UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

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

Insights into the use of Linear Regression Techniques in Response Reconstruction

Master's Dissertation

Student:
Bradley Collins

Student Number:
11202760

February 8, 2021

Abstract

Response reconstruction is used to obtain accurate replication of vehicle structural responses of field recorded measurements in a laboratory environment, a crucial step in the process of Accelerated Destructive Testing (ADT). Response Reconstruction is cast as an inverse problem whereby the desired input is inferred using the measured outputs of a system. ADT typically involves large shock loadings resulting in a nonlinear response of the structure. A promising linear regression technique known as Spanning Basis Transformation Regression (SBTR) in conjunction with non-overlapping windows casts the low dimensional nonlinear problem as a high dimensional linear problem. However, it is determined that the original implementation of SBTR struggles to invert a broader class of sensor configurations. A new windowing method called AntiDiagonal Averaging (ADA) is developed to overcome the shortcomings of the SBTR implementation. ADA introduces overlaps within the predicted time signal windows and averages them. The newly proposed method is tested on a numerical quarter car model and is shown to successfully invert a broader range of sensor configurations as well as being capable of describing nonlinearities in the system.

Acknowledgments

I wish to thank Professor Heyns for the opportunity afforded to me through the CAIM research group. It is through this world class research lab and its team of helpful and insightful technicians that the practical knowledge which informed this dissertation was gained.

Professor Kok, my supervisor, for his technical guidance but more importantly his mentorship during this endeavour. To both Prof. Heyns and Prof. Kok, thank you for your patience and understanding.

I offer my sincerest gratitude to my loved ones, Sydni and my parents, for bearing with me and supporting me through this.

Contents

Acronyms

ADA	AntiDiagonal Averaging
ADT	Accelerated Destructive Testing
APRBS	Amplitude Modulated Pseudo Random Binary Signal
ARX	Autoregressive eXogenous
DOF	Degrees-of-Freedom
FIR	Finite Impulse Response
FIR-DT	Finite Impulse Response with Difference Smoothing and Triangular Weighting
FIR-RR	Finite Impulse Response with Ridge Regression
FIR-T	Finite Impulse Response with Triangular Weighting
LHS	Latin Hypercube Sampling
MFFE	Mean Fit Function Error
NIPALS	Nonlinear Iterative Partial Least Squares
OBF	Orthonormal Basis Functions
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
PLS-SVD	Partial Least Squares by Singular Value Decomposition
POD	Proper Orthogonal Decomposition
RR	Ridge Regression
SBTR	Spanning Basis Transformation Regression
SSA	Singular Spectral Analysis
SVD	Singular Value Decomposition

Chapter 1

Introduction

In the drive for continual improvement in vehicle engineering design, optimized structures and components with lower safety margins and greater reliability are sought [34]. Advances in computational design such as finite element analysis and dynamic modelling combined with fatigue prediction have furthered this goal tremendously in the design phase of development. Nevertheless, there is still a need to dynamically test physical prototypes or existing designs in a controlled laboratory environment. For the analysis to be worthwhile the excitation of the structure in the laboratory environment must induce responses in the structure as though it were being tested in real world operating conditions, the end goal of which is to enable Accelerated Destructive Testing (ADT) of the structure. In ADT the chassis of a vehicle is mounted with its suspension system on a set of hydraulic actuators. The hydraulic actuators then excite the system vertically. Laterally acting forces can also be simulated with additional actuators. An example of an ADT set-up is shown in Figure 1.1.

The excitation of the structure is then carried out for extended periods of time, which allows for the degradation of the structure to be measured in a controlled environment [34]. The structure is not typically excited until catastrophic failure but rather until the degradation in terms of vibration or noise has met a specified threshold [20]. This gives an indication of possible failure points of the system as well as a means of predicting the healthy lifetime of the components. Other insights can be gained from dynamic testing such as better understanding of the dynamics of the system, vibration isolation [34] as well as vibration severities for passenger ride comfort [26].

The core contribution of this dissertation is a new windowing method called AntiDiagonal Averaging ADA which extends capabilities of the Spanning Basis Transformation Regression (SBTR) implementation. This extended linear regression method allows for accurate response reconstruction, a key component of ADT. ADA introduces overlaps within the predicted time signal windows and averages them. The newly proposed method is tested on a numerical quarter car model and is shown to successfully invert a broader range of sensor configurations as well as being capable of capturing nonlinearities in the system.

1.1 Response Reconstruction

The biggest hurdle with ADT is that the inputs to the system such as the displacements or the forces acting on the tyres of the vehicle are difficult or impossible to measure directly in the field. This means the problem must be cast as an inverse modelling or response reconstruction problem [37]. In inverse problems the outputs of the system Z are used in conjunction with model parameters β to determine the inputs U, i.e.

$$U = f(Z, \beta). \tag{1.1}$$

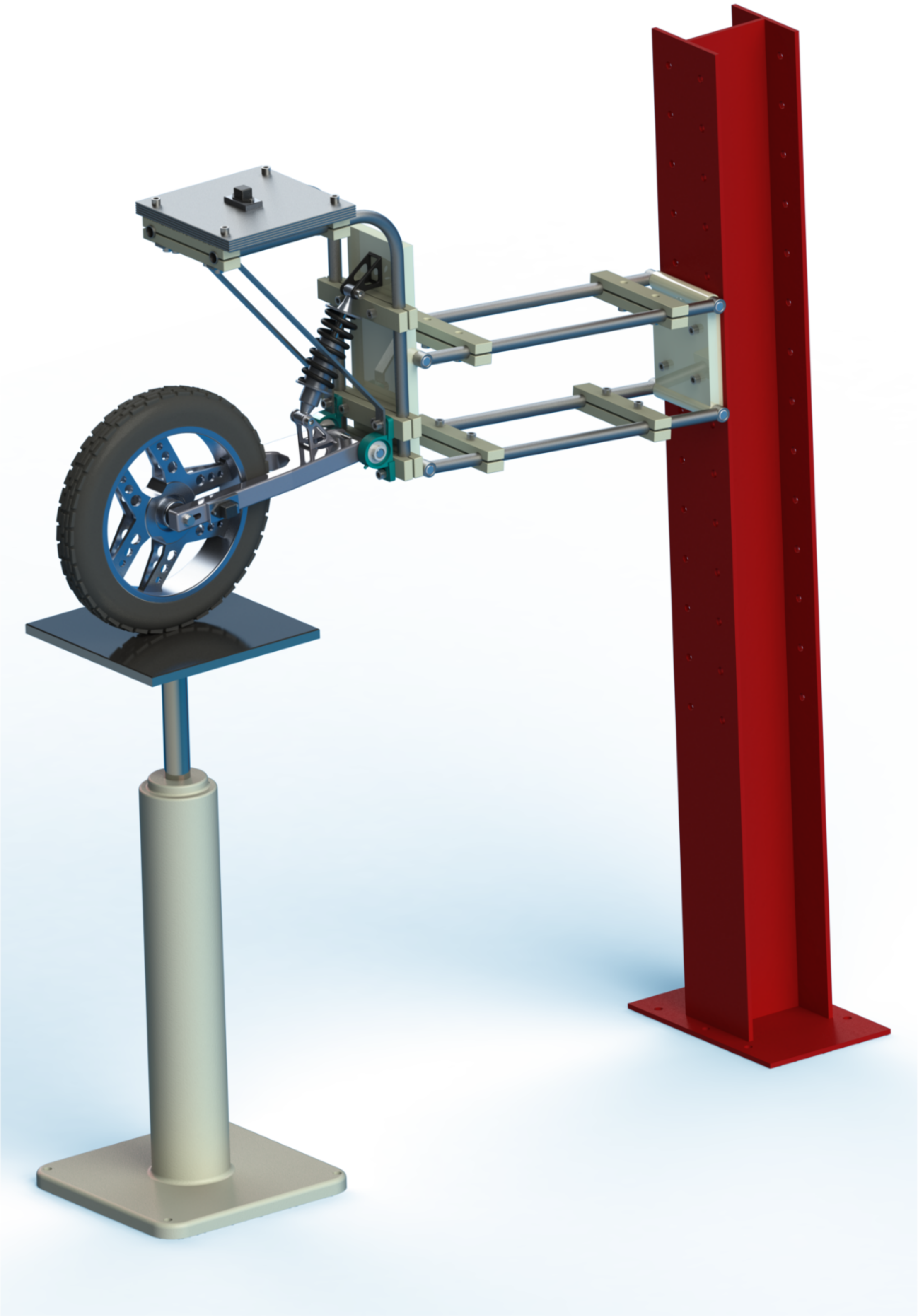


Figure 1.1: An example of an ADT set-up consisting of the rear suspension system of a motorbike. The hydraulic actuator is used to simulate the loads that the motorbike would typically experience in the real world. A range of sensors such as accelerometers and strain gauges are used to capture the dynamic response of the suspension system.

There are two possible choices for creating a model of the system. A mapping of the system can be constructed such that the inputs of the system are used to predict the outputs of the system. This is referred to as the forward problem. The forward problem is then inverted. If the model used to map the problem is nonlinear, an iterative optimization scheme is employed to invert the system. However, the optimization scheme may be prone to local minima.

The second approach is to create a direct inverse of the model whereby the outputs of the system are used to predict the inputs of the system. The inverse method has an inherent stability check since the solution will only be obtained if the direct inverse model is stable [34]. A graphical overview of the two approaches to the inverse problem is shown in Figure 1.2. An overview of the response reconstruction methodology is given in Figure 1.3.

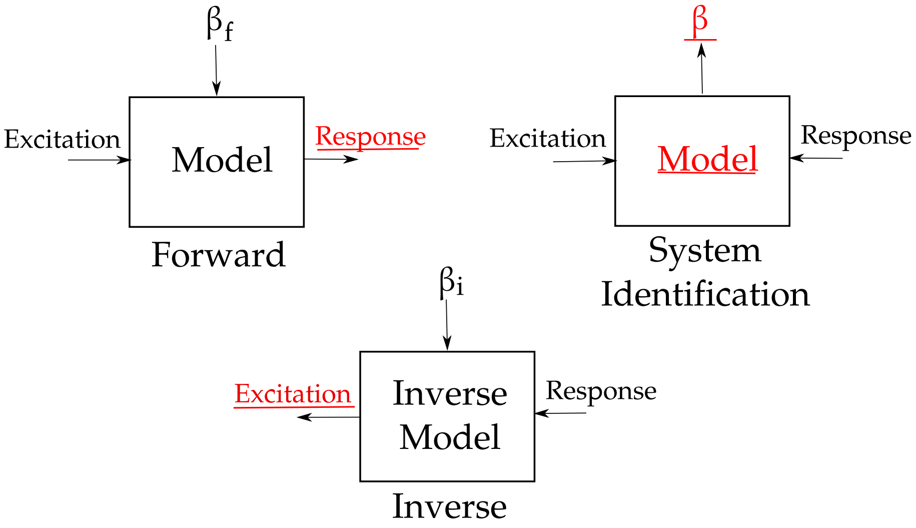


Figure 1.2: Inverse problem overview. In the forward problem we require a set of parameters β_f that when combined with a known set of inputs allow us to predict the unknown response of the system. System identification techniques can be employed to determine β_f by using a set of known inputs and outputs. In the response reconstruction problem we have a known set of outputs but do not know their corresponding inputs. Therefore a set of parameters β_i is required that maps the outputs to the inputs. There are two possible ways of solving this problem, either by first mapping the forward problem using system identification and then inverting the solved forward problem or by using system identification to directly map the known outputs to the inputs. Figure adapted from Uhl [37].

1.2 Non-minimum Phase Systems

For linear time invariant single input single output systems, the forward problem can be written in the Laplace domain as a transfer function $H(s)$ in terms of the transformed output $Z(s)$ and input $U(s)$ such that

$$H(s) = \frac{Z(s)}{U(s)}. \tag{1.2}$$

The transfer function can then be interpreted in terms of its poles and zeros. The poles of the transfer function are the complex values of s that cause the denominator to go to zero and hence the transfer function to approach infinity. To ensure that the system is stable we require that the poles sit in the left-hand side of the complex plane. Stability in this context is defined as a bounded input only resulting in a bounded output. The zeros of the system are the values of s that cause the numerator to approach zeros and thus cause the transfer function to approach

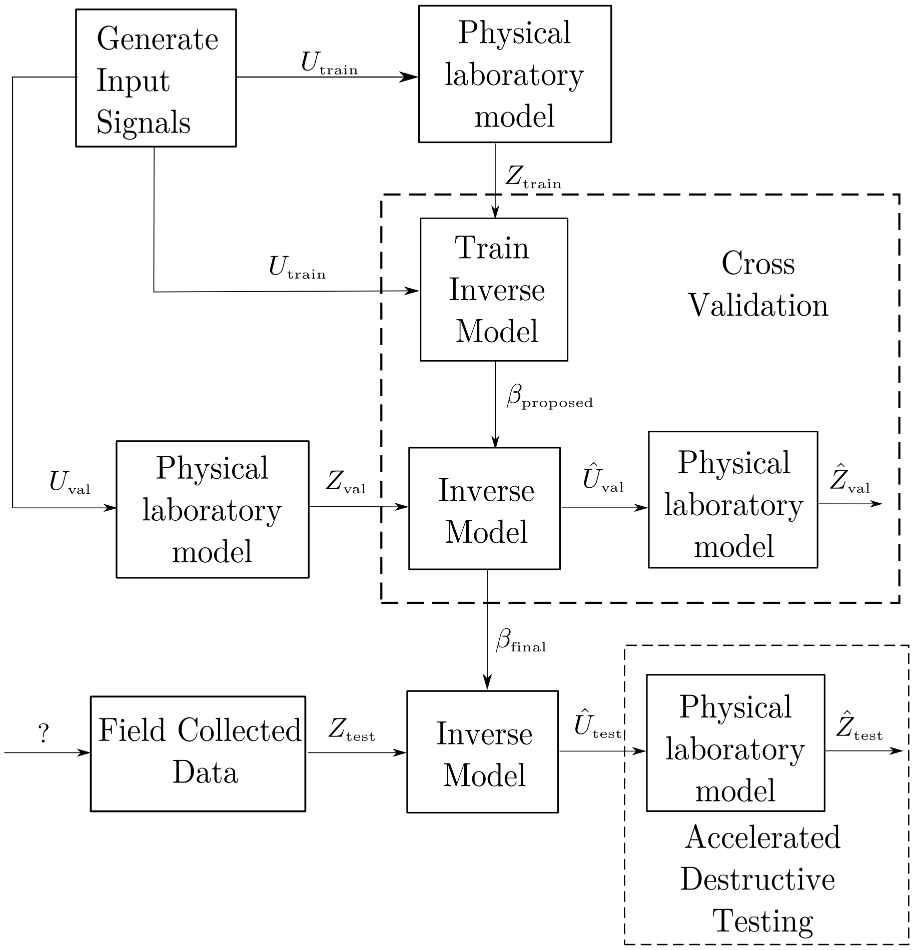


Figure 1.3: Response Reconstruction overview. In the initial phase of response reconstruction, a set of input signals U_{train} are designed in such a manner that they excite the desired dynamics of the system. The choice of the excitation signal is given in Section 4.1. The laboratory test rig is then excited by the inputs to obtain the corresponding outputs Z_{train}. With these known inputs and outputs, an inverse model of the system can be mapped. Direct inverse system identification is used to obtain the model parameters β_{proposed}. Through the course of this dissertation, we will cover why we cannot directly use the model parameters β_{proposed} as will need to employ regularisation. A method of determining the amount of regularisation is cross-validation, which will be covered in Section 4.2. As we will see in Section 2.3, the input U is not unique for a given output Z, therefore we cannot directly compare the reconstructed input $\hat{U}_{\text{val}}$ against the known input U_{val}. Instead, we pass the reconstructed input through the physical model to obtain the reconstructed output $\hat{Z}_{\text{val}}$ which allows for direct comparison against the known output Z_{val}. The discrepancy between Z_{val} and $\hat{Z}_{\text{val}}$ is what we are trying to minimize in the cross-validation step. This means that the cross-validation step requires a physical forward pass through the physical laboratory model. The field collected data of the system Z_{test} (for which we do not know the true inputs U_{test}) can then be inverted to obtain an approximation of the real world input $\hat{U}_{\text{test}}$ given the final set of model parameters β_{final}. The approximated input can now be used to recreate an approximation to the real world response $\hat{Z}_{\text{test}}$ by using the inputs to excite the laboratory test rig. This final input can then be repeated indefinitely for ADT. An important distinction to make here is that even though we are employing inverse methods, we are not particularly interested in the inputs themselves. We are instead interested in the quality of the reconstructed responses.

zero. If we instead rewrite the transfer function such that it maps the outputs to the inputs, i.e. the inverse function H^{-1}

$$H^{-1}(s) = \frac{U(s)}{Z(s)}, \tag{1.3}$$

it becomes clear that the roles of the poles and zeros in the original formulation have now switched. This means that if we require the inverse problem to be stable we require the zeros of the original forward problem to lie in the left-hand plane. Systems that do not exhibit this property are said to be non-minimum phase. A large proportion of systems encountered in response reconstruction are found to be of this nature. The interpretations of poles and zeros do differ for multiple input multiple output and nonlinear systems, however the concept of pole-zero reversal during inversion can be extended to these cases.

In order to obtain an exact inversion of a non-minimum phase system in a bounded time interval, the inversion needs to be performed non-causally. This involves reversing the ordering of the input and output signals [17, 34]. This can only occur in an offline fashion. Fortunately, ADT typically consists of short repeated test signals which are amenable to offline non-causal control.

1.3 System Identification

The forward or inverse system can be identified using three conceptual approaches: white-box, grey-box and black-box modelling [32]. White-box uses first principles physical laws combined with prior knowledge to model the system. Black-box modelling uses general mathematical models to approximate the dynamics of the system. Grey-box modelling is a hybrid of the two methods whereby the first principle approach is augmented by mathematical models to fill in the gaps of the knowledge. This is clearly a spectrum of design approaches. In this dissertation we will be focusing on black-box approaches.

Most common reconstruction techniques are implemented in the frequency domain [1], whereby the discrete Fourier response is multiplied by an inverse or pseudoinverse frequency response function [35]. Raath's Ph.D. thesis [34] highlighted the then known issues of using frequency response techniques in accelerated fatigue testing. It was shown that the frequency response was inaccurate for a number of reasons, that include:

- assuming that the input and output signals are periodic when often they were not. These include sharp impulses from random impacts.
- being unable to model nonlinear models since frequency response analyses assume a linear model.
- requiring long time signals of the order of hours as opposed to minutes or seconds needed for the time domain. This ties in with the issue that low frequency information is easily lost due to spectral leakage where the energy in the lower frequencies is spread over to higher frequencies.
- failing to capture the sequence or causal effects which play an important role in crack propagation

Various time domain techniques have been developed to overcome this, however, they have been shown to be slow or inaccurate [17]. The vehicle structures of interest typically contain many nonlinear components such as springs and pneumatic dampers. Typical control systems will overcome this issue by linearising the system around the operation point. It is then assumed that the system will experience small perturbations around this point. However, in this case it is expected that the system will experience impact loadings and large displacements, which will force the system out of its linear region [34].

1.4 Regularization

For a problem to be well-posed it needs to meet the following criteria [8]:

- The solution is unique.
- A global solution exists for all data.
- The solution to the problem is continuously dependent on the given data.

However, we quickly find that most inverse problems are ill-posed. The first criterion is normally the offending culprit since its easy to construct a forward problem where two different inputs result in the same output. Therefore, the inverse solution is typically not unique. If the problem is ill-posed we may use regularisationtechniques to cast the problem as a more well-behaved problem. These regularisation techniques include [37]:

- Cross validation
- Singular value decomposition
- Iterative methods
- Data filtering
- Tikhonov regularisation.

Singular value decomposition discards small singular values that may blow up when the system is inverted and ensures that the condition number of the system is not too large [29]. A similar technique is used in data filtering where high-frequency signals, that introduce large errors when the system is inverted, are filtered out.

1.4.1 Sources of Uncertainty

In system identification there are two distinct sources of uncertainty when mapping the model of the experiment which contribute to the ill-posedness of the problem. These are aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty is due to random noise that will always be present when measuring the response of the system. Epistemic uncertainty is due to the difference between the real world model that we wish to recreate and the experimental model that we use to represent the real world model [2].

Measurement Error

During training the input signal will be generated by the computer controlling the hydraulic actuators. Therefore, the signal will be deterministic and free of any noise. However, the hydraulic actuators may not necessarily follow the input signal and are subject to noise. This will manifest itself as noise in the output signals and as such is treated as output noise [34].

Model Error

Model uncertainty is due to the mismatch between the experimental set-up in the laboratory and the real world conditions the vehicle will experience. Typically the degrees of freedom of the system are limited in the laboratory environment since it would be physically impractical to have actuators exciting all degrees of freedom a vehicle may encounter in the real world. Model mismatch can be interpreted as a form of extrapolation. The reconstruction algorithms used are making predictions on signals taken from real world experiments in the testing phase which lie outside the laboratory domain on which the inverse model was trained on. It has been demonstrated that regularisation can be used to remedy the effects of model mismatch [2]. By using regularisation we ensure that latent variables that describe both real world and experimental models are used.

1.5 Comparison to Force Identification

A related field to response reconstruction is force identification whereby the inputs of the system are of interest. However, we know that the inputs of the system are not unique. Force identification tackles this problem by enforcing some prior knowledge of the system dynamics to constrain the inputs to reasonable solutions. Bayesian methods have become prevalent in force identification literature since they allow the experimenter to incorporate prior knowledge in a systematic manner [27]. Another benefit of incorporating Bayesian methods is it allows for confidence intervals on the input predictions and model parameters [3].

A noticeable distinction in force identification literature is that a known finite element model of the structure is typically assumed, i.e. a known forward model.

The approach taken in this dissertation of solving the issue of non-uniqueness of the input is mitigated by

- By not focusing on the reconstructed input accuracies.
- Using cross-validation of the reconstructed outputs of the system to determine whether a given inverse model of the system is satisfactory i.e. using a forward pass through the physical system in each cross validation step to determine the model accuracy.

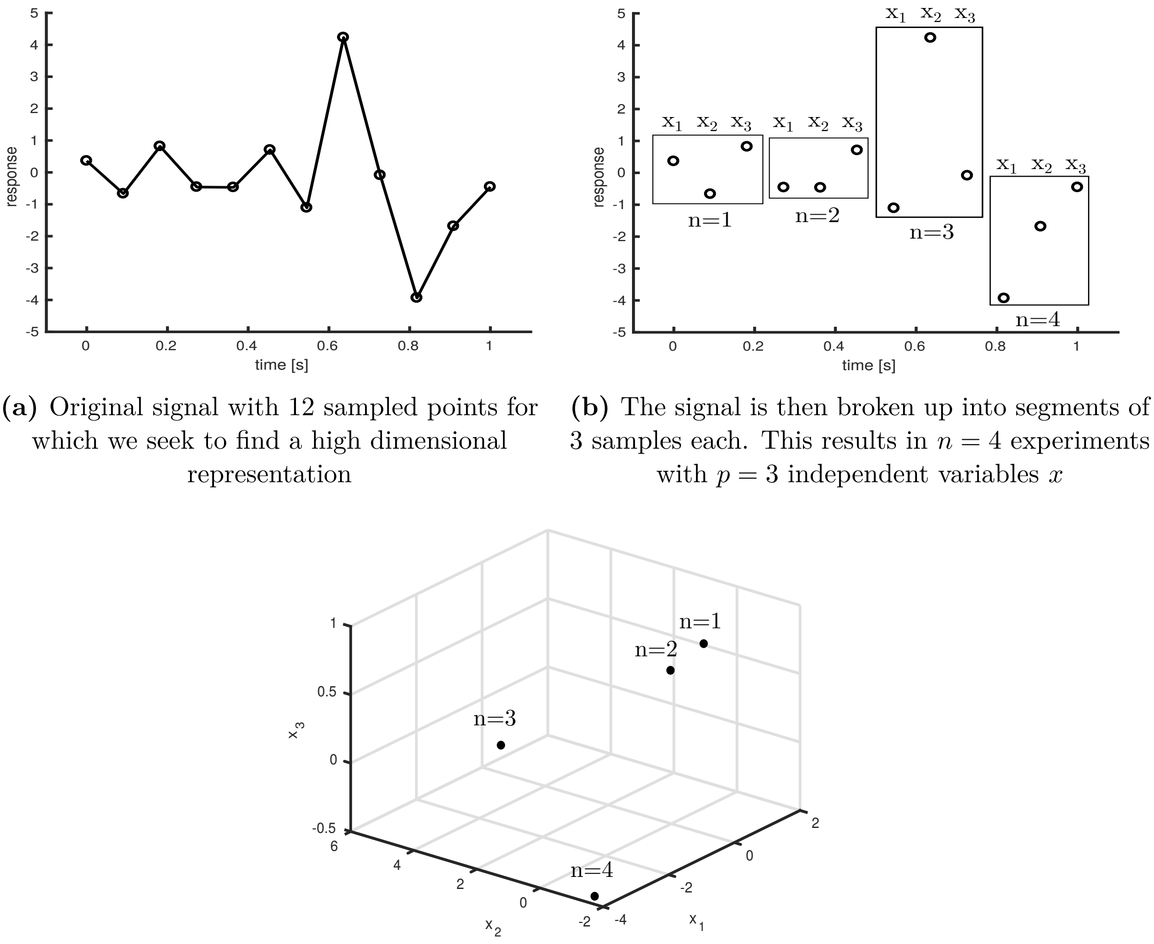
A potential drawback to our approach is that, if implemented naively, the cross-validation can induce undue stress on the system before any ADT can take place.

1.6 Spanning Basis Transformation Regression (SBTR) for Time Domain System Identification

A new method of creating a direct inverse model of a dynamic system was developed by Crous et al. [12]. The methodology was shown to produce high response reconstruction accuracy with low computational cost while using minimal experimental data. The mapping method is based on SBTR which is a linear mapping regression method that is efficient and accurate in solving high dimensional regression problems. The high dimensionality is needed in order to create a flexible linear model that can approximate the nonlinearities of the system. The method can be broken down into three broad steps:

- **Time series representation**: A method of representing the time series in high dimensional space is needed. To this end a series of non-overlapping windows were chosen to represent the inputs and outputs of the system. A graphical overview of this process is shown in Figure 1.4.
- **Regression method**: A means of inferring the inputs using the measured outputs are sought. SBTR was used in order to build a model of this mapping that could deal with the high dimensionality of the problem as well as collinearity typically encountered in system identification.
- **Active learning**: With active learning further training signals are sought that allow for further improvement of the regression model. A method called generalised bootstrapping was used to find novel test signals that further enriched the regression models.

In the original implementation of SBTR with non-overlapping windows [12], the system was validated against a numerical test rig that used the absolute output displacements of the system to infer the inputs. The absolute displacements of the test rig are rarely available due to the impracticality of directly measuring them in operational conditions. Therefore, the most pertinent issue is to validate the SBTR methodology on a more realistic sensor set-up.



(a) Original signal with 12 sampled points for which we seek to find a high dimensional representation

(b) The signal is then broken up into segments of 3 samples each. This results in $n = 4$ experiments with $p = 3$ independent variables x

(c) The $n = 4$ experiments can then be represented as points in 3 dimensional space using the $p = 3$ independent variables as co-ordinates

Figure 1.4: An overview of representing signals in high dimensional space using windowing techniques

1.7 Non-overlapping Windows

Non-overlapping windows is the method originally used in the implementation of SBTR [12]. The original input and response measurements are given by $U \in \mathbb{R}^{m \times q}$ and $Z \in \mathbb{R}^{m \times o}$ where m is the original sequence length in samples and q and o are the number of actuator and sensor channels respectively. The original input signal is thus represented in matrix form as

$$U = \begin{bmatrix} u_1(1) & \cdots & u_{q-1}(1) & u_q(1) \\ u_1(2) & \cdots & u_{q-1}(2) & u_q(2) \\ \vdots & \ddots & \vdots & \vdots \\ u_1(m-1) & \cdots & u_{q-1}(m-1) & u_q(m-1) \\ u_1(m) & \cdots & u_{q-1}(m) & u_q(m) \end{bmatrix}. \tag{1.4}$$

The windowing principle can be used to represent the responses Z as the predictor matrix $X \in \mathbb{R}^{n \times p}$ and the inputs U as the target matrix $Y \in \mathbb{R}^{n \times r}$. The number of observations that the original signal is split into, n, is given by

$$n = \left\lfloor \frac{m}{s_w} \right\rfloor, \tag{1.5}$$

where s_w is the window sample length given by the desired window length in seconds T_w multiplied by the sampling frequency f_s

$$s_w = \lfloor f_s T_w \rfloor. \tag{1.6}$$

This occurs for each time sequence for either an actuator or sensor signal, being appended column-wise, resulting in

$$p = s_w \times o, \tag{1.7}$$

$$r = s_w \times q. \tag{1.8}$$

The final target matrix Y therefore takes on the form

$$Y = \left[\begin{array}{cccccc} \multicolumn{5}{c}{\overbrace{\hspace{12cm}}^{\text{channel 1}}} & \overbrace{\hspace{1cm}}^{\text{channel 2 etc.}} \\ u_1(1) & u_1(2) & \cdots & u_1(s_w-1) & u_1(s_w) & \cdots \\ u_1(s_w+1) & u_1(s_w+2) & \cdots & u_1(2s_w-1) & u_1(2s_w) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ u_1((n-2)s_w+1) & u_1((n-2)s_w+2) & \cdots & u_1((n-1)s_w-1) & u_1((n-1)s_w) & \cdots \\ u_1((n-1)s_w+1) & u_1((n-1)s_w+2) & \cdots & u_1((n)s_w-1) & u_1((n)s_w) & \cdots \end{array}\right]. \tag{1.9}$$

The predictor matrix X takes on a similar shape (not shown). The windowing process can easily be reversed to obtain the original input and output matrices but since the number of rows n were rounded down the reconstructed signal is shortened slightly. A graphical of the non-overlapping windowing process is shown in Figure 1.5.

1.8 Scope of Research

The goal of this research is to build on the groundwork of SBTR in the space of response reconstruction. The going concern is that the methodology, as is, cannot be used directly on

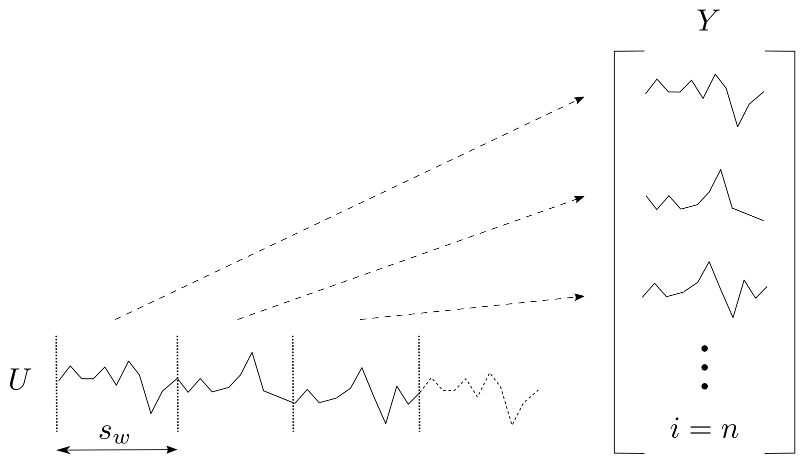


Figure 1.5: An overview for the non-overlapping windowing process. A single actuator signal is shown.

the sensor signals obtained from an experimental rig. This is due to impracticalities of directly measuring absolute displacements. A new methodology will be proposed that uses the windowing method as well as the linear regression concepts used in SBTR but is better suited to a wider range of problems found in response reconstruction. To test these concepts a numerical quarter car model will act as our test rig throughout this dissertation.

1.8.1 Suitability as a black-box model

One possibility of ensuring that the methodology works is to numerically integrate the available signals such that the absolute displacements are obtained. The issue of noise is exacerbated when double integrating acceleration signals resulting in sensor drift. Methods such as Kalman filters can be used to mitigate this [34, 28]. This requires some human intervention as well as prior knowledge of the dynamics of the system to implement.

The proposed methodology will need to be able to handle non-minimum phase systems. As highlighted in the literature review it is known that inverting a non-minimum phase system requires that the system be inverted non-causally. This relates to how the sensor signals are presented to the regression technique. Therefore, the representation of the windows and how they relate to non-minimum phase systems will be explored in this dissertation.

The main goal of this dissertation is to develop a stand-alone black-box methodology that requires minimal prior knowledge to work and is suited to a wider range of response reconstruction problems.

1.8.2 Hyper-parameter investigation

It is to be expected that a number of hyper-parameters and design choices will be required to properly set-up the proposed response reconstruction methodology. These could be determined by running a series of experiments before starting the ADT process. However, we want to avoid unnecessary testing of the system before we even begin ADT. Therefore, we need to investigate methods whereby we can relate these hyper-parameters to easily determined physical constants of the system. To achieve this we will parameterise the physical constants of the quarter car model to determine the relationship between them and the hyper-parameters.

1.8.3 Linear regression performance in terms of response reconstruction

Since SBTR was only used with non-overlapping windows it needs to be tested with the newly proposed windowing methodology to determine whether it confers any specific advantages with regards to response reconstruction. It will be compared against related linear regression methods. In order to measure this, SBTR and related linear regression techniques will be tested in terms of their ability to handle model mismatch, measurement noise and as well as their ability to handle non-linear dynamics.

1.9 Document Overview

An overview of the numerical quarter car model set-up used to investigate the methodology is given in Chapter 2. A derivation and theoretical comparison of related linear regression methods are given in Chapter 3. The methodology used to determine the parameters that govern the windowing and regression methods is outlined in Chapter 4. The shortcomings of non-overlapping windows are highlighted and demonstrated in Section 1.7. The idea of overlapping windows is then introduced to overcome these shortcomings in Sections 5.2 and 5.3. The newly proposed windowing technique is then benchmarked using a broader set of sensor configurations (Sections 5.6) and using a non-minimum phase configuration (Section 5.7). The quarter car model is then parameterised in order to determine how the physical properties of the test rig influence the design choice of the windowing parameters in Chapter 6. Chapter 7 focuses on the performance of the linear regression methods themselves and how they perform in terms of mapping nonlinearities (Section 7.1), robustness to measurement noise (Section 7.2) and handling model mismatch (Section 7.3). An analysis of the regression performance of SBTR in terms of response reconstruction is given in Section 7.4. The proposed methodology is then compared against related system identification techniques in Section 7.5. The final conclusion of the document is then given in Chapter 8.

Chapter 2

Numerical Quarter Car Model

In order to investigate the methods explored in this dissertation a simple two-degree-of-freedom nonlinear mass-spring-damper system representing a quarter car model will be used. The numerical model employed is shown schematically in Figure 2.1. The sprung mass M_A and unsprung mass M_R capture the mass of the body of the vehicle and the suspension-tyre system respectively. These bodies are connected by springs and dampers which represent the dynamics of the suspension system. The unsprung mass is then connected to the road via a spring characterising the tyre stiffness. The system is excited by a road profile u_{road}.

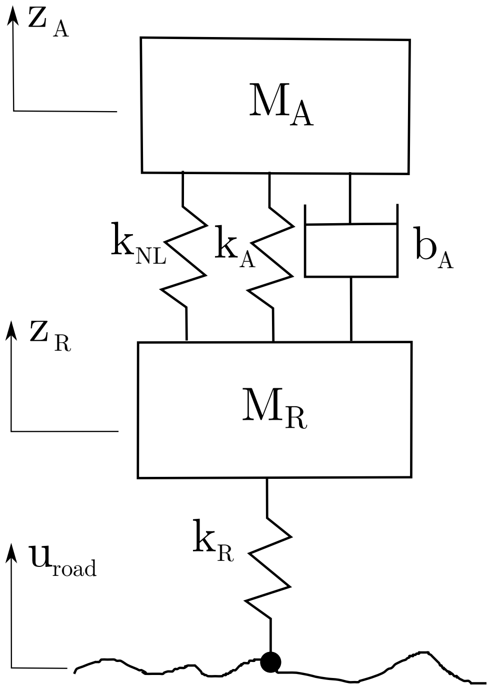


Figure 2.1: Two degree-of-freedom mass-spring-damper representation of the nonlinear quarter car model.

The system behaves according to the following equations of motion:

$$\begin{aligned} \ddot{z}_A &= -\frac{b_A}{M_A}(\dot{z}_A - \dot{z}_R) - \frac{k_{NL}}{M_A}(z_A - z_R)^3 - \frac{k_A}{M_A}(z_A - z_R), && (2.1) \\ \ddot{z}_R &= +\frac{b_A}{M_R}(\dot{z}_A - \dot{z}_R) + \frac{k_{NL}}{M_R}(z_A - z_R)^3 + \frac{k_A}{M_R}(z_A - z_R) - \frac{k_R}{M_R}(z_R - u_{road}), && (2.2) \end{aligned}$$

where the k and b terms are the stiffness and damping coefficients respectively. The nonlinearity is introduced by having cubic stiffening of the sprung mass spring which is controlled by the

k_{NL} term. This gives the following description of the sprung mass spring force, f_A,

$$f_A(\Delta_z) = -k_A\Delta_z - k_{NL}\Delta_z^3, \tag{2.3}$$

where we define a new state of the system representing the deflection of the spring, Δ_z, such that

$$\Delta_z \equiv z_A - z_R. \tag{2.4}$$

The k_{NL} term can be varied to change the severity of the nonlinearity of the system or switched off completely to have the system behave linearly. A hardening spring can be chosen by choosing $k_{NL} > 0$. This results in a spring that becomes stiffer as it goes into compression or tension. Likewise, a softening spring can be implemented by choosing $k_{NL} < 0$. In this study the linear component will always be restorative such that $k_A > 0$. An example of the force profiles for cubic springs, for the softening and hardening cases, is shown in Figure 2.2.

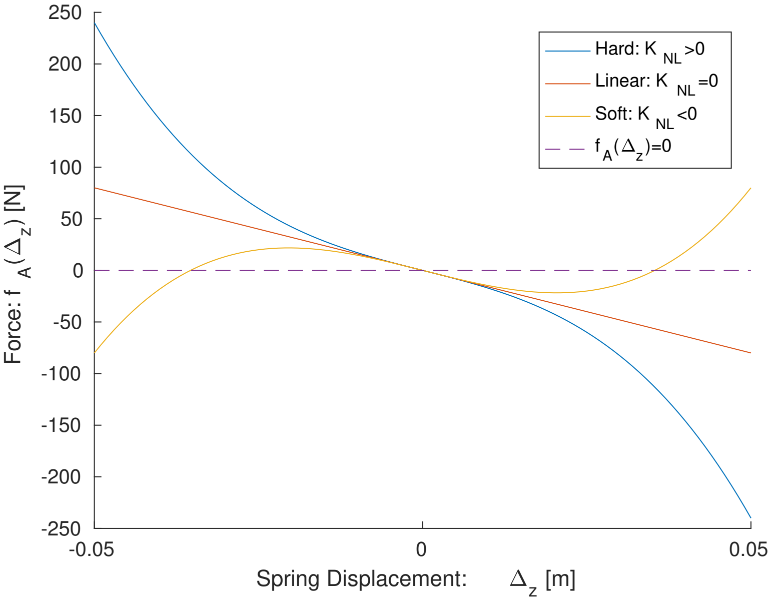


Figure 2.2: Force profiles of cubic stiffening springs. The force of hardening and softening springs are compared against that of a linear spring.

