Faculty of Engineering, Built Environment and Information Technology

A HYBRID GEARBOX CONDITION MONITORING METHODOLOGY USING TRANSFER LEARNING FOR CALIBRATION

by

Luke van Eyk
16026960

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Mechanical Engineering)

in the

Department of Mechanical and Aeronautical Engineering
Faculty of Engineering, Built Environment and Information Technology

Supervised by
Prof. PS Heyns, Dr. S Schmidt

UNIVERSITY OF PRETORIA

2021

DECLARATION OF ORIGINALITY

UNIVERSITY OF PRETORIA

The University of Pretoria places great emphasis upon integrity and ethical conduct in the preparation of all written work submitted for academic evaluation.

While academic staff teach you about referencing techniques and how to avoid plagiarism, you too have a responsibility in this regard. If you are at any stage uncertain as to what is required, you should speak to your lecturer before any written work is submitted.

You are guilty of plagiarism if you copy something from another author's work (e.g. a book, an article or a website) without acknowledging the source and pass it off as your own. In effect you are stealing something that belongs to someone else. This is not only the case when you copy work word-for-word (verbatim), but also when you submit someone else's work in a slightly altered form (paraphrase) or use a line of argument without acknowledging it. You are not allowed to use work previously produced by another student. You are also not allowed to let anybody copy your work with the intention of passing if off as his/her work.

Students who commit plagiarism will not be given any credit for plagiarised work. The matter may also be referred to the Disciplinary Committee (Students) for a ruling. Plagiarism is regarded as a serious contravention of the University's rules and can lead to expulsion from the University.

The declaration which follows must accompany all written work submitted while you are a student of the University of Pretoria. No written work will be accepted unless the declaration has been completed and attached.

Full names of student: Luke Daniel van Eyk

Student number: 16026960

Declaration

1. I understand what plagiarism is and am aware of the University's policy in this regard.
2. I declare that this assignment report is my own original work. Where other people's work has been used (either from a printed source, Internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements.
3. I have not used work previously produced by another student or any other person to hand in as my own.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Signature: 

Date: Monday 31st January, 2022

ABSTRACT

A HYBRID GEARBOX CONDITION MONITORING METHODOLOGY USING TRANSFER LEARNING FOR CALIBRATION

by

Luke van Eyk

Supervisor(s):	Prof. PS Heyns, Dr. S Schmidt
Department:	Mechanical and Aeronautical Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Mechanical Engineering)
Keywords:	CNN, Condition Based Maintenance, Domain Adaptation, Dynamic Gearbox Modelling, Maximum Mean Discrepancy, Spur Gear Modelling, Time Varying Mesh Stiffness, Transfer Learning

Gearboxes are widely utilized as critical components in a large number of engineering applications. Gearboxes are prone to failures and therefore it is advantageous to utilise a condition-based maintenance (CBM) framework to infer the condition of its components. Various data-driven and physics-driven approaches have been developed for the CBM task. In this work, a hybrid approach is proposed where a data-driven and physics-driven approach are combined to infer the condition of the gearbox. The hybrid approach combines the advantages of both approaches and aims to overcome their respective limitations. For the physics-driven approach, a numerical gearbox model is developed. The modelling procedure for the numerical gearbox model introduced a novel approach to gear fault modelling which aims to generalize the introduction of gear faults to a simpler, unified framework. For the data-driven approach, a supervised convolutional neural network (CNN) is utilised to extract features from vibration signals and classify them simultaneously. By generating synthetic data from the physical model and feeding this to the CNN, a hybrid model is developed which may yield the potential for fault identification of the real asset. There is, however, no guarantee that the learned features from the synthetic data (source domain) are transferable to a new domain of signals (target domain), such as those from the real asset. Two transfer learning methods are utilised to calibrate the hybrid model for a change in input data. To investigate the efficacy of transfer learning calibration, two numerical experiments are constructed where the hybrid model is trained on perfect synthetic data (the source domain) and applied to noisy synthetic data with different vibration signatures (the target domain). The results show that an uncalibrated hybrid model fails to transfer to the target domain, but that the calibrated methods perform well on this transfer task. This work highlights the potential of transfer learning-calibrated hybrid methods for condition monitoring of gearboxes.

ACKNOWLEDGEMENTS

I would like to diligently thank important parties involved in the completion of this Masters dissertation:

- I would like to thank my academic supervisor, Prof. Heyns, for introducing me to the vast world of asset condition monitoring, and allowing me to pursue my own interests, whilst simultaneously aligning them to the research foci of the C-AIM department. He has not only been a great source of knowledge but wisdom as well. He left me feeling looked after and provided me with clear direction.
- I would like to thank Dr. Schmidt for his large impact on this work. During our meetings he always showed attention to detail, and picking up potential stumbling blocks I never would have seen. I further thank him for his constant stream of ideas, as well as relevant articles which impacted this work directly. I also would like to thank Dr. Schmidt for always being at the ready when I needed help.
- I would like to thank Prof. Zuo (University of Edmonton, Canada) for sharing an old Masters dissertation with me. I was truly stuck at the start of my study and the dissertation made available to me kickstarted the process. Without this work, my progress would have been significantly slowed.
- I would like to thank the likes of Prof. Hammami (Ecole Nationale d'Ingénieurs de Sfax, Tunisia), Prof. Heyns, Dr. Schmidt, Prof. Chaari (Ecole Nationale d'Ingénieurs de Sfax, Tunisia), Prof. Abbes and Prof. Haddar (Ecole Nationale d'Ingénieurs de Sfax, Tunisia) for allowing me to take part in the MOSCOSSEE'2021 conference. It was an enriching, informative experience.
- I would like to extend special thanks to Ms. B Mokoka. She was always willing to help, solved problems quickly and was generally just a joy to interact with.

I would also like to thank the following parties for supporting me in my personal capacity:

- Firstly, none of this would be possible without the unwavering support from my parents and brother. They have always supported me in all I do and they always provided me with a cup of coffee just when I needed it. I thank them for their continuous emotional support.
- I thank my grandmother for continuously checking in on me and making sure that I am well cared for in the food department. I also thank her for her true interest in my project and her messages of encouragement. If ever I needed to vent she was there.
- I thank my (now deceased) grandfather for instilling in me a true sense of curiosity in this world. He was alive when I started this project, and unfortunately never got to see me finish it. He provided me with all the more reason to see this Master's through, and I can only hope that upon completion I have made him proud. His words have always stuck by me: "Daar's altyd 'n beter manier" (There is always a better way).
- I would like to thank my fellow mechanical engineer and best friend Tristan for his continuous interest and support throughout this process. On multiple occasions, he managed to break me away from the books to accompany him on site visits. These visits helped me clear my mind and gather perspective. I also thank him for being a truly supportive and caring friend.
- I would like to thank my close friend Lara Dreyer for being the person I could run to for objective perspectives on multiple problems I faced. She always showed interest in my mental well-being throughout the process and for that I thank her.
- A special thanks goes out to Dr. Axel Lexmond. Dr. Lexmond was my final year research supervisor who convinced me to pursue a Masters. I truly believe if it were not for him, I would never have pursued this degree. Furthermore, I truly thank Dr. Lexmond for providing me with the strong research basis required to complete a successful research project.

This work is dedicated to my late grandfather, Ken Erasmus (1936-2020)

Table of Contents

List of abbreviations

ANN	Artificial Neural Network
BN	Batch Normalisation
CBM	Condition-Based Maintenance
CNN	Convolutional Neural Network
CPR	Convolution-Pooling-Regularisation
DA	Domain Adaptation
DANN	Deep Adversarial Neural Network
DBM	Deep Boltzmann Machines
DBN	Deep Belief Network
DL	Deep Learning
DNN	Deep Neural Networks
DOF	Degrees of Freedom
EOM	Equation of Motion
FEM	Finite Element Method
FFT	Fast Fourier Transform
GAN	Generative Adversarial Network
GRL	Gradient Reversal Layer
HHT	Hilbert Huang Transform
HI	Health Indicators
IFD	Intelligent Fault Diagnosis
IoT	Internet of Things
KNN	K-Nearest Neighbour
LOA	Line Of Action
LPM	Lumped Parameter Model
LSTM	Long Short-Term Memory
MDOF	Multiple Degrees of Freedom
ML	Machine Learning
MLP	Multi-Layer Perceptron
MMD	Maximum Mean Discrepancy
PE	Potential Energy
PEM	Potential Energy Method
PGM	Probabilistic Graphic Model
PM	Physical Model
RFA	Relative Fault Area
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RNN	Recurrent Neural Networks
RUL	Remaining Useful Life
SAE	Stacked Auto Encoder
SF	Scaling Factor
SNR	Signal to Noise Ratio
SVM	Support Vector Machine
SWM	Square Waveform Method
TDM	Transfer across Different Machines
TIM	Transfer in the Identical Domain
TSA	Time Synchronous Averaging
TL	Transfer Learning
TVMD	Time-Varying Mesh Damping
TVMS	Time-Varying Mesh Stiffness

List of Symbols

Chapter 2
Subscripts:

Symbol	Meaning
1	Pinion Gear Parameter
2	Driven Gear Parameter
a	Axial Component
b	Bending Component
h	Hertzian Component
f	Fillet Foundation Component
s	Shear Component

Roman Symbols:

Symbol	Meaning
A_x	Instantaneous cross-sectional area on the spur tooth
b	Chip length
c	Tip clearance coefficient/ Chip width
c_m	Mean mesh damping value
$\mathbf{C}$	Damping matrix
$\mathbf{C_c}$	Constant Portion of damping matrix
$\mathbf{C_v}$	Varying Portion of damping matrix
d	Distance along the spur tooth at the force contact point
d_1	Transition distance between the base and root circle
d_h	Maximum tooth length from the tooth root
E	Modulus of Elasticity
F	Contacting force
$\mathbf{F}$	Force matrix
G	Shear Modulus
h	Height along the spur tooth at the force contact point
h_a	Addendum coefficient
h_b	Tooth height at the base circle
$h_{q,ave}$	Average effective crack depth
h_s	Spall depth
h_x	Instantaneous height along the spur tooth
I_x	Instantaneous area moment of inertia on the spur tooth
k	Gear meshing stiffness
k_{bm}	Equivalent beam meshing stiffness
K_m	Mean mesh stiffness value
$\mathbf{K}$	Stiffness matrix
$\mathbf{K_c}$	Constant Portion of stiffness matrix
$\mathbf{K_v}$	Varying Portion of stiffness matrix
L	Tooth width
L_c	Position on tooth width that crack terminates
l_s	Spall length
m	Mass/Gear module
$\mathbf{M}$	Mass matrix
$\mathbf{P}$	Projection Matrix
$\bar{r}_c$	Dimensionless tool tip radius
q_0	Left side crack depth
q_1	Variable crack depth

q_2	Right side crack depth
r	Pit radius
$\bar{r}$	Pit radius distribution mean
R_b	Base radius
R_h	Hub radius
R_r	Root radius
S_f	Sector length between the intersection points of the tooth fillet with the root circle
t	Time variable
T_m	Motor torque
T_l	Load resistance
u	Pit centre location
$\bar{u}$	Pit location distribution mean
U	Potential energy/ Strain energy
$\mathbf{U}$	Displacement matrix
$\dot{\mathbf{U}}$	Velocity matrix
$\ddot{\mathbf{U}}$	Acceleration matrix
w_s	Spall width
x	Instantaneous length along the spur tooth
x_1	Spall start position
z	Variable width value across a tooth
Z	Number of spur gear teeth

Greek Symbols:

Symbol	**Meaning**
α	Integration variable representing the equivalent angular domain angle to represent the coordinate (x, h_x)
α_{max}	Angular position α at which the maximum x value is reached
α_0	Inclination angle for the LOA
α_1	Angle between the tooth centreline and the line which connects the tooth origin to the point of tangent contact from an extended contact force
α_2	Half tooth angle of the base circle for a spur gear
α_3	Angle between the tooth centre line and the point at which the profile of the spur gear tooth intersects the root circle ($Z \leq 41$)
α_4	Angle between the tooth centre line and the point at which the profile of the spur gear tooth intersects the root circle ($Z > 41$)
α_5	Angular formulation aid representing the equivalent α_4 value relative to the base radius
δ	Displacement
ΔA_x	Effective tooth cross-sectional area
ΔL	Effective tooth contact length
ΔI_x	Effective tooth area moment of inertia
ζ	Damping ratio
θ	Gear rotation angle/Pit angle
$\bar{\theta}$	Rotational vector containing rotation angles of the full gear/ Pit angle distribution mean
θ_d	Angular duration of a double pair meshing
θ_f	Half angle between the centre point of the gear and the intersection points of the tooth fillet with the root circle
θ_s	Angular duration of a single pair meshing
θ_{tot}	Total angular duration of a single tooth's meshing cycle
μ	Mean damping to stiffness ratio

μ_f	Distance between the base of the tooth and the intersection point of the extended force line and the tooth centre line
ν	Poisson's Ratio/Crack Angle
σ_r	Pit radius distribution standard deviation
σ_u	Pit location distribution standard deviation
σ_θ	Pit angle distribution standard deviation
$\mathbf{K_c}$	Constant Portion of stiffness matrix
ω	angular speed of a rotating shaft

Chapter 3

Roman Symbols:

Symbol	**Meaning**
A_{fault}	Planar area a fault takes up
A_{plane}	Full planar area for a given fault
f_p	Rotation speed of the pinion gear
f_s	Simulation sampling frequency

Chapter 4

Subscripts:

Symbol	**Meaning**
i	Full simulation sample
j	Segment from full simulation sample

Roman Symbols:

Symbol	**Meaning**
f_p	Pinion gear rotational frequency
f_{rs}	Resampling frequency
f_s	Simulation frequency
J	Total number of segments per signal
n	Noise signal
s	Noiseless signal
x	Combined signal
$\bar{x}_r$	Healthy reference signal mean
y	Time-synchronous averaged combined signal

Greek Symbols:

Symbol	**Meaning**
σ_{x_r}	Healthy reference signal standard deviation

Chapter 5

Subscripts:

Symbol	Meaning
c	Class number in sample
d	Domain adaptation parameter
f	Feature extractor parameter
i	Sample number in batch
s	Source domain
t	Target domain
y	Label classifier parameter

Roman Symbols:

Symbol	Meaning
E	Error function
G	Internal model component
N	Total number of samples in a batch
$\mathbf{x}$	Batch of input data
X	Batch of feature vectors
$\mathbf{y}$	Batch of labels

Greek Symbols:

Symbol	Meaning
θ	Model parameters
λ	Relative loss contribution
μ	Learning rate

Calligraphic Symbols:

Symbol	Meaning
$\mathcal{L}$	Loss function

Chapter 1 Introduction

1.1 Background

Gearboxes are commonly found across many industries, including the mining and power generation industries. The gearboxes play a critical role in the process and are usually expected to run continuously. It is therefore of paramount importance that these gearboxes do not incur unplanned downtime, as this will lead to total production shutdown. Traditionally such problems may be solved by preventative maintenance, but this runs the risk of premature failure or premature replacement, e.g. replacing a healthy gearbox. A more sensible approach to the maintenance problem is through Condition-Based Maintenance (CBM).

CBM is a strategy that aims to make maintenance decisions based on an asset's condition (Lei et al. 2018a). This can in turn reduce unplanned downtime or replacement of perfectly healthy components. It follows that CBM is an important strategy for critical machines and components, where downtime must be kept to a minimum, or where components are expensive to replace. CBM is therefore a suitable approach to apply to gearboxes.

CBM consists of two main tasks, namely diagnostics and prognostics. Diagnostics mainly focus on the tasks of anomaly detection, fault isolation and fault classification. Prognostics rather focus on the estimation of Remaining Useful Life (RUL) of an asset (Cubillo et al. 2016). Kundu et al. (2020) state that a diagnostic approach can be followed without a prognostic approach being utilised, but not vice versa. This implies that diagnostics are a fundamental part of any CBM implementation and may therefore be the starting point for any larger CBM approach, should it be diagnostic or prognostic in nature.

Broadly speaking, diagnostic CBM approaches can be thought of as consisting of three main approaches: Physics-driven, data-driven and hybrid approaches.

A physics-driven approach is concerned with modelling the underlying physics of some system. These models may then be used to try and analyse the real system in a numerical environment. One can easily manipulate the geometry of faults and fault sizes in the numerical environment. This allows the creation of a database of synthetic signals which may be analysed to predict how a machine will react under different fault conditions. The generation of a fault database for an asset is extremely valuable, as obtaining such a database experimentally is hard and expensive (Kundu et al. 2020). A further value of developing a physical model lies within its ability to be extended to prognostic approaches, as the physical model allows for faults to be modelled and propagated. Regardless of how the physics-driven approach is implemented, it has one key drawback: The conclusions drawn from the physics-driven approach is generally limited to how well the actual system and faults are modelled. Physical models are simplified versions of the true, complex real asset and therefore do not guarantee that the knowledge obtained from the physical model is transferable to the real asset Kundu et al. (2020).

Data-driven approaches utilise large amounts of data to predict the condition of an asset, with no direct need to understand any underlying physics of the system at hand. The data may be pre-processed by utilising advanced signal processing techniques such as time-synchronous averaging, order tracking or many other signal processing techniques. The processed data is then manipulated to form more meaningful representations, known as features. These features may include hand-crafted statistical features or may involve more modern machine learning-based

approaches, which aim to automatically extract meaningful features (Lei et al. 2020). The features are then combined using either expert knowledge or machine learning techniques to make predictions on the machine condition. The type of predictions that are made depend on whether the data has labels. Where no labels are present, unsupervised approaches are required and where labels exist supervised methods are followed. Supervised methods are much more common in literature, and allow for different system states to be predicted (Lei et al. 2018a). Although supervised methods are more common, they suffer two major drawbacks: Firstly, faults rarely occur in industry, making data for such events scarce. Secondly, where data exists, it is often unlabelled and cannot be used for supervised approaches (Lei et al. 2020).

A hybrid approach attempts to merge the two aforementioned approaches to utilise both methods' strengths whilst eliminating their weaknesses (Kundu et al. 2020, Lei et al. 2020). Hybrid approaches are generally applied as a method to expand diagnostic data-driven models into prognostic models (Lei et al. 2020). Hybrid approaches, however, need not be limited as vessels for expansion to prognostics, but are successfully utilised in a full diagnostic capacity. Works such as Gryllias & Antoniadis (2012), Sobie et al. (2018) and Liu & Gryllias (2021) utilise physical models to generate labelled data which is then used as training data for a data-driven approach. This is a sensible alternative to running experiments, which are expensive and time-consuming. Therefore, it may be stated that these diagnostic hybrid approaches work well to resolve the data scarcity challenge.

Although the diagnostic hybrid approach outlined above proves to be a sensible approach, it has one underlying challenge: The approach makes the strong assumption that the hybrid model which is trained on one domain of data (the synthetic data from the physical model) will transfer well to a different but related domain of data (the measured real data, such as accelerometer signals on industrial gearboxes). That is to say, it assumes that once the data-driven model has been trained on synthetic data from the physical model, it is completely transferable to real data. This will most likely not be true, as a discrepancy will exist between the two domains of data. This domain discrepancy can be traced to the difference between the physical model and the real system. The physical model is a simplification of the much more complex real system where noise, complex lubrication, mounting errors, etc. may be present (Liu & Gryllias 2021).

Therefore, it may be stated that there exists a need to calibrate the hybrid model to work well on a different, but related domain of data. Transfer learning is an attractive choice for calibration, as this field is mainly concerned with transferring knowledge from one domain to another (Pan & Yang 2010). There are many transfer learning approaches that may be followed, but this work will focus specifically on a sub-field known as domain adaptation. Domain adaptation is a transfer learning method that specifically focuses on aligning the feature distributions between two different datasets (Pan & Yang 2010).

In summary, this work will focus on developing a diagnostic hybrid model for detection and classification of faults within a gearbox. Firstly, a physical gearbox model will be created which can generate large amounts of labelled synthetic data. Secondly, a data-driven model will be implemented and trained on the synthetic data. This alone does not yet solve the challenge of utilising such a model on a different, but related dataset, such as the asset on which the physical model is based. Therefore, transfer learning will be investigated as a technique to minimise any variations between domains, acting as a tool to calibrate the diagnostic hybrid model.

1.2 Literature review

Since the focus of this work is on the development of a hybrid model which may be calibrated for data discrepancies, four major areas will be researched. Firstly, the literature regarding physical gearbox models will be researched, as the first portion of the work will focus on generating synthetic data. Secondly, research will be conducted into existing supervised data-driven techniques which may work well in a gearbox CBM context. This is important, as the second portion of the work is concerned with training a data-driven method to classify the synthetic signals. Thirdly, research will be conducted into existing hybrid CBM approaches for rotating machinery. This research is necessary to gather insight into any potential areas that require attention within the hybrid modelling approaches. Finally, research will be conducted into transfer learning approaches for rotating machinery. More specifically, the sub-field of domain adaptation will receive attention, as it is the proposed method in this work to calibrate the hybrid model.

1.2.1 Physics-based modelling of gears

Physics-based models (PBM) have an important role in condition monitoring research and serve as useful tools to generate synthetic data. Synthetic data has the following advantages over experimental data for condition monitoring research:

- Experimental data are expensive and time-consuming to acquire.
- Experimental test-rigs do not provide enough flexibility to model complex failure shapes. Experimental test-rigs allow for faults to be induced and progress naturally, but at the cost of little to no control over the exact fault geometry and intensity to be investigated.
- Experiments are much more difficult to control: Temperature and pressure fluctuations, other machinery nearby, poor experimental setup and many other factors can degrade the level of control in an experiment. These factors make it more difficult to draw conclusions from measured experimental data when developing new condition monitoring methods.

PBMs have evolved over time and have presented quite useful insights into the operation of gearboxes under healthy and faulty conditions. Various gearboxes such as single-stage spur and helical gearboxes; multi-stage spur and helical gearboxes; and planetary gearboxes have been extensively researched. However, the main focus for this work will be on single-stage spur gearboxes due to the scope of this work extending beyond the PBM. Furthermore, single-stage spur gearbox models share a large portion of the modelling process with planetary (which consist of spur gears) and helical gearboxes (Mohammed & Rantatalo 2020). They are therefore the ideal starting point, regardless of which gearbox model is desired.

The modelling of a spur gearbox can generally be split into three separate sub-modelling processes, namely:

- Gear mesh stiffness modelling
- Gear fault modelling
- Gearbox dynamic modelling

The following sub-sections are therefore divided into these three processes.

1.2.1.1 Gear mesh stiffness modelling

The gear mesh interactions between two gears result in excitations that manifest as dominant components in the gearbox vibration measurements (Mohammed & Rantatalo 2020). Therefore, it is important to accurately capture the dynamics between two meshing gears. Hence, much work has been focused on gear mesh modelling (Ma et al. 2015).

The meshing stiffness between two gears vary over time and is further referred to as the time-varying meshing stiffness (TVMS). The TVMS has two main contributing factors. The first factor relates to the fact that the number of teeth in contact during a gear's rotational cycle is not constant. That is to say, that at some stages only a single gear tooth pair are in contact, but at other times two (or more) gear teeth pairs could be in contact. This variation in the number of gear tooth pairs in contact causes a sudden change in the TVMS. The square waveform method (SWM) is generally a simple and phenomenological method of accounting for this effect and is generally utilised to study the phenomenon of gear meshing.

The second effect arises because as a gear rotates, the contact line between a gear tooth pair moves across the tooth faces. This in turn causes not only the location of the force on a given tooth to change but also the angle of the acting force. This effect is much less pronounced but is not negligible. This variation is more complicated to capture and is generally accounted for utilising the finite element method (FEM) or the potential energy method (PEM). Although FEM is considered to be the most accurate approach, it is also very computationally expensive and therefore not ideal in an environment where one would like to digitally experiment with different gear geometries and fault shapes.

The PEM is an attractive alternative to FEM as it aims to deliver the same accuracy as FEM but the speed of an analytical approach. As one objective of this work is to utilise a hybrid approach, many functional evaluations are anticipated and therefore the PEM is sensible to investigate further.

Yang & Lin (1987) showed the first instance of utilising the PEM. The authors considered two main factors which contribute to the TVMS. The first of these factors was the Hertzian contact stiffness, which refers to the local stiffness related to the interaction of two deformable bodies. This deformation can be equated to an equivalent Hertzian stiffness, which is a function of the geometries of the interacting gears. The assumption is made that the two gear teeth in contact are locally parabolic in shape (Yang & Lin 1987).

The second factor considered relates to the axial and bending forces experienced by the gear tooth. At the point of contact between the two gear teeth, a component of the force acts inwards (axially) toward the gear centre and another component towards perpendicular to the gear centre, which adds a bending force to the tooth. To account for these two forces, the cantilever beam theory is introduced. The assumption is made that gear teeth may be modelled as non-uniform cantilever beams. One may then calculate the potential energy stored within these beams (hence why this method is named the potential energy method), and relate this to the stiffness in the beam. This in turn results in a beam stiffness term for each gear tooth pair in contact. Note that the beam stiffness term is comprised of an axial stiffness term and a bending stiffness term.

After this theory was introduced, Tian (2004) introduced an additional stiffness term to the beam stiffness, namely the shear stiffness. This term was also based on beam theory, and once again relied on the potential energy stored within a beam, but now due to the shear force introduced by the force between two contacting teeth.

In the same year, Sainsot et al. (2004) introduced a novel stiffness term to the PEM, which aimed to account for the deflection of the gear body - the portion between the gear hub (through which the shaft passes) and the base of the tooth (where the tooth begins). To model this deflection, circular elastic ring theory from Muskhelishvili (1977) was utilised, and an analytical formula for the fillet foundation stiffness was generated via a polynomial curve fitting method.

As a quick reminder, at this point in the literature, five main stiffness effects are considered, namely three beam stiffness terms (axial, bending and shear), a Hertzian contact stiffness term and finally a fillet foundation stiffness term. These five terms remain relatively unchanged and are used throughout literature.

The next major change in the PEM came about in the work of Liang et al. (2014), where the authors noted that to this point, all gear models were defined relative to the base radius. This assumption is an oversimplification of real gears, where the machining process generates fillets or undercuts at the root of a gear tooth (Hyatt et al. 2014). A more applicable parameter to define where a gear starts, would be the root radius. The base radius may be larger or smaller than the root radius depending on how many gear teeth are present on a given gear. This over or underestimation of base radius causes the calculated volume of material to deviate from the true value, causing inaccuracies during the stiffness modelling process. Liang et al. (2014) introduced corrections to the beam stiffness equations to account for these two cases (root radius larger or smaller than base).

The first set of equations were related to the case when 41 or fewer gear teeth are present (machining leaves a fillet), which refers to the case when the base radius is larger than the root radius. This implied that all prior beam stiffness terms were only accounting for the portion of the gear tooth that started from the base radius onwards, whilst in reality, there was a filleted portion of material between the root radius and base radius that needed to be accounted for. Therefore, Liang et al. (2014) introduced a correction term that added the missing tooth material for this case, increasing the overall meshing stiffness. In the second case, i.e. when more than 41 teeth are present on a gear, one finds that the root radius exceeds the base radius (leading to an undercut). In this case, the amount of gear material is overestimated and the beam stiffness equations need to be reformulated to different bounds entirely. Note that the use of 41 gear teeth specifically arises from standard spur gear theory, where the

assumption is made that the contact angle between gear teeth are 20°. Contact ratios of 20° are common in practice.

Saxena et al. (2015) presented an additional beam stiffness term, namely a torsional stiffness term, which arises due to shaft misalignments that cause one side of a gear tooth contact line to experience higher forces than the other side of the gear tooth contact line. Furthermore, the authors developed theory for the case where the two shafts holding the gears are not parallel (which is a different form of misalignment in comparison to what is described above), by adding misalignment terms to the already existing beam stiffness terms. From the literature, however, it seems that these misalignment terms were not adopted in any other authors' work, except in a follow-up paper (Saxena et al. 2015), where the torsional stiffness effect was accounted for using spall faults.

Ma et al. (2016) addressed a shortcoming of the fillet foundation stiffness term: It was developed with the assumption of single tooth pair contact. In theory and practice, however, it commonly occurs that for periods within the rotation of a gear, two or more pairs of teeth are in contact. This leads to a situation where the fillet foundation term is added to the final stiffness twice (once for each gear tooth pair in contact). This is invalid, as the two pairs of teeth in contact share the same bodies, therefore the combined stiffness effect should be less than what is being accounted for. To fix this problem, Ma et al. (2016) described a FEM-aided method with which one calibrates the relative contribution of each gear tooth pair. In essence, although one has two fillet foundation stiffness terms in the case of two gear tooth pairs meshing, one simply multiplies the second fillet stiffness term with a reduction factor which is determined by FEM.

Regarding the classical PEM, these were the most significant changes to the initial framework presented by Yang & Lin (1987). However, El Yousfi et al. (2020) presented a novel approach to gear stiffness modelling, by discretising the gear tooth in the tooth width and length directions. This method served as a hybrid between the full discretisation of FEM, and the full continuity assumption of the PEM as it involves integral formulations, which are by definition continuous. Thus, although this work offered no benefit in terms of meshing stiffness accuracy over the traditional PEM developed up to this stage, it allowed for a much more robust fault modelling framework, which is the focus of the following sub-section.

1.2.1.2 Fault modelling

The fault modelling process plays an important role in any gearbox model used for condition-based maintenance applications: To observe how different faults affect the dynamics of a system, which in turn allows one to predict behaviours for the real-life asset. Due to its importance, fault modelling of gears receives a healthy amount of attention, with various fault types and gear phenomena being modelled.

Mohammed & Rantatalo (2020) give a review of various fault models from the literature. These models try to capture (but are not limited to) tooth root cracks, tooth profile errors, tooth modifications, broken teeth, chips, tooth surface wear, pitting and spalling. To limit the scope of this research, the more popular fault models (based on initial search engine results) will be researched, with the assumption that they are currently the most relevant to the field of gear fault modelling. These include root crack, broken teeth, chip, spall and pit models. Before showing the fault models, one should note that currently the literature is quite disparate in terms of how faults are modelled and there is a strong need for a unified framework, especially for neophytes in the field of gear tooth modelling (van Eyk et al. 2022). Regardless, the fault models which have been developed will each be discussed in their own sections to follow.

Crack Models

Gear crack models receive a large amount of attention in the literature (Mohammed et al. 2013a). These models refer specifically to the case of gear root cracks, where the crack is initiated near the root of the gear tooth and propagates through the full width of the tooth. It is important to understand that cracks can simultaneously vary in their depth and width along a tooth root, complicating the modelling procedure.

Tian (2004) was according to the author's knowledge the first work to propose a gear crack model. This model assumed a full-width crack of constant depth. This constant width and depth model assumed a constant crack angle, which describes the propagation path into the tooth body. This assumption implied that cracks propagate along a linear path into the depth of a gear, as opposed to some non-linear path. Experimental work by Lewicki (2002) showed that cracks generally follow a near-straight path when propagating into a gear's body, and therefore the straight path assumption is deemed accurate enough for modelling purposes.

The work of Tian (2004) was limited to cracks above the centreline of the tooth. Therefore, Wu et al. (2008) extended this model for cracks beyond the midpoint. Both these works neglected the fillet foundation stiffness term previously discussed. Chaari et al. (2009) did however consider this effect and managed to verify their crack model against the FEM, which verified the proposed crack fault modelling method.

The next major improvement to the crack model came from Chen & Shao (2011). Their work overcomes the constant depth or width assumption for cracks, made by Tian (2004). Instead, the authors designed a parabolic distribution to describe the crack shape within the gear tooth. This work was key to more realistic physical models, as gears do not always experience uniform load across their width, in which case one would thus expect cracks to start propagating from one edge of a tooth and move to the other side. As an added benefit of the work by Chen & Shao (2011), the authors were able to verify their model against the FEM results given by Chaari et al. (2009).

Further improvement to the crack model of Chen & Shao (2011) was made by Mohammed et al. (2013a). Up to this point, the volume of the tooth that does not contribute to the stiffness due to the presence of a crack was assumed to be a straight limiting line drawn from the deepest point of the crack, parallel to the tooth centre-line. One effectively calculates the meshing stiffness as if that portion of the tooth does not exist. What this model fails to consider, is that due to the cracked tooth, the 'unusable' portion of the tooth will bend slightly, adding or subtracting some 'useful' material from the stiffness determination. To account for this, Mohammed et al. (2013a) ran a FEM simulation to determine how stresses are distributed due to a tooth root crack. Based on the stress results, the authors could develop a parabolic limiting line that roughly followed the constant stress gradient from FEM. This method proved to be more accurate compared to previously published methods, specifically for cracks deeper than 30% of the tooth's depth.

Ma et al. (2014) did an in-depth study comparing three combinations of crack fault modelling methodologies. Method 1 consisted of the straight propagation path assumption which started in Tian (2004), with a straight material reduction line. Method 2 consisted of the straight propagation path (like before) but utilising a parabolic material reduction line (as in Mohammed et al. (2013a)). Method 3 consisted of a parabolic crack propagation path, joined with a parabolic material reduction line. All three methods were compared against FEM for various crack depths and angles. It was shown that the model by Mohammed et al. (2013a) was sufficiently accurate and that the straight-line propagation assumption was accurate. This result once again correlated well to experiments conducted in the works of Lewicki (2002), where crack propagation paths were shown to be mostly linear.

Yang et al. (2021) recognise a key issue in the field of crack modelling: Some published work assumes that cracks occur in closed states rather than open states. This implies that the axial compressive stiffness term is left unaltered as it is assumed the crack does still allow for material to 'compress' in this region, transferring the axial load as if no crack existed. Other works assume the crack to be in an open state, which implies the bending of the cracked portion of the tooth is sufficient to bend the tooth tip away from the base, making it impossible to transfer axial forces in the crack region, therefore affecting the axial stiffness term. This is the first work the author has found addressing this problem. Yang et al. (2021) proposed the open state model and verified it against FEM, proving that the open state assumption is the correct one. Furthermore, the results were compared against a closed state assumption and proved to match FEM results better.

Broken Teeth Models

There is not much literature on broken tooth modelling as it is a fairly simple fault type to implement. This fault type can generally be considered a 'full depth crack', if the crack model being utilised is flexible enough.

Therefore, a broken tooth can simply be modelled as a special crack case. Another method, however, is to set the overall meshing stiffness to zero for the tooth being considered, which leads to the same effect (Tian 2004).

Chip Models

Chip models, although relevant to industry, are not as widely studied as other fault types. The first of the three main keystones to this fault type starts with the work of Tian (2004). In this work, only the effective contact surface reduction was considered, i.e. the chip was assumed to have negligible depth. This assumption implies that the beam stiffness terms were left unaffected, with only the Hertzian stiffness term being affected. Following this work, Chaari et al. (2008) proposed a methodology that considered the depth of chips, therefore affecting the beam stiffness terms as well. However, this modelling method assumed that the connecting fracture curves between the vertices are linear. This essentially assumes that the chip's break away from the tooth along straight lines, instead of the natural curve they should follow if the tooth geometry was to be considered. This may lead to large deviations from the actual fracture shape when chips are fairly deep. Therefore, Liu et al. (2021) developed an analytical geometry method that determines the shapes of the fracture curves more accurately. Furthermore, these results were verified against FEM results.

Spall Models

When referring to the literature, there is no clear distinction between spalls and pits (Mohammed & Rantatalo 2020). Sometimes the models are interchangeable and therefore the literature for these two fault types overlap. Generally speaking, pits refer to small, shallow, localised pieces of material that are pitted away from a gear tooth. A spall, however, refers to a collection zone of pits that cause a larger piece of material to be removed. Thus, pits refer to smaller, local phenomena, whereas spalls refer to the larger-scale phenomena.

Initially, the first analytical model for a spall was developed by Chaari et al. (2008), where the spall was very simply modelled as a rectangular shape with a given depth that is removed from the gear tooth. This model was left relatively unchanged until Saxena et al. (2016) unified some of the disjunct works in this field and transformed them into a PEM implementation. In this work, three main spall models were considered, namely rectangular, circular or V-shaped spalls. Luo et al. (2018a) recognised that the previous methods of spall modelling all assumed constant depth spalls, where the case, in reality, would be closer to a varying depth spall. Therefore, Luo et al. (2018a) developed a model which modelled the spall as a spherical cut-out from the tooth face, thus leading to a varying depth on the tooth. In the same year, Luo et al. (2018b) extended this model to be ellipsoidal, which allowed for more elongated, tooth-wide spall shapes of varying depths. This model was further verified against FEM results and was shown to correlate well.

Luo et al. (2019*a*) extended spall modelling by rethinking the entire stiffness modelling process. This new method allowed for a larger array of spall shapes to be modelled. This work verified this fact by comparing their more general model against existing works. The only drawback of this work was the fact that all spall geometries assumed a constant depth. This model was also shown to be capable of modelling pits, making it extremely versatile.

To overcome the constant depth limitation of Luo et al. (2019*a*), the work from El Yousfi et al. (2020) finally presented the field with a method that could model virtually any shape of any geometry and any depth. The only potential limitation of this work was that it utilised a discretisation approach of the gear tooth, which makes the framework of simulation close to that of FEM (although PEM is being used by this approach). To address such a situation, the user has full control over the coarseness of the discretisation and therefore the user has the option between modelling accuracy and simulation speed.

Surface Pit Modelling

Although surface pits are more common than cracks (Liang et al. 2016, Liu et al. 2020b), surface pit modelling receives less attention in the literature. Pitting is generally observed to occur below the pitch line of a tooth because this is the region of the tooth where single tooth meshing occurs. A tooth consists of multiple zones of meshing, corresponding to when neighbouring teeth pairs come in to or out of contact to share the load transfer

between two gears. For certain portions of gear rotation, only a single tooth pair are in contact, which from the perspective of a tooth, is referred to as a single tooth meshing region. The single tooth pair meshing region is therefore associated with higher stresses and thus offers a more conducive environment for pits (Lei et al. 2018b).

Liang et al. (2016) modelled pits as individual cylindrical cut-outs from the tooth face. The authors extended the model to allow for multiple cylindrical pits, allowing for these pits to be distributed across the tooth face as necessary. Their method was verified against FEM and for the given cylindrical spall geometry, coincided well. A limitation of this work was that the pits were fixed in certain locations across the tooth face. In reality, pits do not distribute in an organised manner but follow a more progressive, probabilistic distribution. Therefore, Lei et al. (2018b) improved the pitting field by distributing the pitting faults using a probabilistic approach, setting the mean location for pits within the single tooth meshing region of the tooth.

Another shortcoming from Liang et al. (2016) was that the pits were all assumed to be of constant size. Therefore, Lei et al. (2018b) presented an analytic approach that would not only increase the number of pits for a given fault severity case but also increase their size. This was aimed at simulating reality, where over time, smaller pits grow larger whilst new pits are simultaneously created. This approach was also verified against FEM and the obtained meshing stiffness values never differed by more than 0.65%.

Chen et al. (2019) also used cylindrical pit shapes but implemented a different fault distribution modelling methodology compared to Lei et al. (2018b). The authors modelled pits in local two-dimensional Gaussian clusters, each with its own mean and variance. With this model, it is possible to propagate multiple pitting zones on a single tooth, depending on the severity required for modelling.

Up to this point, no attention had been given to the cylindrical pit shape assumption. In reality, pits do not have constant depths (such as cylinders) and therefore the field needed a newer pit geometry. Liu et al. (2020b) presented a spherical pit, whilst employing the same fault progression methodology as in Lei et al. (2018b).

Meng et al. (2021) also proposed a spherical pitting model, this time with a very detailed write-up. The authors also followed the fault propagation methodology of Lei et al. (2018b). Meng et al. (2021) were further able to prove their model correlated well with FEM.

1.2.1.3 Dynamics modelling

The act of dynamic modelling is the final step within the gearbox modelling framework. Whether the time-varying mesh stiffness (TVMS) was obtained through a FEM model, or a PEM model, both result in a gear mesh stiffness that is a function of gear rotational angle. If a full FEM approach is followed, not only would the gear and faults be modelled in a FEM environment, but the dynamics would be solved in a FEM environment as well. This method, although accurate, may be very time consuming and computationally expensive. An alternative method is to approximate the gearbox as a non-linear lumped parameter model (LPM). The equations of motion are obtained by using a Newtonian or Lagrangian approach, whereafter the equations of motion are solved with numerical integration methods. Although the LPM is not as accurate, it is much quicker to solve and captures the main vibrations of the system.

Within the literature, a vast number of LPMs have been employed. This section does not aim to capture them all, but rather try and show general modelling techniques, assumptions and key findings in the dynamic modelling literature.

Degrees of Freedom Considerations

It is important to select the appropriate degrees-of-freedom (DOF) of the system. If this is not done, it is possible to neglect important dynamic behaviour (Ma et al. 2015, Mohammed & Rantatalo 2020).

In the early literature on LPMs for single-stage gearboxes (i.e. two shafts and two gears), it was generally common to find a two DOF LPM. The first DOF aimed to capture the rotational dynamics of the pinion and the second DOF the gear. Upon further investigation these two degrees could be converted into a single DOF by noticing the rotational displacement of each gear is a multiple of the other (the operating principle of gears). Works such as Yang & Sun (1985), Yang & Lin (1987), Parker et al. (2000) and Richards & Pines (2000) all incorporate this two DOF setup to study gear dynamics.

More recently, a more common eight DOF model is used, with some works simplifying this to a six DOF model. In this model, authors try to account for the dynamics of bearings, shafts, the motor and load on the gearbox. Therefore, starting from the motor, one DOF is assigned to the motor. This motor is then coupled through a flexible shaft (with stiffness and damping terms) to a mass block, which is the combined mass of the pinion gear, pinion side shaft mass and pinion side bearings. This block has three DOFs, namely one for rotation (relating to the gear), one in the x and one in the y directions, relating to the movements of the bearings. This mass block is coupled to another three DOF mass block which simulates the output side gear, shaft and bearing mass. These two blocks are connected through the TVMS term and a damping term which will be discussed in its own section. Finally, the gear block is connected to a load through flexible shafts again, which makes for the eighth and final DOF. Authors such as Tian (2004), Chaari et al. (2008), Khabou et al. (2011), Chaari et al. (2012), Mohammed & Rantatalo (2015), Mohammed et al. (2015), Schmidt (2017) and Luo et al. (2019*b*) all employ this exact model, proving that it is quite a popular choice among researchers.

Some other researchers, however, try and simplify the eight DOF model by neglecting the motor and load DOFs, therefore only considering bearing and gear dynamics. This six DOF model has been used in works such as Chen & Shao (2011), Mohammed et al. (2013b,a), Mohammed & Rantatalo (2015), Mohammed et al. (2015), Liu et al. (2020c), Meng et al. (2020) and Yang et al. (2021).

Mohammed & Rantatalo (2015) questioned the accuracy of employing the six DOF model over an eight DOF for single-stage modelling and argued that if the motor and load inertias were small they may be neglected. In another study, Mohammed et al. (2015) compared four different DOF models over various statistical indicators such as RMS, kurtosis and crest factors to identify how much the different DOF models differ in these features. The four models were the standard eight DOF and six DOF models discussed above, as well as a 12 DOF model which allowed the gears of the eight DOF model to have rotational capability about the x and y axes, adding two more DOF for each mass block. The final model was another six DOF model, but instead of neglecting the motor and load to get to six DOF, rather the tooth contact friction was removed, thereby removing vibrations in the x-direction. Therefore, the reduced six DOF model represented the normal eight DOF model except it did not account for x-direction vibrations.

Mohammed et al. (2015) found that there was no mentionable difference between the 12, 8 and reduced 6 DOF model in terms of statistical indicators, and therefore it is acceptable to go with the reduced six DOF model which neglects inter-tooth friction. The authors further found that the six DOF model with motor and load removed showed statistical indicators that deviated from the three other DOF models. Therefore, it was suggested that removing the motor and load DOFs could cause less accurate dynamics, leading to the deviation in statistical indicators. Mohammed et al. (2015) did however state that if the gearbox has known eccentricities, causing gyroscopic effects, the 12 DOF model needs to be used, as the other models cannot capture these dynamics.

Luo et al. (2019*b*) added two additional DOFs to the normal 8. The additional DOFs accounted for the gearbox casing stiffness and damping. The model further accounted for additional factors such as tooth surface roughness and inter-tooth friction, both of which were given an analytical calculation method. The 10 DOF model was based on an experimental rig. The parameters of the 10 DOF model was calibrated from tests done on the experimental rig. It was shown that the 10 DOF model generated data that matched measured data from the rig quite well.

Feng et al. (2019) presented a 21 DOF model for a single-stage spur gearbox by splitting up the bearings and couplings to have their own masses and DOFs. Their model's data correlated well with the modelled experimental test rig's data. However, it was only possible to match the model's response with experimental data after the model was calibrated. A first round of calibration aimed to match the natural frequencies of the two systems by adjusting joint and bearing stiffnesses. Then, to match the RMS of the simulated and true signal, the meshing stiffness and damping were fine-tuned. The authors utilised a wear model to predict RMS of the system for a given amount of wear and were able to show a great correlation with the actual experimental test rig. This work shows how difficult it can be to fully utilise physical models, as one needs to have experimental information to calibrate the physical model.

A summary of multi-stage gearboxes that do not directly fit into the aforementioned paragraphs, is presented here:

Jia et al. (2003) and Jia & Howard (2006) developed a 26 DOF model with three shafts all accounting for gear, bearing and shaft interactions. Additionally, this model accounted for geometric errors present in the gears, which was directly introduced into the dynamic formulation. Bartelmus et al. (2010) utilised a similar model but only accounted for rotational DOFs, leading to a simplified six DOF model. Ruiz-Botero et al. (2015) also developed a similar model, but this time accounting for the bearing x and y movements, but different from Jia et al. (2003), leading to a 10 DOF model.

Yi et al. (2020) developed an advanced multiple degree of freedom (MDOF) model which coupled an electric motor model with an in-depth gearbox model. The proposed multi-stage gearbox model incorporated a fixed shaft gear set and planetary gear sets, making for a very interesting model.

Damping Considerations

In the literature on dynamic gearbox models, the models are usually calibrated by altering the stiffness and damping parameters to obtain useful physical models and may have as few as 10 to as many as 21 DOFs to model a single-stage gearbox. It is not always possible to obtain the necessary experimental values to calibrate the physical model directly. It may be said that the TVMS (which has dedicated models as discussed in Section 1.2.1.1) is quite close to FEM in most cases (Mohammed & Rantatalo 2020), adding to one's confidence that the stiffness parameter related to gear meshing need not be tuned. Other parameters, however, such as bearing stiffnesses are usually given as constants in the literature. It is possible to construct a separate experimental rig to determine the stiffness and damping coefficients for such elements (such as bearings and shafts). However, one element which may prove difficult to quantify is the damping between gear teeth. The following survey aims to determine the methods with which authors deal with the mesh damping problem.

Philippe (2012) states that damping is a challenging problem to solve and mentions that one of the biggest shortcomings in the gear modelling field is the lack of damping models. The author further states that damping values are generally adjusted to fit experimental results. Before this adjustment stage, however, one of two common models are employed. The first is a Rayleigh damping model, which models the system's damping as being directly proportional to two factors, namely the system mass and the system stiffness. The coefficients in front of the mass and stiffness matrices are then adjusted to fit with experimental results. The second method utilizes modal damping factors, where the damping is once again a function of system masses and stiffness, but now under one term which is scaled by the damping ratio.

Works such as Parker et al. (2000), Khabou et al. (2011), Chaari et al. (2012), Yu (2017) and Wang et al. (2020) all utilise Rayleigh damping. Parker et al. (2000) describe in detail how one may go about determining the Rayleigh factors and is based on experimentation. Yu (2017) however outlines an analytical method based on natural frequencies and modal damping factors to determine the Rayleigh coefficients. As for the second approach, works such as Richards & Pines (2000), Tian (2004), Chaari et al. (2006), Wu et al. (2008), Ruiz-Botero et al. (2015), Wei et al. (2017), Xu et al. (2017), Yi et al. (2020) and Xu et al. (2021) all include modal damping approaches. Parker et al. (2000) indicate that if modal damping is used in the dynamic system, a damping ratio of 7% is a commonly used value. Tian (2004) also used this value in their work, further confirming that this is

a commonly used value and therefore one less parameter to determine experimentally. The above-mentioned works corroborate the statement that models generally utilise one of two damping approaches.

Works such as Jia et al. (2003), Jia & Howard (2006), Bartelmus et al. (2010), Chen & Shao (2011), Mohammed et al. (2013a,b), Mohammed & Rantatalo (2015), Liu et al. (2020c), Meng et al. (2020) and Yang et al. (2021) all utilised some form of damping, but the procedure that was used to obtain the values is unclear.

In earlier works, Yang & Sun (1985) and Yang & Lin (1987) utilised a Hertzian damping formulation, which effectively accounts for energy loss during the interaction between teeth by looking at the area inside the hysteresis loop formed when comparing tooth penetration as a function of compression force. This damping is assumed to be simultaneously proportional to the tooth displacement (interpenetration) and tooth relative velocity (viscous damping).

In recent works, attention has been focused on more accurately modelling the damping effects between teeth. Guilbault et al. (2012) focused mainly on the mechanisms involved in damping of gear transmission systems and the resulting dynamics. Three sources of damping were considered, namely surrounding elements, hysteresis of the gear teeth and oil squeeze film damping. The first two factors were load-dependent, whereas the last factor was load-independent, and a value based strongly on fluid dynamics theory. This model was by no means simple but was a respectable attempt at eliminating uncertainty within gear damping models. The results from this work showed that there is a difference between damping in the case of a loaded vs. unloaded tooth. This implies that damping terms should not be constant, as is a common assumption in literature.

Liu et al. (2015) used the hysteresis model that describes the hydrodynamic contact force on the teeth as a function of gear transmission error, similar to what was seen by Yang & Sun (1985). This loop could be used to estimate the energy dissipation (as one obtains a force vs distance graph) of contacting teeth in hydrodynamic circumstances. The main aim of this work once again was to minimize the number of unknowns during modelling.

Luo et al. (2019*b*) model the meshing damping as a sum between two terms. The first term accounts for oil film damping and the second term for damping between teeth (structural damping). Both terms are given their own analytical formulas with which they may be found. Luo et al. (2019*b*) affirms the fact that the correct values for stiffnesses and damping values are essential to get as close a representation to reality as possible. The authors state that lower values of stiffness generally decrease the system natural frequencies, whilst high damping tends to suppress non-linear behaviour. Furthermore, the authors performed experiments to calibrate the coefficients used in their model. The calibration was successful, as the final RMS and harmonic frequencies matched well with the experimental rig's data.

All of these works highlight that progress has been made in the field of meshing damping. It was seen that although most common approaches in the literature either apply Rayleigh damping or modal damping, some other works started specifically developing damping models.

1.2.2 Data-driven approaches

Data-driven approaches can be split into diagnostic or prognostic approaches. The focus of this work is diagnostic in nature, as already stated in the background. Within the diagnostic field, one may focus on supervised, semi-supervised and unsupervised approaches. The overarching goal of this work is to develop a hybrid model which will receive labelled synthetic from a physics-driven approach. The search within the literature is therefore confined to supervised approaches, as labels from the numerical model are available during training.

Kundu et al. (2020) give an extensive overview of diagnostic and prognostic approaches to gears. For diagnostic data-driven approaches, the authors state that many options exist. For one such approach, a more statistical-based method may be applied where health indicators (HI) can be determined from the raw signal. These HIs can be determined from various domains, such as the time, frequency or time-frequency domains. These HIs are optimal if they are shown to have monotonicity, robustness and a good correlation to faults. For example, if an

HI is not monotonous, the same HI value may refer to multiple degradation states, making for a poor diagnostic tool.

Kundu et al. (2020) highlight a potential drawback of relying on HIs: Some HIs are more suitable at different fault degradation levels. Bishop (2006) also outlines this problem by stating that hand-crafted rules break down in corner cases (such as a change in load, non-stationary conditions, random impulsive signals from other machines nearby, etc.).

Therefore, it may prove advantageous to merge/fuse multiple HIs to do the fault diagnosis task. Also, considering the rise in available measurement data for systems (Lei et al. 2020), a large rise has been seen in recent times in machine learning (ML) based methods (Jia et al. 2016). Lei et al. (2020) recognise four main categories of health state recognition (a framework utilising hand-crafted input features to predict the state of a machine as output), namely: Expert system-based approaches, artificial neural network (ANN) based approaches, support vector machines (SVM) based approaches and other approaches. Among the 'other approaches' categories, one finds machine learning models based on k-nearest neighbour (KNN), decision tree and probabilistic graphical models (PGM). Jia et al. (2016) state that among these approaches, ANN seems to be the most popular.

Both Jia et al. (2016) and Lei et al. (2020) identify the same two common issues across all shallow deep learning models: Firstly, it would prove very advantageous to have a system where faults could be identified automatically from the raw signal. However, due to manual, hand-crafted features being extracted, the role of the diagnostician is still required. Furthermore, the features which might be representative of a fault within a certain asset may not at all be representative of faults in another type of asset, limiting the generalisation ability of shallow approaches. Therefore, to truly remove the diagnostician, one requires an automatic feature recognition system that can mine the most useful features from the data. The second issue plaguing shallow machine learning approaches is that of feature complexity. Shallow models struggle to extract highly complex, non-linear relationships from the training data, limiting their diagnostic accuracy.

Lei et al. (2020) indicate that in modern times, with the advent of the internet of things (IoT), the amount of available data on machine states has massively increased. This implies that more data is available for determining the condition of machines than ever before. Furthermore, due to high sampling frequencies of data and due to faults sometimes manifesting at high vibrational frequencies, the size of datasets quickly become large. Shallow learning methods cannot fully utilise this enormous amount of data, commonly referred to as "big data". Recent interest in deep learning (DL) models have proven to be useful in dealing with the large amount of data.

Zhao et al. (2019) define deep learning quite well as a technique that "attempts to model high-level representations behind data and classify (predict) patterns via stacking multiple layers of information processing modules in hierarchical architectures". A deep learning model aims to deepen architectures of machine learning models, to such an extent, that the machine learning model can automatically extract features, select the relevant features and simultaneously classify the machine condition. It is said that the deeper the architectures become, the more complex and non-linear the features that may be extracted from the data (Jia et al. 2016).

With this, the justification for a large amount of new literature on DL based methods is clear: A deep learning approach not only attempts to be monotonous, robust and well correlated (as for the HI only case), but also attempts to be invariant to different operating conditions (as for the shallow ML cases). Furthermore, a DL approach tries to remove the importance of experts needing to extract manual features, which could impair the overall classification process. Thus, all of the aforementioned shortcomings seem to be solved using a DL model. Thus, a deep learning model is said to be a full package that can convert raw data to a machine condition (Lei et al. 2020).

Common deep learning structures that are applied, are all based on neural networks, leading to the field of deep neural networks (DNN). Common DNN models which have been utilised in the field of machine health monitoring include stacked autoencoders (SAE), deep belief networks (DBN), deep Boltzmann machines

(DBM), convolutional neural networks (CNNs) and recurrent neural networks (RNN) (Zhao et al. 2019, Lei et al. 2020).

From the previous paragraphs, it has become evident that deep learning, or more specifically, the use of deep neural networks seems to be much more successful compared to traditional methods. This is a strong statement, and to justify this, a few examples from the literature are given. Note that most of the works utilise CNNs, as these are generally considered to be the top model choice for fault identification of rotating machinery (Zhao et al. 2020).

Jia et al. (2016) used a stacked auto-encoder (SAE) approach using fast Fourier transform (FFT) information on two separate datasets, namely a bearing and gearbox dataset. The authors based their model on that of Hinton & Salakhutdinov (2006) and found that the novel deep learning approach outperformed the previous ANN methods from their own and other investigations in both accuracy as well as robustness, having the smallest variance across multiple training runs. The authors did however note the main drawback to the novel DNN approach was that of slower training, due to the inherent size increase for deeper architectures. Jing et al. (2017) applied a one-dimensional frequency-CNN approach to classifying a planetary gearbox dataset. This approach outperformed all networks where manual expert features were extracted and fed to a shallow NN. Verstraete et al. (2017) applied a time-frequency based approach to two bearing datasets. The authors constructed a novel CNN architecture, which contained two convolutional layers before a pooling layer instead of the more common single convolutional layer. This new architecture was seen to improve the CNN's robustness against noise, whilst simultaneously reducing the network size. When hand-crafted features were passed to the CNN structure, the performance dropped compared to giving the raw time-frequency domain signal to the CNN. This showed that CNNs are useful for spatial/temporal data, but do not perform well when given hand-crafted features as input, as there is no temporal/spatial component to such inputs.

Han et al. (2019c) applied three different CNNs, one for data from a Fourier transform, one for data from a wavelet transform and one for data from the Hilbert Huang Transform (HHT). It was seen that the wavelet transform was capable of extracting the most information from the gearbox dataset, as this method had the highest classification accuracy. A final finding of the paper was that upon combining the three methods into an ensemble of DNNs, the classification accuracy further increased over the CNN which utilised the wavelet transform data. This showed that the CNN was capable of extracting a large portion of available data, but that different approaches could yield different forms of information, which could increase classification accuracy if combined. This ensemble model could be seen as a shallow network, only receiving highly complex features as input (the outputs of the three models) and simply classifying these features into machine states. This ensemble was capable of outperforming any of the single DNNs. This work showed that feature extraction is the most important part of the machine learning process, as a shallow network outperformed a deep network, purely based on the fact that the features it received were rich with information.

Although DNNs have shown great promise to the field of data-driven diagnostics, a large concern persists: A data-driven approach is only as useful as the quality and quantity of data it has to train off. In many cases neither of these factors are present and therefore, a physics-based approach might yield better results. Obtaining a sufficient amount of failure data is generally not possible with existing assets, as they are seldom allowed to run to failure. Furthermore, setting up experimental setups are limited by cost, time and even fault geometry reproducibility (Kundu et al. 2020). In cases such as these, it might prove useful to combine both a physics-driven and data-driven approach to solving the data shortage problem.

1.2.3 Hybrid approaches

Conventional diagnostic CBM methods are either data-driven or physics-driven, each of which has shortcomings for practical applications. Kundu et al. (2020) recognise that a hybrid model aims to exploit the benefits of both physics-driven and data-driven methods.

A promising approach to diagnostic hybrid models seems to be the generation of synthetic data from a physical model and then training the data-driven model on this synthetic data. This conceptually allows one to reap the benefits of both approaches: For little to no industrial data, one can obtain a network that is trained to

have diagnostic accuracies at least as good as the physical model in isolation. However, as more data becomes available, the hybrid model may be updated with the new data and further improve its accuracy. In works such as Gryllias & Antoniadis (2012), Sobie et al. (2018), Liu & Gryllias (2021), synthetic bearing data is generated by the use of a physical/phenomenological model. This data is then used to train a ML model, which is then applied to real datasets. These hybrid methods showed improved accuracies when compared to their data-driven counterparts which were only trained on the few available data samples from the real asset.

Bobylev et al. (2021) constructed a model to estimate the support stiffness for rotating machinery. The authors constructed a physical model of an experimental setup and were able to show the benefit of a hybrid model in the context where no experimental data is present. More specifically, the model was trained on synthetic data from the physical model and the model was able to predict, with reasonable accuracy, the true support stiffness when applied to experiments. It was further shown that a CNN proved to be the best model for generating robust features across data domains, i.e. synthetic vs. experimental.

From what the author could find, the literature on diagnostic hybrid models was quite limited. To the author's knowledge, no hybrid approaches have been applied to gearboxes. In a recent review from Lei et al. (2018a), the authors show that only 8% of publications related to RUL predictions involved hybrid approaches. Although this figure is related to prognostics, it does give some insight into the shortage of literature on hybrid models.

At this point, diagnostic hybrid models seem to solve the data shortage problem. However, few works - not only limited to hybrid models - focus on the important step of applying the developed models to real, unseen data. That is to say, the models are not tested against a different dataset, to see how robust they are to changes in data. When looking at reviews within the diagnostic condition monitoring field, such as those from Zhao et al. (2019), Kumar et al. (2020) and Lei et al. (2020), the consensus seems to be that although diagnostic models work on one set of data, their performance is not guaranteed on another set of data. Therefore, the developed diagnostic models need some form of calibration to ensure they are robust to changes in data and can more easily be transferred to real engineering scenarios. These reviews all agree that transfer learning serves as a promising approach to calibrate diagnostic models. Therefore, the following section of the literature review is focused on transfer learning.

1.2.4 A tool for calibration: Transfer learning

As mentioned in the previous section, diagnostic models suffer from a lack of data when being applied to industrial environments. Even when data may be present, it will most likely not be balanced, i.e. the healthy class will be overrepresented to a large extent. Furthermore, data will likely be unlabelled and therefore only apply to unsupervised training techniques. One may actively try to prevent these issues by continuously inspecting and labelling data from the real asset, but this defeats the "online" monitoring purpose (Stander & Heyns 2005) and may prove to be costly (Lei et al. 2020).

Lei et al. (2020) propose that the potential solution to this problem is the concept of transfer learning (TL). Transfer learning tries to take a model which was trained on one domain of data and apply it to another domain of data. A domain may be seen as a dataset. In the literature one commonly refers to the source domain (the domain from which a model is trained) and the target domain (the domain to which the source-trained model is to be transferred). For example, the source domain could refer to an experimental dataset (source domain) that trains a DNN and is then expected to function in an industrial environment on industrial data, the target domain.

Although there is more than one way to approach TL, this literature review will show a strong bias towards a specific implementation of TL known as domain adaptation (DA). TL can be thought of as the description for the process and DA as an implementation mechanism. The concept of domain adaptation is not new and seems to arise from image recognition literature (as does much of modern machine learning literature). Another type of TL implementation is also available and is known as inductive methods (Han et al. 2019b). One example of the latter method will be given shortly.

