the fluid film forces $F_{x}$ and $F_{y}$ :

$$
\therefore F_{x}(x, y)=-L_{x} \quad \text { and } \quad F_{y}(x, y)=-L_{y}
$$

Functions $f_{1}(x, y)$ and $f_{1}(x, y)$, defined as:

$$
f_{1}(x, y)=F_{x}+L_{x} \quad \text { and } \quad f_{2}(x, y)=F_{y}+L_{y}
$$

are thus equal to zero when the journal is in its static equilibrium position. The Newton-Raphson method may be employed to iteratively solve the values of $x$ and $y$ for which $f_{1}=f_{2}=0$.

## Newton-Raphson Method:

The solutions for iteration $k$ are denoted $x_{k}$ and $y_{k}$. The values of $f_{1}$ and $f_{2}$ for iteration $k$ in vector form are:

$$
\{R\}_{k}=\left\{\begin{array}{l}
f_{1}\left(x_{k}, y_{k}\right) \\
f_{2}\left(x_{k}, y_{k}\right)
\end{array}\right\}=\left\{\begin{array}{l}
f_{1 k} \\
f_{2 k}
\end{array}\right\}
$$

The Jacobian matrix for iteration $k$ is given by:

$$
[J]_{k}=\left[\begin{array}{ll}
\frac{\partial f_{1 k}}{\partial x} & \frac{\partial f_{1 k}}{\partial y} \\
\frac{\partial f_{2 k}}{\partial x} & \frac{\partial f_{2 k}}{\partial y}
\end{array}\right]
$$

The difference between the solutions from iteration $k$ and the next iteration $k+1$ may be expressed in vector form (as an error vector) as follows:

$$
\{E\}_{k}=\left\{\begin{array}{l}
x_{k+1}-x_{k} \\
y_{k+1}-y_{k}
\end{array}\right\}
$$

$\{E\}_{k}$ is solved from the following matrix equation:

$$
\begin{aligned}
{[J]_{k}\{E\}_{k} } & =-\{R\}_{k} \\
\therefore\{E\}_{k} & =-[J]_{k}^{-1}\{R\}_{k}
\end{aligned}
$$

The solutions to iteration $k+1$ are then given by:

$$
\left\{\begin{array}{l}
x_{k+1} \\
y_{k+1}
\end{array}\right\}=\{E\}_{k}+\left\{\begin{array}{l}
x_{k} \\
y_{k}
\end{array}\right\}
$$

Convergence is achieved when the values of $\{E\}_{k}$ approach zero. This may be tested by calculating the norm of $\{E\}_{\Delta}$ and comparing it to an error tolerance value.

Since analytical expressions for $F_{x}$ and $F_{y}$ are not available in this case (for a finite-length journal bearing), the Jacobian matrix cannot be calculated exactly. [J] $]_{k}$ may however be approximated as follows:

$$
[J]_{k} \cong\left[\begin{array}{cc}
\frac{f_{1}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}-\Delta x, y_{k}\right)}{\Delta x} & \frac{f_{1}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}, y_{k}-\Delta y\right)}{\Delta y} \\
\frac{f_{2}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}-\Delta x, y_{k}\right)}{\Delta x} & \frac{f_{2}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}, y_{k}-\Delta y\right)}{\Delta y}
\end{array}\right]
$$

For very small finite values $\Delta x$ and $\Delta y$. This approximation originates from the fundamental theorem of Calculus:

$$
[J]_{k}=\left[\begin{array}{cc}
\frac{\partial f_{1 k}}{\partial x} & \frac{\partial f_{1 k}}{\partial y} \\
\frac{\partial f_{2 k}}{\partial x} & \frac{\partial f_{2 k}}{\partial y}
\end{array}\right]=\lim _{\Delta x \rightarrow 0 \& \Delta \rightarrow 0}\left[\begin{array}{ll}
\frac{f_{1}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}-\Delta x, y_{k}\right)}{\Delta x} & \frac{f_{1}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}, y_{k}-\Delta y\right)}{\Delta y} \\
\frac{f_{2}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}-\Delta x, y_{k}\right)}{\Delta x} & \frac{f_{2}\left(x_{k}, y_{k}\right)-f_{1}\left(x_{k}, y_{k}-\Delta y\right)}{\Delta y}
\end{array}\right]
$$

Since $F_{x}$ and $F_{y}$ are highly nonlinear functions of $x$ and $y$, load stepping should be implemented in conjunction with the Newton-Raphson method, whereby the loads $L_{x}$ and $L_{y}$ are gradually increased (up to their full values) step-by-step and $x$ and $y$ are solved via the Newton-Raphson method for each load step.