If we consider the equilibrium points of the spring system where $f_A(\Delta_z) = 0$, i.e.

$$\Delta_z(-k_A - k_{NL}\Delta_z^2) = 0, \tag{2.5}$$

we obtain 3 solutions for equilibrium points, namely: $\Delta_z = 0$ and $\Delta_z = \pm\sqrt{\frac{-k_A}{k_{NL}}}$. For the hardening case and restorative linear spring, where $k_{NL} > 0$ and $k_A > 0$, there is only one real solution at $\Delta_z = 0$. In the case where the linear spring is not restorative but still hardening i.e. $k_{NL} > 0$ and $k_A < 0$, all 3 solutions are valid. In the non-restorative case, $\Delta_z = 0$, is an unstable equilibrium point with $\Delta_z = \pm\sqrt{\frac{-k_A}{k_{NL}}}$ being stable equilibrium points. By having two stable equilibrium points, the system can readily switch between oscillating around two equilibrium points causing highly chaotic behaviour of the system. Chaotic systems are outside of the scope of this study and for this reason the problem will be limited to the restorative hardening spring case which has one stable equilibrium point at $\Delta_z = 0$.

In practical applications not all states of the system are directly measurable. Typically the accelerations of the bodies or the strains of the system or the combination of the two are measured. Therefore, the default realistic observable states, $\mathbf{z}$, will be chosen as

$$\mathbf{z} = [\ddot{z}_A, \Delta_z]^T, \tag{2.6}$$

where the strains of the system are approximated by the displacement of the spring Δ_z. However, throughout this dissertation a number of different sensor configurations will be employed.

The model will be simulated using *Simulink*. The solver type will be set to automatic with variable-step size. The results are interpolated by *Simulink* to obtain results sampled at the specified sampling frequency. The default parameters chosen for the numerical quarter car model are given in Table 2.1. From these default values the parameters of the quarter car will be changed depending on the investigation underhand.

Table 2.1: Numerical quarter car model default parameter values and how the system will be parameterised depending on the investigation underhand.

Parameter	Default Value	Investigation
$\mathbf{M_A}$	70 kg	Model mismatch in Section 7.3
$\mathbf{M_R}$	12 kg	Model mismatch in Section 7.3
$\mathbf{k_A}$	1.6×10^3 N m^{-1}	Model mismatch in Section 7.3.
$\mathbf{k_R}$	80×10^3 N m^{-1}	Model mismatch in Section 7.3. Influence on triviality for the displacement case in Section 5.1.1. Sampling frequency influence in Section 6.1
$\mathbf{k_{NL}}$	12.8×10^6 N m^{-3}	Set to zero when dealing with linear case throughout. Regression performance w.r.t. nonlinearity in Section 7.1
$\mathbf{b_A}$	500 N s m^{-1}	Model mismatch in Section 7.3. Window length influence in Section 6.2

2.1 Nonlinearity

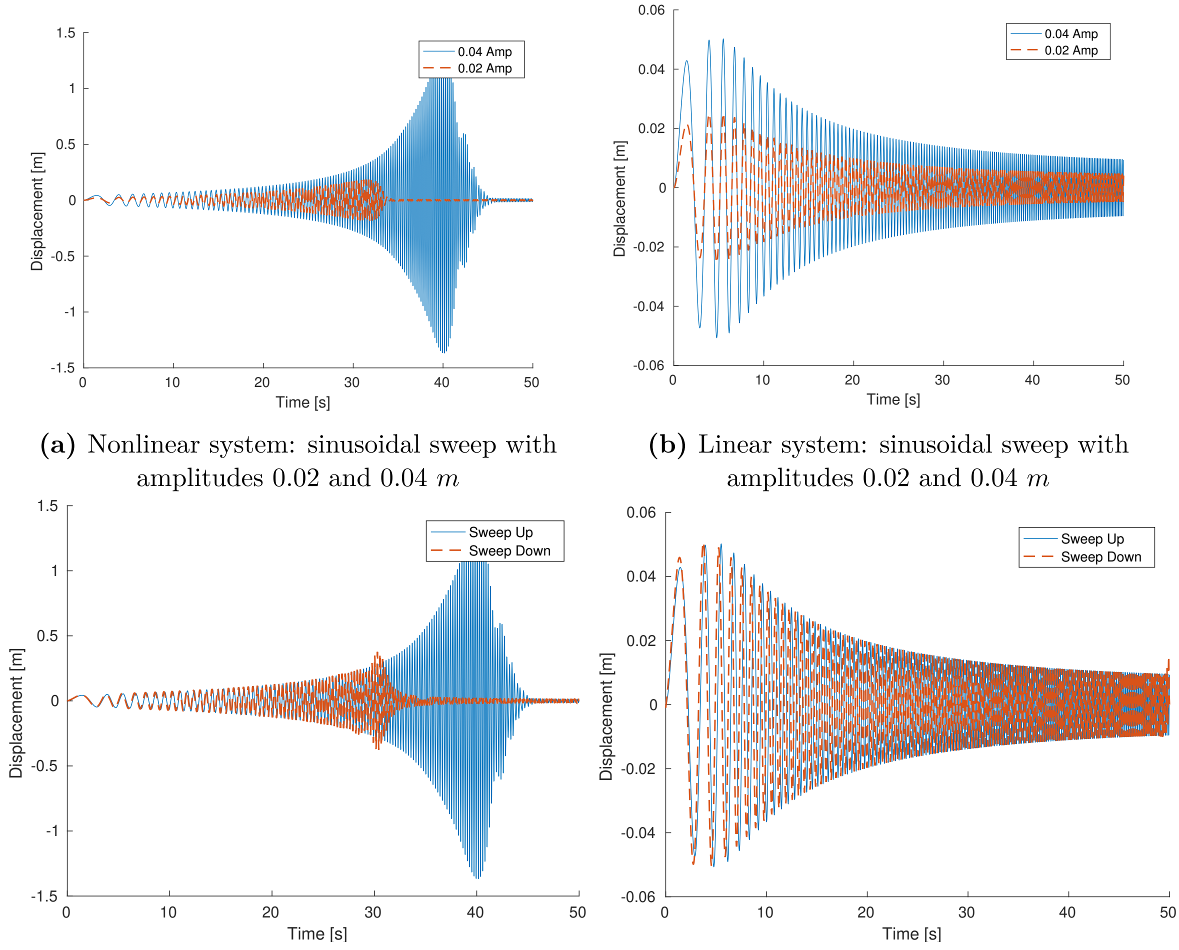
The definition of nonlinearity used in this dissertation is any system that does not follow the principle of superposition. This vague description arises since the familiar expectation that a system behaves linearly is a specific but useful assumption that is employed regularly in engineering. However, there are many phenomena that may cause a system to behave nonlinearly. Researchers may focus on specific cases of nonlinear dynamics but no general definition exists that describes all cases of nonlinearity [5]. The definition chosen here, nonetheless, does allow us to easily visually identify whether a system behaves nonlinearly. A visual method of identifying nonlinearity in a dynamic system [33] is demonstrated by means of the simulated nonlinear quarter car model.

2.1.1 Testing for Nonlinearity

A sinusoidal linear sweep function is used to excite the nonlinear quarter car model covered in Section 2. The sweep input as a function of time $u(t)$ is given as

$$u(t) = A \sin\left(2\pi \left(f_0 + \frac{f_1 - f_0}{T} t\right) t\right), \tag{2.7}$$

where A is the amplitude of the signal, f_0 and f_1 are the initial and final frequency values and T is the timespan over which the sweep occurs. The chosen frequency sweep covers the range from 0.1 to 6 Hz over a period of 50 s. By changing the amplitude from 0.02 to 0.04 m we can note that the envelope of the sprung mass displacement response changes drastically for the nonlinear system as shown in Figure 2.3a whereas the output for the linear system is simply a scaling of the smaller amplitude output as seen in Figure 2.3b. In the second test the amplitude is fixed at 0.04 m but the sequencing of the sweep is reversed. The sweep is firstly swept upwards and then downwards in term of frequency over time. The downward sweep response is then reversed and overlaid on the upward sweep response. In the linear case, shown in Figure 2.3d, the sequence effect has no noticeable effect on the shape of the envelope, whereas for the nonlinear system in Figure 2.3c, we note that the response is heavily distorted by the sequencing of the input. This demonstration further emphasizes the need for accurate response reconstruction in the time domain since both sinusoidal sweeps have the same spectral content but resulted in completely different outputs in the nonlinear case. This method also demonstrates the means of identifying whether the system is nonlinear, but not necessarily the cause of the nonlinearity.



(a) Nonlinear system: sinusoidal sweep with amplitudes 0.02 and 0.04 m

(b) Linear system: sinusoidal sweep with amplitudes 0.02 and 0.04 m

(c) Nonlinear system: increasing and decreasing sinusoidal sweep

(d) Linear system: increasing and decreasing sinusoidal sweep

Figure 2.3: Output responses comparing the tell-tale signs of a nonlinear system to that of the more well behaved linear system

2.2 Forward Problem Validation

The purpose of this section is to verify that the model parameters that we will be using are well-posed for the forward problem. The reasons for this are two-fold. Firstly, we want to ensure that the model parameters are sufficiently nonlinear but do not experience chaotic behaviour. The second reason is to ensure that the employed integration methods are suitable for the problems at hand. We can test whether the system is well-posed by the employing the definition of a well-posed problem. The first requirement is that a unique solution exists. Secondly, any small perturbation to the problem statement or the initial condition of the system should result in a bounded error between the perturbed and the unperturbed system.

To test the first requirement we can verify whether the error norm converges as expected when the step size h of the solver is decreased. To test this the fourth order Runge-Kutta solver will be employed which has a local truncation error $O(h^4)$ [8]. The solver step size, h, will be decremented using $h_1 = 5 \times 10^{-4}$, $h_2 = 2.5 \times 10^{-4}$ and $h_3 = 1.25 \times 10^{-4}$. Since the step size is halved at each iteration, the error norm ratio of the spring displacement outputs is expected to converge to 16 over consecutive step sizes. The error norm ratio is defined as

$$e_{\text{norm ratio}} = \frac{||z_2 - z_1||_2}{||z_3 - z_2||_2}. \tag{2.8}$$

Two sets of the convergence test will be performed. The first test will be performed over a range of damping coefficients b_A and system nonlinearities k_{NL}. The second test will be performed by varying the tyre stiffness values k_R and system nonlinearities k_{NL}. The range of parameter values are given in Table 2.2. In order to avoid using a fixed step size solver throughout this dissertation, MATLAB's auto-solver with variable step size will be benchmarked alongside the Runge-Kutta solver. The Amplitude Modulated Pseudo Random Binary Signal (APRBS) outlined in Section 4.1.1 is used to excite the system.

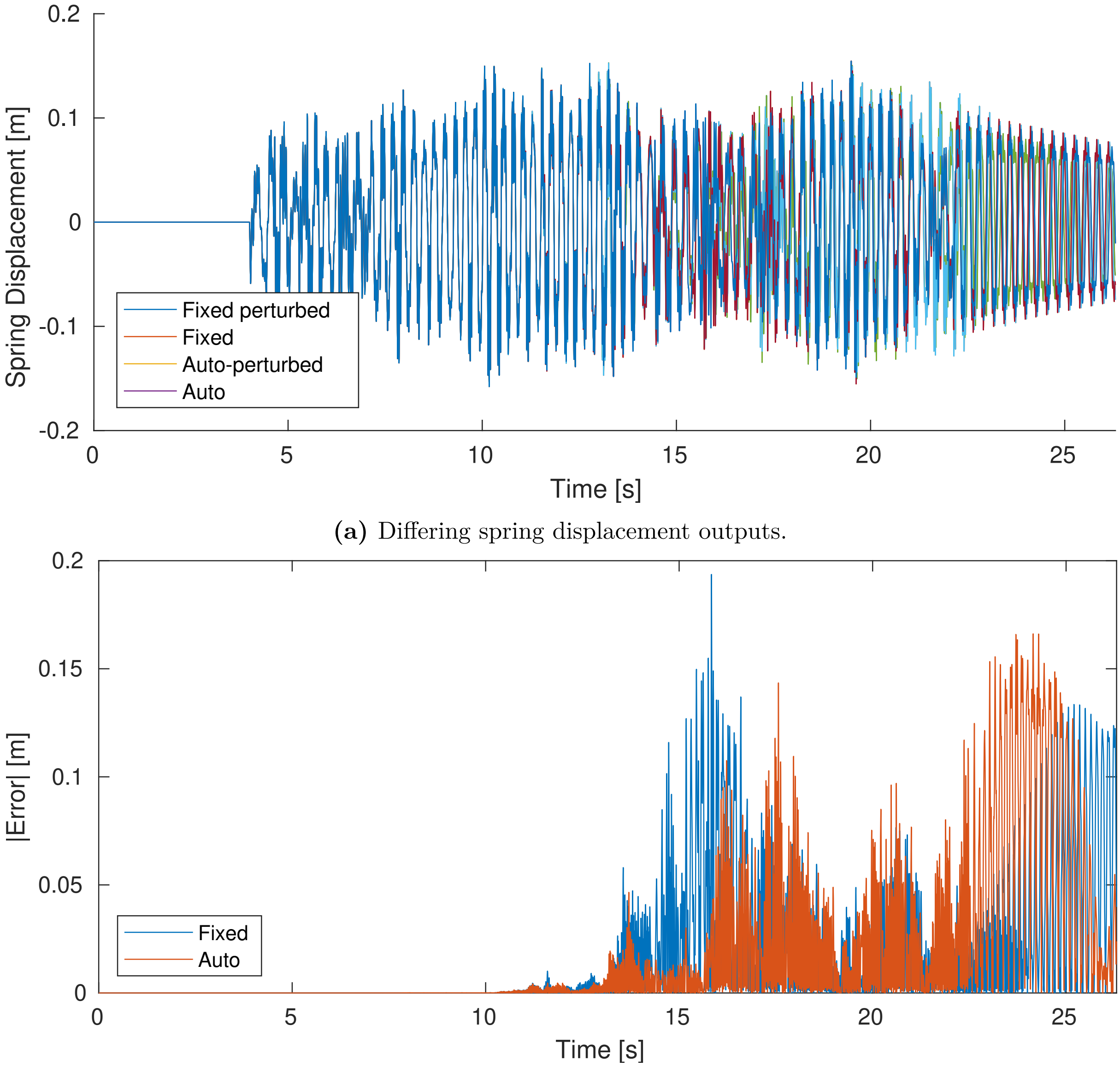
Table 2.2: Quarter car parameter values used to test the well-posedness of the forward problem.

Variable	Details
k_{NL}	1.28×10^n N m^{-3} where $n = [4, 5, 6, 7, 8]$
k_R	$\in [5, 10, 20, 40, 80] \times 10^3$ N m^{-1}
b_A	$\in [50, 2000]$ N s m^{-1} with 30 equal divisions

In order to test the second requirement, a small perturbation to the road profile of 0.0001 m with a duration of 0.001 s will be added at $t = 0$. The error norm, defined as

$$e_{\text{perturb}} = ||z_{\text{perturbed}} - z_{\text{unperturbed}}||_2, \tag{2.9}$$

between the perturbed and unperturbed spring displacement output will be then be compared. As before this will be tested over a range of damping coefficient and system nonlinearities as well as a range of tyre stiffness values and system nonlinearities. The Runge-Kutta solver will be used with step size $h = 1.25 \times 10^{-4}$. To visually demonstrate the effects of the initial perturbation for a numerically ill-behaved problem, the output of the nonlinear system with $k_{\text{NL}} = 1.28 \times 10^7$ N m^{-3} and damping coefficient $b_{\text{A}} = 75$ N s m^{-1} is shown in Figure 2.4a. Note that the signal has a 4 s dead time. The perturbation occurs at the beginning of the dead time. The corresponding absolute error plot is shown in Figure 2.4b. At roughly $t = 11$ s we note that the system response starts diverging even though only a minute perturbation was introduced right in the beginning. This is an indication that the system or the solver is not well-behaved with regards to this configuration.



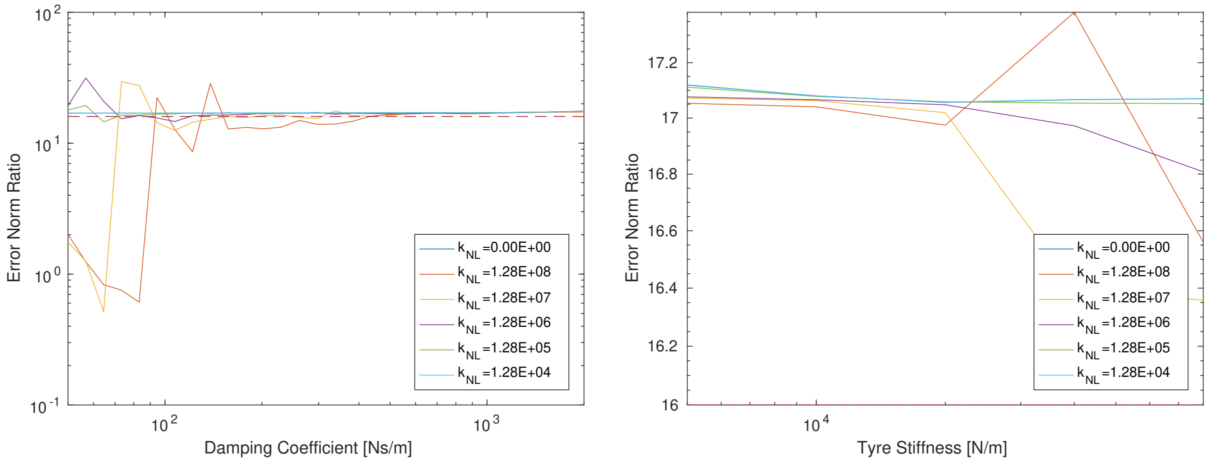
(a) Differing spring displacement outputs.

(b) Spring displacement outputs error.

Figure 2.4: Diverging outputs due to a small initial disturbance in the input profile.

The convergence test results are shown in Figure 2.5. For the damping case, as highlighted in Figure 2.5a, the error ratio eventually converges to 16 for all levels of nonlinearity as the damping coefficient is increased. It is apparent that more damping is required as the nonlinearity of the system increases for the convergence rate to approach 16. Therefore care must be taken when changing the damping parameters to ensure that the system is well-posed. The convergence rate is unaffected by the tyre stiffness values under consideration. The error norm ratio exceeds 16 for all values studied as shown in Figure 2.5b.

The initial condition perturbation results are shown in Figure 2.6. A similar trend occurs as noted in the convergence test. For the damping test, the robustness of the forward problem to an initial condition perturbation decreases as the damping decreases. The robustness also appears to decrease with an increase in nonlinearity. At a certain point of damping all levels of nonlinearity achieve the same error results. For the tyre stiffness test, all values of the nonlinear spring constant achieved the same results. The error due to perturbation increases with the

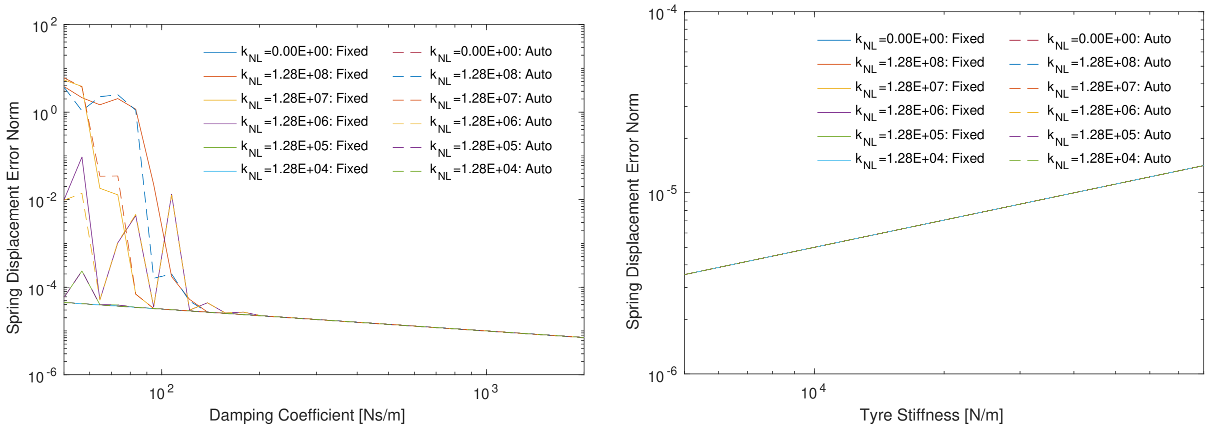


(a) Error norm ratio as a function of nonlinear spring stiffness k_{NL} and damping coefficient b_A

(b) Error norm ratio as a function of nonlinear spring stiffness k_{NL} and tyre stiffness k_R

Figure 2.5: Error norm convergence validation tests

stiffness of the tyre however the errors were reasonable for the parameters under consideration. The auto-solver achieves comparable results to the fixed Runge-Kutta method once the system is using a well-behaved set of parameters.



(a) Error norm as a function of nonlinear spring stiffness k_{NL} and damping coefficient b_A

(b) Error norm as a function of nonlinear spring stiffness k_{NL} and tyre stiffness k_R

Figure 2.6: Error norms as a result of a small initial perturbation.

The validation results give an indication of a suitable range of damping and nonlinear combinations. The suitable parameters are summarised in Table 2.3. For the case where we need to change the tyre stiffness values, all values under consideration should give reasonable results. It is only the damping coefficients that we need to be wary of. The use of the auto-solver with variable step sizes should provide reasonable results provided the determined well-behaved model parameters are used.

2.3 The Inverse Problem

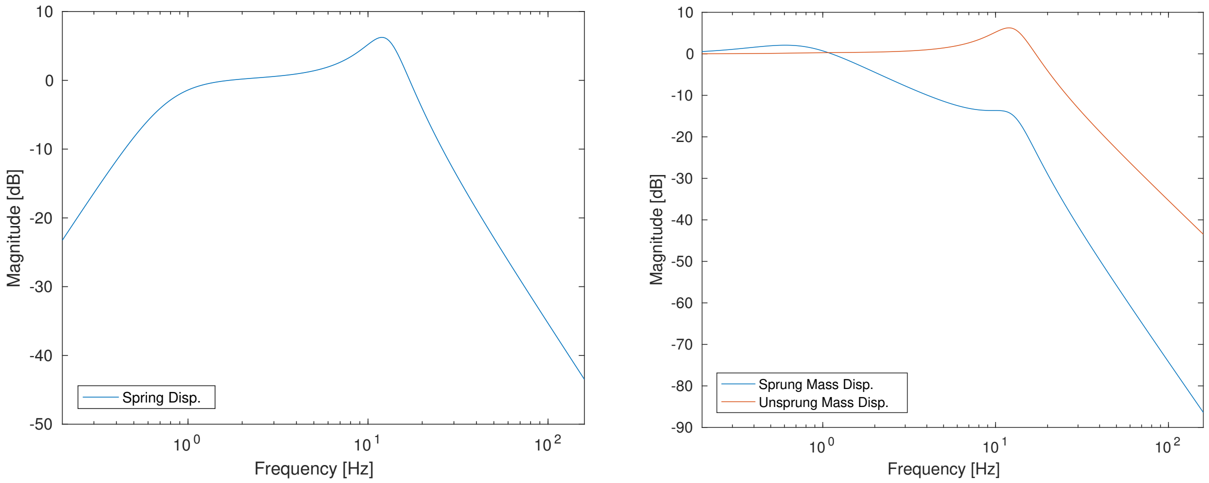
In the forward problem there are features in the input domain that, if varied significantly, will have no or only a minute influence on the output features. These features are referred to as the zeros of the system. In terms of the inverse problem, the opposite problem occurs: small variations in these insensitive output features map to massive variations in the input features. The solution is to minimize or completely remove the influence of these perturbations by means

Table 2.3: Quarter car parameter values deemed to give well-behaved results.

Variable	Details
k_{NL}	1.28×10^n N m^{-3} where $4 \geq n \leq 8$
k_R	from 5×10^3 to 80×10^3 N m^{-1}
b_A	from 125 to 2000 N s m^{-1}

of regularization.

How we express the input and output domains in terms of these features depends on the problem at hand. For linear systems a useful basis is the frequency domain since the output frequencies are only ever linear combinations of the input frequencies. This allows us to easily express where the outputs are most sensitive and insensitive to the inputs through the use of a Bode plot. The Bode plot indicates the gain or attenuation of the input frequency in the output. The Bode plot for the linear quarter car model is plotted in Figure 2.7 in terms of the spring displacement output as well as the absolute displacements of the system. We note that the system acts as a bandpass filter for the spring displacement system, since it attenuates the low and high frequencies. The absolute displacements however act as low-pass filters. For the spring displacement system we could add a number of very high or very low frequencies to the input without affecting the output noticeably.



(a) Relative Spring Displacement

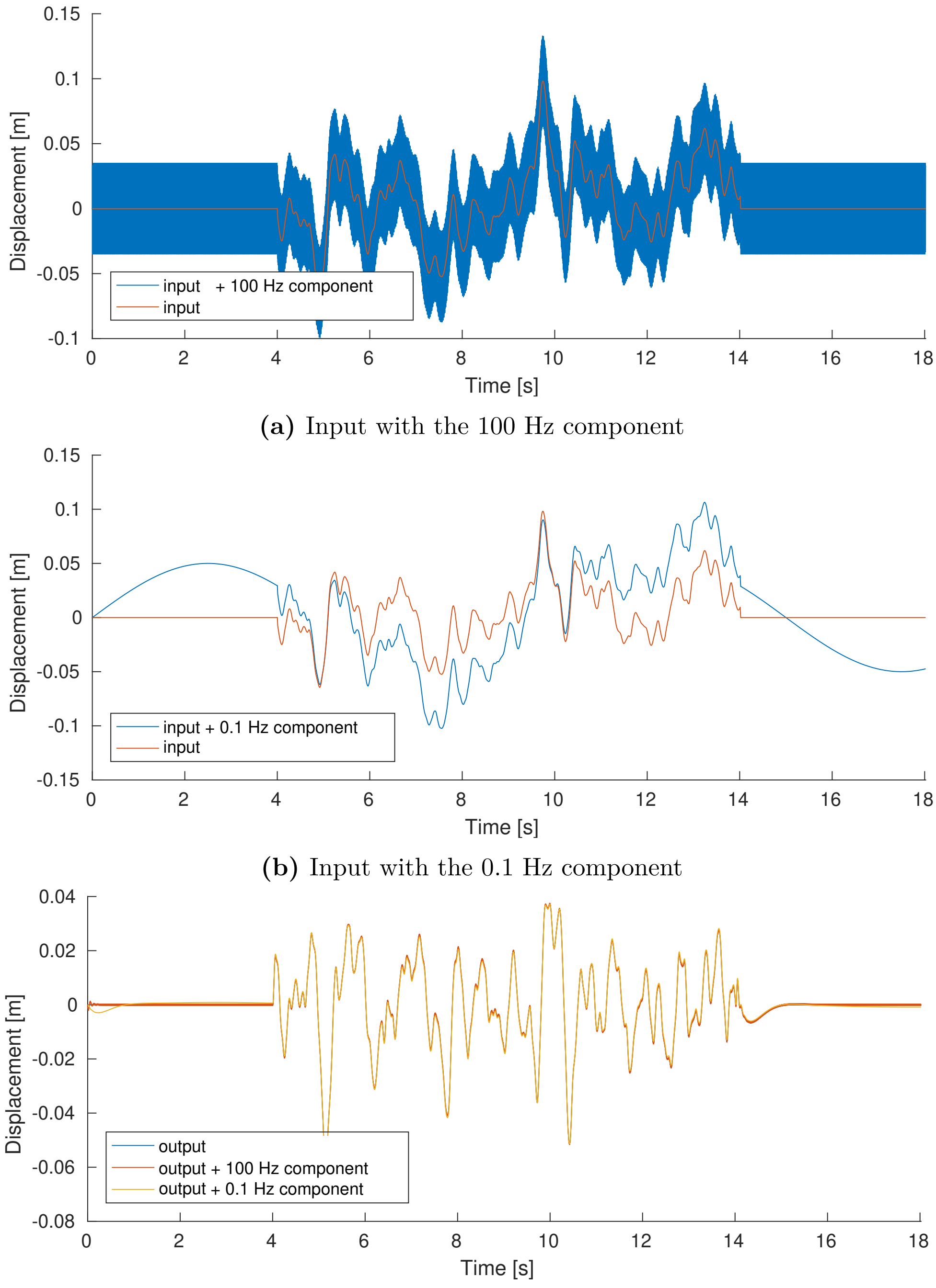
(b) Absolute Displacements

Figure 2.7: Magnitude Bode plot for the linear quarter car set-up comparing relative displacement response to that of absolute displacement.

This is demonstrated in Figure 2.8 where a 0.1 Hz and a 100 Hz signal are added to the road input. The corresponding outputs, plotted in Figure 2.8c, for all 3 inputs are almost identical. In other words many or infinitely many inputs map to the same output. This highlights two important considerations when performing direct inverse methods for response reconstruction:

- The importance of the input accuracy is downplayed since there may be infinitely many valid inputs for a given response. This is referred to as functional reproducibility [31].
- Regularisation is needed in order to obtain sensible input reconstructions since any small perturbation, for instance due to measurement noise, in the insensitive parts of the outputs would lead to large unwieldy reconstructed inputs.

If we instead directly analyse the poles of the linear quarter car set-up,



(a) Input with the 100 Hz component

(b) Input with the 0.1 Hz component

(c) Outputs Comparison

Figure 2.8: Comparison of spring displacement responses when the corresponding inputs have either a low or high frequency spurious component added.

$$\text{poles} = \begin{bmatrix} -20.8172 + 77.8931i \\ -20.8172 - 77.8931i \\ -3.5876 + 3.2512i \\ -3.5876 - 3.2512i \end{bmatrix}, \tag{2.10}$$

we note that the poles of the system are all in the left hand plane. This means that the linear system is stable for the forward model. This is true for all sensor configurations since the poles are only dependent on the system dynamics and are unaffected by the sensor configurations. If we then compare the zeros of the different sensor configurations,

$$\text{zeros}_{\Delta_z} = \begin{bmatrix} 0.2146 \times 10^{-6} \\ -0.2146 \times 10^{-6} \end{bmatrix}, \qquad \text{zeros}_{z_A} = -3.2, \qquad \text{zeros}_{z_R} = \begin{bmatrix} -3.5714 + 3.1784i \\ -3.5714 - 3.1784i \end{bmatrix}, \tag{2.11}$$

we note that the spring displacement zeros Δ_z are not in the left hand plane. Therefore the system that uses the spring displacement as the observable state is not minimum phase. The system that uses either the sprung or unsprung mass displacements, however, will be a minimum phase system. It is therefore expected that in order to invert the relative spring displacement system we will need to perform the inversion non-causally.

Chapter 3

Linear Regression Techniques

In this chapter an overview of the derivation of SBTR is given. But before doing so, related linear regression techniques are covered in order to compare the properties of SBTR in terms of solving linear regression problems.

In linear regression problems the target matrix $Y \in \mathbb{R}^{n \times q}$ of the system is written as a linear combination of the predictor matrix $X \in \mathbb{R}^{n \times p}$, where n is the number of observations and p and q are the respective dimension sizes. The approximation of Y, denoted as $\hat{Y}$, is written in terms of constants $\beta \in \mathbb{R}^{p \times q}$, i.e.

$$\hat{Y} = X\beta. \tag{3.1}$$

Ordinarily we write and solve these problems as a minimisation of the square error of the approximation of Y ($\hat{Y}$) and Y

$$\min_{\beta} ||Y - X\beta||_2, \tag{3.2}$$

which has the known solution [21]

$$\beta = (X'X)^{-1}X'Y. \tag{3.3}$$

The prime superscript denotes the transpose. $(X'X)^{-1}X'$ can be interpreted as a projection of X onto Y. $X'X$ is the covariance matrix for the case when X is mean centred. It is assumed that X and Y are mean centred throughout the rest of this chapter.

For the direct inverse methods that we will be investigating in this dissertation, the windowed outputs Z become the predictor X of the regression problem. Likewise the windowed inputs U become the targets Y. For the type of problems we expect to find in response reconstruction, a large number of the observations of X will be highly collinear. This means that not every observation contributes unique information to the underlying phenomenon and causes X to become singular. Therefore, the inverse of $X'X$ may not exist or is unstable. To overcome this shortcoming, structure is sought in the covariance matrix by finding new latent variables or dimensions which are linear combinations of the original axes. These latent variables seek to explain the maximum amount of variance in either X or Y or a combination of the two with the lowest number of latent variables. The latent variables that do not contribute significantly to the problem are then discarded, thereby reducing the collinearity of the problem.

3.1 Principal Component Analysis (PCA)

A common method of determining the new representations of these dimensions is through the use of Principal Component Analysis (PCA). In PCA we decompose X into R rank 1 matrices

$$X = \sum_{r=1}^{R} M_r \tag{3.4}$$

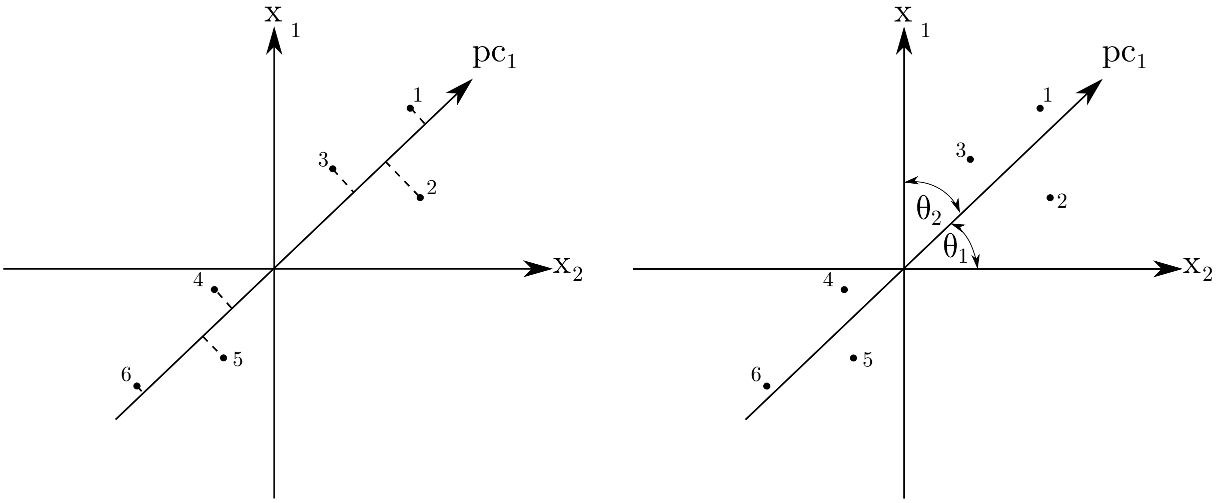
where M_r is the same size as X and R is the original rank of X. This means that each matrix M_r individually spans one of the dimensions originally spanned by X. Each of these matrices M_R can then be broken down into the outer products of two vectors $t \in \mathbb{R}^{n \times 1}$ and $p \in \mathbb{R}^{p \times 1}$ called the scores and loadings respectively such that

$$M_r = t_r p_r', \tag{3.5}$$

or in matrix form as

$$X = TP'. \tag{3.6}$$

The score vectors t can be interpreted as representing the original data points in X projected onto new orthogonal axes called the principal components. These principal components maximize the variance in the original data while minimizing the squared distance of the original data to the new set of axes [21]. The loadings p then represent the cosine of the angles between the original axes and the principal directions [22]. This gives us an indication of weighting of each score on the original axis, in other words how important each axis is in contributing to that score. It also allows for the score to be transformed back into the original axes. A graphical demonstration of PCA is shown in Figure 3.1.



(a) The data points are represented on a new set of axes called the principal components. The data points are projected perpendicularly onto the principal components to obtain the scores T.

(b) The loadings P can be determined by taking the cosine of a unit vector along the principal component. This indicates how much each original axis of the system contributes to the new principal component.

Figure 3.1: A graphical overview of PCA [22].

3.1.1 Singular Value Decomposition (SVD)

One method of performing PCA is by using Singular Value Decomposition (SVD) to factorize X as

$$X = U_x S_x V_x', \tag{3.7}$$

which gives us the left singular vectors $U \in \mathbb{R}^{p \times r}$ and the right singular vectors $V \in \mathbb{R}^{n \times r}$ [21]. The singular vectors are unitary orthogonal matrices which have the property

$$U'U = I, \tag{3.8}$$

with I being the identity matrix. $S \in \mathbb{R}^{r \times r}$ is a diagonal scaling matrix with positive singular values along the diagonal, placed in descending order. If we go back to the original problem of finding structure in the covariance matrix

$$C_{XX} = X'X, \tag{3.9}$$

we can diagonalize it, since we know it is symmetric, to obtain

$$C_{XX} = V\Lambda V' \tag{3.10}$$

where V represents the eigenvectors arranged by columns and the diagonal matrix Λ representing the eigenvalues ordered by size along the diagonal. Substituting Equation (3.7) into Equation (3.9) gives

$$C = V_x S_x' U_x' U_x S_x V_x' = V_x S_x^2 V_x'. \tag{3.11}$$

From this we can conclude that the right singular vectors V obtained during SVD are the eigenvectors or principal directions of the covariance matrix and that the eigenvalues of the covariance matrix λ_i are the squares of the singular values, i.e.

$$\lambda_i = s_i^2. \tag{3.12}$$

SVD can now be related back to PCA with the loading matrix given as

$$P = V \tag{3.13}$$

and the corresponding score matrix given as

$$T = U_x S_x = X V_x. \tag{3.14}$$

The SVD can also be used to create a reduced rank approximation of the original matrix X by selecting the L largest singular values $S \in \mathbb{R}^{L \times L}$ and the corresponding columns of the left $U \in \mathbb{R}^{p \times L}$ and right singular vectors $V \in \mathbb{R}^{n \times L}$. The smaller singular values and their associated vectors are then ignored. This has the benefit of reducing memory requirements to store the original matrix X since only $(p \times L) + (L \times 1) + (n \times L)$ entries need to be stored as opposed to $(p \times n)$. This comes at a cost of loss of information but this is mitigated since it is normally assumed that the smaller singular values are associated with random noise. The SVD has complexity of $O(\min(pn^2, p^2 n))$ and is useful when all of the scores and loading are required [30].