Ganin et al. (2017) define domain adaptation as the process of learning a discriminative predictor (such as a classifier) for the case that a shift between source and target distributions occur. The ultimate goal of DA is to learn a domain mapping from the source domain to the target domain. With such a mapping, classes across domains may be distributed similarly in latent space. In turn, it is hoped that high accuracies for the source domain would lead to similar accuracies for the target domain, given the two domains have similar distributions, and therefore similar classification boundaries. This has the implication that the target domain may be fully unlabelled, and still well classified in practice, as the main focus is aligning domains, not directly optimising target domain classification accuracy.

DA has many forms and may be implemented through various architectural constructs. Recall the goal of DA is to align domains (or features) between two differing domains. Therefore, the main focus of many DA techniques is to find an effective technique to achieve domain alignment or domain invariance as it is sometimes referred to. To simplify the following explanations from literature, a simplification of a general DNN structure is required. DNNs may be seen as being split into two main sections, a feature extraction section and a feature classifier section. At some point, the feature extractor stops, and an array of features are generated from the feature extractor. This will be referred to as the final feature vector. The final feature vector is passed to a normal MLP model, which acts as a feature classifier. The classifier may have multiple layers and tries to learn a mapping between the self-generated features and the desired output. Thus, the DNN is said to consist of a feature extractor and a feature classifier component, with a feature vector as a separating layer in-between the two components.

Many DA approaches are generally formulated by focussing specifically on the feature extractor. The goal is to develop a feature extractor that can simultaneously generate feature vectors that are domain invariant and meaningful. Meaningful features aid the classifier, whereas domain invariant features aid the robustness of the model to domain changes.

One can generally define the transfer learning approach according to two major categories, namely transfer in the identical domain (TIM) and transfer across different machines (TDM). If a TL approach is applied on the same machine type, one obtains what is known as TIM. If, however, one tries to transfer knowledge to a different machine, one obtains what is known as TDM. Both methodologies have the common assumption that the source domain data is fully labelled, but that the target domain is sparsely labelled or even not labelled at all. Regardless of the applied methodology, Lei et al. (2020) show that four common approaches are followed to transfer knowledge, namely feature-based, GAN-based (General Adversarial Network), instance-based and parameter-based approaches. Lei et al. (2020) state that transfer learning seems to be a promising method for utilising experimentally learned knowledge in practice. Therefore, the following sub-sections give examples of successful applications in each of the two TL groups.

1.2.4.1 Transfer in the identical domain (TIM) approaches

TIM is generally focused on transferring knowledge across the same machine, but for differing operating conditions, load conditions or different environments of operation (Lei et al. 2020).

Within the transfer learning literature for diagnostic condition monitoring, three major DA network styles are common. The first is based on the work from Tzeng et al. (2014), which relies on a metric to measure the similarity between domains. Tzeng et al. (2014) utilised the maximum mean discrepancy (MMD) metric to measure similarity between domains, and jointly minimised this metric with classification loss. It was shown that this DA method outperformed other state-of-the-art methods in transferring learned features from image datasets from a source domain to a target domain. It must further be noted that this architecture allowed for completely unsupervised data in the target domain. Do however note that this work was based on an image recognition task, but that the same architecture is easily transferable to the diagnostic CBM task.

The second network style is based on the work from Ganin et al. (2017). The authors replace the MMD metric with a sub-NN, which adversarially tries to determine the domain from which a given sample originated. This prediction is then used to adversarially update the feature extractor, whilst labels are passed to the feature classifier. This, once again jointly leads to domain invariant features that are also meaningful. The work from

Ganin et al. (2017) showed further improved results when compared to that of (Tzeng et al. 2014) on the image classification task. This method is very similar to a GAN, but instead of separately training a generator and discriminator, the process acts as an all-in-one approach. Furthermore, this approach merely attaches another smaller NN, namely the domain classifier to a standard CNN structure and therefore many existing CNN structures can be 'upgraded' for domain invariance by adding this simple extra component to the network. Once again, this approach could be utilised for a target domain that has no labels.

Han et al. (2019a) implemented a TIM DA approach based on the architecture from Ganin et al. (2017). Data was taken from wind turbine, spur and bevel gear datasets at various operating conditions. A CNN was trained on specified load cases and the test cases were set as completely unseen load cases. When applying DA to this approach, an improvement in fault classification accuracy was seen for all datasets. The authors thus showed that the implementation of DA to their existing data-driven model increased the accuracy of the model on unseen conditions.

The third network style involves the utilisation of GANs. GAN-based approaches generally utilise some form of generator and discriminator to adversarially reach an optimum. Examples of such an approach will be given shortly. Chen et al. (2020) applied a GAN based DA method to the rotating machinery fault diagnosis problem. In this work, the DA process was constructed as follows: Firstly, a standard CNN was trained on a source domain until no accuracy increase was seen. A second model was constructed from the first, by duplicating the feature extractor weights. This second model was developed for the target domain feature extraction. After setting up the second model, the weights from the source domain feature extractor were frozen, however, the weights of the target domain feature extractor were not. Then, iteratively, source and target samples were sent through both of their respective networks, resulting in source and target domain feature vectors. These vectors were then passed to a discriminator. Therefore, in a GAN sense, the target domain data features could be seen as the generator network and the domain classifier as the discriminator. Instead of generating samples, however, the generator was aimed at generating feature representations from target samples that match those from the source domain. Therefore, the discriminator and generator were pitted against one another until mutual optimality was reached. At this point, the discriminator could be discarded and the target domain feature extractor could be connected to the source domain's classifier, which had already learnt to map features to accurate outputs. Since the adversarial approach resulted in domain invariant features, one would expect that the classification of the target domain would therefore be similar to the source domain. As a further benefit, once again the process could be conducted in a completely unsupervised manner for the target domain. Chen et al. (2020) showed that this method outperformed other state-of-the-art DA models on a bearing dataset. These models include the model types proposed by Tzeng et al. (2014) and Ganin et al. (2017). However, it must also be stated that these other DA methods still well outperformed the non-DA approaches, once again proving the strong case for DA.

Breaking away from DA approaches, Han et al. (2019b) implemented an inductive approach to the TL task. An inductive approach is geared towards fine-tuning a pre-trained model. The general idea behind this work is as follows: The early layers within a CNN are responsible for extracting general features from a dataset. Therefore, upon training, the front layers of a CNN may already be extracting meaningful and domain invariant features. It is only near the latter part of the network, perhaps even only the classifier part, that features become more domain-specific, and cause a drop in accuracy when transferring data domains. Therefore, Han et al. (2019b) proposed three methods to utilise already trained networks. In the first approach, the pre-trained CNN's feature extractor weights were all frozen and only the classifier weights were fine-tuned on the limited labelled target domain data. This approach assumed that the feature extractor was already general enough, and the overfitting to a domain occurred in the classifier layers. In the second approach, the entire network was allowed to be fine-tuned. This was coupled with the assumption that the whole network could be overfitted to the source dataset and a small amount of fine-tuning was necessary to calibrate the model on the target domain. In the third and final approach, the method was much more similar to DA, where the assumption was that overfitting occurred in the feature extractor. Therefore, in this approach, the classification portion of the network was left frozen and the feature extractor was fine-tuned. All three approaches were tested across two experimental datasets. These approaches were then compared against source domain training only and all three methods showed appreciable improvements in classification accuracy of the target domain. Furthermore, it was seen that

the third approach outperformed the other two approaches. Therefore, this work once again seems to show that the problem of domain invariance may be solved by looking at the feature extractor, not the classifier.

1.2.4.2 Transfer across different machine (TDM) approaches

From the author's experience, the literature on TDM approaches to condition monitoring is quite limited. TDM approaches are considerably harder than TIM approaches. TDM attempts a more difficult task of trying to transfer knowledge to a completely different, but related machine, such as from a motor to a generator Lei et al. (2020). These approaches are truly difficult, as they try and transfer information across two completely different domains. The previous section focused mainly on simple changes in load or changes in speed classification cases. These approaches may be seen as data-driven approaches, as they are not interested in transferring knowledge from one machine (physical model) to another (the real asset). They are only interested in calibrating data within the same machine. The works which will be presented shortly, focus on a much more difficult task: Transferring knowledge from a hybrid model with synthetic data as the source domain to real signals from assets, the target domain. This case is quite difficult as many variables other than speed and load may now vary between domains. Such variables include the DOFs of the hybrid model vs. the real case, temperature effects, effects of other machines in the measurement environment and simulated vs. measured vibration location.

In the work of Liu et al. (2020a), a prognostic hybrid model is built for bearing RUL prediction. Firstly, the authors verified a useful prognostic model against other state-of-the-art models. Then, the authors compared this model against its domain adaptation counterpart. The DA approach utilised the same structure as Ganin et al. (2017). In this work, the source domain was synthetic data produced by a phenomenological bearing model. The target domain was real experimental data. It was shown that the DA approach yielded far better RUL results when applied to the real data compared to the non-DA hybrid model. This showed that with a strong prognostic hybrid model, with a large body of synthetic data of training, one cannot guarantee success for the same model on a different domain. However, if the hybrid model is calibrated with DA, it performs much better, as the domain invariance has been minimised between the synthetic data and real data.

In further works, Liu & Gryllias (2021) focused on the bearing diagnostic task. The authors utilised a CNN hybrid model with data being generated from a phenomenological bearing model. The authors implemented the same architecture as in Ganin et al. (2017), with a slight change to account for differences in class between source and target domain. In this work, the authors assume that small amounts of labelled target domain data were available. Thus, their novel framework would not work in unsupervised target domain cases. Regardless, the authors showed that with small amounts of labelled target domain data, the DA approach outperformed a deep MLP, time-frequency CNN, and their own CNN without DA. This showed that utilising DA is effective at maximising the information contained in limited data. As a further comparison, their method was pitted against other DA approaches and was shown to be most effective. Models which performed close in accuracy to their model, and which could be trained in an unsupervised manner (and is therefore more useful) was shown to be the models already described from Tzeng et al. (2014) and Ganin et al. (2017).

The literature shows evidence that transfer learning, or more specifically, domain adaptation, shows great promise for calibrating a hybrid model when it is expected to be applied to a different dataset.

1.3 Scope of research

The motivation for this work stems from an industrial problem: Inferring the condition of an industrial gearbox from condition monitoring data. The inferred condition of the gearbox may then be used to inform maintenance decisions under a condition-based maintenance framework. The condition monitoring task for complex machinery such as gearboxes, however, is not trivial. Gearboxes run under complex operating conditions such as varying loads, speeds, temperatures, levels of wear on components, etc. This makes physics-driven approaches difficult to calibrate and limit their usefulness for such environments. To further complicate the condition monitoring task, these complex assets generally do not have run-to-failure data which may be used for a data-driven approach. Lastly, if historical data is available, it is often times unlabelled and may be contaminated by noise from sensor measurement or vibrations from other nearby assets. Therefore, data-driven approaches are also limited in scope for industrial gearboxes.

This work proposes a hybrid model that consists of a numerical gearbox model for labelled data generation and a supervised data-driven model to classify gearbox data into different health states. The data-driven portion of the hybrid model is further calibrated through transfer learning to aid in the condition monitoring task. To ensure that the performance and behaviour of the hybrid method and accompanying calibration approach are understood, this work will focus only on the virtual condition monitoring problem. In this virtual problem, synthetic experiments are specially designed to understand the behaviour of the models in different environments.

Hybrid methods are invaluable for the condition monitoring problem as they can realise the benefits from both physics-driven and data-driven approaches simultaneously. The first portion of developing a hybrid gearbox model is the construction of a physical gearbox model. The gearbox model in this work aims to develop a platform where large amounts of labelled synthetic data may be generated. This labelled synthetic data is generated in a controlled, noise-free environment and serves as a good source of data for the data-driven portion of the hybrid approach. Furthermore, the fact that the data is labelled allows for a supervised data-driven method to be employed. This is advantageous, as the labels allow the hybrid model to predict different types of faults, as opposed to only the presence of faults. However, before such a hybrid model may be realised, the literature highlights the following three shortcomings that need to be addressed to achieve the hybrid methodology:

To construct a gearbox model, one needs to accurately model the meshing stiffness between gears, as well as the effect accompanying faults have on this meshing stiffness. The potential energy method has much potential for determining gear meshing stiffnesses, as it is as accurate as finite element methods, but much faster to solve. However, gear fault modelling literature using the potential energy method is disjunct, as different mathematical notations and formulations are used for different fault types. Furthermore, the focus of the published papers is usually for a single fault type, as is evident from the various models described in Mohammed & Rantatalo (2020). There is a need in the gear fault modelling field for a unified approach, so that more robust synthetic datasets may be built. Currently, it appears as if researchers build an entire model around a single fault type, whereas it could be very beneficial to build faults around a single gearbox model (van Eyk et al. 2022).

Secondly, although the literature is focused on building optimal deep learning approaches for gear condition monitoring, it is usually unclear how these data-driven models should be utilised when insufficient amounts of data are present. Hybrid approaches are proposed in the literature, but the literature that was obtained mostly focus on bearing condition monitoring, such as the hybrid works in Gryllias & Antoniadis (2012) and Sobie et al. (2018). Therefore, there is a need for more literature on gearbox condition monitoring applications with limited historical fault data.

Lastly, although the aim of the physical modelling portion of the hybrid modelling framework is to replicate the real asset as well as possible, calibration may be difficult and time-consuming. In addition to this, the physical model is only a simplified version of a much more complicated system and will therefore struggle to capture the full range of complex interactions of the real asset. Therefore, one may expect at least some difference between synthetic data and real data (i.e. epistemic errors). Due to this difference in data, the hybrid model is not guaranteed to perform equally well across different datasets (Lei et al. 2020). That is to say, that a hybrid model trained on synthetic data may not generalise to the real industrial data.

Hence, in this work the following contributions will be made to address these shortcomings:

The first part of this work aims to develop a generalised gearbox model that can address the three important sections of gearbox modelling, namely, gear mesh stiffness modelling, fault modelling and dynamic modelling. To address the generalisation problem, both the gear mesh stiffness modelling and fault modelling portions need to be redefined in a generalised mathematical framework. To verify the correctness of the model, the proposed generalised framework will be compared against published models. The new generalised gear mesh stiffness framework and accompanying fault models will be incorporated in a lumped mass model from the literature and integrated using numerical integration methods.

To address the second issue, which is that of limited data for real gearboxes, a hybrid model will be developed for gearbox fault detection. This model will consist of the aforementioned gearbox model (physics-driven model) and a deep neural network (data-driven model). The gearbox model will generate labelled synthetic vibration data, which may be used to train a supervised data-driven approach. For the supervised deep neural network, no experimental or industrial data is available, and therefore this deep neural network cannot be evaluated in isolation. Therefore, a proven deep neural network that has been applied to a real gearbox dataset will be chosen from the literature to verify its effectiveness.

With the physics-driven model developed and a data-driven architecture selected, the novel hybrid model will be constructed and implemented. Hybrid models of this sort have been constructed before, but are mainly focused on bearings and have not yet, to the author's knowledge, been implemented on gearboxes. Therefore, the second novelty in this work lies in its application to gearbox condition monitoring specifically.

The hybrid model requires validation, and will therefore be tested against a synthetic dataset to validate its effectiveness in identifying faults for different noise profiles. With a strong hybrid model having been developed, the final shortcoming of current works will be addressed, namely the issue of how to calibrate the well-trained hybrid model to classify datasets from industrial gearboxes.

Therefore, to address the third and final shortcoming, a portion of this work will be dedicated to studying and understanding the effectiveness of implementing different transfer learning techniques. Specific focus will be given to a technique known as domain adaptation, as this seems to populate the majority of literature, as it shows great promise (Han et al. 2019a).

The calibrated models may then be compared to their uncalibrated hybrid model counterparts to gain insight into the usefulness of transfer learning techniques for calibrating a hybrid model with limited data in the industrial case. The application of transfer learning as a calibration tool for a hybrid gearbox model is to the author's knowledge novel and has not been utilised before. A portion of other works have applied transfer learning to gearbox datasets, but these works all focused on simpler in-domain tasks, where vast amounts of experimental data were already available, and transfer learning was only applied to learn different load cases within this dataset. This work proposes a method to transfer knowledge from one dataset to a different dataset. In practice, and in alignment with the motivation for this work, this knowledge transfer may occur from a synthetic dataset to a real asset.

In summary, the following novelties result from the work:

- A novel generalised fault modelling framework is proposed that places the focus of fault modelling around the stiffness framework, and not the entire framework around faults. This framework must be modular in the sense that various fault types can be simulated from the same base mathematics, without having to change the stiffness model at all.
- A novel hybrid gearbox model will be developed, which is trained in a supervised manner by utilising a deep neural network combined with labelled synthetic data from the physical gearbox model.
- A novel calibrated hybrid gearbox model which will be robust against a change in input data. This change in input data may represent the motivation for this work: A transfer of knowledge from a synthetic data to industrial data.

The following section briefly outlines how the scope of research will be broken up across the document.

1.4 Document overview

Chapter 2 is aimed at developing the physical gearbox model for the overall hybrid model framework. This chapter starts by constructing a generalised stiffness modelling framework that can easily facilitate various gear tooth faults. The chapter introduces four different fault types which are all directly compatible with the generalised stiffness model. The induced faults are compared to the literature to verify the generalised stiffness

modelling framework. The chapter continues by implementing a dynamic gearbox model, from which two synthetic datasets are created for later experimentation. As a verification step, the vibrations from these datasets are compared against the literature on which they are based.

Chapter 3 develops a supervised data-driven model which can take input vibration data and classify the given signal into a gearbox condition. Furthermore, with the synthetic data from chapter 2 and the data-driven model from chapter 3, the full hybrid model is shown and steps for its implementation are described. Up to this point the full hybrid model is developed and ready for experimentation.

Chapter 4 experimentally proves that the developed hybrid model is robust against noise and that it is capable of identifying different fault types within a gearbox. With the hybrid model trained, a robust baseline has been created against which the transfer learning calibration techniques may be investigated.

Chapter 5 addresses the problem of model robustness against a change in input data. This chapter develops the transfer learning calibration tools for the hybrid models, which aims to calibrate the hybrid model. The calibration tools are aimed at minimising the difference in performance between the hybrid model trained on synthetic data and the hybrid model to be implemented on real data. With the hybrid model calibrated with transfer learning, it is compared against the non-calibrated model from Chapter 4. Comparisons are made by running two synthetic experiments. These experiments show the effectiveness of hybrid model calibration for the fault classification task in a data-scarce domain.

Chapter 6 discusses the overall impact of the work and concludes with suggestions for future research.

Chapter 2 Physics-driven portion of the hybrid framework

Within a condition-based maintenance (CBM) framework, one finds three main approaches to monitoring the condition of assets. These are physics-driven, data-driven and hybrid approaches. From the introduction, it is argued that a hybrid model may be advantageous for the gearbox condition monitoring task, as such an approach aims to maximise the advantages of both data-driven and physics-driven approaches. For this work, a physics-driven model is used to generate data for the data-driven portion of the hybrid framework. A physical model (PM) is chosen over alternative data-acquisition techniques, such as real industrial data or experimental setups for a multitude of reasons.

The main drawback of utilising industrial data directly, is a lack of labelled data and a lack of fault samples. It is unrealistic to expect real assets to have a well-maintained database of labelled signals. Assuming that the data was, however, well labelled, it is furthermore not guaranteed that the signals will contain all the different modes of failure of the asset, if any at all. Assets are generally not run to failure, as this leads to unplanned maintenance, which causes long downtimes and full stoppages of production lines. To overcome the data scarcity and lack of labelling problem, experimental setups similar to the real asset may be proposed. Such experiments, however, also have downsides when compared to PM approaches:

1. A PM provides us with a cost-effective manner to generate data similar to what we would obtain from an actual gearbox. Gearboxes are generally expensive and hard to access for sensory measurements in practice and therefore a PM results in a much less costly experimental setup.
2. A PM allows us to more flexibly induce faults of different shapes and sizes and apply torques of varying amounts quite easily. Thus, data acquisition is much faster when using a PM.
3. A PM allows us to understand at a fundamental level how different faults should affect the gearbox.
4. A PM allows for a controlled, noise-free environment in which clean data can be generated, such that only the underlying physics of the gearbox is captured within the synthetic data.

The modelling behind a PM requires a deep understanding of the physics that drive the dynamics of the asset and implementing these dynamics in a numerical framework. A PM is a powerful tool for both diagnostic and prognostic approaches. The focus of this work, however, is on gearbox fault diagnosis. Therefore the scope of the physical gearbox model is limited to fault modelling and data generation, but not fault propagation models.

The literature identifies two main approaches to gearbox modelling, namely modulation-based techniques and dynamics-based techniques (Mohammed & Rantatalo 2020). This work focuses on the latter. When employing a dynamics-based gearbox model, it is generally seen that works split up the modelling process into three sub-parts:

1. Gear mesh stiffness modelling
2. Gear fault modelling
3. Gear dynamics modelling

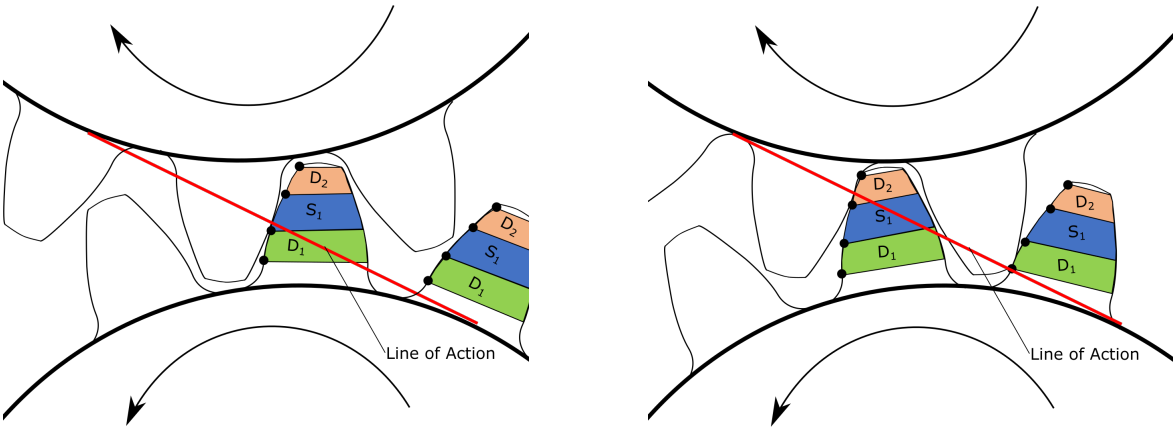
Gear mesh stiffness modelling focusses on modelling the effects which generate meshing stiffness variations between gear pairs. Fault modelling focuses on the methodologies with which we induce faults into the gear model. Finally, dynamics modelling focuses on solving the dynamics of the system, which in turn leads to the main output of the PM, namely synthetic vibration data.

For this work, the Potential Energy (PE) method is used, because it offers a promising compromise between accuracy (continually showing results similar to FEM) and speed (being much faster to solve as opposed to FEM). The PE method allows us to incorporate both mesh stiffness and fault models in one unified framework, which will be developed throughout sections 2.1 (gear mesh stiffness modelling) and 2.2 (fault modelling). The dynamic model which incorporates the fault-altered mesh stiffness will be shown in section 2.3, from which we may finally generate labelled synthetic data, shown in section 2.4.

2.1 Gear mesh stiffness modelling

Gear mesh stiffness modelling receives much attention in literature since the meshing behaviour of gears is the main source of excitation in a gearbox and damage manifests as changes in the gear mesh stiffness. If this is modelled accurately, one should have a good representation of the actual gearbox vibrations.

To show what is meant by "gear meshing stiffness", refer to Figure 2.1. To model the interaction between two gears, we simplify this to two elements that interact by subjecting one another to opposing forces. These forces cause the non-rigid gear teeth to deflect and cause vibrations during rotation. This force-deflection phenomenon may be restated by modelling the stiffness between two gears. The force between two gears ideally acts along the line of action (LOA) (See Figure 2.1). The LOA is important as it ensures that the speed ratio between the two gears stays constant.



(a) Single Tooth Meshing

(b) Double Tooth Meshing

Figure 2.1. Illustration of single and double tooth meshing. Note that for single tooth meshing only a single tooth pair is in contact. During double tooth meshing, however, we see that two pairs of teeth are in contact at a single time. This effect occurs because the gear teeth are confined (and designed) to interact along the line of action.

Two main effects which affect the gear meshing stiffness need to be modelled. Firstly, notice that as the gear rotates (See Figure 2.1), the point of contact between two gear teeth changes its location and angle relative to a given gear tooth. For example, on the left, looking from only the first coloured tooth's perspective, one has a meshing force near the interface between D_1 and S_1. Then, some while later, on the right, one sees the meshing force has moved to a point near the interface between S_1 and D_2. Further notice that between the two contact

cases, the angle of the tooth has changed, but the angle of the LOA has not. Therefore, the relative angle between the meshing force and the tooth face has changed. This simultaneous variation of force location and angle on the tooth affects the stiffness of the tooth.

Secondly, notice that on the left, only a single tooth pair is in contact. On the right, two pairs of teeth are in contact. This implies that at the point at which the meshing changes from a single tooth pair to two tooth pairs, the force is suddenly shared between a larger number of tooth pairs. This leads to a sudden relief in force on the initial single tooth pair. This in turn gives rise to a smaller deflection between the gears overall, hence to a larger equivalent stiffness. This effect is very important to model, as it causes discontinuities in the meshing stiffness, which show up as impulses in the dynamic system.

Each of the above-mentioned effects is extensively addressed in the following sections.

2.1.1 Force variation effect

The force variation effect has received much attention through the PE method. Before detailing the PE method, it is helpful to define a reference coordinate system and parameter names. To simplify modelling, the tooth is assumed to be stationary with only the force (F) varying in its location and angle (α_1) on the tooth (see Figure 2.2). This is a convenient way of setting up the coordinate system, with the only complexity arising when one defines how the force dynamically changes location and angle. This complexity is dealt with later.

Due to the way that gear teeth are manufactured, they are defined by two key radii. These are known as the base radius (R_b) and the root radius (R_r). For gears with fewer than 41 teeth (see Figure 2.2), the cutting tool creates a base radius that is larger than the root radius. However, when we exceed the 41-teeth threshold, we see that the root circle becomes larger than the base circle (see Figure 2.3) (Liang et al. 2014, Wan et al. 2014). This is only valid for standard spur gears and most authors follow this assumption.

The root radius refers to the radius where the gear tooth contacts the gear body and should be used as the reference for all coordinates. In the early years after the potential energy method was proposed, authors did not consider the tooling effect, and hence defined all parameters with reference to the base radius (essentially assuming $R_b = R_r$). The base radius is the location (for a gear with less than 41 teeth) where the fillet of the gear tooth begins, and has no real meaning for a gear with more than 41 teeth, but can still be calculated.

At this point, it is useful to explain the potential energy method. This method assumes a gear tooth to be a non-uniform cantilevered beam and applies beam theory to calculate the potential energy due to strain within this tooth. The potential energy (strain energy) for a beam is defined by Hibbeler (2004) as:

$$U = \frac{F\delta}{2} = \frac{F^2}{2k} \tag{2.1}$$

where F, δ and k relate to the force applied to the beam, the deflection of the beam in the direction of the force and the stiffness of the beam respectively. Note that this equation serves as the general basis from which all stiffness terms will be derived.

Researchers mainly consider five types of potential energy to derive the stiffness terms. The first three stem directly from beam theory, namely a bending (U_b), axial compressive (U_a) and shear (U_s) energy. In addition to these terms, the Hertzian (U_h) line contact and fillet foundation (U_f) energy are also considered. These strain energies are discussed in the subsequent sections.

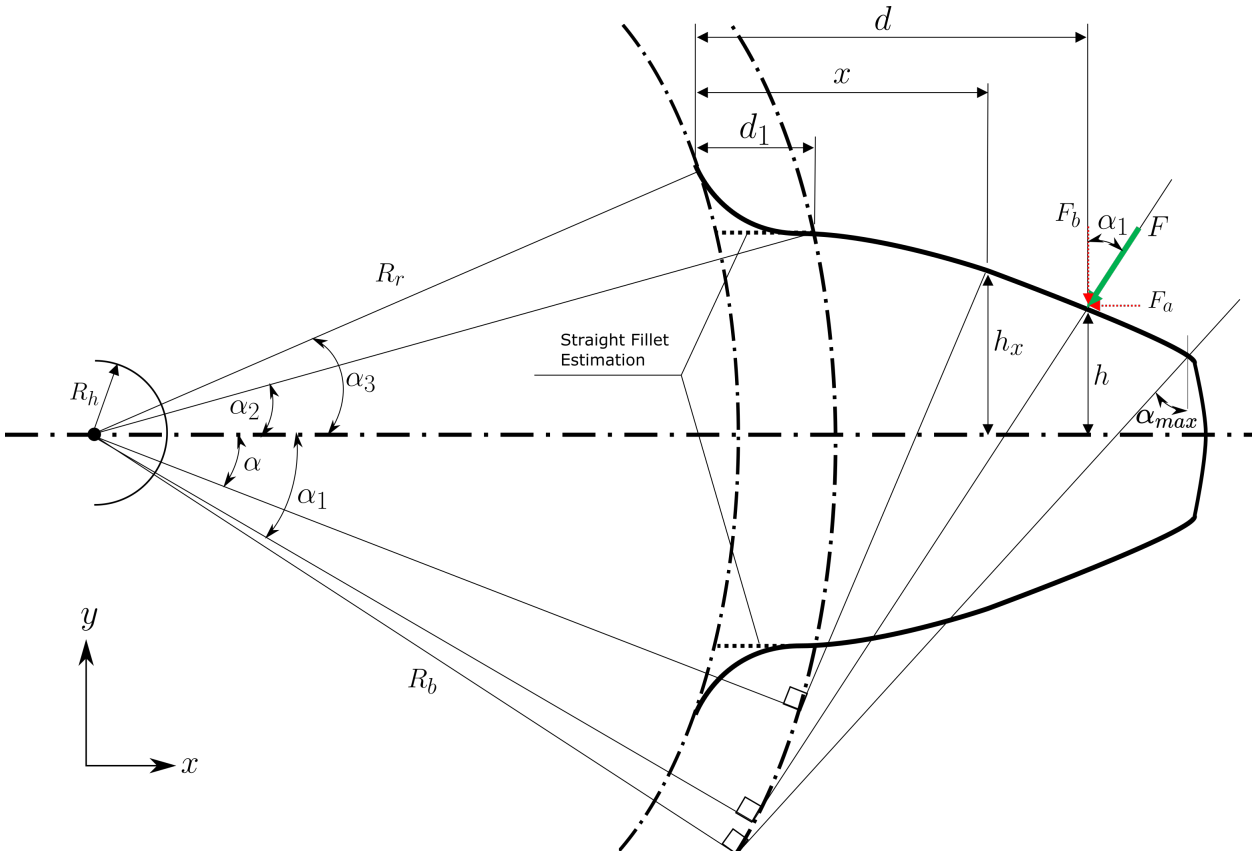


Figure 2.2. Tooth model when the number of gear teeth (Z) is less than or equal to 41 ($Z \leq 41$). This has the effect that the base radius (R_b) is larger than the root radius (R_r). Due to this factor, an extra piece of tooth material needs to be accounted for between the root and base, which was neglected in older methods. This extra portion of material is denoted by d_1.

2.1.1.1 Beam energy terms: U_a, U_b, U_s

The potential energies associated with beam theory, namely axial compressive, bending and shear energies are defined as follows (Tian 2004):

$$U_a = \frac{F^2}{2k_a} = \int_0^d \frac{F_a^2}{2EA_x} dx \tag{2.2}$$

$$U_b = \frac{F^2}{2k_b} = \int_0^d \frac{[F_b(d-x) - F_a h]^2}{2EI_x} dx \tag{2.3}$$

$$U_s = \frac{F^2}{2k_s} = \int_0^d \frac{1.2F_b^2}{2GA_x} dx = \int_0^d \frac{1.2(1+\nu)F_b^2}{2EA_x} dx \tag{2.4}$$

where ν, E and G refer to Poisson's ratio, modulus of elasticity and shear modulus of the gear. The quantities of h, d and x are shown in Figure 2.2 and refer to the height and distance along the tooth at the force contact point, and the instantaneous length along the tooth one is currently at. The instantaneous cross-sectional area and area moment of inertia is given by A_x and I_x respectively. The force terms may be defined as follows (by looking at Figure 2.2 once more):

$$F_b = F \cos \alpha_1 \tag{2.5}$$

$$F_a = F \sin \alpha_1 \tag{2.6}$$

where F refers to the contacting force and α_1 refers to the angle between the tooth centreline and the line which connects the tooth origin to the point of tangent contact of the contact force, were it to be extended. This quantity will be defined shortly.

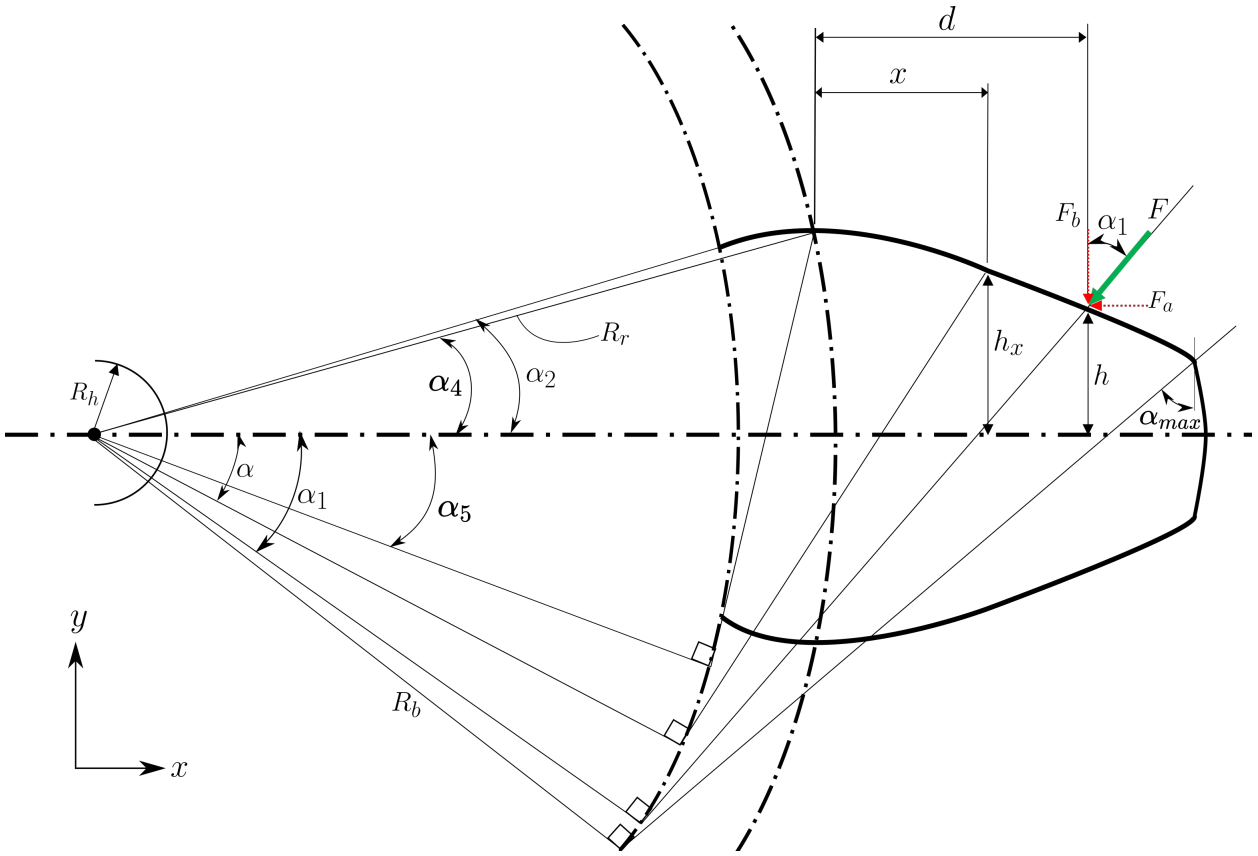


Figure 2.3. Tooth model when the number of gear teeth (Z) is greater than 41 ($Z > 41$). This has the effect that the root radius (R_r) is larger than the base radius (R_b). Due to this factor, an extra piece of tooth material is being accounted for utilising older methods and thus the limits of the tooth need to be redefined, seen by the α_4 parameter.

The tooth figures in Figure 2.2 and Figure 2.3 may be given the coordinate designation of an $x-y$ plane. However, integration along the length of the beam in any designation of an $x-y$ coordinate system is difficult, as we need an expression for the change in area cross-section (A_x) and area moment of inertia (I_x) in terms of a change in the x-direction. Furthermore, we need a definition for the change in tooth height (h) as we progress along x. It is advantageous if one could rather integrate in an angular coordinate system, where all terms are a function of an angular quantity related to x, namely α.

If this is done, we may more easily define our profile of the gear. When converting the $x-y$ coordinate system quantities (x, h_x, d, h, d_1), we will require a new set of angular parameters to relate them to. We define these quantities as follows (Tian 2004):
For the pinion:

$$\alpha_1(\theta_1) = \theta_1 - \frac{\pi}{2Z_1} - (\tan\alpha_0 - \alpha_0) + \tan\left[\arccos\left(\frac{Z_1\cos\alpha_0}{\sqrt{(Z_2+2)^2+(Z_1+Z_2)^2-2(Z_2+2)(Z_1+Z_2)\cos\left(\arccos\left(\frac{Z_2\cos\alpha_0}{Z_2+2}\right)-\alpha_0\right)}}\right)\right] \tag{2.7}$$

For the driven gear:

$$\alpha_1(\theta_1) = \tan\left(\arccos\left(\frac{Z_2\cos\alpha_0}{Z_2+2}\right)\right) - \frac{\pi}{2Z_2} - (\tan\alpha_0 - \alpha_0) - \frac{Z_1}{Z_2}\theta_1 \tag{2.8}$$

where Z_1, Z_2, α_0 and θ_1 refer to the number of teeth on the pinion gear, number of teeth on the driven gear, angle of the LOA (this is from the external observation frame and thus a constant value) and the pinion gear's rotational angle. We now have a formulation that defines our force angle (α_1) as a function of gear rotational angle, which was the main complexity when choosing a stationary $x-y$ coordinate system.

Although we have shown that α_1 is a function of gear rotation angle, the explicit dependency on θ_1 is ignored in subsequent discussions, i.e. $\alpha_1(\theta_1)$ is written as α_1. Finally, the following sub-section deals with the complexities of combining different terms depending on how many pairs of teeth are in contact. Thus, there is no need to formulate α_1 definitions for the case where two pairs of gear teeth are simultaneously in contact here.

Continuing, we find the quantity a_2, which is defined as the half tooth angle of the base circle (Meng et al. 2021):

$$\alpha_2 = \frac{\pi}{2Z} + \tan\alpha_0 - \alpha_0 \tag{2.9}$$

where Z is Z_1 for the pinion and Z_2 for the driven gear.

At this point, a distinction arises. The terms used to define the gear geometry differ depending on the number of gear teeth present. These are presented as:

- Case 1: $Z < 41$, which implies $R_r \leq R_b$
- Case 2: $Z > 41$, which implies $R_b < R_r$

It is useful to note that R_b and R_r are not always necessarily known, but rather a different set of parameters that are used to define the base and root radius. Thus, if the base and root radius is unknown, but the module of the gear is known, the following equations may be used to calculate R_b and R_r (Ma et al. 2014):

$$R_b = 0.5mZ\cos\alpha_0 \tag{2.10}$$

$$R_r = 0.5mZ - (h_a + c)m \tag{2.11}$$

where m refers to the module of the gear and the quantities h_a and c are known as the addendum coefficient and tip clearance coefficient respectively. These are tooling parameters used when cutting gears, and are taken as $h_a = 1$ and $c = 0.25$ in this work, as this is what works such as Ma et al. (2014) and Wan et al. (2014) use and are stated to be the standard values for a spur gear.

Case 1 ($Z \leq 41$):
For this case, we need to define

$$\alpha_3 = \arcsin\left(\frac{R_b\sin\alpha_2}{R_r}\right) \tag{2.12}$$

which aims to locate the angle at which the profile of the tooth reaches the root circle (Meng et al. 2021).

We may now define the transformation equations. Note that for many works the assumption is made that the tooth profile between the tooth base and tooth root is a straight line - Thus h_x will be constant along this region, i.e. the curved region indicated by d_1 in Figure 2.2 is replaced with a horizontal estimate. This assumption holds in this work, as the true fillet profile is difficult to define analytically. Furthermore, the assumption is made that the remainder of the tooth follows an involute profile. The involute profile is a common gear tooth profile that ensures that the line of contact stays constant, making this a valid assumption.

For these assumptions, the following transformation equations are valid (Meng et al. 2021):

$$h = R_b[(\alpha_1+\alpha_2)\cos\alpha_1 - \sin\alpha_1] \tag{2.13}$$

$$d = R_b[(\alpha_1+\alpha_2)\sin\alpha_1 + \cos\alpha_1] - R_r\cos\alpha_3 \tag{2.14}$$

$$d_1 = R_b\cos\alpha_2 - R_r\cos\alpha_3 \tag{2.15}$$

$$x = \begin{cases} x, & \text{if } 0 \le x \le d_1 \\ R_b[(\alpha-\alpha_2)\sin\alpha + \cos\alpha] - R_r\cos\alpha_3, & \text{if } d_1 < x \le d \end{cases} \tag{2.16}$$

$$h_x = \begin{cases} h_{x_1} = R_b\sin\alpha_2, & \text{if } 0 \le x \le d_1 \\ h_{x_2} = R_b[(\alpha_2-\alpha)\cos\alpha + \sin\alpha], & \text{if } d_1 < x \le d \end{cases} \tag{2.17}$$

$$A_x = 2h_xL \tag{2.18}$$

$$I_x = \frac{1}{12}(2h_x)^3L = \frac{2}{3}h_x^3L \tag{2.19}$$

where h and d indicate the height and distance along the tooth at the force contact point. The transition distance between the base and root circle is defined as d_1. The distance and height along the tooth profile are given by x and h_x respectively. The instantaneous cross-sectional area and area moment of inertia is given by A_x and I_x respectively. Finally, L defines the tooth face width. Notice that for x we employ different coordinate systems depending on the tooth region.

This is because the definition of x as a function of α breaks down for any coordinate smaller than the base radius (or in other words for $\alpha > \alpha_2$). If we assume a constant height for any point before d_1, we may formulate an integral in the $x-y$ coordinate system for the region $0 \le x \le d_1$, as we are expecting a constant tooth profile due to the straight-line tooth profile assumption. This makes the integral easy to solve.

We are now equipped to substitute equations (2.13) - (2.17) back into equations (2.2) - (2.4). After substitution we have changed the coordinate system to the angular domain, removed constant terms from the integrals and found the following formulations for the beam related stiffnesses by rearranging terms:

$$\frac{1}{k_a} = \frac{2U_a}{F^2} = \int_0^{d_1}\frac{\sin^2\alpha_1}{2ELh_{x_1}}dx + \int_{-\alpha_1}^{\alpha_2}\frac{R_b(\alpha_2-\alpha)\cos\alpha\sin^2\alpha_1}{EA_x}d\alpha \tag{2.20}$$

$$= \frac{d_1\sin^2\alpha_1}{2ELh_{x_1}} + \frac{R_b\sin^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_2}\frac{(\alpha_2-\alpha)\cos\alpha}{A_x}d\alpha \tag{2.21}$$

$$\begin{aligned}\frac{1}{k_b} = \frac{2U_b}{F^2} &= \int_0^{d_1}\frac{3\left[\cos\alpha_1(d-x)-\sin\alpha_1 h\right]^2}{2ELh_{x_1}^3}dx \\ &+ \int_{-\alpha_1}^{\alpha_2}\frac{R_b^3\left\{1+\cos\alpha_1[(\alpha_2-\alpha)\sin\alpha-\cos\alpha]\right\}^2(\alpha_2-\alpha)\cos\alpha}{EI_x}d\alpha\end{aligned} \tag{2.22}$$

$$\begin{aligned}&= R_b^3\frac{\left[1-\frac{R_r}{R_b}\cos\alpha_1\cos\alpha_3\right]^3 - \left[1-\cos\alpha_1\cos\alpha_2\right]^3}{2EL\cos\alpha_1 h_{x_1}^3} \\ &+ \frac{R_b^3}{E}\int_{-\alpha_1}^{\alpha_2}\frac{\left\{1+\cos\alpha_1[(\alpha_2-\alpha)\sin\alpha-\cos\alpha]\right\}^2(\alpha_2-\alpha)\cos\alpha}{I_x}d\alpha\end{aligned} \tag{2.23}$$

$$\frac{1}{k_s} = \frac{2U_s}{F^2} = \int_0^{d_1}\frac{2.4(1+\nu)\cos^2\alpha_1}{EA_x}dx + \int_{-\alpha_1}^{\alpha_2}\frac{2.4(1+\nu)R_b(\alpha_2-\alpha)\cos\alpha\cos^2\alpha_1}{EA_x}d\alpha \tag{2.24}$$

$$= \frac{1.2(1+\nu)d_1\cos^2\alpha_1}{ELh_{x_1}} + \frac{2.4(1+\nu)R_b\cos^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_2}\frac{(\alpha_2-\alpha)\cos\alpha}{A_x}d\alpha \tag{2.25}$$

Contrary to what is found in literature, equations (2.21), (2.23) and (2.25) are written here with the A_x and I_x terms explicitly visible. This makes the formulas less convenient to work with at this point, but introduces significant advantages for future fault modelling. Although not obvious now, this formulation of the stiffness terms allows for a very simple way to induce faults into the gear mesh stiffness, with no modification necessary

to the stiffness terms. This is an improvement with respect to current practice in the literature, where each different fault type currently requires a different set of stiffness equations.

Case 2 (Z>41):
For this case, we are expecting that the root radius is larger than our base radius (see Figure 2.3). This implies that we should no longer integrate down to the base of the gear, but only to the root. Thus, we have to define a new boundary of integration. We know at which angle this boundary occurs in terms of the **root circle**, namely α_4, however, this is of little use, as we define our coordinate system relative to the **base circle**. Therefore, we need an additional angular quantity, α_5, which defines the equivalent angular point relative to the base circle.

To calculate α_5 is challenging, as we need to solve the following set of equations simultaneously (Liang et al. 2014):

$$R_r \sin\alpha_4 = R_b\left[(\alpha_2-\alpha_5)\cos\alpha_5 - \sin\alpha_5\right] \tag{2.26}$$

$$R_r \cos\alpha_4 = R_b\left[(\alpha_5-\alpha_2)\sin\alpha_5 + \cos\alpha_5\right] \tag{2.27}$$

However, with some manipulation, we may add the two equations after rearranging and squaring to remove α_4 from the formulation, which leads to the following expression which needs to be solved (via Newton's method or any other solver):

$$\left(\frac{R_b}{R_r}\right)^2\left[((\alpha_2-\alpha_5)\cos\alpha_5-\sin\alpha_5)^2+((\alpha_5-\alpha_2)\sin\alpha_5+\cos\alpha_5)^2\right]=1 \tag{2.28}$$

This equation has multiple stable roots, of which the root closest to 0 is required. Visually, one expects $\alpha_5 < \alpha_2$, thus a good first guess is $\alpha_5 = 0$. Then, after solving, ensure $\alpha_5 < \alpha_2$ but also that there are no roots closer to α_2. It seems as if any initial guess between -0.9 and 1 still iterated to the correct α_5 value, thus the required root is not too sensitive to incorrect initial conditions, considering that -1 to +1 radians span a large angle. To be safe, however, and as mentioned before, $\alpha_5 = 0$ is recommended as a starting value.

For some formulas, the value of α_4 will be required, and thus also needs calculation. This may be found from rearranging equation (2.26):

$$\alpha_4 = \arcsin\left(\frac{R_b}{R_r}((\alpha_2-\alpha_5)\cos\alpha_5-\sin\alpha_5)\right) \tag{2.29}$$

We may now define our transformation equations once more. We find that equations (2.13), (2.18) and (2.19) stay precisely the same, and will not be repeated:

$$d = R_b[(\alpha_1+\alpha_2)\sin\alpha_1+\cos\alpha_1]-R_r\cos\alpha_4 \tag{2.30}$$

$$x = R_b[(\alpha-\alpha_2)\sin\alpha+\cos\alpha]-R_r\cos\alpha_4 \tag{2.31}$$

$$h_x = R_b[(\alpha_2-\alpha)\cos\alpha+\sin\alpha] \tag{2.32}$$

Once again, we may set up the stiffness integrals:

$$\frac{1}{k_a}=\frac{2U_a}{F^2}=\int_{-\alpha_1}^{\alpha_5}\frac{R_b(\alpha_2-\alpha)\cos\alpha\sin^2\alpha_1}{EA_x}d\alpha \tag{2.33}$$

$$=\frac{R_b\sin^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_5}\frac{(\alpha_2-\alpha)\cos\alpha}{A_x}d\alpha \tag{2.34}$$

$$\frac{1}{k_b}=\frac{2U_b}{F^2}=\int_{-\alpha_1}^{\alpha_5}\frac{R_b^3\{1+\cos\alpha_1[(\alpha_2-\alpha)\sin\alpha-\cos\alpha]\}^2(\alpha_2-\alpha)\cos\alpha}{EI_x}d\alpha \tag{2.35}$$

$$=\frac{R_b^3}{E}\int_{-\alpha_1}^{\alpha_5}\frac{\{1+\cos\alpha_1[(\alpha_2-\alpha)\sin\alpha-\cos\alpha]\}^2(\alpha_2-\alpha)\cos\alpha}{I_x}d\alpha \tag{2.36}$$

$$\frac{1}{k_s}=\frac{2U_s}{F^2}=\int_{-\alpha_1}^{\alpha_5}\frac{2.4(1+\nu)R_b(\alpha_2-\alpha)\cos\alpha\cos^2\alpha_1}{EA_x}d\alpha \tag{2.37}$$

$$=\frac{2.4(1+\nu)R_b\cos^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_5}\frac{(\alpha_2-\alpha)\cos\alpha}{A_x}d\alpha \tag{2.38}$$

This concludes the required beam related stiffness calculations. The following section deals with the Hertzian contact line stiffness effect.

2.1.1.2 Hertzian line contact term: U_h

The Hertzian energy (U_h) is stored in the contact regions surrounding the line of contact between the two meshing teeth. This occurs due to the local deformation which occurs when two surfaces interact. See Figure 2.4 for a visual representation. The stiffness associated with the contact line between the two gears is modelled with the contacting gear faces assumed to be cylindrical (El Yousfi et al. 2020).

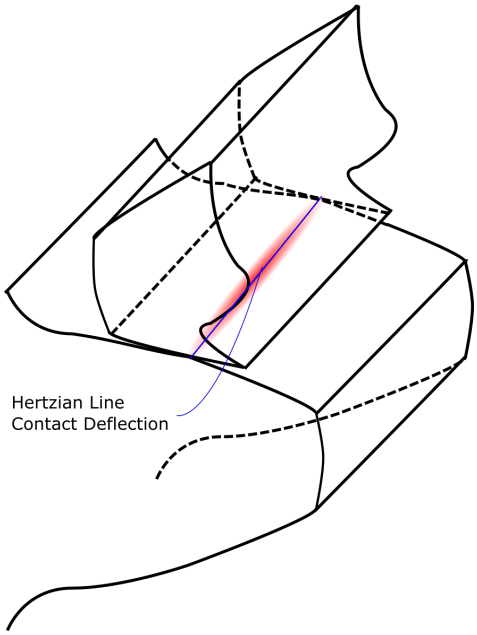


Figure 2.4. Representation of what is meant by the Hertzian line contact stiffness. The portion of the two gears which are in contact deflect locally along the contact line. The deflection model is based upon the assumption that two infinitely long parallel-axis cylinders are in contact.

The Hertzian energy term, with the cylindrical contact assumption, is therefore defined as follows:

$$U_h = \frac{F^2}{2k_h} = \frac{F^2}{2} \times \frac{4(1-\nu^2)}{\pi EL} \tag{2.39}$$

which leads to the equivalent stiffness term (Yang & Lin 1987):

$$\frac{1}{k_h} = \frac{2U_h}{F^2} = \frac{4(1-\nu^2)}{\pi EL} \tag{2.40}$$

2.1.1.3 Fillet foundation term: U_f

Finally, the fillet foundation stiffness term needs to be determined. This term tries to account for the effect of the gear's body deflecting under a load on the tooth. This term has received special modelling attention and has been approximated by a parametric formula (Sainsot et al. 2004). The modelling procedure uses the notation defined in Figure 2.5.

We may show the energy stored in tooth base deflection as a parametric equation from Sainsot et al. (2004):

$$U_f = \frac{F^2}{2k_f} = \frac{F^2}{2} \times \frac{\cos^2\alpha_1}{EL}\left[L^*\left(\frac{\mu_f}{S_f}\right)^2 + M^*\left(\frac{\mu_f}{S_f}\right) + P^*(1+Q^*\tan^2\alpha_1)\right] \tag{2.41}$$

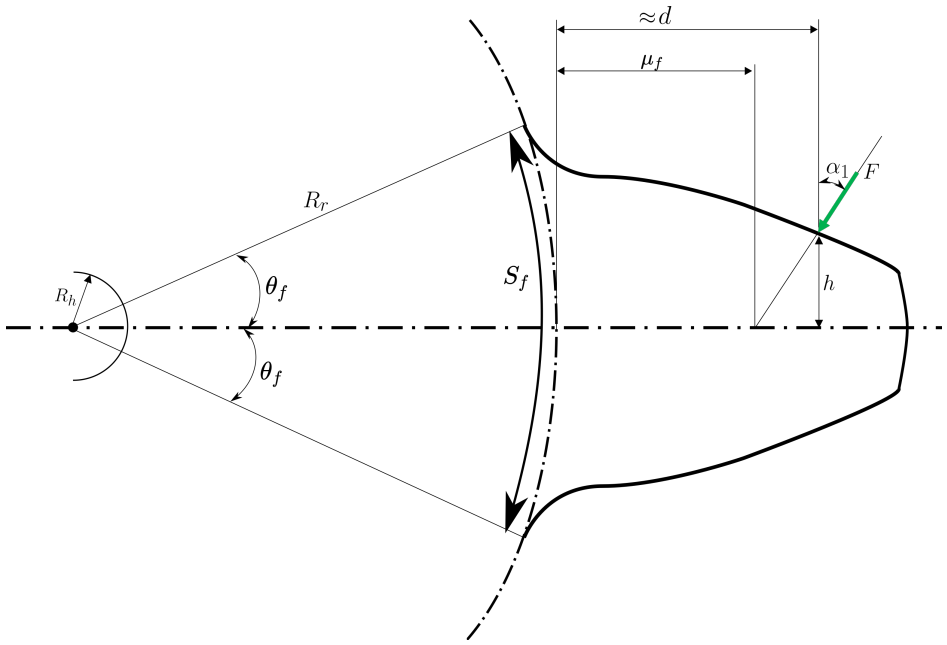


Figure 2.5. Representation of the important parameters required to calculate the fillet effect.

with the equivalent stiffness term given as:

$$\frac{1}{k_f} = \frac{2U_f}{F^2} = \frac{\cos^2\alpha_1}{EL}\left[L^*\left(\frac{\mu_f}{S_f}\right)^2 + M^*\left(\frac{\mu_f}{S_f}\right) + P^*(1+Q^*\tan^2\alpha_1)\right] \tag{2.42}$$

We may define μ_f and S_f (according to Figure 2.5) as follows:

$$\mu_f = d - h\tan\alpha_1 \tag{2.43}$$

$$S_f = 2R_r\theta_f \tag{2.44}$$

$$\theta_f = \frac{1}{Z}\times\left[\frac{\pi}{2} + 2\tan\alpha_0\times(h_a-\bar{r}_c) + \frac{2\bar{r}_c}{\cos\alpha_0}\right] \tag{2.45}$$

where $h_a = 1$ (as from the definitions in equations (2.10) and (2.11) and $\bar{r}_c = 0.2$ as taken from Sainsot et al. (2004) and is defined as the dimensionless tool tip radius.

Finally, we define L^*, M^*, P^* and Q^* by substituting their coefficient values into the following equation, defined by Sainsot et al. (2004):

$$X^*(h_i,\theta_f) = \frac{A_i}{\theta_f^2} + B_i h_i^2 + \frac{C_i h_i}{\theta_f} + \frac{D_i}{\theta_f} + E_i h_i + F_i \tag{2.46}$$

where $h_i = R_r/R_h$, with R_h being defined as the hub radius of the gear. A_i, B_i, C_i, D_i, E_i and F_i may be found in Table 2.1.

Table 2.1. Coefficients required for the polynomial curve fit for fillet foundation stiffness (Sainsot et al. 2004).

	A_i	B_i	C_i	D_i	E_i	F_i
L*(h,θ_f)	-5.574E-5	-1.9986E-3	-2.3015E-4	4.7702E-3	0.0271	6.8045
M*(h,θ_f)	60.111E-5	28.100E-3	-83.431E-4	-9.9256E-3	0.1624	0.9086
P*(h,θ_f)	-50.952E-5	185.50E-3	0.0538E-4	53.3000E-4	0.2895	0.9236
Q*(h,θ_f)	-6.204E-5	9.0889E-3	-4.0964E-4	7.8297E-3	-0.1472	0.6904

This fillet foundation stiffness term assumes only a single tooth pair in contact. If one were to fully account for

this effect for the double tooth region, one must employ a FEM simulation. The FEM simulation is required to find a calibration term that is multiplied with the fillet stiffness during the double tooth meshing region. This procedure is outlined in Ma et al. (2016). This approach, however, is time-consuming and would draw away from the focus of this work, as it requires a FEM model to be developed. The motivation for using the PEM was that the FEM may be avoided.

It is difficult to gauge the effect that FEM calibration alone has on a model, as Ma et al. (2016) simultaneously considered multiple additional improvements to the gear mesh stiffness modelling framework. It was however argued that neglecting the FEM calibration would result in an overestimation of the mesh stiffness. Therefore, the current work acknowledges that the resulting mesh stiffness may be overestimated. The overall trajectory of this work is not towards a fully representative physical model, as a data-driven approach will accompany and potentially "calibrate" any major differences incurred between a developed model and a true model. Therefore, to save time and maintain focus on the key aspects of this work, the FEM calibration will not be implemented.

Lastly, it seems from the literature that the effective face width term (L) for the fillet foundation stiffness is never reduced by faults, and therefore the same approach is used here. That is to say that for the fillet foundation stiffness, the L parameter will never be affected by faults.

This completes the process of modelling the gear meshing stiffness. The following section provides the process of accounting for the number of teeth in mesh at any given time and is the final modelling step before we may generate mesh stiffness results.

2.1.2 Tooth pair variation effect

At this point, a set of equations have been developed which can give the stiffness at a given rotational position θ_1, for both the pinion or driven gear. For a given rotation angle, we may write the equivalent stiffness between two gear teeth as follows (Tian 2004):

$$k_t(\theta_1) = \frac{1}{\frac{1}{k_h} + \frac{1}{k_{bm1}} + \frac{1}{k_{f1}} + \frac{1}{k_{bm2}} + \frac{1}{k_{f2}}} \tag{2.47}$$

where θ_1 refers to the **pinion gear** rotational angle. The k_h term is shared between a tooth pair, but k_f is not, and therefore the pinion and driven gear fillet foundation stiffnesses are added independently as k_{f1} and k_{f2} respectively. The k_{bm} parameter refers to the equivalent 'beam stiffness' of a gear (pinion or driven) and is given by:

$$k_{bmi} = \frac{1}{\frac{1}{k_{ai}} + \frac{1}{k_{bi}} + \frac{1}{k_{si}}} \tag{2.48}$$

where $i = 1$ refers to the pinion and $i = 2$ refers to the driven gear. Thus, we calculate k_{bm1} and k_{f1} by using the definition of α_1 from equation (2.7) (the pinion gear), and calculate k_{bm2} and k_{f2} by using the definition of α_1 from equation (2.8) (the driven gear). Very importantly, when calculating the stiffness terms utilising either $i = 1$ or $i = 2$, the relevant quantities must change throughout the stiffness terms as well. For example, in the stiffness equations, R_b refers to the base radius, therefore, when calculating for the pinion, utilise $R_b = R_{b1}$ and for the driven gear $R_b = R_{b2}$, where R_{b1} is not necessarily equal to R_{b2}, as the pinion and driven gear have different properties. Lastly, regardless of which gear is being calculated for, the gear rotation angle always refers to the pinion gear. That is to say that θ_1 is always used, as was made clear in both equations (2.7) and (2.8).

Recall once again that for any given angle θ_1 of the pinion gear, we get a new k_t value, as a change in θ_1 affects the value of α_1, in turn affecting the length of the tooth over which we integrate. Thus, it becomes key to define the limits of θ_1, which will be limited to the gear tooth shape. To do this, it is important to understand what stages a tooth experiences during meshing. Referring back to Figure 2.1 one sees three meshing zones, namely D_1, S_1 and D_2.