## H. 4 Analytical Solution for the Short Bearing Approximation

Setting $\partial p / \partial \theta=0$ in eqn. (1.9), p. 9, yields:

$$
\frac{\partial}{\partial z}\left(\frac{h^{3}}{6 \mu} \frac{\partial p}{\partial z}\right)=\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}
$$

Since $h(\theta)$, not a function of $z$ :

$$
\begin{aligned}
\frac{h^{3}}{6 \mu} \frac{\partial^{2} p}{\partial z^{2}} & =\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t} \\
\frac{\partial^{2} p}{\partial z^{2}} & =\frac{6 \mu}{h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right) \\
\therefore \frac{\partial p}{\partial z} & =\frac{6 \mu z}{h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)+c_{1} \\
\therefore p(\theta, z) & =\frac{3 \mu z^{2}}{h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)+c_{1} z+c_{2}
\end{aligned}
$$

where $c_{1}$ and $c_{2}$ are constants.

Assuming that full end-leakage is allowed at the bearing edges, the BCs from eqn. (1.13) are applied to solve these constants:

BC 1: $p(\theta,-l / 2)$ :

$$
\begin{equation*}
0=\frac{3 \mu l^{2}}{4 h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)-\frac{l}{2} c_{1}+c_{2} \tag{a}
\end{equation*}
$$

BC 2: $p(\theta, l / 2)=0$ :

$$
\begin{equation*}
0=\frac{3 \mu l^{2}}{4 h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)+\frac{l}{2} c_{1}+c_{2} \tag{b}
\end{equation*}
$$

Solving equations [ $a$ ] and $[b]$ simultaneously, yields:

$$
\begin{aligned}
c_{1} & =0 \quad \text { and } \quad c_{2}=-\frac{3 \mu l^{2}}{4 h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right) \\
\therefore p(\theta, z) & =\frac{3 \mu z^{2}}{h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)-\frac{3 \mu l^{2}}{4 h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right) \\
& =\frac{3 \mu}{h^{3}}\left(z^{2}-\frac{l^{2}}{4}\right)\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)
\end{aligned}
$$

which is the exact/analytical solution of the fluid film pressure distribution for the short bearing approximation. $h(\theta), \partial h / \partial \theta$ and $\partial h / \partial t$ are defined in eqns. (3.22), (3.23) and (3.24) respectively.

The cavitation boundary locations ( $\theta_{1}$ and $\theta_{2}$ ) are determined by calculating where $p(\theta, z)$ is equal to zero:

$$
p=\frac{3 \mu}{h^{3}}\left(z^{2}-\frac{l^{2}}{4}\right)\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)=0
$$

Since $3 \mu / h^{3} \neq 0$ and $\left(z^{2}-l^{2} / 4\right) \neq 0$ :

$$
\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}=0
$$

Substituting in eqns. (3.22), (3.23) and (3.24) and solving for $\theta$ yields:

$$
\begin{gathered}
\tan \theta=\frac{\omega y+2 \dot{x}}{\omega x-2 \dot{y}} \\
\therefore \theta_{2}=\operatorname{atan} 2(\omega x-2 \dot{y}, \omega y+2 \dot{x}) \quad \text { and } \quad \theta_{1}=\theta_{2}+\pi
\end{gathered}
$$

The fluid film forces acting on the journal are calculated by integrating $p(\theta, z)$ over the loadcarrying region of the bearing (thus excluding the cavitation region):

$$
\begin{aligned}
\left\{\begin{array}{c}
F_{x} \\
F_{y}
\end{array}\right\} & =-\int_{\theta_{1}}^{\theta_{2}} \int_{-l / 2}^{l / 2} p(\theta, z) r\left\{\begin{array}{c}
\cos \theta \\
\sin \theta
\end{array}\right\} d z d \theta \\
& =-r \int_{\theta_{1}}^{\theta_{2}} \int_{-l / 2}^{l / 2} \frac{3 \mu}{h^{3}}\left(z^{2}-\frac{l^{2}}{4}\right)\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)\left\{\begin{array}{c}
\cos \theta \\
\sin \theta
\end{array}\right\} d z d \theta
\end{aligned}
$$

$$
=\frac{\mu r l^{3}}{2} \int_{\theta_{1}}^{\theta_{2}} \frac{1}{h^{3}}\left(\omega \frac{\partial h}{\partial \theta}+2 \frac{\partial h}{\partial t}\right)\left\{\begin{array}{c}
\cos \theta \\
\sin \theta
\end{array}\right\} d \theta
$$

which may be evaluated using numerical integration. Employing the trapezoidal rule yields:

$$
\left\{\begin{array}{c}
F_{x} \\
F_{y}
\end{array}\right\}=\frac{\mu r l^{3}}{4} \sum_{i=1}^{n}\left[G\left(\theta_{i}\right)\left\{\begin{array}{c}
\cos \theta_{i} \\
\sin \theta_{i}
\end{array}\right\}+G\left(\theta_{i-1}\right)\left\{\begin{array}{c}
\cos \theta_{i-1} \\
\sin \theta_{i-1}
\end{array}\right\}\right] \Delta \theta
$$