3.2 Principal Component Regression (PCR)

In PCR the input matrix X in Equation (3.1) is replaced with the scores matrix T such that

$$\hat{Y} = T\beta, \tag{3.15}$$

with the scores associated with low variance being discarded [21]. This then has the solution

$$\beta = (T'T)^{-1}T'Y. \tag{3.16}$$

This ensures that $T'T$ is easily invertible. Solving the collinearity problem as well as reducing random noise can be achieved by discarding scores of principal components associated with small eigenvalues. To see why PCR allows for easier inversion, we can substitute $T = XV$ in Equation (3.15) and $T = US$ in Equation (3.16) to get

$$\hat{Y} = XV_x(T'T)^{-1}T'Y \tag{3.17}$$
$$= XV_x(S_x'U_x'U_xS_x)^{-1}S_x'U_x'Y. \tag{3.18}$$

Using the property of $U'U = I$ and the diagonal property of S to get $(S_x'S_x)^{-1}S_x' = S_x^{-1}$ results in

$$\hat{Y} = X(V_xS_x^{-1}U_x')Y. \tag{3.19}$$

The diagonal S_x^{-1} term is simple to calculate as the diagonals of S_x^{-1} are the reciprocals of each singular value of S, i.e.

$$S_{x,ii}^{-1} = \frac{1}{s_i}. \tag{3.20}$$

This explains why truncating small singular values helps to improve stability by increasing the condition number of the problem. The singular values, that are close to zero, are blown up when inverted. These extreme inverted singular values would otherwise dominate the larger singular values which are of importance to the underlying model.

3.3 Tikhonov Regularization

Another form of regularization known as Tikhonov regularization can be shown to treat the truncation of small singular values in a more smooth manner. Tikhonov regularization uses a regularization matrix Γ with the solution

$$\beta = (X'X + \Gamma'\Gamma)^{-1}X'Y, \tag{3.21}$$

where Γ is typically chosen as a scaling of the identity matrix, $\Gamma = \alpha I$, commonly referred to as Ridge Regression (RR). RR has the solution in terms of the SVD of X [21]

$$\hat{Y} = X(V_xDU_x')Y, \tag{3.22}$$

where the entries of the diagonal matrix D are given by

$$D_{ii} = \frac{s_i}{s_i^2 + \alpha^2}. \tag{3.23}$$

Where in PCR there is a sharp cut off of the singular values, RR instead smoothly decays the inverted singular values through the choice of the parameter α.

3.4 Spanning Basis Transformation Regression (SBTR)

In PCR we demonstrated that the regression problem is made easier by choosing scores that maximize the covariance of X. However, there is no direct coupling to the output space Y when extracting the scores. This means that there is no way to distinguish if the truncated information is random noise or information that varies poorly in X (hence having low singular values) but co-varies strongly between Y and X and is therefore useful in predicting Y. One method to solve this problem is by building an inner relationship between the scores of X and Y [22]. An outer relationship is built for X in terms of scores and loadings as was done with PCR, i.e.

$$\hat{X} = TP'. \tag{3.24}$$

To proceed Y is also decomposed terms of its own scores R and loadings Q, i.e.

$$\hat{Y} = RQ'. \tag{3.25}$$

The outer relationships are then related to one another by mapping their corresponding scores using a linear relationship

$$R = T\beta, \tag{3.26}$$

such that

$$\hat{Y} = T\beta Q'. \tag{3.27}$$

One way of achieving this is to decompose X and Y independently of one another using PCA and then map the extracted scores using Equation (3.26). This is referred to as the *over-simplified model* [22] and as SBTR [11]. SBTR is presented as a generalized framework that maps the arbitrarily chosen spanning bases selected from an input to the spanning bases of the output through the use of scalings and rotations. However, if SVD is used to determine the set of spanning bases, SBTR reverts to the *over-simplified model* [22].

We start SBTR by decomposing X as we did in Equation (3.7) and obtain the scores and loadings using Equations (3.13) and (3.14). Likewise, we decompose Y to get

$$Y = U_y S_y V_y', \tag{3.28}$$

and then determine the corresponding loadings R and Q

$$\begin{aligned} R &= U_y S_y, &(3.29)\\ Q &= V_y. &(3.30)\end{aligned}$$

Solving for β in Equation (3.26) and then substituting Equation (3.13) and using the same orthogonal properties used to simplify Equation (3.18), gives

$$\begin{aligned} \beta &= (T'T)^{-1}T'R, &(3.31)\\ &= (S_x'U_x'U_xS_x)^{-1}S_x'U_x'R, &(3.32)\\ &= S_x^{-1}U_x'R. &(3.33)\end{aligned}$$

We then substitute the solved β into Equation (3.27) to get the prediction equation for Y,

$$\hat{Y} = XPS_x^{-1}U_x'RQ'. \tag{3.34}$$

We can expand out the equation in terms of the original SVD matrices to give

$$\hat{Y} = XV_xS_x^{-1}U_x'U_yS_yV_y'. \tag{3.35}$$

In summary the following remarks can be made about SBTR:

- SBTR can be interpreted as creating weighted sums between the scores extracted from X and Y which are extracted independently of one another.
- Like PCR we have the choice of choosing the number of scores. In the case of SBTR we have the choice of selecting the number of retained scores, L, for X and Y independently of one another. In the original implementation of SBTR [11], the number of retained scores for X and Y were set equal, i.e. $L_x = L_y$. This is the method that will be followed throughout this dissertation.
- SBTR can be interpreted as PCR with an additional step of having a reduced rank approximation of Y. If all the scores for the outputs Y are retained then SBTR reverts to PCR.

3.5 Partial Least Squares by Singular Value Decomposition (PLS-SVD)

In SBTR the scores are still chosen such that they maximize the variance in their own respective spaces without considering how the chosen scores affect the mapped variance in each other. If instead the scores were chosen such that they maximized the variance in each other, this would result in potentially fewer latent variables being required to fully explain all the variance in Y. This would result in a parsimonious and hopefully more true representation of the underlying dimensionality of the problem. This is equivalent to finding the scores that maximize the covariance of $X'Y$. This is the underlying idea behind Partial Least Squares (PLS). PLS can be performed using either iterative or direct methods. The more popular methods of performing PLS iteratively are the NIPALS [22] or the SIMPLS algorithms [14]. The NIPALS algorithm has complexity $O(ipql)$[30] where i is the number of iterations to convergence, which is relatively quick, and l is the number of retained scores. Therefore the iterative methods are suitable if the number of scores required are low. It is also possible to perform PLS using SVD if all the loadings and scores are required. It is referred to as Robust Canonical Analysis [36] where it was used for regression and as PLS-SVD [7] where it was used for analysis. We will be referring to the method as PLS-SVD. However it must be noted that the different methods of performing PLS are not equivalent and that the scores and loading extracted by the direct methods and the iterative methods differ from one another [38].

To perform PLS-SVD we decompose the covariance matrix $X'Y$ using SVD to obtain

$$X'Y = U_{xy}S_{xy}V_{xy}', \tag{3.36}$$

where $U_{xy} \in \mathbb{R}^{p\times r}$, $S_{xy} \in \mathbb{R}^{r\times r}$ and $V_{xy} \in \mathbb{R}^{q\times r}$. The inputs X are then projected onto the left singular vectors to obtain the scores for X such that

$$T = XU_{xy}, \tag{3.37}$$

while Y is projected on to the right singular vectors to obtain the scores for Y to give

$$R = YV_{xy}. \tag{3.38}$$

The corresponding loadings for X and Y are the left and right singular vectors respectively, i.e.

$$\begin{aligned} P &= U_{xy}, && (3.39) \\ Q &= V_{xy}. && (3.40) \end{aligned}$$

Solving for β in Equation (3.26) and then substituting Equations (3.37) and (3.38), we obtain

$$\begin{aligned} \beta &= (T'T)^{-1}T'R, && (3.41) \\ &= (U'_{xy}X'XU_{xy})^{-1}U'_{xy}X'YV_{xy}. && (3.42) \end{aligned}$$

We then substitute the solved β into Equation (3.27) to get the prediction equation for Y

$$\begin{aligned} \hat{Y} &= XU_{xy}(U'_{xy}X'XU_{xy})^{-1}U'_{xy}X'YV_{xy}V'_{xy}, && (3.43) \\ &= XU_{xy}(U'_{xy}X'XU_{xy})^{-1}U'_{xy}X'Y. && (3.44) \end{aligned}$$

If we write out the covariance matrix between the scores T and R and expand it in terms of the original X and Y using Equations (3.37) and (3.38), we obtain

$$T'R = U'_{xy}X'YV_{xy}, \tag{3.45}$$

where $X'Y$ is the original covariance matrix. Expanding the original covariance matrix using Equation (3.36) and using the orthogonal properties of the singular vectors result in

$$\begin{aligned} T'R &= U'_{xy}U_{xy}S_{xy}V'_{xy}V_{xy}, && (3.46) \\ &= S_{xy}. && (3.47) \end{aligned}$$

This means that the covariance matrix of T and R is simply the diagonal matrix containing the singular values of $X'Y$. This results in the solution for the linear relationship between the scores T and R, in terms of β such that

$$\beta_{ii} = \frac{s_i}{t'_i t_i}, \tag{3.48}$$

where t_i is the i^{th} column of T. The off-diagonals of the β coefficient matrix are equal to 0, i.e.

$$\beta_{ij} = 0, \tag{3.49}$$

where $i \neq j$ [36]. However for the implementation in this dissertation the method outlined in Equation (3.44) will be used. The collinearity problem can be mitigated by truncating the smaller singular values and their corresponding singular vectors.

3.6 Scaling

The outputs Y and inputs X of the system record different types of signals which will have different variances across them. We may also find that the sensors contain constant biases that need to be accounted for. Therefore it would be pertinent to scale the inputs and outputs such that the rows have a mean of 0 and a variance of 1 [21]. This is achieved using the z-score stated as

$$\bar{X} = \frac{X - \mu_p}{\sigma_p}, \tag{3.50}$$

where μ_p is the row mean

$$\mu_p = \frac{1}{N}\sum_{n=1}^{N} x_{p,n}, \tag{3.51}$$

and σ_p is the row standard deviation

$$\sigma_p = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_{p,n} - \mu_p)^2}. \tag{3.52}$$

Since the regression methods covered extract the latent variables that explain the greatest variance in the input and output spaces it makes sense to scale them accordingly such that one type of signal is not favoured unnecessarily over another. Scaling the matrices also has the benefit of improving the conditioning of the matrices.

3.7 Conclusion

The regression methods covered seek new orthogonal axes or latent variables that maximize the explained variance in either X (PCR and RR) or in $X'Y$ (PLS-SVD) or in X and Y independently of one another (SBTR). The issue of collinearity can be mitigated by truncating the number of retained scores. The choice of retained scores will be determined through the use of cross validation and will be covered in the following chapter.

The latent variables are calculated using iterative methods such as NIPALS or directly using SVD. Iterative methods scale $O(ipnl)$ whereas direct methods scale $O(\min(pn^2, p^2n))$. Iterative methods are used when the number of latent variables required are significantly less than the original dimensionality of the problem. For easy comparison against the SBTR algorithm, direct methods that implement SVD will be considered and benchmarked in this dissertation. Therefore SBTR will be benchmarked against PCR, PLS-SVD and RR.

Chapter 4

Model Building and Validation

In the previous chapter we demonstrated that the number of retained latent variables L extracted by the different regression methods need to be truncated in order to solve the collinearity problem. It was stated in the objectives of this dissertation that the use of overlapping windows will be investigated. The use of overlap introduces another variable, proportion overlap γ, that needs to be optimized in addition to the window length T_w. This chapter covers the process of optimizing for these hyper-parameters.

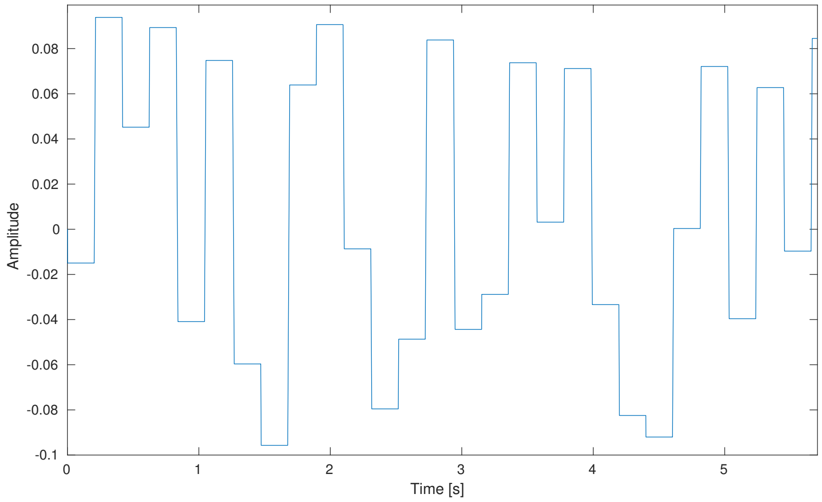
4.1 Choice of Excitation Signals

Before we can begin building a direct inverse model of our plant we need good quality data since the quality of the excitation signal places an upper bound on the accuracy of any subsequent model that we wish to build [32]. In the case of response reconstruction we have the benefit that we can design the signals we wish to train on. There are two possible methods of designing excitation signals: model-free and model-based methods. In model-based methods subsequent excitation signals are chosen such that they improve the accuracy of the model [15]. Initially we have little prior knowledge of the system and of the real world input signals, therefore we need to employ model-free methods. In model-free methods we design an excitation signal that offers the best distributed coverage of the operating condition. We assume we have some prior knowledge of the range of the operating condition. A popular choice in system identification literature for this is the Amplitude Modulated Pseudo Random Binary Signal (APRBS).

4.1.1 Amplitude Modulated Pseudo Random Binary Signal

Since we are working with nonlinear structural systems we know that the system responses are functions of input frequencies as well as the amplitude at which we excite the system. Therefore a signal that covers the necessary frequencies as well as covering the expected amplitude range of operations is required. The APRBS attempts to cover the amplitude operating conditions with a series of step responses that are fairly well distributed over the input range. An example of an APRBS is shown in Figure 4.1.

In order to specify the profile, a set of N design points d_n are chosen to define the amplitudes of the steps. The design points are sampled from the desired range $[u_{\min}, u_{\max}]$ using Latin Hypercube Sampling (LHS). LHS splits the design space into N intervals with one design point placed randomly in each interval. LHS then iteratively optimises the design points such that each design point is the maximum distance away from its neighbours. This provides a random but equally spread set of design points. Since no physical system can achieve an instantaneous change in displacements required for a true step input, the step is instead approximated by a ramp function. The slope of the ramp is determined by the maximum allowed velocity $v_{\max}$ that can safely or accurately be performed by the actuator. The slope of the ramp affects



,

Figure 4.1: APRBS example.

the frequency content of the signals with higher velocities resulting in higher frequencies being excited [16]. The length of the step is then specified by the hold time T_h. The testing time is limited, therefore the maximum number of steps that best cover the input space in the shortest time is sought. The hold time T_h must be small enough to fit as many steps in but must be long enough that the steps actively excite the system at that point. The hold time T_h is typically set to be at least the length of the largest time constant $T_{\mathrm{c,max}}$ of the system [32]. This can be determined with a simple step test of the system if no prior knowledge is known.

4.1.2 Road Profile

In order to generate a separate test set to determine how well the direct inverse model performs on unseen data, the ISO 8608 standard [19] for specifying road profiles will be employed. The ISO 8608 standard defines inputs that are distinct from APRBS while still being representative of real world operating conditions. The profiles are characterised by the standard in the frequency domain where the spectral density S_z is given by

$$S_z(\phi) = A(\phi)^{-n}, \tag{4.1}$$

for the given spatial frequency ϕ with units m^{-1}. The A term represents the roughness coefficient of the road whereas n represents the road index of the profile. The A coefficient controls how large the amplitudes are at each frequency whereas n controls how quickly the amplitudes decay as functions of frequency. By altering these two coefficients the varying types of profiles such as ploughed agricultural land to smooth gravel highways can be produced. The spatial frequencies ϕ are limited between 0.01 and 10 m^{-1}. The former represents the broad changes in the landscape which have negligible effects on vehicle dynamics while the upper limit on the frequency represents small scale variations which are filtered out by the tyre [25]. When generating the profiles only the amplitude information is given by the ISO 8608 standard, therefore in order to generate time signals, a uniformly random signal is generated for the phase signal. This generates a displacement signal as a function of distance. The velocity of the vehicle must therefore be chosen in order to generate a displacement signal as a function of time.

4.1.3 Sampling Frequency

When recording our training input and output signals we have the choice in selecting the sampling frequency. It is recommended for the case of correlation and spectral methods to choose a sampling frequency $f_s = 10f_c$ where f_c is the upper limit of the frequencies we are interested in [5]. For parameter based models $f_s = 2.5f_c$ is recommended. For ADT in vehicles we are rarely interested in frequencies above 100 Hz therefore a default sampling frequency of $f_s = 1000$ Hz was chosen for training and testing the linear regression techniques. However it is demonstrated in Section 6.1 that this value can be reduced significantly.

4.1.4 Preprocessing

Certain windowing techniques will truncate some of the samples in the testing and training sets. To ensure that there is a fair comparison between the different data sets a dead time will be appended and prepended. The dead times will be excluded when calculating the cost function during cross validation as well as when reporting the final accuracies of the predictions. The constant initial and final conditions also allow for different signals to be appended to one another without introducing unwanted jumps.

4.2 Cross Validation

We know that we cannot use all of the scores extracted due to measurement noise in our observations, response reconstruction being an ill-posed problem as well as the issue of collinearity. Therefore we select fewer scores or latent variables than the problem allows while still maximizing the amount of variance explained. In order to determine the number of retained scores we can vary the number of retained scores to see how well they predict on an unseen validation set in a process called cross validation. However, cross validation can be misleading if it is implemented without considering dependencies between observations. If the validation set is removed once the data has already been windowed with overlaps, then the validated set will be correlated to the training set due to the overlaps introduced. If the validation set is first removed from the middle portion of the dataset and then windowed, then care must be taken when splitting and merging the training set to ensure that no unintended overlap is introduced between the separated training segments. A simpler solution to this problem is implemented by removing a single validation set from either the beginning or end of the dataset before windowing. In this work a validation set was created independently of the training set. In the original implementation of SBTR k-fold cross validation was implemented on the data set after they were split into windows [12]. This was allowable since overlap was not implemented. It was also computationally tractable to implement since the number of observations obtained were low allowing for fast calculations during SVD. The hold out set implemented in this work has the benefit of reducing the computational overhead introduced by k-fold cross validation but comes at the cost of producing a more biased estimate of the validation error [6].

4.3 Choice of Cost Function

We have the choice of either using the errors of the approximated inputs or the approximated outputs as the cost function of the optimisation scheme. In response reconstruction, we are interested in producing an accurate output response of the system since a unique input may not exist. The downside of this is that in order to obtain the output error, the approximated input needs to be passed through the test rig. This needs to occur for every loop in the cross validation step. This is not an issue for the numerical model since it is cheap to compute, however, in the real world this would result in significant fatigue of the experimental rig and would take considerable time to run. Therefore it is necessary to limit the number of forward evaluations in the cross validation step. In the evaluation of the methods investigated for response reconstruction, the output error will be used in the cross validation step. Since we need to measure and compare response and input reconstruction accuracies across different types of signals, we need a normalized measure of error. The Mean Fit Function Error (MFFE) [9] will be used to report the final test accuracies of the reconstructed input and output signals. MFFE is defined as

$$\text{MFFE} = 100 \times \frac{\sum_{m=1}^{M} |e_0|}{\sum_{m=1}^{M} |z_0|} [\%], \tag{4.2}$$

where e_0 is the error between the true output z_0 and the approximate output $\hat{z}_0$, i.e.

$$e_0 = z_0 - \hat{z}_0. \tag{4.3}$$

The signals under consideration have been mean centred such that

$$z_0 = z - \mu_z, \tag{4.4}$$

$$\hat{z}_0 = \hat{z} - \mu_{\hat{z}}. \tag{4.5}$$

4.4 Training Procedure

The cross validation algorithm consists of two sub-routines: an outer routine which incorporates the windowing parameter grid search and an inner subroutine which optimises for the number of latent variables.

4.4.1 Window Loop

A graphical overview of the training process is shown in Figure 4.2 with focus on the window parameter search. The window optimisation loops over the window length, $T_{w,i}$, and the amount of overlap, γ_j. The training set U_{train} and Z_{train} as well as validation output Z_{val} are then windowed accordingly. The z-score parameters, σ_{ij} and μ_{ij}, are then calculated using only the training dataset and applied to both the training and validation set. The training set is then decomposed using SVD according to the regression method specified. The full SVD is calculated so that we can simply truncate it to the required number of latent variables as opposed to recalculating it every time we want a different number of latent variables. The decomposed SVD and the range of latent variables $L_{\text{max},ij}$ are then passed to the latent variable optimisation loop.

4.4.2 Latent Variable Loop

A graphical overview of the training latent variable optimisation loop is shown in Figure 4.3. The regression coefficients β_k are then calculated using the reduced number of latent variables, L_k. The approximated windowed validation inputs $\hat{Y}_{\text{val}}$ are then predicted using the windowed validation outputs $\hat{X}_{\text{val}}$. The approximated windowed validation inputs are rescaled and then merged using the specified windowing methods to obtain the approximated input $\hat{U}_{\text{val}}$. The merged inputs are then passed through the test rig to obtain the approximated output $\hat{Z}_{\text{val}}$. The MFFE is then calculated between the true output Z_{val} and the approximated output $\hat{Z}_{\text{val}}$. The optimal number of latent variables L_{min} and the corresponding minimum MFFE are then returned from this loop to the windowing loop as seen Figure 4.2. This minimum MFFE result is then used in the window loop to find the corresponding optimal window parameters $T_{w,\text{min}}$ and γ_{min}.

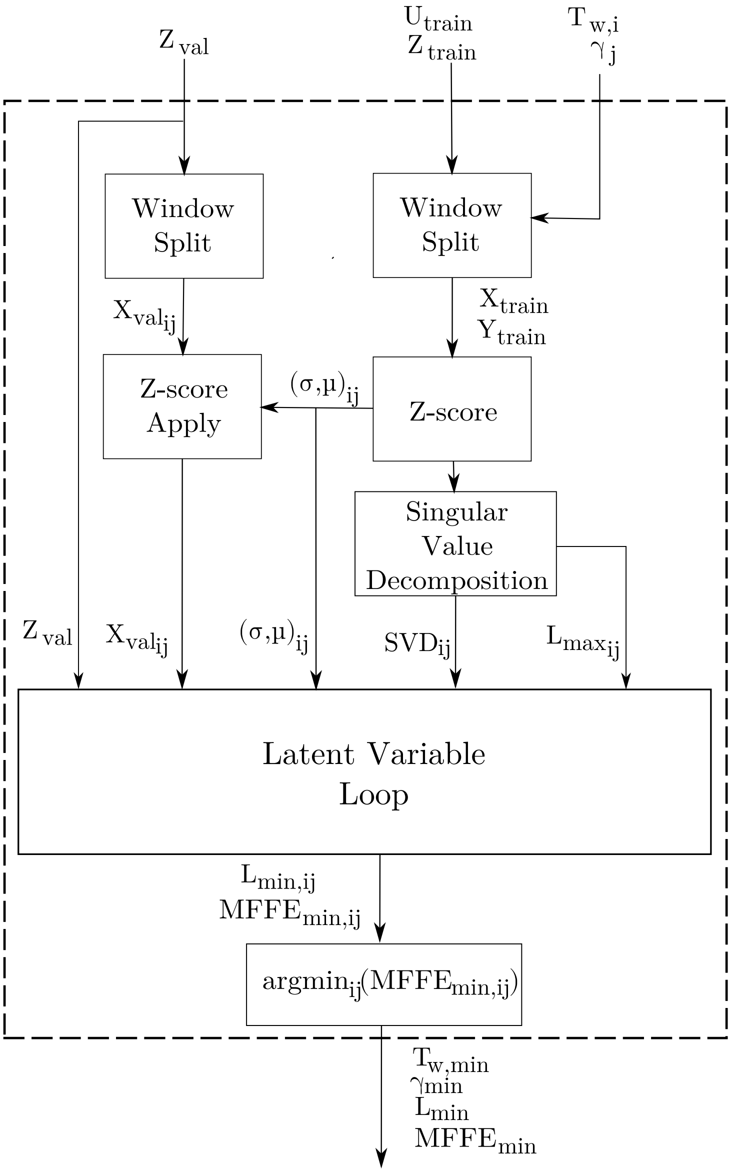


Figure 4.2: Overview of the cross validation procedure used to determine the optimal hyper-parameters.

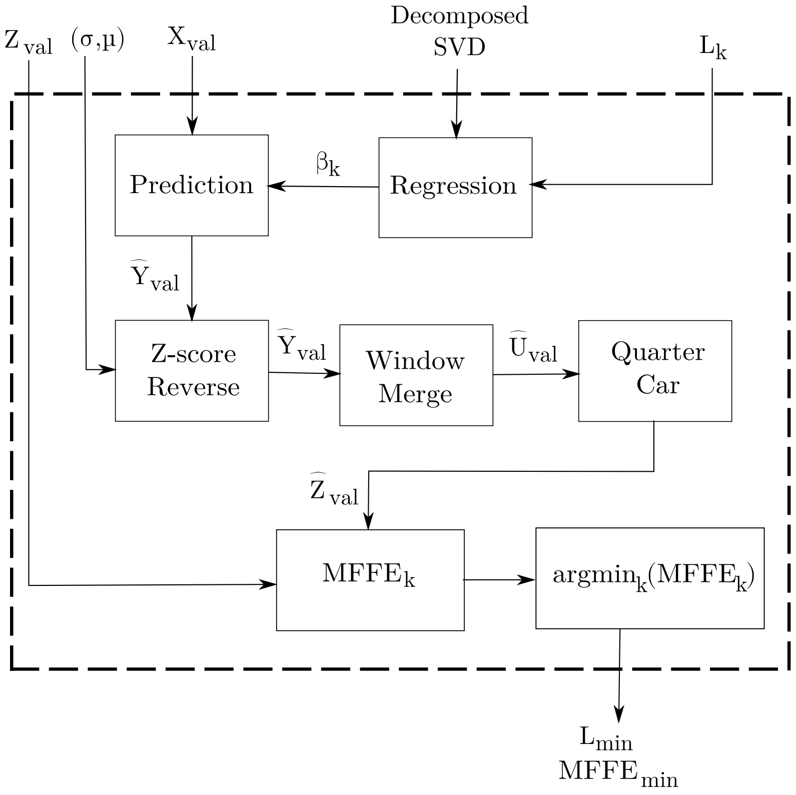


Figure 4.3: Overview of the latent variable optimisation loop.

4.4.3 Final Training Step

In the final training step, the training set is concatenated with the validation set. This newly combined set is then windowed with the optimised window paramaters $T_{w,\text{min}}$ and γ_{min}. The new z-score parameters $[\sigma, \mu]$ are then calculated. The combined set is decomposed and used in the regression step with the optimised number of latent variables L_{min} to determine the final regression coefficients β_{final}.

4.5 Prediction

A graphical overview of the prediction step and the approximation of the output is shown in Figure 4.4. Once the training step is complete it is relatively straightforward to use the optimized parameters to make further predictions. The test output signal Z_{test} needs to be preprocessed first before predictions can be made. The test signal is windowed and z-scored normalised, using the parameters determined during the training phase, to obtain the predictor matrix, X_{test}. The prediction step then occurs using the regression coefficients β_{final} obtained during training to obtain the approximate target matrix, $\hat{Y}_{\text{test}}$. The windowing and z-score normalisation are then reversed before passing the approximated input $\hat{U}_{\text{test}}$ into the test rig to obtain the approximated output, $\hat{Z}_{\text{test}}$.

4.6 Experimental Procedure Summary

The experimental procedures which follow will use the framework outlined in this section so it is worthwhile giving a high-level overview of the general procedure. Each experiment will tweak the methodology slightly depending on the variables of interest but in general, they will follow the general framework:

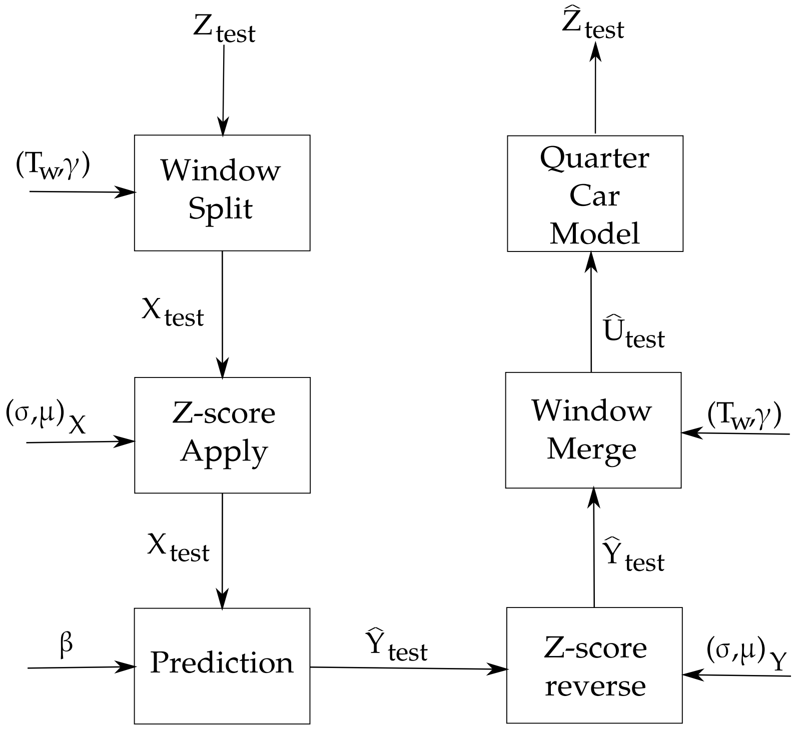


Figure 4.4: Overview of the prediction procedure.

1. Generate a set of training and validation input signals using APRBS as outlined in Section 4.1.1.
2. Generate a set of test input signals using a road profile as outlined in Section 4.1.2.
3. Pass these input signals through the physical laboratory set-up. In this case, we will be using a numerical model to simulate this step as outlined in Section 2.
4. Window these input and output pairs using a windowing method, the window parameters include the window length T_w and the amount of overlap γ. Scale these windowed outputs and inputs according to Section 3.6.
5. Use the windowed pair of training input and output signals to create a direct inverse linear regression model with a regularisation term, for most of the regression methods covered this is the amount retained latent variables L. For the case of ridge regression, it is the regularisation constant α.
6. Use this regularised inverse model with the validation output to get an approximate windowed validation input. Unscale these inputs.
7. Unwindow the approximate validation input using the appropriate method.
8. Pass the unwidowed approximate validation input through the physical laboratory set-up. This gives us an approximate validation output signal.
9. Compute response reconstruction error between approximate and true validation outputs using Equation (4.2).
10. Repeat steps 4-9 using different regularisation constants in a grid search such that the validation output reconstruction is minimized. (It may be necessary to perform an outer grid search in step 4 on the window parameters when comparing windowing methods).
11. Window the test output.
12. Use the model that minimizes the response reconstruction error to create an approximate windowed input given the test output. Reverse the scaling.
13. Unwindow the approximate test input.

14. Pass this approximate input through the physical experimental set-up to get an approximated output.
15. Compare the test approximated output and the true output using Equation (4.2).

Chapter 5

Windowing Methods

Since we are working with physical systems we can take advantage of the fact that the system will experience damping and that part of the energy of the system will be lost to the environment. Therefore we can assume that the states of the system will not be influenced by inputs or states that occurred far back in time. This allows for long measurement sequences to be subdivided into windows where the final state of the system in that window is only influenced by the events that occur in that window. Therefore we can treat each window as an independent experiment allowing for more observations with more manageable sequence lengths. The length of the window is closely related to the amount of damping in the system. The relationship between window length and damping will be further explored in Section 6.2.

5.1 Non-overlapping Windows Investigation

In this section the non-overlapping windowing method will be tested using SBTR on a slightly broader scope of sensor configurations. In this study two types of sensor set-ups will be used. The first set-up uses the absolute displacements of the sprung and unsprung mass. This represents the sensor set-up used in the original implementation of SBTR [12]. The second set-up uses the more practical set-up of using acceleration and displacement readings.

For this investigation the default parameters for the quarter car will be used as stated in Table 2.1. However, the nonlinearity of the model will be switched off. No measurement noise will be present as well. This represents the ideal case and is the bare minimum required of the regression technique. APRBS will be used for training and validation sets. The lengths of the signals are 26.3 s and 15.9 s respectively. The parameters of the signals, as discussed in Section 4.1.1, are given in Table 5.1.

Table 5.1: APRBS parameters used to generate the training and validation signals used in the non-overlapping windows numerical investigation.

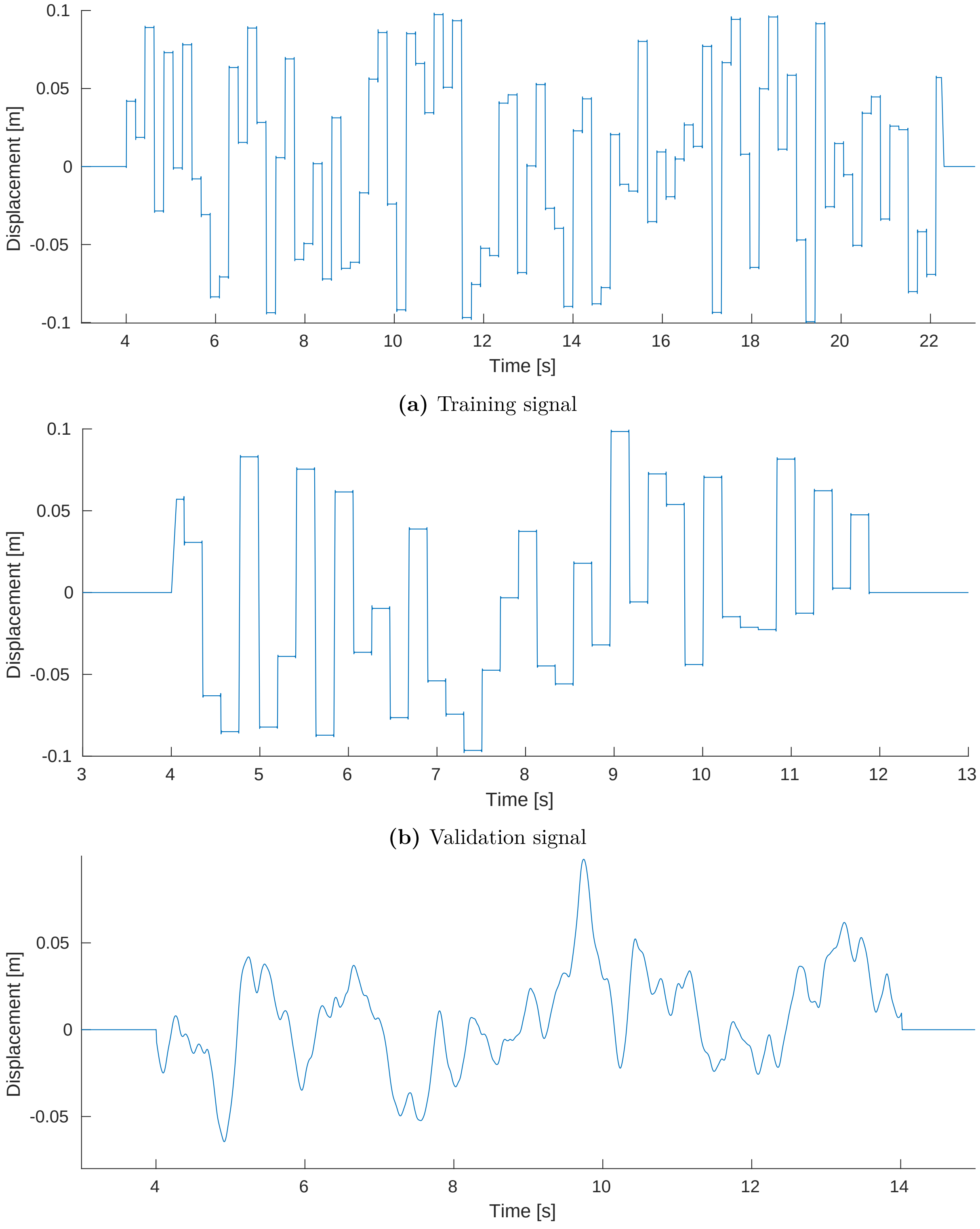
T_s $[s]$	T_h $[s]$	$v_{\mathbf{max}}$ $[ms^{-1}]$	$u_{\mathbf{max}}$ $[m]$	$u_{\mathbf{max}}$ $[m]$
0.001	0.2	10	-0.1	0.1

A 10 s long road profile specified by the ISO 8608 standard will be used for testing the model. The parameters of the road profile, as discussed in Section 4.1.2, are given in Table 5.2. The generated signals are shown in Figure 5.1 with a zoomed-in slice of the training signal shown in Figure 5.2.