The following explanation follows only the middle tooth pair from Figure 2.1. During the period of rotation where the contact force falls within D_1, the middle coloured tooth, as well as the left uncoloured tooth will be in simultaneous contact with the opposite pair of teeth on the driven gear, leading to the first double meshing region,

D_1. Then, after some rotation, the contact force enters region S_1. For this stage, only the middle tooth pair is in contact, leading to the first (and only) single tooth meshing region, S_1. Finally, after some more rotation, two tooth pairs are again in contact, but this time the middle and right tooth. This is shown as double tooth meshing region D_2. Note that if one looks from the perspective of the right-most tooth, this is double tooth meshing region 1. Therefore, for all tooth pairs, the second double tooth meshing region will be the first double tooth meshing region for the following tooth pair. This implies that for each gear on a tooth, we may simply calculate the stiffness terms over the full region from the start of D_1 to the end of D_2 independently and then superimpose the results for all gear tooth pairs. The superposition will occur with offsets depending on where teeth begin and end in the angular domain. This description might be unclear at present, but it will be clarified shortly. For now, it will be useful to show the angular duration of a double pair meshing region (D_1 and D_2), known as θ_d and a single pair meshing region (S_1), known as θ_s. The formulation is given by Tian (2004):

$$\begin{aligned}\theta_d = &\tan\left(\arccos\frac{Z_1\cos\alpha_0}{Z_1+2}\right)-\frac{2\pi}{Z_1}\\ &-\tan\left[\arccos\frac{Z_1\cos\alpha_0}{\sqrt{(Z_2+2)^2+(Z_1+Z_2)^2-2(Z_2+2)(Z_1+Z_2)\cos\left(\arccos(\frac{Z_2\cos\alpha_1}{Z_2+2})-\alpha_0\right)}}\right]\end{aligned} \tag{2.49}$$

$$\theta_s = \frac{2\pi}{Z_1}-\theta_d \tag{2.50}$$

Recalling that we have two double pair meshing regions and one single pair meshing region, we can define our superposition limits for each gear as a function of θ_1 as follows:

$$(n-1)\times\frac{2\pi}{Z_1}\leq\theta_{1,n}\leq(n-1)\times\frac{2\pi}{Z_1}+\theta_t \tag{2.51}$$

where $\theta_t = 2\theta_d+\theta_s$ and where $\theta_{1,n}$ is used to refer to the rotation limits for gear n. Thus, for a given tooth n (where n $\in[1,Z_1]$ and $n\in\mathbb{Z}$), the stiffness terms should be 0 for any angle outside the given bounds, as the gear will not be in mesh and not contribute to the system's stiffness. Now, to set up the stiffness evaluation over a full gear rotation, we simply calculate the stiffness array for each tooth and superimpose the results as follows:

$$k_t = \sum_{n=1}^{Z_1} k_{t,n}(\bar{\theta}) \tag{2.52}$$

where $\bar{\theta}$ refers to an array of θ_1 values within $[0,2\pi]$.

A slight issue occurs at the first gear tooth and the last gear tooth, as they overlap in physical space, but in a computational array do not. Therefore, the last double meshing region from the final tooth must be removed and added to the first double meshing region for the first tooth. That is to say the values in the final meshing region

$$(Z_1-1)\times\frac{2\pi}{Z_1}+\theta_s+\theta_d\leq\theta_{1,Z_1}\leq(Z_1-1)\times\frac{2\pi}{Z_1}+\theta_t \tag{2.53}$$

must be removed and simply added to the first double meshing region of the first tooth,

$$0\leq\theta_{1,1}\leq\theta_d \tag{2.54}$$

If one employs the stiffness modelling in this way, it makes fault modelling much easier, as we can induce faults on single teeth, and have the results show up during the tooth superposition phase. For now, the procedure mentioned above is seen in Figure 2.6.

We see that individual tooth stiffnesses only have stiffness values over a given period of pinion rotation angle, and are zero otherwise. If we add all Z_1 tooth stiffness arrays together, we obtain the solid line figure, which is the total equivalent stiffness between the pinion and driven gears during any given rotation angle of the pinion. This is precisely what is required when employing a lumped parameter model, where we are constantly updating the shaft rotation angle and would like to know what the equivalent stiffness term should be.

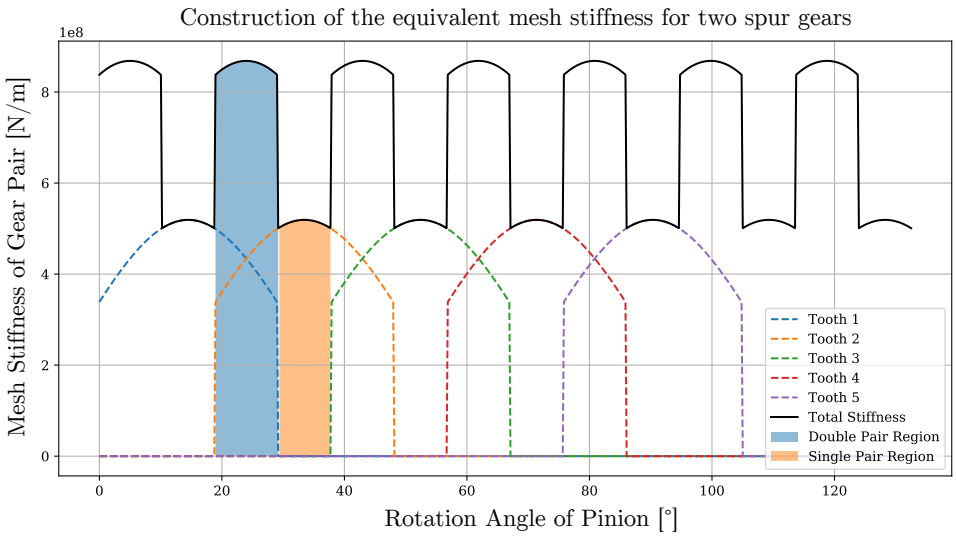


Figure 2.6. Graphic representing how individual stiffnesses of teeth are superimposed to obtain the overall meshing stiffness between two gears for any given rotation angle of the pinion gear

Note that Figure 2.6 contains both meshing behaviours discussed at the start of this chapter. Firstly, the effect of force location variation is captured by the dotted figures (the individual teeth). This shows up as appreciable stiffness variations over a given tooth. Secondly, the effect of different meshing pair numbers is considered based on the location of individual teeth stiffness locations, where overlapping teeth regions (blue coloured region) represent the double pair meshing region and the orange region represents a single pair of teeth in mesh. This effect is larger in mesh stiffness amplitude variation and shows up as discontinuities in the mesh stiffness, which is the main cause of vibrations within a gearbox.

This completes the stiffness modelling portion for the spur gears. The following section explains how various fault types can be induced such as to influence the overall mesh stiffness.

2.2 Gear fault modelling

The true value of the physical model within the hybrid modelling context is realised within fault modelling. The modelling of faults allows the creation of fault data that is otherwise not available from industrial gearboxes. The more accurate the formulations in this section, the better correspondence the generated signals will have with actual gearbox signals. This section presents four fault types:

1. Root Cracks
2. Chips
3. Spalling
4. Surface Pits

After the fault types are given, a generalised stiffness formulation is presented, which can incorporate any of these faults, purely based on the analytical PEM method. To the author's knowledge, this is the first work that attempts to reconcile different fault mechanisms into a single formulation.

2.2.1 Cracks

Crack fault modelling is quite mature in the literature. This is the only fault mechanism considered in this work which does not act on the surface of a gear tooth, but rather into the body of the gear tooth. The model utilised in this work is based on the work of Chen & Shao (2011). In this work, firstly, the crack path is assumed to be constrained to an inclined plane into the gear tooth, i.e. the crack depth propagation into the tooth is linear. This

assumption has been shown to be adequate based on experimental evidence (Lewicki 2002). The crack model utilises an angular value (see Figure 2.7(b)), ν, which defines the angle of the crack plane.

Secondly, the model assumes that the crack depth varies as a function of the tooth width (see Figure 2.7(a) and Figure 2.7(c)). The crack depth shape is modelled using a parabola. The parabolic modelling is split into two cases, with case 1 being shown as a solid line (partial crack width progression case) and a dashed line (full crack width progression case) in Figure 2.7(c).

The first case occurs when the crack has not propagated across the full tooth's width. For this case, one specifies q_0, which is the initial crack depth at one of the tooth edges, and W_c, which describes the position on the tooth width at which the crack ends. This crack scenario is thus defined as (Chen & Shao 2011):

$$q_1(z) = \begin{cases} q_0\sqrt{\frac{L_c - z}{L_c}}, & z \in [0, L_c] \\ 0, & z \in [L_c, L] \end{cases} \tag{2.55}$$

where $q_1(z)$ describes the crack depth as a function of tooth width for the first scenario.

The second case occurs when a full-width crack is present on a tooth. One still specifies the initial crack depth q_0, but now since $L_c = L$, we specify how deep the crack is on the opposite side of the gear side surface, q_2. The corresponding crack model will thus be (Chen & Shao 2011):

$$q_2(z) = \sqrt{\frac{q_2^2 - q_0^2}{L} z + q_0^2} \tag{2.56}$$

where $q_2(z)$ describes the crack depth as a function of tooth width for the second scenario.

After these parameters are chosen, a double integral needs to be performed. More specifically, it is necessary to integrate across the tooth length, as for all stiffness terms, and the tooth width (since the crack depth varies across tooth width. This adds complexity to the proposed generalised approach, and ideally one would want to only take an integral across the tooth length (to fit the proposed generalised model), as in equations (2.2) - (2.4). In their current form, the stiffness terms would require solving by the following integral:

$$k_i = \int_0^L \int_0^x f_i(x,z)\,dx\,dz \tag{2.57}$$

where i refers to the relevant stiffness term (axial, bending, shear, Hertzian or fillet). The quantity $f_i(x,z)$ indicates the terms required to calculate stiffness term k_i. Finally, dx and dz indicate a small step in the length of the tooth and a small width slice of the tooth. As an illustration, consider equation (2.2), which would now have to be rewritten as:

$$U_a^* = \int_0^L \int_0^d \frac{F_a^2}{2EA_x^*(x,z)}\,dx\,dz \tag{2.58}$$

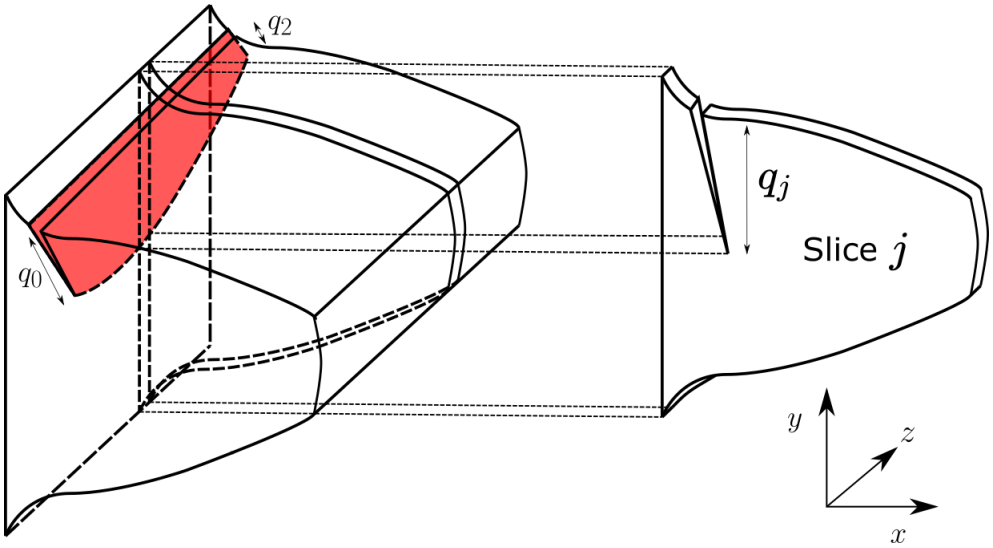
where U_a^* and $A_x^*(x,z)$ indicate a fault-modified axial potential energy and some modified cross-sectional area based on the crack parameters. Note that A_x^* is both a function of the x and z-directions, and may be defined as:

$$A_x^*(x,z) = L \cdot h_q^*(x,z) \tag{2.59}$$

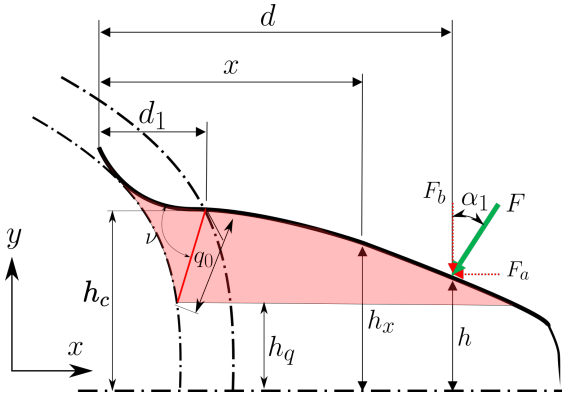
where $h_q^*(x,z)$ indicates the height bound (see Figure 2.7(b)) of the material at a depth x and width z along the tooth.

This formulation shows that a double integral is required to obtain a single stiffness term. It would prove advantageous if the integral along the z-direction could be solved beforehand, leaving only an integral along a single dimension. Before a solution to this problem is proposed, it is useful to understand how one models a varying width tooth crack.

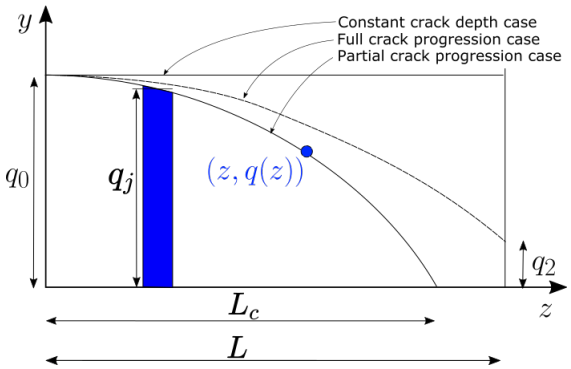
Firstly, one divides the tooth width into n equally small slices. Then, choosing a particular slice j, we obtain an average crack depth at slice j of q_j. Such a slice is seen in Figure 2.7(a). Depending on the specified depth of



(a) Illustration of the three-dimensional nature of the root crack. Note that the crack is induced with varying width from one edge (q_0) to the other edge (q_2). On the right-hand side of the figure is a single slice j, which results in a uniform crack depth of q_j.



(b) Illustration of how the effective height of the tooth is assumed to be reduced within the crack depth region seen in red. We define a crack depth (q_0) and angle (ν) which determine how much material is virtually removed from the tooth stiffness. Figure adapted from Chen & Shao (2011).



(c) Illustration of how the varying crack width is modelled. Here z describes the position on the tooth width, and $q(z)$ defines the variable crack depth (which was previously assumed as a constant q_0) as a function of the tooth width. Figure adapted from Chen & Shao (2011).

Figure 2.7. Illustration of (a) a three-dimensional view of the methodology proposed by Chen & Shao (2011) to improve upon the traditional crack modelling method, (b) the traditional model used to define constant width cracks and (c) the formal notation of the methodology from Chen & Shao (2011).

crack, one normalises the depth to be perpendicular to the tooth length (x-direction), using the crack plane angle, ν (See Figure 2.7(b)):

$$q_{j,norm} = q_j \cdot \sin(\nu) \tag{2.60}$$

where $q_{j,norm}$ indicates the normalised slice depth. Subsequently, the difference in height between the half tooth height (h_c in Figure 2.7(b)) and the normalised slice depth ($q_{j,norm}$) is determined. This gives a quantity called $h_q(q_j)$ (Similar to h_q in Figure 2.7(b)). We may define h_c and $h_q(q_j)$ as:

$$h_c = R_b \sin(\alpha_2) \tag{2.61}$$

$$h_q(q_j) = h_c - q_{j,norm} \tag{2.62}$$

When one integrates across the tooth length (the x-direction integral), one removes all volume associated with tooth heights above $h_q(q_j)$ from the integral. It is assumed that the crack disconnects that entire portion of the tooth, making the effective tooth cross-sectional area much smaller. This effect can be visualised by considering Figure 2.7(b). When integrating, the only area on the tooth which contributes to the stiffness is the area not in red. The integral (more accurately Riemann sum) can then be taken for all n tooth width slices, each with a unique effective tooth height value, $h_q(q_j)$.

The proposed method to remove the extra integral is to pre-solve one of the integrals. More specifically, it will be useful if the integral along the z-direction (further denoted as the z-integral) could be pre-solved. This is possible if the z-integral has an analytical solution. The only parameter which varies as a function of z is the crack depth, $q(z)$ (See Figure 2.7(c)). The true quantity of interest, however, is $h_q(q_j)$, which may be used to quantify the depth of the crack in the meshing stiffness coordinate system.

If we replace the width-varying quantity $h_q(q_j)$ with an equivalent non-varying quantity, the z-integral is removed. The proposed non-varying quantity is the average crack height ($h_{q,ave}$), which may be calculated by using the mean value theorem:

$$h_{q,ave,k} = \frac{1}{L}\int_0^L h_q(q_j)dz \tag{2.63}$$

where k refers to the crack scenario, as described by equations (2.55) and (2.56). After applying the mean value theorem, the average effective crack depth parameters may be found from:

$$h_{q,ave,1} = h_c - \left(\frac{2q_{0,norm}L_c}{3L}\right)\sin(\nu) \tag{2.64}$$

$$h_{q,ave,2} = h_c - \left(\frac{2}{3} \times \frac{q_{2,norm}^3 - q_{0,norm}^3}{(q_{2,norm}^2 - q_{0,norm}^2) + \varepsilon}\right)\sin(\nu) \tag{2.65}$$

where $q_{0,norm}$ and $q_{2,norm}$ are the normalised crack depth limits, calculated from equation (2.60) and ε is a very small number introduced in code (the author uses 1e-9) to mitigate division by zero errors for the case where $q_0 = q_2$.

Figure 2.8 demonstrates the result of applying this formulation. This compares the true crack depth variation ($q(z)$) across a tooth's width in orange to the equivalent average crack depth ($h_{q,ave,k}$) which results from applying the mean value formulation. For both of these graphs, the effective area under the curves is equal, making the formulations equivalent. Essentially, we have transformed the $A_x^*(x,z)$ quantity into $A_x^*(x)$, by removing the dependency of z through taking an average, equivalent heigh parameter. That is to say, the formulation has now become:

$$A_x^* = L \cdot h_q^*(x,z) = L \cdot h_{q,ave,k}(x) \tag{2.66}$$

To the author's knowledge, this is the first time such a formulation has been derived. This formulation is not only equivalent to the integral along the width dimension (z), but also an analytical counterpart to a tedious Riemann sum required for the slicing method utilised by Liang et al. (2014) and Mohammed & Rantatalo (2015). This yields an exact and much faster result. Therefore, not only is this formulation useful for the generalised model mentioned earlier, but also for any crack model which employs a varying crack depth as a function of tooth width. Note, however, that this averaging formulation differs from the z-integral method in the sense that it will

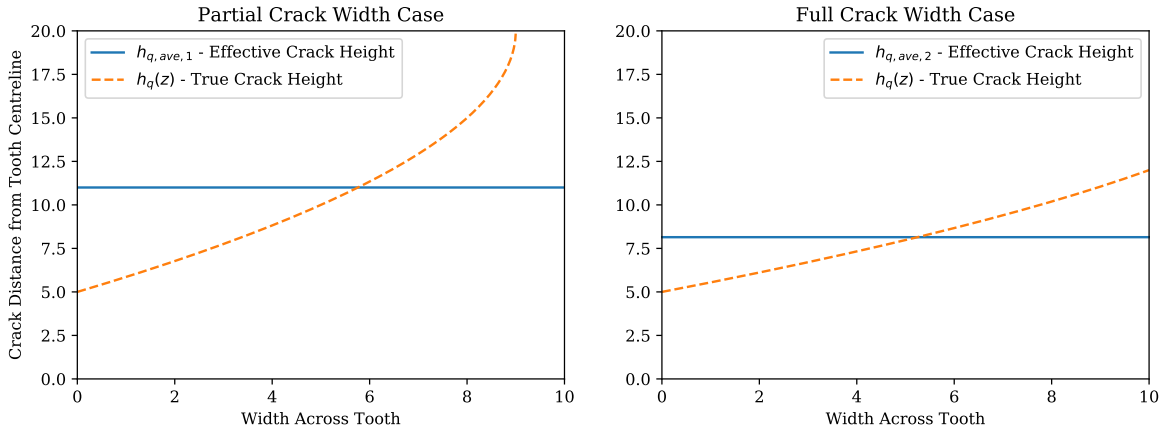


Figure 2.8. Demonstration of the new crack formulation method where a varying crack height profile is converted into its average counterpart.

not result in the same x vs. A_x or x vs. I_x curve, which give the change in cross-sectional area and change in area moment of inertia as a function of tooth depth (x). However, since we are only interested in the **net** integral beneath these curves, the results on the stiffness formulation should be equivalent.

We are now equipped to define the effect a crack fault has on the integrals in equations (2.21), (2.23), (2.25), (2.34), (2.36), (2.38). We redefine the terms for h_{x1}, A_x and I_x. A new variable notation is used to indicate this.

$$h_{x,crack-fault,k} = \begin{cases} h_x, & h_x \leq h_{q,ave,k} \\ \max\left(0, \frac{h_x + h_{q,ave,k}}{2}\right), & h_x > h_{q,ave,k} \end{cases} \tag{2.67}$$

$$A_{x,crack-fault,k} = 2L(h_{x,crack-fault,k}) \tag{2.68}$$

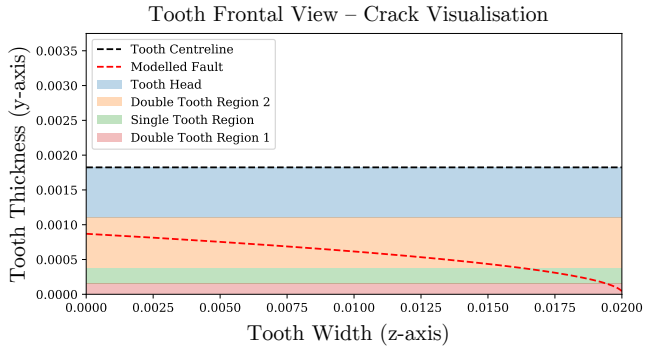
$$I_{x,crack-fault,k} = \frac{2L}{3}(h_{x,crack-fault,k})^3 \tag{2.69}$$

where k once again refers to the crack scenario. Although it seems inconvenient to define the height parameter bounds in terms of the average effective crack depth parameter ($h_{q,ave,k}$), it turns out to be very simple to implement in code.

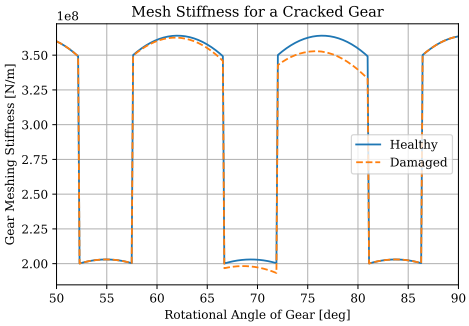
The definition of h_x only depends on the angular variable α (see equations (2.17)) and (2.32), therefore, if required, these bounds can be rewritten as angular bounds. However, this is tedious and results in four unique crack cases to be considered, which unnecessarily overcomplicates the modelling process. The given formulation is equivalent to the four unique angular cases commonly found in the literature.

Note that with this formulation we are also able to accommodate crack faults that cross past the centreline of the tooth. This has the further implication that when $h_{q,ave,j}$ is set to $-h_c$, the resulting equivalent height will be 0 (from the max function), and so too will the area and area moment of inertia. This leads to the same mathematical conclusion as a broken tooth. Thus, this model is further able to model broken tooth faults accurately, which was previously dealt with as a completely separate model in Tian (2004).

An example from this model is given in Figure 2.9(a), where a full-width, non-uniform crack is introduced on a gear tooth. The resulting stiffness from such a crack is seen in Figure 2.9(b).



(a) A visualisation of a crack induced on a gear tooth (Frontal view of tooth).



(b) Resulting gear mesh stiffness from a cracked gear tooth.

Figure 2.9. Illustration of how an induced crack on a gear tooth affects the gear meshing stiffness.

2.2.2 Chips

Chips have not been investigated nearly as much as cracks. Three main models exist in the literature as discussed in Section 1.2.1.2. Preliminary investigations in this work showed that the depth of a chip has very little effect on the overall stiffness change. Hence, the model proposed by Tian (2004), which neglected the influence of the chip depth, is used in this work. The model proposed by Tian (2004) is hyperbolic in nature, and as mentioned above, only accounts for the change in tooth width at the surface of the tooth. That is to say that the effective width of the tooth is assumed to have a much larger effect on the overall stiffness reduction compared to the shallow depth of the chip. The chipping fault therefore only acts on the L term, as the depth effect which would have affected the A_x and I_x terms are being assumed negligible. Therefore, only the Hertzian stiffness term is affected by the chip.

The hyperbolic modelling of the chip profile allows for chips to be induced at the corner of the tooth. Then we may specify with a parameter the depth of the chip down the tooth face (b) as well as the width of the chip across the tooth tip (c). An illustration of this model is given in Figure 2.10.

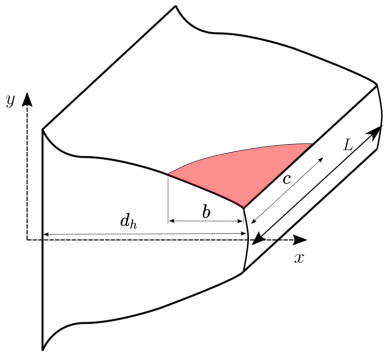


Figure 2.10. Chip model utilised in this work. Adapted from Tian (2004)

Before the formulation for the chip effect may be given, the d_h parameter needs to be defined. The d_h parameter defines the maximum tooth length from the tooth root and may be found from:

$$d_h = \begin{cases} R_b[(\alpha_{max} - \alpha_2)\sin\alpha_{max} + \cos\alpha_{max}] - R_r\cos\alpha_3, & R_r \leq R_b \\ R_b[(\alpha_{max} - \alpha_2)\sin\alpha_{max} + \cos\alpha_{max}] - R_r\cos\alpha_4, & R_b < R_r \end{cases} \tag{2.70}$$

where α_{max} defines the angular position α at which the maximum distance x is reached (See Figure 2.2 and Figure 2.3). The α_{max} parameter may be described by substituting the angle θ_{tot} into equation (2.7). We define

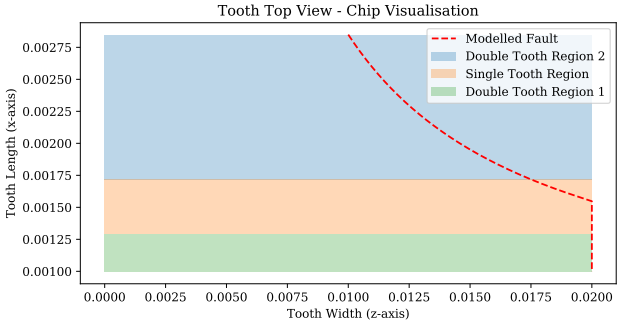
θ_{tot} as follows:

$$\theta_{tot} = 2\theta_d + \theta_s \tag{2.71}$$

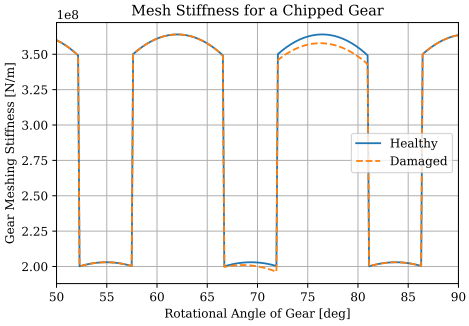
Therefore, $\alpha_{max} = \alpha_1(\theta_{tot})$. With this parameter fully defined, we may finally show the chip formulation. Note, once again, that the formulation only acts on the tooth width (L) as a function of distance along the tooth (x). We therefore leave the A_x and I_x parameters unchanged for chip models. We may describe the effective width function as follows (Tian 2004):

$$L_{chip-fault} = \begin{cases} L, & 0 \leq x \leq d_h - b \\ L - \left(\frac{d_h c}{b} - \frac{1}{x} \left[\frac{d_h^2 c}{b} - d_h c \right] \right), & d_h - b \leq x \leq d_h \end{cases} \tag{2.72}$$

At this stage, we have a chip formulation that only requires two parameters, b and c to fully define the chip's size. Note that the induced chip fault (Figure 2.11(a)) acts only on the tooth face and not into the gear body, as for the crack case. The meshing stiffness in Figure 2.11(b) reflects the fact that the fault was induced to affect the second double tooth meshing region, as well as a small portion of the single tooth meshing region.



(a) A visualisation of a chip induced on a gear tooth (Top view).



(b) Resulting gear mesh stiffness from a chipped gear tooth.

Figure 2.11. Illustration of how an induced chip on a gear tooth affects the gear meshing stiffness.

2.2.3 Spalls

The modelling of spalls and pits are relatively interchangeable, as was seen in the literature review. Although the state-of-the-art spall models have advanced quite a bit from the first models, the choice was made to utilise the older modelling technique proposed by Chaari et al. (2008), as this model has a larger amount of literature to compare results against.

The spall model from Chaari et al. (2008) is rectangular, having constant depth. This model can account for changes in spall width and length on the tooth face, as well as depth changes. Therefore, this fault type affects the effective width of the tooth due to the rectangular surface shape (L_x) but also affects the effective tooth cross-sectional area (A_x) and effective area moment of inertia (I_x) due to the spall depth effect. This implies that this fault type affects the beam stiffness terms as well as the Hertzian term.

The rectangular spall model allows for spalls to be introduced anywhere across the length of the tooth, having any width (w_s), length (l_s) or depth (h_s) and starting at location x_1. An illustration of this model is given in Figure 2.12.

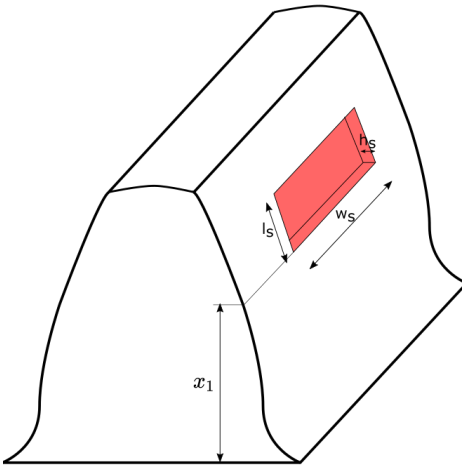


Figure 2.12. Spall model utilised in this work. Adapted from Chaari et al. (2008)

The formulation for the spall model is relatively simple, and may be given as follows (Chaari et al. 2008):

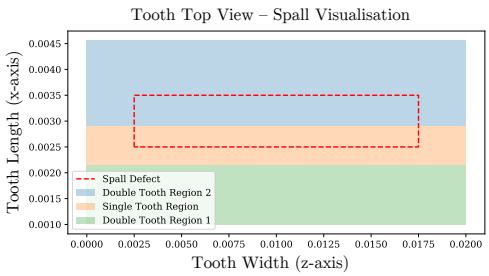
$$L_{spall-fault} = \begin{cases} L, & x \in [0, x_1] \text{ or } [x_1 + l_s, d_h] \\ L - w_s, & x \in [x_1, x_1 + l_s] \end{cases} \tag{2.73}$$

$$A_{spall-fault} = \begin{cases} A_x, & x \in [0, x_1] \text{ or } [x_1 + l_s, d_h] \\ A_x - h_s w_s, & x \in [x_1, x_1 + l_s] \end{cases} \tag{2.74}$$

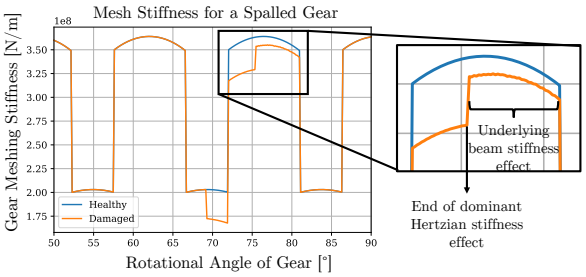
$$I_{spall-fault} = \begin{cases} I_x, & x \in [0, x_1] \text{ or } [x_1 + l_s, d_h] \\ I_x - \frac{w_s}{12} h_s^3, & x \in [x_1, x_1 + l_s] \end{cases} \tag{2.75}$$

where $0 \leq x_1 \leq d_h$ and d_h is defined in equation (2.70).

At this stage, we have a spall formulation that requires four parameters, namely l_s, w_s, h_s and x_1 to fully define the spall's size and location. Notice again, that as was the case with the chip formulation, the induced fault (Figure 2.13(a)) acts on the tooth face plane. Further note that the meshing stiffness (Figure 2.13(b)) reflects the fact that the fault was induced to affect the second double tooth meshing region, as well as a small portion of the single tooth meshing region.



(a) A visualisation of a spall induced on a gear tooth.



(b) Resulting gear mesh stiffness from a spalled gear tooth.

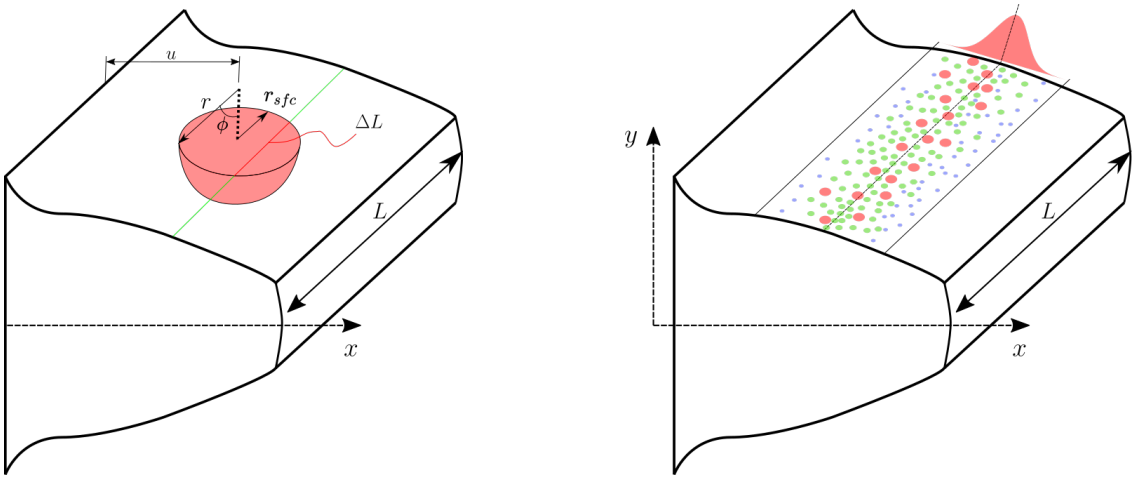
Figure 2.13. Illustration of how an induced spall on a gear tooth affects the gear meshing stiffness.

A subtle, but very important phenomenon is present in Figure 2.13(b). Notice that at 68° the spall fault instantly results in a reduction in mesh stiffness. This continues to around 76°. Past 76 °, the spall fault has been passed, however, the stiffness is still reduced. The main reduction in gear meshing stiffness between 68° and 76° comes from the Hertzian stiffness term. However, after the spall width effect is felt (past 76°), we still see a reduction in stiffness where no Hertzian effect should be present, as the tooth width is not affected anymore. What is occurring, is that the **depth** of material that was removed during the spall portion remains removed for all meshing angles after the spall is induced. Thus, beyond 76° we are observing reductions in the bending and axial stiffness terms, due to the reduction in the effective cross-sectional area and area moment of inertia. This showcases the importance of modelling multiple stiffness effects on the gear tooth.

2.2.4 Pits

The pitting model employed in this work is based on the work in Meng et al. (2021). This model is close to the state-of-the-art in terms of parametric modelling of pits, given its recent development. It was however found that the paper based on this model had minor editorial mistakes, which are corrected in this work.

The pits for this model are all modelled as spheres of "negative volume" or spheres that merely subtract from the volume of whatever object they come in contact with - in this case, a gear tooth. This model may be observed in Figure 2.14(a). A sphere has a centre location relative to the tooth base of u. The sphere has radius r, which corresponds to the size of the pit. Finally, what converts this sphere into a more realistic pit shape, is the contact angle, θ. The contact angle defines the angle between the intersection curve of the sphere and the tooth surface (assumed to be parallel to the tooth centreline) and the sphere centre. Thus, the larger the intersection angle, the deeper the sphere is sunk into the gear tooth, removing more material. Therefore, we can range pit geometries from shallow, wide pits (small θ, large r), to deep thin pits (large θ, small r). This gives the model great flexibility for modelling pits. Due to the nature of pits, it is not enough to have a single model to describe a



(a) Pit model utilised in this work. Size exaggerated for clarity.

(b) Pit distribution model followed in this work.

Figure 2.14. Individual and distributed pit models. Both models were adapted from Meng et al. (2021).

single pit. Pits occur in clusters and are generally distributed across a gear tooth, more likely to be concentrated in the single tooth contact region. In the single tooth region, the forces realised by each gear tooth is higher, as only that single tooth pair is carrying the full torsional load, therefore fault development is exacerbated in this region. When modelling these pits, two main factors have to be accounted for:

1. The location of pits needs to be modelled to concentrate around a certain region (single tooth meshing region).
2. The size of pits needs to be flexible to describe older pits (larger/wider/deeper) and newer pits (smaller/narrower/shallower).

To account for the two factors above, a secondary model for pit modelling is generally required, which can account for multiple pits, each with varying size, and distribute them flexibly across a tooth face.

To solve the first challenge, we simply generate multiple pits (in an array), where the centres of these pits (u_i) are sampled from a normal distribution with mean u_p and standard deviation σ_u. Thus:

$$u_i \sim \mathcal{N}(u_p, \sigma_u) \tag{2.76}$$

where i indicates sampled pit number i. The mean of the distribution may thus be placed at the required area in the single tooth meshing region, with the standard deviation being used to constrain how wide the pits are distributed. With the fault location challenge solved, the challenge of varying fault age/size is still open. Meng et al. (2021) solve this by simply defining 3 sphere radii and 3 contact angle values, which result in 9 unique pit shapes. They then continue to generate differing amounts of each type depending on how severe the pitting case being described is. In this work, however, a more generalised approach is followed. Instead of hard-coding pit shapes, they are once again sampled from normal distributions as follows:

$$r_i \sim \mathcal{N}(r_p, \sigma_r) \tag{2.77}$$

$$\theta_i \sim \mathcal{N}(\theta_p, \sigma_\theta) \tag{2.78}$$

Therefore, any given pit is drawn based on three normal distributions, one for its location, and two for its shape. The logic behind this approach is simple: By drawing the pit shape from a normal distribution, we are assuming that there will be a small number of very large pits (older pits), and a small number of very small pits (more recent), but with the majority of the pits having roughly the same size (the mean pit shape parameters). With these pit shapes defined, we distribute most of them near the single tooth meshing region, and fan them out to become fewer and fewer as we move away from the single tooth region, hence the location distribution.

The effect of this technique may be seen in the visualisation given in Figure 2.14(b). The variation in location is shown visually by the normally distributed faults, whilst the shape variation is shown as different sized and coloured pits across the tooth width.

At this point, it becomes meaningful to formulate these pits in such a way that they once again fit into the generalised stiffness model being proposed. First, we notice that the pit model not only reduces the effective contact length of the tooth by ΔL (see Figure 2.14(a)), but also reduces the effective cross-sectional area (ΔA_x) and area moment of inertia (ΔI_x) across the tooth length and width simultaneously.

Again we encounter a fault shape that needs to be integrated into both the x and z-directions, which is not ideal for the generalised stiffness model. We therefore need to pre-integrate across the z-dimension once again. This has however been done by Meng et al. (2021) and results in the following formulation:

$$x_i = \begin{cases} x, & x \in [u_i - r_i \sin\theta_i, u_i + r_i \sin\theta_i] \\ 0, & x \notin [u_i - r_i \sin\theta_i, u_i + r_i \sin\theta_i] \end{cases} \tag{2.79}$$

$$\Delta L_{pit,i} = 2\sqrt{(r_i \sin\theta_i)^2 - (u_i - x_i)^2} \tag{2.80}$$

$$\Delta R_i = \sqrt{r_i^2 - (u_i - x)^2} \tag{2.81}$$

$$\Delta A_{x,pit,i} = \frac{\Delta R^2 (2\theta - \sin 2\theta)}{2} \tag{2.82}$$

$$\Delta I_{x,pit,i} = \frac{\Delta R^4}{72}\left(18\theta - 9\sin 2\theta \cos 2\theta - \frac{64\sin^6\theta}{2\theta - \sin 2\theta}\right) + \frac{A_x \Delta A_x \left(h_x - \left(\frac{4\Delta R \sin^3\theta}{3(2\theta - \sin 2\theta)} - \Delta R\cos\theta\right)\right)^2}{(A_x - \Delta A_x)} \tag{2.83}$$

where i refers to the ith pit sampled from the N total pits. Note that a small correction was made to the bounds of this formulation (equation (2.79)) when compared to Meng et al. (2021). The bounds used in this formulation make geometrical sense, where the bounds set by Meng et al. (2021) seem to be erroneous. Referring to Figure 2.14(a), one observes that the intersection between the sphere and the tooth surface results in a circle with radius

r_{sfc}. This quantity may be described by:

$$r_{sfc} = r\sin\theta \tag{2.84}$$

Therefore, to find the bounds of influence for a pit, r_{sfc} should be used. Meng et al. (2021), however, utilises r and therefore the correction was made in this work.

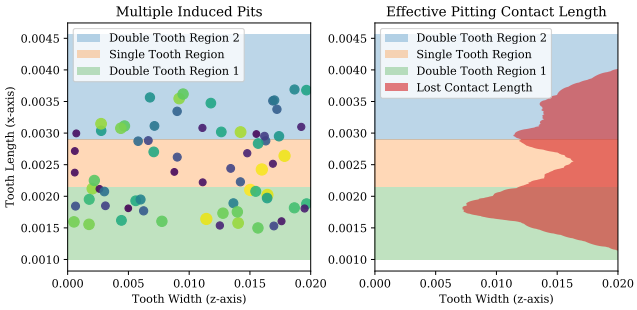
After sampling N pits, the equivalent terms to be used for the generalised model may be found from (Meng et al. 2021):

$$L_{pit-fault} = L - \sum_{i=1}^{N} \Delta L_{pit,i} \tag{2.85}$$

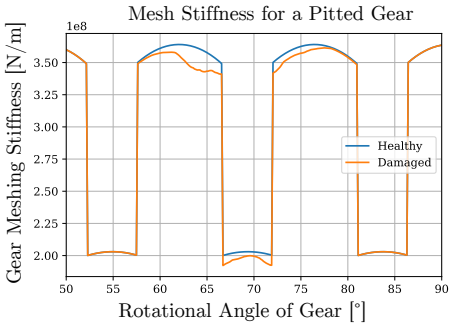
$$A_{x,pit-fault} = A_x - \sum_{i=1}^{N} \Delta A_{x,pit,i} \tag{2.86}$$

$$I_{x,pit-fault} = I_x - \sum_{i=1}^{N} \Delta I_{x,pit,i} \tag{2.87}$$

Throughout this formulation, no attention was given to the width location of the pit. This is because the width information is lost when pre-integrating across the width of the tooth. It is therefore meaningless to specify a width dimension for a given pit fault. Finally, an illustration of induced pits with their equivalent effect on the effective tooth width is given in Figure 2.15(a). The resulting gear meshing stiffness for these pits is seen in Figure 2.15(b).



(a) A visualisation of a spall induced on a gear tooth. Dark colours indicate smaller, newer pits such as blue and dark green. Lighter colours, such as lime and yellow indicate larger, older pits. These pits are not drawn to scale and only serve as an illustrative aid.



(b) Resulting gear mesh stiffness from a spalled gear tooth.

Figure 2.15. Illustration of how an induced spall on a gear tooth affects the gear meshing stiffness.

Note the stochastic nature of the pitting faults on the stiffness graphs. This makes the pitting model unique, as this is the only model with irregular stiffness changes. This roughness may be attributed to the fact that fault locations and shapes are randomly sampled.

2.2.5 General formulation

To summarise the progress and modelling flow of the gearbox model, refer to Figure 2.16. This figure outlines the procedures necessary to develop the generalised stiffness model. At this point, attention will be directed to the portion of work that makes the stiffness model truly general: The incorporation of faults into the stiffness model. This is shown in Figure 2.16. To aid in clarity, for each fault type, a summary of the modelling terms, which parameters they affect, and where in the work the formulas may be found, are given in Table 2.2.

Recall momentarily that there are two main sets of stiffness equations, depending on which case is present. Case 1 refers to when the base radius is larger than the root radius ($Z \leq 41$). Case 2 refers to when the root radius is larger than the base radius ($Z > 41$).

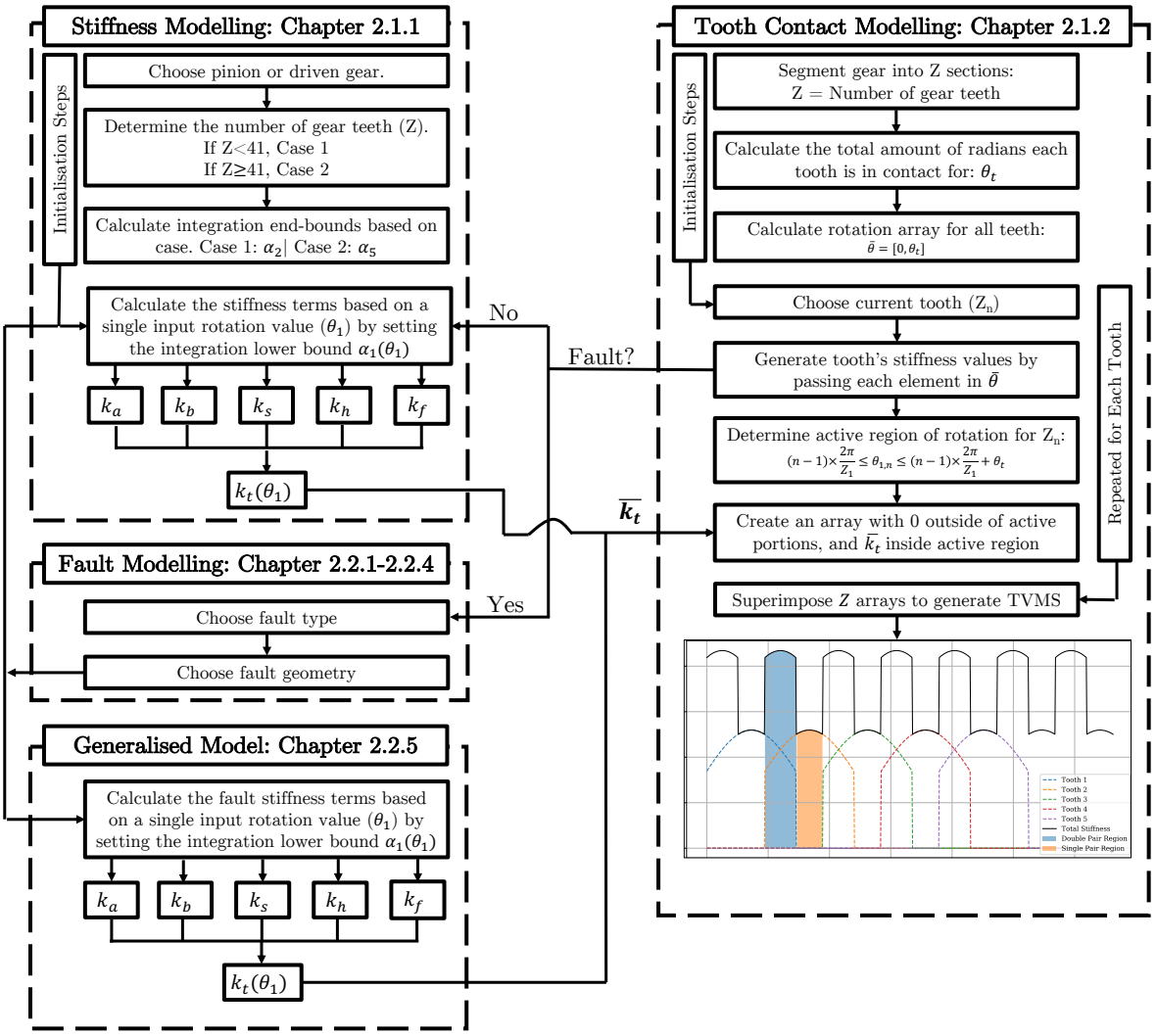


Figure 2.16. Diagram indicating the flow of Chapter 2.

Table 2.2. Summary of fault types and their applicable formulas.

Fault type	Fault Modelling Terms	Affected Parameter(s)	Equation Numbers
Crack	q_0, q_2, L_c, ν	h_x, A_x, I_x	(2.67), (2.68), (2.69)
Chip	b, c	L	(2.72)
Spall	l_s, w_s, h_s, x_1	L, A_x, I_x	(2.73) (2.74) (2.75)
Pit	r, θ, u	L, A_x, I_x	(2.85) (2.86) (2.87)

The new general formulation may now be re-stated for each case as follows:
Case 1 (Z≤41):

$$\frac{1}{k_a} = \frac{d_1 \sin^2\alpha_1}{2EL_{fault}h_{x,fault}} + \frac{R_b \sin^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_2} \frac{(\alpha_2-\alpha)\cos\alpha}{A_{x,fault}}d\alpha \tag{2.88}$$

$$\frac{1}{k_b} = R_b^3 \frac{\left[1-\frac{R_r}{R_b}\cos\alpha_1\cos\alpha_3\right]^3 - [1-\cos\alpha_1\cos\alpha_2]^3}{2EL_{fault}\cos\alpha_1 h_{x,fault}^3} + \frac{R_b^3}{E}\int_{-\alpha_1}^{\alpha_2} \frac{\{1+\cos\alpha_1[(\alpha_2-\alpha)\sin\alpha-\cos\alpha]\}^2(\alpha_2-\alpha)\cos\alpha}{I_{x,fault}}d\alpha \tag{2.89}$$

$$\frac{1}{k_s} = \frac{1.2(1+\nu)d_1\cos^2\alpha_1}{EL_{fault}h_{x,fault}} + \frac{2.4(1+\nu)R_b\cos^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_2} \frac{(\alpha_2-\alpha)\cos\alpha}{A_{x,fault}}d\alpha \tag{2.90}$$

Case 2 (Z>41):

$$\frac{1}{k_a} = \frac{R_b \sin^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_5} \frac{(\alpha_2-\alpha)\cos\alpha}{A_{x,fault}}d\alpha \tag{2.91}$$

$$\frac{1}{k_b} = \frac{R_b^3}{E}\int_{-\alpha_1}^{\alpha_5} \frac{\{1+\cos\alpha_1[(\alpha_2-\alpha)\sin\alpha-\cos\alpha]\}^2(\alpha_2-\alpha)\cos\alpha}{I_{x,fault}}d\alpha \tag{2.92}$$

$$\frac{1}{k_s} = \frac{2.4(1+\nu)R_b\cos^2\alpha_1}{E}\int_{-\alpha_1}^{\alpha_5} \frac{(\alpha_2-\alpha)\cos\alpha}{A_{x,fault}}d\alpha \tag{2.93}$$

And for all cases:

$$\frac{1}{k_h} = \frac{4(1-\nu^2)}{\pi EL_{fault}} \tag{2.94}$$

Recall that the fillet foundation stiffness is not affected by faults, and need not be restated. For the equations (2.88) - (2.94), $h_{x,fault}$, L_{fault}, $A_{x,fault}$ and $I_{x,fault}$ indicate the new effective tooth height, length, cross-sectional area and area moment of inertia. The subscript $fault$ indicates the fault type one wishes to use. For example:

In the case of a crack, we will substitute the $h_{x,fault}$, $A_{x,fault}$ and $I_{x,fault}$ terms from equations (2.88) - (2.94) with $h_{x,crack-fault,j}$, $A_{x,crack-fault,j}$ and $I_{x,crack-fault,j}$ (equations (2.67) - (2.69)), but set the L_{fault} term back to L.

For a chip case, only the L_{fault} term will be modified to $L_{chip-fault}$ (equation (2.72)), with the $h_{x,fault}$, $A_{x,fault}$ and $I_{x,fault}$ terms being reverted to h_x, A_x and I_x.

Therefore, the proposed generalised model simply requires the use of the relevant equations in Table 2.2 to be substituted into equations (2.88) - (2.94), resulting in the equivalent stiffnesses found in the literature. This is a great simplification in the implementation of faults, as it utilises a generalised stiffness formulation, which only requires analytical definitions of $h_{x,fault}$, L_{fault}, $A_{x,fault}$ and $I_{x,fault}$ to be defined for any new fault type which might be modelled. This is a novel idea, as the current literature gives a set of equations for each specific fault type investigated. Note, however, that although the equations may be novel in the way they are written, they produce the same results found in various published works.

To illustrate the general nature of this method, five fault types are introduced on the same gear, two teeth apart, all utilising the new generalised model. This is found in Figure 2.17. Note that the broken tooth fault type is

only an extreme case of the crack model, and therefore not necessarily a *new* case. It does however serve as an illustration that the newly proposed crack model also correctly models a broken tooth, as other modern methods do too.

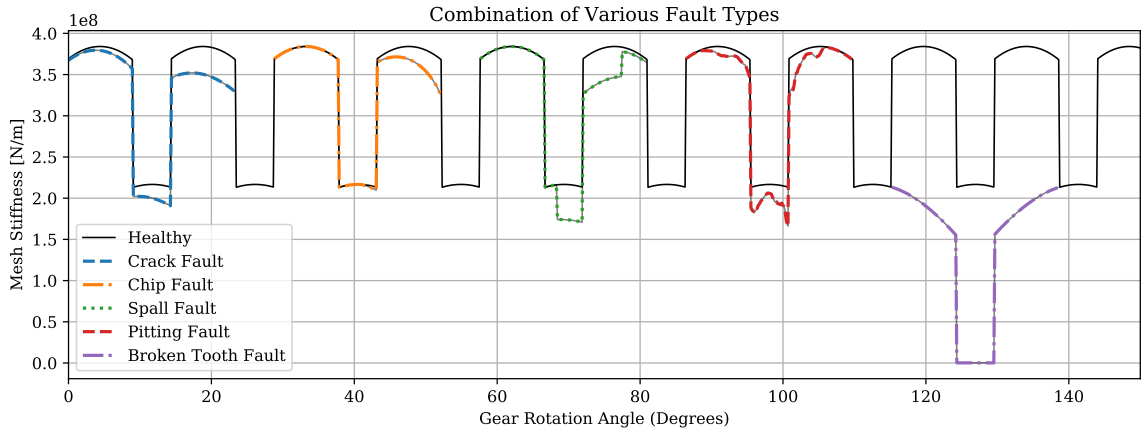


Figure 2.17. Different fault models implemented onto the same gear. This showcases the flexibility of the proposed generalised model to model virtually any analytical fault type in literature.

All of the modelling methodology to this point, as general as it may be, has still not proven to be equivalent to existing works. Therefore, a short section will be given to serve as verification against works in literature.

2.2.6 Verification of the generalised model

The verification procedure is simple. The modelling parameters have been taken from the literature and inserted into the proposed generalised model. The stiffness is then calculated and compared for shape and amplitude against existing works. To aid in the comparison, the generated graphs are also visually matched to those in the literature for optimal readability and comparability.

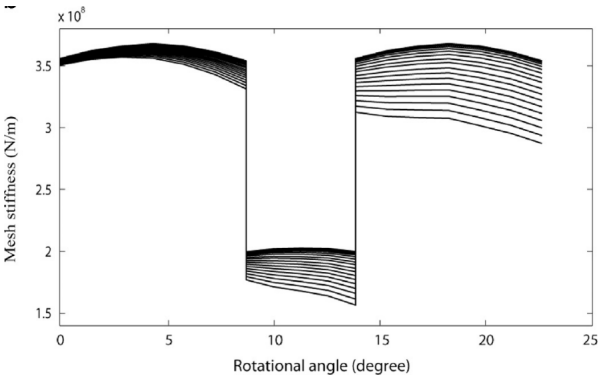
2.2.6.1 Crack verification

To verify the crack model, Mohammed et al. (2013b) is used as a baseline. Mohammed et al. (2013b) not only have a FEM verified model, but also consider all five stiffness effects modelled in this work, making the results of the generalised model directly comparable. Mohammed et al. (2013b) introduced 20 different combinations of fault parameters to generate 20 stiffness graphs (see Figure 2.18(a)). Copying these exact stiffness experiments (Table 6 in Mohammed et al. (2013b)), and applying them to the generalised stiffness model resulted in Figure 2.18(b).

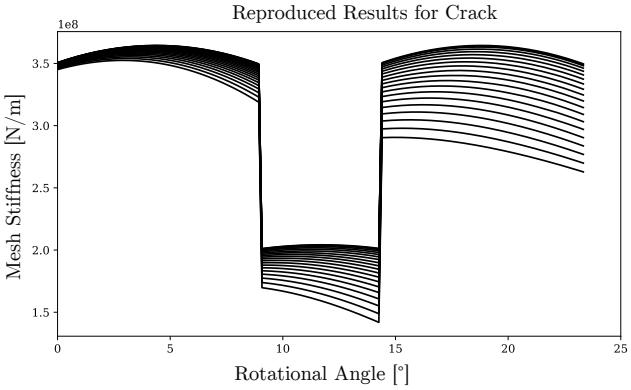
Not only do the shapes of the stiffness graphs correspond very well, but so too do the amplitudes. The generalised stiffness model tends to result in slightly lower stiffnesses for deeper cracks. It is not immediately clear why the slight difference is present, but many modelling factors could lead to small deviations. To name a few: The method in which the contact ratio is calculated differs from Mohammed et al. (2013a). The hub radius is not defined in the works of Mohammed et al. (2013a) and an estimated value had to be taken. Finally, Mohammed et al. (2013a) do not show the angular domain formulation for the stiffness terms, therefore it is hard to gauge which assumptions were made regarding angular parameters. Therefore, given this large set of unknowns from literature, the correspondence is still very good and deemed sufficient.

2.2.6.2 Chip verification

To verify the chip model, Tian (2004) is used as a baseline. Tian (2004) did not include the fillet foundation stiffness effect. Furthermore, Tian (2004) did not account for the stiffness from the tooth area between the tooth root and base. These terms were therefore suppressed in the generalised model to make results comparable. The meshing stiffness resulting from the chip introduced by Tian (2004) is seen in Figure 2.19(a). The stiffness results after copying and running the exact chip parameters from Tian (2004) through the generalised stiffness model may are seen in Figure 2.19(b).

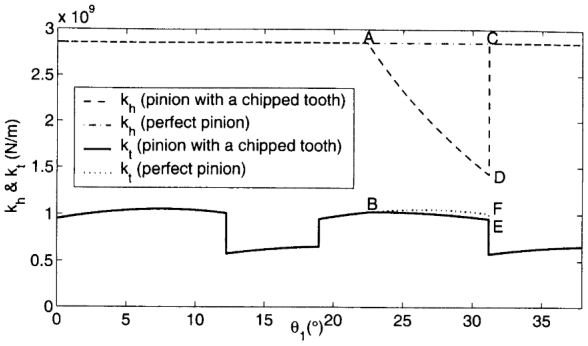


(a) Stiffness experiment results from Mohammed et al. (2013b).

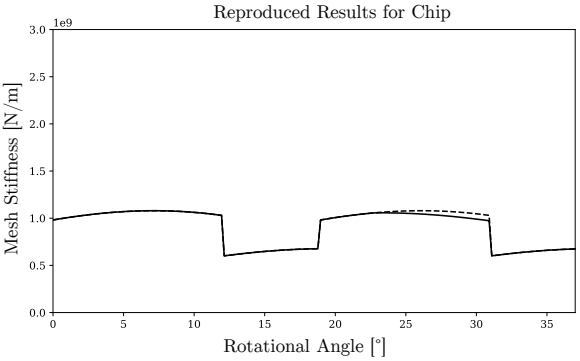


(b) Stiffness results for a crack generated by the generalised stiffness model utilising identical parameters as in Mohammed et al. (2013b).

Figure 2.18. Comparison between literature and generated results for a crack from the generalised stiffness model.



(a) Stiffness experiment results from Tian (2004).



(b) Stiffness results for a chip generated by the generalised stiffness model utilising identical parameters as in Tian (2004).

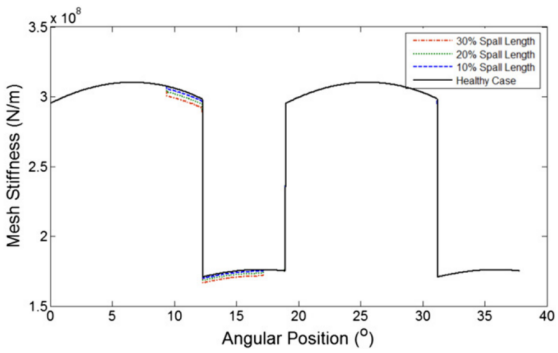
Figure 2.19. Comparison between literature and generated results for a chip from the generalised stiffness model.

Once again perfect correlation in shape is observed, but even more, the amplitudes are exact. This once again shows that the same generalised model which has been verified against cracks is furthermore verified against chips too.

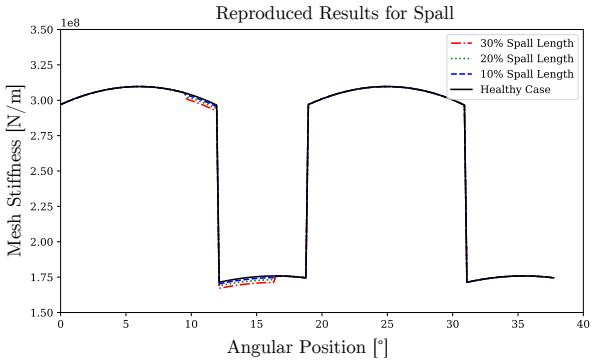
2.2.6.3 Spall verification

To demonstrate the validity of the spall model, Saxena et al. (2016) are used as a baseline. The authors included all relevant stiffness terms, making this model highly comparable to the generalised stiffness model. The spall introduced by Saxena et al. (2016) may be seen in Figure 2.20(a). The stiffness results after copying and running modified spall parameters from Saxena et al. (2016) through the generalised stiffness model are seen in Figure 2.20(b).