## Appendix I: Finite Element Rotor System Models Tree Diagrams

## I. 1 1D Rotor System Model







## I. 2 3D Rotor System Model






[^0]:    ${ }^{1}$ The primary electricity utility in South Africa

[^1]:    2 'Unstable' meaning that the rotor vibration amplitude continues to increase with each rotation of the rotor

[^2]:    ${ }^{3}$ This frequency depends on the friction characteristics of the rotor and stator

[^3]:    ${ }^{4}$ For a finite bearing, boundaries subject to Gümbel BCs are technically also free boundaries (but not curved) since their exact positions are not know before the fluid film pressure distribution around the entire journal has been calculated

[^4]:    ${ }^{5}$ MSC.Nastran 2012.1 was used for this dissertation

[^5]:    ${ }^{6}$ MSC. Patran 2010.1 was used for this dissertation
    ${ }^{7}$ The analysis of an asymmetric rotor generally requires a rotating coordinate system

[^6]:    ${ }^{8}$ Appropriate Microsoft .Net Framework is also required; The Express Edition requires a Microsoft Platform SDK, modification of Windows registry keys and additional code in the SConscript build configuration files

[^7]:    ${ }^{9}$ Whether oil whirl and/or oil whip is present depends on the setup of the system (which influences the journal bearing load) and the unbalance loads - newer versions of this rotor kit includes a centreline adjuster that may be used to adjust the journal position within the journal bearing housing

[^8]:    ${ }^{10}$ This is assumed since the predominant lateral deformation of the rotor arises from bending of the shaft and, for small bending deformations, the shaft cross-section should therefore retain its shape

[^9]:    ${ }^{11}$ PFA 4.34 was used in this case

[^10]:    ${ }^{12}$ User-modifiable parameters or code are indicated by "::" in the code comments
    ${ }^{13}$ Through which bearing forces are effectively transferred

[^11]:    ${ }^{14}$ This damped natural frequency differs for each load case due to the different vibrating masses used for each load case

[^12]:    ${ }^{15}$ Through which the coupling forces are effectively transferred

[^13]:    ${ }^{16}$ Each addition to or change in the MSC.Patran model is considered to be a step in the construction of the model

[^14]:    ${ }^{17}$ "Coupling connection" refers to the stiffness and damping of the flexible coupling element

[^15]:    ${ }^{18}$ This punch (PCH) file output request may be specified within MSC.Patran

[^16]:    ${ }^{19}$ The spectrogram of the horizontal displacement of node 5 is almost identical and is thus not included here

[^17]:    ${ }^{20}$ Two types of clearances are supported: radial and top flat surface - these are discussed in more detail during the description of RotorDisplClearance.m in section 3.2.1.7

[^18]:    ${ }^{21}$ Since these forces are applied to the acting centres of the bearings and the coupling connection, they do not bend the rotor

[^19]:    ${ }^{22}$ Contact force/stiffness as a function of penetration (or rather deformation of the contacting bodies) may be derived from equation (3-72) on p. 118 of Shigley's Mechanical Engineering Design, 8th ed. (Budynas \& Nisbett, 2008)

[^20]:    ${ }^{23}$ Determining the extent of bearing anisotropy caused by misalignment may, however, be more involved

[^21]:    ${ }^{24}$ In order to extract the results from an off-centreline rotor node or element, the SEIDspec parameter (within the code of each program) needs to specify the ID of the rotor superelement (superelement 1) and not the residual structure (superelement 0 )

[^22]:    ${ }^{25}$ To apply different BCs to these two boundaries (where the boundary subject to Reynolds BCs is curved) would require that the non-rectangular solution domain be transformed into a rectangular domain (by specifying a new coordinate) and using an iterative method, such as the Newton-Raphson method, to solve the nonlinear free boundary problem (Prata et al., 1988) - this could add unnecessary complexity and computation time

[^23]:    ${ }^{26}$ Naturally, this spin speed function should correspond with the spin speed function defined in the BDF input file
    ${ }^{27}$ 'Journal bearing model' referring only to the finite journal bearing model captured in FiniteHDJB.f90

[^24]:    ${ }^{28}$ The 1D support elements connected to node 2563 prevent the AUTOSPC feature of MSC.Nastran from automatically constraining this (rotational) DOF

[^25]:    ${ }^{29}$ This is an approximation since the rotor is also subject to dynamic unbalance forces at 500 rpm which are ignored in this case

[^26]:    ${ }^{30}$ The spin speed is thus approximated as constant for each pulse-to-pulse segment

[^27]:    ${ }^{31}$ The motor spin direction could not be reversed

[^28]:    ${ }^{32}$ In program RotorVibSP.m, these pulse locations are provided by function TachoSpinSpeed.m