The entire investigation will be completed using a 1000 Hz sampling rate. The window sizes and number of latent variables will determined by grid search with $T_w \in [0.1, 4]$ s with 25 equally

Table 5.2: Road profile parameters used to generate the test signal used in the non-overlapping windows numerical investigation.

n	A	ϕ_{min} $[m^{-1}]$	ϕ_{max} $[m^{-1}]$	ϕ_{int} $[m^{-1}]$	v $[ms^{-1}]$
10	6.5×10^{-4}	0.5	10	3.5×10^{-4}	5



(a) Training signal

(b) Validation signal

(c) Test signal

Figure 5.1: Training, validation and test input signals used in the non-overlapping windows numerical investigation

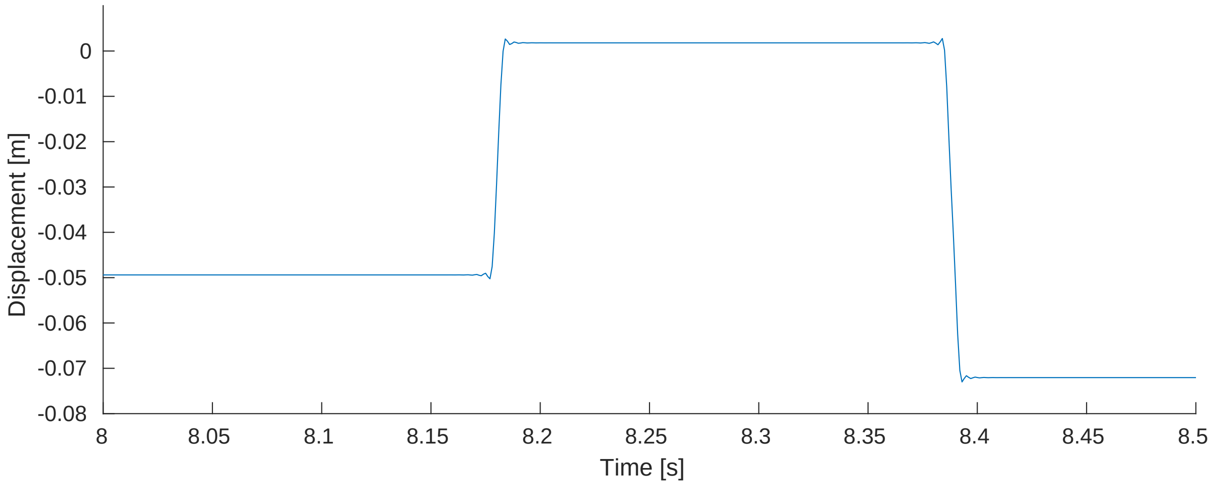


Figure 5.2: Training signal zoomed in.

spaced intervals and $L \in [1, L_{\max}]$ with d_L spaced divisions given as

$$d_L = \min(40, \max(10, \lceil L_{\max}/10 \rceil)). \tag{5.1}$$

A plot of the function used to determine the number of latent variable divisions is shown in Figure 5.3.

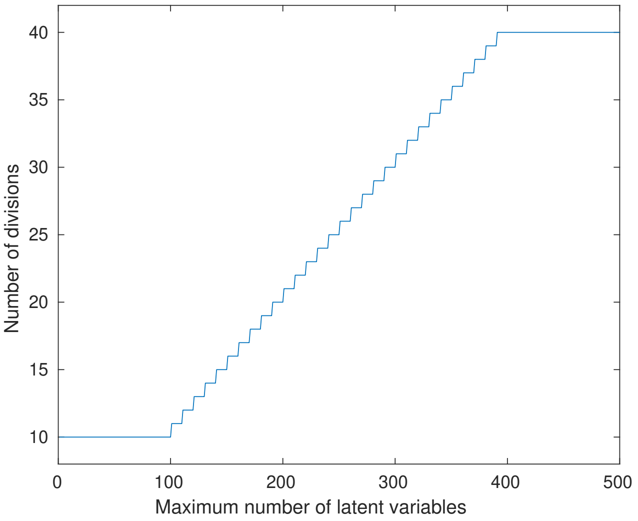


Figure 5.3: The number of grid search divisions used to determine the number of latent variables as a function of maximum possible number of latent variables.

For the relative displacement and acceleration set-up the spring displacement MFFE value will be the cost function. In the absolute displacement case the sprung mass displacement MFFE value will be the cost function.
A summary of the numerical experiment parameters is given in Table 5.3.

5.1.1 Results

The MFFE values for the reconstructed outputs and inputs are shown in Table 5.4.

The absolute displacement set-up performs remarkably well whereas the acceleration and spring displacement set-up performs poorly across the outputs and the inputs. There are 3 orders of

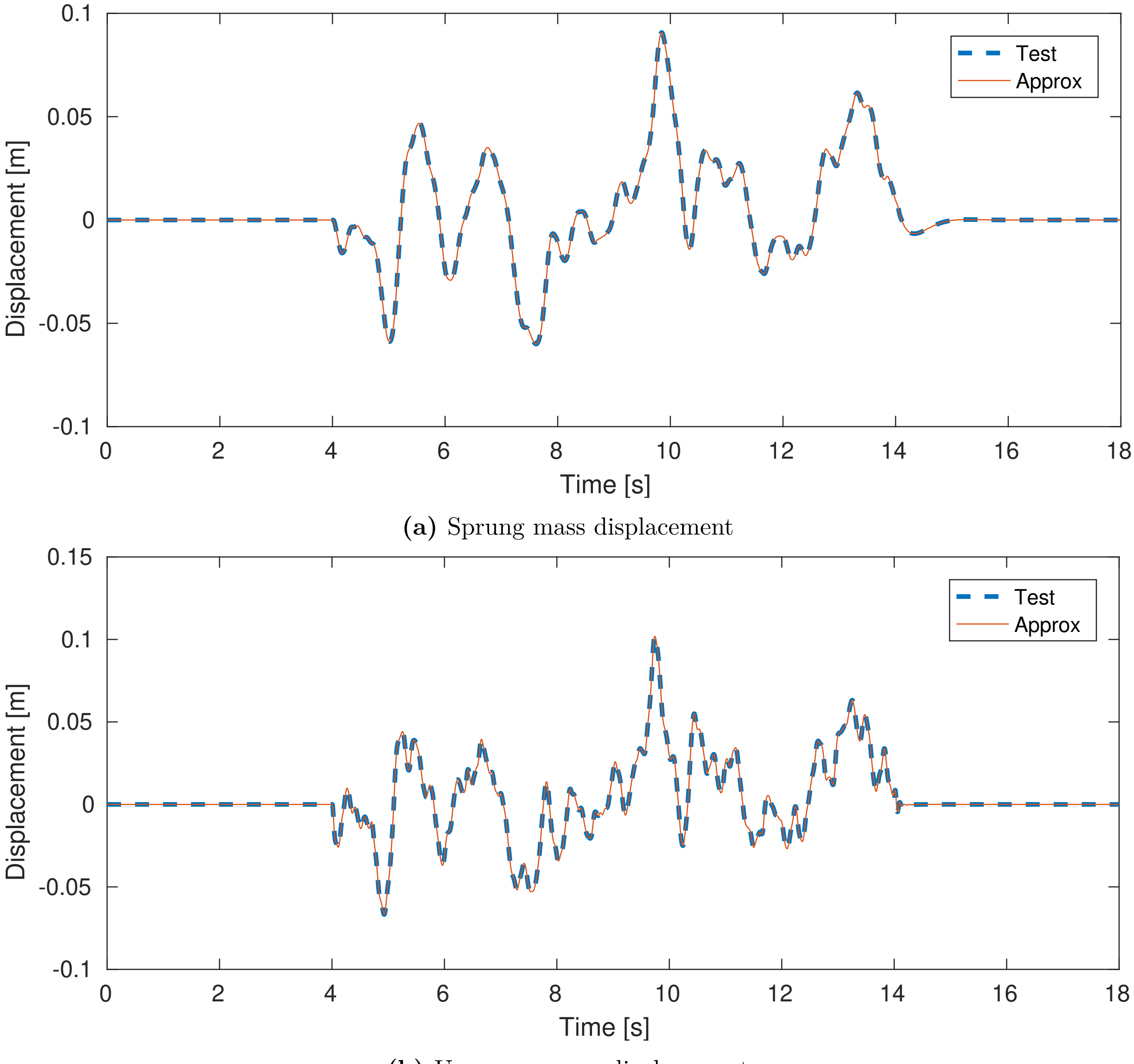
Table 5.3: Experimental design for non-overlapping windowing method performance. (Variables of interest shown first)

Variable	**Details**
Window method	No overlap
Sensor Configuration	Absolute displacement on both DOF, Relative Displacement + Sprung mass acceleration
Window Length $\mathbf{T_w}$	$\in [0.1\text{ s}, 6\text{ s}]$ with 25 equally spaced divisions.
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced divisions
Regression method	SBTR
QC parameters	Default values but without the nonlinear term k_{NL}; Table 2.1
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

Table 5.4: MFFE [%] results for the approximated input and output signals using non-overlapping windows and SBTR on the linear quarter car model.

	Training			**Validation**			**Test**		
	$\mathbf{u}_{\text{road}}$	$\mathbf{z_A}$	$\mathbf{z_R}$	$\mathbf{u}_{\text{road}}$	$\mathbf{z_A}$	$\mathbf{z_R}$	$\mathbf{u}_{\text{road}}$	$\mathbf{z_A}$	$\mathbf{z_R}$
Abs. Disp.	5.4×10^{-4}	5.1×10^{-4}	6.1×10^{-4}	5.8×10^{-4}	1.6×10^{-2}	2.3×10^{-2}	1.8×10^{-2}	1.8×10^{-2}	1.7×10^{-2}
	$\mathbf{u}_{\text{road}}$	$\mathbf{\ddot{z}_A}$	$\mathbf{\Delta_z}$	$\mathbf{u}_{\text{road}}$	$\mathbf{\ddot{z}_A}$	$\mathbf{\Delta_z}$	$\mathbf{u}_{\text{road}}$	$\mathbf{\ddot{z}_A}$	$\mathbf{\Delta_z}$
Rel. Disp. + Acc.	66.11	47.11	38.96	64.64	28.07	32.61	75.49	80.72	46.66

magnitude difference in the test MFFE outputs when comparing the two different sensor set-ups. This is visually confirmed if we compare the recreated test output signals in Figures 5.4 and 5.5. The input results between the two sensor set-ups are compared in Figure 5.6. It becomes obvious when zooming into the recreated inputs in Figure 5.6b that the acceleration and relative displacement sensor set-up experiences sharp discontinuities when moving from one non-overlapping window to the next. This is clearly noticeable at $T \approx 9.45$ s and $T \approx 10.05$ s. The discontinuities in the inputs result in large spikes in the acceleration signal as seen in Figure 5.5a. It is not visually noticeable whether the absolute displacement set-up experiences this jump between each window.

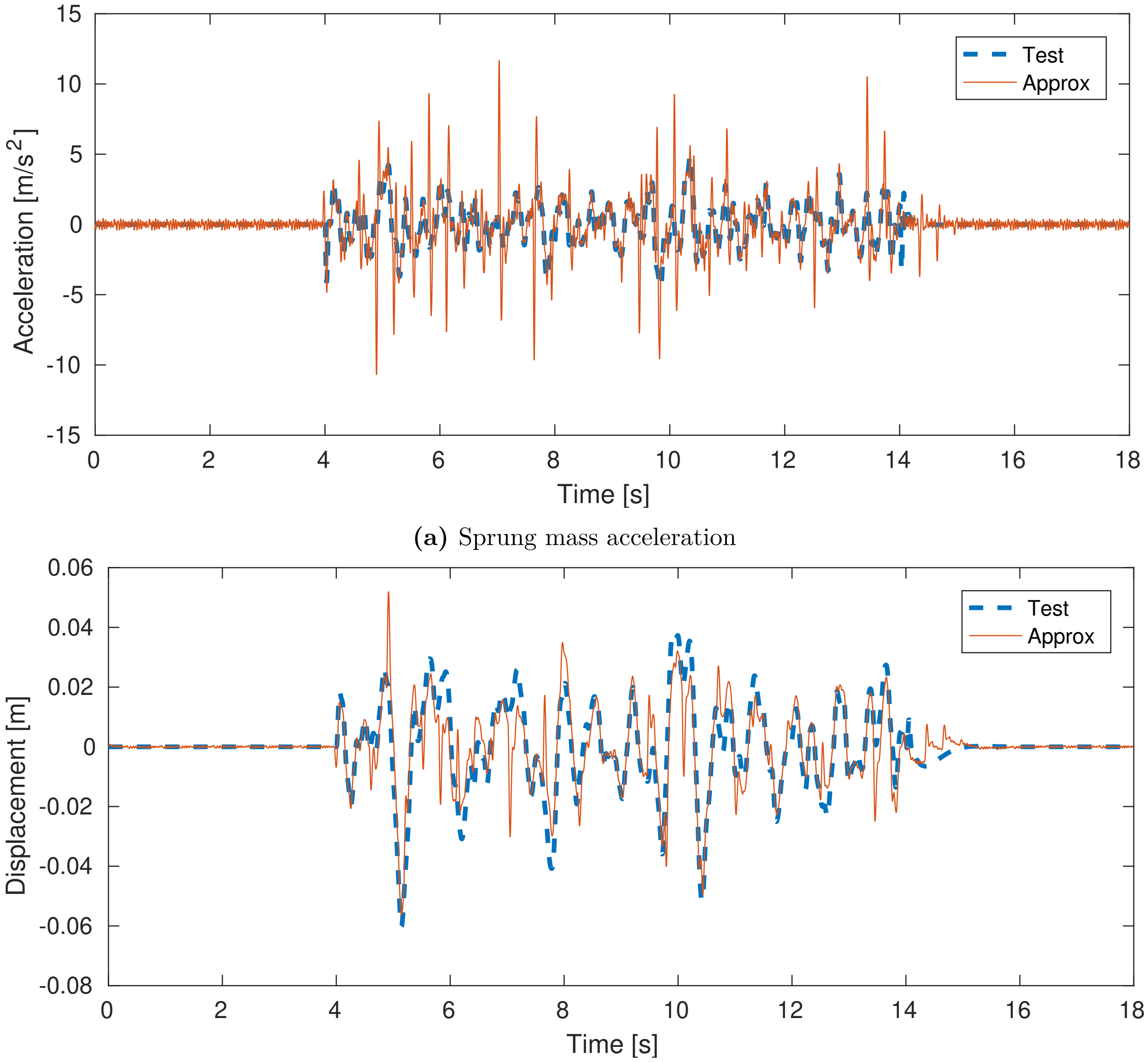


(a) Sprung mass displacement

(b) Unsprung mass displacement

Figure 5.4: Recreated outputs using non-overlapping windows and SBTR. Absolute displacement set-up.

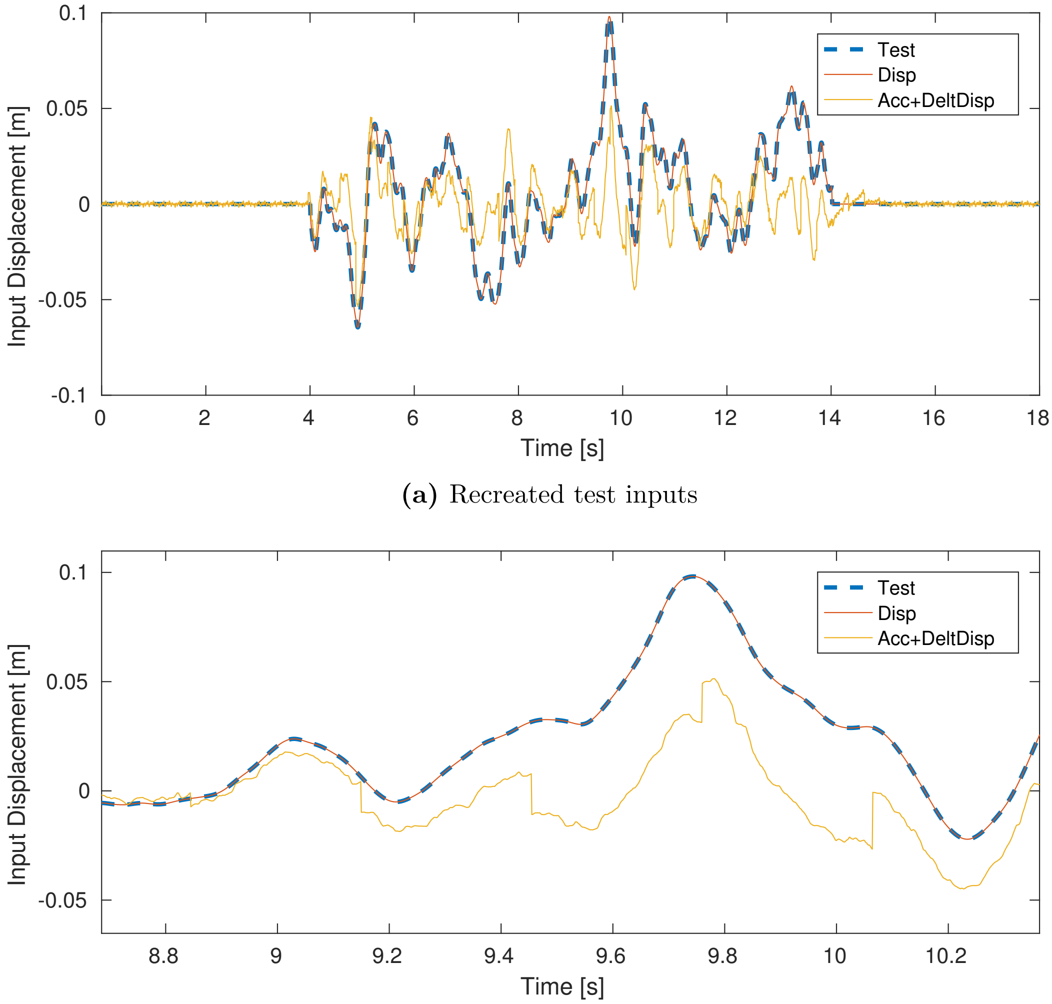
There are two potential causes proposed for the discrepancy in performance. The first is that the stiffness associated with the unsprung mass is too high. If this is the case, the relative displacement between the input and sprung mass will be negligible. The second cause for the discrepancy is more fundamental in that the system fails to approximate integration.



(a) Sprung mass acceleration

(b) Spring Displacement

Figure 5.5: Recreated outputs using non-overlapping windows and SBTR. Acceleration and spring displacement set-up.



(a) Recreated test inputs

(b) Recreated test inputs, zoomed in

Figure 5.6: Comparison between the acceleration and spring displacement set-up and the absolute displacement set-up test inputs. Non-overlapping windows and SBTR are used.

Trivially stiff tyre stiffness

If the tyre is too stiff, its absolute displacement would act as an accurate proxy for the input. This results in a trivial problem. This is demonstrated in Figure 5.7, where the absolute displacement of the unsprung mass is plotted against the road profile input for decreasing values of the unsprung mass stiffness k_R. For the large stiffness values, i.e. $k_R > 10000$, the output displacements lie close to the inputs. For the less stiff values, the output diverges from the inputs. To test the influence of the unsprung mass stiffness k_R, the absolute displacement set-up experiment will be repeated except with a range of k_R values. The MFFE scores for the different stiffness values are recorded in Table 5.5. We note that the MFFE test scores for the inputs and outputs increase with a decreasing sprung mass stiffness, however, the test scores are all less than 1%. This indicates that the tyre stiffness does play a role in the accuracy of the non-overlapping window implementation but not enough to explain the discrepancy between the acceleration and relative displacement set-up and the absolute displacement set-up results.

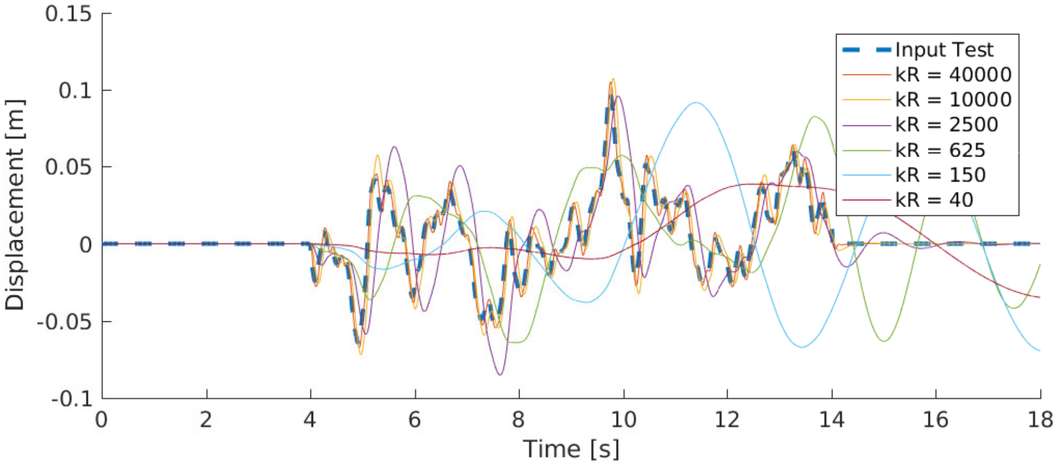


Figure 5.7: The blue dashed line represents the road profile input. The coloured solid lines represent the absolute displacement of the unsprung mass at different stiffness values. The stiff value displacements are almost identical to the input.

Need for integration

The second potential cause for the disparity is that the acceleration set-up needs to perform integration in order to obtain all the $\mathbf{z}$ terms in Equations (2.1) and (2.2). Whereas in the absolute displacement set-up differentiation needs to be performed to obtain all the z terms. In an ideal case differentiation can be numerically approximated with just two points in time allowing for almost instant inversion. Integration would, on the other hand, require a full history of points in time to approximate the integral or, in the case of damping, up to the point in time where the energy would have dissipated. The assumption of the non-overlapping window SBTR method is that each window is an independent observation. The regression technique is therefore unaware of any inputs or outputs occurring outside of each window. This concept of integration approximation will be explored further in Section 5.6.

We can visually get an intuition of this by plotting the errors in each independent window on top of one another to obtain a sense of where the errors occur within each window. The absolute displacement set-up and the acceleration and spring displacement set-up window errors are plotted in Figures 5.8 and 5.9 respectively. If we look at the errors for the input of the absolute displacement in Figure 5.8c we notice large spikes in the beginning and ends of the windows. The regression method has few or no preceding or proceeding outputs at the extremes of the window and is therefore penalised in terms of prediction accuracy. These spikes quickly die out

Table 5.5: Input and output reconstruction MFFE [%] using non-overlapping windows and SBTR for different unsprung mass stiffness values k_R

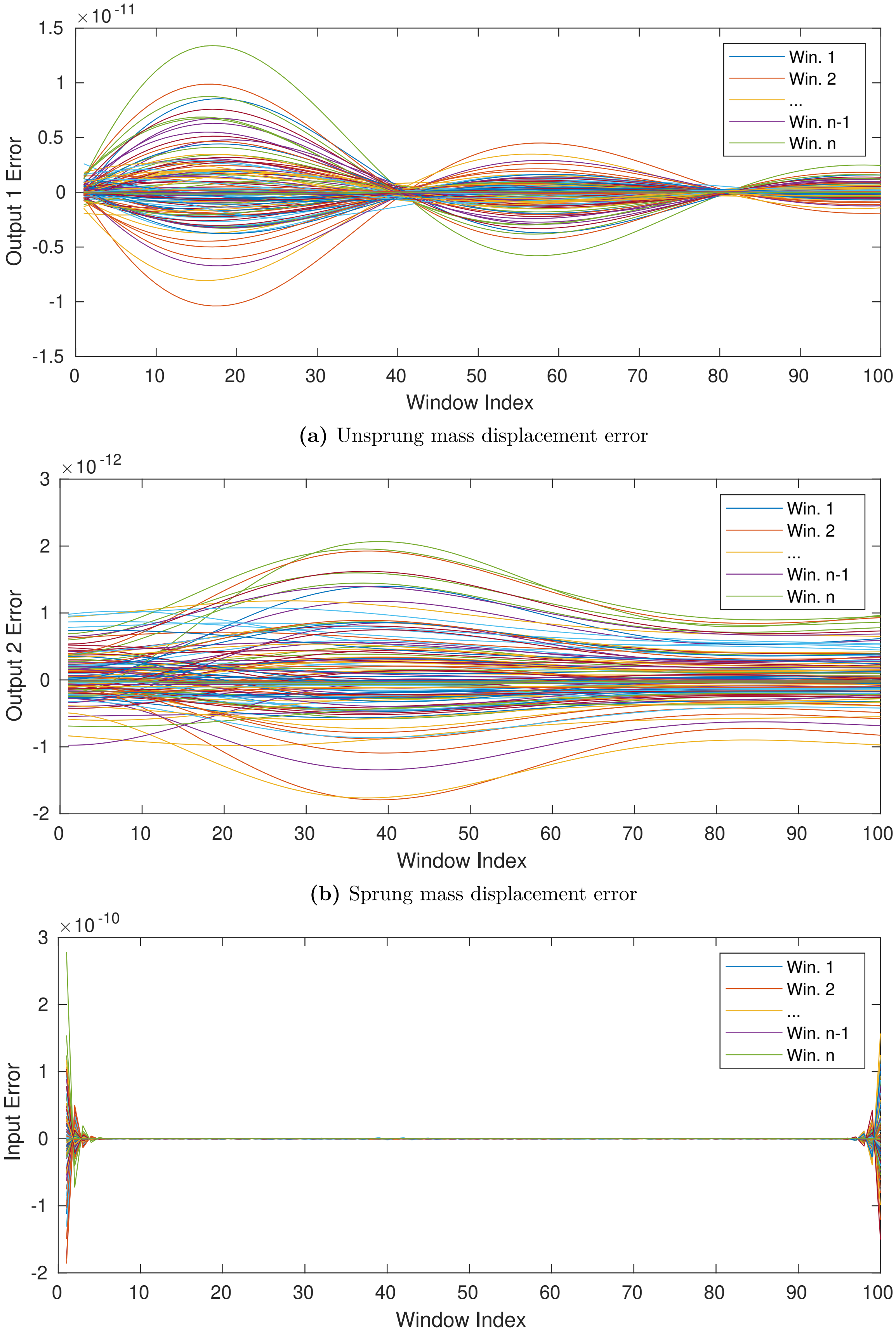
	Training			Validation			Test		
$\mathbf{k_R}$	$\mathbf{u_{road}}$	$\mathbf{z_A}$	$\mathbf{z_R}$	$\mathbf{u_{road}}$	$\mathbf{z_A}$	$\mathbf{z_R}$	$\mathbf{u_{road}}$	$\mathbf{z_A}$	$\mathbf{z_R}$
40000	3.4×10^{-3}	3.2×10^{-3}	3.8×10^{-3}	3.6×10^{-8}	8.5×10^{-3}	1.1×10^{-2}	4.0×10^{-3}	3.9×10^{-3}	3.7×10^{-3}
10000	9.0×10^{-4}	8.1×10^{-4}	8.8×10^{-4}	2.4×10^{-6}	1.5×10^{-2}	1.8×10^{-2}	8.8×10^{-3}	8.0×10^{-3}	7.6×10^{-3}
2500	3.4×10^{-3}	5.2×10^{-3}	5.3×10^{-3}	3.8×10^{-4}	1.2×10^{-2}	1.1×10^{-2}	4.0×10^{-3}	3.0×10^{-3}	2.5×10^{-3}
625	2.9×10^{-3}	9.6×10^{-2}	1.1×10^{-1}	3.0×10^{-3}	2.0×10^{-1}	2.2×10^{-1}	1.1×10^{-2}	2.7×10^{-2}	3.9×10^{-2}
150	6.0×10^{-3}	4.0×10^{-2}	4.1×10^{-2}	1.4×10^{-1}	8.2×10^{-2}	8.4×10^{-2}	1.3×10^{-2}	6.6×10^{-2}	6.6×10^{-2}
40	1.5×10^{-1}	4.4×10^{-1}	4.4×10^{-1}	2.8×10^{0}	1.1×10^{-1}	1.1×10^{-1}	1.9×10^{-1}	9.0×10^{-2}	9.0×10^{-2}

since the derivatives can be quickly estimated using a small number of points. The error spikes then manifest themselves as delayed errors in the outputs of the system as seen in Figures 5.8a and 5.8b. The output errors then converge towards the ends of the window. The output errors of the acceleration and spring displacement set-up, as shown in Figures 5.9a and 5.9b, follow a similar trend of starting off widely distributed in the beginning of the window while becoming narrowly distributed towards the end of the window. However, the corresponding input errors as plotted in Figures 5.9c follow a different trend as the error distribution increases along the length of the window.

If we examine the optimised hyper-parameter results of the cross validation in Table 5.6 we note that the acceleration and spring displacement set-up is highly regularised with 12 out of a potential 86 latent variables being used. In the absolute displacement set-up the system used all of the potential latent variables. The system was limited by the number of singular values available in the decomposition input matrix U. Due to the nature of the SBTR algorithm setting the number of latent variables from the input and output matrices equal to one another, this has placed an artificial lower limit in the number of latent variables being extracted from the output matrix Z. We also note that the absolute displacement set-up used the smallest allowed window length which is another indication that the absolute displacement system can be inverted almost immediately.

Table 5.6: Optimised hyper-parameter results and corresponding windowed matrix sizes for the numerical investigation of non-overlapping windows.

Sensors	**L**	$\mathbf{L_{max}}$	$\mathbf{T_W}$[s]	$\mathbf{n \times r}$	$\mathbf{n \times p}$
Abs. Disp.	100	100	0.10	263 × 100	263 × 200
Rel. Disp. + Acc.	12	86	0.305	86 × 305	86 × 610

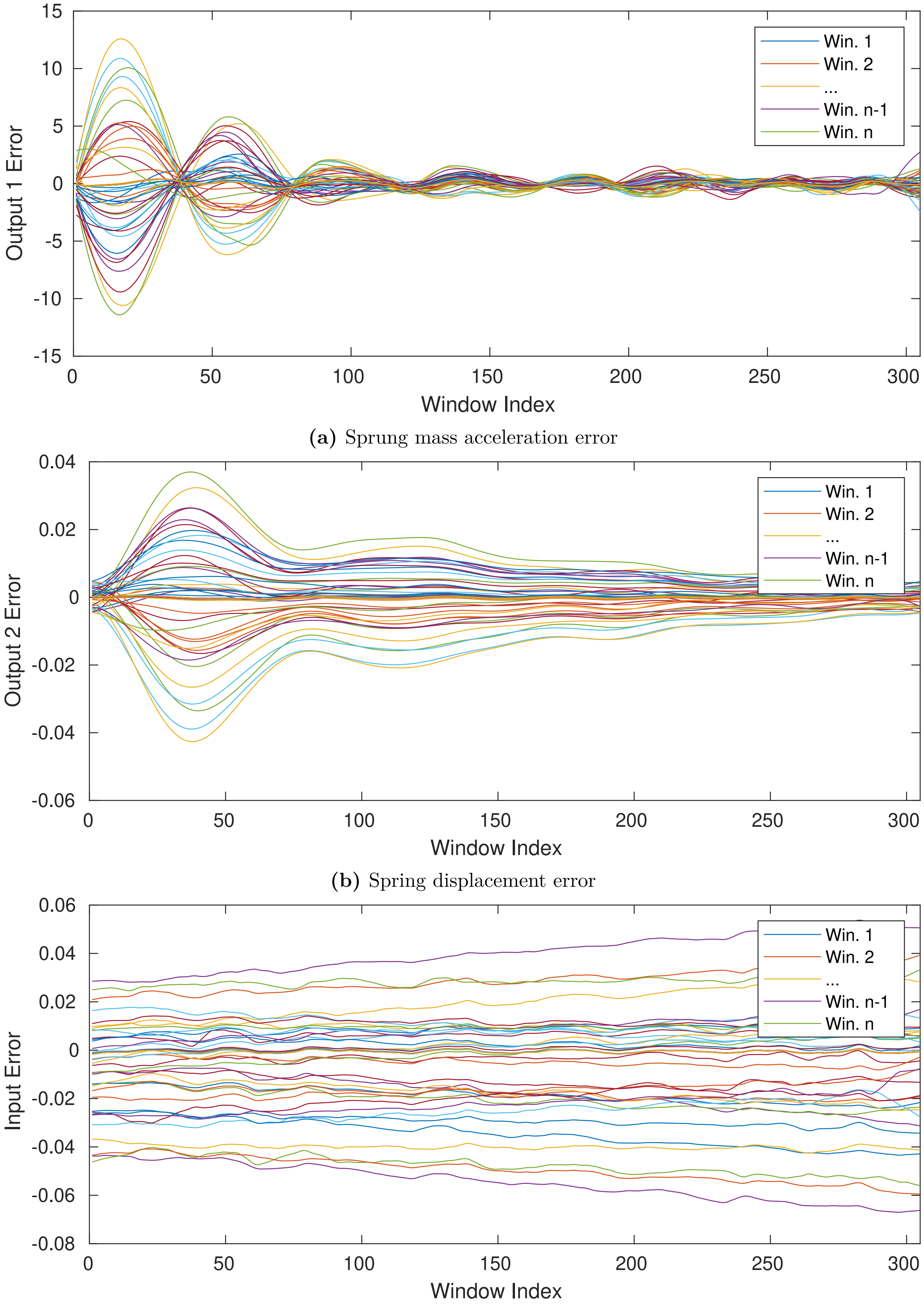


(a) Unsprung mass displacement error

(b) Sprung mass displacement error

(c) Input displacement error

Figure 5.8: Output and input errors plotted for each sample in each window. Absolute displacement sensor set-up.



(a) Sprung mass acceleration error

(b) Spring displacement error

(c) Input displacement error

Figure 5.9: Output and input errors plotted for each sample in each window. Acceleration and spring displacement sensor set-up.

5.1.2 Conclusion

In this numerical investigation a substantial flaw in the implementation of the non-overlapping window method in response reconstruction has been highlighted. The method performs well when predicting the inputs of the system when absolute displacements of the system are used but performs poorly when using a more practical sensor set-up of acceleration and spring displacement. The investigation was performed on a base-line system where no nonlinearities or measurement noise were introduced. It was shown that having a high spring stiffness in the system may make the system input easier to obtain in the case of the absolute displacement set-up but it was not the main cause of discrepancy. The most likely cause of the difference in performance is that in the absolute displacement set-up, the algorithm needs to approximate differentiation in order to obtain all the states of the system. In the acceleration and spring displacement set-up the algorithm needs to perform integration over the entire time history to approximate the states of the system. This places different constraints on parameters of the windows.

In response reconstruction the selection of sensors may be limited due to practical reasons and therefore we would want a response reconstruction method that is applicable to a wide range of reasonable sensor set-ups. A more practical issue in terms of the algorithm is that of minimizing the known errors in the regions of the window that are known to be high. One way of overcoming this is to use overlapping windows and then use the parts of the predicted window that are believed to have low errors and discard the rest.

5.2 Overlapping Windows

In Section 1.7 it was highlighted that, by using non-overlapping windows, errors were introduced by the discontinuities between each window. Two possible solutions to this problem are explored in this chapter. The first proposed solution is to introduce overlaps between each window and then to only use parts of the window that are believed to predict outputs with low errors. The second proposed solution is to introduce overlap but instead of discarding the unwanted sections, the predicted overlapping windows are blended using smooth weighted averaging. In both approaches we need to optimize for a suitable overlap sample length. The overlap sample length s_γ is defined by a proportion γ of the proposed window sample length s_w, i.e.

$$s_\gamma = \lfloor \gamma s_w \rfloor \,. \tag{5.2}$$

The stride of the window, s_τ, is then given by

$$s_\tau = s_w - s_\gamma. \tag{5.3}$$

This results in the same number of columns p and r defined by Equations (1.7) and (1.8) except now the number of rows or observations, n, is equal to

$$n = \left\lfloor \frac{m - s_\gamma}{s_\tau} \right\rfloor . \tag{5.4}$$

The target matrix Y therefore has the form

$$Y = \begin{array}{c} \begin{array}{cccccc} \multicolumn{4}{c}{\overbrace{\hspace{26em}}^{\text{channel 1}}} & & \overbrace{\hspace{2em}}^{\text{channel 2 etc.}} \end{array} \\ \begin{bmatrix} u_1(1) & u_1(2) & \cdots & u_1(s_w-1) & u_1(s_w) & \cdots \\ u_1(s_\tau+1) & u_1(s_\tau+2) & \cdots & u_1(s_\tau+s_w-1) & u_1(s_\tau+s_w) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\ u_1((n-2)s_\tau+1) & u_1((n-2)s_\tau+2) & \cdots & u_1((n-2)s_\tau+s_w-1) & u_1((n-2)s_\tau+s_w) & \cdots \\ u_1((n-1)s_\tau+1) & u_1((n-1)s_\tau+2) & \cdots & u_1((n-1)s_\tau+s_w-1) & u_1((n-1)s_\tau+s_w) & \cdots \end{bmatrix} \end{array}. \tag{5.5}$$

The predictor matrix X takes on a similar form (not shown).

The simplest way of reconstructing the approximated input $\tilde{U}$ is to take s_γ samples from the beginning of each observation in window i in the total number of observations n. This is demonstrated in Figure 5.10a. We do this for each given actuator signal j in the total number of actuators q, which results in

$$\hat{U}_{[(i-1)s_\tau+1:(i)s_\tau,j]} = \hat{Y}_{[i,(j-1)s_w+1:(j-1)s_w+s_\tau]}. \tag{5.6}$$

It is also possible to construct the input using the ends of each window, i.e.

$$\hat{U}_{[s_\gamma+(i-1)s_\tau+1:s_\gamma+(i)s_\tau,j]} = \hat{Y}_{[i,(j)s_w-s_\tau:(j)s_w]}. \tag{5.7}$$

However we noted earlier that the errors are large at the extremes of the window. Therefore s_γ samples should rather be taken from the middle of the window. The starting point s_α is given by

$$s_\alpha = \left\lfloor \frac{s_w - s_\gamma}{2} \right\rfloor, \tag{5.8}$$

which means $\hat{U}$ can be constructed as

$$\hat{U}_{[(i-1)s_\tau+s_\alpha:(i)s_\tau+s_\alpha-1,j]} = \hat{Y}_{[i,(j-1)s_w+s_\alpha:(j-1)s_w+s_\alpha+s_\tau-1]}. \tag{5.9}$$

An overview of the middle sampling method is shown in Figure 5.10b.

We may find that at times small unwanted discontinuities may arise between the stitched together segments. The solution is to use a weighted average of prediction that overlap with one another to smooth these discontinuities. An overview of the weighted averaging merging is shown in Figure 5.11.