Note that for this case, the values from Saxena et al. (2016) are not used, but are slightly modified. The only deviation from Saxena et al. (2016) is the spall width, which according to Saxena et al. (2016) was noted as



(a) Stiffness experiment results from Saxena et al. (2016).



(b) Stiffness results for a spall generated by the generalised stiffness model utilising modified parameters as in Saxena et al. (2016).

Figure 2.20. Comparison between literature and generated results for a spall from the generalised stiffness model.

4mm. This width did not make physical sense given the gear parameters. After trying different values, a spall width of 1mm was found to match up the results.

This correction does not affect the amplitude of the stiffness reduction, but only the location where the spall starts and ends as a function of the rotation angle. A 4mm spall width covers more than one tooth rotation, which is not physically possible. To ensure that the mistake was not in the proposed generalised model, a separate comparison was done against Chaari et al. (2008), and indeed the width parameter, when compared to Chaari et al. (2008), matched perfectly. Thus, this serves as a second confirmation that the generalised stiffness model does interpret the spall parameters correctly. The reason why Chaari et al. (2008) is not used for the verification here, is that it is earlier work and also does not account for all stiffnesses in the same manner as modern techniques.

The comparison with the literature shows the excellent amplitude correlation and also excellent shape correlation of all faults. The slight upward trend of stiffnesses during the single tooth meshing region is perfectly captured by the developed model.

2.2.6.4 Pit verification

To finalise the verification process, the work by Meng et al. (2021) is used as reference. Not only is the pit model built around this work, but also, this work is very recent and should represent close to state-of-the-art modelling results.

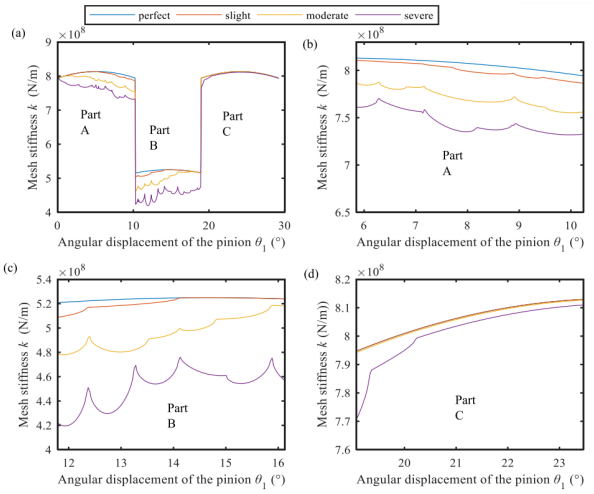
Note that Meng et al. (2021) models one additional effect that this work does not contain: A fillet foundation stiffness correction term. The fillet foundation stiffness term assumes only a single gear tooth pair in contact. Therefore, it is incorrect to extend the formula utilised for this case to the region where two gear tooth pairs are in contact. For the double gear tooth contact, a correction factor is introduced to decrease the fillet foundation stiffness, which is now shared between two load-bearing teeth.

There is no analytical method to determine this factor and therefore it is generally neglected in the literature, as it requires a simulation from a FEM model of the gear to determine the correction factor. This level of accuracy is beyond the scope of this work. Therefore, it should be noted although all the parameters are set to be identical to Meng et al. (2021), there is a slight deviation in meshing stiffness due to this factor not being considered.

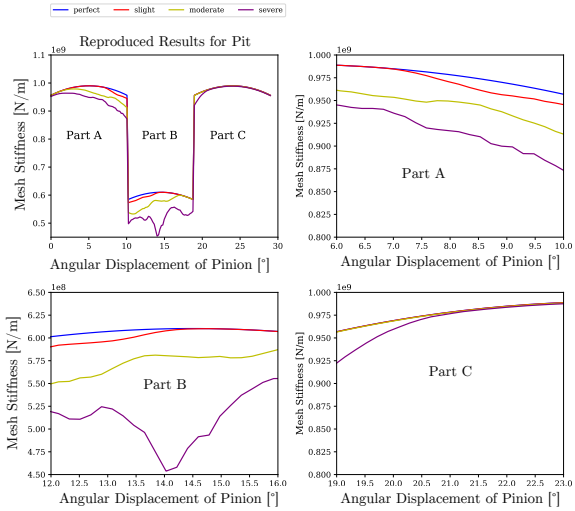
Furthermore, Meng et al. (2021) present a combination of predefined pit shapes, which is tedious to reproduce. Instead, the work presented here rather shows the Gaussian pit geometry sampling technique discussed in the pit

section. Thus, the shapes of the faults do not match exactly either, but the general tendency of the pits to cause stochastic reductions is demonstrated.

The different pit intensities introduced by Meng et al. (2021) and their effect on meshing stiffness may be seen in Figure 2.21(a). The stiffness results from the proposed generalised stiffness model are seen in Figure 2.21(b).



(a) Stiffness experiment results from Meng et al. (2021).



(b) Stiffness results for different pit combinations generated by the generalised stiffness model utilising the parameters as in Meng et al. (2021).

Figure 2.21. Comparison between literature and generated results for a pitting fault from the generalised stiffness model.

The figures showcase the capability of the generalised stiffness model to reproduce pits according to Meng et al. (2021). One may note that Meng et al. (2021) has slightly more rounded deviations, whereas the general stiffness model has slightly smoother reductions. This is because instead of predefining the pit shapes exactly, they are randomly sampled and distributed, leading to a smoother distribution of pit shapes and locations.

2.3 Gear dynamics modelling

The final step of generating the gearbox model is solving the dynamics of a well-defined system. A common method of solving the underlying dynamics of the system is to utilise a lumped parameter model (LPM). Such a model simplifies the physical system into groupings of masses, each with their own degrees of freedom. This gives rise to a multiple degrees of freedom (MDOF) system. These degrees of freedom are connected via springs and dampers, allowing for dynamic interactions in the MDOF system. Each DOF, therefore, has its own equation of motion (EOM) describing the movement of a given mass in a particular direction (the DOF).

Generally, these interactions between DOFs are coupled, which imply that the dynamics of each DOF cannot be solved independently, but rather a simultaneous solving approach is required. To solve the system of equations simultaneously, the EOMs are written in matrix form:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}(t)\mathbf{u}(t) = \mathbf{F}(t) \tag{2.95}$$

where $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ refer to the mass, damping and stiffness matrices of the system respectively. The acceleration, velocity and displacement vectors of the system are given as $\ddot{\mathbf{u}}(\mathbf{t})$, $\dot{\mathbf{u}}(\mathbf{t})$ and $\mathbf{u}(\mathbf{t})$ respectively. These vectors describe the response of the system. Finally, $\mathbf{F}(\mathbf{t})$ is the force vector, which describes any external forces or torques that add energy to the system.

This set of equations is solved using a solving algorithm such as the Runge-Kutta method or a Newmark algorithm. The latter is chosen in all work to follow. From Section 1.2.1.3 in the literature review, it was found that the most common single-stage gearbox models employed either a six DOF or eight DOF model. However, we identified the following challenges from the literature:

- Generally, the presented models do not have all the accompanying values to reproduce their results.
- A majority of works do not consider the effects of practical measurement of gearbox signals, i.e. not having direct access to the modelled DOFs. For example, a large body of work reports on the y accelerations of the pinion gear, but in practice, one will not be able to put an accelerometer on the gear directly. Instead, a more feasible approach would be to place the accelerometer on the gearbox casing. This casing acts as a new mass with its own damping and stiffness (through the bolts which connect the base of the gearbox to the gearbox casing) which needs to be accounted for.

The work of Luo et al. (2019*b*) presented sufficient values and considered the gearbox casing for vibration measurement. Therefore, the basis of the dynamic model for this work is directly based on the model from Luo et al. (2019*b*). The gear meshing stiffness and damping modelling approach used by these authors is different to the methods used in this work, and therefore deviations are expected in the shape and the amplitude of the system responses.

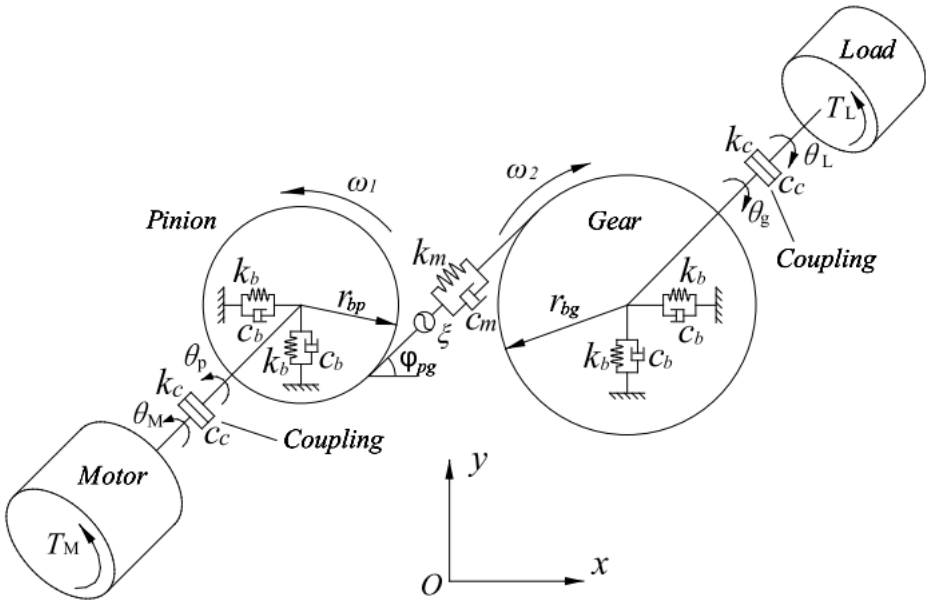
The parameters from the model of Luo et al. (2019*b*) were derived from experiments directly and therefore instils confidence in their use in this work. The model from Luo et al. (2019*b*) utilises 10 DOFs. Eight of these DOFs are the DOFs commonly used in literature and two are added to account for the casing x and y movement. Their model is presented in Figure 2.22.

A benefit of building a model with more DOFs than is commonly found in the literature (such as the model from Luo et al. (2019*b*)), is that it then becomes trivial to retain the DOFs required by simply removing selected DOFs such as the gearbox casing DOFs. This implies that other lower DOF literary works can be used for comparison if needed, by simply removing the required DOFs. In a sense, the spirit of this dynamic modelling section is to over-design the DOFs and remove DOFs for comparisons later. Before showing the complete model, it must be mentioned that although the model setup by Luo et al. (2019*b*) is used, their method of solving/grouping terms is different from what will be used in this work. The method utilised in this work is more similar to works from Chaari et al. (2012), with a few improvements. The LPM for this work may be seen in Figure 2.23.

The 10 DOF model may be lumped into five major mass/inertia groups. Firstly, the gearbox casing is given a mass, m_f. For the other four mass/inertia groups, a system of labelling similar to that of Chaari et al. (2012) is followed. Elements in this labelling system receive two subscripts. The first set of subscripts split the model into two main mass groups, namely group 1 (subscript 1), associated with input side elements and group 2 (subscript 2), associated with output side elements. The second set of subscripts indicate the relationship between each set of elements within a group. The motor is the first element in group 1, and therefore receives the second subscript as 1 ($I_{1,1}$). The pinion gear is the second element of group 1, and therefore receives the second subscript 2 ($I_{1,2}$). Similarly, the output gear and load are labelled as $I_{2,1}$ and $I_{2,2}$ respectively. The pinion gear and driven gear receive two mass/inertial values, as they represent two simultaneous groupings of mass. The mass values (m_1 and m_2) indicate a grouping of masses for the bearings, shafts and gears into a single mass for the given group. The inertia values ($I_{1,2}$ and $I_{2,1}$) indicate the rotational inertia of the gear for a given group. This group-based labelling system is further followed for the stiffnesses, damping values and degrees of freedom (x, y and θ). Finally, the two gears are coupled through a time-varying mesh stiffness ($k_t(t)$), which was modelled in section 2.1 and 2.2, and a time-varying damping ($c_t(t)$), which will be modelled shortly.

To model the time varying damping, however, it is first necessary to define matrices found in equation (2.95). Recalling that 10 DOFs need to be modelled, the displacement matrix of equation (2.95) is defined as:

$$\mathbf{u}(t) = \{x_1, y_1, \theta_{1,1}, \theta_{1,2}, x_2, y_2, \theta_{2,1}, \theta_{2,2}, x_f, y_f\} \tag{2.96}$$



(a) Dynamic model of gear mesh

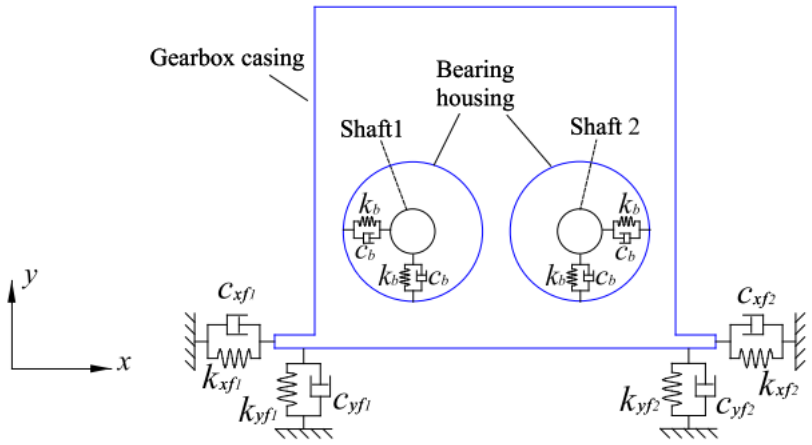


Figure 2.22. 10 DOF model used by Luo et al. (2019*b*).

where x_1, x_2, y_1 and y_2 refer to the pinion and gear x-direction movement and the pinion and gear y-direction movement respectively. $\theta_{1,1}$, $\theta_{2,2}$, $\theta_{1,2}$ and $\theta_{2,1}$ refer to the motor, load, pinion gear and driven gear rotational movements respectively. Finally, x_f and y_f refer to the gearbox casing x and y-direction movement. These quantities are clearly labelled in Figure 2.23.

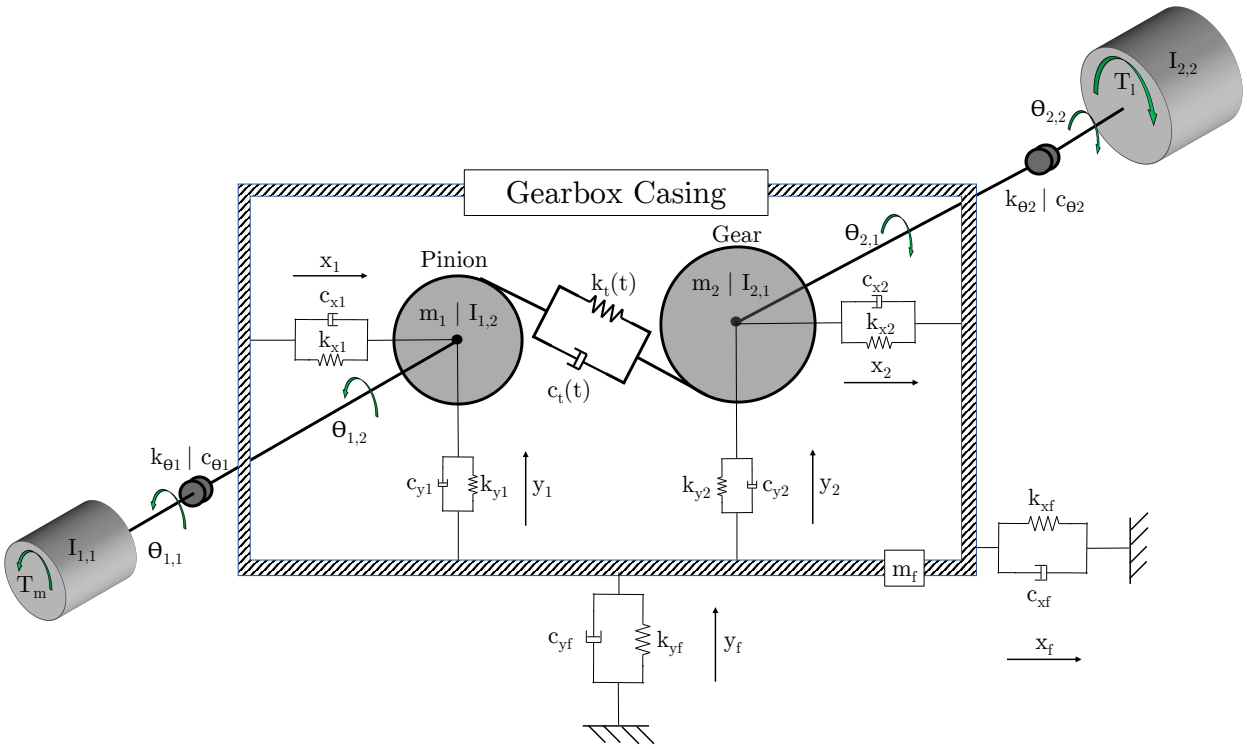


Figure 2.23. 10 DOF model used in this work. Adapted from Luo et al. (2019*b*).

The mass matrix in equation (2.95) may be constructed as:

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{1,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_{2,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{2,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_f & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_f \end{bmatrix} \tag{2.97}$$

The stiffness and damping matrices consist of constant and time-varying values. To account for this fact, it is useful to split these matrices into a steady and time-varying portion. That is to say:

$$\mathbf{K} = \mathbf{K_c} + \mathbf{K_v}(t) \tag{2.98}$$

$$\mathbf{C} = \mathbf{C_c} + \mathbf{C_v}(t) \tag{2.99}$$

where $\mathbf{K_c}$, $\mathbf{C_c}$, $\mathbf{K_v}(t)$, $\mathbf{C_v}(t)$ refer to the constant stiffness and damping matrix and the time-varying stiffness and damping matrices respectively.

The constant matrices may be set up as follows:

$$\mathbf{K_c} = \begin{bmatrix} k_{x1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_{x1} & 0 \\ 0 & k_{y1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_{y1} \\ 0 & 0 & k_{\theta 1} & -k_{\theta 1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -k_{\theta 1} & k_{\theta 1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{x2} & 0 & 0 & 0 & -k_{x2} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{y2} & 0 & 0 & 0 & -k_{y2} \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{\theta 2} & -k_{\theta 2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -k_{\theta 2} & k_{\theta 2} & 0 & 0 \\ -k_{x1} & 0 & 0 & 0 & -k_{x2} & 0 & 0 & 0 & k_{xf}+k_{x1}+k_{x2} & 0 \\ 0 & -k_{y1} & 0 & 0 & 0 & -k_{y2} & 0 & 0 & 0 & k_{yf}+k_{y1}+k_{y2} \end{bmatrix} \tag{2.100}$$

$$\mathbf{C_c} = \begin{bmatrix} c_{x1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c_{x1} & 0 \\ 0 & c_{y1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c_{y1} \\ 0 & 0 & c_{\theta 1} & -c_{\theta 1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -c_{\theta 1} & c_{\theta 1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{x2} & 0 & 0 & 0 & -c_{x2} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{y2} & 0 & 0 & 0 & -c_{y2} \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{\theta 2} & -c_{\theta 2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -c_{\theta 2} & c_{\theta 2} & 0 & 0 \\ -c_{x1} & 0 & 0 & 0 & -c_{x2} & 0 & 0 & 0 & c_{xf}+c_{x1}+c_{x2} & 0 \\ 0 & -c_{y1} & 0 & 0 & 0 & -c_{y2} & 0 & 0 & 0 & c_{yf}+c_{y1}+c_{y2} \end{bmatrix} \tag{2.101}$$

The time variant parts of the stiffness and damping arrays are solely dependent on the time-varying mesh stiffness (TVMS). Although it is said to be time-varying, in actual fact the meshing stiffness is inherently a function of the gear rotation angle, as seen in equation (2.47). The angle θ_1 is however a function of time as the gear rotates, and one may thus simply convert the meshing stiffness to a time-varying quantity by employing:

$$\theta_1(t) = \int_0^t \omega_s(t)dt \tag{2.102}$$

where $\omega_1(t)$ indicates the angular speed of the pinion shaft and t denotes the time.

The TVMS acts between two main clusters of mass, each with three DOFs and therefore acts upon six DOFs. To further complicate the matter, the TVMS acts as an inclined spring, due to the contact line between the gears being at an angle. Thus the coordinate system of the TVMS needs to be rotated to the global coordinate system so that it can be related to each DOF. It needs to be noted that various DOFs are coupled to the TVMS along the line of action and therefore the net displacement along the line of action is a function of multiple DOFs.

It is useful to describe this displacement along the line of action ($\delta(t)$) using a projection array:

$$\delta(t) = \mathbf{u}(t) \cdot \mathbf{P}^T \tag{2.103}$$

where $\mathbf{P}$ defines the projection array needed to convert a DOF to its equivalent along the line of action. For the 10 DOF model, this is defined as:

$$\mathbf{P} = \{\sin\alpha_0, \cos\alpha_0, 0, R_{b1}, -\sin\alpha_0, -\cos\alpha_0, -R_{b2}, 0, 0, 0\} \tag{2.104}$$

Note that $\mathbf{P} \in \mathbb{R}^{1\times 10}$. A useful property of this formulation is that the time-varying stiffness matrix may be set up by multiplying the displacement along the line of action with the correct projection term to convert it back to the relevant DOF. The projection term is found directly in the projection matrix. The time-varying stiffness matrix is therefore defined as:

$$\mathbf{K_v}(t) = k_t \cdot \delta(t) \cdot \mathbf{P} \tag{2.105}$$

$$\mathbf{K_v}(t) = k_t \cdot \mathbf{u}(t) \cdot \mathbf{P^T P} \tag{2.106}$$

where k_t is the TVMS found in equation (2.52). The formulation in equation (2.105) showcases what is happening mathematically. The displacement $\delta(t)$ is projected to the relevant DOF through multiplication with **P**, giving a displacement quantity. The displacement is then multiplied by the TVMS to give a force, as is required in a LPM modelling environment. It is however easier to use the second form, where the projection matrices multiply to form a matrix quantity which may be added to the constant stiffness matrix.

To model the time-varying mesh damping (TVMD) is a more complicated task, as there are multiple damping factors to consider. This is beyond the scope of this work. Rather, a commonly used approach from the literature will be employed. This approach models the TVMD as being proportional to the TVMS. Some approaches in literature simply model the damping ratio as a constant value which is derived as a function of the average gear mesh stiffness. Other authors, such as Tian (2004) do not make such a limiting assumption and develop a time-varying damping term that is directly proportional to the meshing stiffness. This latter method is employed in this work and is implemented as follows:

Firstly, the mean damping value between gear teeth is determined. This is found by rearranging the damping ratio formulation and by substituting the effective mass of the formulation with that of the pinion and bearing blocks such as in Tian (2004). This yields:

$$c_m = 2\zeta\sqrt{k_m\frac{m_1 m_2}{m_1+m_2}} \tag{2.107}$$

where k_m and c_m denote the mean mesh stiffness and damping. ζ defines the damping ratio of the system, and m_1 and m_2 define pinion and driven side mass blocks.

A common value for ζ in the literature is 0.07 and is therefore used in this work. For a practical setup, this value may be experimentally determined. Having obtained a constant damping value, most authors stop here. Tian (2004), however, continued by using this average mesh damping value in relation to the average mesh stiffness value to find a scaling constant, leading to the following formulation:

$$c_t(t) = \mu k_t(t) \tag{2.108}$$

where $k_t(t)$ and $c_t(t)$ refer to the time-varying mesh and damping stiffness and μ is found from:

$$\mu = \frac{c_m}{k_m} \tag{2.109}$$

where c_m is defined above and k_m is found by taking the average stiffness value over a meshing cycle.

With the complexity of stiffness and damping being shown, the final term from equation (2.95) to be determined is the force array. Only two forces are acting on the model, namely the input torque from the motor and the load resistance torque. Therefore, one simply needs to add these torques to the model as follows:

$$\mathbf{F}(t) = \begin{bmatrix}0 & 0 & T_m & 0 & 0 & 0 & 0 & T_l & 0 & 0\end{bmatrix} \tag{2.110}$$

where T_m and T_l refer to the motor input torque and the load resistance torque, respectively. Note that it is only necessary to know the one to model the other, due to the geared relationship resulting in:

$$T_l = -\frac{Z_2}{Z_1}T_m \tag{2.111}$$

A negative sign is added behind the load resistance torque to be consistent with the direction of the rotational DOF of the driven side. The load resistance torque acts opposite this direction and therefore must receive a negative value.

The relationship given in equation (2.111) is only valid at steady-state, where the motor speed has reached a steady value. Before this time (run-up), the motor will supply a higher torque (at low speed) than the load is consuming, leading to a net force on the system, which in turn makes the motor spin faster, reducing its torque output until this value reaches the load resistance torque value. This effect may be modelled into the system by creating a speed-dependent motor torque, as is seen in Khabou et al. (2011). However, for this work, it is not necessary to model these dynamics, and we are simply interested in the steady-state regime. Therefore, upon solving the dynamics of the system with the Newmark algorithm ($\beta_1 = 0.5$ and $\beta_2 = 2.25$ for the form in

Zienkiewicz (2013)), the shaft speed and motor torque are simultaneously fixed in the iteration loop. This leads to unrealistic dynamic responses for the first few iterations, but then quickly converges to the desired steady-state space.

At this point, the physical gearbox model is complete and synthetic data may now be generated by solving the dynamics as outlined above..

2.4 Synthetic signal generation

The hybrid approach proposed in this work relies on labelled synthetic data from the gearbox model. Therefore, this section is purely aimed at describing and generating synthetic datasets for the hybrid model. Furthermore, this section simultaneously sets out to verify the synthetic data being generated, to ensure the dynamic models being used are trustworthy.

Before data is generated, it is important to outline the current shortcomings from the literature, as these will be the main points of investigation for this work. Therefore, we have identified two main investigations for which we need to generate data:

1. Many works in literature that utilise physical models do not consider practical measurement complexities. These works focus on generating fault diagnosis techniques for the pinion gear vibration. However, in practice one is not able to directly mount an accelerometer on the pinion gear. It is more likely that accelerometers are mounted to the gearbox casing. Therefore, a dataset will be generated that can compare the vibrations directly on the pinion gear to the vibrations from the gearbox casing. The dynamic model from Luo et al. (2019*b*) includes both of these vibration locations as DOFs and will therefore be utilised. Therefore, this dataset will allow for the effect of measurement location on condition monitoring predictions to be investigated.
2. When designing a physical model, one will generally find deviations between the modelled signal and the true asset. This is expected, as the physical model is a low DOF estimate of a much higher DOF reality. Techniques that are developed on the synthetic data may not work for the real data. Therefore, a dataset will be generated that allows one to compare the results from two identical physical models, with the only difference being the number of DOFs used to model the system dynamics. For this dataset, the six DOF model from Meng et al. (2020) will be used as the lower DOF model. The same model will be increased to eight DOFs, with motor and load DOFs being introduced. Therefore, this dataset will allow for the effect of model fidelity on condition monitoring predictions to be investigated.

In Chapter 2.3, the 10 DOF dynamic model of Luo et al. (2019*b*) was given. It was stated that starting with a 10 DOF dynamic model allows the author to easily step down to a lower DOF model if necessary. For the first dataset, relating to measurement locations, the developed 10 DOF model will be used and the parameters from Luo et al. (2019*b*) are directly implemented.

For the second dataset, the same 10 DOF dynamic model is used, with the necessary DOFs removed to reach an eight or six DOF model. However, for these two models, the values from Meng et al. (2020) will be used as opposed to those from Luo et al. (2019*b*). A different model's values is utilised in this case because it allows for a second independent source from literature against which the current dynamic model may be verified. Note that the model from Meng et al. (2020) is a six DOF model, and therefore the eight DOF model will utilise the values from Meng et al. (2020) for six of the DOFs and utilise values from Luo et al. (2019*b*) for the missing two DOFs.

The following sections are divided according to the dataset being created. The first dataset is called Dataset A and is split into two sub-datasets. These sub-datasets are split according to the measurement location, namely Dataset A-pinion for horizontal pinion gear vibrations (x_1) and Dataset A-casing for horizontal gearbox casing vibrations (x_f). The second dataset is called Dataset B and is also split into two sub-datasets. These sub-datasets are split according to the number of DOFs utilised for dynamic modelling, namely Dataset B-6 for the six DOF

model and Dataset B-8 for the eight DOF model. For both DOF models, the pinion vertical vibrations will be captured (y_1).

For each dataset, four fault types, namely cracks, chips, spalls and pits will be modelled and simulated at 20 different intensities each. The details to these faults will be given shortly. A summary of the datasets that are generated is given in Figure 2.24.

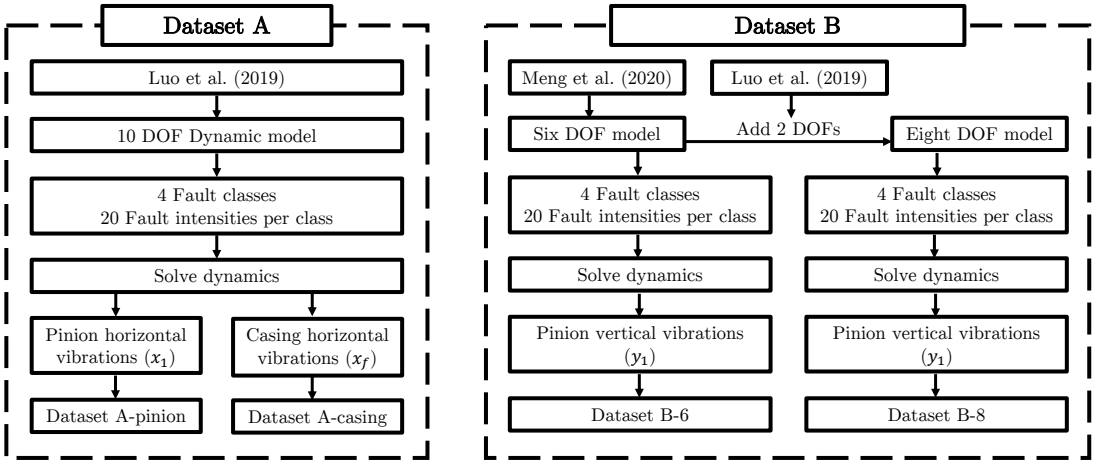


Figure 2.24. Illustration of the models and works used for the dataset creation.

At this stage it becomes useful to define how data is generated and compiled into a dataset. By utilising the developed generalised stiffness model, one may choose a fault type, intensity and location on the pinion gear and run a dynamic simulation. Depending on how long the simulation is run, varying lengths of signals will result. For Dataset A and B, a 0.4s simulation is run with a simulation frequency of $f_s = 200$kHz. A single simulation result may be seen in Figure 2.25.

A gear crack fault of 50% intensity is chosen on the fifth pinion tooth for illustrative purposes. The initial part of the simulation is transient (due to estimated initial conditions) and therefore only the steady-state portion of the signal must be used. This steady signal is called a simulation s_i. Further, note the rotational position of the input shaft (not the pinion gear) being shown. This information is a simple by-product of the simulated dataset and is shown as an aid to visualise the rotation of the input shaft. In practice, this quantity may be measured by a tachometer, or by utilising tacholess methods. The input shaft rotation is given as opposed to the pinion gear rotation, as in practice, the tachometer will more likely be placed nearer to the input shaft, rather than near the gear itself. The reason for this is that access to the pinion gear may be limited or impossible. It is therefore expected that a tachometer will more likely be placed outside of the gearbox itself, near the motor, where tachometers may more easily be mounted.

The steady-state signal is split into sub-signals, each representing a single pinion gear rotation. This is done by observing when the rotational value of the input shaft passes 0, indicating that a new rotation has begun. The rotational position of the shaft follows a steady trend, because of the fixed motor speed. Sample s_i is now split into j portions, each indicating a single rotation from the pinion gear by the method described above. This results in a set of signals, each labelled by $s_{i,j}$. Note that each sub-signal may not be of the same length and therefore a resampling of the sub-signal is enforced. In this work, a resampling per full gear rotation is enforced such that the length of any rotation $s_{i,j}$ is 4096 points.

Finally, the resampled signals may be standardised. We may standardise a signal $s_{i,j}$ as follows:

$$s_{n|i,j} = \frac{s_{i,j} - \bar{s_r}}{\sigma_{s_r}} \tag{2.112}$$

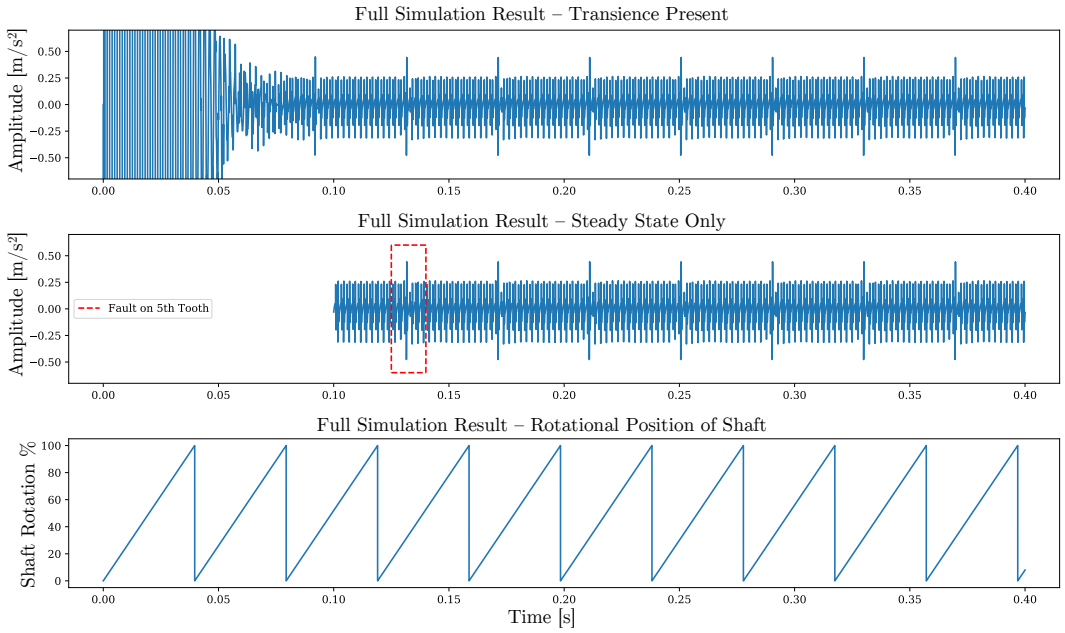


Figure 2.25. Illustration of a single generated synthetic signal. The top portion of the figure shows one full 0.4s simulation. Note the transients for roughly the first 100ms. Due to this, only the steady portion is extracted, seen in the middle figure. The bottom figure indicates the rotation % of the pinion shaft. This signal may be used between zero-crossings to split the full vibration signal into synchronised sub-signals.

where $s_{n|i,j}$ refers to the standardised signal for the jth rotation of the ith simulation sample. $\bar{s_r}$ and σ_{s_r} refer to the mean and standard deviation of the reference signal, respectively. The reference signal should be chosen as the signal from a healthy case, where no faults are present. This allows all samples to be standardised relative to the same reference signal. The healthy signal is chosen, as it is expected that during the operation of a gearbox, the first data which is captured from a gearbox is from a healthy state. All future vibrations should then be normalised relative to that initial healthy state. Should the initial data captured from the gearbox contain a fault, this may still be used as the reference signal, as long as all future samples up to maintenance are standardised relative to the same reference signal. To complete the process, each standardised rotation sample, $s_{n|i,j}$ is labelled and stored in a dataset.

With the necessary steps outlined, we may define our dynamic models' parameters in their respective sub-sections. After the parameters are given, the faults for each dynamic model are given and Dataset A and B are created. Finally, the resulting datasets are analysed and preliminary observations from the above-mentioned datasets are given.

2.4.1 10 DOF dynamic model

This chapter utilises the 10 DOF model from Luo et al. (2019*b*) and specifically focuses on extracting the horizontal vibrations from the pinion gear (x_1) and gearbox casing (x_f). As stated in the previous chapter, the parameters for the 10 DOF model from Luo et al. (2019*b*) were mostly determined experimentally and are therefore at least representative of a real gearbox. The goal of this section is to show the reproduced results and give the parameters required to produce such results. The results may then be compared to the direct results from Luo et al. (2019*b*). It is however reasonable to expect that these results will not yield the same vibration shape or amplitudes. The reasons behind this statement are as follows:

The method employed to model the meshing stiffness and damping between gear teeth is different from this work and the work in Luo et al. (2019*b*). In their work, a large amount of attention is given to damping mechanisms whereas this work simply employs a proportional damping mechanism. Therefore, due to the difference in implementation for the meshing stiffness and damping, a vibration shape and amplitude difference is expected.

Furthermore, the dynamic formulation of Luo et al. (2019*b*) accounts for inter-tooth friction, which this work does not. Again, this will likely affect the shape and amplitude of simulated vibrations. Table 2.3 shows the parameters employed in this work.

Table 2.3. Necessary Parameters to Create Dynamic Simulation for 10 DOF model from Luo et al. (2019*b*).

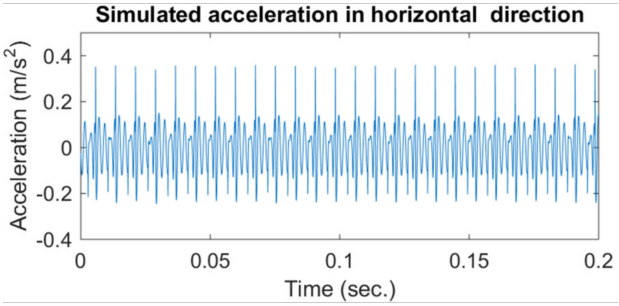
Symbol	Description	Value	Units
Z_1	Number of pinion gear teeth	16	-
Z_2	Number of driven gear teeth	48	-
m_1, m_2	Tooth module	3.175	mm
L_1, L_2	Tooth width	16	mm
α_0	Pressure angle	20	$^\circ$
E	Young's modulus	200	GPa
ν	Poisson's ratio	0.3	-
R_h	Hub radius[1]	9.76	mm
m_1	Mass of pinion block	1.272	kg
m_2	Mass of gear block	3.5367	kg
m_f	Mass of gearbox casing	18.509	kg
$I_{1,1}$	Mass moment of inertia of motor	0.016107	kgm^2
$I_{1,2}$	Mass moment of inertia of pinion block	0.0001751	kgm^2
$I_{2,1}$	Mass moment of inertia of gear block	0.006828	kgm^2
$I_{2,2}$	Mass moment of inertia of load	0.005153	kgm^2
T_m	Motor torque	$7 \times \frac{16}{48}$	Nm
T_l	Load torque	7	Nm
$k_{x1}, k_{x2}, k_{y1}, k_{y2}$	Bearing stiffnesses	8.5364×10^7	N/m
k_c	Coupling stiffness	330	Nm/rad
k_{xf}	Gearbox casing screw x stiffness	1.9912×10^8	N/m
k_{yf}	Gearbox casing screw y stiffness	2.036×10^8	N/m
$c_{x1}, c_{x2}, c_{y1}, c_{y2}$	Bearing damping	2.134×10^4	Ns/m
c_c	Coupling damping	23.1	Nm rad/s
c_{xf}	Gearbox casing screw x damping	1995.56	Ns/m
c_{yf}	Gearbox casing screw y damping	2005.80	Ns/m

With the parameters all defined, the TVMS is generated with the aid of the generalised stiffness model. Then, with the appropriate fault type, shape and location decided, they too are incorporated into the stiffness model. Finally, the dynamics of the system are solved using the Newmark algorithm (at a solution frequency of 200 kHz), which samples the correct stiffness value from the pre-generated TVMS (with faults already factored in) as a function of pinion gear rotation angle. As mentioned in Chapter 2.3, both the pinion motor speed as well as input and output torques are fixed during solving to represent steady-state conditions. Two such conditions are analysed, namely a $f_{p1} = 8.1$Hz and $f_{p2} = 25.2$Hz case[2], to correspond with the work from Luo et al. (2019*b*). Figure 2.26 shows the comparison of horizontal gearbox casing vibrations (x_f) for f_{p1} and Figure 2.27 for f_{p2}.

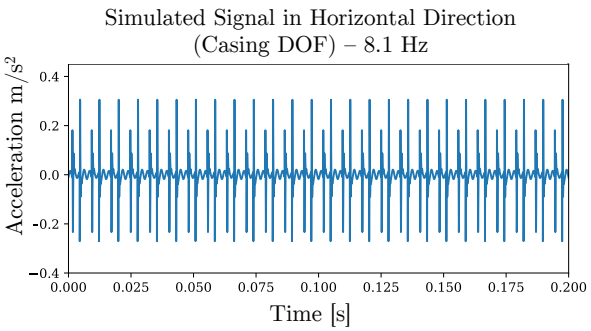
It may be seen that as expected, the signals differ in shape. This is expected as the damping model in this work is much simpler than that of Luo et al. (2019*b*). However, the amplitudes of vibration are mostly correct, showcasing that the decision to neglect friction and use a simplified damping model, did not significantly affect the amplitude of vibrations. Furthermore, key sharp vibration peaks and troughs are also seen to be similar between the developed model and Luo et al. (2019*b*). This is important as these sharp vibrations are caused

[1]This quantity is not reported in the literature. Instead, it was adjusted (within realistic bounds) to such an extent that the resulting stiffness graph has the same maximum amplitude as that from the literature.

[2]Recall that a sampling frequency of 200kHz was used to simulate the graphs. This implies that any resampled frequencies above the Nyquist frequency of 100kHz could alias without a filter. Therefore, a quick sanity check is appropriate to ensure the resampled results are not aliased. Given a frequency of rotation of the pinion gear (f_p) of 25.2 Hz, one may infer that the resampled frequency is $f_{rs} = 4096 \times f_p = 103$ kHz, which is slightly over the Nyquist limit. This implies that frequencies between 100kHz and 103kHz will alias into the 0 to 3 kHz bands without a filter. However, the amplitude of the frequencies near 100kHz was seen to be negligibly small, and therefore a filter was not implemented.

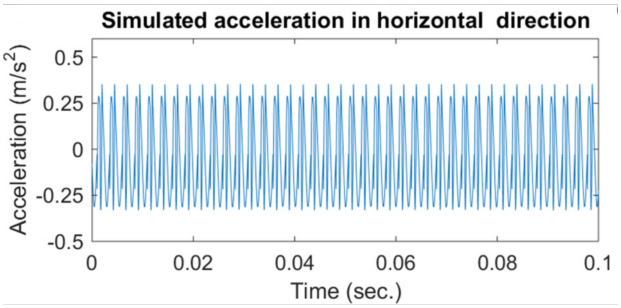


(a) Simulated acceleration signal from the works of Luo et al. (2019*b*) for $f_{p1} = 8.1$Hz.

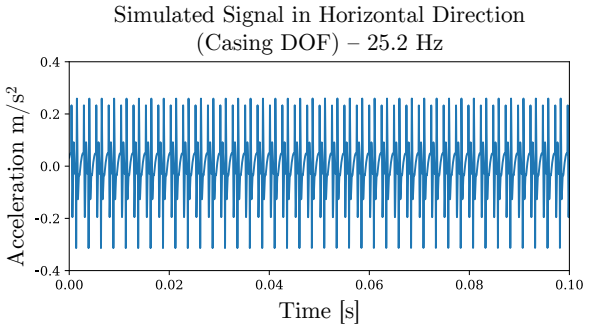


(b) Recreated acceleration signal from the works of Luo et al. (2019*b*) for $f_{p1} = 8.1$Hz.

Figure 2.26. Comparison between literature and generated results for the simulated acceleration signal from the works of Luo et al. (2019*b*) for $f_{p1} = 8.1$Hz.



(a) Simulated acceleration signal from the works of Luo et al. (2019*b*) for $f_{p2} = 25.2$Hz.



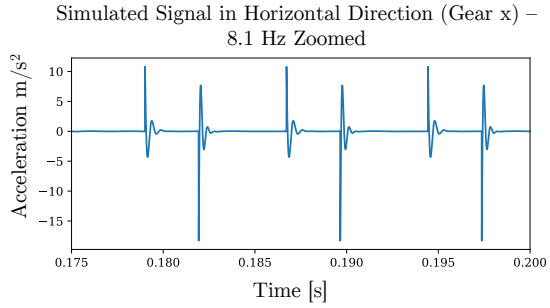
(b) Recreated acceleration signal from the works of Luo et al. (2019*b*) for $f_{p2} = 25.2$Hz.

Figure 2.27. Comparison between literature and generated results for the simulated acceleration signal from the works of Luo et al. (2019*b*) for $f_{p2} = 25.2$Hz.

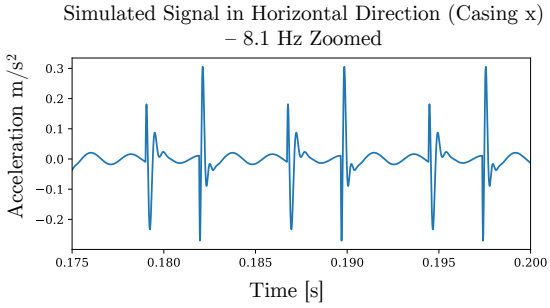
by the sudden discontinuity in gear meshing stiffness during single to double tooth transition (or vice versa), showing that the TVMS is likely modelled correctly.

With the dynamic model developed, we may do a preliminary comparison between the two DOFs which are extracted (x_1 and x_f). The case for f_{p1} is run and may be seen in Figure 2.28. The vibration amplitude on the gearbox casing horizontal direction is much smaller than its pinion gear counterpart. This important factor showcases that although a vibration model may be developed for gear fault detection, it may be crucially important to model the vibration path all the way to the measurement location, as it is seen that not only are amplitudes different, but also vibration shape. Thus, it may not be sufficient to infer the gearbox condition purely based on gear vibrations.

This concludes the dynamic modelling of the 10 DOF model from the works of Luo et al. (2019*b*). It was shown that although the vibration shapes may not perfectly match, other important qualities such as amplitude and impulsive behaviour were successfully captured.



(a) Simulated acceleration signal for $f_{p1} = 8.1$Hz. Vibrations extracted are for the x-direction on the pinion gear.



(b) Simulated acceleration signal for $f_{p1} = 8.1$Hz. Vibrations extracted are for the x-direction on the pinion gear.

Figure 2.28. Comparison between simulated gear and casing vibrations for the model based on the works in Luo et al. (2019*b*). Notice the differences in shape and amplitude.

2.4.2 Six and eight DOF dynamic models

As stated earlier, a six and eight DOF model will also be developed. The six and eight DOF models are based on the 10 DOF model developed in Chapter 2.3, with the only difference being that the eight DOF system does not incorporate the gearbox body (2 DOFs) and the six DOF model further does not incorporate the motor and load.

Meng et al. (2020) proposed a six DOF model that is directly used to set up the parameters for the reduced six DOF model here. This work presents another independent opportunity to verify all work done thus far, should the resulting dynamics match the results from Meng et al. (2020).

There is much more similarity between the factors which are considered between the work of Meng et al. (2020) and the work here. The only missing factor from this work is the consideration of inter-tooth friction. Otherwise, all other modelling aspects seem similar. It is therefore expected that the reproduction of the results from Meng et al. (2020) must be closely matched if the proposed generalised model is correct.

Although the six DOF model has no motor and load component, one may still apply the motor and load torques in the dynamic analysis. This is done by adding these forces directly on the pinion and driven gear directly (i.e. removing the coupling stiffness and damping from the eight DOF model). As stated before, for all models, both the motor and load torque will be fixed as well as the motor rotational speed. For the six DOF model, the pinion gear speed will thus be fixed.

For the eight DOF model, two additional DOFs are required. The additional DOFs come from the motor and load. The two DOFs are implemented from the 10 DOF model from Luo et al. (2019*b*). However, the values for the motor and load inertias are taken from Chaari et al. (2012) and the coupling values between motor and pinion and the driven gear and load from Mohammed et al. (2015). For both the six and eight DOF dynamic models, only a single rotational frequency is considered, namely $f_p = 40$Hz. However, three different load cases are considered, namely a 10, 20 and 30 Nm load case. The necessary parameters to recreate the dynamics of either the six DOF or eight DOF model are given in Table 2.4.

[3]This quantity is not reported in the literature. Instead, it was adjusted (within realistic bounds) to such an extent that the resulting stiffness graph has the same maximum amplitude as that from the literature.
[4]The motor inertia was taken from Chaari et al. (2012).
[5]The load inertia was taken from Chaari et al. (2012).
[6]The coupling stiffness value was taken from Mohammed et al. (2015).
[7]The coupling damping value was taken from Mohammed et al. (2015).

Table 2.4. Necessary Parameters to Create Dynamic Simulation for eight DOF model (and six DOF model) from Meng et al. (2020). Note that three different load and motor torque values are used, indicated by the values separated between the vertical bars.

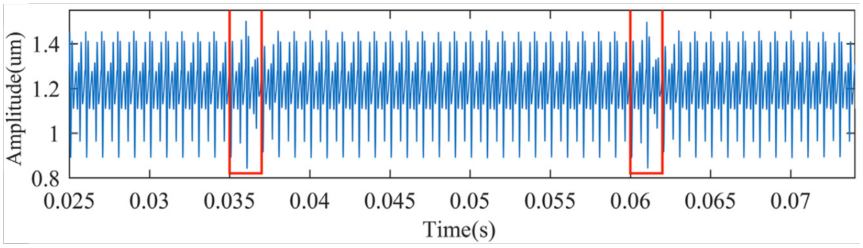
Symbol	Description	Value	Units
Z_1	Number of pinion gear teeth	25	-
Z_2	Number of driven gear teeth	30	-
m_1, m_2	Tooth module	2	mm
L_1, L_2	Tooth width	20	mm
α_0	Pressure angle	20	$^\circ$
E	Young's modulus	200	GPa
ν	Poisson's ratio	0.3	-
R_h	Hub radius[3]	9.76	mm
m_1	Mass of pinion block	0.3083	kg
m_2	Mass of gear block	0.4439	kg
$I_{1,1}$	Mass moment of inertia of motor[4]	0.0043	kgm^2
$I_{1,2}$	Mass moment of inertia of pinion block	0.96×10^{-4}	kgm^2
$I_{2,1}$	Mass moment of inertia of gear block	2×10^{-4}	kgm^2
$I_{2,2}$	Mass moment of inertia of load[5]	0.0045	kgm^2
T_m	Motor torque	$(10\|20\|30) \times \frac{25}{30}$	Nm
T_l	Load torque	10| 20| 30	Nm
$k_{x1}, k_{x2}, k_{y1}, k_{y2}$	Bearing stiffnesses	6.56×10^8	N/m
k_c	Coupling stiffness[6]	1×10^4	Nm/rad
$c_{x1}, c_{x2}, c_{y1}, c_{y2}$	Bearing damping	1.8×10^3	Ns/m
c_c	Coupling damping[7]	10	Nm rad/s

With all the parameters defined, the TVMS is generated. Then, with the necessary fault type, shape and location decided, they too are incorporated into the stiffness model. Finally, the dynamics of the system are solved using the Newmark algorithm (at a solution frequency of 200 kHz), which samples the correct stiffness value from the pre-generated TVMS (with faults already factored in) as a function of pinion gear rotation angle. The resulting signal may be found in Figure 2.29(b) with the equivalent signal from the works of Meng et al. (2020) in Figure 2.29(a).

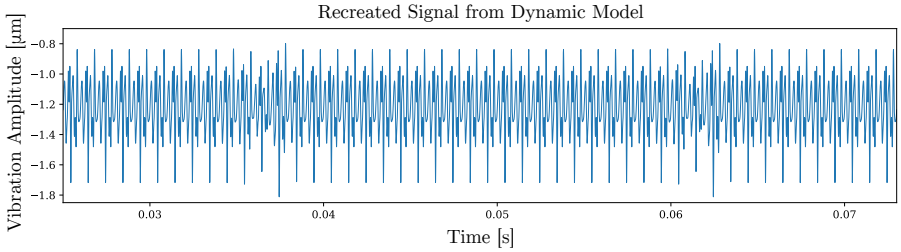
As expected, the two vibration signals correspond well in shape and amplitude. It is also clear that the generated signal differs in vibration mean, but this may be attributed to a potential difference in the choice of positive DOF directions between the generated signals and the works of Meng et al. (2020). To further test the model, a spalling defect was introduced to match the signals in the red boxes. It was found that the dynamic model managed to capture these dynamics well, as the slight drop and then gain in amplitude in the fault region is reproduced in this work.

This reproduction should further corroborate the validity of the developed model and at this point, the verification is deemed sufficient to trust the model at hand. The advantage of having two dynamic models based on the same parameters is that we may now compare the effects of different numbers of DOFs. Therefore, in Figure 2.30(a) the acceleration signal of the pinion vertical direction (y_1) for the six DOF model is given and compared to its eight DOF counterpart in Figure 2.30(b).

It is quite clear that with the increase in DOF, the signal tends to be more damped. This makes sense as two additional DOFs extract energy from the system through damping. This is important, as it shows that indicators that might traditionally be used on a developed model with a certain number of DOFs cannot always account for the amount of damping expected in practice and may become invalid. From a frequency perspective, the two DOF models seem to match each other in vibration shape, therefore one may expect that frequency-based features may perform better at extracting universal features.

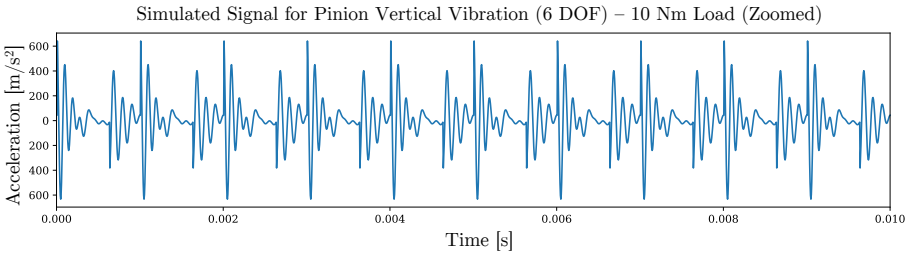


(a) Displacement signal from Meng et al. (2020). Vibrations shown are for the y-direction on the pinion gear.

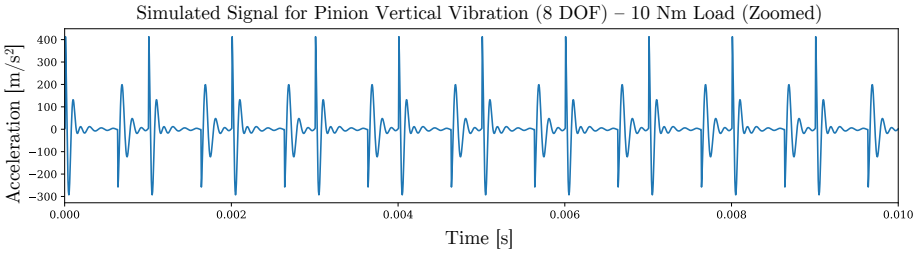


(b) Simulated displacement signal for $T_l = 20$Nm. Vibrations shown are for the y-direction on the pinion gear.

Figure 2.29. Comparison between the work of Meng et al. (2020) and the developed dynamic model.



(a) Simulated acceleration signal for $T_l = 20$Nm. Vibrations extracted are for the y-direction on the pinion gear on the six DOF model.



(b) Simulated acceleration signal for $T_l = 20$Nm. Vibrations extracted are for the y-direction on the pinion gear on the eight DOF model.

Figure 2.30. Visual comparison between the same acceleration measurements for the same load torque based on the work of Meng et al. (2020). The only difference between the two is the number of DOFs.

These questions need not be addressed here and will be given attention in a later chapter. This concludes the description of the two dynamic models that were created from the same base code. The results seem to indicate that the developed models are performing correctly from a numerical standpoint and may be trusted to do numerical analyses. We may now utilise the generalised stiffness model to create various faults, and in turn solve the dynamics to create Dataset A and Dataset B.

2.4.3 Introduction of faults

Before training a data-driven model, one needs to create a meaningful dataset. Therefore, this section sets out to describe the different faults that were simulated for the 10 DOF model from Luo et al. (2019*b*) and the eight and six DOF model from Meng et al. (2020).

Firstly, one needs to decide on the physical quantities to be used when designing the faults. These mainly depend on the size of the gear tooth. Most of the fault parameters are defined relative to some tooth property and therefore fine-tuning these values is not too difficult. The following sub-sections outline the exact fault parameters used to generate Dataset A and B.

2.4.3.1 Dataset A - Fault parameters and analysis for dynamic model A-pinion and A-casing

For Dataset A, four fault types are considered, namely a crack, chip, spall and pit. Each fault type has its own parameters and is given in Table 2.5.

Table 2.5. Parameters used to generate 20 fault cases for each fault type for Dataset A. The minimum value indicates a healthy gear tooth state and a maximum value the worst fault intensity for a given choice of parameters.

Crack Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Left Side Crack Depth	q_0	0	5	mm
Right Side Crack Depth	q_2	0	5	mm
Crack Angle	ν	90	90	$^\circ$
Crack Width	W_c	16	16	mm
Chip Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Chip Depth	b	1	1	mm
Chip Width	c	0	16	mm
Spall Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Spall Length	l_s	1	1	mm
Spall Width	w_s	0	15	mm
Spall Depth	h_s	1	1	mm
Spall Start Location	x_1	4	4	mm
Pit Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Mean Pitting Line	u	4.387	4.387	mm
Standard Pitting Deviation	σ_u	296	296	μm
Sampled Pit Location	u_i	$\mathcal{N}(u,\sigma_u)$		mm
Sampled Pit Angular Size	θ_i	$\mathcal{U}(30,30)$		$^\circ$
Sampled Pit Radius	r_i	$\mathcal{U}(80,80)$		μm
Number of Samples	N.A.	0	2000	-

Note that when a value in Table 2.5 changes from a minimum to a maximum, this is implemented in 20 steps, all linearly interpolated. The minimum value refers to the fully healthy tooth (and thus minimum fault intensity) and the maximum value to the highest damaged tooth case (and thus the maximum considered fault intensity). Therefore, the resulting dataset contains 20 fault intensities for each fault type.

To minimise external variability during data analysis, the faults are made as simple as possible with only a single variable ever being changed. Thus, it may be seen that for cracks, a constant width crack is assumed ($q_0 = q_2$), with a constant crack angle and width. Therefore, the depth of crack is the only variable being considered. In this case, the maximum crack depth is the same as the thickness of the tooth and therefore indicates a fully broken tooth. The visualisations of the crack case are found in Figure 2.31 (Top Left). For the chip case, the chip is assumed to have a constant depth and thus the only varying quantity is its width across the tooth. This may be seen in Figure 2.31 (Top Right). Spalls are assumed to have constant depth, length and starting location, with only the spall width across the tooth being varied. This is seen in Figure 2.31 (Bottom Left).

Finally, the pits are also simplified by fixing the pit radius, angle and possible mean location. Thus, the only variation is the number of pits on the tooth. Due to the variability when drawing pit samples, a special approach is followed to build up the pit dataset: Firstly, 2000 samples are drawn at random. Then, to build up different fault intensities, the first n pit samples are always taken in the same order from the list of 2000 pits. For example, for fault intensity 5, the first 500 pits are taken from the 2000 pit samples. For fault intensity 15, the first 1500 pit samples are taken from the 2000 pit samples. This method of sampling ensures that larger pit intensities always contain the previous intensity's pits and some new pits. This approach aims to make the pit evolution as realistic as possible. The case where all 2000 pits are present is seen in Figure 2.31 (Bottom Right).

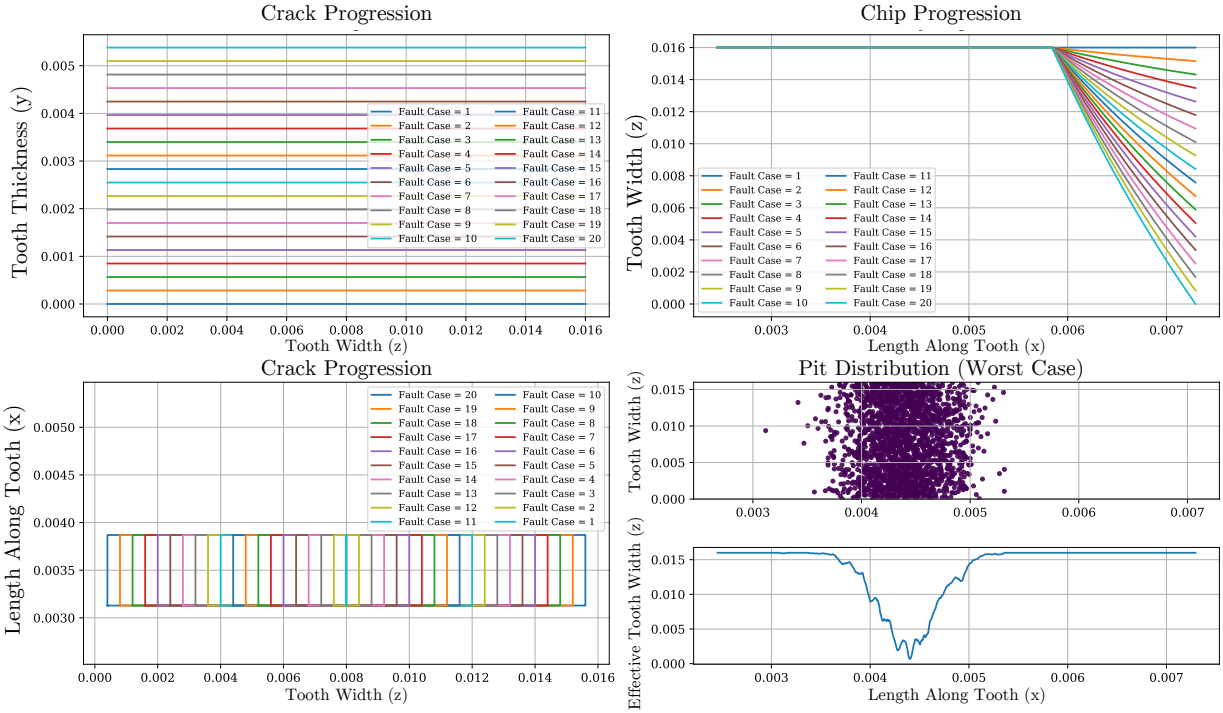


Figure 2.31. Visualisation of fault progressions in Dataset A for the four considered fault types, namely cracks, chips, spalls and pits.