This creates an intermediate averaged windowed target matrix $\hat{Y}^*$. This process can be expressed as

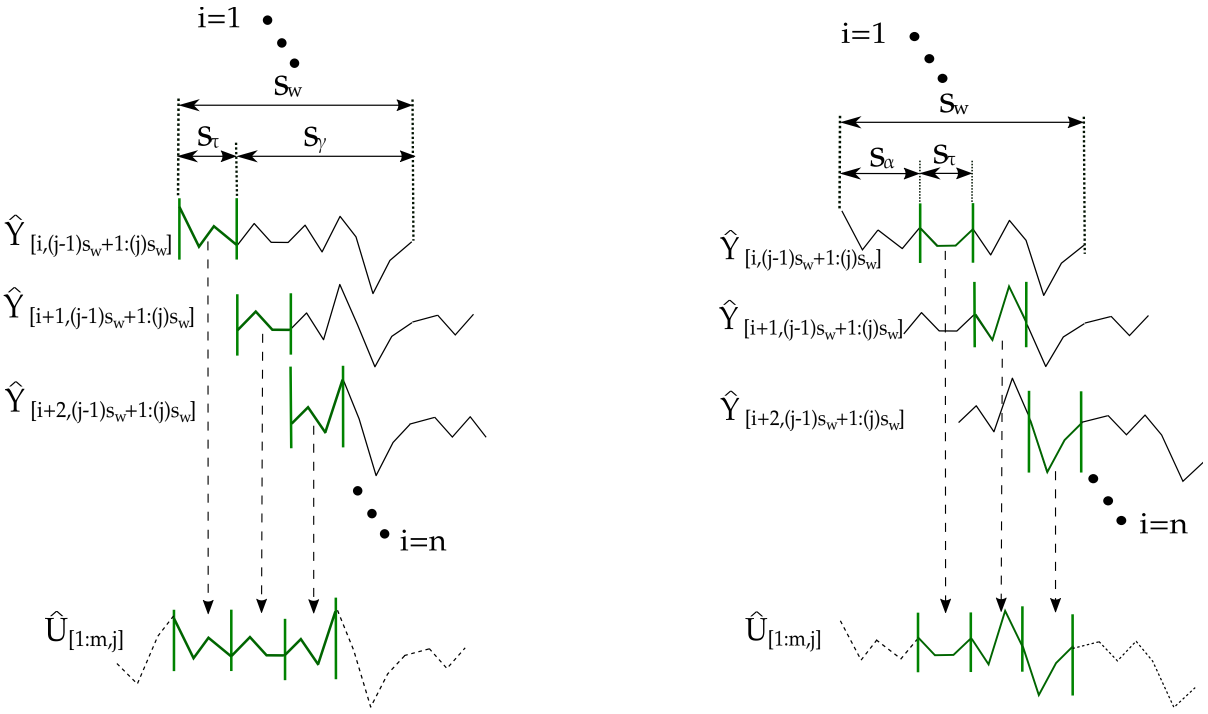
$$\hat{Y}^*_{[i,(j-1)s_w+s_\tau+1:(j)s_w]} = (w) \circ \hat{Y}_{[i,(j-1)s_w+s_\tau+1:(j)s_w]} + (1-w) \circ \hat{Y}_{[i+1,(j-1)s_w+1:(j-1)s_w+s_\gamma]} \tag{5.10}$$

where $\circ$ denotes the Hadamard product. Since the first s_τ samples do not overlap, the first s_τ from the intermediate weighted average matrix remains the same as the original windowed matrix, i.e.

$$\hat{Y}^*_{[i,(j-1)s_w+1:(j-1)s_w+s_\tau]} = \hat{Y}_{[i,(j-1)s_w+1:(j-1)s_w+s_\tau]}. \tag{5.11}$$

If we define w (for the case of two overlapping signals) as

$$w = 0.5 \cdot \mathbb{1} \tag{5.12}$$



(a) First merging: the first s_τ for each observation i , with window length s_w, of the approximate target matrix $\hat{Y}$ is used to reconstruct approximate input $\hat{U}$. s_y denotes the amount of overlap between each successive observation.

(b) Middle merging: s_τ samples beginning at s_α are taken for each observation i , with window length s_w, of the approximate target matrix $\hat{Y}$ to reconstruct the approximate input $\hat{U}$.

Figure 5.10: An overview of merging overlapping windows.

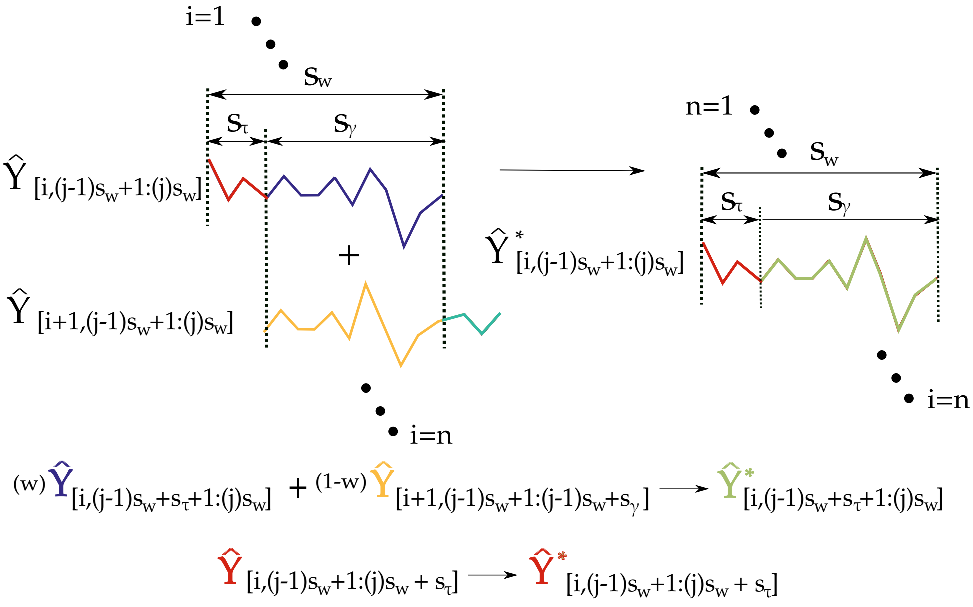

i=1
Sw
Sτ
Sγ
Ŷ [i,(j-1)sw+1:(j)sw]
+
Ŷ [i+1,(j-1)sw+1:(j)sw]
i=n
n=1
Sw
Sτ
Sγ
Ŷ* [i,(j-1)sw+1:(j)sw]
i=n
(w) Ŷ[i,(j-1)sw+sτ+1:(j)sw] + (1-w) Ŷ[i+1,(j-1)sw+1:(j-1)sw+sγ] → Ŷ*[i,(j-1)sw+sτ+1:(j)sw]
Ŷ [i,(j-1)sw+1:(j)sw + sτ] → Ŷ* [i,(j-1)sw+1:(j)sw + sτ]

Figure 5.11: An overview of the weighted average merging method. The overlapping segments of length s_γ for each observation i of $\hat{Y}$ with length s_w are weighted averaged with the corresponding segment in the $(i+1)^{th}$ window using w. An intermediate averaged target matrix $\hat{Y^*}$ is then obtained. Since the first s_τ samples do not overlap the first s_τ samples are taken as is from $\hat{Y}$.

where $\mathbb{1} \in \mathbb{R}^{s_\gamma \times 1}$ is a vector containing ones, we simply treat each signal with equal weighting. An alternative is to define the weight using the sigmoid function, given by

$$w_{k=1}^{s_\gamma} = \frac{1}{1 + e^{-6 + k\frac{12}{s_\gamma}}} \tag{5.13}$$

and plotted in Figure 5.12.

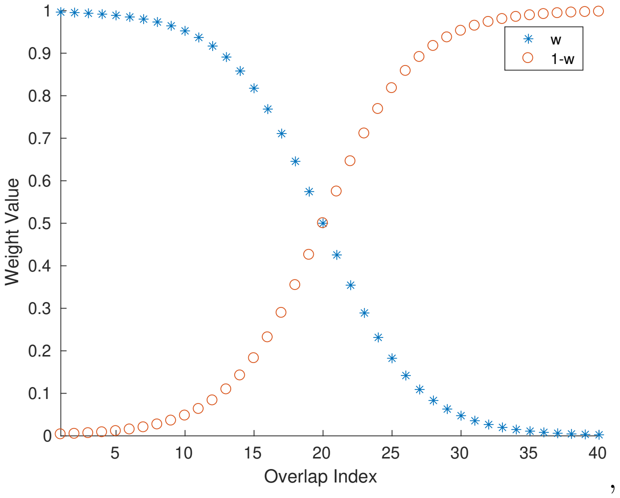


Figure 5.12: Sigmoidal weights used in sigmoidal weighted averaging, created for an overlap with $s_\gamma = 40$ samples.

The w term initially weights the i^{th} window highly with $w = 1$ and then drops off to zero over the length of the overlap. The $1 - w$ initially gives no weight to the $(i+1)^{th}$ window but eventually gives full weight to the $(i+1)^{th}$ window. This attempts to weigh down samples that are sampled from the beginning of each window observation while emphasizing samples that are extracted from the middle of the window. Once the intermediate weighted window predictions $\hat{Y}^*$ are calculated, the approximated input $\hat{U}$ can be created by using the middle sampling method described in Equation (5.9). It can be noted that the i^{th} window may overlap with the $(i+2)^{th}$ window and so on, if the overlap exceeds half the length of the previous window i.e. $\gamma > 0.5$. However, the averaging methods covered in this section do not consider the information contained in the $(i+2)^{th}$ window. We will explore methods that average over more than two windows in Section 5.3.

5.2.1 Numerical Investigation

In this section we will test the different methods of merging overlapping windows. The procedure will follow similarly to that outlined in Section 5.1, however, in this case the overlap proportion γ needs to be determined by grid search as well. The candidate overlap values are $\gamma \in [0, 0.99]$ with 20 equal linearly spaced intervals. In this set-up we will only be focusing on the sprung mass acceleration and spring displacement sensor configuration. Five different window merging techniques will be investigated, namely:

- *First*: the prediction is taken from the beginning of the window.
- *Middle*: the prediction is taken from the middle of the window.
- *Last*: the prediction is taken from the end of the window.
- *Average*: two overlapping windows are averaged with equal weighting.
- *Sigmoidal*: two overlapping windows are averaged using sigmoidal weighting.

The numerical experiment procedure parameters are given in Table 5.7.

Table 5.7: Experimental design for overlapping windows. (Variables of interest shown first).

Variable	Details
Windowing methods	First, Middle, Last, Average, Sigmoidal
Window Proportional overlap γ	$\in [0, 0.99]$ with 20 equally spaced intervals
Sensor configuration	Sprung mass acceleration + spring displacement
Window lengths $\mathbf{T_w}$	$\in [0.1\text{ s}, 6\text{ s}]$ with 25 equally spaced divisions.
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced divisions
Sampling frequency $\mathbf{f_s}$	1000 Hz
Regression method	SBTR
QC parameters	Default values; Table 2.1 however with no nonlinearity term k_{NL}
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

Results

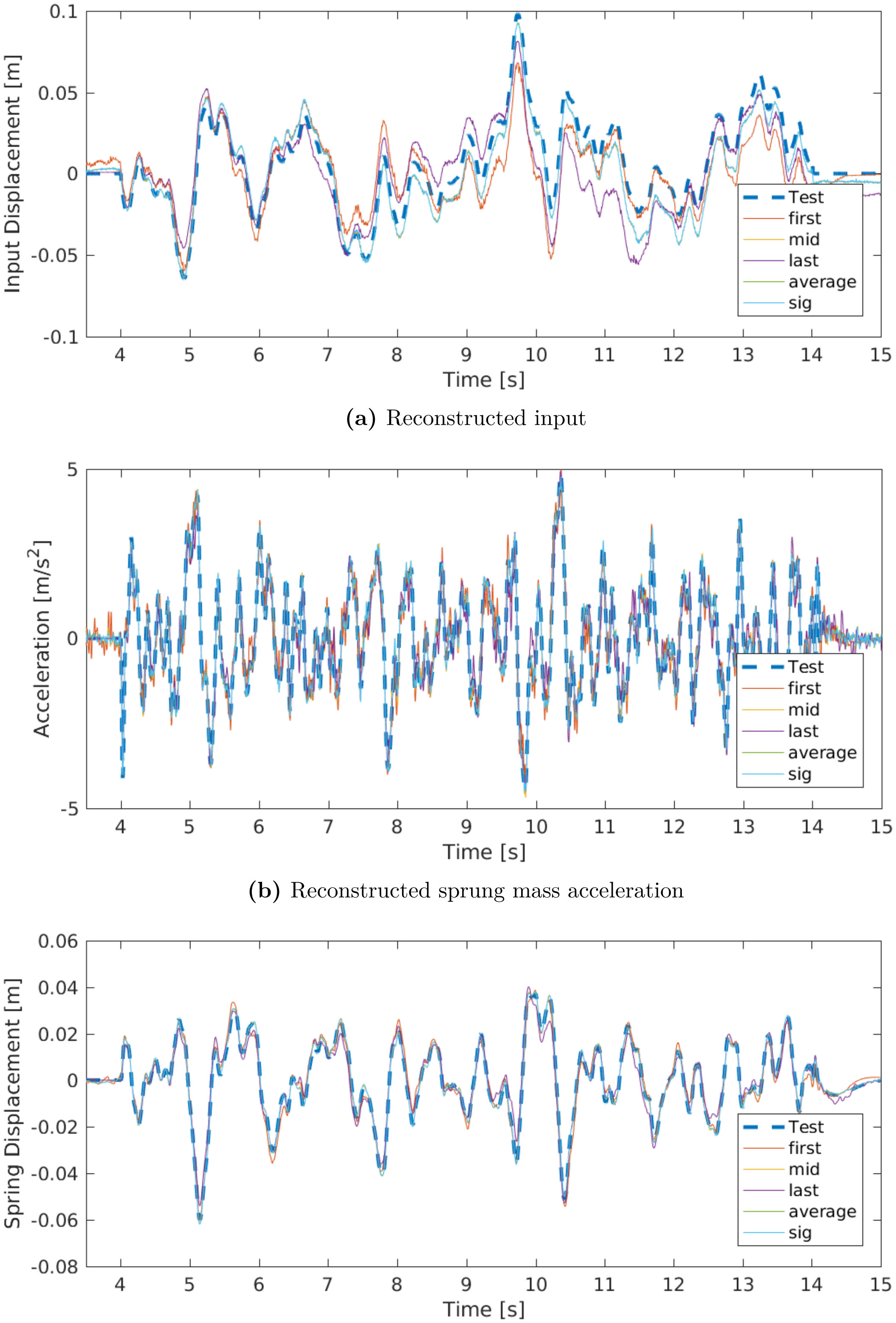
The recreated input and output signals using the different window merging techniques are shown in Figure 5.13.

The MFFE results of the grid search optimisation are shown in Table 5.8.

Table 5.8: MFFE [%] results for the approximated input and output signals using different windowing methods on the linear quarter car model. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

	Training			**Validation**			**Test**		
	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta_z}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta_z}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta_z}$
First	19.06	26.04	13.94	43.70	15.15	10.56	46.97	21.26	14.48
Middle	6.81	13.60	7.56	24.99	12.97	6.05	28.13	10.43	4.92
Last	21.43	9.18	25.92	58.00	19.10	15.48	50.63	22.03	18.94
Average	6.88	12.69	7.35	24.90	9.57	5.58	28.23	9.16	4.66
Sigmoidal	6.81	12.65	7.39	24.77	9.53	5.58	28.27	8.92	4.65

Using the *Last*, or the *First* merging techniques produced the worst results as can be seen in the MFFE results as well as sharp unwanted spikes they produce in the acceleration reconstruction as demonstrated in Figure 5.13b. The *Middle*, *Average* and *Sigmoidal* produced the best results and performed similarly. The weighted averaging techniques, *Average* and *Sigmoidal*, performed slightly better than just using *Middle* alone. When comparing the results from Table 5.8 against those using non-overlapping windows in Table 5.4, we note that all of the investigated merging techniques achieved significantly better results than the non-overlapping window techniques employed for the acceleration and spring displacement set-up.



(a) Reconstructed input

(b) Reconstructed sprung mass acceleration

(c) Reconstructed spring displacement

Figure 5.13: Recreated input and output reconstruction results using different overlapping window methods with SBTR. The tests were performed on a linear quarter car.

The hyper-parameter results of the grid search optimisation are shown in Table 5.9.

Table 5.9: Optimized hyper-parameter results for the numerical investigation of non-overlapping windows and corresponding training matrix sizes.

	$\mathbf{L_{retained}}$	$\mathbf{L}_{\max}$	$\mathbf{T_w}$[s]	γ	$\mathbf{n} \times \mathbf{r}$	$\mathbf{n} \times \mathbf{p}$
First	210	633	3.542	0.99	633 × 3542	633 × 7084
Middle	344	474	4.525	0.99	474 × 4524	474 × 9048
Last	331	633	3.542	0.99	633 × 3542	633 × 7084
Average	341	474	4.525	0.99	474 × 4524	474 × 9048
Sigmoidal	344	474	4.525	0.99	474 × 4524	474 × 9048

All five windowing methods used the maximum amount of overlap allowed by the grid search to obtain their best results. This is corroborated by looking at the spring displacement output reconstruction MFFE results using *Sigmoidal* merging in Figure 5.14. The MFFE are plotted as a function of proportional overlap and window length. It can be noted that the MFFE drops off quickly as the proportional overlap approaches the maximum allowed value of 0.99. It is clear that the response reconstruction benefits significantly from a large overlap. This suggests that even better reconstruction accuracy can be achieved by increasing the overlap even further. The optimised window lengths also increased significantly when compared to the window lengths obtained using non-overlapping windows.

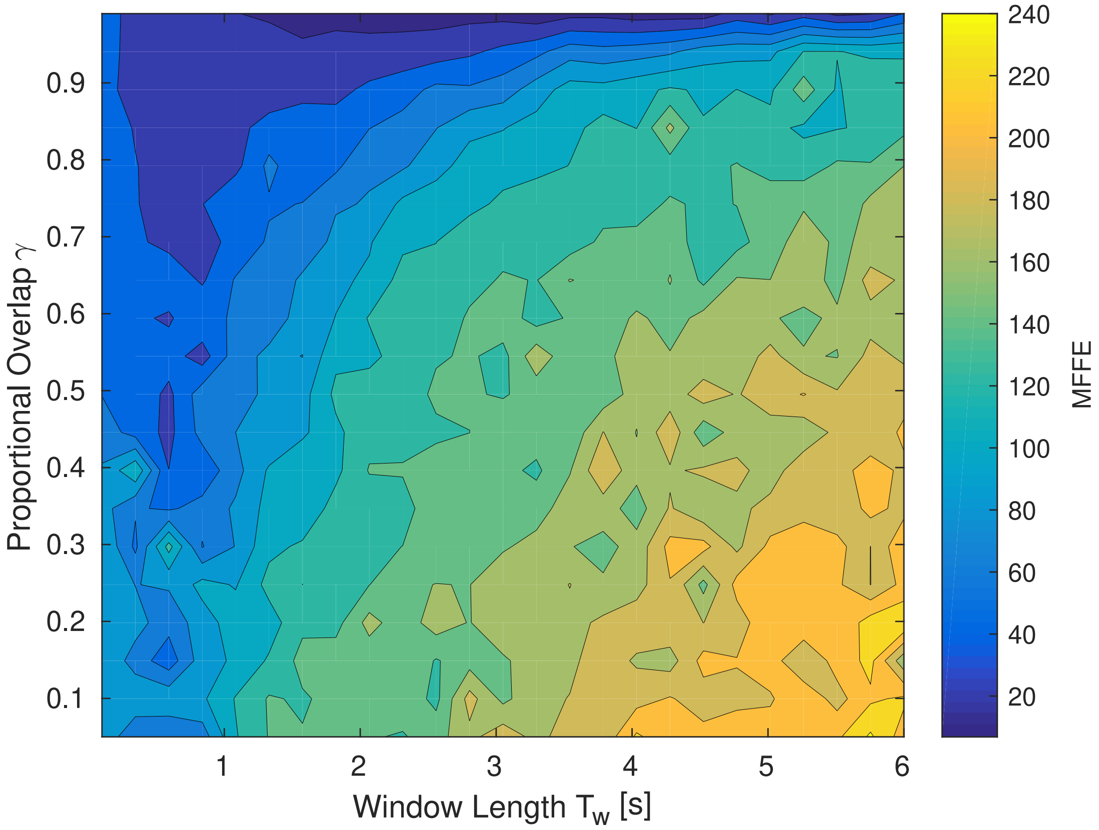


Figure 5.14: Spring displacement reconstruction MFFE [%] as a function of window overlap γ and window length T_w. Sigmoidal window merging was used.

5.2.2 Conclusion

Overall, these results confirm the need for overlapping windows when performing SBTR with systems that need to perform integration to obtain the inputs. The best performance results were achieved when the maximum allowed amount of overlap was used. This suggests that allowing even more overlap may be beneficial. It was shown that selecting the middle section of a reconstructed window produces the lowest error as compared to selecting the beginning or end section of the window.

The slight improvement in MFFE results when employing weighted averaging techniques suggests that combining the different predictions from different windows may be a fruitful method of producing even better results. The averaging techniques only averaged across two overlapping windows even though the overlap existed between more than just two windows. The original but misguided assumption was that introducing some overlap, i.e. $\gamma < 0.5$ would be satisfactory and that averaging over more than two windows would not be required. In the next section the maximum achievable overlap will be investigated as well as a method of using these omitted overlapping segments to create the weighted average.

5.3 AntiDiagonal Averaging (ADA)

We noted in Section 5.2 that better reconstruction results could be achieved by setting the amount of overlapping to the extreme such that the stride is one sample, i.e. $s_\tau = 1$. This results in the following windowed target matrix Y

$$Y = \begin{array}{c} \begin{array}{cc} \overbrace{\hspace{20em}}^{\text{channel 1}} & \overbrace{\hspace{2em}}^{\text{channel 2 etc.}} \end{array} \\ \begin{bmatrix} u_1(1) & u_1(2) & \cdots & u_1(s_w - 1) & u_1(s_w) & \cdots \\ u_1(2) & u_1(3) & \cdots & u_1(s_w) & u_1(s_w + 1) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\ u_1(n-1) & u_1(n) & \cdots & u_1(n + s_w - 3) & u_1(n + s_w - 2) & \cdots \\ u_1(n) & u_1(n+1) & \cdots & u_1(n + s_w - 2) & u_1(n + s_w - 1) & \cdots \end{bmatrix} \end{array}. \tag{5.14}$$

The windowed predictor matrix X takes on a similar form (not shown). We can simply average over the anti-diagonals of the windowed data matrix $\hat{Y}$ to reconstruct the approximated input $\hat{U}$. To compute the average response $\hat{u}(k)$ we average all the anti-diagonal terms of $\hat{Y}_{i,j}$, such that

$$\hat{u}(k) = \frac{1}{n_{\text{diag}}} \sum \hat{Y}_{i,j}, \tag{5.15}$$

for which $i + j = k + 1$ and n_{diag} is the number of elements in the anti-diagonal. This process is known as Hankelization, called ADA in this dissertation, which is the same process followed in Singular Spectral Analysis (SSA) [23]. SSA is a time series analysis tool. In time series analysis the inputs of the system of interest are typically unknown or infeasible to measure. Time series analysis is typically used in the analysis and forecasting of large complex systems such as climatology and economics [18]. Time series are then cast as autoregressive problems for prediction purposes whereby current outputs of the system are a function of past outputs such that

$$z(k) = f(z(k-1), \ldots, z(k-n)). \tag{5.16}$$

In SSA outputs of the system are windowed using a stride of $s_\tau = 1$. The corresponding windowed matrix is referred to as the trajectory matrix. This is known as the embedding step in SSA. The windowed matrix is then decomposed using SVD. The singular values of the system are then analysed to separate different types of signals such as linear trends, polynomials and sinusoids from the noise of the system. The separated signals are then reconstructed using ADA to achieve an approximated time signal. SSA is closely related to Proper Orthogonal Decomposition (POD) typically used in engineering problems [10]. In POD the separate observations are separated in space so in effect they use the matrix form given in Equation (1.4) whereas in SSA the lagged windows of the signal are used as different observations. In this case we are simply borrowing the concept of ADA from the reconstruction step used in SSA but instead using it in a regression problem.

An example signal with $m = 7$ samples windowed with maximum overlap and window length $s_w = 3$ results in the following equation

$$\begin{bmatrix} \hat{u}(1) & \hat{u}(2) & \hat{u}(3) \\ \hat{u}(2) & \hat{u}(3) & \hat{u}(4) \\ \hat{u}(3) & \hat{u}(4) & \hat{u}(5) \\ \hat{u}(4) & \hat{u}(5) & \hat{u}(6) \\ \hat{u}(5) & \hat{u}(6) & \hat{u}(7) \end{bmatrix} = \begin{bmatrix} z(1) & z(2) & z(3) \\ z(2) & z(3) & z(4) \\ z(3) & z(4) & z(5) \\ z(4) & z(5) & z(6) \\ z(5) & z(6) & z(7) \end{bmatrix} \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \beta_{1,3} \\ \beta_{2,1} & \beta_{2,2} & \beta_{2,3} \\ \beta_{3,1} & \beta_{3,2} & \beta_{3,3} \end{bmatrix}. \tag{5.17}$$

Here z is the response signal used to predict the inputs u. The linear coefficients β are computed using any of the techniques presented in Chapter 3. To gain insight into the workings of ADA we can write out the set of equations that infer $\hat{u}(3)$, i.e.

$$\hat{u}(3)_1 = \beta_{1,3}z(1) + \beta_{2,3}z(2) + \beta_{3,3}z(3), \tag{5.18}$$

$$\hat{u}(3)_2 = \beta_{1,2}z(2) + \beta_{2,2}z(3) + \beta_{3,2}z(4), \tag{5.19}$$

$$\hat{u}(3)_3 = \beta_{1,1}z(3) + \beta_{2,1}z(4) + \beta_{3,1}z(5). \tag{5.20}$$

We can then average over all the $\hat{u}(3)$ predictions to get the final prediction of $\hat{u}(3)$

$$\hat{u}(3) = \frac{1}{3}\left(\hat{u}(3)_1 + \hat{u}(3)_2 + \hat{u}(3)_3\right) \tag{5.21}$$

$$= z(1)\left(\frac{\beta_{1,3}}{3}\right) + z(2)\left(\frac{\beta_{2,3}+\beta_{1,2}}{3}\right) + z(3)\left(\frac{\beta_{3,3}+\beta_{2,2}+\beta_{1,1}}{3}\right) + z(4)\left(\frac{\beta_{3,2}+\beta_{2,1}}{3}\right) + z(5)\left(\frac{\beta_{3,1}}{3}\right). \tag{5.22}$$

If we rewrite the average of the β multiplying with a particular z term as a new constant e.g. $\beta_2 = \frac{\beta_{2,3}+\beta_{1,2}}{2}$, we obtain

$$\hat{u}(3) = \beta_1\frac{1}{3}z(1) + \beta_2\frac{2}{3}z(2) + \beta_3 1z(3) + \beta_4\frac{2}{3}z(4) + \beta_5\frac{1}{3}z(5). \tag{5.23}$$

Here we note that the ADA emphasizes the middle most term with decreasing emphasis placed on proceeding and preceding terms. It in effect creates a triangular windowing function. If we add a corresponding weight term w e.g. $w_2 = \frac{2}{3}$ we can rewrite the equation generally as

$$\hat{u}(k) = \beta_1 w_1 z(k - s_w) + \cdots + \beta_{s_w} w_{s_w} z(k) + \cdots + \beta_{k+s_w} w_{k+s_w} z(k + s_w). \tag{5.24}$$

This result demonstrates that ADA is an indirect method of creating a weighted moving average filter. In system identification this is known as a Finite Impulse Response (FIR) model. More specifically this an example of a non-causal weighted FIR model. The weights can be arbitrary and are a prior design choice. If we forgo the ADA method and use the weighted FIR model we can be more creative with the choice of weighting. An overview and comparison of ADA against FIR models are given in Section 7.5.

5.3.1 Numerical Investigation

In order to compare the performance of the ADA window merging method, the investigation of overlapping windows conducted in Section 5.2.1 will be repeated. This time the overlap will be fixed such that the overlap is maximal. In other words the stride of the window is one sample, i.e. $s_\mathcal{T} = 1$. The overlapping windowing methods previously covered will be included as well as the ADA windowing method. An overview of the numerical investigation parameters are shown in Table 5.10.

Table 5.10: Experimental design for maximum overlap. (Variables of interest shown first).

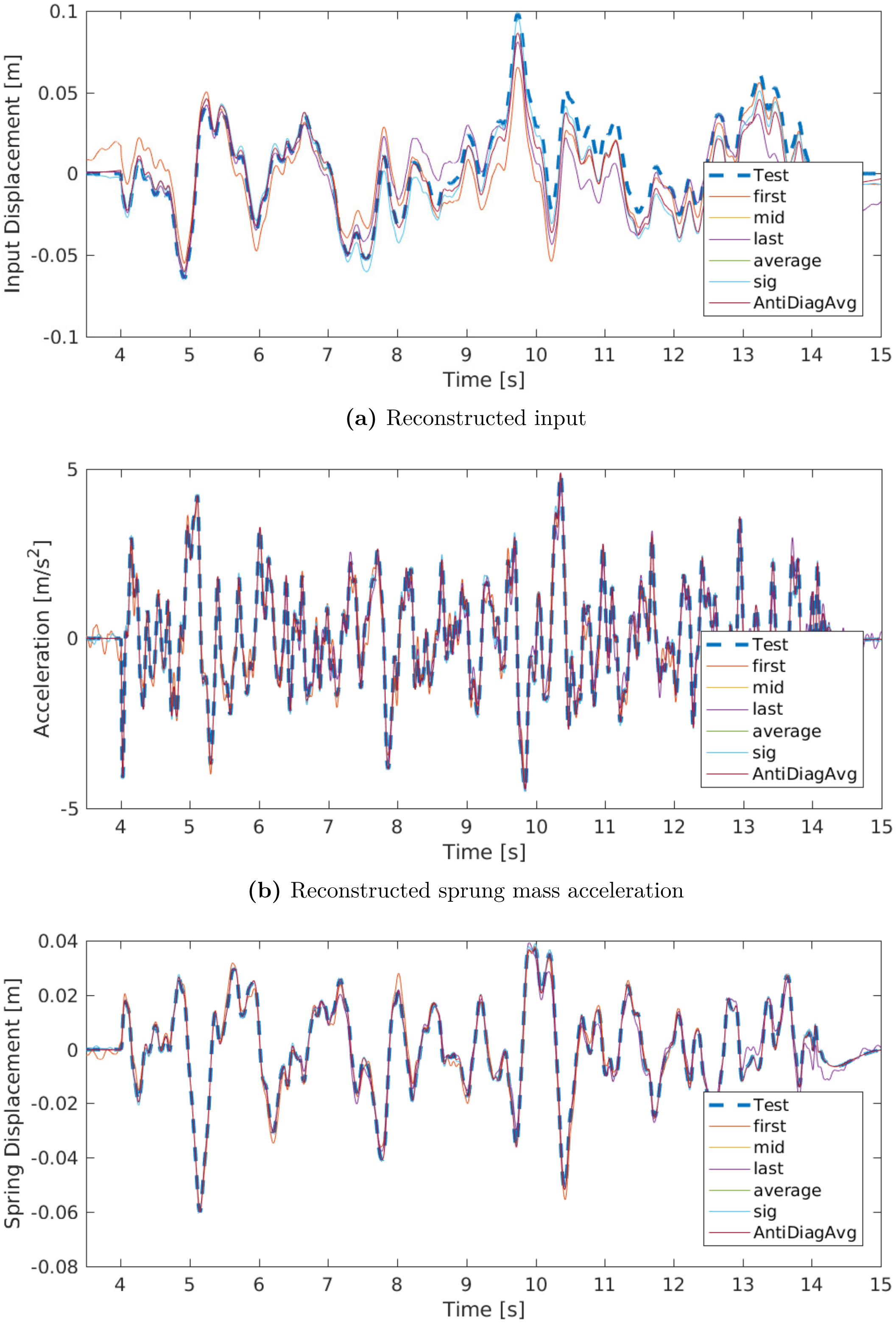
Variable	**Details**
Windowing methods	First, Middle, Last, Average, Sigmoidal, ADA
Window proportional overlap γ	Maximum
Sensor Configuration	Sprung mass acceleration + spring displacement
Window lengths $\mathbf{T_w}$	$\in [0.1\ \mathrm{s}, 6\ \mathrm{s}]$ with 25 equally spaced divisions.
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced divisions
Regression methods	SBTR
QC parameters	Default values; Table 2.1 however with no nonlinearity term k_{NL}
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

Results

The recreated input and output signals using the different window merging techniques are shown in Figure 5.15. The reconstruction MFFE results of the grid search optimisation are shown in Table 5.11.

Table 5.11: MFFE [%] results for the approximated input and output signals using different windowing methods with maximum overlap on the linear quarter car model. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility as discussed in Section 2.3.

	Training			**Validation**			**Test**		
	$\mathbf{u}_{\mathrm{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\mathrm{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\mathrm{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$
First	17.52	9.24	10.98	32.24	8.20	7.03	40.83	14.39	12.43
Middle	6.46	2.50	6.43	13.51	2.71	1.97	23.37	4.94	3.75
Last	21.20	10.56	14.92	26.84	9.82	8.92	47.43	17.05	13.20
Average	6.51	2.57	6.43	13.47	2.68	2.15	23.37	4.91	3.75
Sigmoidal	6.49	2.54	6.43	13.51	2.71	1.97	23.37	4.91	3.75
ADA	7.85	0.45	0.37	20.77	0.11	0.46	28.49	0.46	1.41



(a) Reconstructed input

(b) Reconstructed sprung mass acceleration

(c) Reconstructed spring displacement

Figure 5.15: Recreated input and output results using different overlapping window methods with maximal overlap.

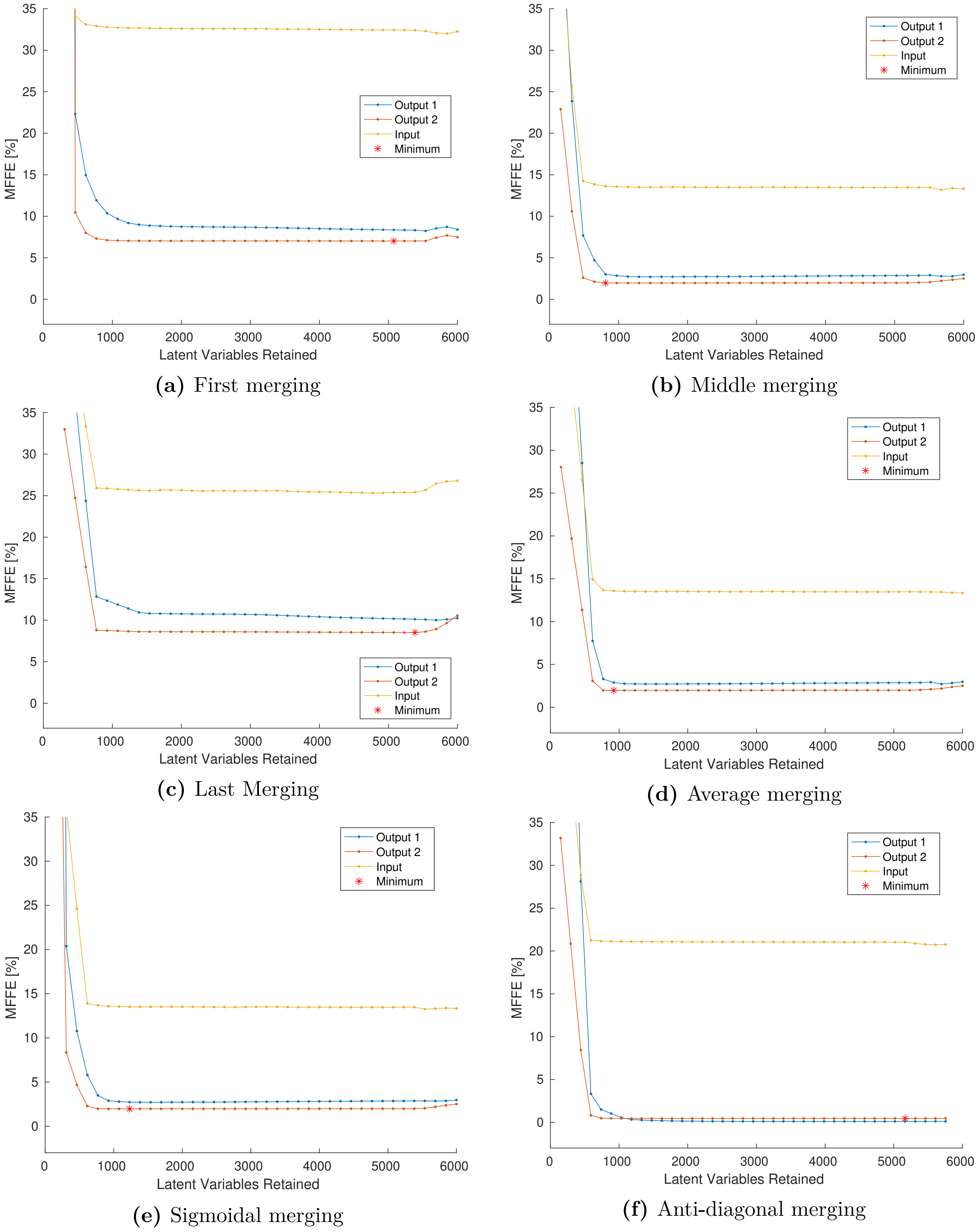
All of the previously tested merging methods achieved a small improvement in MFFE results when compared to the results obtained in Table 5.8. This small improvement comes at a high cost of computational effort due to the massive increase in the window matrices that need to be be decomposed. However, if we inspect the ADA results we note a marked improvement in results. The test results for the reconstructed acceleration signal $\ddot{z}_A$ dropped by an order of magnitude when compared to the second best result achieved by *Sigmoidal* merging. This confirms our hypothesis that averaging over multiple windowed observations will produce better response reconstruction results. It is interesting to note that while the output reconstruction MFFE results decreased, the corresponding input MFFE for ADA actually increased. This is acceptable for the case of response reconstruction since we know that this current sensor set-up allows for an off-set in the input results as discussed in Section 2.3.

The hyper-parameter results of the grid search optimisation are shown in Table 5.12. The number of latent variables available as well as the number of latent variables used have also increased when compared to the results obtained in Table 5.9. The latent variable results indicate that the ADA method allows for the use of a larger proportion of latent variables than the other weighted average methods, i.e. *sigmoidal* and *average*.

Table 5.12: Hyper-parameter results and corresponding training matrix sizes for the numerical investigation of overlapping windows with maximum overlap on the linear quarter car model.

	$\mathbf{L_{retained}}$	$\mathbf{L}_{\max}$	$\mathbf{T_w}$[s]	$\mathbf{n \times r}$	$\mathbf{n \times p}$
First	5417	6000	6.000	20307 × 6000	20307 × 12000
Middle	1189	6000	6.000	20307 × 6000	20307 × 12000
Last	5468	6000	6.000	20307 × 6000	20307 × 12000
Average	1002	6000	6.000	20307 × 6000	20307 × 12000
Sigmoidal	1054	6000	6.000	20307 × 6000	20307 × 12000
ADA	5421	5754	5.754	20553 × 5754	20553 × 11508

In order to investigate the effects of the number of retained latent variables on the windowing method, we need to look at the validation results in the latent variable grid-search. To this end, the latent variable grid-search results are shown in Figure 5.16. Each plot represents the validation MFFE results obtained during the latent variable grid-search at the optimal window length value T_w for each windowing method. All of the plots show a quick drop in MFFE as the number of latent variables retained are increased. The majority of the MFFE values then plateau or decrease extremely slowly before rising slightly at the end. This indicates the point at which the effects of the inversion of the smaller singular values begin to become noticeable. In this case the upper limit of the number of latent variables extracted from the predictor matrix X is limited by the size of the target matrix Y since the inputs have less channels than the outputs. If we instead use PCR, we can increase the upper limit of the number of the number of latent variables extracted since the number of extracted latent variables is only dependent on the size (and rank) of X. The latent variable grid-search results using PCR are shown in Figure 5.17. Here, the detrimental effects of using too many latent variables on the output validation performance becomes more apparent. The exception to this is the ADA windowing method which, in both the SBTR and PCR cases, is not penalised by retaining a large number of latent variables. ADA appears to only suffer from choosing too few latent variables. This indicates that the ADA is implementing another layer of regularisation by imposing a structure onto the shape of the β coefficients. However, it is still necessary to perform cross validation to determine the number of latent variables to retain due to effects such as measurement noise and model mismatch as outlined in Sections 7.2 and 7.3. In this investigation and throughout this document the full range of latent variables is used for the cross validation step. This is done to illustrate the need for regularisation as well as to ensure consistent results across experiments. A more practical approach would be to stop searching once the validation results have plateaued.