It is useful to gain a measure for the relative fault intensity of each of these faults. For example, a full-width chip does not reduce the stiffness of a tooth nearly as much as a very deep crack. Therefore, a relative fault intensity measure is proposed. To calculate this measure, it is important to decide in which plane a fault predominantly acts. For example, a crack acts in the plane going through the girth of the tooth, whereas chips, pits and spalls mainly act on the face plane of the tooth. With this considered, the relative fault area (RFA) may be defined as:

$$RFA = \frac{A_{fault}}{A_{plane}} \tag{2.113}$$

where A_{fault} indicates the area in the relevant plane the fault is taking up and A_{plane} indicates the full area of the chosen plane. For example, a full-width crack that goes halfway through the tooth will have half the area of the full plane and thus yield an RFA value of 0.5. A broken tooth is thus an RFA of 1 and a perfectly healthy tooth a value of 0. This metric is purely used as a quick and easy measure of relative fault contribution when looking at the vibration signals.

When doing these calculations for each fault type and intensity, Figure 2.32 results. One sees that a simple

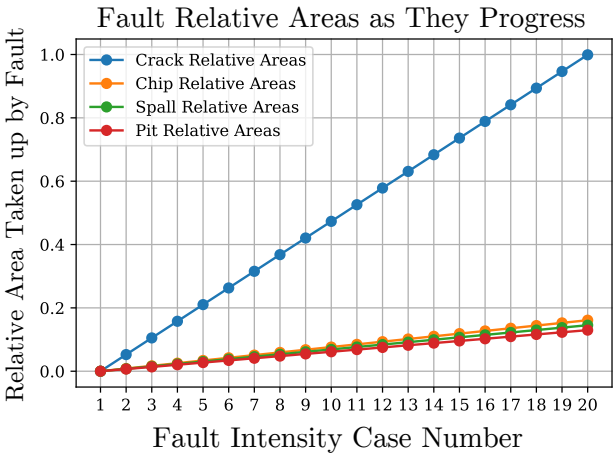


Figure 2.32. Relative fault areas for the four fault types in Dataset A.

progression of crack faults is much more impactful in terms of RFA. One therefore expects larger stiffness effects to arise due to cracks than any of the other simulated faults. This further implies that it is much easier to pick up for example a 50% propagated crack than a 50% propagated spall (within this numerical context). This is to say that the dataset will be slightly imbalanced in terms of relative fault intensities, in that the crack faults are more significant than their counterparts. This is not a disadvantage, however, as it adds a meaningful point of diversity in the dataset which may be investigated in future experiments.

One may finally take the given fault values, substitute them into the generalised stiffness formulation and determine the resulting TVMS. It is arbitrarily decided to induce the faults on the 5th gear tooth. Figure 2.33 shows the effects for all 20 fault intensities for all four fault types.

One can see that cracks have a much larger effect on the TVMS compared to other fault types. This is due to a crack's mechanism which renders any material above the crack as 'missing', making that portion of the tooth contribute nothing to the overall meshing stiffness. One can also see for a maximum crack case, the TVMS drops to 0 during the single tooth meshing region, which is exactly what would be expected from a broken tooth - no stiffness contribution. Spalls present quite a discontinuous drop in TVMS amplitude and are a direct result of the modelling choice made to use rectangular cut-outs. If a more modern approach were utilised, the spalls would reduce the TVMS in a much smoother manner. The pits seem to present irregular stiffness deviations, which are due to the random sampling nature of this fault mechanism. In reality, this would be the case too, so it is not unrealistic. Finally, chips present smooth reductions in TVMS but are very small reductions, bringing into question how easily this fault type may be identified.

Finally, with the faults modelled and incorporated into the TVMS, one may solve the dynamic system using the Newmark algorithm. The solving frequency is set at 200 kHz and the solver is set to run up to 0.4s. The initial conditions are set to zero displacement, velocity and acceleration for all DOFs, except for the input shaft, which is set to a fixed rotational speed according to the speed case being simulated. Furthermore, the system torques are fixed according to the appropriate load for the speed case. Due to the zeroing of initial conditions, the system

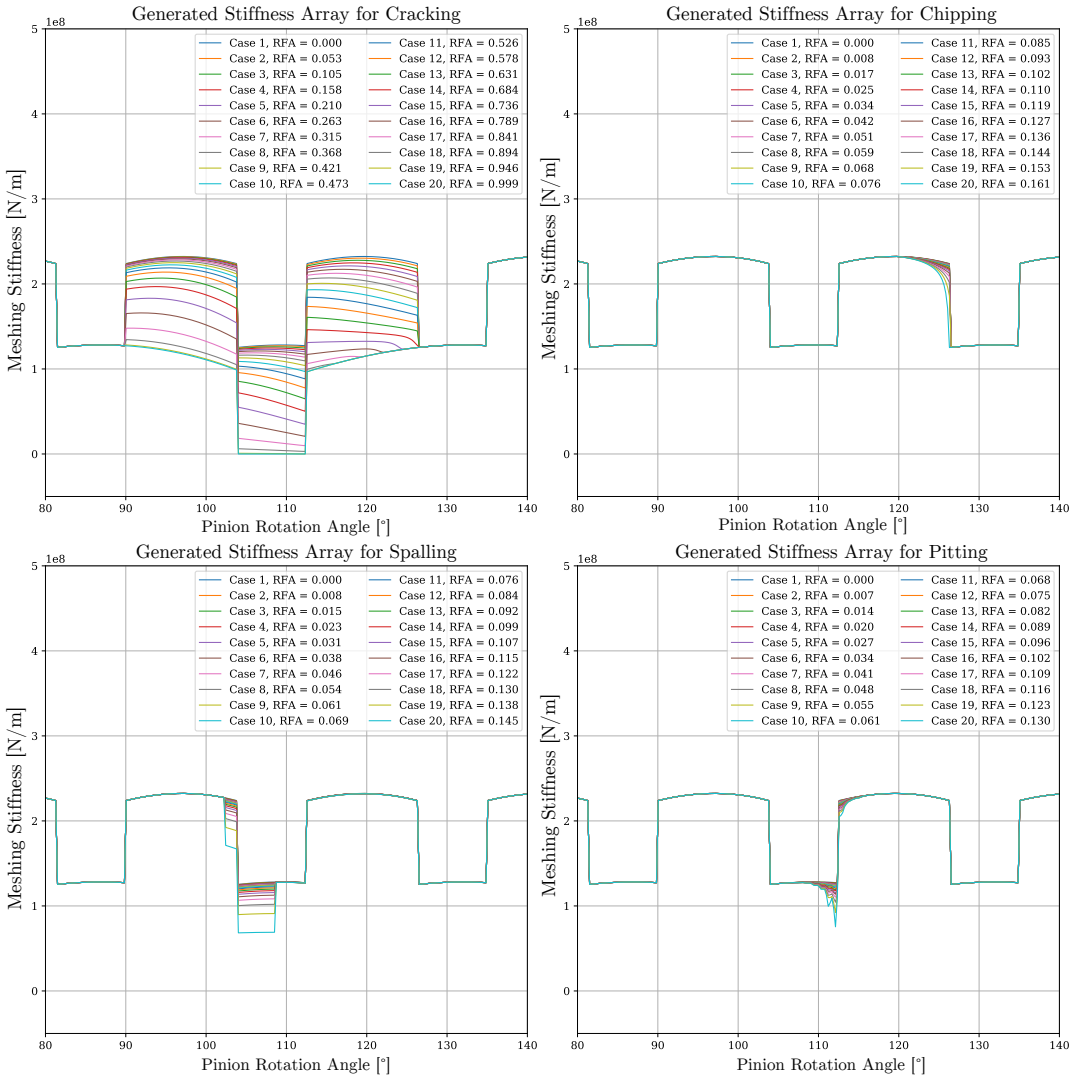
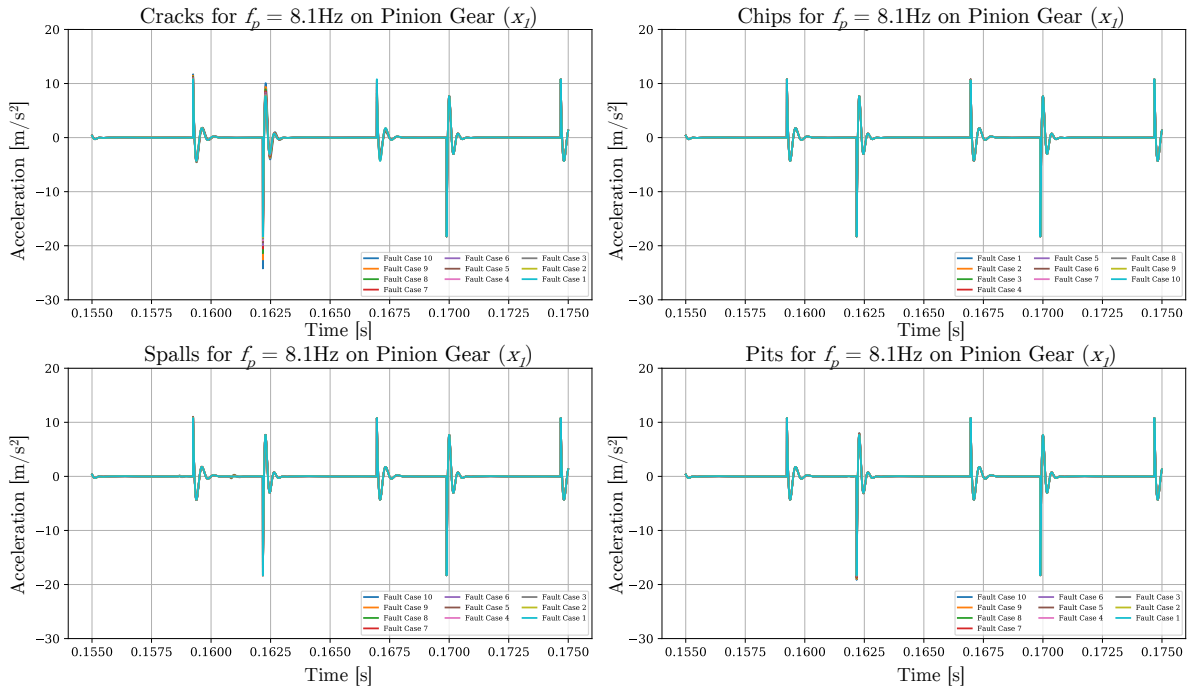


Figure 2.33. Effective TVMS for the four considered fault types in Dataset A.

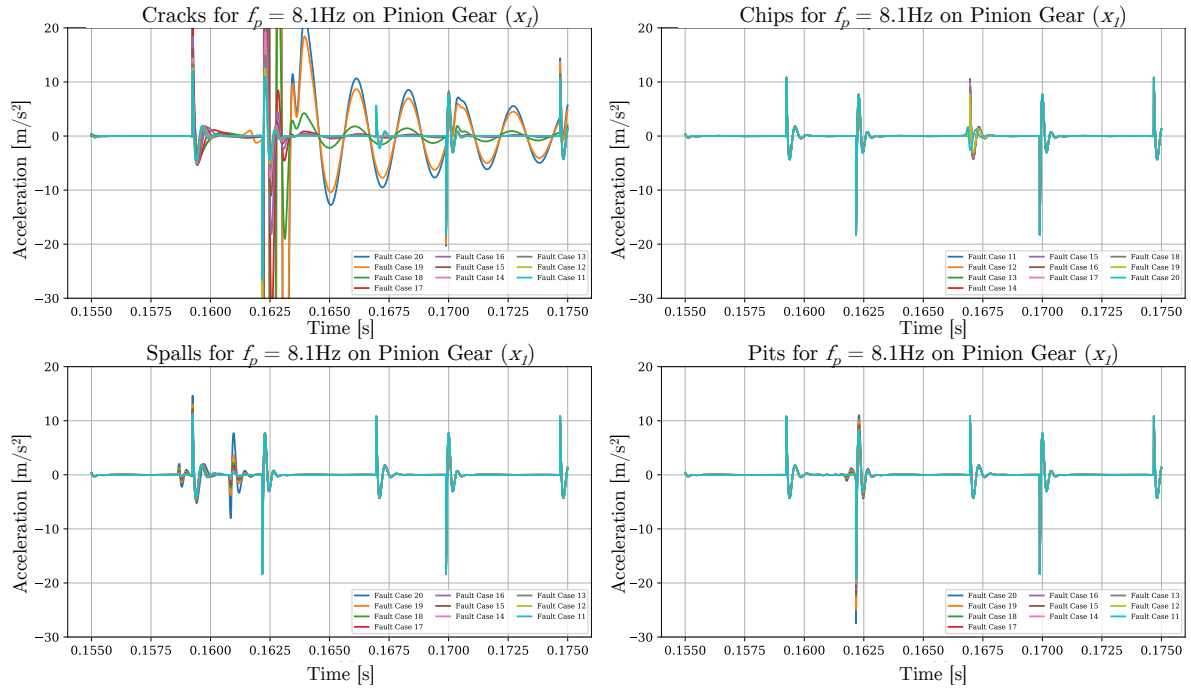
takes a few iterations to reach steady-state. The steady-state portion is visually determined and extracted, which is then used to investigate the resulting time and frequency domain signals. For the time domain signals, 20 fault intensities prove difficult to plot. Therefore, Figure 2.34(a) shows the first 10 fault intensities for each fault case and Figure 2.34(b) the last 10 for the pinion gear x-direction. Similarly, Figure 2.35(a) shows the first 10 fault intensities for each fault case and Figure 2.35(b) the last 10 for the gearbox casing x-direction.

Moving on to the frequency domain, we note that with 16 teeth rotating at $f_{p1} = 8.1$ Hz, we expect the gear mesh frequency and its harmonics to show up at $n \times 129.6$ Hz intervals. Furthermore, given the fact that the simulations are run for steady-state, a successful dynamic model should still render amplitude and frequency modulations around these harmonics to showcase gear faults, similar to what is seen in Chaari et al. (2008). The results from taking the frequency spectra for the pinion gear and gearbox casing x-directions are shown in Figure 2.36(a) and Figure 2.36(b) respectively. Note that a zoomed-in version is given, as for a solving frequency of 200 kHz, frequency bands are available up to 100 kHz, which is unnecessary.

Although the case for f_{p1} was given in the figures, the case for f_{p2} was also simulated and was used as the basis for Dataset A. Therefore, Dataset A-pinion and Dataset A-casing are comprised of the f_{p2} case. The higher

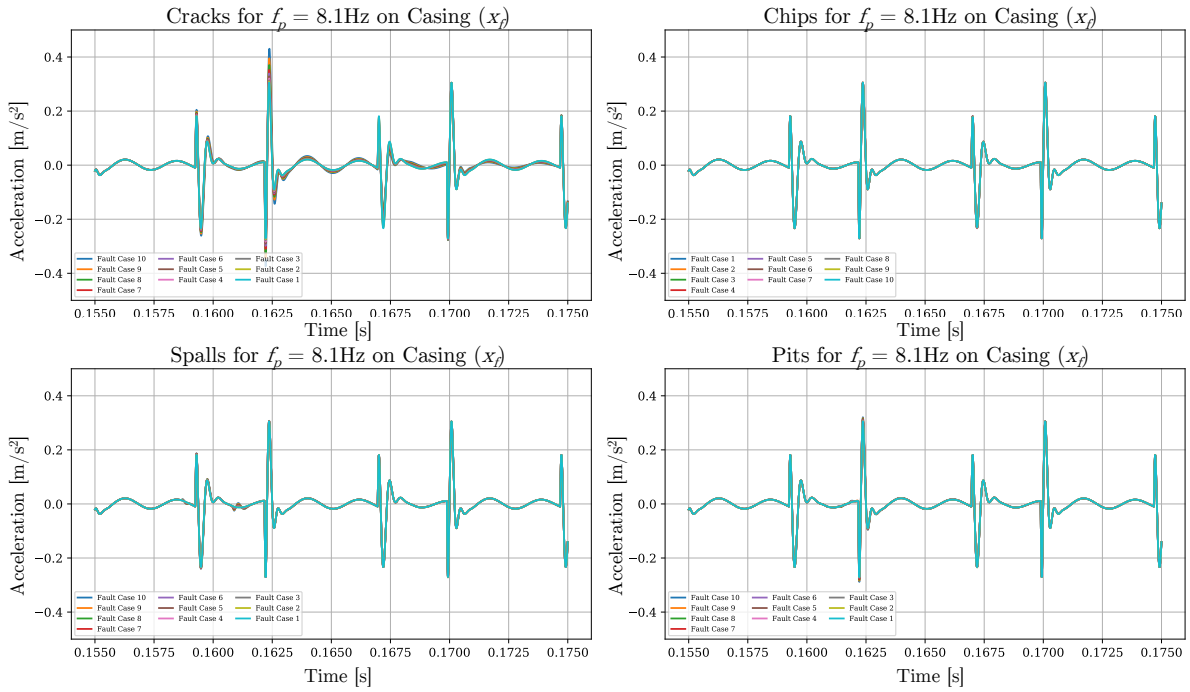


(a) Resulting Vibration signals on the pinion gear x-direction for the first 10 fault intensities.

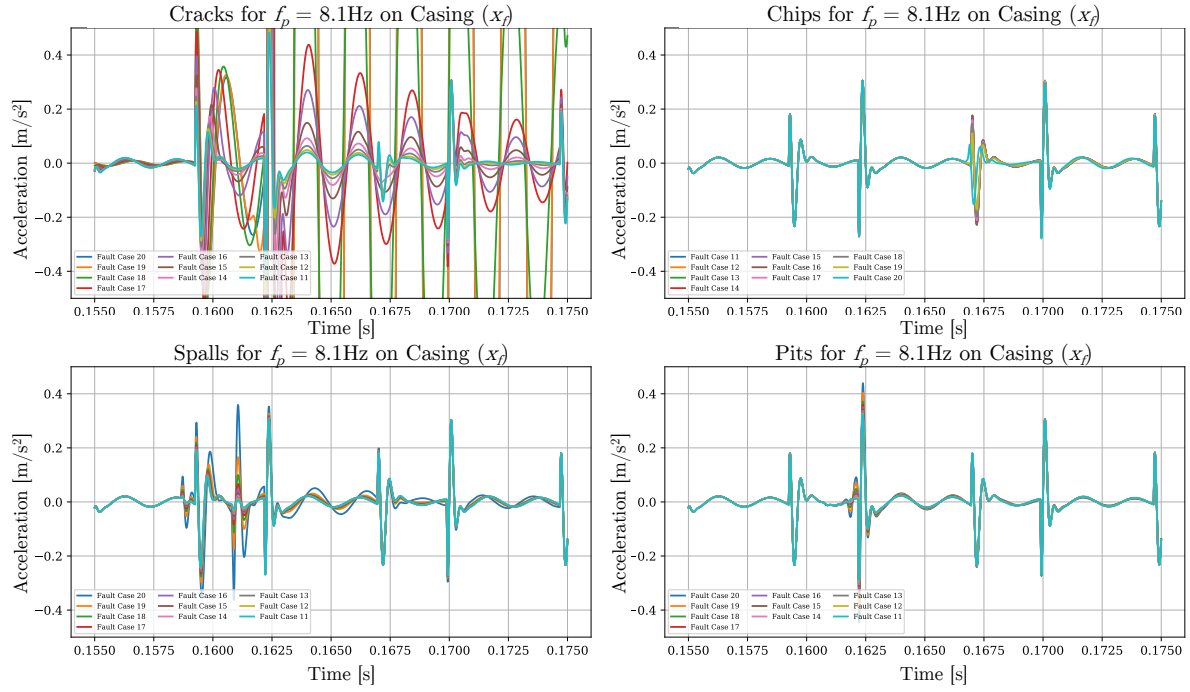


(b) Resulting Vibration signals on the pinion gear x-direction for the last 10 fault intensities.

Figure 2.34. Vibration signals generated for various fault intensities over various fault types for the pinion gear x vibrations in Dataset A.

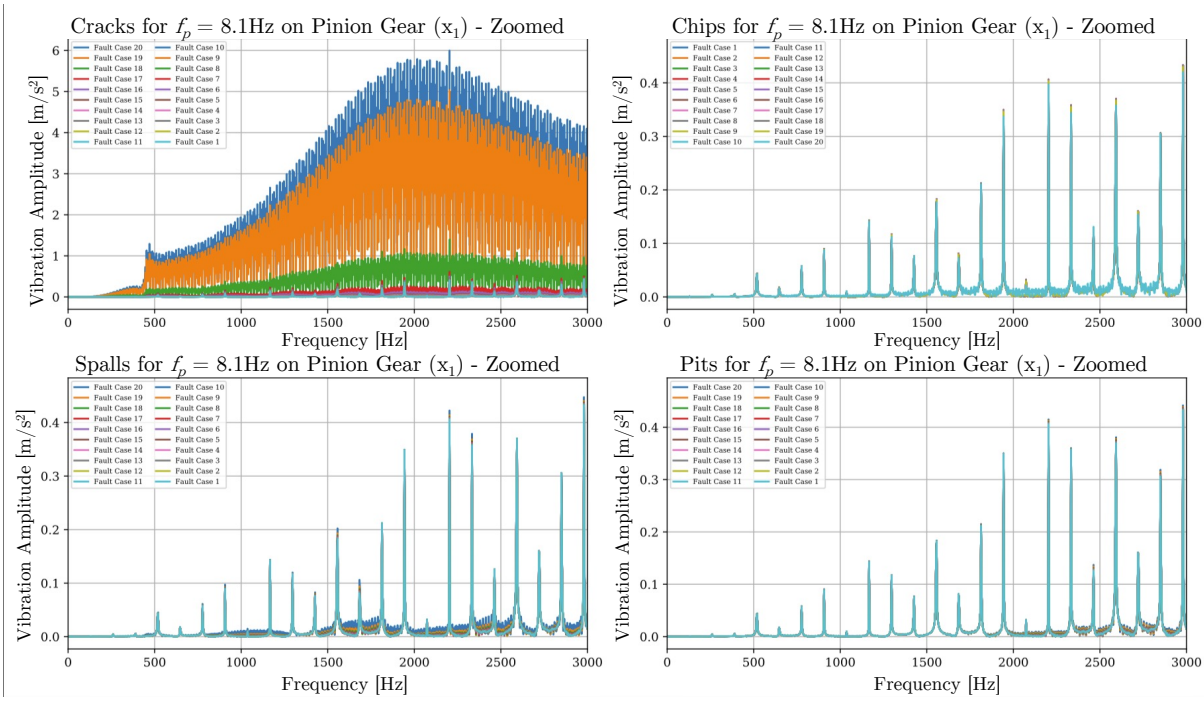


(a) Resulting Vibration signals on the gearbox casing x-direction for the first 10 fault intensities.

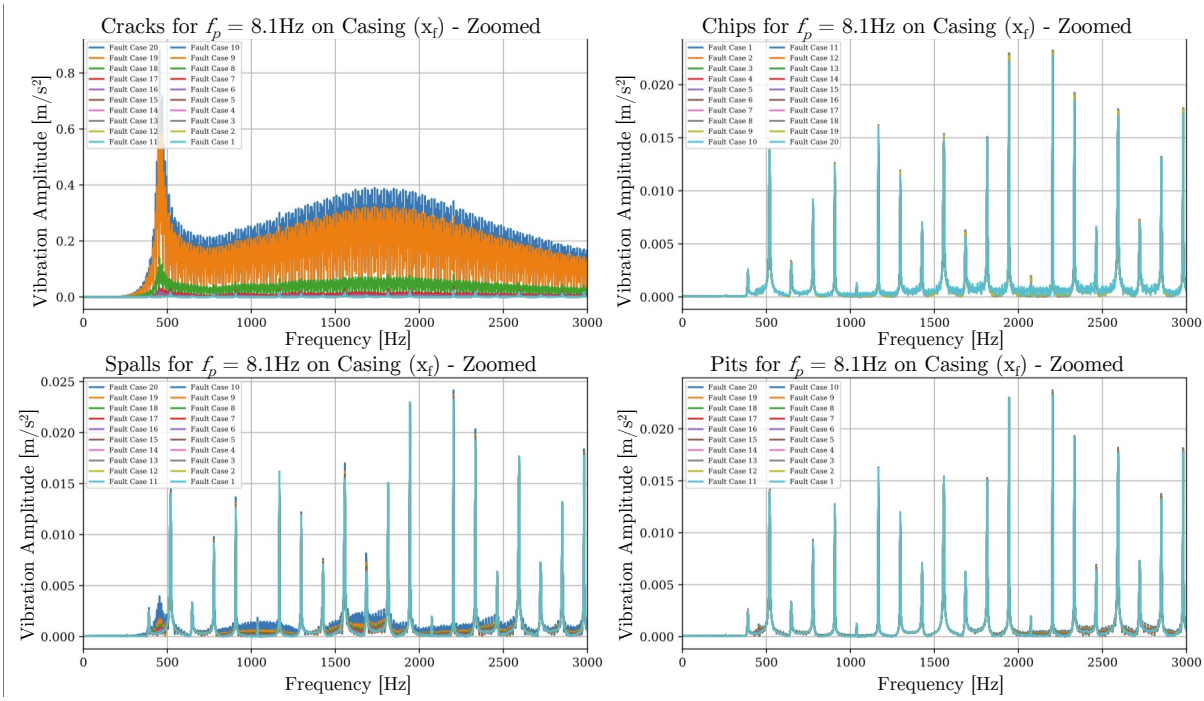


(b) Resulting Vibration signals on the gearbox casing x-direction for the last 10 fault intensities.

Figure 2.35. Vibration signals generated for various fault intensities over various fault types for the gearbox casing x-direction in Dataset A.



(a) Resulting FFT on the pinion gear *x*-direction for all fault intensities



(b) Resulting FFT on the gearbox casing *x*-direction for all fault intensities

Figure 2.36. Frequency domain spectra for various fault intensities over various fault types for Dataset A.

speed case (f_{p2}) is arbitrarily chosen for Dataset A. However it may prove advantageous to use f_{p2} over f_{p1} for future investigations where time-synchronous averaging is used, as the f_{p2} simulation contains more revolutions for the given simulation time.

2.4.3.2 Dataset B - Fault parameters and analysis for dynamic model B-6 and B-8

For Dataset B, an identical process to the previous section is followed. Therefore, only the key information and resulting graphs is shown here. The fault parameters for the six and eight DOF models differ from the 10 DOF model, as the values for the six and eight DOF models are based on Meng et al. (2020). Therefore, a new set of parameters are required. Furthermore, only the 20 Nm load case is shown here, although all three load cases are simulated. The parameters for creating the faults may be found in Table 2.6.

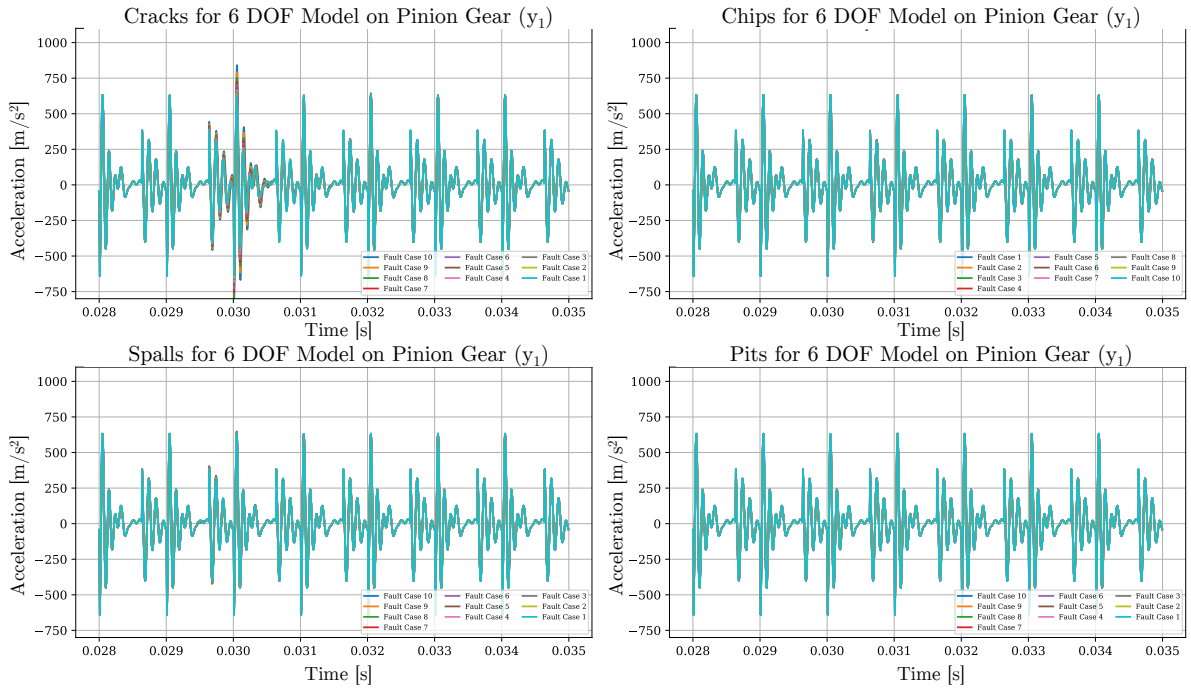
Table 2.6. Parameters used to generate 20 fault cases for each fault type for Dataset B. The minimum value indicates a healthy gear tooth state and a maximum value the worst fault intensity for a given choice of parameters.

Crack Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Left Side Crack Depth	q_0	0	3.644	mm
Right Side Crack Depth	q_2	0	3.644	mm
Crack Angle	ν	90	90	$^\circ$
Crack Width	W_c	20	20	mm
Chip Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Chip Depth	b	1.07	1.07	mm
Chip Width	c	0	20	mm
Spall Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Spall Length	l_s	526	526	μm
Spall Width	w_s	0	19	mm
Spall Depth	h_s	1	1	mm
Spall Start Location	x_1	2.374	2.374	mm
Pit Parameters				
Quantity Name	**Symbol in Report**	**Min. Value**	**Max. Value**	**Unit**
Mean Pitting Line	$\bar{u}$	2.5	2.5	mm
Standard Pitting Deviation	σ_u	210.4	210.4	μm
Sampled Pit Location	u_i	$\mathcal{N}(\bar{u},\sigma_u)$		mm
Sampled Pit Angular Size	θ_i	$\mathcal{U}(30,30)$		$^\circ$
Sampled Pit Radius	r_i	$\mathcal{U}(80,80)$		μm
Number of Samples	N.A.	0	2000	-

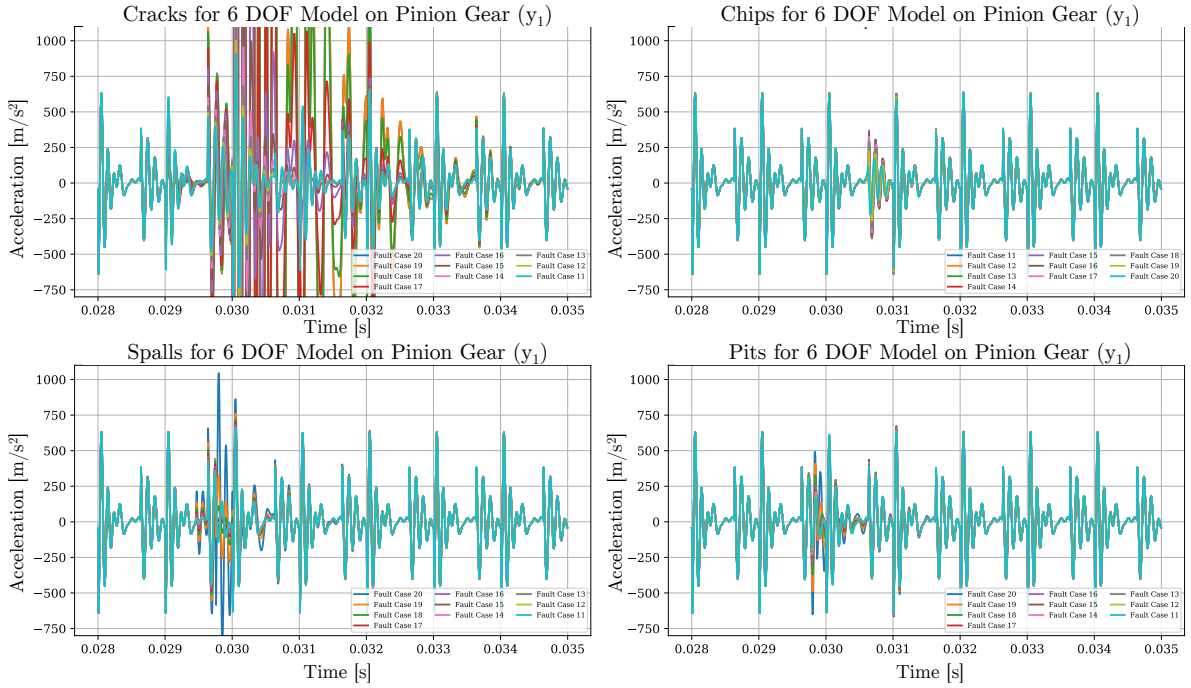
These parameters are substituted into the generalised stiffness model and the resulting TVMS is determined. An arbitrary choice is made to induce the fault on the fifth gear tooth. The stiffness reductions look similar to those in Figure 2.33 and are thus not shown here. The final task is to once again show the resulting time and frequency responses of the dynamic model.

Once again, 20 fault intensities prove difficult to plot. Therefore, Figure 2.37(a) shows the first 10 fault intensities for each fault case and Figure 2.37(b) the last 10 for the six DOF system pinion y-direction. Similarly, Figure 2.38(a) shows the first 10 fault intensities for each fault case and Figure 2.38(b) the last 10 for the eight DOF pinion gear y-direction.

Moving on to the frequency domain, we note that with 25 teeth rotating at $f_p = 40$ Hz, we would expect the gear mesh frequency and its harmonics to show up at 1000 Hz intervals. As mentioned before, since the simulations were run at steady-state, a successful dynamic model should render amplitude and frequency modulations around these harmonics to showcase gear faults. The results from taking the frequency spectra for the pinion gear y-directions for the six and eight DOF models are shown in Figure 2.39(a) and Figure 2.39(b) respectively. A zoomed version is given as the frequency bins continue to 100 kHz.

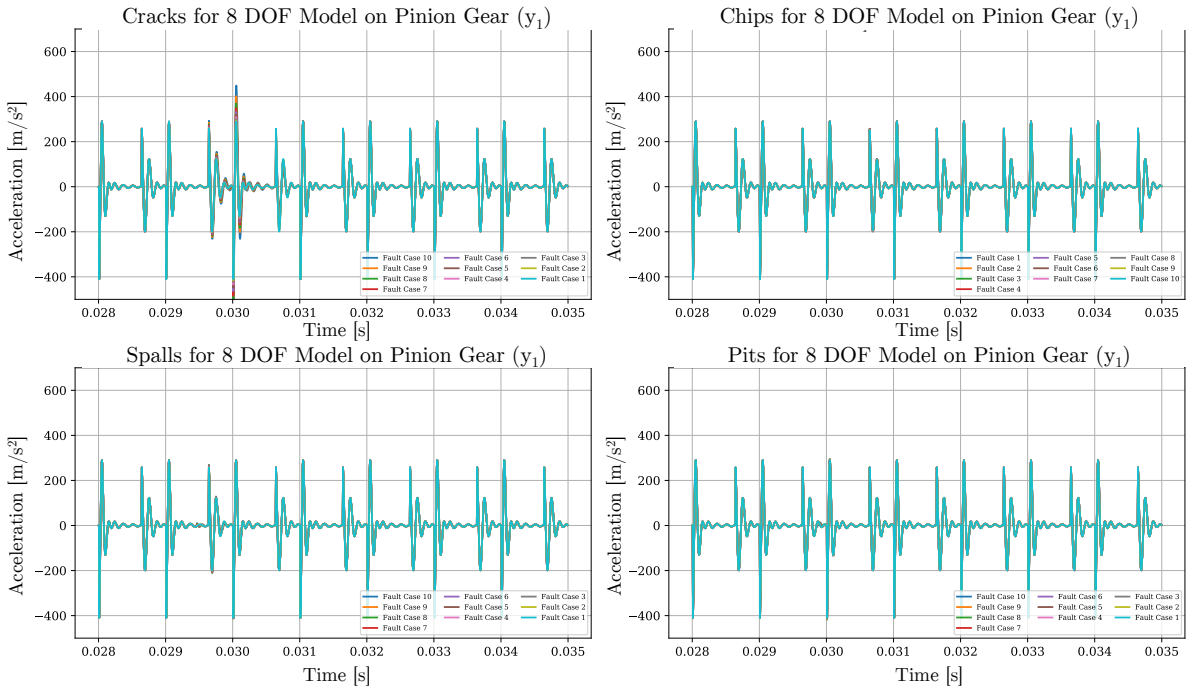


(a) Resulting Vibration signals on the pinion gear *y*-direction for the first 10 fault intensities of the six DOF model.

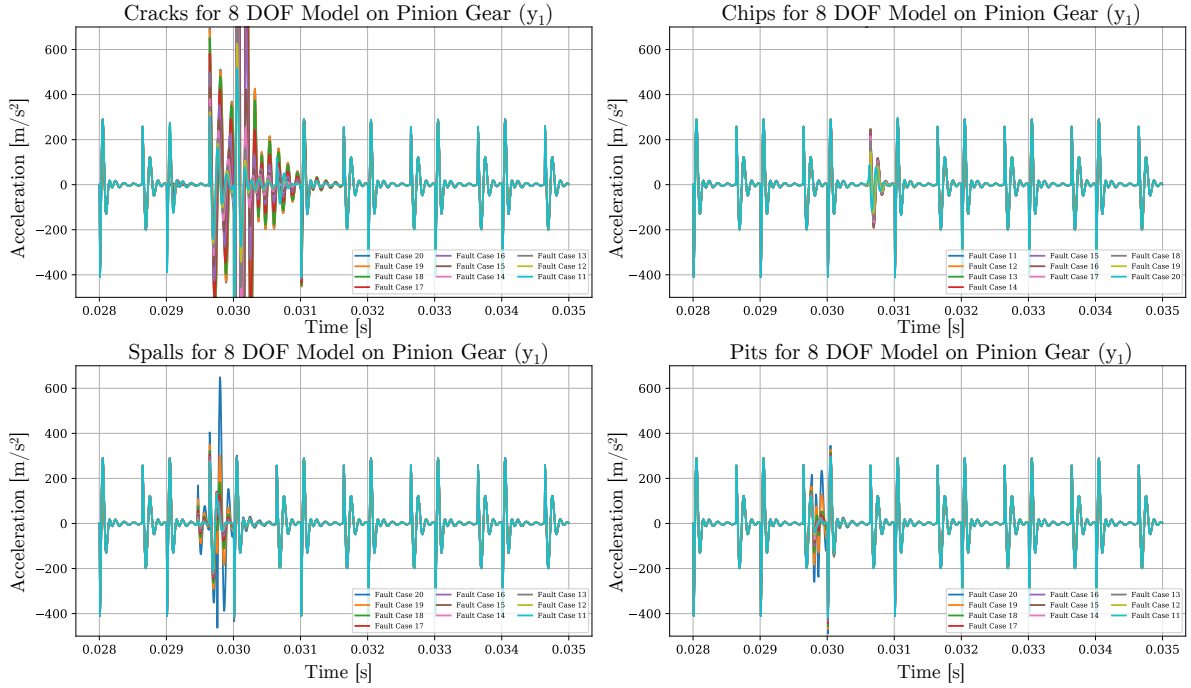


(b) Resulting Vibration signals on the pinion gear *y*-direction for the last 10 fault intensities of the six DOF model.

Figure 2.37. Vibration signals generated for various fault intensities over various fault types for the pinion gear *y* vibrations of the six DOF model.

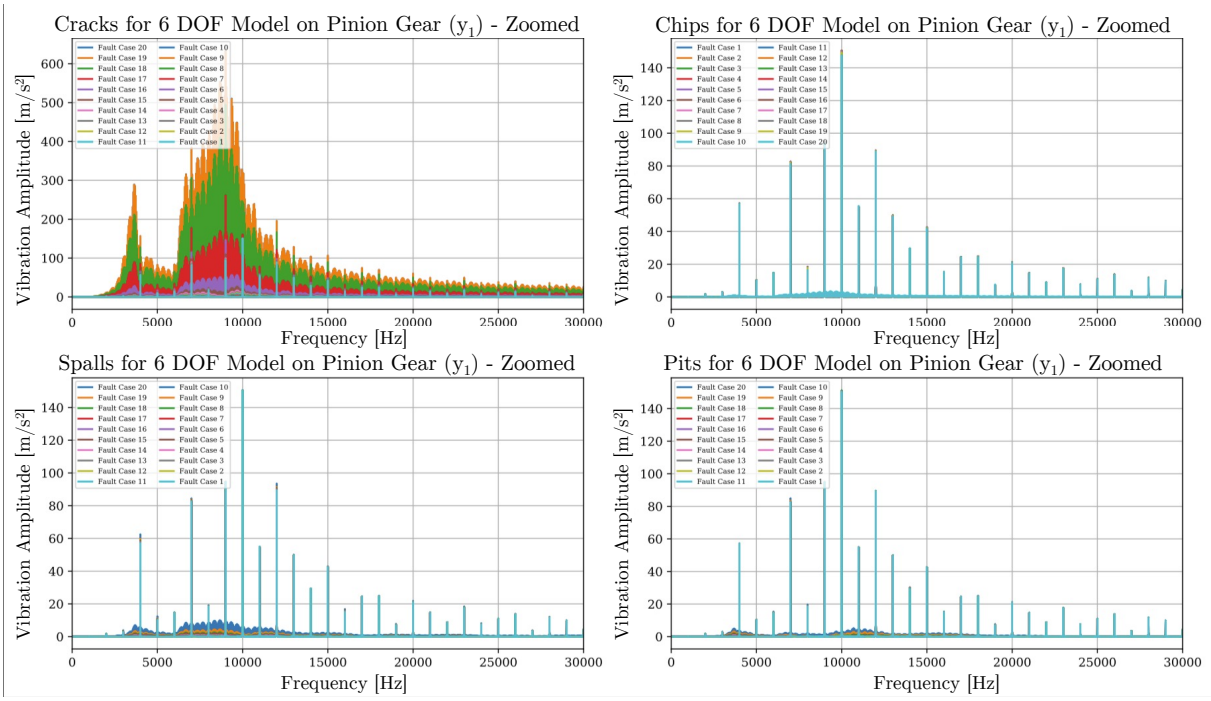


(a) Resulting Vibration signals on the pinion gear *y*-direction for the first 10 fault intensities of the eight DOF model.

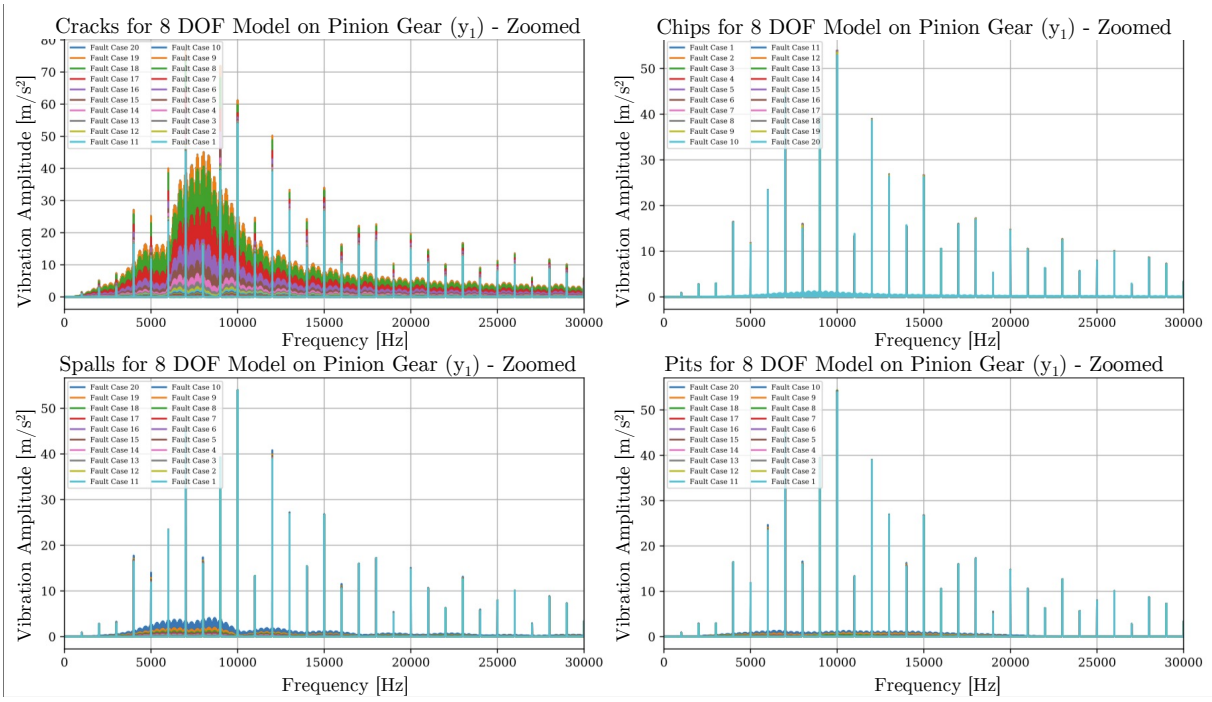


(b) Resulting Vibration signals on the pinion gear *y*-direction for the last 10 fault intensities of the eight DOF model.

Figure 2.38. Vibration signals generated for various fault intensities over various fault types for the pinion gear *y* vibrations of the eight DOF model.



(a) Resulting FFT for the six DOF model y_1 vibrations for all fault intensities.



(b) Resulting FFT for the eight DOF model y_1 vibrations for all fault intensities.

Figure 2.39. Frequency domain spectra for various fault intensities over various fault types for the six and eight DOF models.

This concludes the showcasing of Dataset B's results. Although three load cases are simulated, only the 20 Nm load case will be used to create Dataset B. Therefore, Dataset B-6 and Dataset B-8 are both simulated at a load of 20 Nm. The decision to use the 20 Nm load is completely arbitrary but may serve as a good median between the 10 Nm and 30 Nm datasets.

2.4.4 Analysis of results for both datasets

When observing **Figures 2.34(a), 2.35(a), 2.37(a)** and **2.38(a)**, one observes that quite early on in the fault intensities, crack faults may visually be picked up. One also finds that for very severe crack intensities in **Figures 2.34(b), 2.35(b), 2.37(b)** and **2.38(b)** the vibrations reach massive amplitudes (too high to meaningfully show on the plots) which are expected from a large stiffness reduction in the affected tooth. Finally, although a bit harder to see for Dataset B, all four datasets showed visible frequency modulations for the severe crack cases in the time domain (**Figures 2.36(a), 2.36(b), 2.39(a)** and **2.39(b)**).

When observing **Figures 2.34(b), 2.35(b), 2.37(b)** and **2.38(b)**, it is seen that chips, spalls and pits are only really visually detectable near the larger end of their fault intensities. Furthermore, for chips, larger faults resulted in a decrease in vibration amplitude. This result was unexpected and therefore a small experiment was undertaken to determine whether chips always result in a vibration amplitude reduction. The results were mixed, with different loads and speeds resulting in increases or decreases in vibration amplitude, with no clear pattern. Therefore, the results here will be taken as is, but warrants further investigation in the future. For spalls and pits, however, the vibration signals were seen to increase with an increase in fault intensity. Chips, spalls and pits only really showed noticeable vibration changes near the last few vibration intensities, which represent relatively large fault sizes. This implies that these fault types may be extremely difficult to detect in an industrial setup where noise may easily hide the small amplitude deviations from the healthy signal.

In conclusion, it was seen that the amplitude of vibration near cracks, pits and spalls in **Figures 2.34, 2.35, 2.37** and **2.38** tends to increase, whereas chips tend to decrease the vibration amplitude. These results show the importance of numerical models in a diagnostic framework, as three of the four fault types tend to increase vibrations in their acting region as they increase in intensity, but chips on the other hand tend to decrease vibrations. The differing shapes/signatures of different fault types across these figures also show the importance of a numerical model, as it shows that a specific fault may be coupled to a specific vibration signature, which is not only unique on a fault to fault basis, but to a dataset to dataset basis. Therefore, for both of the aforementioned reasons, simple features which work on one fault type and one dataset may not at all apply to other fault types or datasets. This justifies the need for a stronger fault classification framework that can account for such different signals. The next chapter deals with this task, but it is informative to visually illustrate this.

When analysing the frequency signals in **Figures 2.36(a), 2.36(b), 2.39(a)** and **2.39(b)**, it is seen that the gear mesh frequency and its harmonics show up in all four fault types at the predicted frequencies for each dataset. Furthermore, it is observed (very strongly for cracks) that all fault types produce the predicted amplitude and frequency modulations around the meshing frequency harmonics. This behaviour is a good indication that the dynamic models may be trusted. It is also seen for the crack, pit and spall case that with increasing fault intensity one may expect to find increasing peak frequencies at the gear meshing harmonics. The chip case, however, tends to show the opposite behaviour. This correlates to what was observed in the time domain. It was also seen that the most intense crack cases produced extremely strong frequency modulations, which may be attributed to the harsh stiffness drop for large crack sizes. This stiffness drop acts as a large impulsive force on the system and excites the system harmonics.

As a concluding remark, it is once again evident that features that might work well on one fault type cannot necessarily work well on another fault type or on another dataset, as not only does the location of the strongest amplitude and frequency modulations change between fault types, so too does the effect a fault has on the peaks of the gear mesh harmonics. Therefore, it is once again emphasized that a strong fault classification framework is necessary.

2.5 Concluding remarks

To conclude this chapter, the first novelty of the work has been presented: A generalised stiffness model that can be used to model multiple gear fault types with relative ease. This model was not only presented but also verified against existing works for correctness. Furthermore, crack faults were reformulated to fit within the generalised framework. Not only did this reformulation ease the use of cracks within the generalised framework but also is fully compatible with existing methods. Therefore, if a generalised framework is not applied, the crack reformulation developed in this work may be used to eliminate one of the integrals for crack formulation.

Furthermore, three dynamic models were developed and verified against the works on which they were based, namely Luo et al. (2019*b*) for the 10 DOF dynamic model and Meng et al. (2020) for the eight and six DOF models. From these dynamic models, two datasets were generated, namely Dataset A and Dataset B. These datasets were further evaluated for correct behaviour, and showed that the developed dynamic models produce vibrations that seem realistic.

Chapter 3 Data-driven portion: Constructing the full hybrid framework

After generating labelled datasets from the physical gearbox model, we require a data-driven model to convert this data into useful diagnostic information. Data-driven models may be defined as being unsupervised or supervised depending on the availability of labels. For the task of gearbox condition monitoring, supervised methods have strong appeal, as they are not only capable of detecting faults, but may further classify the faults into a certain class, such as cracks, chips, spalls or pits. Therefore, the focus of this work is on supervised data-driven approaches, as we have labels available to us through the physical model.

The focus of this chapter is to develop a supervised data-driven model which can utilise the synthetic data from the physical model. In turn, the combination of the synthetic data with the supervised data-driven model will be known as the hybrid model. The hope is that the hybrid model will be able to differentiate between different fault states on gears by utilising vibration data. Although the vibration data in the hybrid model is synthetic, it is hoped that the hybrid model will be able to generalise to the real asset on which the gearbox model was based.

The use of a hybrid condition monitoring model is not a novel idea and has been applied to other works before. This is however seen as novel work in the gearbox space and sets out to prove the untapped usefulness of hybrid models for gearbox fault classification. Section 3.1 sets out to define the data-driven portion of the hybrid model. Section 3.2 continues by showcasing the complete hybrid framework which will be used in subsequent chapters for numerical experimentation.

3.1 Supervised data-driven model

A supervised data-driven model requires labelled data and tries to classify input data into the different labels used for training. This section is used to argue which supervised data-driven model is most suitable for this work and justify the parameters which will be used for the chosen data-driven model.

3.1.1 Approach type selection

Data-driven techniques have advanced over time, with various challenges being solved by new approaches. Traditional approaches aim to extract expert features from data, and then manually utilise expert knowledge to classify the set of features into certain classes. Two potential drawbacks are present with such an approach. Firstly, an expert is needed to extract relevant hand-crafted features from the data. Secondly, the hand-crafted features have to be selected by an expert and analysed to predict the condition of the asset. Therefore, manual approaches suffer from an over-reliance on experts, making the condition monitoring task time consuming and costly (Khan & Yairi 2018, Lei et al. 2020).

To solve the issue of the generalisability of features, shallow machine learning techniques such as artificial neural networks (ANNs) have been proposed. These techniques aim to utilise the expert features as input and output the predicted condition of the machine. These methods lower the need for expert knowledge, as one no longer needs an expert to select the features and classify them into a machine health state. These techniques automatically learn the relationship between the expert features and the asset health state. However, these techniques still rely on an expert to set up the features (Lei et al. 2020).

Therefore, in more modern approaches, deep learning techniques have seen great success. Deep learning techniques completely remove the need for experts, as features are automatically extracted within the deeper architectures. The automatically extracted features are then fed to a feature extractor, which automatically classifies the features into a given machine health state. Therefore, deep learning techniques fully remove the need for an expert, as raw data is used as input and the machine health state is generated as output (Lei et al. 2020). There is also evidence to suggest that the features which are extracted by the deep learning approaches are more meaningful than the hand-crafted expert features. In works from Jia et al. (2016) and Verstraete et al. (2017), it was shown that deep learning methods (automatic features) outperformed shallow methods (which utilise expert features).

Therefore, this work will employ a deep learning model. More specifically, a deep neural network (DNN) will be utilised. DNNs are commonly utilised in the literature for condition monitoring tasks and will therefore provide the author with a large reference framework when developing the DNN for this work. A popular form of a DNN is called the multi-layer perceptron (MLP). This model stacks multiple densely connected layers with the idea that layers near the start extract general features and layers near the end start to extract more case-specific features. Although MLPs are more useful than their shallow counterparts, they may suffer from the fact that some forms of data have temporal or spatial features.

MLPs are invariant to the order in which inputs are fed. This is a direct result of the architecture in which inputs are connected to all nodes in the following layer. Therefore, MLPs cannot take advantage of temporal or spatial information when it exists. Therefore, although MLPs have shown their use in many applications, there are other DNNs better suited to the task. Some newer deep architectures have been developed to extract useful temporal or spatial features. Two popular choices include convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Among RNNs, a popular choice is the long short term memory (LSTM) network. Some works combine the two network types to gain the benefits of each approach.

Recall that the overarching goal of this work is to develop a hybrid model that can be calibrated through transfer learning. Therefore, it does not matter which DNN is chosen, because any DNN can be utilised and calibrated within the hybrid model. However, it will be advantageous to choose a DNN which is suited to temporal data, such as a CNN or RNN. Although both methods have been shown to work well on temporal data (which is the type of data one will deal with for vibration signals), the CNN will be the method of choice going forward.

The CNN is commonly used in literature (Han et al. 2019b), has strong temporal capabilities and allows for faster training than an LSTM. It is reiterated, however, that the focus of this work is not on optimising the data-driven model, but rather investigating a hybrid model in a larger context. Therefore, although a CNN is used, the conclusions from this work may be extended to other DNN architectures. With the decision made to employ a CNN, one needs to take a group of new factors into account, such as architectural choices, hyperparameter choices, loss functions, input data format etc. The following sections aim to address these concerns. Firstly, however, to ease future explanations, a short overview of the CNN framework will prove useful.

3.1.2 Convolutional neural networks for supervised learning

Recall that a supervised data-driven approach is being utilised, as labels are available during training. Therefore, the following descriptions pertain to supervised CNN approaches, which will only be referred to as CNN approaches going forward. CNNs receive significant attention in the literature and have proven to be valuable where data is spatial or temporal. As with any DNN, the parameters necessary to extract meaningful features and make strong classifications are generally not clear and some degree of fine-tuning is necessary.

The general architecture behind a CNN is, however, quite standard and follows the following pattern: The front layers of the CNN are composed of convolutional layers and pooling layers (Zhao et al. 2019, Lei et al. 2020). Furthermore, it is common to see some form of regularisation applied between pooling and convolution layers. Works that include such normalisation include Ganin et al. (2017), Han et al. (2019a,b), Chen et al. (2020), Liu & Gryllias (2021). These blocks of convolution-pooling-regularisation (CPR) can be stacked to deepen the NN and increase the feature extraction capabilities. After sufficiently many CPR blocks have been stacked, one generally flattens out all features into a single vector (Lei et al. 2020), which is commonly referred to as the latent

feature space. This latent space contains (hopefully) the most meaningful and separable feature descriptions possible.

This latent space may then be fed into a normal MLP framework near the end of the CNN. This tail-end MLP acts as a classifier (Lei et al. 2020). Recall the choice to not use an MLP from start to finish was to extract temporal information from the input data, which has been achieved by the time values reach the latent space. At this point, one is only interested in finding some set of separable non-linear combinations of these features to classify the signal. Therefore, an MLP will suffice, as the latent space does not necessarily have a temporal component to it.

Therefore, whenever future portions of work refer to the feature extractor, reference is being made to the full set of stacked CPR blocks, which take a raw signal as input and output the latent features space. When portions of work refer to the classifier, reference is being made to the posterior MLP structure of the network, which takes the latent feature space as input and outputs a classification from the given features. When these two components are combined, one is essentially converting raw signals into classifications, and this combination is what will be referred to as the data-driven approach.

3.1.3 Architectural considerations

When a CNN is implemented, one needs to consider how wide and deep the CNN must be made. Generally, the deeper the CNN, the richer the features which may be extracted. This has been shown empirically on rotating machinery data in works such as Han et al. (2019a,b), with the structure they call CNN3 - This name will consistently be used from this point forward. This however comes at the cost of a larger network, which is more expensive to train. Since the novelty of this work is not focused on building a new CNN structure, as this has been done many times in literature, it will be wasteful to start from scratch and optimise a new CNN structure. It is more useful to draw from a proven architecture. Therefore, the architecture from CNN3 will be replicated, as this has shown a good balance between the minimum number of layers required and high classification accuracy on rotating machinery diagnostics.

Furthermore, it is a common trend that with deepening CNN layers, one also increases the number of convolution kernels whilst decreasing the kernel size. The logic behind this is as follows:

- Regarding kernel size: As one convolves and pools across each layer, the number of features shrinks as well, therefore one needs to shrink the corresponding kernel size.
- Regarding the number of kernels: The first layers of a CNN only extract very general features from the data. As one goes deeper, the number of ways in which the previous layers' features can be combined to extract meaningful features increases exponentially. Therefore, the number of kernels is increased to capture as many meaningful features as possible.

Therefore, this structure is beneficial to maximise feature richness and will be applied. This same trend is seen in CNN3, further confirming this as a good choice of architecture. Another consideration is the type of pooling layer to use. Generally, most works use max-pooling and therefore this pooling technique will be used in this work.

It is further seen that after a convolution and pooling layer, one adds some type of regularisation layer. This is generally preferred as NNs have a large number of parameters (magnitudes more than the amount of available input data) and may therefore be prone to overfitting. One therefore needs a regularisation technique to prevent or at least minimise overfitting to the training data.

Two types of regularisation are common among CNNs, namely batch normalisation (BN) and dropout. These layers may be used in isolation or combination. BN tries to address a common issue with DNNs: Weight updates during backpropagation are made with the assumption that the feature distribution in already updated layers is still the same as before the backpropagation. This is not the case and one can think of the shallower layers

chasing a moving target, as backpropagation works from the deep end to the shallow end. This is referred to as internal covariate shift of the network (Ioffe & Szegedy 2015).

To combat this issue, BN is generally accepted to aid in reducing the internal covariate shift. For BN to be implemented, as is evident in the name, one needs to train the network using mini-batches or full batches. This is however common practice within networks and can almost be assumed as a default. Therefore, BN applies to almost all networks. BN takes the batch of activation outputs from the previous layer and standardises them to zero mean and variance of one. This has the effect that during backpropagation, the assumption of feature distributions (which are now standardised) are more accurate - therefore one minimises the internal covariate shift. This speeds up training and simultaneously acts as a form of regularisation, as weights are continually standardised (Ioffe & Szegedy 2015).

Dropout is another (quite popular) technique to regularise a network. The concept is simple: A dropout layer may be inserted after any normal NN layer (convolution, max-pooling, densely connected etc.). This layer then acts as a type of randomised filter in which one specifies some dropout percentage. Then, during training, a certain percentage of nodes are not allowed to propagate their signals forward, essentially passing a zero value to the activation function. It acts as if the node from which the signal came 'dropped out' of the network. A larger dropout percentage refers to a larger fraction of nodes being turned off during training. The key concept behind dropout is that it randomly turns off nodes during each training epoch (Zhang et al. 2018).

Since the dropouts are random, the network is "forced" into learning general features (Zhang et al. 2018) instead of overfitting to specific nodes, generating highly specific features. If dropout is not applied, it is easily conceivable that a network may learn to overfit on specific data points in the training dataset and oversaturate a set of weights along different input to output paths. If, however, these oversaturated paths are randomly interrupted during training, the network is forced to utilise a combination of all weights and nodes as a whole to learn useful features instead of over-relying on one specific set of weights and nodes. This therefore acts as a regularisation method, as it attempts to prevent any overfitting attempts by the network.

For this work, some preliminary tests were conducted with both BN and dropout being applied. Although BN resulted in faster convergence, the overall accuracy suffered slightly when compared to dropout. Furthermore, tests with BN showed diverging classification accuracies, whereas dropout showed more consistent accuracy results, usually converging to a maximum value. Therefore, from the preliminary results, dropout was chosen as the regularisation technique for the CNN. This concludes architectural considerations. The following section deals with some smaller nuances of building and training a NN.

3.1.4 Hyperparameter and loss considerations

As with any data-driven method, NNs have important hyperparameters which need to be determined. One of the most important hyperparameters to deep learning is that of the activation function. Deep networks are prone to what is known as the 'vanishing gradient' problem. The deeper the network gets, the smaller the backpropagation signals become, essentially stagnating the weight updates for the deeper layers. Common activation functions such as the sigmoid or hyperbolic tan all suffer from the vanishing gradient problem. It has however been shown that the rectified linear unit (ReLU) overcomes the vanishing gradient problem (Zhu et al. 2016). Therefore, ReLU activations are quite commonly found in DNNs, and will therefore also be used in this work.

Another important consideration is the loss function. Loss functions need to be determined based on the task at hand. For this work fault classification is attempted. Therefore, categorical cross-entropy is a good loss function, as it is mathematically well suited to the backpropagation process, where large deviations from optimal values result in large weight updates. For this reason, categorical cross-entropy is a common choice for classification.

One must also decide which optimisation algorithm to use. The optimisation algorithm defines the methodology with which the loss function is used to update the network weights. A good default choice is Adam (adaptive

moment estimation), as it has been shown to work well in comparison to other adaptive gradient techniques (Ruder 2016).

A final consideration for the network is the batch size. The batch size does not necessarily have a large impact on training accuracy, but more on training speed. There is however a trade-off between the two. A very small batch size takes up little RAM, but also is a noisy representation of the entire dataset and will thus have quite stochastic training behaviour. Smaller batches also lead to longer training times. A very large batch size takes up a large amount of RAM, but is faster to train and will converge to a more favourable minimum and be much steadier during training. Therefore, for each problem it is different, but generally, there is some good balance between a small enough batch to not fill up all RAM but also large enough to be representative of the entire dataset, leading to more steady training. For this work, a default batch size of 60 was chosen as a good trade-off.

3.1.5 Data considerations

A supervised data-driven approach is not solely dependent on the model, but rather also on the input data itself. As Andrew Ng says: Data is food for a NN. Therefore, one needs to take care in selecting the format of data to be fed to the NN.

The first consideration when dealing with vibrational signals is the domain in which the data will be presented: time, frequency or time-frequency domain. For this work, the raw time domain data is fed to the CNN structure. The reason behind this is twofold. Firstly, the use of the time or frequency domain leads to a 1D CNN structure, which is much faster to train. As the overarching goal of this work deals with transfer learning research and not necessarily architecture or data optimisation, this choice seems justified, as 1D CNNs train faster than 2D CNNs. Even if the 1D techniques have lower accuracies, they will prove or disprove the use of transfer learning much faster than training a 2D CNN.

Secondly, the time domain is chosen over the frequency domain purely due to its popular appeal. From the literature review, it was seen that the transfer learning approaches all utilised the raw vibration (time domain) signals for the experiments. Therefore, to minimise variables during investigations in this work, a proven data domain is chosen, namely the time domain.

With the time domain being the established domain of choice, one still needs to consider the most useful format of the data. Some preliminary investigations (not reported on here) experimented with utilising individual meshing vibration signals. Thus, the method splits one full rotation into n equal segments, where n corresponded to the number of teeth on the pinion gear. A healthy tooth would thus receive a healthy label and a faulty tooth a fault type label. This is format may be advantageous, as it allows the user to not only detect a fault, and fault type, but further classify the location of the fault. This data format however proved to be quite troublesome. These unfavourable results are therefore not included in the current work, but a brief hypothesis is proposed as to why the poor results were seen:

The tooth-by-tooth vibration signals are seen to vary in shape between different fault types. That is to say, different fault types are visually unique, and should therefore be capable of being classified. However, since the data is fed into the CNN as temporal data, it is one-dimensional and therefore difficult to truly capture the spatial variations in the different vibration shapes, which would be a two-dimensional input. Furthermore, the signals are too short to carry any meaningful temporal information and therefore the CNN is not suited to this task, at least in the one-dimensional case. Perhaps if the data is fed as an image, a 2D CNN would perform better. This is however expensive, as a vibration signal is only a line in two-dimensional space, implying there is a large area of useless data around the signal. As stated earlier, the goal of this work is not data optimisation and this route was not explored further.

Instead, a full rotation's worth of data was fed to the CNN. Therefore, if a pinion gear had 16 teeth, the input signal would consist of a fixed-length input of 16 gear mesh signals. If the signal contained a faulty tooth, the entire signal is labelled as a fault, otherwise, it is labelled as healthy. This representation loses the benefit of identifying the exact fault location, but it is not unrealistic to expect one to build a tooth by tooth NN after the

fact to predict which tooth is faulty - This is however once again not the focus of the present work. The longer vibration data format also holds truer to the types of formats found in the literature, and should result in easier convergence of models, as it is a proven data input format. It should also be said that due to the nature of the longer data format, it inherently contains a strong temporal component and therefore should be well suited to the strengths of a 1D CNN.

This concludes the necessary justifications for the CNN model which will be used. The final subsection is aimed at showing the structure and exact parameter values for the data-driven model.

3.1.6 Full data-driven model

The CNN structure to be employed for this work is shown in Figure 3.1.

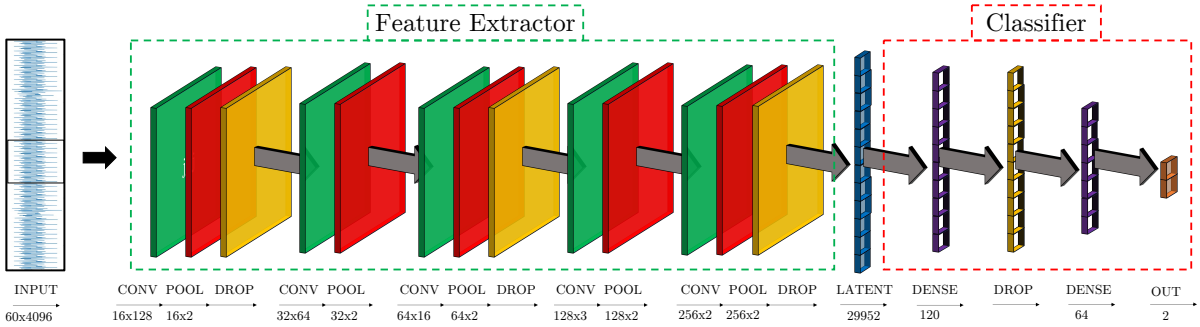


Figure 3.1. The data-driven model to be used in this work, namely a CNN. Within the first portion of the CNN, namely the feature extractor, green, red and yellow layers represent convolution, max-pooling and dropout layers respectively. The blue layer between the feature extractor and classifier is a one-dimensional feature vector, which is referred to as the latent feature space. In the final portion of the CNN, the classifier may be seen with purple, yellow and orange vectors representing densely connected, dropout and output layers respectively. Numbers beneath the feature extractor layers are given as $a \times b$, where a refers to the number of kernels and b the kernel size. In the classifier layers, numbers represent the number of nodes, all densely connected. For the input, full rotation signals are used and are represented by 4096 values. Input signals are passed in batch sizes of 60. These parameters may be changed, but for illustrative purposes, these values are used here.

With the CNN structure given, it is clear that from raw time domain data, one obtains a binary prediction at the other end of the network relating to the health state of the machine, i.e. healthy or faulty. This could also be adapted to a fault type identifier network by simply having five output nodes instead of two. The green portion of the network refers to the feature extractor, as previously described. After the feature extractor, the feature latent space is obtained. This latent space may be seen as an array of meaningful features to describe the raw data. The red portion of the network refers to the classifier portion of the CNN. The classifier takes latent features as input and tries to find meaningful combinations of these features to classify the signal as healthy or faulty.

Furthermore, note the naming and numbering beneath the relevant layers. As described earlier, the trend of increasing kernel numbers with decreasing kernel size is seen in this network. Additionally, note that all max-pooling layers are of size 2, the minimum required amount for information compression.

This architecture removes the regularisation layer once every second CPR block. The choice was made to copy the architecture directly from Han et al. (2019a), without deviation, therefore the omission of the dropout layers every 2 blocks are copied.

As a final note, note that from the large latent space representation of features, the information gets compressed to 120 nodes, and then further to 64. Finally, these 64 nodes are combined to make the final prediction on the health state of the input signal.

With the architecture described, the necessary parameters to implement this model may be found in Table 3.1.

Table 3.1. Necessary parameters to set up the data-driven model in Tensorflow 2.

CNN Parameter Description	Parameter Choice
Training Environment	Google Colab
Batch Size	60
Input Data Size	4096
Convolutional Stride Length	1
Max-Pooling Stride Length	1
Dropout Rate	Case Dependent
Hidden Layers Activation Function	ReLU
Output Layer Activation Function	Softmax
Optimiser	Adam (Default Tensorflow Settings)
Loss Function	Categorical Cross-Entropy
Weight Initialiser	He-Uniform (Tensorflow)
Deep Learning Framework	Tensorflow 2 (Google Colab)
Training GPU	Tesla K80 (Google Colab)

By combining the illustration of the CNN with the tabulated values, the data-driven model may be constructed within Tensorflow 2. Note that a fixed value for the dropout percentage is not given. This is because dropout is case dependent and therefore will be varied to suit the desired dataset. Han et al. (2019a) recommend a dropout rate of 0.5%, and this will be used by default.

3.2 Showcasing the hybrid approach for gearbox condition monitoring

At this stage, we have two synthetic datasets from section 2.4 and the data-driven model defined in section 3.1. Therefore, the hybrid approach may finally be described. An illustration of the proposed hybrid approach may be seen in Figure 3.2.
The framework for developing the hybrid model may be split into multiple steps:

1. A single synthetic vibration signal is generated from the physical gearbox model. The gearbox model requires fault parameter, speed, load and geometry information to generate a synthetic signal.
2. The steady-state portion of the signal is identified and cropped.
3. Within the steady-state portion, the signal is windowed to a single rotation of the input shaft. Single shaft rotation points may be identified through the use of a tachometer. The benefit of a physical model is that when the dynamics are solved, one automatically has full knowledge of the input shaft angular displacement. Therefore, identifying single rotation windows is simple.
4. The windowed signal is resampled to 4096 points. This is necessary, as slight variations in shaft speed may cause some windows to be longer than others. Furthermore, the CNN requires a fixed input data size.
5. The resampled, windowed signals are then standardised and compiled, along with their labels, into a database.
6. Steps 1-5 are repeated for every possible fault type and intensity to be included in the database.
7. The database of different fault types and intensities may be fed to the CNN in batches of 60 for training or testing.
8. If training:

    (a) Dropout is turned on, as the model may potentially overfit on the data if regularisation is not implemented.
    (b) Labels are passed through the CNN, allowing the model to update the weights to best fit the given training data.

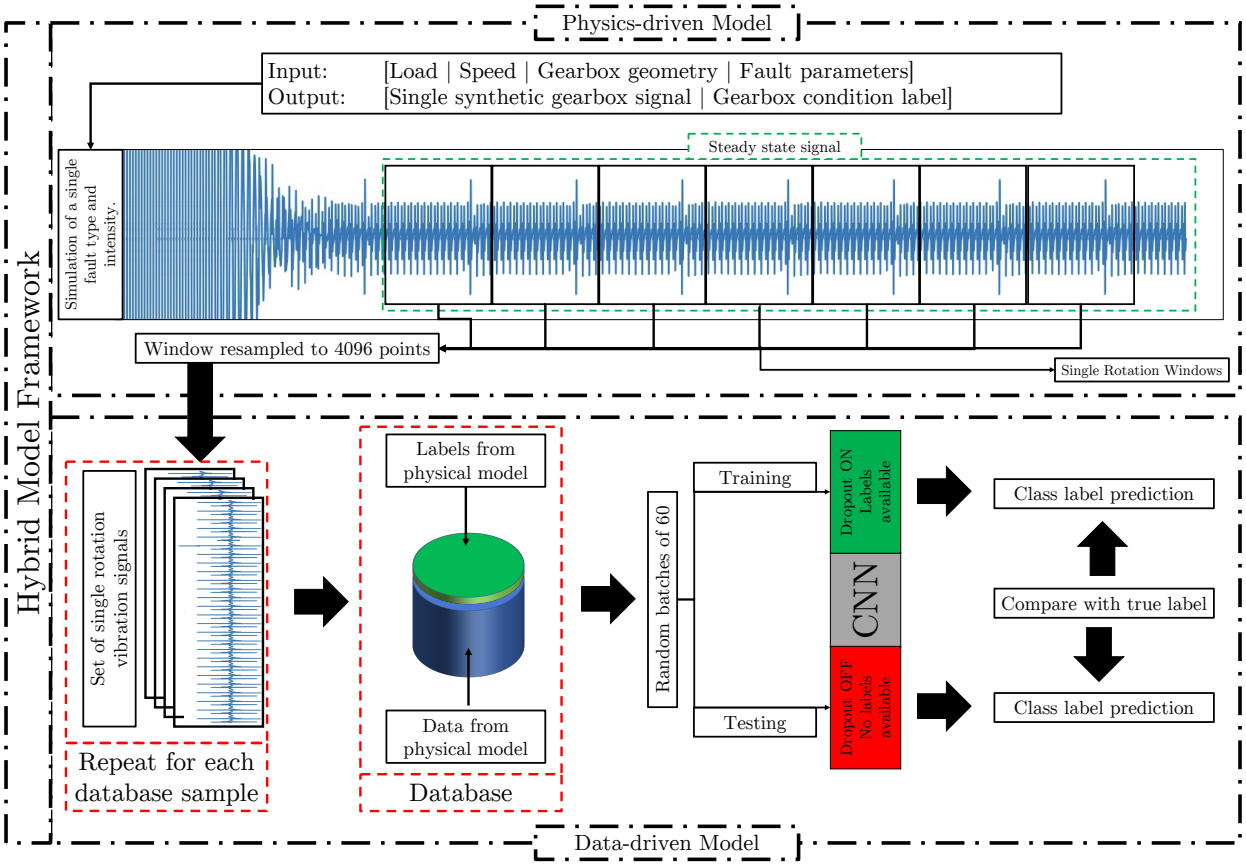


Figure 3.2. Illustration of the hybrid modelling framework. The top portion of the figure shows the result from the physical model, namely synthetic, labelled signals. The bottom portion of the figure shows the data-driven model, which utilises the output of the physical model to train a CNN.

(c) The weights may be updated by comparing the CNN's predicted class label to the true class label.

9. If testing:

(a) Dropout is turned off, as the model is no longer adapting its weights, therefore no regularisation is necessary.
(b) Labels are not passed the CNN, as the model is not adapting weights, only testing its performance on an unseen dataset.
(c) The performance of the model may be evaluated against the true test set labels, to determine how well the model generalises to unseen data.

The framework outlined above describes the method of operation for all future training and testing in this work. If the hybrid model had to be applied to industry, the only difference to the framework above is that step 1 is removed, and replaced with actual labelled accelerometer data. However, in practice labels may not be available. In this case, the hybrid model is pre-trained with synthetic data as outlined above. After pre-training, the weights are frozen. Finally, the real, unlabelled accelerometer data can then replace step 1, and only the testing portion of Figure 3.2 will be utilised, where real gearbox conditions are predicted.

3.3 Concluding remarks

A proven data-driven architecture has been selected and fully defined. The necessary arguments have been presented to justify the type of data-driven approach, namely a CNN. Furthermore, the architectural shape, hyperparameters, loss functions and data considerations were all discussed and motivated.

Finally, the full hybrid model framework has been proposed, and the method of combining the physical model with the data-driven model has been shown. The hybrid model training procedure was described and serves as the basis for how training will be implemented going forward. At this point in the work, it is sensible to test the hybrid model from the generated datasets. Therefore, the following chapter focuses on identifying whether the hybrid model approach is feasible.

Chapter 4 Numerically investigating the usefulness of the hybrid model: A test for noise robustness

It has been stated that a hybrid model may solve the major drawbacks faced when employing a data-driven or physics-driven approach in isolation (Kundu et al. 2020). However, the developed hybrid model from Chapter 3 has not yet been proven to work. Therefore, this chapter aims to set up a numerical experiment to test the hybrid model. The main goal of this work is to understand how well the developed hybrid model works, but furthermore, investigate how it may be adapted to apply to real industrial systems. Adaptation steps such as modifications to the synthetic data, data pre-processing, and model calibrations may be necessary to apply the hybrid model to industrial data.

The hybrid model is trained on synthetic data, which is generated in a controlled numerical environment. However, if the hybrid model is ultimately implemented on an industrial gearbox, industrial data will be used, which is not captured in a controlled environment. Industrial data may therefore differ from synthetic data in multiple ways. For example, data captured on an industrial gearbox may be affected by factors such as temperature, vibrational excitation from nearby machines, motor dynamic effects, dynamic loads, measurement noise etc.

It is infeasible within the scope of this work to investigate all of these effects on the hybrid model's capability to classify faults. Therefore, this chapter will only focus on investigating the classification robustness of the hybrid model in noisy environments. Noise is the simplest of the aforementioned effects to incorporate into the existing datasets and is therefore chosen. As this work does not utilise industrial data, the generated datasets from the previous chapter are used for the noise investigation. This implies that for the numerical investigation to be completed, the synthetic datasets must be infused with noise.

An important part of training a data-driven model, and therefore the second half of the hybrid model, is good quality data. It may prove useful to pre-process the noisy data by utilising signal processing techniques, making the fault classification task simpler. Therefore, a second, but parallel investigation may be done to determine the usefulness of signal pre-processing techniques for the hybrid approach. For this work, time-synchronous averaging (TSA) will be used, as it is a well-known signal processing method to combat noise.

Therefore, the aim of this chapter is twofold:

- Firstly, the aim is to investigate the hybrid model's classification accuracy in identifying multiple fault types at varying intensities within a noisy environment.
- Secondly, the aim is to investigate whether the hybrid model classification accuracy can benefit from the use of TSA.

The outline of the chapter is given in Figure 4.1. The first portion of the work focuses on pre-processing the synthetic datasets and the second portion on investigating the classification accuracy of the hybrid model on the

pre-processed datasets. A more detailed description of the process is given in the respective chapters denoted in Figure 4.1.

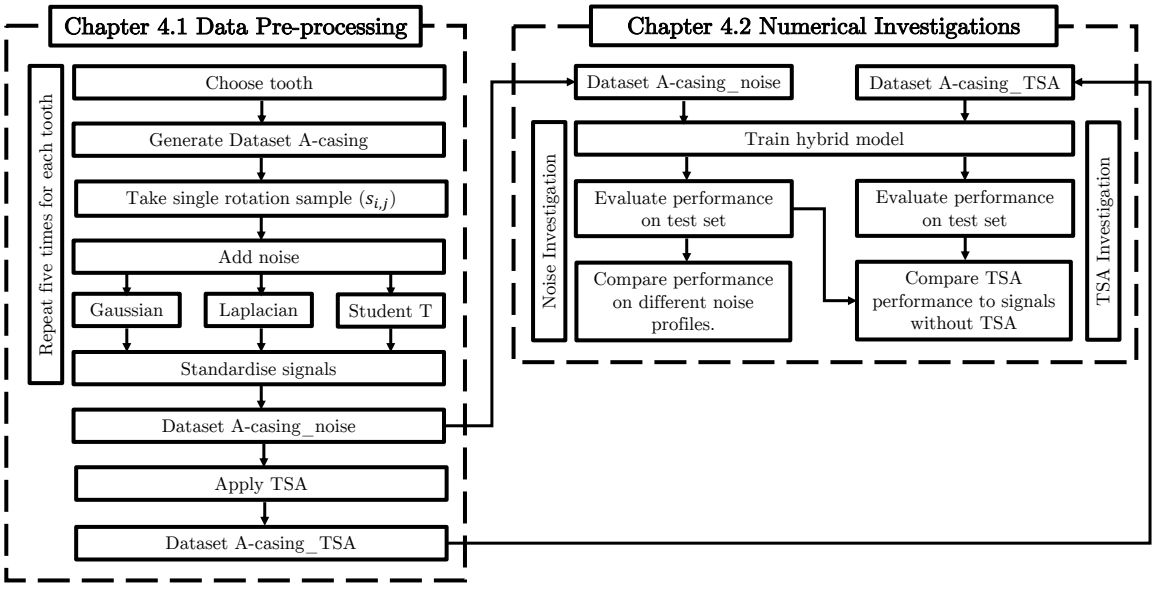


Figure 4.1. Outline of Chapter 4. The first portion of Chapter 4 focuses on pre-processing Dataset A-casing. The latter portion of the chapter investigates the effect of different noise profiles as well as the application of TSA on the hybrid model's predictive accuracy.

For both investigations, Dataset A (based on the works from Luo et al. (2019*b*)) is used. Recall that Dataset A is further split into Dataset A-pinion and Dataset A-casing, relating to the horizontal vibration of the pinion gear (x_1) and gearbox casing (x_f) respectively. In the investigations, Dataset A-casing is utilised. This choice is not arbitrary: The faults are modelled directly on the pinion gear, therefore the x vibrations directly on the pinion gear could potentially be much easier to classify. Therefore, to make the classification problem more challenging, and closer to the industrial case, the choice is rather made to utilise the gearbox casing vibrations. These vibrations comprise of multiple elements in the 10 DOF model, and may "hide" the faults more effectively. Furthermore, it may be argued that in the industrial setup, accelerometers will likely be mounted on the gearbox casing and this dataset will therefore be a better training dataset for a model which needs to be employed directly in industry.

The first portion of this chapter sets out to pre-process the casing dataset. The pre-processing involves the extraction of single rotation samples from Dataset A-casing, which are then infused with different noise profiles. Next, the signals are re-standardised to ensure that all forms of data are internally consistent within the dataset. This dataset is then recompiled as Dataset A-casing_noise. Finally, TSA is applied to the standardised, noisy data to create a second dataset, namely Dataset A-casing_TSA.

The latter portion of the chapter contains the numerical investigations that aim to not only test the robustness of the hybrid model in noisy environments, but also the efficacy of using signal processing techniques to aid the hybrid model. This is followed by the showcasing of the investigation results and conclusions from the investigations.

4.1 Synthetic dataset pre-processing

The simulation data containing the casing x-vibrations from Dataset A are used for this chapter. The vibrations of interest are measured on a different location than where the fault is induced (on the pinion gear) and therefore this dataset represents an adequate challenge for the hybrid model. Furthermore, three types of noise are introduced, namely Gaussian noise, Laplacian noise and student-T noise with 2 degrees of freedom.

The motivation for adding noise, as stated earlier, is to investigate how accurately the hybrid model can classify faults within a noisy environment. To further test how different noise profiles may affect the hybrid model accuracy, three different noise profiles are used. Gaussian noise is a common assumption and may serve as a baseline. Laplacian and student-T noise distributions have larger tails, and therefore more impulsive noise is expected. Therefore, these noise profiles may serve as references to see whether impulsive noise affects the hybrid model's predictive performance over the shorter-tailed Gaussian noise.

Pre-processing of Dataset A-casing is necessary to include the noise profiles outlined above. Furthermore, before TSA can be applied, the noise has to be infused into Dataset A-casing. Therefore, the necessary steps to transform Dataset A-casing to the noise-infused Dataset A-casing_noise and finally Dataset A-casing_TSA is outlined below:

1. Extract the single rotation portions from Dataset A-casing.
2. Add unique noise at a certain noise level to each rotation sample and restandardise the signal relative to a noise-infused healthy sample. Append the sample to Dataset A-casing_noise.
3. Apply time-synchronous averaging (TSA) to Dataset A-casing_noise to obtain Dataset A-casing_TSA.

The following sub-sections aim to detail the steps above.

4.1.1 Noise modelling and standardisation

As outlined in Figure 4.1, we start by extracting a single rotation sample from Dataset A-casing, $s_{i,j}$. This sample signifies the jth full rotation window within the ith full-length steady-state portion of the simulation. This single rotation is infused with noise, according to a certain noise level (defined shortly).

To make the three noise profiles comparable, we simulate the distributions to be as standard as possible, and then scale the distributions to match a certain noise level. For the standard distributions, the Gaussian is assumed as zero mean with unit variance. The Laplacian distribution utilises a location of zero and a scale of one. Finally, the student-T utilises 2 degrees of freedom.

A Gaussian noise profile is a general assumption from literature, whereas a Laplacian or student-T noise profile will simulate thicker tails, corresponding to more impulsive noise, which may be closer to industrial noise profiles. Zak et al. (2017) conducted work related to a heavy-duty gearbox operating in a harsh environment. They state that the signals of interest within such an environment are expected to be impulsive. They continue to state that models of industrial data are moving away from Gaussian noise profiles to more impulsive noise profiles. Therefore, the investigation is well suited to determine how less impulsive (Gaussian) and more impulsive (Laplacian and student-T) noise profiles affect the hybrid model's classification performance.

To determine the exact amount of noise to be added, a signal to noise ratio (SNR) value is defined. For this work, a SNR of 1 is chosen. This implies that the power of the noise is equal to the power of the signal. One may also define the SNR in terms of a decibel scale, which is used for this work as follows:

$$SNR = \left(\frac{RMS_{signal}}{RMS_{noise}} \right)^2 \tag{4.1}$$

$$SNR_{db} = 10 \log_{10}(SNR) \tag{4.2}$$

where RMS refers to the root mean square of the signal. It is initially unclear how one must manipulate the noise parameters to correspond to the relevant RMS. Therefore, a more general approach is taken. Firstly, the RMS of the signal is determined. This is a known value and is simple to calculate. Then, knowing the predefined SNR_{dB} value, we rewrite equation (4.1) and substitute it into equation (4.2) as follows:

$$RMS_{noise} = RMS_{signal} \times 10^{-\frac{SNR_{dB}}{20}} \tag{4.3}$$

This is adequate information to calculate the required RMS for the noise to result in the necessary SNR_{dB}. However, it is still unclear how one should adjust the parameters of the noise profiles to correspond to the required SNR_{dB}. Therefore, instead of developing mathematical relations between noise RMS and noise

parameters, a simpler approach is followed. A standard noise distribution is created (zero mean, variance of one, scaling parameters of 1 etc.) and the RMS of this standard noise distribution is determined. Depending on what the required noise RMS is, a scaling factor is calculated with which all values in the noise profile are multiplied to obtain the necessary SNR_{dB}. Thus:

$$SF = \frac{RMS_{noise,required}}{RMS_{noise,default}} \tag{4.4}$$

where $RMS_{noise,required}$ refers to the value from equation (4.3) and $RMS_{noise,default}$ refers to the noise generated by the default noise parameters described earlier. Therefore, one multiplies the default noise profiles with SF to correct the noise amplitude to the correct RMS value. This results in noise signals labelled $n_{i,j}$, which have corresponding indices directly related to the synthetic signals $s_{i,j}$. At this point the final noisy signal ($x_{i,j}$) is found by applying:

$$x_{i,j} = s_{i,j} + n_{i,j} \tag{4.5}$$

As a final step, the noise-infused signals are standardised. The process is identical to Chapter 3.2. Therefore, the standardised noise-infused signal is generated from:

$$x_{n|i,j} = \frac{x_{i,j} - \bar{x_r}}{\sigma_{x_r}} \tag{4.6}$$

where $x_{n|i,j}$ refers to the standardised signal for the jth rotation of the ith normalised, noise-infused sample. $\bar{x_r}$ and σ_{x_r} refer to the mean and standard deviation of the reference signal, respectively. The reference signal, as described in Chapter 3.2 is set as the healthy case, where no faults are present.

At this point a discussion is necessary to explain the construction of Dataset A-casing_noise: Dataset A_casing contains four fault types, with 20 different fault intensities, and J full rotations, all arbitrarily simulated on the fifth gear tooth. Therefore, Dataset A_casing contains a matrix of 80 (4×20) by J unique signals. However, this dataset fails to account for faults on different pinion teeth. Since signals are aligned according to single shaft rotations, this implies that this dataset always contains faults in the same phase of rotation. This will lead to a strong bias when training the hybrid model, as it will only have examples of faults on the fifth gear tooth. It is therefore important to expand the dataset to include examples from each possible pinion tooth.

Dataset A_casing is constructed from a 10 DOF model which has 16 pinion gear teeth. Therefore, to construct Dataset A-casing_noise, the data from Dataset A-casing is generated 16 times over (once for each tooth location), each time with the 80 unique signals of J rotations for each tooth. Therefore, Dataset A-casing_noise contains 1280 unique signals of J rotations. These unique signals, however, only represent a single example for each possible fault type, intensity and location. Therefore, to include multiple unique samples for any given fault type and intensity on a tooth, the dataset is generated five times over. That is to say, for a given tooth, with a given fault and given intensity, five examples with J rotations are generated, infused with noise as outlined above and appended to Dataset A-casing_noise. This results in a dataset with 6400 (4 fault types $\times$20 intensities $\times$16 pinion teeth $\times$5 duplications) unique samples, with J rotations each.

Finally, since only the first fault intensity is representative of a healthy gearbox state, and nineteen other intensities represent a fault state, the number of healthy samples are expanded. The dataset may be roughly balanced to contain equal parts healthy and fault samples, by generating a healthy signal for every other signal generated. That is to say, 6400 unique healthy signals, with J rotations each is generated. Therefore, the final noise dataset, Dataset A-casing_noise, consists of 12800 unique signals, each with J rotations. This process is repeated for each noise case.

A single set (i) of J standardised noise signals for all three noise cases are seen in Figure 4.2. These signals represent a crack fault on the fifth gear tooth. It is clear that the Laplacian noise is more impulsive and is a direct function of the larger tails of the Laplacian distribution. We do however only observe a slight difference from the Gaussian noise, with no major outlier values being seen. When looking at the student-T distribution, however, one sees much larger impulsive behaviour. This is once again attributed to larger tails associated with this distribution. This distribution leads to some large outliers, which imitates random impulses found in industry-type signals (Schmidt et al. 2020).

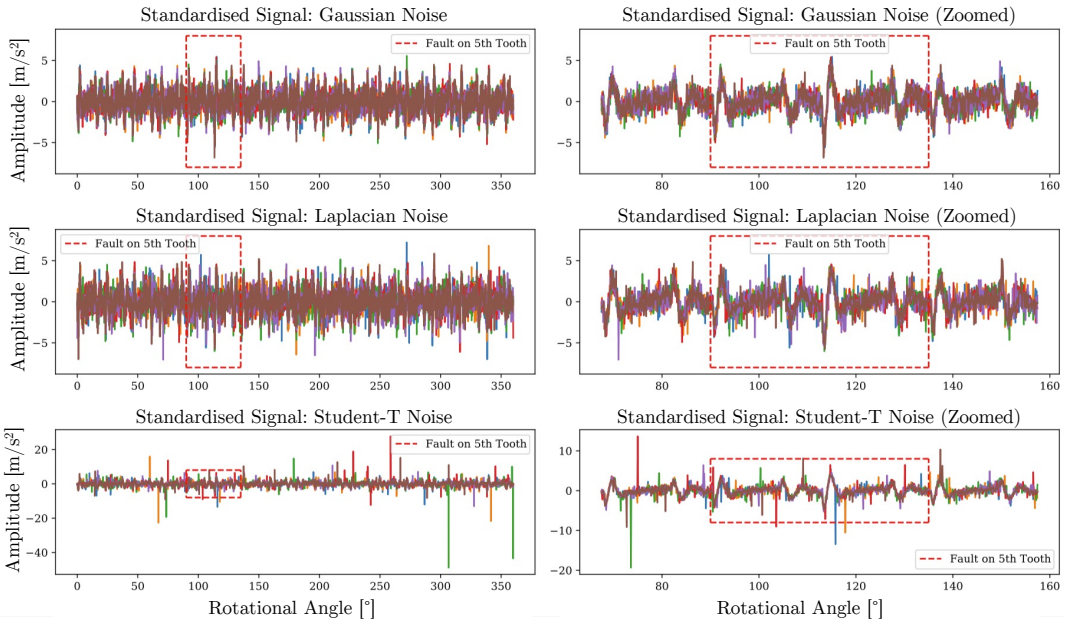


Figure 4.2. Illustration of J rotations' worth of noise-infused signals. Signal amplitudes are shown as a function of the pinion gear rotation angle. Note that these are the resampled signals and therefore contain 4096 data points.

The Gaussian noise does not fully hide the crack fault but does seriously impede its detection, at least in the time domain. The more impulsive Laplacian and student-T noise fully hide the crack faults, making the diagnostic task extremely difficult.

4.1.2 Noise dataset adaptation: Signal processing through time-synchronous averaging (TSA)

As a measure of eliminating some of the noise, a simple time-synchronous averaging (TSA) technique is applied to the noise-infused signals. In this work, TSA may be applied to align with pinion gear or driven gear rotations. Since the faults are induced on the pinion gear, the decision is made to take averages over full pinion shaft rotations. This is necessary, as this aligns the samples, such that the fault signal always occurs in the same phase within the rotation. If the driven shaft rotations were used, the fault signals would be averaged out of phase and would not be detected. The TSA is defined, using the existing notation as:

$$TSA = y_i = \sum_{j=1}^{J} \frac{x_{n|i,j}}{J} \tag{4.7}$$

where J refers to the number of full rotations initially extracted from s_i. The result will be i signals, each having been averaged over their respective J noise infused signals. For this work, six full rotations are extracted per 0.4s simulation. Ideally one would utilise more rotations, as TSA averages out the noise. The average of the noise becomes closer to the expected value, with increasing numbers of signals to average over. Therefore, a longer simulation of 4s was used, resulting in $J = 60$ rotations. The resulting TSA signals (for the same crack fault) are seen in Figure 4.3. The process is repeated for each set of samples from Dataset A-casing_noise to generate a second dataset, Dataset A-casing_TSA. This dataset has the same shape as Dataset A-casing_noise, except all J rotations are averaged away. Therefore, Dataset A-casing_TSA consists of 12800 unique TSA signals, with a single time-synchronised rotation sample. Once again, this process is repeated for each noise case.

The benefits of using TSA in the hybrid modelling process is clear. TSA attenuates the noise components and makes it possible to see the crack damage. One can observe that the cracks are more easily discerned in the TSA case (Figure 4.3), compared to the normal noisy case (Figure 4.2). When looking at the different noise cases, it becomes clear that the TSA removes most of the noise from the Gaussian and Laplacian cases, but that there are still some slight remnants of impulsivity in the student-T case. This statement is not be taken at face value however, as the numerical investigations to follow investigate the hybrid model's fault classification accuracy for

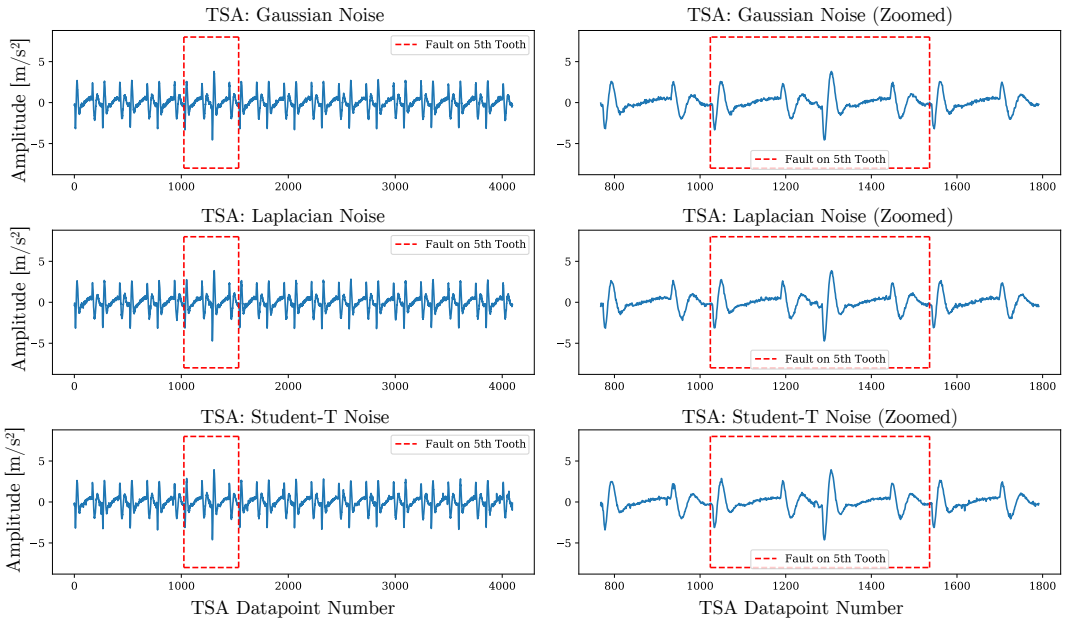


Figure 4.3. Illustration of the resulting signals after 60 rotations' worth of data used for the TSA.

each noise case. This allows one to draw comparisons between the effect of different levels of noise impulsivity on fault classification accuracy of the hybrid model.

4.2 Numerical investigations

With the two datasets constructed, the numerical investigations may be detailed. Firstly, note that two datasets are utilised for the investigations, namely Dataset A-casing_noise (See Figure 4.2) and Dataset A-casing_TSA (See Figure 4.3). Recall there is a strong bias towards healthy data in both datasets, as there are five classes, but half of the data is taken up by this single healthy class. This decision was made intentionally: The hybrid model is rewarded more for classifying a signal as healthy, due to there being more healthy examples relative to any one fault class. This data imbalance serves as a method to bias the network to minimize if not completely remove false positives. False positives in a fault detection scenario could cause unnecessary maintenance and should be avoided.

Dataset A-casing_noise contains 12800 samples, but with $J = 60$ rotations for each sample. This implies that the two datasets are not currently directly comparable, as Dataset A-casing_TSA only has a single rotation (the TSA signal) for every sample. Therefore, to make the datasets comparable, the first rotation from Dataset-A_casing is taken, and the other 59 rotations are discarded. The breakdown of the datasets for the noisy or TSA case may be found in Table 4.1. Note that six datasets are essentially created, three from each noise type being studied

Table 4.1. Breakdown of the data for the three different noise cases, for either the noisy or TSA case.

Class	Gaussian Dataset: No. Samples	Laplacian Dataset: No. Samples	Student-T Dataset: No. Samples
Healthy	6400	6400	6400
Crack	1600	1600	1600
Chip	1600	1600	1600
Spall	1600	1600	1600
Pit	1600	1600	1600

(Gaussian, Laplacian and student-T) and two for the two processing cases (noise only and TSA). Further note

that within each class (except for the healthy class), the samples are further sub-divided into 20 fault intensities, with 1 being healthy and 20 being the maximum fault to be induced. These faults are clearly outlined in Chapter 2.4.3.1.

With the datasets fully defined, the setup of the numerical investigation may receive attention. Six main investigations are conducted (one for each of the six datasets outlined in the previous paragraph), where the hybrid model is fed the synthetic datasets and tasked with minimising the cross-entropy loss. The hybrid model is tasked with identifying the exact class from which a signal is drawn, i.e. fault type classification. Therefore, a multi-class classification problem is being undertaken. The goal of these six experiments is to draw two main conclusions:

- How do different levels of impulsive noise affect the fault classification task of the hybrid model?
- How useful are signal processing techniques as pre-processors for hybrid (or data-driven) models?

Therefore, the following structure is set up: The datasets (outline in Table 4.1) are all randomly split such that 66.7% of the data is utilised for training and the remaining 33.3% for testing. Within the training dataset, 20% of the data is reserved as a validation dataset. The validation dataset serves as a test for when the network starts overfitting the training samples. When this happens, the accuracy between the training set and the validation set diverge. At this point, the training is stopped and the weights from that epoch are taken as the final model.

After some preliminary testing, a dropout rate of 10% seemed to be more than sufficient to regularise the model. Batch sizes of 256 samples are used. Both these values are different from the default parameters given in Chapter 3.1.4 and 3.1.6. As mentioned in Chapter 3.1.6, these values are only default values and can change depending on the experiment and available GPU memory. With some experimentation, these values were shown to work well, and are thus changed from the default recommendations.

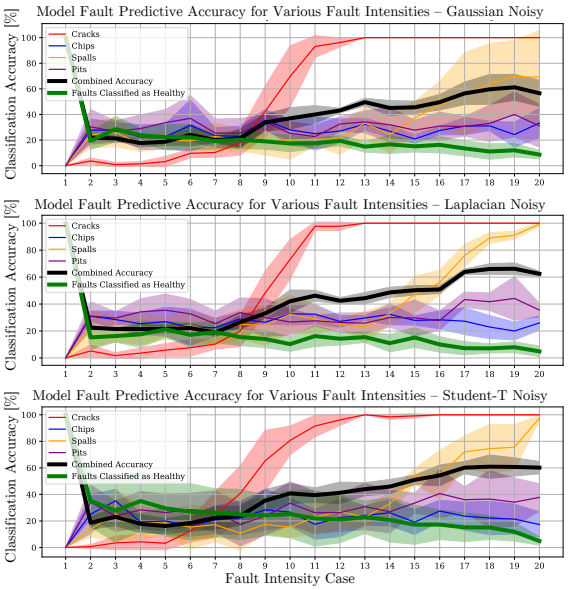
Finally, each classification experiment is run 5 times to generate a more meaningful interpretation of the model's classification capabilities and to try and quantify the uncertainty between experimental runs. This concludes the setup of the experiment, and all that is left is to train the hybrid model to classify different fault types.

4.3 Experimental results

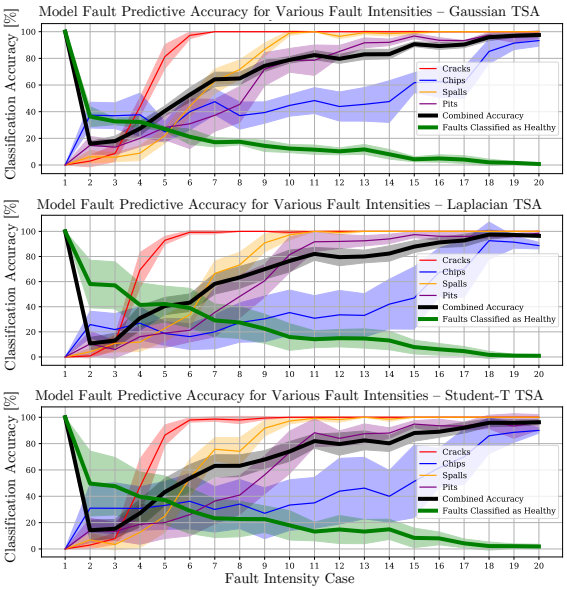
The results of all six experiments are compiled into Figure 4.4. The left half of the figures represent the case where no signal processing was done beforehand (called the noisy case) and the right half of the figures indicate the case where TSA has been applied. Looking down the rows, the top row indicates the results for Gaussian noise, with the middle row indicating the results for Laplacian noise and the bottom row results for student-T noise.

For each figure, six results are being plotted simultaneously. The red, blue, yellow and purple results respectively refer to the model accuracies for the crack, chip, spall and pit type faults at varying intensities. The black result plots the overall model classification accuracy for a given fault intensity. Finally, the green result represents the percentage of faults that are labelled as healthy samples. This result aims to showcase the value of the model as a raw fault detector. Therefore, the lower this result (except for the lowest intensity case, which represents a true healthy case, and thus 100% success is wanted), the better the model will do if it was to be implemented as a simple anomaly detector. That is to say, if the model were only trained to detect the presence of faults, but not label the fault type, this is the curve that results.

The bands around the parameters refer to one standard deviation over the 5 experimental runs. These figures alone do not tell the full story, however, and calibration is needed. Recall that in Chapter 2.4.3.1 a relative fault area (RFA) metric was created (equation (2.113)), which aimed to make results between different fault intensities more comparable. Therefore, the compilation of results in Figure 4.5 shows the corrected results according to relative fault area. As a reminder, the RFA represents the relative area between the fault and the full area of the plane the fault acts upon. As an additional insight into the success of the hybrid model, confusion

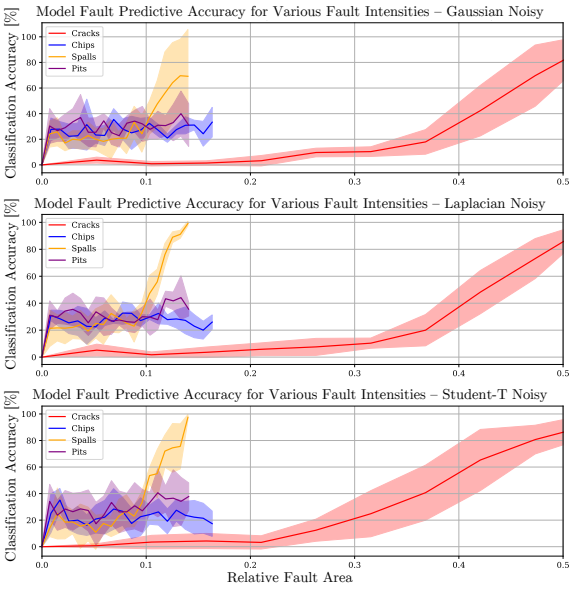


(a) No pre-processing applied (Noisy).

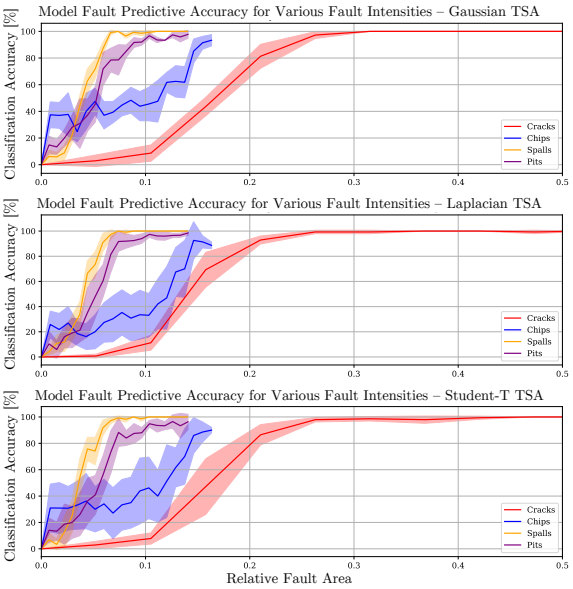


(b) TSA pre-processing applied.

Figure 4.4. Comparison between each fault class' classification accuracy for different fault intensities. The top row shows the results for Gaussian Noise, the middle for Laplacian and the bottom for student-T. One standard deviation of results is given by the transparent bands.



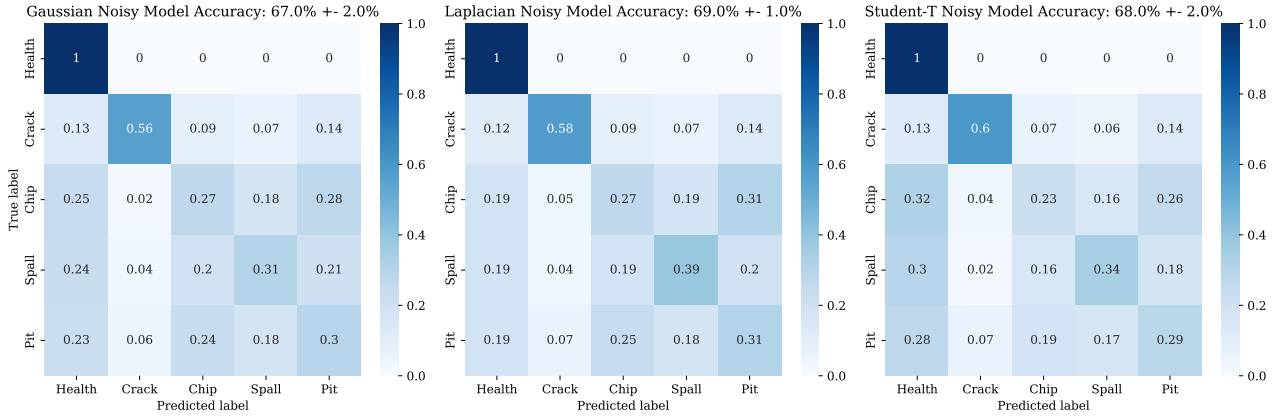
(a) No pre-processing applied (Noisy).



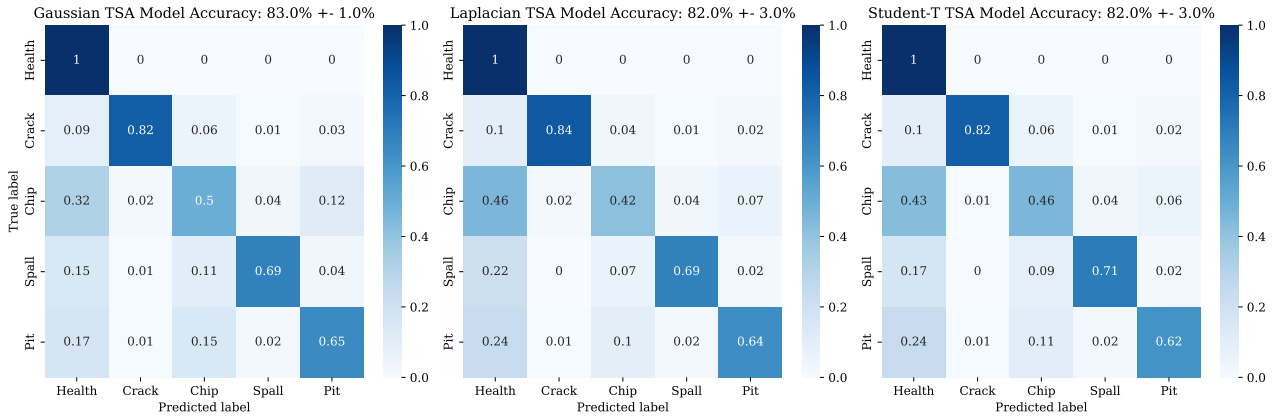
(b) TSA pre-processing applied.

Figure 4.5. Compilation of each fault class' classification accuracy for different relative fault areas. The top row shows the results for Gaussian Noise, the middle for Laplacian and the bottom for student-T. Standard deviations of results are given by the transparent bands. All results above 50% RFA are removed for clarity.

matrices for each experiment are given in Figure 4.6. The shown results are the compiled result over all five



(a) No pre-processing applied (Noisy).



(b) TSA pre-processing applied.

Figure 4.6. Compilation of confusion matrices over the five classes. Each figure shows the confusion matrix over all 5 runs of an experiment compiled into a single matrix. The title of the graphs indicates the noise type, pre-processing method and the overall model classification accuracy, followed by one standard deviation of results.

runs. Therefore, it may be seen that for the healthy class, over all six experiments, not a single sample was misclassified. The following section will discuss the given results.

4.4 Discussion of results

When comparing the results of Figure 4.4 and Figure 4.5, it immediately becomes clear that the experiments without any TSA all suffer severely in classification accuracy. This is most likely because the noise hides most of the faults making classification of faults at low intensities impossible. When looking at the confusion matrices in Figure 4.6, it also becomes clear that having data that has gone through signal processing, results in an immediate classification accuracy increase of about 13-16%. Therefore, it is without a doubt a good idea to apply useful signal processing techniques as pre-processors for hybrid (or data-driven) models.

With the usefulness of TSA being proven, the focus now turns to the set of results that are concerned with TSA approaches. That is to say, no further conclusions will be drawn from the noisy results (Figure 4.4(a), **4.5(a)** and **4.6(a)**), as they are quite poor. Therefore, the discussion to follow is only relevant to the TSA cases.

Firstly, it will prove valuable to look at the fault classification capabilities of the hybrid model for the three noise cases. It is quite clear that there is a strong imbalance between the crack fault case compared to the other fault types when looking at Figure 4.5(b). However, this is a good test for the model to see how it handles data of vastly varying fault intensities. For comparison, a fully broken tooth is present at the 20th crack fault intensity, leading to a 100% RFA value. However, the maximum fault value from any other fault was around 16%. This strong discrepancy should showcase the robustness of the model to not overfit to the crack class, which has really strong fault signals compared to the other three types.

When looking across all three results from Figure 4.5(b), a trend becomes clear: For a given RFA, spalls are the easiest fault type to identify (although they seem more difficult to detect for very small sizes), followed by pits then chips then cracks. These results only hold for the RFA metric, as it is the only metric proposed to somewhat standardise results across faults.

Furthermore, one sees that at about 7% RFA, all spalls were mostly identified correctly, with pits reaching near-perfect results at around 15% RFA. Chips seemed to struggle even more, never really reaching full classification success, but seem to be close to 100% classification success at an RFA of around 18%. Finally, cracks only seem to converge to fully identifiable faults at an RFA of around 27%.

The conclusion to be drawn from this limited experiment suggests that cracks could be the most difficult fault type to recognise at low intensities. This statement is supported by the literature: Wu et al. (2008) applied multiple statistical indicators to simulated crack faults of varying depths. All statistical indicators were shown to be incapable of identifying cracks up to around 40% propagation into the tooth. Therefore, it seems as if cracks at low intensities are difficult to detect.

Spalls are seen to be the easiest fault type to identify. When referring to the stiffness graphs of the fault types (Figure 2.31), one can gather some insight into why this is. The fault types which result in a more sudden stiffness reduction, such as spalls and pits, are the easiest to identify early on. However, the fault types which result in smooth and continuous stiffness reductions, such as chips and cracks, are much harder to identify. This makes physical sense, as the sharper stiffness drops can be associated with larger impulses on the gearbox, leading to stronger identifiability.

When moving along to a noise comparison, the results from Figure 4.4(b) become more useful, as they contain two extra metrics. When looking at the variance bars around the results, it is clearly shown that Gaussian noise results in more consistent model accuracies, whereas the more impulsive Laplacian and student-T results showcase a greater variance in model performance. It was found that the total model accuracy for the Gaussian set was 83% over all classes, with a standard deviation of 1%. The Laplacian and student-T results were tied at a slightly lower 82% accuracy with a larger 3% variance.

This is an important finding, as it suggests the model's accuracy in practice may vary slightly based on the impulsivity of surrounding noise. Overall, however, the current hybrid model seems to be quite robust against noise, when combined with a powerful pre-processor such as TSA.

Besides the slight variability in model accuracy between noise types, it proves informative to look at the green metric, which indicates the percentage of healthy results labelled as healthy (for fault intensity 1) and the percentage of fault cases labelled as healthy (fault intensities 2 through 20). The lower this value, the better, since the rejection of a fault sample (even if it is misclassified) as a healthy case, makes for a strong anomaly detector. It is seen that only around very high fault intensities (18) does the model become a really strong anomaly detector. However, respectable results are already seen around the 10th fault intensity, where around 80% of fault samples are correctly identified as a fault. These results hold over all three noise cases.

A final trend may be seen, which serves as a sanity check: As the fault intensity increases, so too does the identifiability of the fault type and a decrease in the variance of model performance. Therefore, as the faults

become larger, the model simultaneously becomes more aware and certain of their presence. This makes physical sense, as larger faults should inherently be more identifiable, and in turn, should lead to less variance during prediction.

As a final piece of experimental information, the confusion matrices from Figure 4.6(b) prove useful. Firstly, when looking at the number of healthy samples identified as healthy, we see that for all noise types the model is 100% accurate. This is a good result, as it showcases the model's ability to restrict false fault classifications, which as mentioned earlier are undesirable in industry. This naturally comes with the trade-off that fault samples will be misclassified as healthy, but the author believes that this scenario is more acceptable (at lower fault intensities) compared to continually misclassifying healthy samples. When referring back to figure 4.4(b), we see that after fault intensities of size 17, all faults are correctly classified as faults. Therefore, although the model misclassified faults as being healthy, the model fully identified faults when their intensity grew large enough. This is highly desirable behaviour, and shows that the model is capable of identifying a true health state, but also a true fault state, given a large enough fault intensity.

When looking at the number of times a fault was misclassified as healthy, it is seen that chips are the most likely class to be mistaken for a healthy case over all three noise types. This is important from an anomaly detection perspective. If a simple anomaly detector is set up, a chip signal will more likely register as a healthy signal rather than a fault signal. This may therefore suggest that although cracks were seen to be the hardest fault type to identify, from a pure anomaly detection perspective, chips might be the more difficult task.

4.5 Concluding remarks

In conclusion, two new datasets were developed for the numerical investigations of this chapter, namely Dataset A-casing_noise and Dataset A-casing_TSA. These datasets were used to compare the usefulness of signal pre-processing techniques in improving the hybrid model classification accuracy. Furthermore, the datasets were used to identify the effect of impulsive noise types on the classification accuracy of the hybrid model. A few key points were identified from the numerical investigations.

Firstly, TSA was proven to be a valuable pre-processing technique and dramatically improved the performance of the hybrid model. This was because TSA aims to reduce noise within a signal, which in turn increases the identifiability of smaller faults.

Secondly, the hybrid model proved to be quite robust against the more impulsive noise types but did show a slight drop in accuracy, as well as a higher variance for more impulsive noise types. This was not large, however, and therefore the model seems to perform quite well in an environment with impulsive noise, assuming it is paired with TSA.

When comparing the identifiability of faults, cracks and chips proved to be hard to identify at low fault intensities. This is predicted to be a function of their smooth reduction on the gear mesh stiffness. Faults that produce more sudden stiffness reductions were seen to be capable of being identified earlier.

Although the experiments were posed as a multi-class problem, the performance of the model as a simple anomaly detector was also proven to be quite strong. The model seemed to perform well not only as a fault classifier, with accuracies around 83% over all fault intensities but also as an anomaly detector, with 80% of faults being detected at half of their maximum fault intensities.

With all these conclusions, the case has been made that the developed hybrid model, paired with TSA, seems fairly robust against impulsive noise types and has shown to be capable of good fault classification capabilities. Furthermore, the model has proved to be equally capable as an anomaly detector.

This concludes the second novelty of the work, i.e. a hybrid gearbox model, which has been trained on multiple fault types. To the author's knowledge, this type of hybrid model has not yet been developed for a gearbox.

At this stage, a final key question remains: Has the hybrid model truly learnt the underlying fault mechanisms to a strong enough degree that it will fare well in practice? It seems as if much of the literature focuses on developing physical and data-driven models, but do not comment on how exactly they are to be applied to a completely unlabelled industrial dataset. Although the developed hybrid model has been constantly verified against existing works and has now been shown to be robust against a synthetic dataset, no comments can yet be made as to how this model will fare against a completely different domain of data, similar to what is seen in industry. This is the focus of the following chapter.

Chapter 5 Transfer learning: A tool for hybrid model calibration

Up to this point, a hybrid model for gearbox condition monitoring has been developed and shown to be capable of distinguishing different fault types from one another. However, the performance of the hybrid model is only known for the synthetic dataset it was trained on. It is still unknown whether the trained hybrid model will classify faults equally well on a different dataset, ideally an industrial dataset. It is expected that the other factors not taken into account up to this point (temperature, vibrational excitation from nearby machines, motor dynamic effects, dynamic loads etc.) will lead to mentionable differences between the synthetic datasets and a true, industrial dataset. Therefore, a form of calibration between the existing, pre-trained hybrid model and a different dataset (ideally the industrial data) is required. This chapter is dedicated to developing a calibration framework, to finally try and deliver a truly useful and proven model which may be ready to face industrial type datasets.

Transfer learning (TL) is proposed as a strong method of calibration for the hybrid model. Transfer learning tries to utilise knowledge from one domain and transfer it to a different domain. In this case, the "knowledge" refers to the underlying structure between the input data and exit class prediction of the hybrid model. We would therefore like to transfer this knowledge over to a different domain, or dataset, essentially calibrating the pre-trained hybrid model to work on a different domain of data.

A TL framework is generally suited to problems that have two different domains of data: First, one requires a large body of labelled data. This is known as the source domain. The source domain may have originated from experiments or in this case a physical model. The key aspect of the source domain is that it contains labelled data. Secondly, one has a second set of very limited and potentially unlabelled data, known as the target domain. This target domain may be a different dataset, ideally the industrial dataset. Transfer learning tries to find some mapping between the data in the source domain and the data in the target domain, to ensure model performance stays consistent across different datasets.

TL is a relatively new field within machine learning and even newer in the field of condition monitoring. Within transfer learning, various types of transfer learning approaches exist. The approach used in this work is known as domain adaptation (DA) and has been popular in recent works. The problem currently addressed by DA techniques is that within the same NN, one finds two distinct sets of feature mappings in the latent space: one for the source and one for the target domain. This discrepancy between feature distributions arises due to the input data to the network between source and target domains being different.

DA attempts to find a mapping between the latent features within the source and target domains and align them. If one can simultaneously extract features that are meaningful (high classification success in source domain) and also domain invariant (source and target domain latent distributions look similar), one can most likely also achieve a similar classification accuracy within the unlabelled target domain. This is a powerful idea, but one needs to find a viable method to align the domains. The following subsections discuss two existing solutions to the domain alignment problem. These are maximum mean discrepancy convolutional neural networks (MMD-CNN) and deep adversarial neural networks (DANN).