(a) First merging

(b) Middle merging

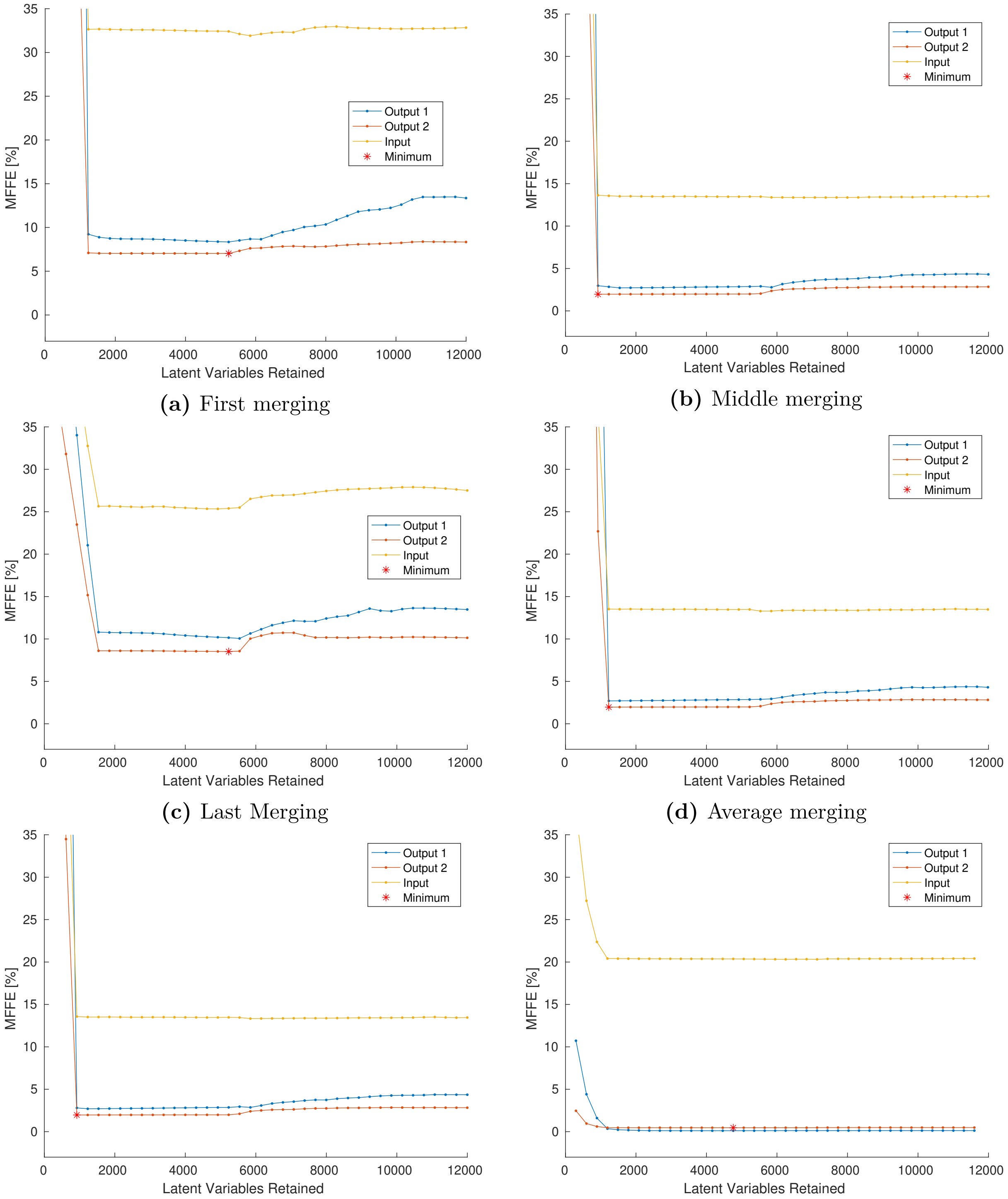
(c) Last Merging

(d) Average merging

(e) Sigmoidal merging

(f) Anti-diagonal merging

Figure 5.16: Latent variable validation grid-search results for different window merging methods using SBTR.



(a) First merging

(b) Middle merging

(c) Last Merging

(d) Average merging

(e) Sigmoidal merging

(f) Anti-diagonal merging

Figure 5.17: Latent variable validation grid-search results for different window merging methods using PCR

5.4 Complexity

It is worth comparing the matrix sizes created by the different windowing methods used to solve the response reconstruction problem. The main computational cost in the regression methods covered is the SVD, which scales $O(\min(np^2, n^2p))$. The matrix sizes for the best performing windowing techniques for the spring displacement and acceleration problem in Sections 1.7, 5.2 and 5.3 are compared in Table 5.13. The corresponding number of operations to perform SVD are also stated. The first thing to note is that the complexity of the problem changes fundamentally when we go from no overlap to using ADA. The number of operations required increase by 5 orders of magnitude. The SBTR algorithm is now penalised twice by the increased complexity, since it needs to decompose both the target Y and predictor matrix X. This is in comparison to regression methods such as PCR and RR which only need to decompose the predictor matrix. The introduction of overlapping has somewhat undermined the main benefit of using the non-overlapping windows method which is its relative computational ease. Clearly, it is understandable that we would wish to use a windowing method that uses no overlaps since the matrices are substantially smaller but by adding overlap we allow the algorithm to solve a more general class of problems. In all the performed numerical experiments, a highly conservative sampling frequency of 1000 Hz was used so using a more suitable sampling frequency will alleviate this burden somewhat.

Table 5.13: Training set sizes using different methods of windowing and the corresponding number of operations needed to perform SVD.

Window Method	Size		No. of Operations	
	X	Y	X	Y
No Overlap	86×610	86×305	2.26×10^6	4.51×10^6
Overlap ($\gamma = 0.99$)	474×9048	474×4524	1.02×10^9	2.03×10^9
ADA	20553×11508	20553×5754	6.80×10^{11}	2.72×10^{12}

5.5 Varying the Stride with ADA

We showed in the Section 5.4 how the complexity of the problem rapidly increases when the overlap approaches the maximum. At one extreme we explored the method of taking variable overlaps but we discarded most of the information when making a prediction. On the other hand we took the maximum overlap and used all the information to make a prediction by taking the average of all the windows. In this section we will investigate whether there is a happy medium of the two by exploring the idea of varying the stride and still averaging the prediction.

ADA will be performed on the linear and nonlinear quarter car as done in Section 5.3. The experiment will be conducted at a range of stride values. The experimental setup overview is given in Table 5.14.

5.5.1 Result

The MFFE results of the linear and nonlinear experiments are shown in Table 5.15. For the linear case the best validation MFFE for the recreated spring displacement occurs with a stride of 16. In the nonlinear case the best recreated validation spring displacement occurs at a stride length of 2. This is not a significant reduction as compared to the linear case but the stride can be pushed to between 4 and 8 for the nonlinear case without the recreation error degrading too much. An obvious reason for the eventual degradation of the performance as the stride

Table 5.14: Experimental design for ADA with varying strides. (Variables of interest shown first).

Variable	Details
Window Strides $\mathbf{s}_\tau$	$[2, 4, 8, 16, 32, 64]$
Windowing method	ADA
Sensor Configuration	Sprung mass acceleration and spring displacement
Window lengths $\mathbf{T_w}$	$\in [0.1\text{ s}, 6\text{ s}]$ with 25 equally spaced divisions.
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced divisions
Regression method	SBTR
QC parameters	Default values; Table 2.1
Nonlinearity Constant $\mathbf{k_{NL}}$	Linear case: 0; Nonlinear case: $1.28 \times 10^7\ \mathrm{N\,m^{-3}}$
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

increases is that the number of latent variables available eventually decreases. This can be seen by looking at hyper-parameters in Table 5.16. In both the linear and nonlinear case going from a stride of 1 to 2 yields the same number of latent variables but comes with the benefit of reduced computational and memory cost. An interesting point to note is that the proportion of latent variables retained decreases as the stride length increases. This indicates that the problem requires more regularization as the stride is increased.

Doubling the stride of the windowing techniques for the a given window length halves the number of observations. The complexity scales $O(\min(np^2, n^2p))$, therefore halving the matrix size can reduce the complexity by half or a quarter. Varying the stride is a viable option to decrease memory and computational cost but it introduces another hyper-paramter that needs to be optimised. A practical solution to this would be to fix the stride to 1 and instead use the sampling frequency as means of reducing the computational expense. A means of choosing the sampling frequency is given in Section 6.1.

5.6 ADA as a Generalized Black-Box Model

In Section 5.1.1 it was highlighted that one of the reasons why the non-overlapping window method failed was due to its inability to implicitly approximate integration. This integration is required to estimate the unmeasured states of the system. The same problem arises in state-space control whereby the full states of the system are needed to properly control the system. It is not always possible to measure these states. So typically an "observer" is designed whereby the system dynamics are used to estimate these unmeasured states through the means of a Kalman filter. However it is known that a properly implemented general black-box model will implicitly parameterise this observer [34, 28]. The goal of this section is to demonstrate ADA's fitness as a generalised black-box model.

To contrast this black-box approach, we could instead use non-overlapping windows as we would need to numerically integrate the acceleration signals twice to obtain displacements and

Table 5.15: MFFE [%] results for the approximated input and output signals using different stride lengths with ADA. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

Linear Case									
	Training			**Validation**			**Test**		
$\mathbf{s}_{\mathcal{T}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$
1	7.79	0.12	0.49	21.03	0.11	0.46	31.65	0.86	2.09
2	7.80	0.12	0.49	20.69	0.11	0.45	31.65	0.86	2.09
4	8.45	0.13	0.51	20.96	0.13	0.45	31.84	0.79	2.05
8	8.52	0.33	0.51	21.08	0.47	0.46	31.86	0.84	2.06
16	6.72	0.55	0.29	18.61	1.31	0.41	32.84	1.56	2.00
32	7.62	0.59	0.45	16.93	2.13	0.84	33.73	1.96	1.42
64	1.43	0.27	0.16	23.61	4.90	2.00	35.06	7.27	2.55

Nonlinear Case									
	Training			**Validation**			**Test**		
$\mathbf{s}_{\mathcal{T}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\boldsymbol{\Delta}_{\mathbf{z}}$
1	6.89	0.21	0.16	36.76	0.26	0.22	32.49	1.29	1.97
2	6.88	0.21	0.16	36.76	0.26	0.21	32.49	1.29	1.97
4	6.88	0.23	0.17	36.78	0.27	0.22	32.51	1.28	1.97
8	6.85	0.24	0.23	38.46	0.61	0.30	31.99	0.79	1.23
16	4.47	0.19	0.13	51.59	1.43	0.83	41.62	2.63	2.70
32	8.34	1.04	0.66	69.58	4.47	2.31	66.90	5.10	4.49
64	20.89	4.91	3.44	55.63	11.18	6.08	127.81	17.94	17.47

Table 5.16: Hyper-parameter results and corresponding training matrix sizes for the numerical investigation of ADA with differing stride lengths on the linear quarter car model.

Linear Case

$\mathbf{s}_\tau$	$\mathbf{L_{retained}}$	$\mathbf{L}_{\max}$	$\mathbf{T_w}$[s]	$\mathbf{n \times r}$	$\mathbf{n \times p}$
1	5607	5753	5.75	20557 × 5750	20557 × 11500
2	5476	5754	5.75	10279 × 5750	10279 × 11500
4	2644	5139	5.75	5140 × 5750	5140 × 11500
8	1095	2570	5.75	2570 × 5750	2570 × 11500
16	749	1270	6.00	1270 × 6000	1270 × 12000
32	487	681	4.52	681 × 4520	681 × 9040
64	330	349	2.07	379 × 2067	379 × 4134

Nonlinear Case

$\mathbf{s}_\tau$	$\mathbf{L_{retained}}$	$\mathbf{L}_{\max}$	$\mathbf{T_w}$[s]	$\mathbf{n \times r}$	$\mathbf{n \times p}$
1	5508	5508	5.51	20799 × 5508	20799 × 11016
2	5508	5508	5.51	10400 × 5508	10400 × 11016
4	3656	5161	5.51	5200 × 5508	5200 × 11016
8	1917	2631	5.26	2631 × 5262	2631 × 10524
16	1229	1377	4.28	1377 × 4279	1377 × 8558
32	502	750	2.31	750 × 2312	750 × 4624
64	208	391	1.33	391 × 1329	391 × 2658

use these as inputs. This shifts the methodology away from a black-box approach towards a grey-box approach whereby we need to tailor the methodology using a physical understanding of the system for it to work. A drawback of this approach is that of sensor drift. Any noise in the raw acceleration signal will accumulate as an error as a result of numerical integration, further obscuring the dynamics of the system.

5.6.1 Numerical Investigation

To contrast these two methods we will set-up a numerical experiment whereby we measure the accelerations on both DOFs of the quarter-car model. We will then numerically integrate these signals to get the velocities and then integrate again to get the displacements of both DOFs. At each stage of integration the response reconstruction process will be performed using non-overlapping windows and the ADA windowing method. Since we expect that the real-world signals will contain noise, we will repeat the experiment but with a low level of noise. The level of noise for this investigation is defined in percentage terms, $\eta_{\%}$, of the standard deviation for each channel o of the outputs z. The noise is assumed to be Gaussian with zero mean, resulting in

$$\mathbf{z}_{\mathrm{o,noisy}} = \mathbf{z}_{\mathrm{o}} + \mathcal{N}\left(0, \eta_{\%}^2\, \sigma_{z_o}^2\right). \tag{5.25}$$

A noise level of 2% will be used for this experiment. A zoomed in slice of the test sprung mass acceleration signal, comparing the noisy signal to the noise-free signal, is shown in Figure 5.18.

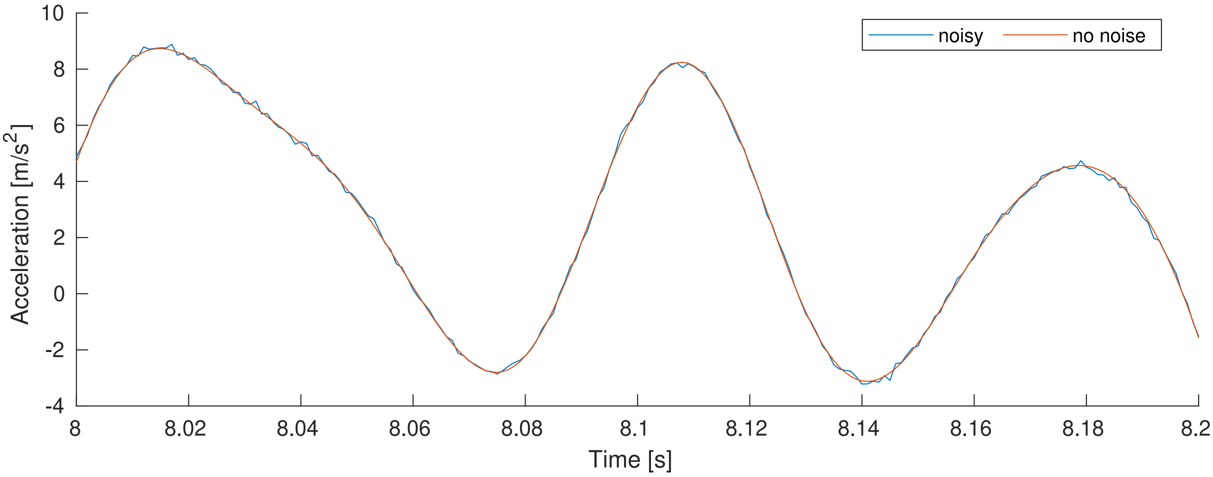


Figure 5.18: A zoomed in slice of the sprung mass acceleration signal with and without noise.

A summary of the experimental design is given in Table 5.17.

5.6.2 Results

The results of the integration test are given in Table 5.18. We note that ADA manages to reconstruct the response of the system using the raw acceleration signals whereas the non-overlapping windows method does not. When we inspect the noise-free results using the states of the system that were obtained using numerical integration we note that ADA manages to reconstruct the responses regardless of what state is fed to it. The non-overlapping window method only ever achieves a decent reconstruction when displacements are used. If we compare the noisy results we note that ADA reconstructs the outputs for both the acceleration and velocity cases fairly well but struggles with the displacement case. It is at this point where the issue of signal drift begins to dominate. In the case of non-overlapping windows we see that it struggles with the noisy case for the displacement case due to the signal drift. The main point of the results is that ADA will implicitly integrate the states of the system with or without noise.

Table 5.17: Experimental design for integration approximation. (Variables of interest shown first).

Variable	**Details**
Sensor Configuration	Raw Acceleration on both DOFs, Velocity (integrated), Displacement (double integrated)
Noise level, $\eta_{\%}$	off and with 2 %
Windowing method	No overlap, ADA
Window length $\mathbf{T_w}$	$\in [0.1\text{ s}, 6\text{ s}]$ with 25 equally spaced divisions.
Window stride $\mathbf{s}_\tau$	1 for ADA
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced divisions
Sampling frequency $\mathbf{f_s}$	1000 Hz
Regression method	SBTR
QC parameters	Default values; Table 2.1, no nonlinearity term k_{NL}
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

Table 5.18: MFFE [%] results for the approximated input and output signals for the integration approximation experiment. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

	Training			**Validation**			**Test**		
Raw Acceleration	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\ddot{\mathbf{z}}_{\mathbf{R}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\ddot{\mathbf{z}}_{\mathbf{R}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\ddot{\mathbf{z}}_{\mathbf{R}}$
ADA	5.63	0.04	0.11	20.56	0.04	0.10	32.04	0.37	0.79
No Overlap	38.48	25.97	22.73	55.46	29.41	21.80	164.50	73.58	44.73
ADA, Noisy	15.84	3.09	3.36	30.46	2.55	2.69	35.66	2.83	2.92
No Overlap, Noisy	36.64	27.11	24.36	60.75	32.02	25.97	167.46	74.25	45.36
Velocity	$\mathbf{u}_{\text{road}}$	$\dot{\mathbf{z}}_{\mathbf{A}}$	$\dot{\mathbf{z}}_{\mathbf{R}}$	$\mathbf{u}_{\text{road}}$	$\dot{\mathbf{z}}_{\mathbf{A}}$	$\dot{\mathbf{z}}_{\mathbf{R}}$	$\mathbf{u}_{\text{road}}$	$\dot{\mathbf{z}}_{\mathbf{A}}$	$\dot{\mathbf{z}}_{\mathbf{R}}$
ADA	3.59	0.11	0.31	9.10	0.17	0.47	16.92	1.03	1.72
No Overlap	47.05	27.89	27.25	53.31	26.56	26.70	91.57	49.81	42.39
ADA, Noisy	5.35	13.63	5.62	14.78	4.93	1.16	24.45	5.04	5.04
No Overlap, Noisy	40.17	25.40	28.90	50.58	18.56	22.04	115.01	48.70	53.74
Displacement	$\mathbf{u}_{\text{road}}$	$\mathbf{z}_{\mathbf{A}}$	$\mathbf{z}_{\mathbf{R}}$	$\mathbf{u}_{\text{road}}$	$\mathbf{z}_{\mathbf{A}}$	$\mathbf{z}_{\mathbf{R}}$	$\mathbf{u}_{\text{road}}$	$\mathbf{z}_{\mathbf{A}}$	$\mathbf{z}_{\mathbf{R}}$
ADA	0.00	0.00	0.00	0.02	0.02	0.02	0.02	0.03	0.02
No Overlap	0.00	0.00	0.00	0.02	0.02	0.02	0.02	0.03	0.02
ADA, Noisy	6.23	94.49	212.24	20.63	3.19	8.73	41.77	136.49	36.85
No Overlap, Noisy	23.46	42.50	134.53	28.36	49.09	22.13	59.54	60.20	63.85

In Figures 5.19 and 5.20 we see that the non-overlapping window again creates non-smooth responses for acceleration and velocity configurations. If we look at the displacement reconstructions in Figure 5.21 we see that for the noisy cases the sensor drift begins to dominate. If we compare the hyperparameters in Table 5.19 we note that ADA and the non-overlapping window both choose smaller window lengths for the case of displacement signals when no noise is present. This indicates that both windowing methods favour shorter windows since differentiation can be performed with shorter windows.

In summary the results indicate that overlapping windows and by extension ADA allow for a wider range of problems to be solved by allowing the regression problem to approximate both numerical integration as well as differentiation.

Table 5.19: Optimized hyper-parameter results and corresponding training matrix sizes for the numerical investigation of integration approximation.

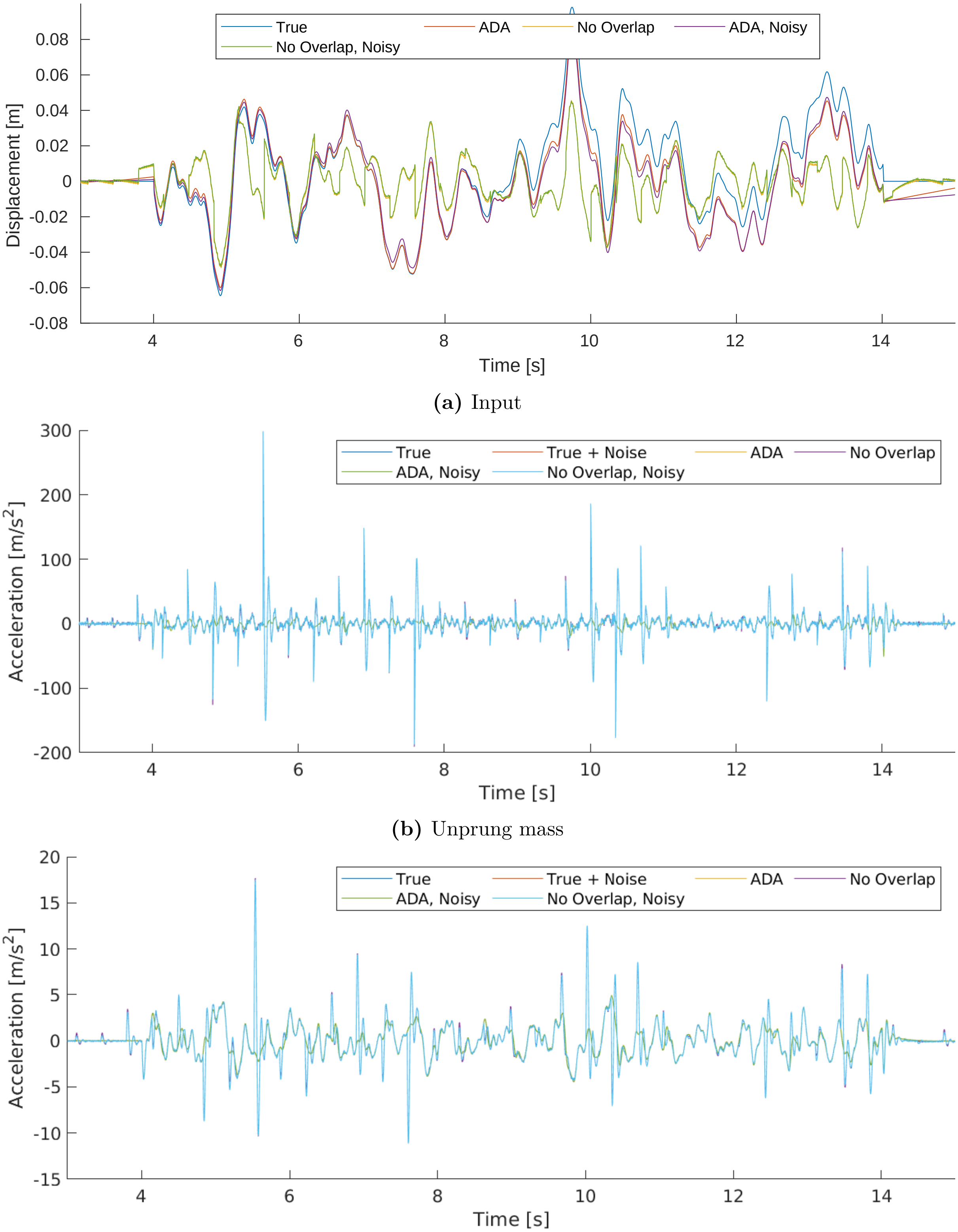
Raw Acceleration	$\mathbf{L}_{\text{retained}}$	$\mathbf{L}_{\text{max}}$	$\mathbf{T_w}$[s]	$\mathbf{n \times r}$	$\mathbf{n \times p}$
ADA	5753	5999	6.00	20307×6000	20307×12000
No Overlap	28	72	0.35	76×345	76×690
ADA, Noisy	5016	5016	5.02	21291×5016	21291×10032
No Overlap, Noisy	30	69	0.35	76×345	76×90
Velocity	$\mathbf{L}_{\text{retained}}$	$\mathbf{L}_{\text{max}}$	$\mathbf{T_w}$[s]	$\mathbf{n \times r}$	$\mathbf{n \times p}$
ADA	5098	6000	6.00	20307×6000	20307×12000
No Overlap	14	100	0.10	263×100	263×200
ADA, Noisy	3536	4948	5.02	21291×5016	21291×10032
No Overlap, Noisy	27	64	0.35	76×345	76×690
Displacement	$\mathbf{L}_{\text{retained}}$	$\mathbf{L}_{\text{max}}$	$\mathbf{T_w}$[s]	$\mathbf{n \times r}$	$\mathbf{n \times p}$
ADA	85	85	0.10	263×100	263×200
No Overlap	92	100	0.10	263×100	263×200
ADA, Noisy	2575	4770	4.77	21537×4770	21537×9540
No Overlap, Noisy	10	24	1.08	24×1083	24×2166

5.7 Non-minimum Phase Systems

It was highlighted in Section 1.2 that in order to invert a non-minimum phase system, the inversion would need to occur non-causally. In this section we will demonstrate that ADA is capable of handling non-minimum phase systems. In Section 2.3 it was shown that the spring displacement sensor configuration contained an unstable zero:

$$\text{zeros}_{\Delta_z} = \begin{bmatrix} 0.2146 \times 10^{-6} \\ -0.2146 \times 10^{-6} \end{bmatrix}. \tag{5.26}$$

Therefore, we would expect this sensor configuration would need to be inverted non-causally. To demonstrate this we will compare two different sensor configurations: the aforementioned spring displacement set-up and a sprung mass only displacement set-up. The latter was chosen

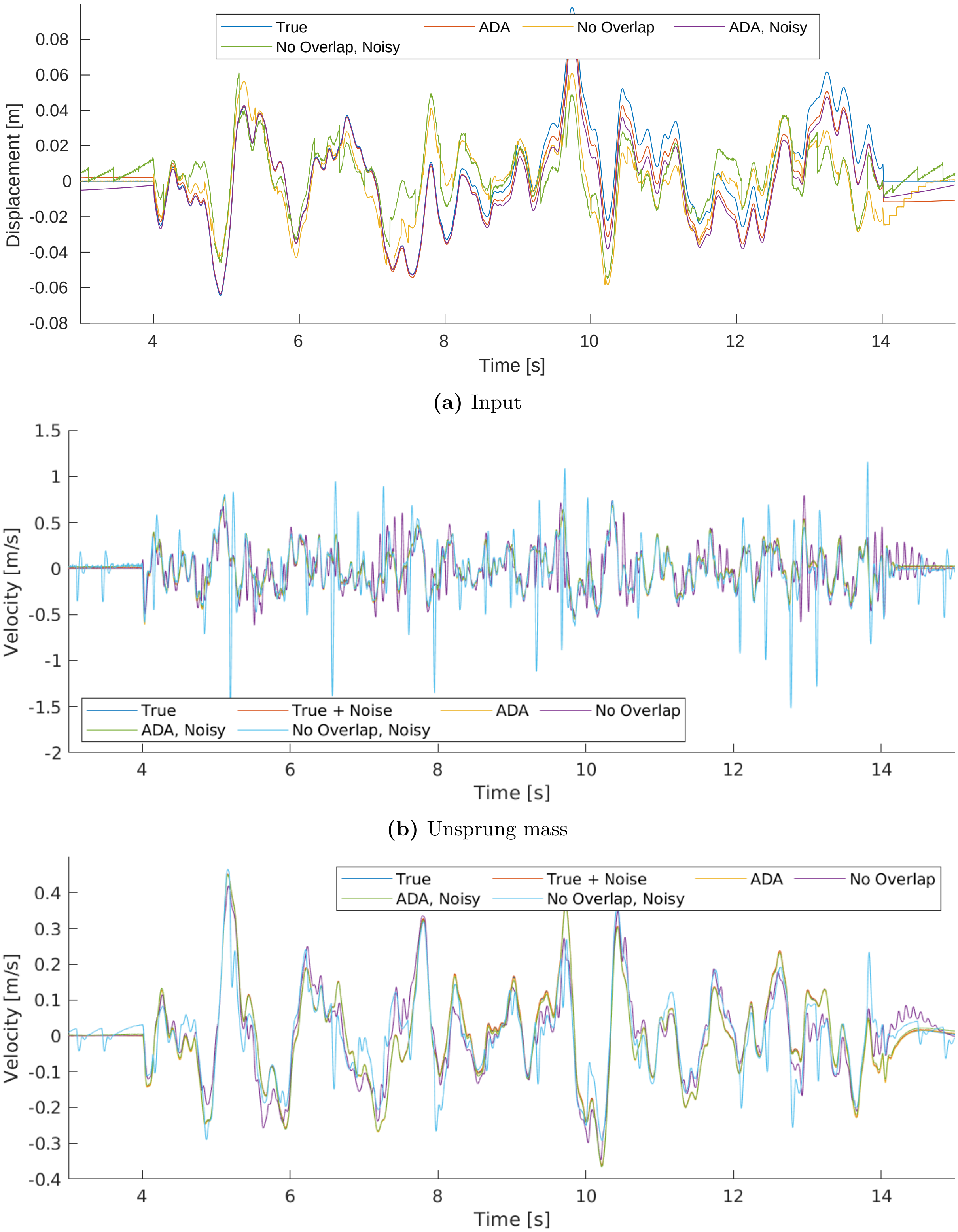


(a) Input

(b) Unprung mass

(c) Sprung mass

Figure 5.19: Response reconstructions for the case of raw acceleration sensor values.

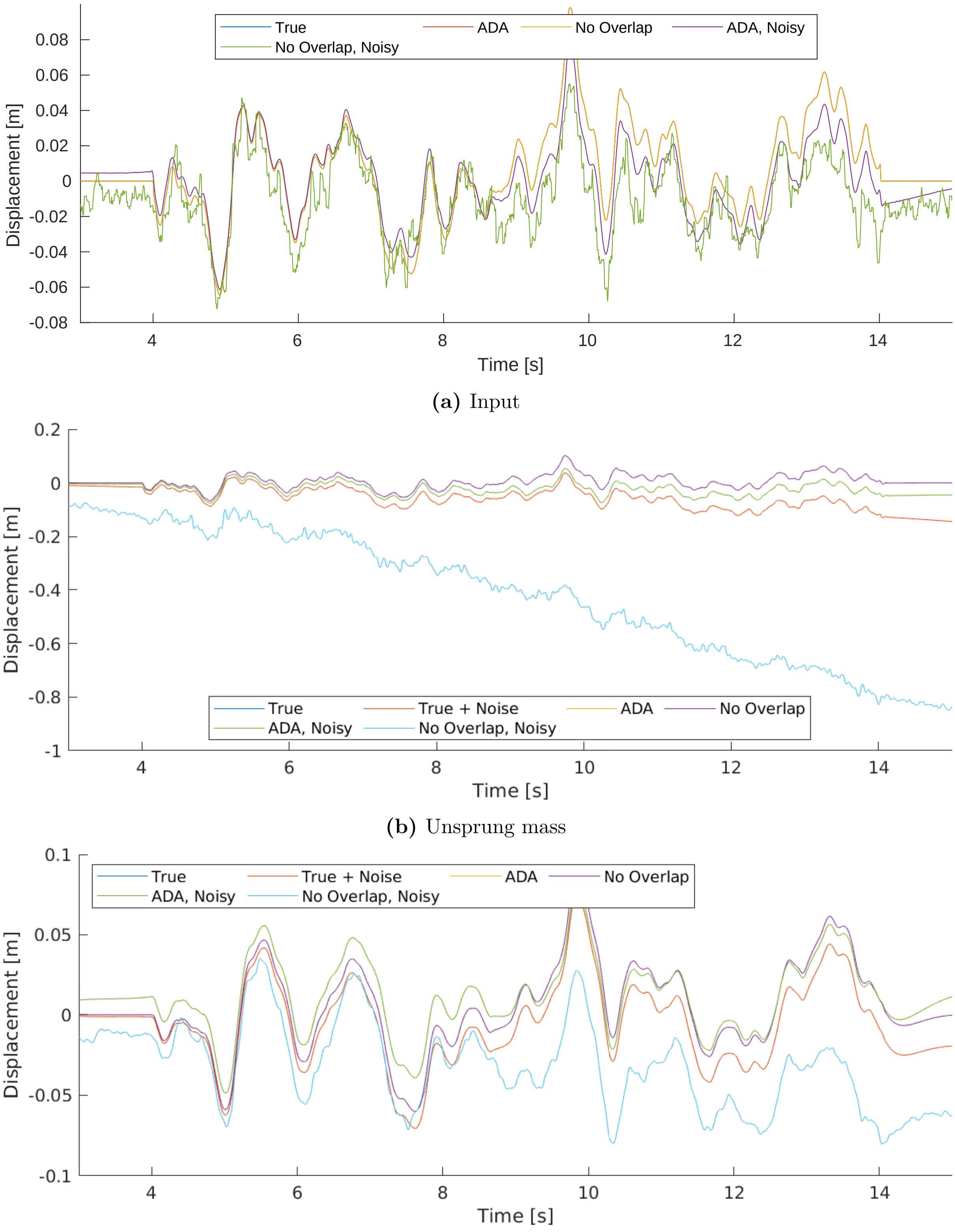


(a) Input

(b) Unsprung mass

(c) Sprung mass

Figure 5.20: Response reconstructions for the case of velocity sensor values. Velocity values were computed by numerically integrating acceleration readings.



(a) Input

(b) Unsprung mass

(c) Sprung mass

Figure 5.21: Response reconstructions for the case of displacement sensor values. Displacement values were computed by numerically integrating acceleration readings twice.

since we know that it has a singular well-behaved zero sitting in the left hand plane as shown in Section 2.3:

$$\text{zeros}_{z_A} = -3.2. \tag{5.27}$$

We would thus expect this configuration to be able to be inverted causally. We will invert these systems using the following windowing methods: *No Overlap*, *First*, *Middle*, *Last*, ADA. The *First* windowing is the completely acausal case since its predictions are based on all the values ahead of it in the same window period. The *Last* is the purely causal case since its predictions are based on all the predictor values preceding it for a given window period. ADA and *Middle* sit somewhere inbetween these two in terms of causality. *No Overlap*, the windowing method with no overlap is included for completeness. A summary of the experimental design is given in Table 5.20.

Table 5.20: Experimental design for testing non-minimum phase inversion. (Variables of interest shown first).

Variable	**Details**
Sensor Configuration	Spring displacement only, $\mathbf{\Delta_z}$, Sprung mass displacement only, $\mathbf{z_A}$
Windowing method	*No Overlap*, *First*, *Middle*, *Last*, ADA
Window lengths $\mathbf{T_w}$	$\in [0.1\ \text{s}, 6\ \text{s}]$ with 25 equally spaced divisions.
Window Stride $\mathbf{s_\tau}$	1 for all windowing methods except for *No Overlap*
Number of retained latent variables L	$\in [1, L_{\text{max}}]$ with $\min(40, \max(10, \lceil L_{\text{max}}/10 \rceil))$ equally spaced divisions
Sampling frequency $\mathbf{f_s}$	1000 Hz
Regression method	SBTR
QC parameters	Default values; Table 2.1,no nonlinearity term k_{NL}
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

5.7.1 Results

The response reconstruction results for the spring displacement sensor configuration are shown in Table 5.21. Here we note that ADA performs the best in terms of the recreated output with *Middle* performing second best. If we compare *First* and *Last* we note that they perform similarly with *First* performing slightly better. The recreated test responses for the different windows are shown in Figure 5.22. The optimised hyper-parameters for the spring displacement case are shown in Table 5.22. We note that *Middle* is heavily regularised.

The response reconstruction results for the sprung mass displacement sensor configuration are shown in Table 5.23. Here we note that *Middle* performs the best in terms of the recreated output with *Last* coming in a close second, suggesting that a causal inversion is better suited to this problem. Here we note that *First* has performed the worst. The recreated test responses for the different windows are shown in Figure 5.23. The optimised hyper-parameters for the sprung mass displacement case are shown in Table 5.24.

Comparing the non-minimum to the minimum phase cases it appears that *Middle* seems to be the better choice in general but with ADA performing better in the non-minimum phase

Table 5.21: MFFE [%] results for the approximated input and output signals for the numerical investigation into non-minimum phase systems. Spring displacement case (non-minimum phase system). The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

	Training		**Validation**		**Test**	
Window Method	$\mathbf{u}_{\text{road}}$	$\mathbf{\Delta z}$	$\mathbf{u}_{\text{road}}$	$\mathbf{\Delta z}$	$\mathbf{u}_{\text{road}}$	$\mathbf{\Delta z}$
ADA	8.46	0.51	21.04	0.46	31.83	2.05
First	17.51	7.98	32.44	7.03	43.86	11.20
Middle	6.56	2.02	13.58	1.96	22.49	3.93
Last	21.20	9.39	25.38	8.53	47.82	17.16
No Overlap	32.26	19.38	57.79	26.17	96.03	39.26

Table 5.22: Optimized hyper-parameter results and corresponding training matrix sizes for the numerical investigation into non-minimum phase systems. Spring displacement case (non-minimum phase system).