The remaining portion of the chapter is structured as follows: Firstly, a thorough description of the mathematical definitions behind training a standard CNN will be given. This is important, as it eases the explanation for the content to follow. After the training process is outlined, the two DA techniques' architectures and training structures will be given. Finally, numerical experiments will be set up to test the efficacy of these techniques as a calibration mechanism. The results will be discussed and concluding remarks will be made on these techniques.

5.1 Preliminary theory: A formal outline for training a CNN

The task of training the CNN may be described as a loss minimisation problem. Before showing the mathematical constructs to achieve this, refer to Figure 5.1. This figure visually explains which portions of the network are being referenced when speaking of a "feature extractor", "label classifier" and the "latent features".

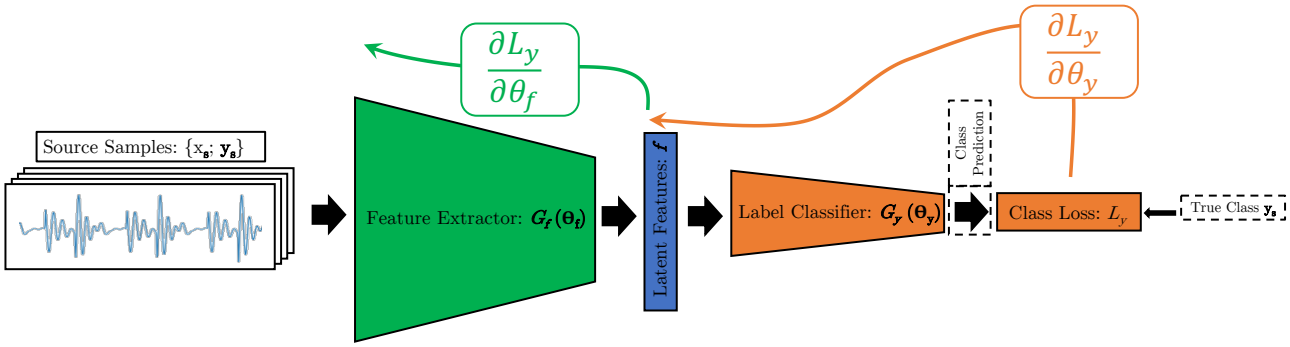


Figure 5.1. CNN structure used to optimise a classification task. Take note of the various portions of the CNN and their respective symbols. A batch of samples $\mathbf{x_s}$ is passed through the feature extractor, at which point it is mapped to the latent space. This latent space representation is then passed through the classifier, which predicts the classes of the original batch of samples. This prediction is compared to the true set of batch labels, $\mathbf{y_s}$ to generate a loss. This loss is then backpropagated back through the classifier (orange arrow) and extractor (green arrow) to update the network weight parameters ($\theta_\mathbf{f}, \theta_\mathbf{y}$).

The classification loss for this work employs cross-entropy loss, as this is suited to a classification problem. The classification loss ($\mathcal{L}_y$) for a batch of samples may therefore be defined as:

$$\mathcal{L}_{y,C}(G_f(G_y(\mathbf{x})),\mathbf{y}) = -\frac{1}{C\cdot N}\sum_{c=1}^{C}\sum_{i=1}^{N} y_{i,c}\cdot\log\left(G_f(G_y(x_i))\right) \tag{5.1}$$

where G_f and G_y refer to the feature extractor and label classifier portions of the CNN. Furthermore, $G_f \mapsto \mathbb{R}^p$ and $G_y \mapsto \mathbb{R}^C$. N, C and p define the number of samples (in a batch), number of classes and number of features in the latent space, respectively. The values of $\mathbf{x}$ and $\mathbf{y}$ represent the full batch of input data and the corresponding one-hot encoded labels. The value of $y_{i,c}$ represents the true label for the ith sample in the batch.

As a special case for this formulation, a binary class cross-entropy is defined. The classification loss ($\mathcal{L}_y$) for a batch of samples from 2 classes only, is defined as:

$$\mathcal{L}_{y,2}(G_f(G_y(\mathbf{x})),\mathbf{y}) = -\frac{1}{2N}\sum_{i=1}^{N}\left[y_i\times\log\left(G_f(G_y(x_i))\right)+(1-y_i)\times\log\left(1-(G_f(G_y(x_i)))\right)\right] \tag{5.2}$$

where $\mathbf{y}$ has now changed from a matrix to a vector that contains the labels but is now coded with a binary value, 0 or 1, instead of being one-hot encoded as [0,1] or [1,0]. This vector is therefore of length N and y_i now simply refers to the ith label.

Therefore, for fault type detection, equation (5.1) may be minimised, and for simple fault detection, equation (5.2) may rather be minimised. To minimise the classification loss, one may note that G_f and G_y are comprised

of parameters θ_f and θ_y respectively. Therefore, the error function to be optimised may be defined as:

$$E(\theta_f, \theta_y) = \mathcal{L}_y(\theta_f, \theta_y) \tag{5.3}$$

where we try to optimise θ_f and θ_y such that optimal parameters $(\widehat{\theta_f}, \widehat{\theta_y})$ can be found:

$$(\widehat{\theta_f}, \widehat{\theta_y}) = \underset{\theta_f, \theta_y}{\operatorname{argmin}}\, E(\theta_f, \theta_y) \tag{5.4}$$

The optimisation algorithm to be used will be chosen on a case to case basis, but will either be steepest gradient descent (SGD) or Adam. For the SGD case, the weight optimisation may be defined as:

$$\theta_{f,i+1} = \theta_{f,i} - \mu \left(\frac{\partial E_i(\theta_f, \theta_y)}{\partial \theta_{f,i}} \right) \tag{5.5}$$

$$\theta_{y,i+1} = \theta_{f,i} - \mu \left(\frac{\partial E_i(\theta_f, \theta_y)}{\partial \theta_{y,i}} \right) \tag{5.6}$$

where μ refers to the learning rate of the network and i refers to the training step.

5.2 Domain adaptation technique 1: Maximum mean discrepancy

The first method of DA to be discussed is that of MMD. As stated earlier, the goal behind DA is to align the source and target domains. This is also known as achieving minimal domain invariance. To minimise the domain invariance, a good approach is to minimise the difference in latent feature distributions for each domain. Therefore, if one has a loss function that can measure the difference between distributions in latent space, one can directly optimise this distance to be a minimum. MMD aims to be such a measure of dissimilarity between distributions.

MMD can be defined as a measure of the distance between feature means. MMD has rigorously been described in other works, and will therefore not be given here, as it interrupts the flow of the work to follow. Instead, the reader is referred to Borgwardt et al. (2006) for any leaps in mathematics made here. In short, for a source domain with data $\mathbf{X_S}$ and a mini-batch of samples $\mathbf{x_s}$ and a target domain with data $\mathbf{X_T}$ with mini-batch of samples $\mathbf{x_t}$, the squared MMD (the squared variant has a property which will be discussed shortly) for a batch of samples is defined as follows:

$$MMD^2(\mathbf{x_s}, \mathbf{x_t}) = ||\mathbb{E}_{\mathbf{x_s} \sim \mathbf{X_S}} \phi(\mathbf{x_s}) - \mathbb{E}_{\mathbf{x_t} \sim \mathbf{X_T}} \phi(\mathbf{x_t})||^2_{\mathcal{H}} \tag{5.7}$$

$$= \mathbb{E}_{\mathbf{x_x}, \mathbf{x_s}' \sim \mathbf{X_S}} k(\mathbf{x_s}, \mathbf{x_s}') + \mathbb{E}_{\mathbf{x_t}, \mathbf{x_t}' \sim \mathbf{X_T}} k(\mathbf{x_t}, \mathbf{x_t}') - 2\mathbb{E}_{\mathbf{x_s} \sim \mathbf{X_S}, \mathbf{x_t} \sim \mathbf{X_T}} k(\mathbf{x_s}, \mathbf{x_t}) \tag{5.8}$$

where $k(x,y) = \langle \phi(x), \phi(y) \rangle$. Here, $k(\cdot)$ refers to a kernel function and $\phi(\cdot)$ refers to a specific representation of the data input into the function. It has been shown for this squared formulation, that if a Gaussian kernel is employed, a property exists that guarantees that if and only if two distributions are identical, the MMD will be 0 (Borgwardt et al. 2006). Therefore, this metric serves as a strong measurement of the distance between feature distributions, and minimising this function during training may lead to domain invariant features. Note that the form in equation (5.8) is inconvenient from a coding perspective. An empirical biased estimate of this value is given in Gretton et al. (2012). Note that the biased version of the estimate is used for coding convenience:

$$\mathrm{MMD}^2(\mathbf{x_s}, \mathbf{x_t}) = \frac{1}{M^2} \sum_{i,j=1}^{M} k(x_{s,i}, x_{s,j}) + \frac{1}{N^2} \sum_{i,j=1}^{N} k(x_{t,i}, x_{t,j}) - \frac{2}{MN} \sum_{i,j=1}^{M,N} k(x_{s,i}, x_{t,j}) \tag{5.9}$$

where M and N refer to the number of samples per batch for the source and target domains respectively. Furthermore, note that the kernel function returns a matrix with rows representing the batch size and columns too representing the batch size. Thus, the i and j indices correspond to the row and column value in the resulting kernel matrix to be selected. In essence, we are simply adding together all the kernel matrix values and dividing by the number of elements in the matrix, or more simply put: We are simply finding the mean kernel value. This ties into the statement made earlier, stating that MMD is essentially a measure of distance between feature means.

Note that the kernel matrix dimensions are comprised of the selected batch sizes. At this point, it is explicitly stated that the source and target domain batch sizes will be made equal. Therefore, the final form of the squared

MMD metric may be written as:

$$\text{MMD}^2(\mathbf{x_s}, \mathbf{x_t}) = \frac{1}{N^2}\sum_{i,j=1}^{N} k(x_{s,i}, x_{s,j}) + \frac{1}{N^2}\sum_{i,j=1}^{N} k(x_{t,i}, x_{t,j}) - \frac{2}{N^2}\sum_{i,j=1}^{N} k(x_{s,i}, x_{t,j}) \tag{5.10}$$

where N is the fixed batch size across both domains.

At this point, a useful loss function has been derived, which is capable of meaningfully measuring domain misalignment. Noting that the goal of DA is to minimise domain invariance, one may therefore manipulate the structure from Figure 5.1, by placing the MMD loss as a secondary loss function directly after the latent feature space. This modification is seen in Figure 5.2

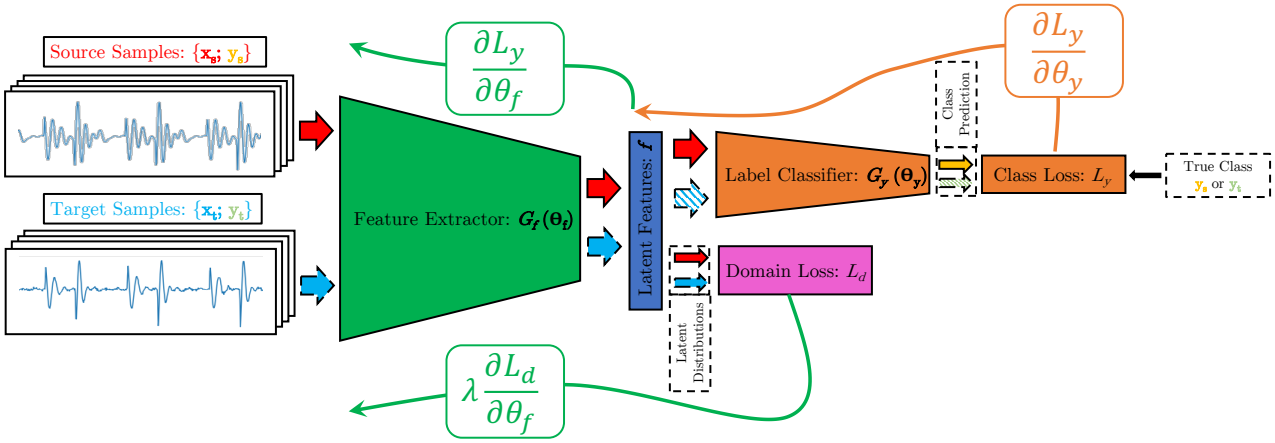


Figure 5.2. Modified CNN structure to incorporate MMD as a domain adaptation technique. Notice that the modified network now has a second loss function L_d, which measures the difference between the latent feature distributions of the two domains of input data. A batch of samples $\mathbf{x_s}$ and $\mathbf{x_t}$ are simultaneously passed through the feature extractor, essentially forming one large batch. This batch of samples is then mapped to the latent space. This latent space representation is then simultaneously passed to two network elements. Firstly, the latent representations are passed through the classifier, which predicts the classes of the original batch of samples. This prediction is compared to the true set of batch labels, $\mathbf{y_s}$ and $\mathbf{y_t}$ (if it exists) to generate a loss. This loss is then backpropagated back through the classifier (orange arrow) and extractor (top green arrow) to update the network weights ($\theta_\mathbf{f}, \theta_\mathbf{y}$). Secondly, the latent representations are directly utilised by the *MMD* loss function, to generate a similarity loss between the source and target batches. This loss is then backpropagated through only the feature extractor (bottom green arrow) to update the extractor weights ($\theta_\mathbf{f}$).

To align the domains, one would like to minimise the variance between the latent feature space for the two domains at hand. Therefore, an MMD loss function is connected to the output of the feature extractor. The domain loss may be defined as:

$$\mathcal{L}_d(\theta_f) = MMD^2(G_f(\mathbf{x_s}, \mathbf{x_t})) \tag{5.11}$$

With this definition, the overall training loss function for the MMD-CNN may be modified as follows:

$$E(\theta_f, \theta_y) = \mathcal{L}_{y,s}(\theta_f, \theta_y) + \mathcal{L}_{y,t}(\theta_f, \theta_y) + \lambda \mathcal{L}_d(\theta_f) \tag{5.12}$$

where s and t refer to the samples from the source and target domain respectively. Note that in a fully unsupervised target domain, no target domain classification loss ($\mathcal{L}_{y,t}(\theta_f, \theta_y)$) component will exist. The λ parameter represents the relative loss contribution between the domain loss and the classification loss. This parameter is simply inserted as a potential aid during training. To minimise the given training loss, we try to optimise θ_f and θ_y such that optimal parameters ($\widehat{\theta_f}, \widehat{\theta_y}$) can be found:

$$(\widehat{\theta_f}, \widehat{\theta_y}) = \underset{\theta_f, \theta_y}{\text{argmin}}\, E(\theta_f, \theta_y) \tag{5.13}$$

Once again the SGD weight optimisation may be defined:

$$\theta_{f,i+1} = \theta_{f,i} - \mu \left(\frac{\partial E_i(\theta_f, \theta_y)}{\partial \theta_{f,i}} \right) \tag{5.14}$$

$$\theta_{y,i+1} = \theta_{y,i} - \mu \left(\frac{\partial E_i(\theta_f, \theta_y)}{\partial \theta_{y,i}} \right) \tag{5.15}$$

Note that if the form of equation (5.12) is observed, the feature extractor parameters (θ_f) are jointly optimised from the classification loss and the domain loss, whereas the label classifier parameters (θ_y) are only optimised from the classification loss. This fully covers the mathematical process behind the MMD-CNN. At this point, it will prove useful to give a more process-based description of how this new network is trained. Before describing the process, however, take note of the forward arrows in Figure 5.2. The solid and dashed outline arrows indicate the flow of samples and features through the MMD-CNN for the source and target domains respectively.

The arrows after the label classifier indicate the flow of prediction labels from the source and target domain. Further note that where arrows are hatched, it indicates that data need not necessarily be passed. This occurs when labels are not present, such as an unsupervised target domain approach. Finally, the longer, backwards-pointing arrows at the top and bottom of the MMD-CNN show the backpropagation route of the different loss functions through the network and represent the optimisation algorithm at work. Now to describe the training procedure:

Firstly, a simultaneous batch of source and target samples are fed to the feature extractor. Due to assumptions mentioned earlier, these batches must be of equal size. Therefore, the input batch consists of N source samples and N target samples to form a full batch of $2N$ samples. The source and target samples are then converted through many convolutional layers until finally, they are output as vectors in a latent feature space. The feature vectors are passed, without labels, to the domain loss function. No labels are required as MMD simply measures the similarity between a set of distributions. The domain loss is calculated and backpropagated back through the feature extractor to optimise for domain invariance.

Simultaneously, only the labelled latent features are passed as input to the label classifier. This will always include source domain samples, but may also include target domain samples where labels exist. Therefore, this framework is fully compatible with an unsupervised, semi-supervised or fully supervised target domain, making for a robust training structure. The classification loss prediction is then derived from the input features and the classification loss is calculated. The loss is then backpropagated through both the label classifier and the feature extractor, to optimise for feature uniqueness.

The aim of this process is that the simultaneous backpropagation through the feature extractor will result in unique, domain invariant features which reach high classification accuracies in both the target and source domains. Note that target labels need not be present for this process. This works because the domain loss function has the sole task of developing domain invariant features, implying that faults that produce one set of feature distributions in the source domain, may also be the same fault in the target domain which produces a similar feature distribution. If this is true, then the label classifier, which was solely trained on source data, will still do a good job at identifying different fault types within the target domain. The optimism behind this statement will be tested in the numerical experiments.

5.3 Domain adaptation technique 2: Adversarial domain adaptation

The second method of DA investigated in this work involves a more adversarial approach. In this approach, instead of directly measuring the distance between two distributions with some loss function, we hand the problem down to a sub-NN which is tasked with finding its own internal representation for measuring domain discrepancy. This sub-NN will be known as the domain classifier G_d, which is the adversary in the title of this method. An illustration of the new architecture may be seen in Figure 5.3.

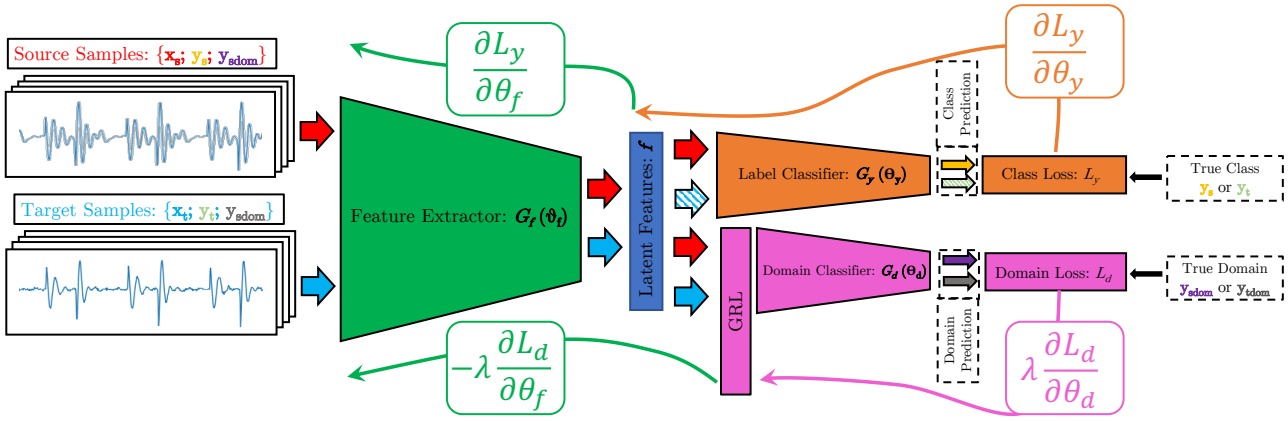


Figure 5.3. Modified CNN structure to incorporate the deep adversarial approach as a domain adaptation technique. Notice that the modified network now has a second classifier, namely G_d, which is the domain classifier and a corresponding loss function L_d, which adversarially tries to determine the domain from which the latent features came. A batch of samples $\mathbf{x_s}$ and $\mathbf{x_t}$ are simultaneously passed through the feature extractor, essentially forming one large batch. This batch of samples is then mapped to the latent space. This latent space representation is then simultaneously passed to two network elements. Firstly, the latent representations are passed through the label classifier, which predicts the classes of the original batch of samples. This prediction is compared to the true set of batch labels, $\mathbf{y_s}$ and $\mathbf{y_t}$ (if it exists) to generate a loss. This loss is then backpropagated back through the label classifier (orange arrow) and extractor (top green arrow) to update the network weights ($\theta_\mathbf{f}$, $\theta_\mathbf{y}$). Secondly, the latent representations are passed to the domain classifier, which predicts which domain the batch of latent features come from. This prediction is compared to the true domain from which the batch of samples came from, y_{sdom} and y_{tdom}, to generate a domain loss. This loss is then backpropagated through the domain classifier (purple arrow) to optimise the domain classifier's weights ($\theta_\mathbf{d}$) towards better domain prediction accuracy. At the gradient reversal layer (GRL), the backpropagation signal is inverted and passed through the feature extractor (bottom green arrow) to update the extractor weights ($\theta_\mathbf{f}$).

The approach outlined in Figure 5.3 is known as a deep adaptation neural network (DANN). The main distinguishing factor between the baseline CNN and DANN, is the domain classifier. The domain classifier receives the latent feature vectors paired with labels indicating the domain from which the features came. The domain classifier is then optimised to discern the domain of origin for a given feature distribution. Simultaneously, we use this optimisation to adversarially train the feature extractor to produce domain invariant features to confuse the domain classifier. This adversarial approach therefore aims to simultaneously train a feature extractor that delivers domain invariant and meaningful features.

The current uncertainty with this approach is defining how the feature extractor should be updated such as to minimise the success of the domain classifier. A brilliant suggestion made by Ganin et al. (2017) (the first work to utilise this adversarial structure), is to employ a gradient reversal layer (GRL). The GRL acts as a normal pass-through layer during forward propagation. However, during backpropagation, the gradients determined by the optimisation process are reversed. At first, this seems counter-intuitive, but upon further investigation, this is a good suggestion:

The backpropagation process of NNs utilises error gradients to update parameters in such a manner as to minimise a certain loss function. However, one can argue that if this gradient update was reversed, the opposite will be achieved, i.e. maximisation of loss. This is where the gradient reversal layer plays a role. Although the gradients within the domain classifier represent the optimal step to minimise the domain classification error, a reversal of this gradient before being backpropagated to the feature extractor would therefore serve to maximise the domain classification error. This gradient reversal layer therefore acts to make the training process an adversarial one, where the first half of gradient backpropagations (in the domain classifier) aim to minimise domain classification error and the second half (in the feature extractor) aim to maximise domain classification error. Paired with the label classifier gradients, the feature extractor is simultaneously optimised for meaningful

and domain invariant features. Notice that this process requires no complicated loss function, rather just domain labels and a simple binary cross-entropy loss function for the domain predictions.

The formal definition of the process outlined above requires two losses that need to be optimised, namely:

$$\mathcal{L}_y(\theta_f, \theta_y) = \mathcal{L}_{y,C}(\theta_f, \theta_y) \tag{5.16}$$

$$\mathcal{L}_d(\theta_f, \theta_d) = \mathcal{L}_{y,2}(\theta_f, \theta_d) \tag{5.17}$$

where θ_d refers to the parameters within the domain classifier. With the loss functions defined, the joint loss function of the network may therefore be defined as:

$$E(\theta_f, \theta_y, \theta_d) = \mathcal{L}_{y,s}(\theta_f, \theta_y) + \mathcal{L}_{y,t}(\theta_f, \theta_y) - \lambda \left[\mathcal{L}_{d,s}(\theta_f, \theta_d) + \mathcal{L}_{d,t}(\theta_f, \theta_d)\right] \tag{5.18}$$

where λ once again serves as a parameter that may be used to define the relative contribution between the domain classifier and the label classifier, and need not be a constant value throughout the optimisation process. Ganin et al. (2017) dynamically increase this value throughout optimisation. Note that the loss function is reduced by the domain loss terms ($\mathcal{L}_d(\theta_f, \theta_d)$), which is a direct result of the GRL. Also note that for the case of an unsupervised target domain, the target domain classification loss ($\mathcal{L}_{y,t}(\theta_f, \theta_y)$) will not exist. The domain classification loss ($\mathcal{L}_{d,t}(\theta_f, \theta_d)$) must however still exist, as the labels from the domains are known. Finally, the optimisation goal of the adversarial setup may be defined as follows:

We try to optimise θ_f, θ_d and θ_y such that optimal parameters ($\widehat{\theta_f}, \widehat{\theta_y}, \widehat{\theta_d}$) can be found through a min-max problem:

$$(\widehat{\theta_f}, \widehat{\theta_y}) = \underset{\theta_f, \theta_y}{\operatorname{argmin}}\, E(\theta_f, \theta_y, \widehat{\theta_d}) \tag{5.19}$$

$$\widehat{\theta_d} = \underset{\theta_d}{\operatorname{argmax}}\, E(\widehat{\theta_f}, \widehat{\theta_y}, \theta_d) \tag{5.20}$$

Note that this formulation requires a saddle point to be reached, as it is an adversarially posed min-max problem. The domain classifier tries to maximise the loss function due to the way it is set up in equation (5.18), with a reversed sign. Maximisation of the loss function will therefore result in a stronger domain classifier, which will serve as a stronger adversary. However, we simultaneously try to minimise the loss function using the feature extractor and label classifier, which tries to overcome the adversary. Therefore, in error space, the optimisation is trying to reach a saddle point which results in the strongest adversarial component possible for the minimum classification error possible. Therefore, once a saddle point is reached, the network may be said to be optimised.

Note the elegance of this method: Usually generative adversarial networks (GANs) are employed to reach such a min-max optimisation problem, whereas this adversarial approach is somewhat simpler and easier to implement. Finally, for the SGD case, the gradient updates may be given as:

$$\theta_{f,i+1} = \theta_{f,i} - \mu \left(\frac{\partial E_i(\theta_f, \theta_y, \theta_d)}{\partial \theta_{f,i}}\right) \tag{5.21}$$

$$\theta_{y,i+1} = \theta_{f,i} - \mu \left(\frac{\partial E_i(\theta_f, \theta_y, \theta_d)}{\partial \theta_{y,i}}\right) \tag{5.22}$$

$$\theta_{d,i+1} = \theta_{f,i} - \mu \left(\frac{\partial E_i(\theta_f, \theta_y, \theta_d)}{\partial \theta_{d,i}}\right) \tag{5.23}$$

At this point, the mathematical description of the optimisation problem is fully described, and it will prove useful to give a general overview of the training process involved in the adversarial approach.

Referring to Figure 5.3, note once again that solid and dashed arrows are present, representing the source and target domain data flow for forward propagation, respectively. The arrows exiting the label classifier represent the predicted class label for the source and target domain samples. The arrows exiting the domain classifier represent the predicted domain label for the source and target domain samples. After the feature extractor, the target domain's arrows are hatched. This indicates that in the case of an unsupervised target domain, this data

flow does not occur and can be neglected. With this information, the process required to implement DANN may be described in full:

Firstly, a simultaneous batch of source and target samples are fed to the feature extractor. Due to assumptions mentioned earlier, these batches must be of equal size. Therefore, the input batch consists of N source samples and N target samples to form a full batch of $2N$ samples. The source and target signals are then converted through many convolutional layers until finally, they are output as a latent feature space. The feature vectors are passed, with their domain labels, to the domain classifier. The domain classifier then further tries to predict with a single output the domain from which the latent feature sample has come. The domain loss is then calculated and firstly backpropagated through the domain classifier to optimise domain classification. However, just before reaching the latent feature layer, the gradients are reversed and backpropagated through the feature extractor to optimise for domain invariance. Note that the domain classifier must be optimised to more easily predict the domain of origin for features, as any adversarial approach relies on a strong adversary to generate equally strong features to fool the adversary.

Simultaneously, only the labelled latent features are passed as input to the label classifier. This will always include source domain samples, but may also include target domain samples where labels exist. Therefore, this framework is fully compatible with an unsupervised, semi-supervised or fully supervised target domain, making for another robust training structure. The classification loss prediction is then derived from the input features and the classification loss is calculated. The loss is backpropagated through both the label classifier and the feature extractor, to optimise for feature uniqueness.

The expectation for DANN, is that the simultaneous backpropagation through the feature extractor will result in unique, domain invariant features which will reach high classification accuracies in both the target and source domains. In the unsupervised target domain case, no classification labels are required. It is expected that the adversarial domain classifier will optimise for domain invariant features, such that if the label classifier reaches high accuracies on the labelled source domain, the same should be true for the target domain, as the latent features spaces are as close to identical as possible. The validity of this claim will be tested in the next section.

5.4 Numerical experiment: Transfer learning as a calibration tool

The overarching goal throughout this work has been the development of a hybrid model for gearbox condition monitoring. Chapter 4 showed that the hybrid model is perfectly capable of detecting different fault types at various intensities on different teeth of a pinion gear. However, this success is constrained to the synthetic dataset on which the hybrid model was trained. Currently, we have no indication as to how a pre-trained hybrid model will perform on a different dataset, such as for a real industrial gearbox. It is expected that a discrepancy will exist between the synthetic data and real data, as discussed in the introduction to this chapter. Therefore, transfer learning is proposed as a tool to calibrate the hybrid model, minimising the differences between a synthetic dataset and a real dataset. Before real data is used, however, it may be advantageous to test the calibration methods in a pure numerical environment, where external variables may be controlled. If the numerical experiments show promise, we may move along to industrial data [1].

Therefore, this chapter sets out to set up numerical experiments with two main goals:

1. To determine whether a pre-trained hybrid model can be applied directly to a different (but related) dataset, and maintain high accuracy.
2. To determine whether transfer learning may be used to calibrate the differences between two different (but related) datasets, resulting in a higher fault detection accuracy when compared to an uncalibrated hybrid model.

[1] For this work, no industrial data was available and therefore experiments with real data is left as future work

To truly understand the efficacy of using transfer learning for calibration, it is decided to develop numerical experiments, as these provide a controlled environment with which one can gain a deeper understanding of the advantages and shortcomings of the proposed calibration technique. Therefore, the entirety of this chapter utilises synthetic data which is generated in a controlled numerical environment, with known labels.

A summary of the structure of the numerical investigations is given in Figure 5.4. The remainder of this section aims to explain in greater depth the contents of this figure.

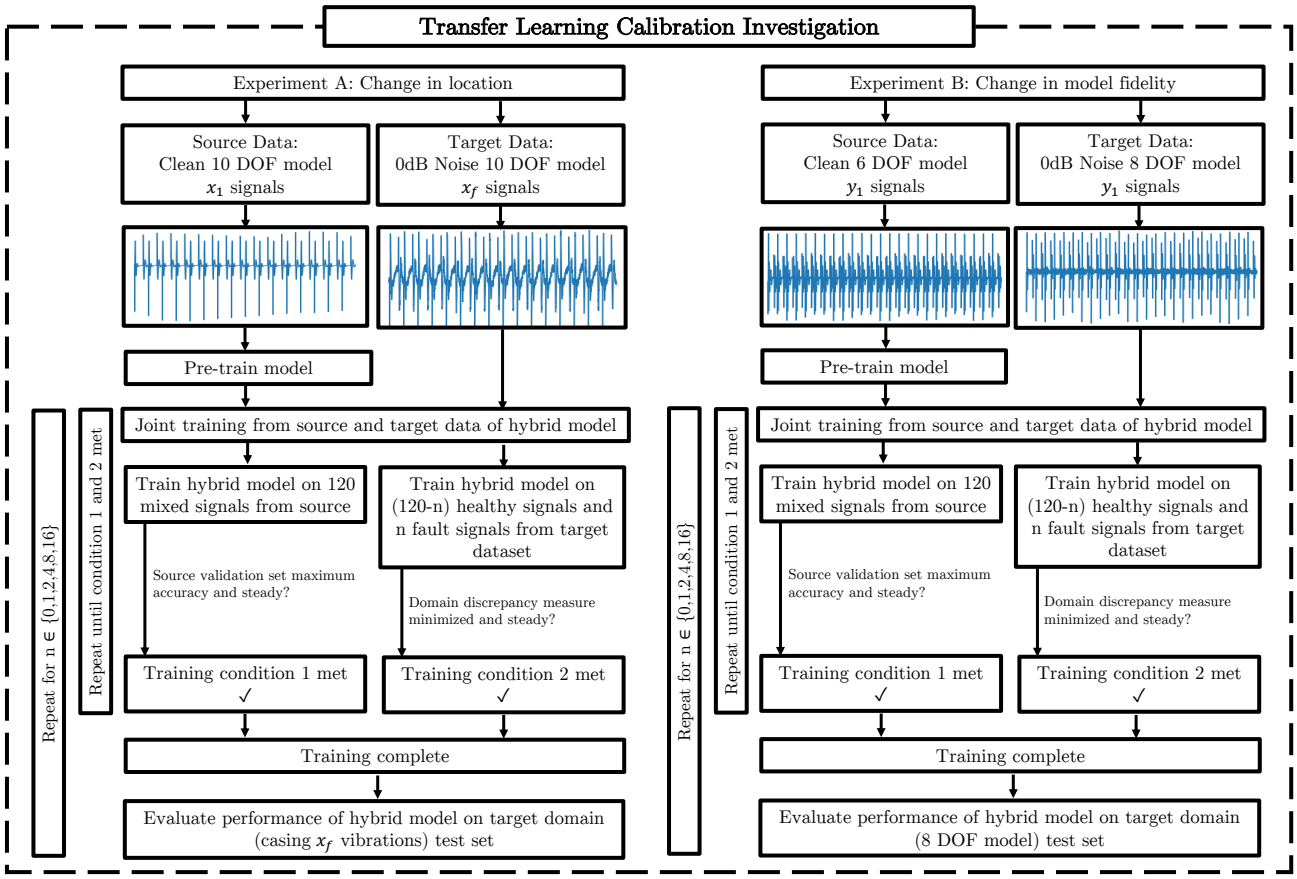


Figure 5.4. Basic outline of the numerical investigations for this section. Take note that the full set of experiments are repeated for each model type. That is to say, for any given model (MMD-CNN, DANN, Baseline CNN) with or without labels (supervised and unsupervised), this framework will be run.

At this point, three main hybrid models have been developed. The first hybrid model is the strong baseline CNN, which is outlined in great detail in Chapter 4. Although the justification for using a CNN in this work is given in sub-section 3.1.1, this work is not constrained by a CNN model and may be extended to any DNN model. The baseline CNN does not employ transfer learning in its architecture, and can therefore be used as a good control against which findings can be compared. This is a strong baseline, as Chapter 4 showed the strong fault classification and fault detection capabilities of the baseline CNN. This model may further be used to address the first goal of the experiments.

The second model utilises the same fundamental structure as the baseline CNN, but incorporates an MMD loss function to create a transfer learning approach called MMD-CNN. Finally, the baseline CNN was once again adapted with an adversarial component to deliver a deep adaptation neural network (DANN). These models may be compared against the baseline CNN to address the second goal of the experiments.

Note that these experiments are exceptionally difficult: Much of the works in literature simply try TIM (transfer in the identical machine), where simple transfers on the same asset with similar vibration patterns are done, with only differing load cases. These approaches assume that labelled data already exists for the asset of interest. From the outset of this work, it has been reiterated that the assumption for this work is that labelled data is scarce for gearboxes. Therefore, although these methods have been shown to work well in theory, they rely on an assumption that is misaligned with this work.

This work is investigating the much harder case of TDM (transfer across different machines), which has the task of transferring knowledge between two completely different (but similar in nature) machines. It can be stated that TDM is being applied, as the goal of this work is to develop a hybrid model which may be pre-trained on synthetic data (machine 1), and has to be applied to a real gearbox (machine 2). The complexity of the TDM approach for this work lies within the assumption that the real gearbox data will have very few or no labels at all.

The first subsection of this chapter outlines the synthetic datasets which will be used to test the three models. Then, the experimental setups for each dataset are explained in detail, followed by the results. Finally, the results are discussed and the validity of domain adaptation techniques for transfer to industry are commented upon.

5.4.1 Datasets: An overview

The two datasets which will be used, originate from Chapter 3, namely Dataset A and B. Both of these datasets are identical in format and have been extensively explained in their respective sections. A quick overview of the properties for both datasets is however given:

- The datasets consist of four fault types, namely cracks, chips, spalls and pits.
- The datasets consist of twenty fault intensities for each fault type.
- The datasets contain each fault and intensity for each possible tooth on the pinion gear.
- Dataset A is split into two sub-datasets, both generated from a 10 DOF dynamic gearbox model based on Luo et al. (2019*b*):
  - Dataset A-pinion is generated from the horizontal vibrations on the pinion gear of the gearbox.
  - Dataset A-casing is generated from the horizontal vibrations of the gearbox casing.
- Dataset B is also split into two sub-datasets. Both datasets utilise the vertical pinion acceleration:
  - Dataset B-6 is generated from the six DOF dynamic gearbox model based on Meng et al. (2020).
  - Dataset B-8 is generated from the eight DOF dynamic gearbox model based on Meng et al. (2020).

Starting from the aforementioned properties, the datasets are further manipulated to have the following properties for this investigation:

- The datasets are minimised to only have a single unique vibration signal per fault type, intensity and location. That is to say for a given tooth on the pinion gear, there exists only one sample for a given fault type and intensity. This is a major reduction in data, as the default datasets have five duplications. The decision to remove all duplications are twofold. Firstly, this drastically speeds up the rate of experimentation, as the dataset is instantly cut into a fifth of its original size. Secondly, and more importantly, this limited data environment is used to simulate the limited labelled data seen in industry. It therefore tests the robustness of the models, with very limited data being available to learn from.
- For each fault type, with a given intensity, for a given tooth, a healthy sample is generated. On top of this, an additional set of healthy samples are added to the dataset, to bias the dataset slightly towards healthy data. Therefore, slightly more than 50% of the dataset consists of healthy examples. The task of not misclassifying a healthy sample is set as a high priority. Misclassifying a healthy sample is undesirable for a real gearbox, as this may lead to the gearbox prematurely being maintained.

Between Dataset A and Dataset B, only two main differences are observed: Firstly, the number of samples. Dataset A has fewer samples, as the model from which it is generated has fewer pinion gear teeth, implying fewer variations in possible fault locations. Secondly, another difference is the exact load and speed conditions under which each dataset is generated. For this chapter, the following load and speed cases from Dataset A and Dataset B are selected:

- Dataset A: 7 Nm load at a 25.2Hz rotation speed.
- Dataset B: 20 Nm load at a 40Hz rotation speed.

As a final note, recall that Dataset A and B are further split up into two separate sub-datasets. The main reason for creating the sub-datasets from Chapter 3 will become clear shortly.

When drawing from the results from the baseline CNN in Chapter 4, one notices that very low fault intensities are difficult to classify. Since the goal behind this chapter's investigations is to validate the use of transfer learning as a calibration tool, this variable needs to be removed. If not, poor results may not necessarily be attributed to a single methodology (calibration vs. no calibration), but rather to the simple fact that "unclassifiable" samples are present in the datasets. Therefore, the following simplifications to Dataset A and B are proposed:

In Chapter 4, it was seen that the highest fault intensity cases for all fault types were accurately classified. Therefore, the experiments to follow will only use the three top fault intensities within Dataset A and B. That is to say for all fault types, the 18th, 19th and 20th fault intensity cases may be used for training, and any lower intensity discarded for the sake of the experiment. As a slight change to this, it was also seen from Chapter 4, that the crack fault types were vastly larger in their relative fault areas, whereas the other three fault types were fairly similar. Therefore, to make the dataset more standardised, for only the crack fault type, instead of selecting the 18th-20th fault intensities, the 10th to 13th fault intensities were rather selected. With these nuances explained, a finer explanation of each dataset is given.

5.4.1.1 Dataset A: 10 DOF model with a change in measurement location

The breakdown of Dataset A-pinion and A-casing after applying the reductions as mentioned in the introduction to this section may be seen in Table 5.1. Note the very limited data being presented for each fault case. Only 48 samples for each fault are available; that is three intensities over 16 possible tooth locations.

Table 5.1. Breakdown of Dataset A

Class	Dataset A - Pinion: No. Samples	Dataset A - Casing: No. Samples
Healthy	240	240
Crack	48	48
Chip	48	48
Spall	48	48
Pit	48	48
Total	432	432

5.4.1.2 Dataset B: Six and eight DOF models

The breakdown of Dataset B-6 and B-8 after applying the reductions as mentioned in the introduction to this section may be seen in Table 5.2. Note once more the very limited data. The number of samples is slightly higher than before, as this dataset involves 25 possible tooth locations over three intensities.

5.4.2 Experimental setup

The purpose of the experiments is to try and mimic industrial problems as closely as possible with only the available synthetic data. Furthermore, the experiments try to determine the usefulness of transfer learning techniques for calibrating discrepancies in data. The first experiment deals with the issues of sensor placement, and the second experiment with the issue of model complexity. The first experiment may be seen as a data-oriented problem and the second experiment a physics-focused problem.

Table 5.2. Breakdown of Dataset B

Class	Dataset B - 6 DOF: No. Samples	Dataset B - 8 DOF: No. Samples
Healthy	375	375
Crack	75	75
Chip	75	75
Spall	75	75
Pit	75	75
Total	675	675

For both experiments, the transfer learning methods are compared to the baseline CNN. Therefore, to make results truly comparable and consistent across both experiments, a general framework of experimentation is set up. The following framework holds for both experiments:

The source domain data contains the full range of fault types, intensities and locations for the training and testing dataset. The target domain, however, by default contains only healthy samples for the training set. The testing set, however, has the full range of fault types, intensities and locations available. This strict limitation on target domain training data is done to ensure the assumption of limited to no labelled data in the target domain is realised. Recall that ideally, the target domain can be set as real industrial gearbox signals. When historical data for an asset is available, this might only include healthy vibration measurements, as machines are generally maintained before they reach a deteriorated state. It is however possible that a few fault examples may be contained within the captured dataset. For this reason, the target domain training dataset is allowed to contain varying amounts of fault examples to simulate this fact. This is however the fundamental parameter to be varied in the experiments and will be explained shortly. For now, a quick focus on training set sizes is informative.

Due to the expected limited (target) data for the target domain, we limit the target domain training dataset to 120 samples only. As stated before, these training examples are by default 120 healthy examples only, unless the specific experiment includes fault examples into the training dataset. Where these fault samples are included, healthy samples are replaced, such that the size of the dataset always stays fixed at 120 samples.

For all investigations, the decision is made to test the anomaly detection capabilities of uncalibrated and calibrated hybrid approaches. That is to say, the hybrid model is tasked with identifying whether a fault exists, rather than being tasked with specifying the exact fault type. The motivation for this is as follows: Identifying an anomaly is a binary classification problem. However, identifying a specific fault type is a multi-class classification problem. These experiments aim to investigate the effect of calibrating the hybrid model, not necessarily predicting a fault type and therefore the simpler binary classification problem is chosen. It is sensible to start with the simpler binary classification problem and build up to a multi-class classification problem, were it deemed necessary. Therefore, although different fault types and intensities are generated, they are all labelled under the same fault label.

At this point, the two types of approaches need to be evaluated, namely uncalibrated and calibrated approaches. Firstly, among the uncalibrated approaches, two types of training are undertaken:

1. One baseline CNN is trained only on the labelled source domain training dataset, which represents the current state-of-the-art for hybrid models: An assumption that a model trained on synthetic data can perform well on a different dataset. This may be seen as evaluating the value of the physical modelling portion of a hybrid model, as only synthetic data is being used for training the model, with no information from the target domain being incorporated. This method may be thought of as an unsupervised approach, as no information from the target domain is utilised, making the predictions for the target domain completely unsupervised.

2. Another baseline CNN is trained on the very limited target domain training dataset (120 examples), and not shown source data at all. This case aims to represent a purely data-driven approach to fault detection with a deep NN. Note that even though some fault examples may be present in the target domain training dataset, this network relies on labelled examples, making this approach quite optimistic. That is to say, that even if this approach works well, it still has the strong assumption that the target domain was already labelled in the first place.

Both these cases are trained using an Adam optimiser (with Tensorflow defaults). This is chosen over SGD, as pre-experimentation showed quicker convergence times for Adam. The author tested multiple optimisers and no real accuracy difference was observed. Therefore the only meaningful difference is training speed.

Batch sizes are fixed to 40 samples per batch, per domain (i.e. 80 samples in total), as this shows a good balance between RAM management and training speed. The source-only method receives the full source domain training set, whereas the target-only method receives 120 labelled target domain examples, with different numbers of faults present in the dataset depending on experiment type. The training is stopped once a small validation dataset (10% of the training dataset) stops climbing in classification accuracy. Furthermore, from preliminary testing, a dropout rate of 0.5% was shown to work well for all experiments, and therefore this rate is used.

For the calibrated approaches (transfer learning approaches), the following training scenarios are undertaken:

1. The MMD-CNN and DANN training datasets are set up with 120 random samples from the fully labelled source training dataset, and 120 samples from a completely unlabelled and limited target training dataset. This is called the unsupervised training scenario and is the most important one: This scenario does not require labels for the target domain. This method may directly be compared to scenario 1 in the uncalibrated approaches above. Note that the source domain is also limited to 120 samples to ensure both the source and target domains have the same number of samples.
2. The MMD-CNN and DANN training datasets are set up with 120 random samples from the fully labelled source training dataset, and 120 fully labelled (although limited) samples from the target training dataset. This is called the supervised training scenario. This scenario represents a stronger assumption: Labels are available for the target domain. As already stated, this is a strong, potentially unrealistic assumption. This method may directly be compared to scenario 2 in the non-transfer learning scenario. Note once again the source domain is limited to 120 samples, to match the number of target domain samples.

Following preliminary experiments, for both supervised and unsupervised cases, SGD will be used, as it proved to be the most stable and reliable option across many other tested optimisers. The learning rate is fixed at 0.003. Once again, batch sizes of 40 samples from each domain are used. Thus, the network simultaneously receives 40 source and 40 target samples to create a total batch size of 80. Also, as mentioned before, 120 source and 120 target samples will be made available for training. Notice that this is quite a small number of the total available dataset, and tries to mimic the issue of limited industry data.

For both supervised and unsupervised approaches, the networks are pre-trained with data from the full source domain dataset (more than 120 samples) as if they were standard hybrid models. From preliminary experimentation, this pre-training proved to play a key preparatory role before the transfer learning took place. If this step was neglected, the networks would more easily diverge and reach either one extreme of high source domain classification accuracy, but with no domain invariance, or another extreme of complete domain invariance, but with no source or target domain classification accuracy. Thus, this pre-training may be seen as a priming step to get the model into a local minimum in error space.

The priming step for all calibrated approaches is fixed at 100 epochs. After priming, the networks are trained simultaneously on the new joint batch of 40 samples from each domain, which are now sampled from the limited 120 samples in each domain. The transfer learning techniques need to be eased into the training process,

otherwise, their domain alignment effects may be too strong early in training, leading to low classification accuracy. Therefore, a growing λ value is employed to increase the relative loss contribution of the transfer learning techniques. This value starts at 0 (thus no domain alignment) and ends at 1 (equal contribution to label classification loss). The pattern used to increment the domain alignment contribution is to increase the λ value by 0.05 every 25 epochs of training. Therefore, a minimum of 500 ($25/0.05 = 500$) epochs is run before stoppage conditions may be implemented.

The training is stopped when two criteria are met. Firstly, the source validation accuracy must be high and stable. Then, if this has been ensured, the training is only stopped once the domain discrepancy measure has stabilised. If the domain discrepancy measure is still moving towards an optimum, whilst the source validation accuracy is high, the network is allowed to continue. If at some point the source validation accuracy starts dropping, the training is stopped. This will indicate the point that the domain discrepancy measure is moving the weights into a general but not meaningful representation, as the label classification accuracy is starting to suffer. For DANN, ideally one would like a domain classification accuracy of 50% (random guessing) and for MMD one would like a domain similarity score of 0. Therefore, these are the optimum goals to try and achieve, whilst keeping source validation high.

Finally, for both experiments, a dependent and independent variable is chosen to experiment upon. The dependent variable is set as the classification accuracy of each training case. The independent variable is set as the number of fault samples in the target domain, starting from 0 and ending at 16, with increments in powers of two. Thus, the effect of increasing the number of labelled target domain fault samples is investigated against the classification accuracies of uncalibrated and calibrated frameworks. Note that the set of 0 to 16 fault samples that are made available are completely random. Therefore, for example, in the case where a single fault is introduced, this single fault may be a crack for one experimental run, but then a chip in the next and so forth. Therefore, multiple runs are required to truly gauge the usefulness of the calibration techniques. Therefore, these experiments are re-run five times each to gain confidence in the models' true performance.

This concludes the general framework that both datasets' experiments must follow. The finer nuances for each dataset are explained in the following sections. A summary of the key experimental framework is given in Table 5.3.

Table 5.3. Summarised training setups for uncalibrated and calibrated approaches.

Uncalibrated Cases:	
Experimental Parameter	**Experimental Setting**
Batch Size	40
Dropout Rate	0.5%
Optimiser (Learning Rate)	Adam (Default)
End of Training Criterion	Validation Loss Maximum
Target Domain Fault Samples Available	[0,1,2,4,8,16]
Number of Experimental Repetitions	5
Calibrated Cases:	
Experimental Parameter	**Experimental Setting**
Batch Size	40/40 (Source/Target)
Dropout Rate	0.5%
Optimiser (Learning Rate)	SGD (0.003)
λ Parameter	Growing from 0 to 1 with 0.05 increase each 25 epochs
End of Training Criterion	Source Validation Loss Maximum Domain Discrepancy Measure Minimum
Target Domain Fault Samples Available	[0,1,2,4,8,16]
Number of Experimental Repetitions	5

5.4.2.1 Experiment A: Difference in vibration location - Dataset A

The first numerical experiment is formulated around the placement of sensors. It is common practice in the literature to utilise the pinion gear vibrations for analyses. This assumption, however, may not be realised in practice. Often access to the pinion gear is difficult. Therefore, it is more likely that an accelerometer will be placed on the gearbox casing. However, the difference in measurement location may have mentionable effects on the measured vibration signal. Therefore, this experiment aims to utilise Dataset A to investigate how well the baseline CNN performs if the sensor location is changed. Furthermore, the effect of calibrating the CNN is analysed by employing MMD-CNN and DANN. Recall that the transfer learning approaches (calibrated approaches) discussed in this work require a source and a target domain. Therefore the following setup is proposed:

The source domain is set as the horizontal vibrations directly synthesized on the pinion gear (Dataset A-Pinion). Dataset A-Pinion represents the many works that try to develop models based on pinion gear vibrations. The target domain is set as the horizontal vibrations synthesized on the gearbox casing of the 10 DOF model (Dataset A-Casing). Dataset A-Casing represents the types of measurement a true sensor would take in practice. This difference in data due to sensor placement can be seen as a data-focused problem, i.e. where the input data may be vastly different due to the sensor placement in practice compared to an experimental setup.

As a further measure to try and align the experiments with a more realistic industrial transfer task, the source domain is generated free of noise, with only the vibrations based on the raw physical model being used. The idea behind this is that all the necessary underlying physics behind a fault may be built into the raw, but noise-free vibration signal. Therefore, the aim is that the network trained on the source domain will be capable of inferring the true fault mechanisms to make correct classifications. The target domain, however, will be infused with Laplacian noise to the extent that the SNR value becomes 1, i.e. the power of the underlying signal and the noise is equal. This aims to simulate one of many factors present in industrial signals, where the underlying physics may be hidden by noise. Therefore, the task at hand is not only to create a general model which classifies well on both the source and target domains but can further generalise to more noisy cases.

5.4.2.2 Experiment B: Difference in model complexity- Dataset B

The second numerical experiment is formulated around the design complexity of physical models, which in turn relates directly to the hybrid model. When creating a physical model, we create a low fidelity estimate of the true asset. This implies that the data generated from a physical model may be different from the real asset, as the physical model cannot always capture all the necessary complex interactions of a real asset. Therefore, a discrepancy between the data generated from a physical model and the real asset may be expected. At this point, it is unknown whether the hybrid model can overcome such discrepancies in the data and needs to be investigated. Therefore, this experiment aims to utilise Dataset B to investigate how well the baseline CNN performs if the number of DOFs is changed between a source and target domain. Furthermore, the effect of calibrating the CNN is analysed by employing MMD-CNN and DANN. Recall that the transfer learning approaches (calibrated approaches) discussed in this work require a source and a target domain. Therefore the following setup is proposed:

The source domain is set as the vertical vibrations on a pinion gear of a lower-dimensional gearbox model (Dataset B-6). Dataset B-6 represents the lower dimensional physical model that the hybrid model may be trained on before being employed to a larger DOF reality (industry). The target domain is also set as the vertical vibrations of the pinion gear, but this time on a higher-dimensional model (Dataset B-8). Dataset B-8 represents the higher dimensional "true asset" on which the model will need to be employed. This difference in DOF between modelling and implementation can be seen as a more physics-focused problem, i.e. the discrepancy between source domain and target domain results are purely due to the physical modelling process.

For the same reasons as given before, the source domain is left noise-free and the target domain is infused with Laplacian noise to result in an SNR of 1.

Admittedly it is far-fetched to expect an eight DOF model to represent a real industrial asset over a six DOF model. However, the purpose of the experiment is not to accurately model a true asset with even higher DOFs. The main purpose is to investigate whether even a small change in DOFs is enough to reduce the classification accuracy of a hybrid model pre-trained on a dataset that is generated with only six DOFs. The aim is that calibration will eliminate any discrepancies in results if any exist in the first place. The following section shows the results for each of the experiments outlined here.

5.4.3 Experimental results

This section showcases the experimental results. The accuracies given in these results refer to the target domain test set accuracies. To further gain insight into the training of the calibrated approaches (MMD-CNN and DANN), a second set of results are given. These results only utilise the training runs from experiment A to draw some interesting conclusions. For these results, a single run with four faults in the target domain is investigated. Finally, a third and final set of results are given which showcase the latent feature space. The goal behind this is to visualise whether the calibrated models are creating domain invariant features.

With regards to the results which are shown shortly, it must be noted that the target domain test set comprises of all the samples outlined in Table 5.1 and Table 5.2 minus the sampled 120 values used for training. Therefore, there is no data leakage in any of the results.

5.4.3.1 Summarised experimental results

The results from Experiment A may be found in Table 5.4, which provides insight into the pure performance of each network. As a visual aid in determining trends more easily, the same results are plotted in Figure 5.5. Similarly, the results from Experiment B may be found in Table 5.5, with the visual equivalent found in Figure 5.6.

Table 5.4. Results for Experiment A: Investigation into the classification accuracy of calibrated and uncalibrated models when attempting to change the data from one sensor location to another. Maximum unsupervised accuracy for each fault case is indicated with blue font and maximum supervised accuracy for each fault case is indicated in purple font. The maximum classification accuracy regardless of unsupervised or supervised is further expressed in **bold**.

Number of Fault Samples in Training Set	Classification Accuracy on Target Domain Test Set [%]					
	Source Training Only (Unsupervised)	DANN (Unsupervised)	MMD-CNN (Unsupervised)	Target Training Only (Supervised)	DANN (Supervised)	MMD-CNN (Supervised)
0	51.73 ± 0.0	74.02 ± 12	80.92 ± 5.0	48.28 ± 0.0	84.14 ± 2.7	**88.51 ± 1.6**
1	51.73 ± 0.0	87.13 ± 5.7	**92.73 ± 3.8**	48.74 ± 0.6	87.13 ± 4.7	89.20 ± 2.1
2	51.73 ± 0.0	84.37 ± 7.9	85.84 ± 7.1	48.51 ± 0.5	84.37 ± 4.1	**89.20 ± 2.7**
4	51.73 ± 0.0	84.83 ± 5.2	82.30 ± 9.5	49.43 ± 0.0	86.90 ± 3.7	**88.74 ± 2.6**
8	51.73 ± 0.0	91.26 ± 3.3	89.43 ± 5.5	56.78 ± 3.0	**94.94 ± 3.7**	90.57 ± 4.9
16	51.73 ± 0.0	88.05 ± 4.3	93.33 ± 3.2	76.09 ± 0.9	93.79 ± 3.3	**94.25 ± 1.9**

Table 5.5. Results for Experiment B: Investigation into the classification accuracy of calibrated and uncalibrated models when attempting to change the fidelity of data from a six DOF model to an eight DOF model. Maximum unsupervised accuracy for each fault case is indicated with blue font and maximum supervised accuracy for each fault case is indicated in purple font. The maximum classification accuracy regardless of unsupervised or supervised is further expressed in **bold**.

Number of Fault Samples in Training Set	Classification Accuracy on Target Domain Test Set [%]					
	Source Training Only (Unsupervised)	DANN (Unsupervised)	MMD-CNN (Unsupervised)	Target Training Only (Supervised)	DANN (Supervised)	MMD-CNN (Supervised)
0	46.67 ± 0.0	82.07 ± 7.4	**91.26 ± 2.2**	53.33 ± 0.0	83.56 ± 5.8	76.73 ± 6.1
1	46.67 ± 0.0	85.48 ± 8.4	**92.44 ± 2.0**	53.33 ± 0.0	84.59 ± 8.1	83.61 ± 3.5
2	46.67 ± 0.0	88.30 ± 4.2	90.07 ± 6.1	53.33 ± 0.0	90.37 ± 3.8	**93.63 ± 2.7**
4	46.67 ± 0.0	87.85 ± 4.2	**93.19 ± 3.4**	56.44 ± 2.3	90.22 ± 9.1	86.96 ± 5.2
8	46.67 ± 0.0	88.30 ± 5.5	**94.37 ± 1.5**	63.11 ± 2.1	93.63 ± 4.1	90.81 ± 4.8
16	46.67 ± 0.0	88.74 ± 3.5	93.78 ± 3.5	66.97 ± 2.0	93.33 ± 5.8	**93.93 ± 2.7**

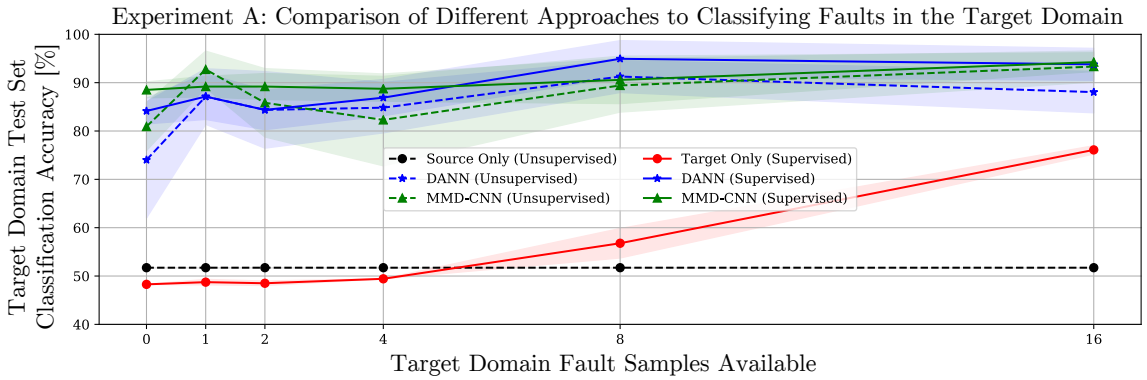


Figure 5.5. Resulting classification accuracies for Experiment A on the target domain for various approaches. This experiment involves the investigation of the classification accuracy of calibrated and uncalibrated models when attempting to change the data from one sensor location to another. Supervised techniques are shown with solid lines and unsupervised techniques with dashes. All experiments were run 5 times and therefore a standard deviation band is given.

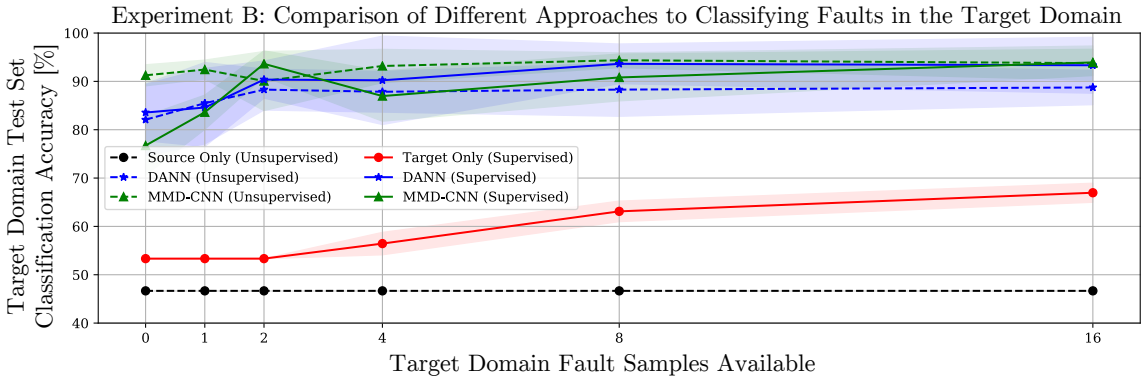


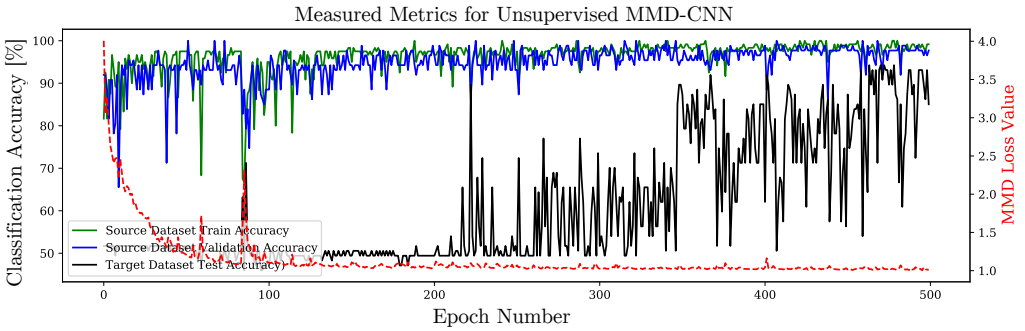
Figure 5.6. Resulting classification accuracies for Experiment B on the target domain for various approaches. This experiment involves the investigation of a change in model fidelity from a six DOF model to an eight DOF model on target domain classification accuracy. Supervised techniques are shown with solid lines and unsupervised techniques with dashes. All experiments were run 5 times and therefore a standard deviation band is given.