Window Method	$\mathbf{L}_{\text{retained}}$	$\mathbf{L}_{\text{max}}$	$\mathbf{T}_{\mathbf{w}}[\text{s}]$	$\mathbf{n \times r}$	$\mathbf{n \times p}$
ADA	5342	5753	5.75	20553×5754	20553×5754
First	5580	5999	6.00	20307×6000	20307×6000
Middle	896	6000	6.00	20307×6000	20307×6000
Last	5596	5999	6.00	20307×6000	20307×6000
No Overlap	18	39	0.59	44×591	44×591

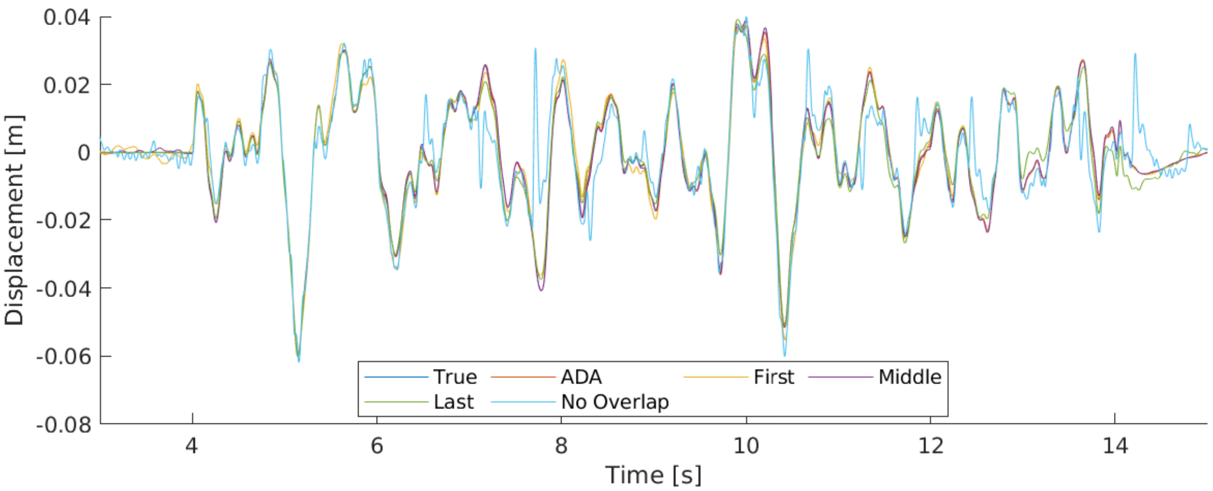


Figure 5.22: Reconstructed test response or the numerical investigation into non-minimum phase systems. Spring displacement case (non-minimum phase system).

configuration. This makes sense since both methods sit between the two extremes acausal or fully causal inversion. There are obviously poor choices for windowing these systems with *Last* performing worse on the non-minimum phase system and *First* performing poorly on the minimum phase system. So in general ADA or *Middle* would be the preferred choices.

Table 5.23: MFFE results for the approximated input and output signals for the numerical investigation into non-minimum phase systems. Spring displacement case (minimum phase system).

	Training		**Validation**		**Test**	
Window Method	$\mathbf{u}_{\text{road}}$	$\mathbf{z}_{\mathbf{A}}$	$\mathbf{u}_{\text{road}}$	$\mathbf{z}_{\mathbf{A}}$	$\mathbf{u}_{\text{road}}$	$\mathbf{z}_{\mathbf{A}}$
ADA	0.39	0.47	0.64	0.84	0.90	0.95
First	16.92	14.19	17.63	15.36	17.92	20.30
Middle	0.12	0.08	0.16	0.16	0.09	0.11
Last	0.07	0.07	0.16	0.20	0.10	0.12
No Overlap	0.03	0.04	15.77	4.45	17.30	9.09

Table 5.24: Optimized hyper-parameter results and corresponding training matrix sizes for the numerical investigation into non-minimum phase systems. Sprung mass displacement case (minimum phase system).

Window Method	$\mathbf{L}_{\text{retained}}$	$\mathbf{L}_{\text{max}}$	$\mathbf{T}_{\mathbf{w}}$**[s]**	$\mathbf{n} \times \mathbf{r}$	$\mathbf{n} \times \mathbf{p}$
ADA	6000	6000	6.00	20307×6000	20307×6000
First	22	345	0.35	25962×345	25962×345
Middle	2746	5975	6.00	20307×6000	20307×6000
Last	5048	5507	5.51	20799×5508	20799×5508
No Overlap	24	31	0.84	31×837	31×837

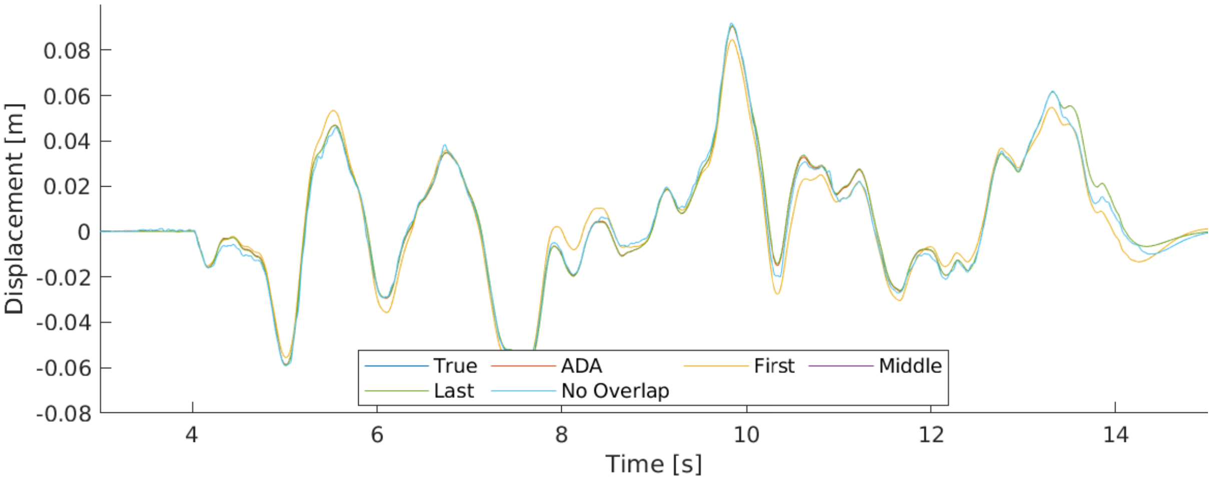


Figure 5.23: Reconstructed test response for the numerical investigation into non-minimum phase systems. Sprung mass displacement case (minimum phase system).

5.8 Conclusion

Through the course of this chapter it has been highlighted that non-overlapping windows can only invert systems where only differentiation is needed to approximate the internal states of the system. It was initially believed that the issue lay in simply extracting parts of the window that were expected to be more correct. This is shown to be true to some extent in the non-minimum phase system investigation (Section 5.7) whereby causality does play a role in the response reconstruction accuracy. However, the main underlying issue is the ability to approximate integration. This is shown to be the case in Section 5.6 whereby it was demonstrated that ADA is a generalised black-box model. This means that it can invert systems where the unobserved states of the system need to be approximated with integration or differentiation. ADA produces the most accurate response reconstructions over a wider range of problems as compared to the other overlapping window methods. It was demonstrated that ADA produces these results by adding another form of regularisation by imposing a form on the shape of the linear regression coefficients. The introduction of overlapping windows greatly increased the computational complexity of the problem as demonstrated in Section 5.4. However, methods such as varying the stride were shown to mitigate this computational overhead.

Chapter 6

Choosing Hyper-parameters A Priori

We stated that the one of the goals of this dissertation is to develop a black-box methodology of creating an inverse model of the system. The number of hyper-parameters that will need to be determined via cross-validation would be expensive both in terms of computation and in terms of unnecessary physical testing of the rig. The physical testing is required since we optimise for the recreated output accuracy of the system. Therefore each cross validation iteration requires a physical experimental trial. If we relax the requirement of having no prior knowledge of the system we could perhaps reduce the number of hyper-parameters by choosing them based on easily determined physical properties of the rig. In this chapter we will look at means of choosing the sampling frequency and window length based on this assumption.

6.1 Sampling Frequency

Simply choosing a high sampling frequency, as was done in the previous numerical experiments, may eliminate the need to search for an optimal sampling frequency but choosing too high a sampling frequency produces significant computational cost and memory requirement. Two methods will be investigated in order to approximate the required sampling frequency using an initial test of the test rig. The first method relates the sampling frequency to the fastest time constants $T_{c,\min}$ of the system through the means of a step input test. The second method uses correlation tests on the outputs of the system to determine a reasonable sampling frequency.

For the step input test the system is given a displacement step input. The displacements of the sprung and unsprung masses are then recorded. For the sake of this investigation we are going to assume that we have access to the absolute displacements of the masses. The time constant T_c for each mass is then defined as the time the system takes to first reach 63% of the final displacement value. In order to properly resolve the fastest time constant we would need to sample at least at the Nyquist rate, which is twice the frequency of interest, i.e. $f_{\mathrm{s}} = 2f_{\mathrm{c}} = 2\frac{1}{T_{\mathrm{c}}}$ [32].

For the autocorrelation tests, the desired sampling frequencies are defined by the first minimum of the autocorrelation of the outputs $\phi(z,z)$, where the correlation function $\phi(x,y)$ is given as

$$\phi(x,y) = \frac{\Sigma_{k=1}^{N-\tau}[x(k)-\mu_x][y(k+\tau)-\mu_y]}{\sqrt{\Sigma_{k=1}^{N}[x(k)-\mu_x]^2}\sqrt{\Sigma_{k=1}^{N}[y(k)-\mu_y]^2}}. \tag{6.1}$$

The autocorrelation of each sensor is then taken and the fastest minimum lag is then used to determine the frequency of interest. In the case of a linear system it is known that the system can only produce frequencies in the outputs that are present in the input signal. However, this is not the case for a nonlinear system where it is possible to produce frequencies that are much higher than the input content. Therefore, a test that takes this nonlinear relationship into consideration is required. A proposed method is to look at the autocorrelation of the outputs squared i.e. $\phi(z^2, z^2)$ [4]. In the next section the step input and the linear correlation design will be tested on a linear quarter car set-up. The test will then be performed on a nonlinear quarter car model using the previous methods as well as the nonlinear autocorrelation method. The purpose of the tests is to determine the suitability of this method as a rule of thumb when deciding upon a sampling frequency when using ADA combined with SBTR.

6.1.1 Procedure

In order to test the performance of ADA with regards to different sampling frequencies, the signals will need to be downsampled before performing regression and prediction on them. To achieve this, all training signals will be passed through the quarter car model and measured at 1000 Hz. The output and input signals will then be downsampled using MATLAB's `resample` function. The `resample` function low-pass filters the downsampled signal to avoid anti-aliasing issues. The regression and prediction steps will then be performed with the downsampled set of signals. Once a downsampled input approximation is computed it is then upsampled to 1000 Hz using the `resample` function. The upsampled approximate input is then passed through the quarter car model. The corresponding outputs are then compared against the original signal sampled at 1000 Hz to obtain a final MFFE result. This will then be repeated for a number of sampling frequencies. To reduce computational overhead, the window length will be fixed to 6 s.

The effects of the downsampling will also be tested at different unsprung mass stiffness k_{R} configurations. This is done in order to change the time constant of the system. The procedure is then repeated for the nonlinear case. The autocorrelation tests will be performed using the APRBS plotted in Figure 5.1a as the input.

An arbitrary benchmark at which the sampling frequency achieves an MFFE of 1% will be chosen in order to compare the frequencies suggested by the step and autocorrelation methods. In order to create a rule of thumb for potential initial guesses for the sampling frequency a ratio, r_f, will be calculated that denotes the ratio between the frequency suggested by either the step method or the correlations method and the corresponding point at which the spring displacement validation reconstruction reaches 1% MFFE. The ratio r_f is then rounded to the nearest integer, i.e.

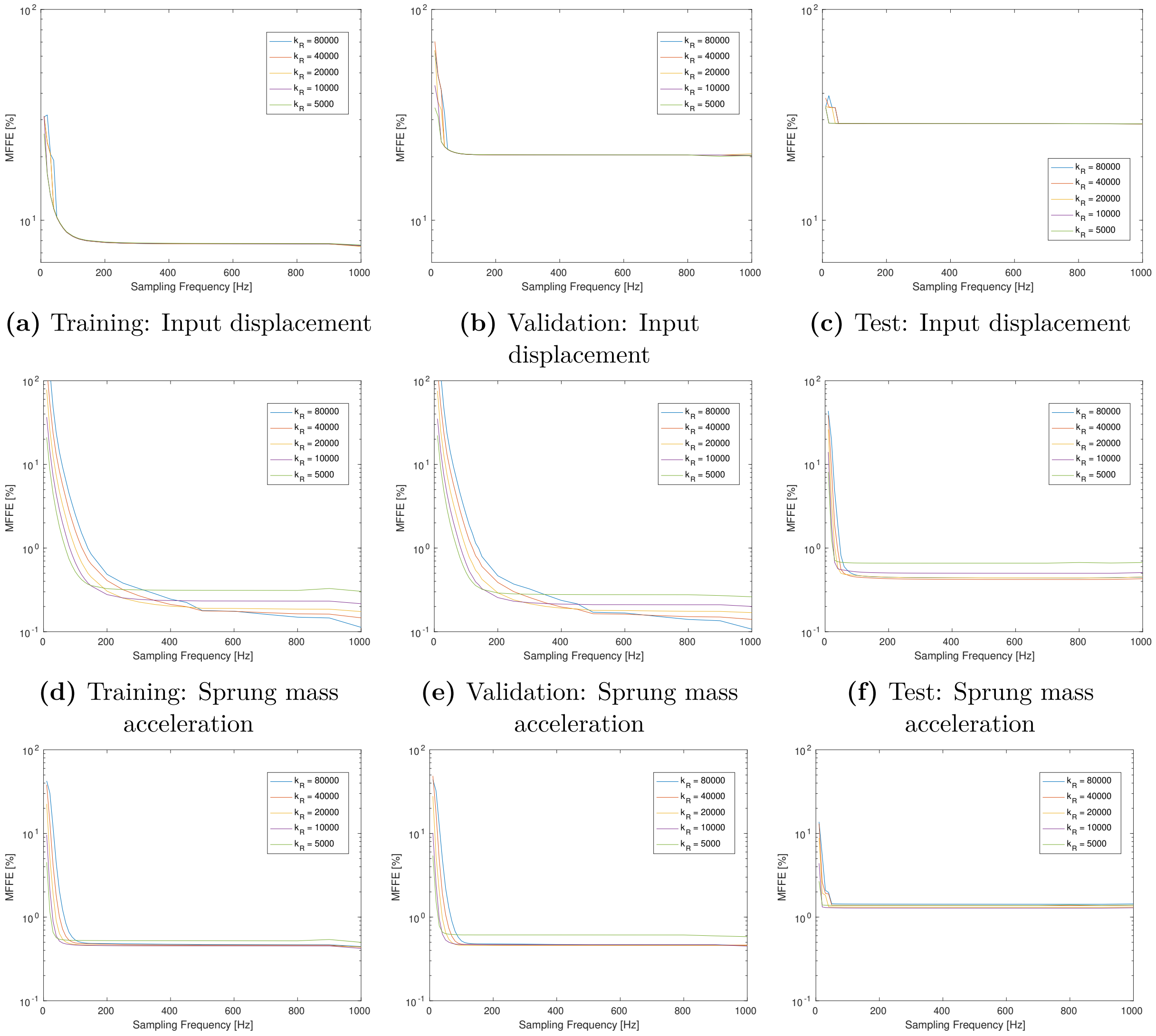
$$r_f = \left\lfloor \frac{f_{1\%}}{f_{\mathrm{method}}} \right\rceil . \tag{6.2}$$

An overview of the numerical experiment variables is given in Table 6.1.

6.1.2 Results

The linear system reconstruction results for the input and output channels as functions of sampling frequency f_{s} are shown in Figures 6.1. The linear system MFFE results for the outputs show that the error quickly drops off initially. The error then proceeds to decrease slowly as the sampling rate increases. This suggests that a trade-off is needed. We note in general that increasing the stiffness of the system requires a higher sampling frequency to achieve the same MFFE on the outputs. This trend only holds true until the response accuracies start plateauing, as can be seen in Figure 6.1e.

Visually, the validation results suggest that sampling above 175 Hz is wasteful in terms of the spring displacement reconstruction accuracy for all of the k_{R} configurations for the linear case. This value will be used for any linear numerical experiment going forward in the dissertation for convenience. Fixing this value reduces the computational expense as compared to the 1000 Hz used previously without having to optimise for the sampling frequency.



(a) Training: Input displacement

(b) Validation: Input displacement

(c) Test: Input displacement

(d) Training: Sprung mass acceleration

(e) Validation: Sprung mass acceleration

(f) Test: Sprung mass acceleration

(g) Training: Spring displacement

(h) Validation: Spring displacement

(i) Test: Spring displacement

Figure 6.1: Input and output reconstruction MFFE results as a function of sampling frequency. The stiffness value k_R was varied to change the time constants of the system. Performed on the linear quarter car model.

Table 6.1: Experimental variables for the investigation into the effects of sampling frequency on response reconstruction. (Variables of interest shown first).

Variable	Details
Sampling Frequency $\mathbf{f_s}$	$\in [50, 1000]$ Hz with 15 equally spaced steps
QC parameters	Default values; Table 2.1 with $k_R \in [5, 10, 20, 40, 80] \times 10^3$ N m
$\mathbf{k_{NL}}$	Linear case: 0; Nonlinear case: 1.28×10^7 N m^{-3}
Windowing method	ADA
Regression method	SBTR
Sensor Configuration	Sprung mass acceleration + spring displacement
Window length $\mathbf{T_w}$	6 s
Window proportional overlap γ	Maximum
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced steps
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

The nonlinear system reconstruction results are shown in Figure 6.2. The MFFE results are noisier than those achieved by the linear system with less noticeable plateauing of the MFFE values. However there it is still a case of diminishing returns. For convenience sake, 350 Hz will be used for any nonlinear numerical investigations going forward.

Step Test

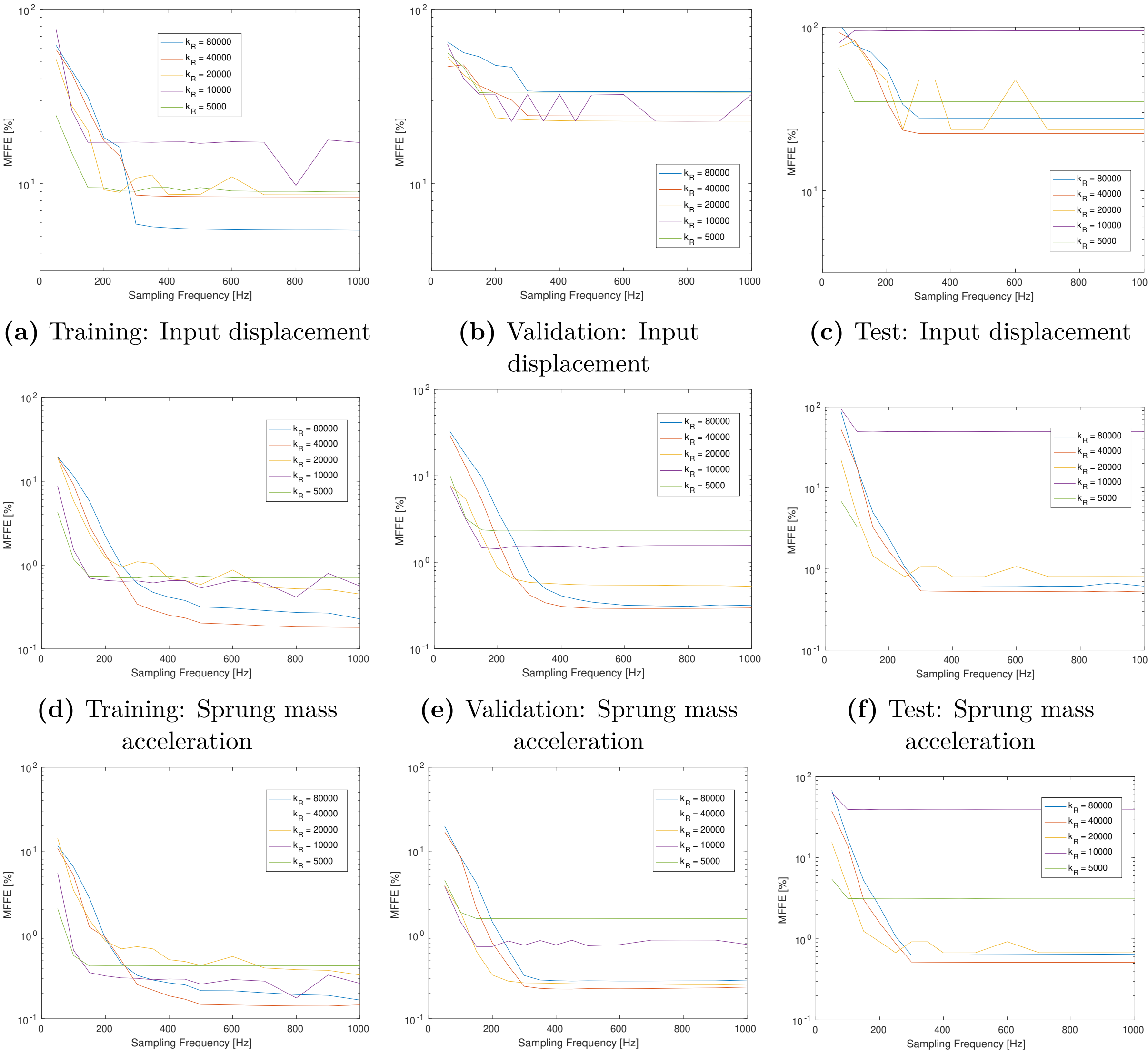
The corresponding step input results for tyre stiffness values $k_R = 80000$ N m^{-1} are plotted in Figures 6.3. The frequency suggestions, f_{step} , as provided by the step test methods are compared in Table 6.2. By using the step input method, it seems that using 2 to 3 times the fastest frequency suggestion, in this case the value associated with the unsprung mass z_{R}, provides an adequate sampling rate for the linear case. The r_f ratio range widens to between 5 and 14 when switching to the nonlinear case.

Linear Autocorrelation Test

The autocorrelation results of the outputs, for tyre stiffness values $k_R = 80000$ N m^{-1}, are plotted in Figures 6.4. The frequency suggestions, $f_{\phi(z,z)}$, as provided by the linear correlation test are compared in Table 6.2. The linear autocorrelation methods suggests a frequency ratio r_f range of 3 to 7 times the proposed frequency for the linear case. The range shifts considerably to between 35 to 40 for the nonlinear case.

Nonlinear Autocorrelation Test

The autocorrelation results of the outputs, for tyre stiffness values $k_R = 80000$ N m^{-1}, are plotted in Figure 6.5 for the linear and nonlinear case respectively. The frequency suggestions,$f_{\phi(z^2,z^2)}$, as provided by the linear correlation test are compared in Table 6.2. The frequency ratio values, r_f, provided by the nonlinear autocorrelation test suggest that a range between 2 and 4 would



(a) Training: Input displacement

(b) Validation: Input displacement

(c) Test: Input displacement

(d) Training: Sprung mass acceleration

(e) Validation: Sprung mass acceleration

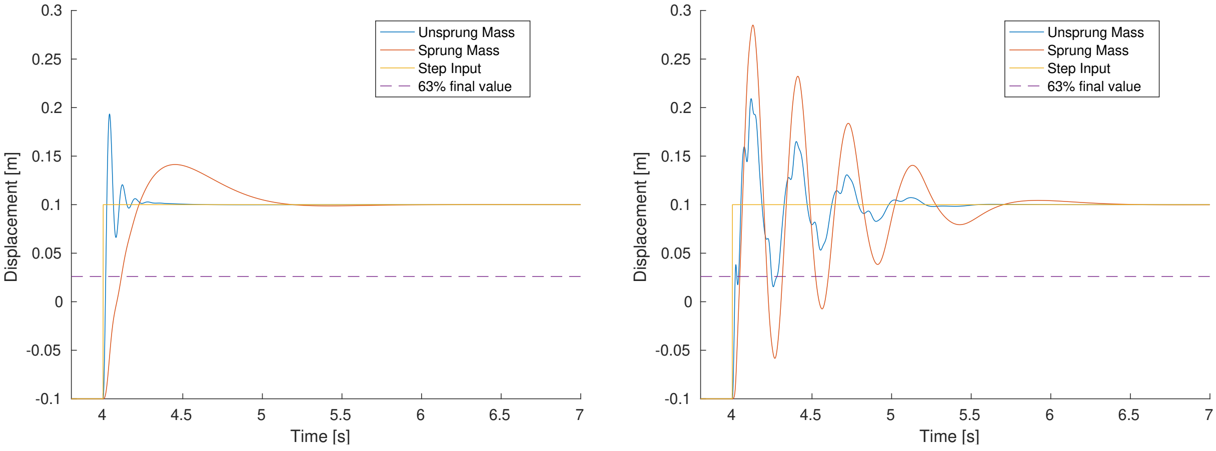
(f) Test: Sprung mass acceleration

(g) Training: Spring displacement

(h) Validation: Spring displacement

(i) Test: Spring displacement

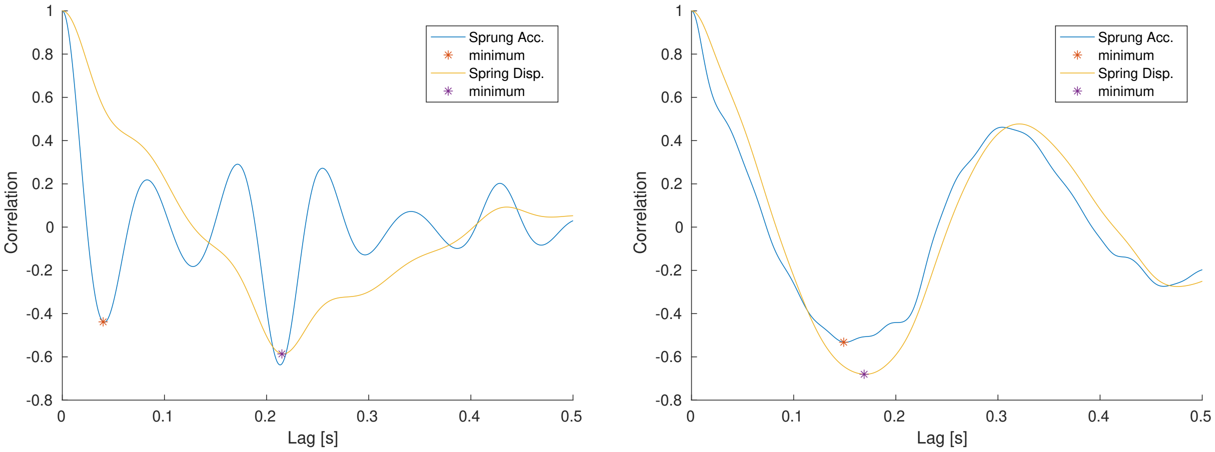
Figure 6.2: Input and output reconstruction MFFE results as a function of sampling frequency. The stiffness value k_R was varied to change the time constants of the system. Performed on the nonlinear quarter car model.



(a) Linear System

(b) Nonlinear System

Figure 6.3: Step responses for the quarter car model with unsprung stiffness $k_R = 80000$. The step response is used to approximate the time constant of the system by determining the time taken to reach 63% of the final value.

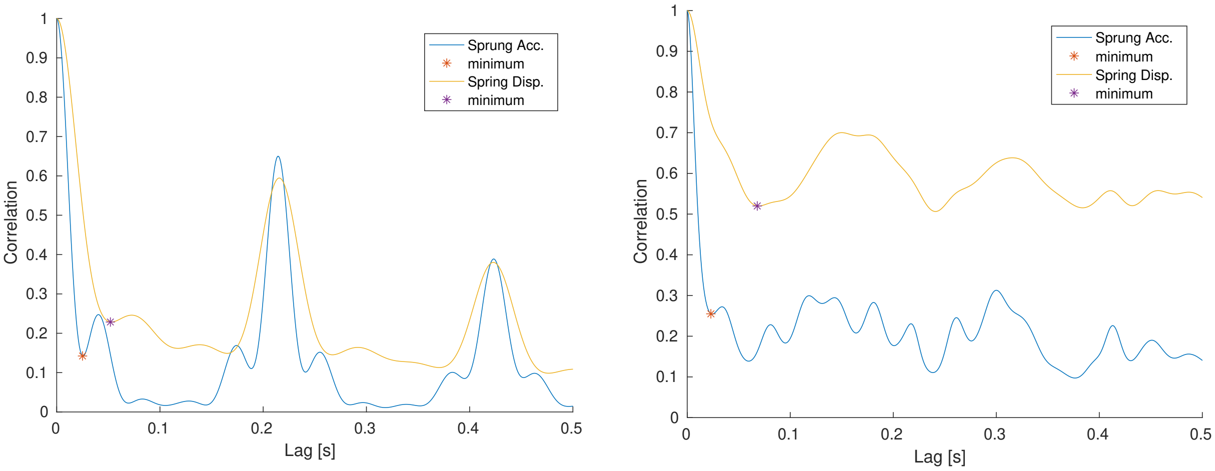


(a) Linear quarter car

(b) Nonlinear quarter car

Figure 6.4: Linear autocorrelation values of the outputs for the linear quarter car with unsprung mass stiffness $k_R = 80000\ \mathrm{N\,m^{-1}}$.

work for the linear case. The range widens to between 4 and 20 for the nonlinear system, however, a value of 10 times the suggested sampling frequency would work for most of the linear and nonlinear configurations.



(a) Linear quarter

(b) Nonlinear quarter car

Figure 6.5: Nonlinear autocorrelation values of the outputs for the quarter car with unsprung mass stiffness $k_R = 80000$.

Table 6.2: Results comparing suggested sampling frequencies suggested by the step test, f_{step} , linear autocorrelation test, $f_{\phi(z,z)}$ and nonlinear autocorrelation test, $f_{\phi(z^2,z^2)}$. r_f denotes how many times faster the sampling needs to occur to achieve 1% MFFE, $f_{1\%}$, for the spring displacement reconstruction.

k_{R}	f_{step} [Hz]				$f_{\phi(z,z)}$ [Hz]				$f_{\phi(z^2,z^2)}$ [Hz]				$f_{1\%}$ [Hz]
Linear Case													
	z_{R}	r_f	z_{A}	r_f	$\ddot{z}_{\text{A}}$	r_f	Δz	r_f	$\ddot{z}_{\text{A}}$	r_f	Δz	r_f	Δz
80000	62.50	2	8.70	8	25.00	3	4.65	15	40.00	2	19.23	4	67.39
40000	41.67	2	8.77	6	16.13	4	4.59	12	26.32	2	7.14	8	52.28
20000	27.03	2	8.06	5	11.24	4	4.46	9	16.95	3	8.33	5	39.65
10000	16.95	2	6.62	5	4.69	7	4.15	8	9.17	4	7.35	5	29.77
5000	10.42	3	5.18	6	4.44	7	3.47	8	8.13	4	6.13	5	27.20
Nonlinear Case													
	z_{R}	r_f	z_{A}	r_f	$\ddot{z}_{\text{A}}$	r_f	Δz	r_f	$\ddot{z}_{\text{A}}$	r_f	Δz	r_f	Δz
80000	55.56	5	21.28	11	6.71	35	5.92	39	43.48	6	14.71	16	228.40
40000	19.23	10	15.62	13	5.95	33	5.81	34	50.00	4	12.05	16	192.17
20000	13.89	10	11.36	12	4.57	30	4.44	31	40.00	4	9.01	16	135.42
10000	10.00	14	8.26	16	3.33	40	3.32	40	6.80	20	6.67	20	130.66
5000	6.99	-	6.02	-	2.60	-	2.47	-	5.43	-	5.38	-	-

6.1.3 Conclusion

By increasing the stiffness of the numerical quarter car we were able to influence the sampling frequency requirement of the response reconstruction algorithm. The linear case demonstrated the general trend of a stiffer system requiring a higher sampling frequency. The methods outlined in this section can be used to help guide the selection of a sampling frequency. However, the results achieved are at best a rule of thumb that may help identify an initial guess for the appropriate sampling frequency. The results obtained using the nonlinear autocorrelation provided a reasonable range for an initial guess both for the linear and nonlinear case. It is suggested that an initial guess of 10 times the frequency suggested by the nonlinear autocorrelation test is used. The initial estimate can be relaxed if the computational costs are too high or increased if the reconstruction results are unsatisfactory.

6.2 Window Length

In the previous chapters the search for a suitable window length was determined by a grid search cross validation over a large range of window length values T_{w}. Searching over a large range of potential window lengths using the cross validation method results in multiple experiments on the physical test rig which leads to unnecessary fatiguing of the system. Therefore, a methodology is needed that can narrow down the search for a suitable window length without requiring multiple experiments. To this end two different approaches will be used to determine an estimate of the window length T_{w}.

The first proposed method is to perform a step input test of the system and to use the absolute displacements of the masses to determine the settling time of the system. The settling time of the system is defined as the time taken to approach 95 % or 99 % of the final value, denoted as $t_{\mathrm{step},95\%}$ and $t_{\mathrm{step},99\%}$ respectively.

The second proposed method is to look at the correlation function between the inputs u and the outputs z of the system to determine the delay of the system. If we use the correlation function defined in Equation (6.1) we can then use the 95 % confidence interval for the $\phi(u, z)$ value to determine a suitable cut-off point for the window length T_w. An approximation for the 95 % confidence interval for a given correlation function is given as $\pm 1.96/\sqrt{N}$, where N is the number of sample points [5].

In the following section the proposed methods will be tested on the numerical quarter car model.

6.2.1 Procedure

In order to change the settling time of the test rig the damping constant b_{A} of the system will be varied over a range of values: $b_{\mathrm{A}} \in [125, 250, 500, 1000, 2000]\ \mathrm{N\,s\,m^{-1}}$. To determine whether the methodologies will be suitable for linear as well as nonlinear systems, the quarter car will be tested with k_{NL} set to 0 and 1.28×10^7 for the linear and nonlinear case. The sampling frequency of the system will be set to 350 Hz and 400 Hz for the linear and nonlinear case respectively. The window length at which spring displacement reconstruction MFFE reaches 1 % for the validation set will be chosen as a benchmark and is denoted as $t_{1\%}$. The correlation between the inputs and outputs will be performed using the APRBS plotted in Figure 5.1a as the input. An overview of the numerical experiment variables is given in Table 6.3.

Similarly to choosing a sampling frequency, a rule of thumb for potential initial guesses for the window length T_{w} will be determined using a ratio, r_t. The value r_t denotes the ratio between the time suggested by either the step method or the correlation method and the corresponding point at which the spring displacement validation reconstruction reaches 1% MFFE. The ratio r_t is then rounded to the nearest integer, i.e.

$$r_t = \left\lceil \frac{t_{1\%}}{t_{\text{method}}} \right\rceil . \tag{6.3}$$

The sensor with the longest suggested window length will be used to determine the window length ratio.

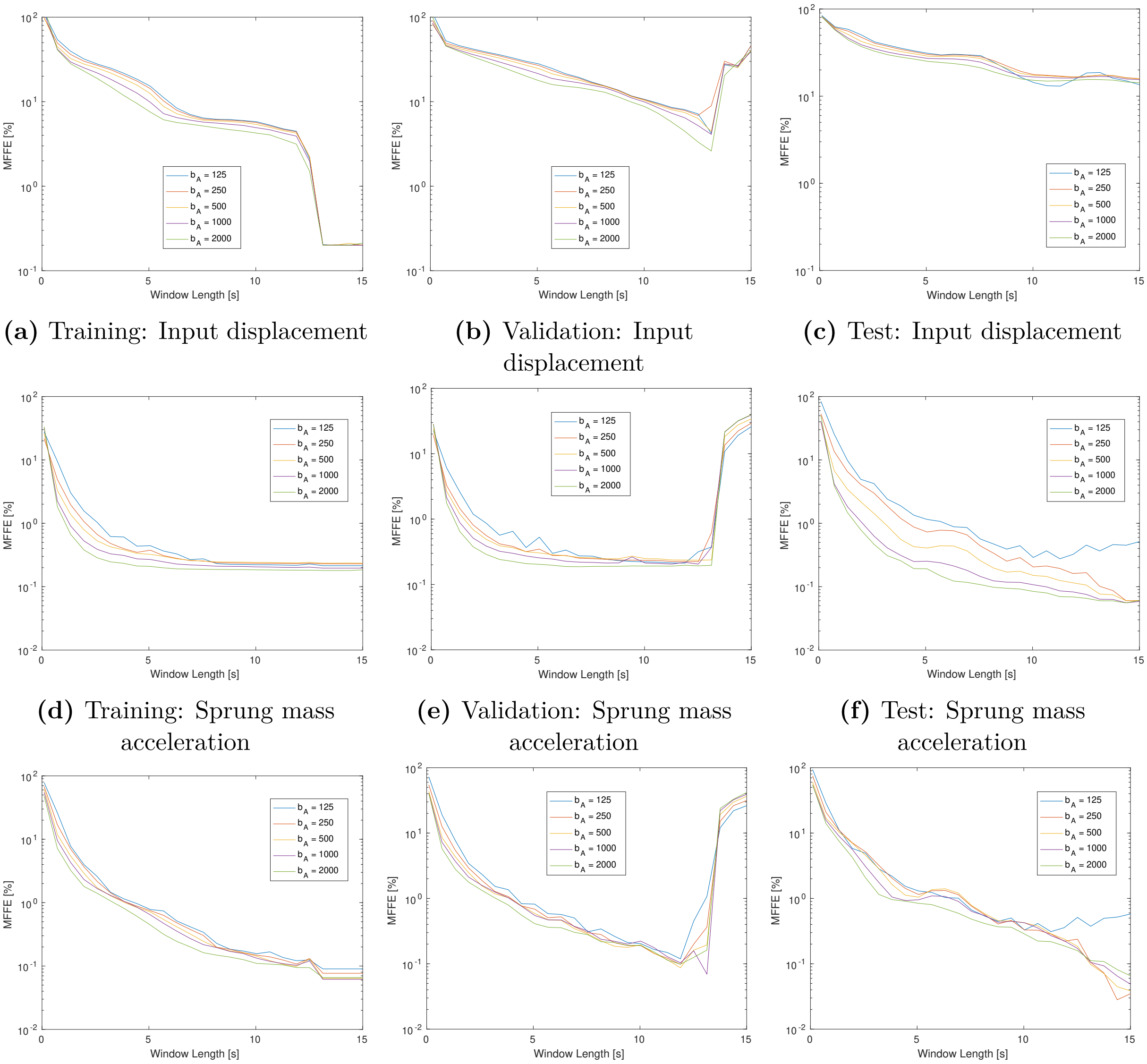
Table 6.3: Experimental variables for the investigation into the effects of window length on response reconstruction. (Variables of interest shown first).