5.4.3.2 Training-specific results: Experiment A

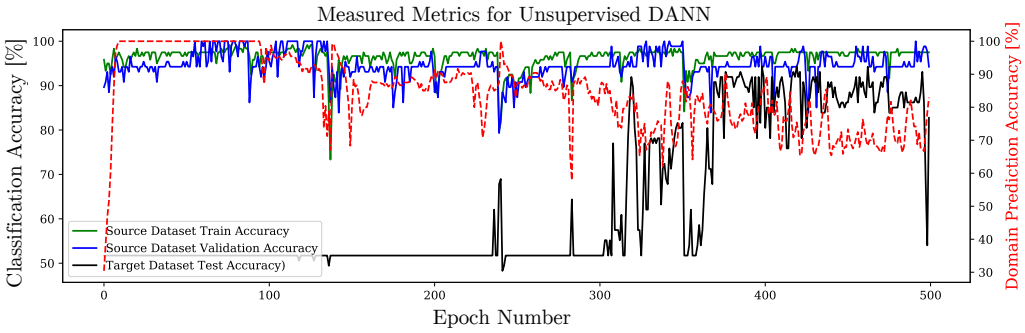
To gain some understanding of the training process of the transfer learning models, two main figure sets are given. Firstly, a batch of training curves for the unsupervised and supervised versions of both MMD and DANN is given. This is only be done for Experiment A, as Experiment B shows similar results, and therefore adds nothing extra to the discussion. This is shown in Figure 5.7.

Secondly, the confusion matrices for the final training epoch are given. Note that the final epoch is not necessarily the best epoch or the epoch which was selected for the final accuracy, but it does give a good insight into how the network is classifying samples near convergence. This is shown in Figure 5.8.

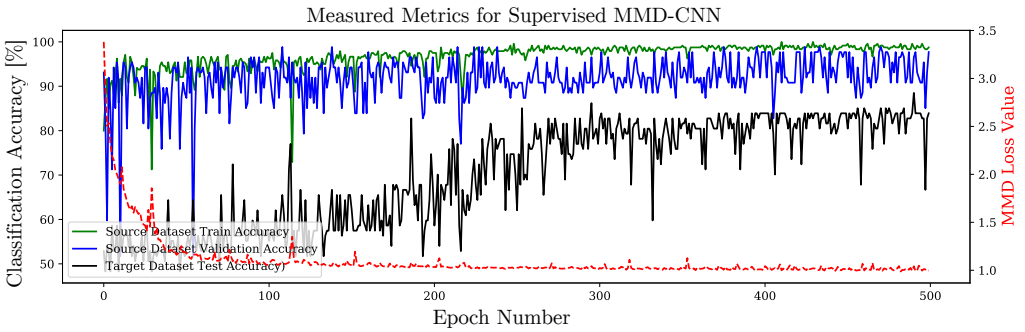
All results are for a single training run, on four fault samples available in the target domain. It is also seen that during the experimental runs, the patterns which are shown for four available faults also hold for other numbers of available faults. Therefore, only the case for four target domain faults is shown.



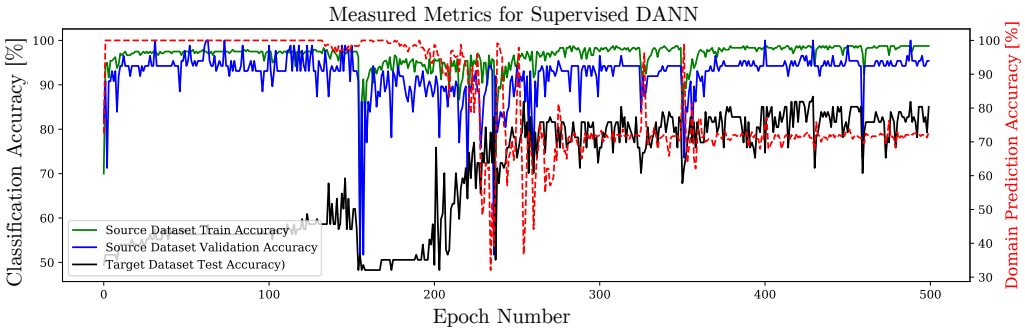
(a) Unsupervised training graphs for MMD approach.



(b) Unsupervised training graphs for DANN approach.

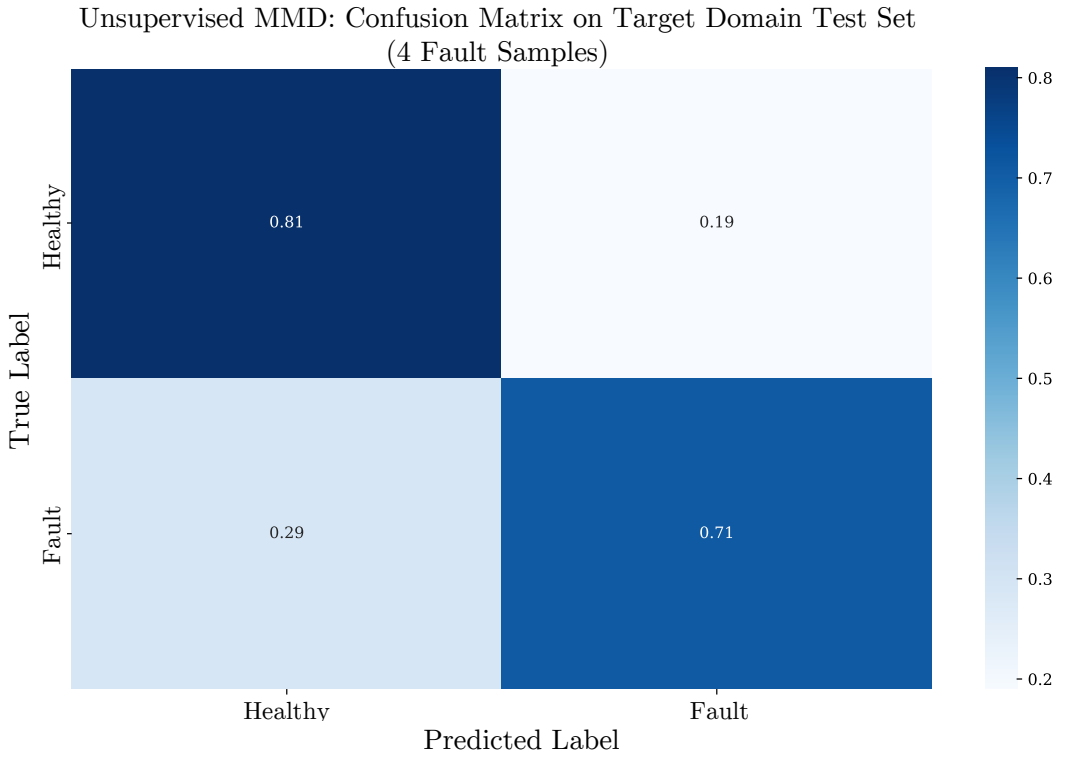


(c) Supervised training graphs for MMD approach.

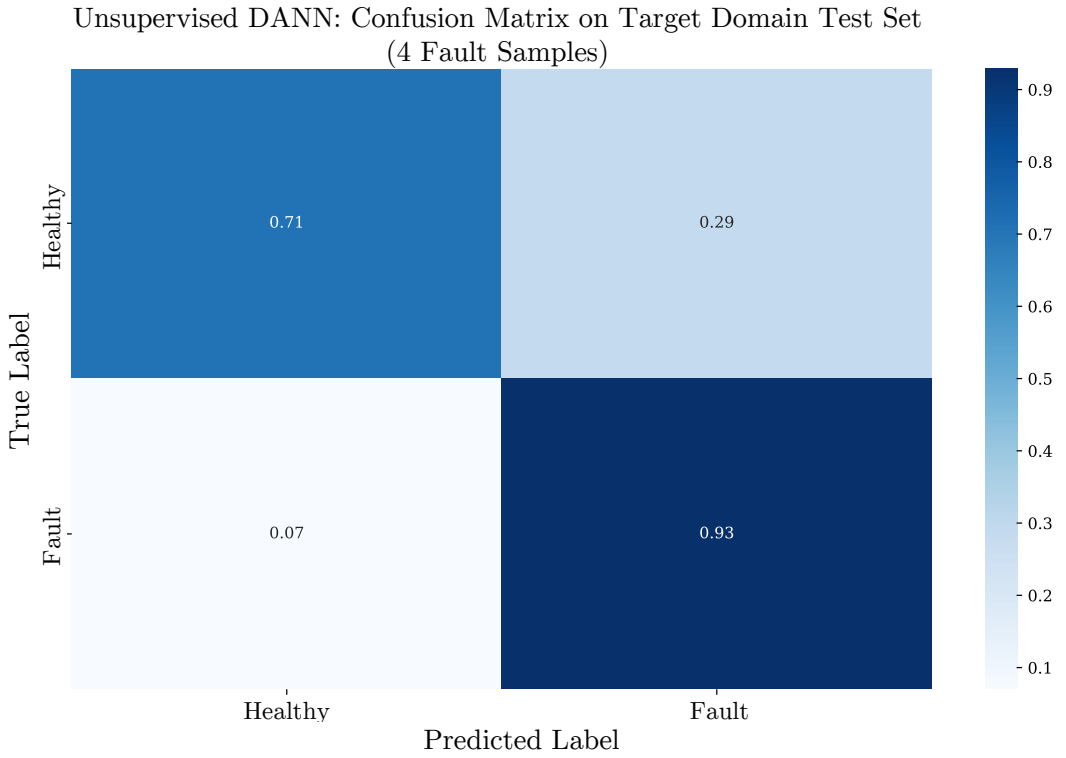


(d) Supervised training graphs for DANN approach.

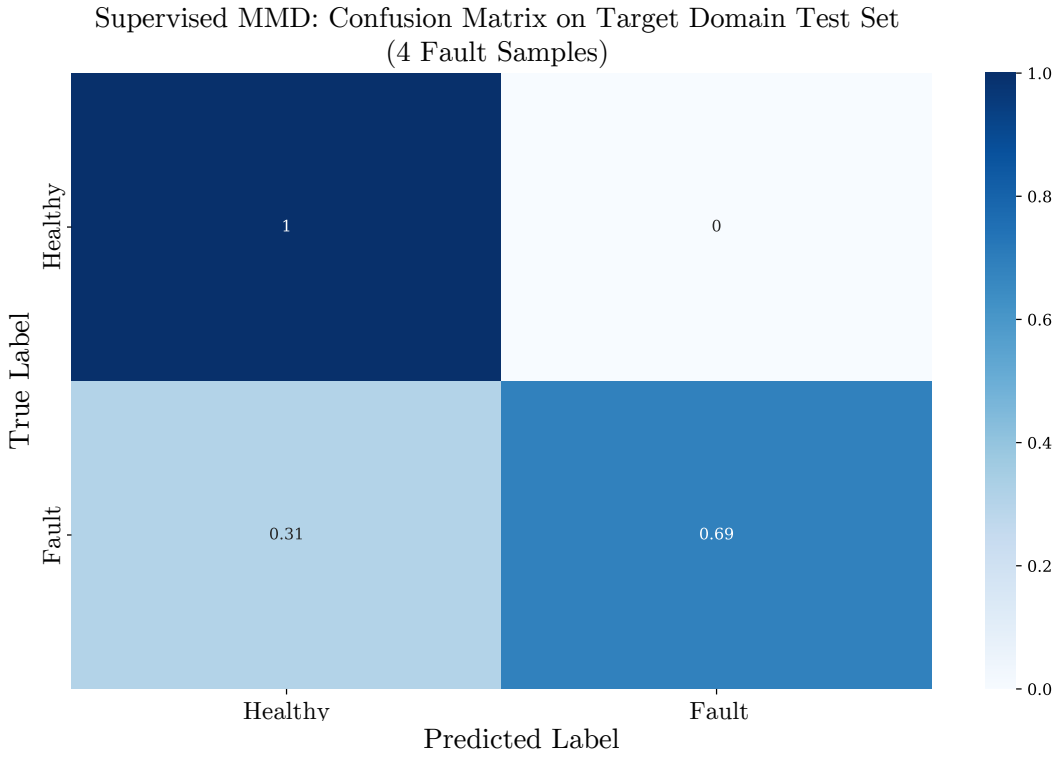
Figure 5.7. Compilation of training runs for the unsupervised and supervised versions of MMD and DANN. These showcase the dynamics of training each type of model.



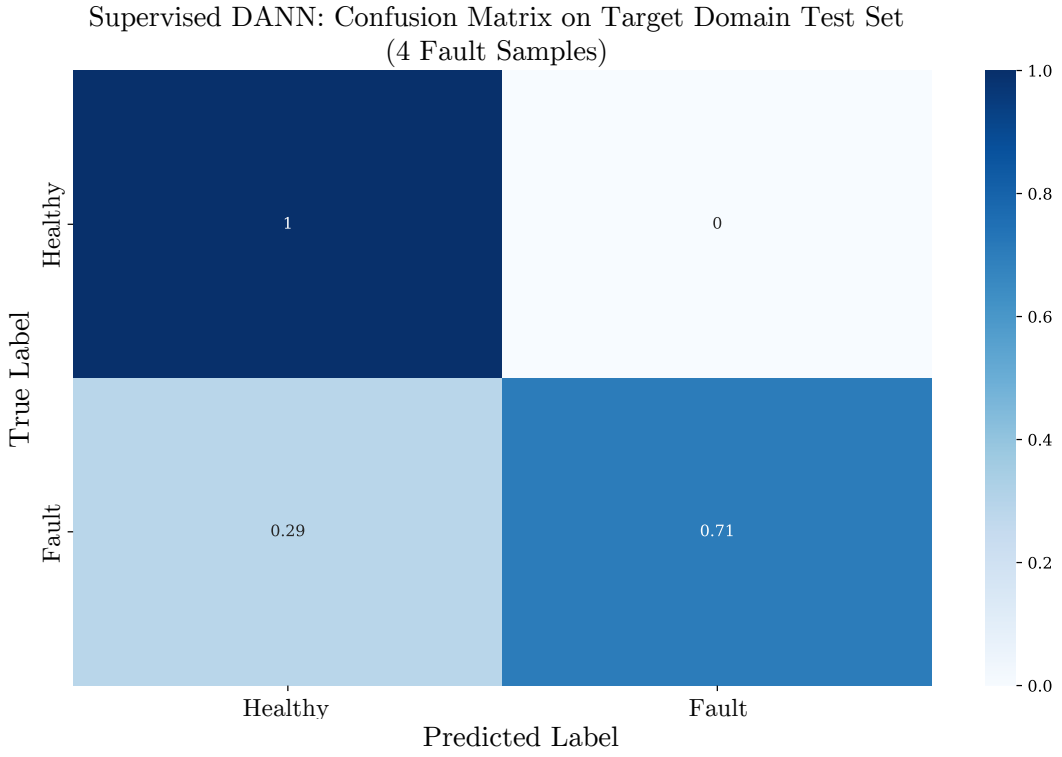
(a) Unsupervised confusion matrix for MMD approach.



(b) Unsupervised confusion matrix for DANN approach.



(c) Supervised confusion matrix for MMD approach.



(d) Supervised confusion matrix for DANN approach.

Figure 5.8. Compilation of confusion matrices for the unsupervised and supervised versions of MMD and DANN. These showcase how each model tends to classify (or misclassify) samples.

5.4.3.3 Domain adaptation evaluation: Experiment A

As a final attempt to understand the usefulness of domain adaptation, and to ensure it is working correctly, a zoomed feature space is given for the unsupervised and supervised versions of both MMD and DANN. The first and last epoch results for the source and target domains are given to serve as a 'before calibration' and 'after calibration' comparison. These figures compare the similarity between domains visually. A zoomed version is given simply due to the size of the feature space, which contains roughly 30000 features, which is impossible to display on a screen or paper. Furthermore, a single sample of the final epoch is extracted and shown for both the source and target domains. The compilation of these graphs is found in Figure 5.9

Take note that for all of the figures in Figure 5.9, the following applies: For the source and target domain plots, each row represents a single sample from the 120 samples available. The columns represent the features. For the lower figure, a single sample is plotted simultaneously from the source and target domain to compare the similarity of samples between domains.

5.5 Discussion of results

When looking at the classification accuracy results of all the different methods for Experiment A, a few key points are observed. In the unsupervised cases, that is to say, no labelled target domain data is utilised, the baseline CNN performs very poorly, only classifying 51.73% of samples correctly. In fact, upon deeper investigation, it is seen that all of the baseline CNN's predictions are to the fault class. This result is not shown in any results given in this chapter but was observed from the confusion matrices during training. Thus, when changing the domain of data, the baseline CNN simply classifies all signals as anomalous, which is essentially a pointless model. This shows that by simply changing the position of the sensor, the developed hybrid model completely fails at the fault detection task, as all samples are identified as faults.

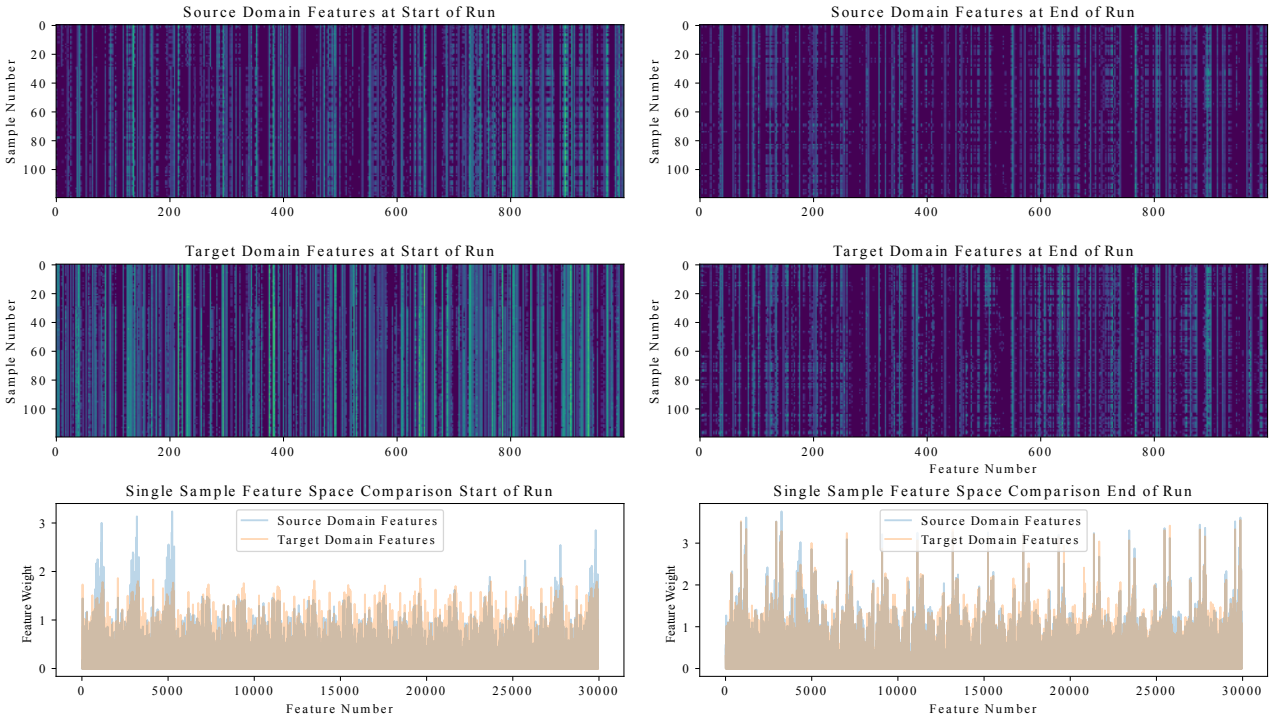
Moving on to the calibrated techniques, much more promising results are seen. The MMD-CNN seems to perform slightly better than DANN and can classify faults in an entirely unseen domain with a minimum of 81% and a maximum of 93% accuracy. This is an improvement of at least 30% over the uncalibrated baseline CNN, clearly proving the worth of transfer learning for calibration.

In the supervised cases, it is seen that the baseline CNN performed worse than the unsupervised baseline CNN initially, but then starts to improve. The baseline CNN trained on target domain data is seen to classify all samples as healthy (initially), and improved to classify some fault examples correctly. Once again, these results were seen during training and not shown here directly. The highest classification accuracy for the uncalibrated hybrid model is for 16 available fault examples, at only 76% accuracy.

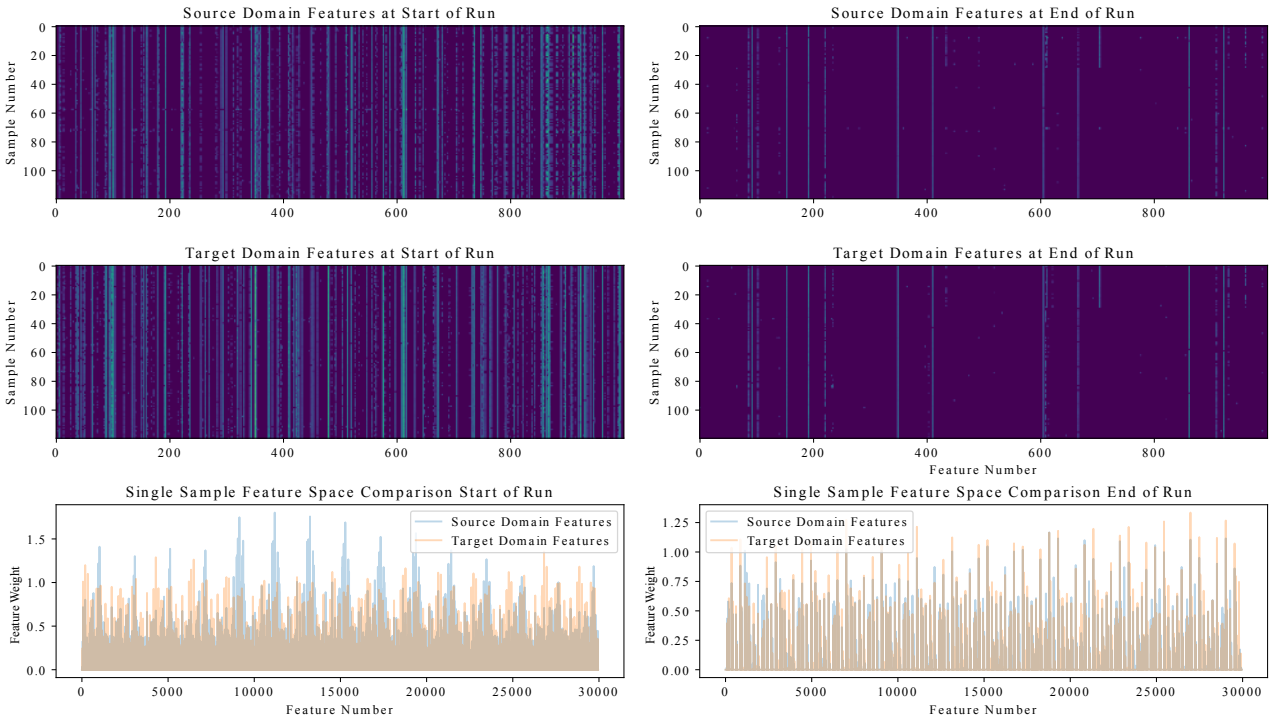
Once again, MMD seems to slightly outperform the DANN approach. It is also clear that the calibrated techniques are capable of classification accuracies 20-30% higher than the baseline CNN. This once again proves the major benefit of applying a transfer learning calibration when a data shift is present.

Special attention should be paid to the case where zero fault samples are made available in the target domain: Recognise that the calibrated techniques can classify faults in the target domain even though they have not been shown any prior faults. The MMD approach can classify with 81% accuracy with no labels, and with 89% accuracy with labels. This is an extremely promising result, as it shows that even if no historical fault data is available for an asset (which may be the case more often than not), the calibrated models may be capable of identifying faults on these assets.

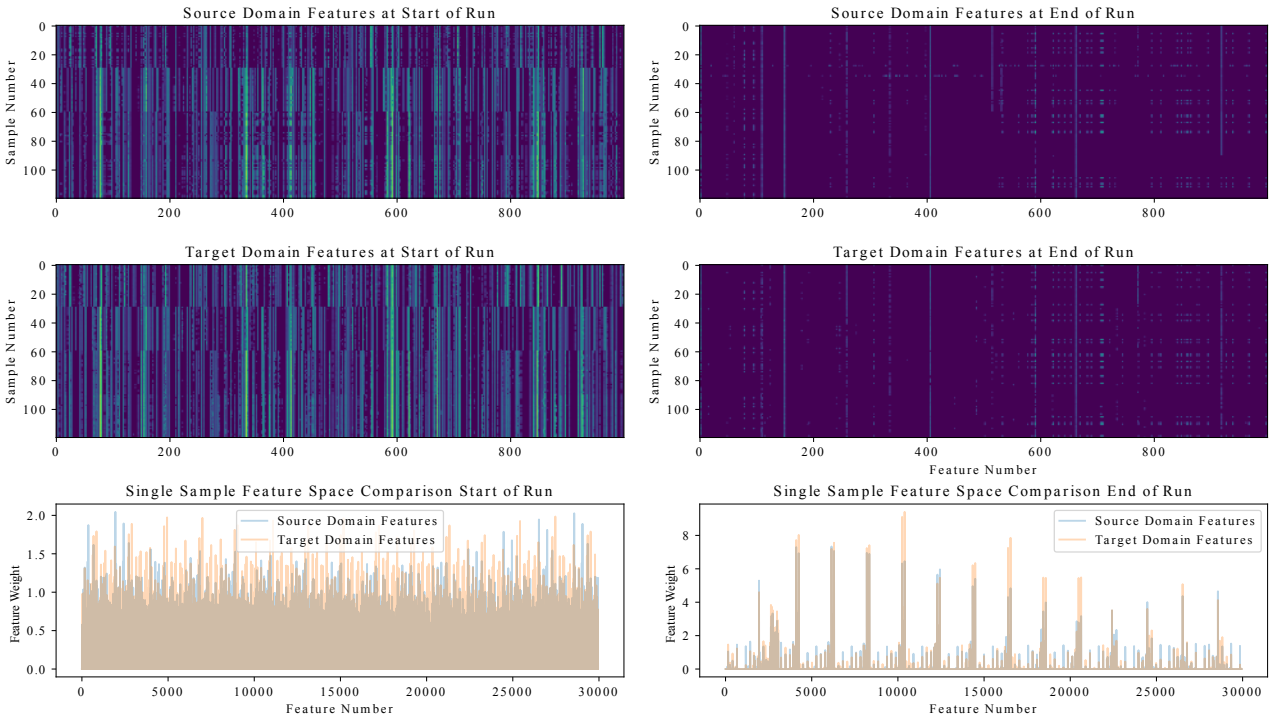
When comparing the supervised methods to the unsupervised methods, it is seen that for Experiment A, the supervised approaches generally outperform the unsupervised approaches by a few percent. As a sanity check, it is also seen that with an increasing amount of available target training data, all techniques which utilise this data, increase in classification accuracy.



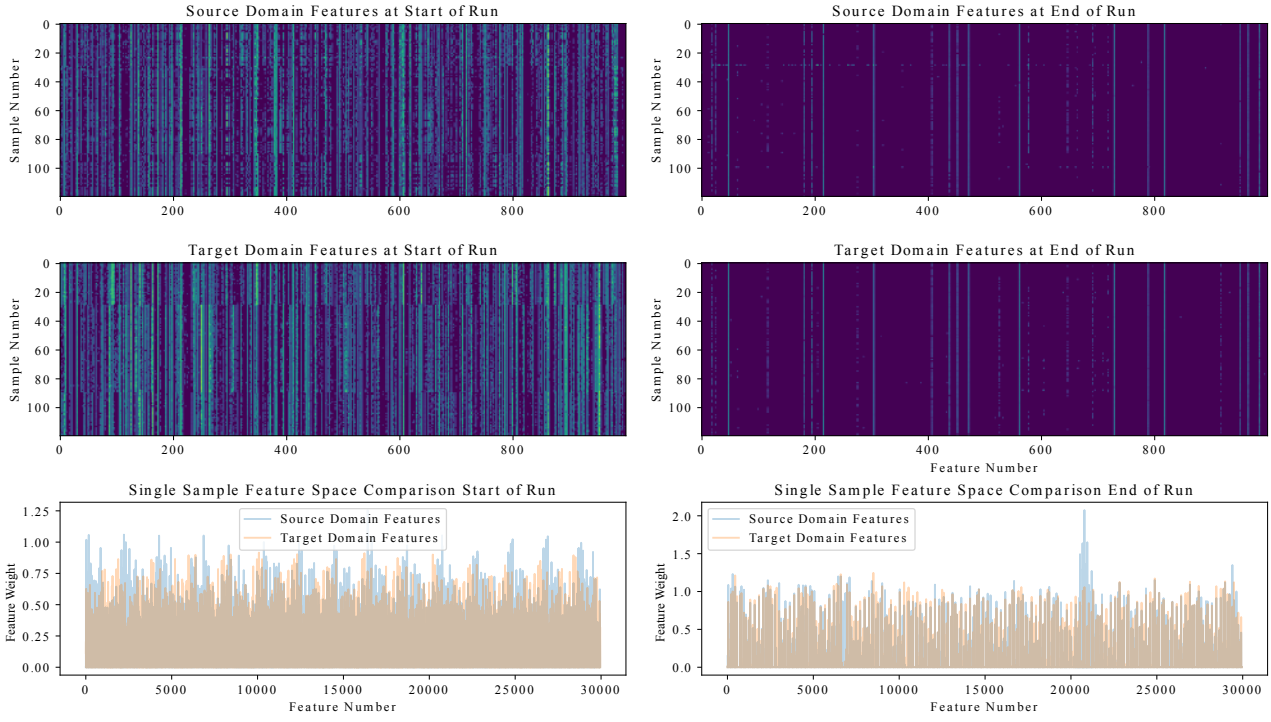
(a) Feature space comparisons for unsupervised MMD.



(b) Feature space comparisons for unsupervised DANN.



(c) Feature space comparisons for supervised MMD.



(d) Feature space comparisons for supervised DANN.

Figure 5.9. Compilation of feature space comparisons for the unsupervised and supervised versions of MMD and DANN. These figures give insight into the validity of claims about calibration made earlier in this chapter, i.e. whether transfer learning as a calibration tool can generate general but meaningful features.

Moving on to Experiment B, another set of useful observations may be made. Similar to Experiment A, all calibrated techniques prove vastly superior to the baseline CNNs. It is also observed that the source domain trained CNN defaults to classifying all signals as faulty, which is a pointless model. Also, the target domain trained CNN defaults to classifying all signals as healthy, and with enough fault samples, learns to classify up to 67% accuracy.

In this experiment, MMD-CNN is shown to be superior in the unsupervised case but is about equal in performance in the supervised case. Interestingly, the unsupervised approaches seem to fare better than the supervised methods, which is perhaps a bit counter-intuitive. However, this seemingly better result will be explained shortly when referring to the confusion matrices. As a final observation, note again the high accuracies obtained by the calibrated techniques for no fault samples present. The worst result from the calibrated approaches, where no target domain fault samples are given, still outperforms the baseline CNN which is trained on 16 available fault samples. This experiment further showcases that calibrated techniques adapt well to a change in data input, in this case, a change in model fidelity, whereas a baseline CNN may not be good enough for most cases.

These are important findings, as the general indication up until this point is that the developed hybrid model seems quite robust. These results, however, challenge this conclusion, and without any doubt showcase that the gap between works from literature, such as the developed hybrid model, and industry-like signals is too large and requires more robust methods. The transfer learning-calibrated models have shown to be very robust, and their potential impact on industrial condition monitoring seems promising. No concrete comments may be made with regards to industrial data directly, as experiments with such data is beyond the scope of this work. However, we may comment on the efficacy of this method to deal with shifts in data domains.

Moving on to the training-specific results, some further observations can be made. Firstly, when looking at the training runs, one can see that the unsupervised cases are much more stochastic, and have no clear convergence point. Admittedly, these graphs do not show the full training run, but when compared to their supervised counterparts, it becomes clear that the supervised approaches reach convergence much sooner and are much more stable. Furthermore, the supervised cases show good convergence of the domain discrepancy measure, where this is not always the case for the unsupervised cases, or more so DANN.

A promising observation from the training graphs is that the classification accuracy of the target domain test dataset (which is unknown in practice) seems to be predicted quite well by the validation dataset in the source domain. Therefore, upon implementation of a transfer learning calibration technique, if done correctly, the source domain validation dataset, which one does have access to, may be a very good indicator of which epochs also have good target domain classification accuracy. This is most clearly observed in Figure 5.7(d), where sudden drops in validation accuracy also correspond to sudden drops in target domain accuracy.

When looking at the confusion matrices for the calibrated techniques, an interesting observation is made: Although the supervised cases may have accuracies similar to their unsupervised counterparts (see Table 5.4 and Table 5.5), a clear shortcoming is observed. The unsupervised cases do not classify 100% of healthy cases correctly. This is a potential issue, as it may result in false alarms. This issue may be fixed with another pre-processing/filtering model which simply tries to identify definite healthy examples and only passes the possible fault examples to the unsupervised models. It is however preferred that these models can inherently identify all healthy signals automatically. However, it is promising to see that the supervised methods classify 100% of healthy samples correctly, making for a very useful fault detector. As mentioned in section 5.4.1, the datasets were biased towards healthy data, therefore this bias towards correctly classifying health samples is seen. This naturally comes with the trade-off that fault samples will be misclassified as healthy, but the author believes that this scenario is more acceptable (at lower fault intensities) compared to continually misclassifying healthy samples.

As a final point of discussion, the evaluation of the effectiveness of domain adaptation as a method to calibrate the models is given. When referring to Figure 5.9, a clear trend is seen. In all cases, the first epoch results in features that look visually dissimilar between the two domains. This is not surprising as the model is pre-trained

on only source domain data, leading to a clear discrepancy when a target domain data sample is fed into the network. Further note that the first epoch feature space has a large number of active features, indicating that the features are not necessarily very meaningful, but perhaps overfitted to the source domain.

When the results for the final epoch are observed, a much better picture emerges. Visually, source and target domain feature space look similar, implying that the domain adaptation process works correctly to create domain invariant features. Further notice that the features are much more sparse, indicating that the learnt features are much more specific and perhaps therefore meaningful. These features are simultaneously domain invariant and meaningful, which is why the classification success of the calibrated techniques are so high. These figures further support the idea that creating a domain invariant feature space is possible and may result in good classification accuracies in a domain with limited to no labelled data.

5.6 Concluding remarks

To conclude this chapter, the following conclusions are drawn from both Experiment A and B as well as the further investigations into the training process of the calibration methods, and the feature space that results from these calibration techniques.

Firstly, it is clear that the minor task of changing the location of vibration measurement or simply increasing the model fidelity greatly impacted the success of the already successful hybrid model. This is important, as this shows that a hybrid model itself is not sufficient to directly solve problems in industry. It is however clear that regardless of labels in the target domain, the calibrated methods vastly outperform their uncalibrated counterparts, making these techniques strong candidates to bridge the gap between literature and industry.

It is further shown that even if no fault samples exist in the target domain, the calibrated methods can learn useful mappings such that faults can be detected in the target domain. This is an important finding, as this implies that with a hybrid model and some target domain healthy data, useful conclusions may already be drawn on the state of a target domain without ever seeing a fault example. Generally, it is also concluded that MMD-CNN seems to show slightly more promise than DANN, but this may simply be a function of the transfer problem at hand. Regardless, it is abundantly clear that whether MMD-CNN or DANN is chosen, either should vastly outperform the uncalibrated approaches.

It is also seen that unsupervised methods are more difficult to train. They struggle to converge due to their stochastic training curves. However, it is found that if the validation accuracy in the source domain is followed, a good estimate may be made to ascertain a good epoch to stop the training. It is also seen that the unsupervised methods are not able to identify healthy samples with 100% accuracy, whereas supervised methods are. This is not a great problem but complicates the fault classification task.

Finally, it may be concluded that domain adaptation as a transfer learning calibration technique, creates a feature space that is much more meaningful and invariant between domains, leading to the promising results obtained from these experiments.

Therefore, the main takeaway from this chapter and the entirety of this work should be this: Transfer learning calibration techniques show great promise in aligning the source and target domains for hybrid models. Calibrated methods are seen to vastly outperform uncalibrated methods and therefore show great promise to the condition monitoring task of gearboxes.

With this, the third and final novelty of the work has been delivered: A hybrid model which, if coupled with one of the shown calibration techniques (MMD-CNN or DANN), is very robust to a change in the data domain, so much so that it may potentially bridge the gap between synthetically generated data and industrial data.

Chapter 6 Conclusions and recommendations

6.1 Key conclusions from this research

The main task of this research was to develop a hybrid model which could be used for gearbox condition monitoring. The hybrid model would not only act as a strong fault classifier but would be capable of transferring its hybrid knowledge from one domain of data to another. Ideally, these domains would be the synthetic data domain and real industrial measurements. To develop the necessary hybrid model, a decision was made to fuse a supervised data-driven model with synthetic data generated from a physics-driven gearbox model.

The first part of this work therefore set out to develop a physical gearbox model which may be used to generate synthetic data. During this modelling process, a shortcoming in literature was addressed, namely a disjunct set of literature that include different modelling equations for different fault types. This work successfully developed a generalised stiffness model, paired with four fault models, namely cracks, chips, pits and spalls. The generalised stiffness model aimed to generalise the gear stiffness modelling process, making future works more comparable and reproducible. Furthermore, such a framework made it simple to generate large amounts of synthetic data. This is the first instance of such a generalised model and represents the first novel contribution of this work. The generalised model was further verified against existing works. During the modelling process, an advancement to crack models was also made, by reducing the necessary dimensions of integration by one, increasing the efficiency of simulating crack models. This too was verified against existing works. Finally, two independent dynamic models were built and verified against existing works. The dynamic models were shown to have correct amplitudes, shapes and frequency behaviour.

The second part of the work focused on the data-driven portion of the hybrid model, with special attention being given to the verification of the developed hybrid model. The synthetic data from the physics-driven model was infused with noise and fed to a CNN. This experiment showed two key findings. Firstly, it was shown that signal processing techniques such as TSA significantly improve model performance in noisy environments. Therefore, it is advised that when possible, TSA be applied as a pre-processor for a data-driven model. Secondly, it was shown that when paired with TSA pre-processing, the hybrid model performed well at identifying different fault types under three different noise profiles. Therefore, with the hybrid model successfully identifying faults of different intensities and locations, it was shown that the hybrid model showed promise in detecting gearbox faults. This verified hybrid model represents the second novel contribution of the work, as to the author's knowledge, no other hybrid gearbox model (in this format) exists.

As well as the hybrid model seemed to perform on sufficient labelled data, it was unknown how it would generalise to a different set of related data. Upon investigation, it was shown that the model accuracy suffered severely when trying to apply the hybrid model to different datasets. This brought the usefulness of the hybrid model into question. Two transfer learning calibration techniques were presented as solutions to minimise the discrepancy between different datasets and showed outstanding results. It was shown that when calibration approaches were applied to the hybrid model, the model was able to identify faults within a domain of data it had never seen before. This is an important property for such a hybrid model, as it raises the confidence that this model may be capable of generalising to industrial assets, finally bridging the gap between literature and practice. This serves as the third and final novel contribution of the work, namely a transfer learning calibrated hybrid gearbox model trying to bridge the gap between data from two completely different datasets.

6.2 Recommendations for future work

The following items present a list in order of importance that may be pursued to build upon this work. It may be useful to split the research into the two main components which define the hybrid model, namely the physics-driven side and the data-driven side.

Future research for the physics-driven model:

1. Literature containing dynamic gearbox models commonly assume damping conditions that are not necessarily proven. They are convenient but do not represent the true dynamics of the system. Some works do however build quite advanced damping models. Therefore, if the current physical modelling portion of the hybrid model is to be improved, this is the first area of research that should receive attention.
2. It was found from the initial hybrid model experiments that the identifiability of faults seemed to correlate directly to the fault's effect on the gear meshing frequency. Therefore, it is logical to research this point further. Is there a direct correlation between the "sharpness" of stiffness reductions and identifiability of faults, or was this just chance?
3. The gearbox model developed in this work only focused on spur gears. However, many critical industrial assets utilise helical gears, and therefore future work could be focused on building the same generalised structure for helical gears.
4. Some of the fault models developed in this work are quite simple and are not state-of-the-art. Therefore, in future research, these models should be updated to try and match the state-of-the-art models.
5. Although the literature on the potential energy method has shown good correlation with FEM, one main issue is still present: When the fillet foundation stiffness term is calculated for two simultaneous gear tooth pairs in mesh, an overestimation of stiffness occurs. This is because the theory for the fillet foundation stiffness term is based on a single tooth pair meshing assumption. To calibrate this overestimation, a FEM gear model is required. This, however, defeats the purpose of the potential energy method in the first place, as one would like to avoid expensive FEM models and rather utilise equally accurate analytical methods. Therefore, this is an open issue that needs to be addressed, such that the fillet foundation term can be accurately calculated without the need for FEM.

Future research for the data-driven model:

1. First and foremost, the developed calibrated hybrid models must be tested on a real dataset. Therefore, the most logical continuation of this work is to first and foremost redo the experiments from Chapter 5. Instead of the developed synthetic dataset in the target domain, it may be replaced by a real dataset. This will truly test the efficacy of actually applying the developed transfer learning-calibrated hybrid model to industrial problems.
2. Most of the transfer learning works in literature seem to focus on time domain signals. A useful experiment would be to investigate the effectiveness of perhaps changing over to the frequency or time-frequency domain.
3. The current transfer learning techniques aim to manipulate the feature space. An interesting and meaningful investigation would be to look into the effectiveness of applying the transfer learning calibration techniques at different layer depths. Taking this idea to its extreme, perhaps a domain adaptation layer straight at the input of the network would suffice, directly modifying input signals instead of the latent space.
4. Another large area of concern during the present research was finding stoppage conditions during training of the transfer learning calibrated models. It was not always clear whether a network had converged and it is believed that there is substantial untapped performance potential. Some training runs might have been stopped too late (overfitting) or too early (too little learning), and finding the balance is still an open question. Finding a balance is difficult, as the transfer learning techniques generally have competing training goals, leading to a tug of war between loss function optimisation. This tug of war makes it difficult to determine when an optimum has been reached and must receive further attention.

References

Bartelmus, W., Chaari, F., Zimroz, R. & Haddar, M. (2010), 'Modelling of gearbox dynamics under time-varying nonstationary load for distributed fault detection and diagnosis', *European Journal of Mechanics, A/Solids* **29**(4), 637–646.

Bishop, C. M. (2006), *Pattern recognition and machine learning*, Information science and statistics, Springer, New York, NY.

Bobylev, D., Choudhury, T., Miettinen, J., Viitala, R., Kurvinen, E. & Sopanen, J. (2021), 'Simulation-based transfer learning for support stiffness identification', *IEEE Access* **9**(Ml), 1–1.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H. P., Schölkopf, B. & Smola, A. J. (2006), 'Integrating structured biological data by Kernel Maximum Mean Discrepancy', *Bioinformatics* **22**(14), 49–57.

Chaari, F., Baccar, W., Abbes, M. S. & Haddar, M. (2008), 'Effect of spalling or tooth breakage on gearmesh stiffness and dynamic response of a one-stage spur gear transmission', *European Journal of Mechanics, A/Solids* **27**(4), 691–705.

Chaari, F., Bartelmus, W., Zimroz, R., Fakhfakh, T. & Haddar, M. (2012), 'Gearbox vibration signal amplitude and frequency modulation', *Shock and Vibration* **19**(4), 635–652.

Chaari, F., Fakhfakh, T. & Haddar, M. (2006), 'Dynamic analysis of a planetary gear failure caused by tooth pitting and cracking', *Journal of Failure Analysis and Prevention* **6**(2), 73–78.

Chaari, F., Fakhfakh, T. & Haddar, M. (2009), 'Analytical modelling of spur gear tooth crack and influence on gearmesh stiffness', *European Journal of Mechanics, A/Solids* **28**(3), 461–468.

Chen, T., Wang, Y. & Chen, Z. (2019), 'A novel distribution model of multiple teeth pits for evaluating time-varying mesh stiffness of external spur gears', *Mechanical Systems and Signal Processing* **129**, 479–501.

Chen, Z., He, G., Li, J., Liao, Y., Gryllias, K. & Li, W. (2020), 'Domain adversarial transfer network for cross-domain fault diagnosis of rotary machinery', *IEEE Transactions on Instrumentation and Measurement* **69**(11), 8702–8712.

Chen, Z. & Shao, Y. (2011), 'Dynamic simulation of spur gear with tooth root crack propagating along tooth width and crack depth', *Engineering Failure Analysis* **18**(8), 2149–2164.

Cubillo, A., Perinpanayagam, S. & Esperon-Miguez, M. (2016), 'A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery', *Advances in Mechanical Engineering* **8**(8), 1–21.

El Yousfi, B., Soualhi, A., Medjaher, K. & Guillet, F. (2020), 'New approach for gear mesh stiffness evaluation of spur gears with surface defects', *Engineering Failure Analysis* **116**(December 2019), 104740.

Feng, K., Borghesani, P., Smith, W. A., Randall, R. B., Chin, Z. Y., Ren, J. & Peng, Z. (2019), 'Vibration-based updating of wear prediction for spur gears', *Wear* **426-427**(September 2018), 1410–1415.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. & Lempitsky, V. (2017), *Domain-Adversarial Training of Neural Networks*, Springer International Publishing, Cham, pp. 189–209.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. (2012), 'A kernel two-sample test', *Journal of Machine Learning Research* **13**, 723–773.

Gryllias, K. C. & Antoniadis, I. A. (2012), 'A Support Vector Machine approach based on physical model training for rolling element bearing fault detection in industrial environments', *Engineering Applications of Artificial Intelligence* **25**(2), 326–344.

Guilbault, R., Lalonde, S. & Thomas, M. (2012), 'Nonlinear damping calculation in cylindrical gear dynamic modeling', *Journal of Sound and Vibration* **331**(9), 2110–2128.

Han, B., Yang, X., Ren, Y. & Lan, W. (2019c), 'Comparisons of different deep learning-based methods on fault diagnosis for geared system', *International Journal of Distributed Sensor Networks* **15**(11).

Han, T., Liu, C., Yang, W. & Jiang, D. (2019a), 'A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults', *Knowledge-Based Systems* **165**, 474–487.

Han, T., Liu, C., Yang, W. & Jiang, D. (2019b), 'Learning transferable features in deep convolutional neural networks for diagnosing unseen machine conditions', *ISA Transactions* **93**, 341–353.

Hibbeler, R. C. (2004), *Engineering mechanics: statics*, Vol. 1, Pearson Education.

Hinton, G. & Salakhutdinov, R. (2006), 'Reducing the dimensionality of data with neural networks', **313**(July), 504–507.

Hyatt, G., Piber, M., Chaphalkar, N., Kleinhenz, O. & Mori, M. (2014), 'A review of new strategies for gear production', *Procedia CIRP* **14**(December 2014), 72–76.

Ioffe, S. & Szegedy, C. (2015), 'Batch normalization: Accelerating deep network training by reducing internal covariate shift', *32nd International Conference on Machine Learning, ICML 2015* **1**, 448–456.

Jia, F., Lei, Y., Lin, J., Zhou, X. & Lu, N. (2016), 'Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data', *Mechanical Systems and Signal Processing* **72-73**, 303–315.

Jia, S. & Howard, I. (2006), 'Comparison of localised spalling and crack damage from dynamic modelling of spur gear vibrations', *Mechanical Systems and Signal Processing* **20**(2), 332–349.

Jia, S., Howard, I. & Wang, J. (2003), 'The dynamic modeling of multiple pairs of spur gears in mesh, including friction and geometrical errors', *The International Journal of Rotating Machinery* **9**(6), 437–442.

Jing, L., Zhao, M., Li, P. & Xu, X. (2017), 'A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox', *Measurement: Journal of the International Measurement Confederation* **111**, 1–10.

Khabou, M. T., Bouchaala, N., Chaari, F., Fakhfakh, T. & Haddar, M. (2011), 'Study of a spur gear dynamic behavior in transient regime', *Mechanical Systems and Signal Processing* **25**(8), 3089–3101.

Khan, S. & Yairi, T. (2018), 'A review on the application of deep learning in system health management', *Mechanical Systems and Signal Processing* **107**, 241–265.

Kumar, A., Gandhi, C. P., Zhou, Y., Kumar, R. & Xiang, J. (2020), 'Latest developments in gear defect diagnosis and prognosis: A review', *Measurement: Journal of the International Measurement Confederation* **158**.

Kundu, P., Darpe, A. K. & Kulkarni, M. S. (2020), 'A review on diagnostic and prognostic approaches for gears', *Structural Health Monitoring* p. 147592172097292.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T. & Lin, J. (2018a), 'Machinery health prognostics: A systematic review from data acquisition to RUL prediction', *Mechanical Systems and Signal Processing* **104**, 799–834.

Lei, Y., Liu, Z., Wang, D., Yang, X., Liu, H. & Lin, J. (2018b), 'A probability distribution model of tooth pits for evaluating time-varying mesh stiffness of pitting gears', *Mechanical Systems and Signal Processing* **106**, 355–366.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N. & Nandi, A. K. (2020), 'Applications of machine learning to machine fault diagnosis: A review and roadmap', *Mechanical Systems and Signal Processing* **138**, 106587.

Lewicki, D. G. (2002), 'Gear crack propagation path studies-guidelines for ultra-safe design', *Journal of the American Helicopter Society* **47**(1), 64–72.

Liang, X., Zhang, H., Liu, L. & Zuo, M. J. (2016), 'The influence of tooth pitting on the mesh stiffness of a pair of external spur gears', *Mechanism and Machine Theory* **106**, 1–15.

Liang, X., Zuo, M. J. & Pandey, M. (2014), 'Analytically evaluating the influence of crack on the mesh stiffness of a planetary gear set', *Mechanism and Machine Theory* **76**, 20–38.

Liu, C. & Gryllias, K. (2021), 'Simulation-driven domain adaptation for rolling element bearing fault diagnosis', *IEEE Transactions on Industrial Informatics* **3203**(c), 1–1.

Liu, C., Mauricio, A., Qi, J., Peng, D. & Gryllias, K. (2020a), 'Domain adaptation digital twin for rolling element bearing prognostics', *Annual Conference of the PHM Society* pp. 1–10.

Liu, F. H., Theodossiades, S., Bergman, L. A., Vakakis, A. F. & Mcfarland, D. M. (2015), 'Analytical characterization of damping in gear teeth dynamics under hydrodynamic conditions', *Mechanism and Machine Theory* **94**, 141–147.

Liu, J., Chen, S., Bergés, M., Bielak, J., Garrett, J. H., Kovačević, J. & Noh, H. Y. (2020c), 'Diagnosis algorithms for indirect structural health monitoring of a bridge model via dimensionality reduction', *Mechanical Systems and Signal Processing* **136**.

Liu, J., Wang, C. & Wu, W. (2020b), 'Research on meshing stiffness and vibration response of pitting fault gears with different degrees', *International Journal of Rotating Machinery* **2020**.

Liu, Y., Shi, Z., Shen, G., Zhen, D., Wang, F. & Gu, F. (2021), 'Evaluation model of mesh stiffness for spur gear with tooth tip chipping fault', *Mechanism and Machine Theory* **158**, 104238.

Luo, Y., Baddour, N., Han, G., Jiang, F. & Liang, M. (2018b), 'Evaluation of the time-varying mesh stiffness for gears with tooth spalls with curved-bottom features', *Engineering Failure Analysis* **92**(June), 430–442.

Luo, Y., Baddour, N. & Liang, M. (2018a), 'Evaluation of time-varying mesh stiffness of gears with tooth spalls modeled as spherical shapes', *Proceedings - 2018 Prognostics and System Health Management Conference, PHM-Chongqing 2018* pp. 94–99.

Luo, Y., Baddour, N. & Liang, M. (2019*a*), 'A shape-independent approach to modelling gear tooth spalls for time varying mesh stiffness evaluation of a spur gear pair', *Mechanical Systems and Signal Processing* **120**, 836–852.

Luo, Y., Baddour, N. & Liang, M. (2019*b*), 'Dynamical modeling and experimental validation for tooth pitting and spalling in spur gears', *Mechanical Systems and Signal Processing* **119**, 155–181.

Ma, H., Song, R., Pang, X. & Wen, B. (2014), 'Time-varying mesh stiffness calculation of cracked spur gears', *Engineering Failure Analysis* **44**, 179–194.

Ma, H., Zeng, J., Feng, R., Pang, X., Wang, Q. & Wen, B. (2015), 'Review on dynamics of cracked gear systems', *Engineering Failure Analysis* **55**, 224–245.

Ma, H., Zeng, J., Feng, R., Pang, X. & Wen, B. (2016), 'An improved analytical method for mesh stiffness calculation of spur gears with tip relief', *Mechanism and Machine Theory* **98**, 64–80.

Meng, Z., Shi, G. & Wang, F. (2020), 'Vibration response and fault characteristics analysis of gear based on time-varying mesh stiffness', *Mechanism and Machine Theory* **148**, 103786.

Meng, Z., Wang, F. & Shi, G. (2021), 'A novel evolution model of pitting failure and effect on time -varying meshing stiffness of spur gears', *Engineering Failure Analysis* **120**(February 2020), 105068.

Mohammed, O. D. & Rantatalo, M. (2015), 'Dynamic response and time-frequency analysis for gear tooth crack detection', *Mechanical Systems and Signal Processing* **66-67**, 612–624.

Mohammed, O. D. & Rantatalo, M. (2020), 'Gear fault models and dynamics-based modelling for gear fault detection – A review', *Engineering Failure Analysis* **117**(May), 104798.

Mohammed, O. D., Rantatalo, M. & Aidanpää, J. O. (2013a), 'Improving mesh stiffness calculation of cracked gears for the purpose of vibration-based fault analysis', *Engineering Failure Analysis* **34**, 235–251.

Mohammed, O. D., Rantatalo, M. & Aidanpää, J. O. (2015), 'Dynamic modelling of a one-stage spur gear system and vibration-based tooth crack detection analysis', *Mechanical Systems and Signal Processing* **54**, 293–305.

Mohammed, O. D., Rantatalo, M., Aidanpää, J. O. & Kumar, U. (2013b), 'Vibration signal analysis for gear fault diagnosis with various crack progression scenarios', *Mechanical Systems and Signal Processing* **41**(1-2), 176–195.

Muskhelishvili, N. I. (1977), 'Basic equations of the plane theory of elasticity', *Some Basic Problems of the Mathematical Theory of Elasticity* pp. 89–104.

Pan, S. J. & Yang, Q. (2010), 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering* **22**(10), 1345–1359.

Parker, R. G., Vijayakar, S. M. & Imajo, T. (2000), 'Non-linear dynamic response of a spur gear pair: modelling and experimental comparisons', *Journal of Sound and Vibration* **237**(3), 435–455.

Philippe, V. (2012), On the modelling of spur and helical gear dynamic behaviour, *in* 'Mechanical Engineering', InTech.

Richards, D. A. & Pines, D. J. (2000), 'Effects of tooth damage on spur gear vibrations', *Smart Structures and Materials 2000: Smart Structures and Integrated Systems* **3985**(301), 306.

Ruder, S. (2016), 'An overview of gradient descent optimization algorithms', pp. 1–14.

Ruiz-Botero, J. A., López-López, J. F. & Quintero-Riaza, H. F. (2015), 'Design, modeling and dynamic simulation of three double stage gearboxes with different module, mesh stiffness fluctuation and different level tooth breakage', *Revista Facultad de Ingenieria* **1**(74), 117–131.

Sainsot, P., Velex, P. & Duverger, O. (2004), 'Contribution of gear body to tooth deflections - A new bidimensional analytical formula', *Journal of Mechanical Design, Transactions of the ASME* **126**(4), 748–752.

Saxena, A., Parey, A. & Chouksey, M. (2015), 'Effect of shaft misalignment and friction force on time varying mesh stiffness of spur gear pair', *Engineering Failure Analysis* **49**, 79–91.

Saxena, A., Parey, A. & Chouksey, M. (2016), 'Time varying mesh stiffness calculation of spur gear pair considering sliding friction and spalling defects', *Engineering Failure Analysis* **70**, 200–211.

Schmidt, S. (2017), A cost-effective diagnostic methodology using probabilistic approaches for gearboxes operating under non-stationary conditions, Masters dissertation, University of Pretoria.

Schmidt, S., Heyns, P. S. & Gryllias, K. C. (2020), 'A methodology using the spectral coherence and healthy historical data to perform gearbox fault diagnosis under varying operating conditions', *Applied Acoustics* **158**, 107038.

Sobie, C., Freitas, C. & Nicolai, M. (2018), 'Simulation-driven machine learning: Bearing fault classification', *Mechanical Systems and Signal Processing* **99**, 403–419.

Stander, C. J. & Heyns, P. S. (2005), 'Instantaneous angular speed monitoring of gearboxes under non-cyclic stationary load conditions', *Mechanical Systems and Signal Processing* **19**(4), 817–835.

Tian, X. (2004), Dynamic simulation for system response of gearbox including localized gear faults, Masters dissertation, University of Alberta.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K. & Darrell, T. (2014), 'Deep domain confusion: maximizing for domain invariance'.

van Eyk, L., Heyns, P. S. & Schmidt, S. (2022), A short review of gear fault modelling in a hybrid modelling context, *in* 'Modelling and Simulation of Complex Systems for Sustainable Energy Efficiency', Vol. 20, pp. 242–258.

Verstraete, D., Ferrada, A., Droguett, E. L., Meruane, V. & Modarres, M. (2017), 'Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings', *Shock and Vibration* **2017**, 1–17.

Wan, Z., Cao, H., Zi, Y., He, W. & He, Z. (2014), 'An improved time-varying mesh stiffness algorithm and dynamic modeling of gear-rotor system with tooth root crack', *Engineering Failure Analysis* **42**, 157–177.

Wang, S., Xieeryazidan, A., Zhang, X. & Zhou, J. (2020), 'An improved computational method for vibration response and radiation noise analysis of two-stage gearbox', *IEEE Access* **8**, 85973–85988.

Wei, J., Zhang, A., Qin, D., Lim, T. C., Shu, R., Lin, X. & Meng, F. (2017), 'A coupling dynamics analysis method for a multistage planetary gear system', *Mechanism and Machine Theory* **110**(October 2016), 27–49.

Wu, S., Zuo, M. J. & Parey, A. (2008), 'Simulation of spur gear dynamics and estimation of fault growth', *Journal of Sound and Vibration* **317**(3-5), 608–624.

Xu, H., Qin, D., Liu, C., Yi, Y. & Jia, H. (2017), 'Dynamic modeling of multistage gearbox and analysis method of resonance danger path', *IEEE Access* **PP**, 1.

Xu, H., Qin, D., Liu, C. & Zhang, Y. (2021), 'An improved dynamic model updating method for multistage gearbox based on surrogate model and sensitivity analysis', *IEEE Access* **9**, 18527–18537.

Yang, D. C. & Lin, J. Y. (1987), 'Hertzian damping, tooth friction and bending elasticity in gear impact dynamics', *Journal of Mechanical Design, Transactions of the ASME* **109**(2), 189–196.

Yang, D. C. & Sun, Z. S. (1985), 'A rotary model for spur gear dynamics', *Journal of Mechanical Design, Transactions of the ASME* **107**(4), 529–535.

Yang, L., Wang, L., Yu, W. & Shao, Y. (2021), 'Investigation of tooth crack opening state on time varying meshing stiffness and dynamic response of spur gear pair', *Engineering Failure Analysis* **121**(December 2020), 105181.

Yi, Y., Tan, X., Xuan, L. & Liu, C. (2020), 'Dynamic interaction behavior of an electric motor drive multistage gear set', *IEEE Access* **8**(April), 66951–66960.

Yu, W. (2017), 'Dynamic modelling of gear transmission systems with and without localized tooth defects', (May).

Zak, G., Wyłomańska, A. & Zimroz, R. (2017), 'Data-driven iterative vibration signal enhancement strategy using alpha stable distribution', *Shock and Vibration* **2017**.

Zhang, W., Li, C., Peng, G., Chen, Y. & Zhang, Z. (2018), 'A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load', *Mechanical Systems and Signal Processing* **100**, 439–453.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. & Gao, R. X. (2019), 'Deep learning and its applications to machine health monitoring: A survey', *Mechanical Systems and Signal Processing* **115**(8), 1–14.

Zhao, Z., Li, T., Wu, J., Sun, C., Wang, S., Yan, R. & Chen, X. (2020), 'Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study', *ISA Transactions* **107**, 224–255.

Zhu, H., Ting, R., Wang, X., Zhou, Y. & Fang, H. (2016), 'Fault diagnosis of hydraulic pump based on stacked autoencoders', *2015 IEEE 12th International Conference on Electronic Measurement and Instruments, ICEMI 2015* **1**, 58–62.

Zienkiewicz, O. C. (2013), 'The Finite Element Method: Its basis and fundamentals', *The Finite Element Method: its Basis and Fundamentals* p. iii.