Variable	**Details**
Window lengths $\mathbf{T_w}$	Linear: $\in$ [0.1, 15] s with 30 equally spaced intervals. Nonlinear: $\in$ [0.1, 10] s with 20 equally spaced intervals.
QC parameters	Default values; Table 2.1 with $b_{\text{A}} \in [125, 250, 500, 1000, 2000]$ N s m^{-1}
$\mathbf{k_{NL}}$	Linear case: 0; Nonlinear case: 1.28×10^7 N m^{-3}
Sensor Configuration	Sprung mass acceleration + spring displacement for correlation tests. Absolute Displacements on both DOF for step tests.
Regression method	SBTR
Window method	ADA
Window proportional overlap γ	Maximum
Number of retained latent variables L	$\in [1, L_{\text{max}}]$ with $\min(40, \max(10, \lceil L_{\text{max}}/10 \rceil))$ equally spaced steps
Sampling Frequency $\mathbf{f_s}$	$\in [50, 1000]$ Hz with 15 equally spaced steps
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

6.2.2 Results

The linear MFFE results for the validation and test set for the different damping coefficients are plotted in Figure 6.6. In the linear case the algorithm experiences overfitting once the window length exceeds a threshold. This is indicated by the training errors either staying constant or decreasing while the corresponding validation errors and test errors increase. The test results seem to be unaffected by the overfitting whereas the validation results experience a sudden jump in error results. This is possibly due to the fact that the test inputs use the road profile which do not contain the higher frequency components found in the APRBS. The threshold occurs at $T_w = 13.15$ s which is half the length of the original training input signal. This is the point where the maximum potential rank is achieved. This is highlighted in Figure 6.7. The number of observations n is equal to the dimension q for the matrix Y at this point. Therefore the maximum number of latent variables can potentially be extracted which allows the system to readily overfit. We also note the general observation that as the damping increases the required window length for a given output MFFE decreases. This is most evident in Figure 6.6f.

The nonlinear case results are plotted in Figure 6.8. The nonlinear MFFE results show a much clearer minimum as a function of window length. In the case of $b_{\text{A}} = 250$ we note poor performance in the testing phase of response reconstruction. This dramatic effect is possibly due



(a) Training: Input displacement

(b) Validation: Input displacement

(c) Test: Input displacement

(d) Training: Sprung mass acceleration

(e) Validation: Sprung mass acceleration

(f) Test: Sprung mass acceleration

(g) Training: Spring displacement

(h) Validation: Spring displacement

(i) Test: Spring displacement

Figure 6.6: Input and output reconstruction MFFE results as a function of window length. The damping constant b_A was varied to change the settling times of the system. Performed on the linear quarter car model.

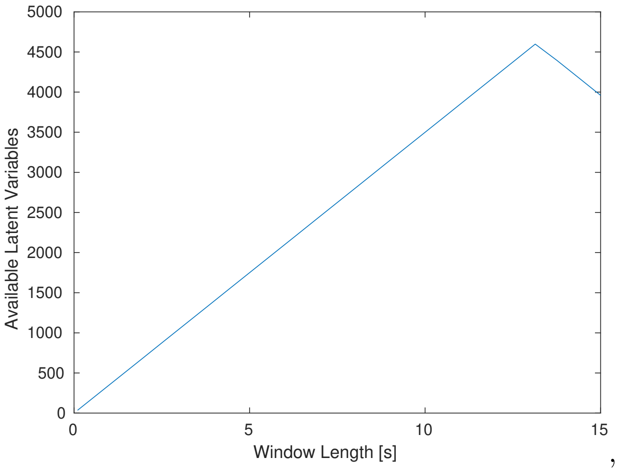


Figure 6.7: Number of latent variables available plotted against the window length. At half the length of the original training input the number of latent variables reach a maximum.

to the significant ill-posedness of the nonlinear quarter car model as discussed in Section 2.2.

Settling Time

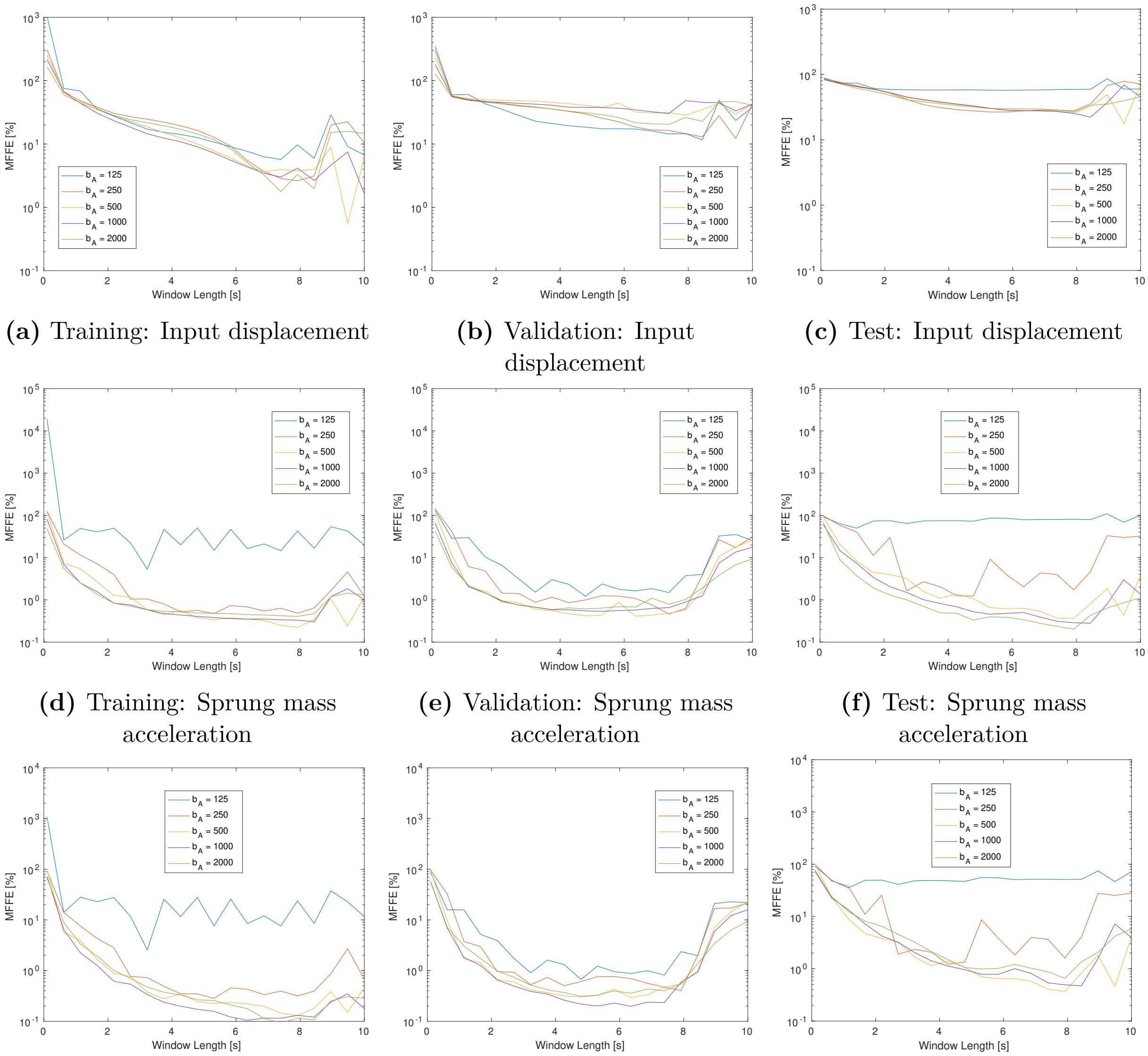
A sample step for the nonlinear system with damping constant $b_A = 500\ \mathrm{N\,s\,m^{-1}}$ is shown in Figure 6.9. The results of the step test are shown in Table 6.4. If we look at the slowest settling times for the 95% mark we note that they occur for sprung mass displacements, z_A. Using the slowest settling times suggests a ratio r_t from anywhere between 4 and 19 for the linear case and between 1 and 7 for the nonlinear case. If we look at the slowest settling time, this time setting the threshold at 99% suggests using a ratio between 2 and 4 for the linear case and between 1 and 6 for the nonlinear case. This is a much tighter range in the linear case. Therefore, the settling time may be a good indicator for window length for a linear system.

Correlation Test

A sample correlation test for the nonlinear system with damping constant $b_A = 500\ \mathrm{N\,s\,m^{-1}}$ is shown in Figure 6.10. The results of the correlation test are shown in Table 6.4. In the case of the correlation test the suggested values $t_{\phi(z,u)}$ were larger than $t_{1\%}$. Therefore, the ratio r_t is inverted. Using the inverted ratios r_t provided by the correlation test suggests using a value between 1 and 4 for both the linear and nonlinear case. This renders a tighter and more reasonable range of values. A sensible suggestion would be to start with an inverted ratio r_t of 2.

6.2.3 Conclusion

By varying the damping in the system we were able to influence the required window length to achieve a given response reconstruction accuracy. In the linear case it was apparent that increasing the damping coefficient lead to a shorter window being required. We have also highlighted the need for choosing sensible values for the window length T_w. Unlike the case of sampling frequency, where being cautious and choosing a high sampling rate results in no adverse effects in terms of reconstruction error, the window length is sensitive to choosing too large a value as the reconstruction error quickly deteriorates. A sensible method of choosing a starting point or at least a range of starting points for the window length has been proposed using the correlation function between the inputs and outputs of the system.



(a) Training: Input displacement

(b) Validation: Input displacement

(c) Test: Input displacement

(d) Training: Sprung mass acceleration

(e) Validation: Sprung mass acceleration

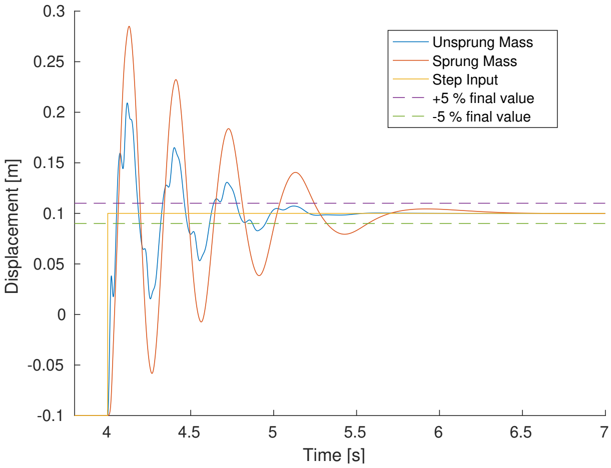
(f) Test: Sprung mass acceleration

(g) Training: Spring displacement

(h) Validation: Spring displacement

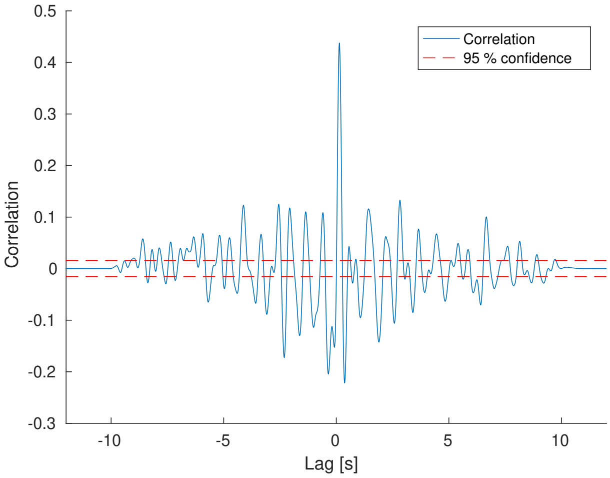
(i) Test: Spring displacement

Figure 6.8: Input and output reconstruction MFFE results as a function of window length. The damping constant b_A was varied to change the settling times of the system. Performed on the nonlinear quarter car model.



,

Figure 6.9: Step response used to determine the settling time of the system. Nonlinear case with $b_A = 500\ \mathrm{N\,s\,m^{-1}}$.



,

Figure 6.10: Correlation between input and spring displacement, $\phi(\Delta_z, u_{road})$. Used to approximate the settling time of the system. Nonlinear case with $b_A = 500\ \mathrm{N\,s\,m^{-1}}$.

Table 6.4: The results of the step input and auto-correlation tests. The r_t term denotes the ratio rounded up that will result in a 1% MFFE score for the spring displacement reconstruction. *Denotes the inverse ratio.

b_{A}	$t_{\mathrm{step},95\%}$				$t_{\mathrm{step},99\%}$				$t_{\phi(z,u)}$				$t_{1\%}$
							Linear Case						
	z_{R}	r_t	z_{A}	r_t	z_{R}	r_t	z_{A}	r_t	r_t	Δz	r_t	$\ddot{z}_{\mathrm{A}}$	Δz
125	0.58	18	2.86	4	0.88	12	2.96	4	10.40	1	10.28	1	10.49
250	0.28	14	1.53	3	0.52	8	2.28	2	9.82	2*	9.69	2*	3.96
500	0.14	28	0.90	4	0.31	13	1.09	4	9.76	3*	9.64	2*	3.89
1000	0.09	41	0.67	6	0.21	18	1.55	3	9.27	2*	9.63	2*	3.88
2000	0.20	17	0.17	19	0.24	14	1.33	2	9.54	3*	9.61	3*	3.24
							Nonlinear Case						
	z_{R}	r_t	z_{A}	r_t	z_{R}	r_t	z_{A}	r_t	$\ddot{z}_{\mathrm{A}}$	r_t	Δz	r_t	Δz
125	2.91	2	2.95	2	2.93	2	2.96	2	12.38	2*	10.11	1	6.85
250	1.93	1	2.87	1	2.46	1	2.94	1	10.11	4*	8.39	3*	2.44
500	0.95	9	1.57	5	1.19	7	2.22	4	9.76	1	8.89	1	8.36
1000	0.55	13	0.74	9	0.77	9	1.28	6	9.74	1	8.83	1	7.07
2000	0.39	6	0.35	7	0.43	6	0.46	5	9.16	4*	9.61	4*	2.44

Chapter 7

Regression Methods

In the previous chapters we established that ADA is a suitable method of splitting and merging the sliding windows. In this chapter we will investigate the suitability of the regression methods that map the windowed outputs to the windowed inputs. For this purpose we will be comparing SBTR against 3 different linear regression techniques: PLS-SVD, PCR and RR.

7.1 Performance on Varying Degrees of Nonlinearity

In this study we want to compare the performance of the regression techniques in terms of their ability to handle increasing levels of nonlinearity. To this end the response reconstruction will take place with the nonlinearity term k_{NL} in Equation (2.1) being ramped up for each iteration. The values to be used are $k_{NL} \in 1.28 \times 10^n \ \mathrm{N\,m^{-3}}$ where $n = [4, 5, 6, 7, 8]$. The linearised values for the spring stiffness as a function of spring displacement are shown in Figure 7.1. The linearised spring stiffnesses are plotted over the expected range of the spring displacement. This demonstrates how quickly the different values of nonlinearity diverge from a purely linear spring.

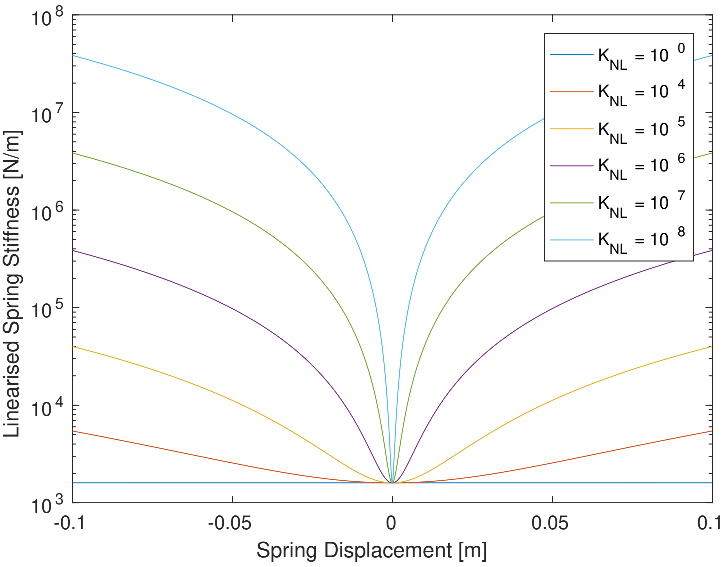


Figure 7.1: The linearised values for the spring stiffness as a function of spring displacement and increasing levels of the nonlinear spring stiffness constant k_{NL}. The values are plotted for the expected range of spring displacement values.

A grid search will be performed to find the appropriate window length and amount of regularisation. The grid search will be performed for $T_w \in [0.1, 6]$ s with 25 equally spaced divisions. The grid search divisions used to search for the number of latent variables used in PCR and PLS-SVD will follow the same method as outlined in Equation (5.1). The potential α values used to regularise RR will be spaced equally on a log scale within the range $\alpha \in [s_{\min} \times 10^{-5}, s_{\max}]$, where s are the singular values. 30 equally spaced divisions will be used. An overview of the numerical investigation is given in Table 7.1.

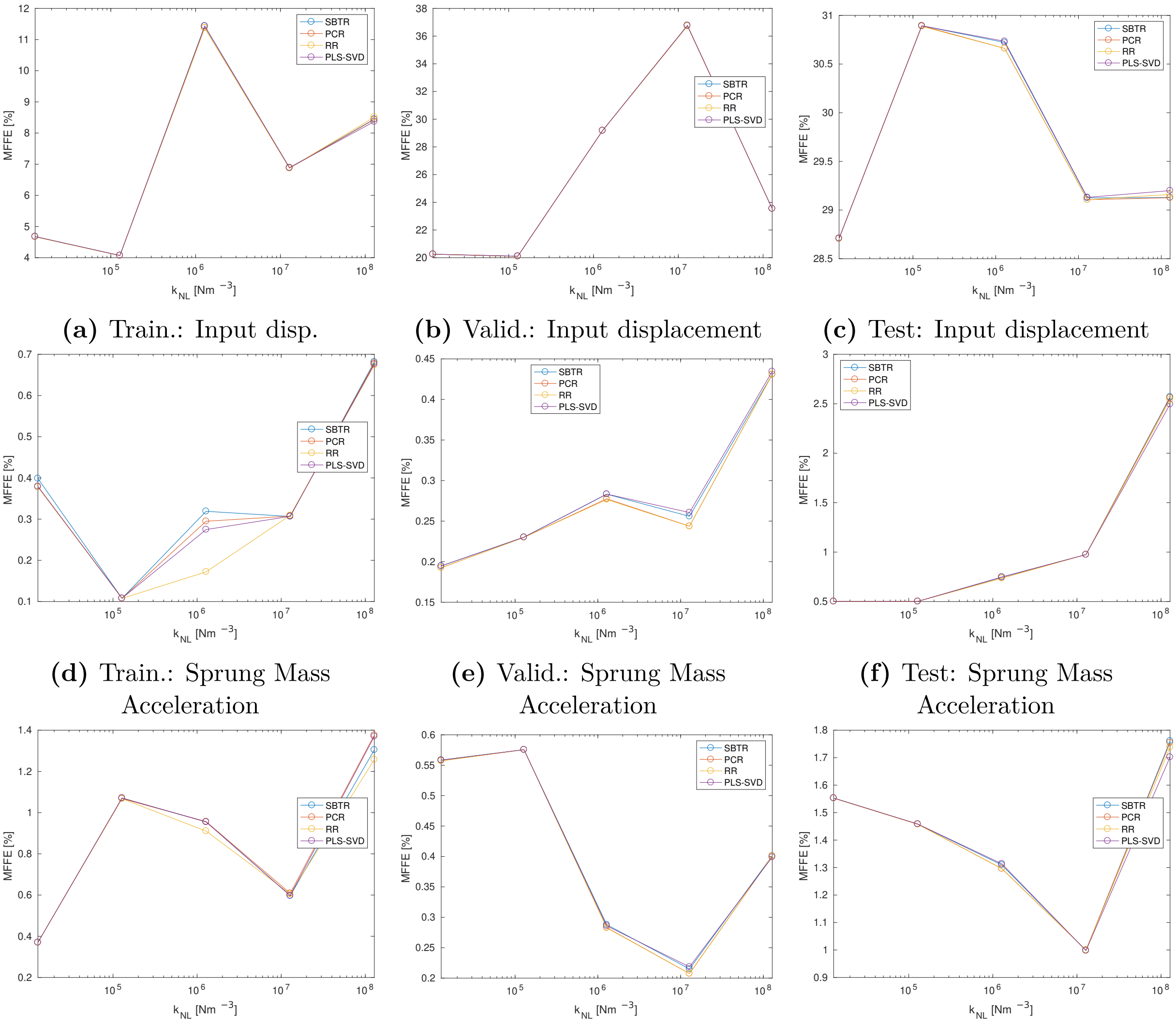
Table 7.1: Experimental design for regression method performance on varying levels of nonlinearity. (Variables of interest shown first).

Variable	**Details**
Regression method	SBTR, PLS-SVD, PCR and RR
QC parameters	Default values; Table 2.1 except with varying levels of nonlinearity k_{NL}
Sensor Configuration	Sprung mass acceleration + spring displacement
Window method	ADA
Nonlinearity constant $\mathbf{k_{NL}}$	1.28×10^{n} N m^{-3} where $n = [4, 5, 6, 7, 8]$
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced divisions
Ridge regression regularisation constant α	$\in [s_{\min} \times 10^{-5}, s_{\max}]$ with 30 divisions spaced logarithmically
Winodw lengths $\mathbf{T_w}$	$\in [0.1\text{ s}, 6\text{ s}]$ with 25 equally spaced divisions.
Window proportional overlap γ	Maximum
Sampling frequency $\mathbf{f_s}$	1000 Hz
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

7.1.1 Results

The training, validation and test results for the various levels of nonlinearity are plotted in Figure 7.2. The only noticeable difference in performance occurs for the training reconstruction error for the sprung mass acceleration as demonstrated in Figure 7.2f. Otherwise, there is remarkably insignificant difference between the various regression methods. Overall the regression methods performed well in reconstructing the outputs of the system over a range of nonlinear spring stiffness values. The reconstructed inputs and outputs for the nonlinear quarter car model with $k_{\text{NL}} = 1.28 \times 10^{7}$ are shown in Figure 7.3. Qualitatively, we can note that the signals produced by the different regression methods are similar and that they have managed to recreate the nonlinear response of the system fairly well.

The latent variable cross validation results for the $k_{\text{NL}} = 1.28 \times 10^{7}$ iteration are shown in Figure 7.4. Again we note the trend highlighted in Section 5.3 of the ADA cross validation results flat-lining after a certain number of retained latent variables. PCR and SBTR perform similarly in that they both begin to converge to the same values at roughly 2000 retained latent variables. PLS-SVD appears to require significantly more latent variables when used in conjunction with ADA. The PLS-SVD results converge at roughly 4500 retained latent variables.



(a) Train.: Input disp.

(b) Valid.: Input displacement

(c) Test: Input displacement

(d) Train.: Sprung Mass Acceleration

(e) Valid.: Sprung Mass Acceleration

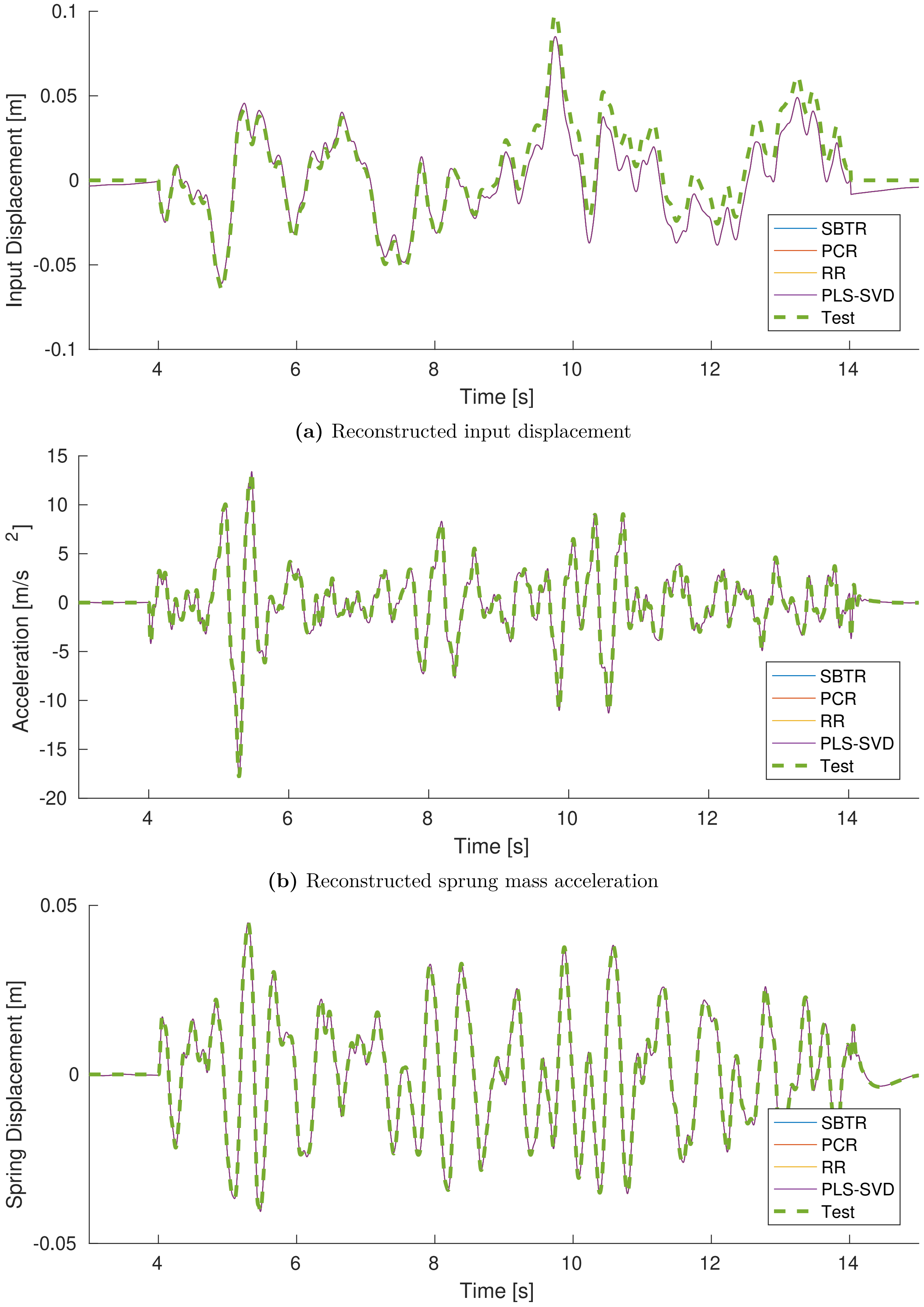
(f) Test: Sprung Mass Acceleration

(g) Train.: Spring Displacement

(h) Valid.: Spring Displacement

(i) Test: Spring Displacement

Figure 7.2: Response reconstruction MFFE results for increasing levels of nonlinearity

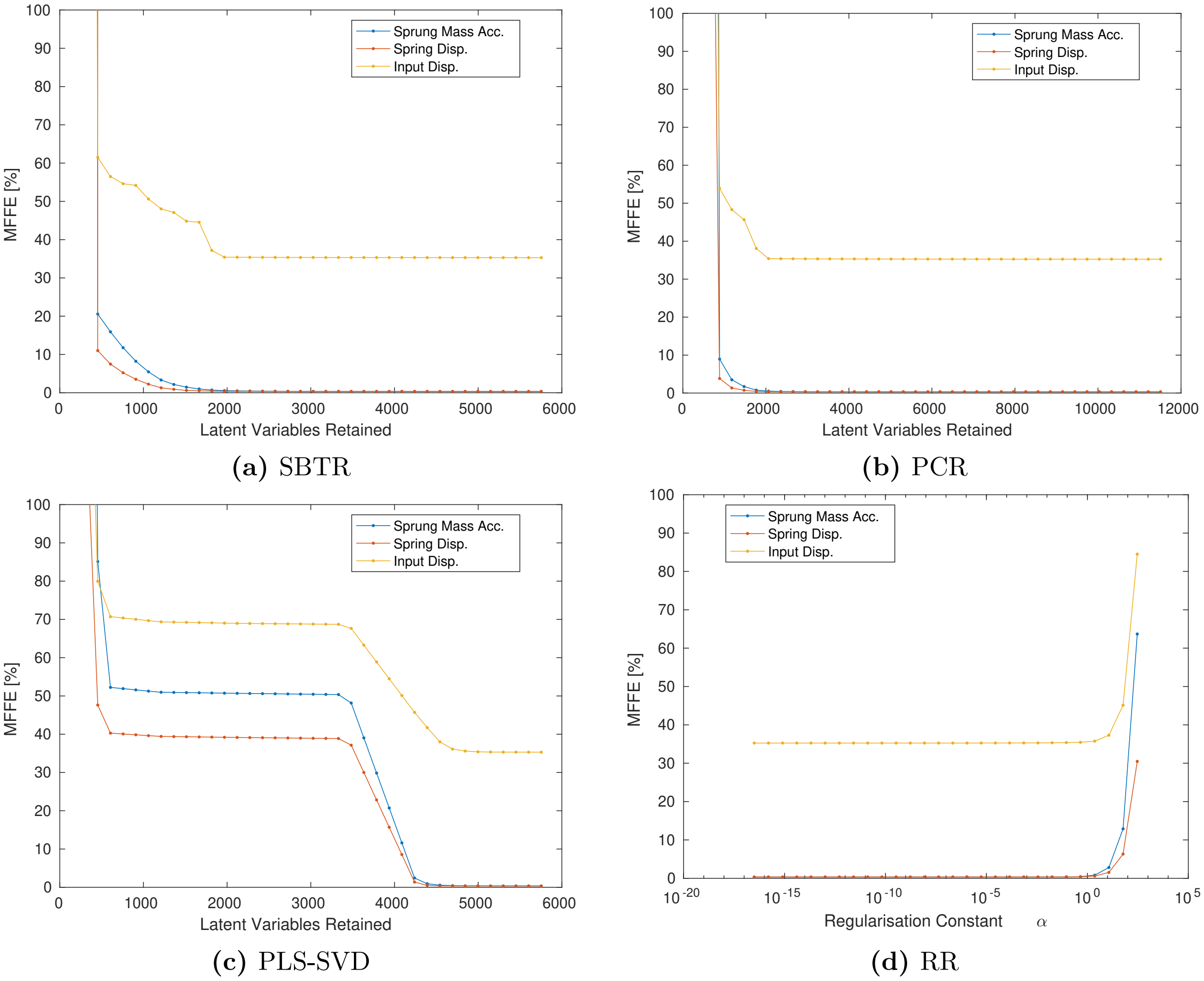


(a) Reconstructed input displacement

(b) Reconstructed sprung mass acceleration

(c) Reconstructed spring displacement

Figure 7.3: Reconstructed inputs and outputs using ADA and different regression methods. Performed on the nonlinear quarter car with nonlinear spring stiffness $k_{NL} = 1.28 \times 10^7\ \mathrm{N\,m^{-3}}$.



(a) SBTR

(b) PCR

(c) PLS-SVD

(d) RR

Figure 7.4: Cross validation results for the nonlinear quarter car with with nonlinear spring stiffness $k_{NL} = 1.28 \times 10^7$.

7.1.2 Conclusion

It appears that the choice of the regression method has little to no effect on performance in terms of the accuracy of the output reconstruction as a function in terms of the nonlinearity of the system. One noticeable difference in terms of performance is the number of latent variables required for the regression methods. It appears that the number of latent variables required by PLS-SVD is significantly larger than those required by SBTR or PCR. In this dissertation, we focused on simply ramping up the nonlinearity constant k_{NL} of the system. A more in-depth approach would be to test against a wider class of nonlinearities such as discontinuities or saturations.

7.2 Robustness to Measurement Noise

The purpose of this chapter is to determine whether the different linear regression methods combined with ADA presents differences in accuracy when measurement noise is present in the output channels. To test this, SBTR will be benchmarked against PCR, RR and PLS-SVD, at various levels of measurement noise.

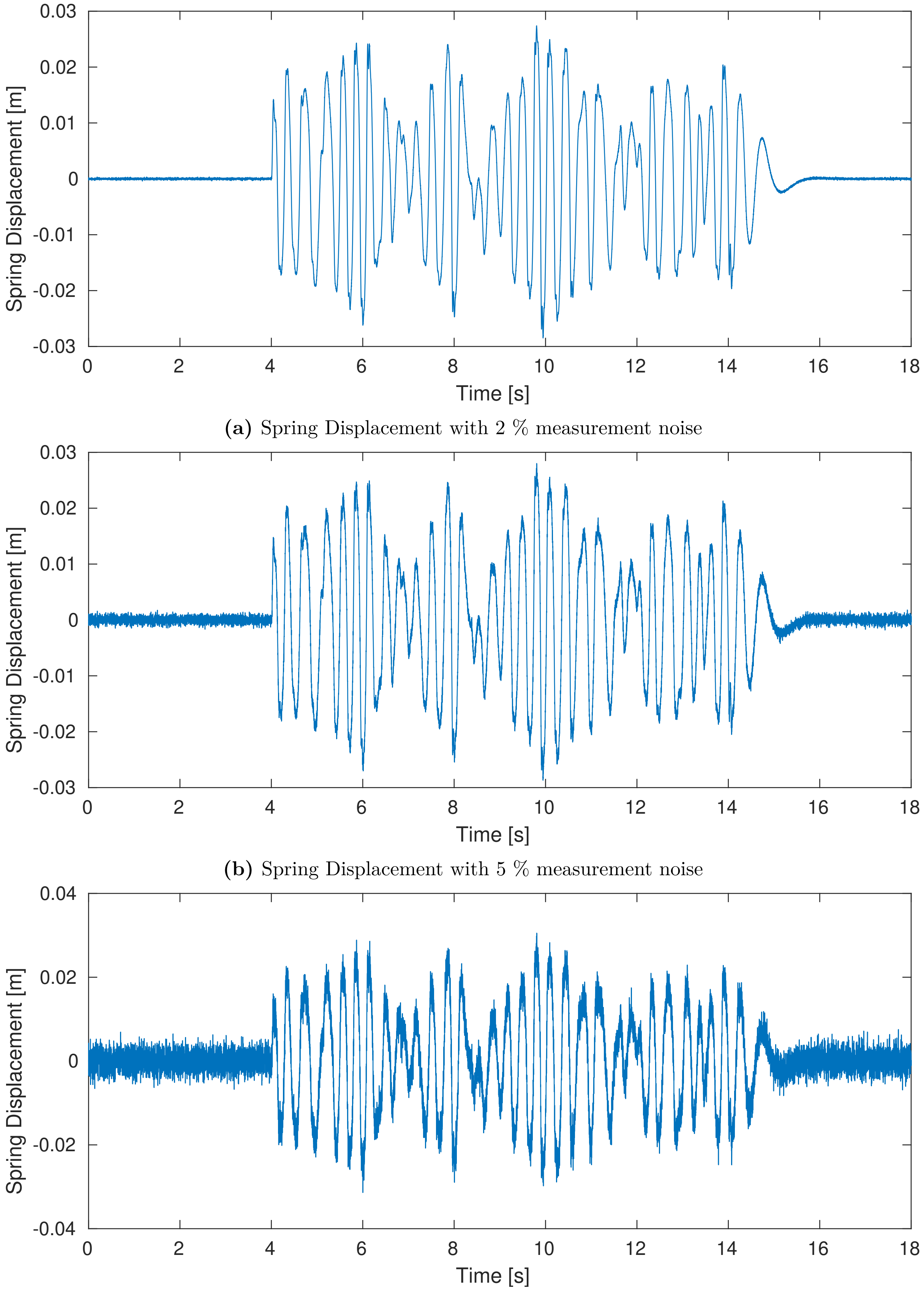
The various regression methods will be performed at 3 different noise levels. The noise levels will be set to 2, 5 and 20 % as defined by Equation (5.25). The spring displacement output corrupted by the different levels of noise are shown in Figure 7.5.

The numerical investigation will be conducted with a linear and nonlinear quarter car set-up. Due to the stochastic nature of the numerical experiment, the experiment will be performed 30 times. The corresponding mean and standard deviation of the MFFE between the true and the approximated noisy signal will be recorded. We can create a baseline MFFE by taking the average MFFE between the noiseless test outputs and the noisy outputs over all the test iterations. We would expect that the best approximation that could be achieved is that of the baseline results. The probabilities of the given regression methods achieving an MFFE less than that of SBTR, i.e. $P(\text{MFFE} < \text{MFFE}_{\text{SBTR}})$, will be calculated. An overview of the numerical investigation design is shown in Table 7.2.

7.2.1 Results

For the linear case, whose test reconstruction results are highlighted in Table 7.3, we note that the input reconstruction accuracy is only slightly affected by the increase in the noise levels. The mean varies only slightly with the increase in noise. The standard deviation of the input does, however, increase noticeably with the increase in noise. In terms of the output reconstruction, the accuracy degrades with the increase in noise levels. If we compare the approximated MFFE against the baseline MFFE we note that the approximated results are initially close to the baseline results but the difference between the baseline and the approximated results get progressively larger as the noise level increases. The probabilities of the other regression method obtaining a better MFFE for the spring displacement are all close to 0.5 which means there is no clearly better regression method in terms of spring displacement recreation. This observation changes slightly when we compare the probabilities for sprung mass acceleration, where RR obtains a probability in the range 0.2 to 0.4. However the mean values are not significantly different. Overall, for the linear case, it appears that no regression method performs significantly better than any other method.

By comparing the results for the nonlinear case in Table 7.4, it becomes obvious that PLS-SVD has failed to produce reasonable results. This is the case even when no noise is present. In order to reduce the computational time for running 30 experiments, the signals were sampled at 350 Hz whereas in Section 7.1 the signals were sampled at 1000 Hz. The remaining regression methods performed similarly without being affected by the downsampling. It was revealed in



(a) Spring Displacement with 2 % measurement noise

(b) Spring Displacement with 5 % measurement noise

(c) Spring Displacement with 20 % measurement noise

Figure 7.5: Test spring displacement outputs contaminated with increasing levels of noise

Table 7.2: Experimental design for regression method performance on varying levels of noise. (Variables of interest shown first).

Variable	**Details**
Reg. method	SBTR, PLS-SVD, PCR and RR
QC parameters	Default values; Table 2.1
Nonlinearity constant $\mathbf{k_{NL}}$	Linear: 0, Nonlinear: 1.28×10^{7} N m^{-3}
Noise level $\eta_{\%}$	$\in [2, 5, 20]$ %
Sensor Config.	Sprung mass acceleration + spring displacement
Window length $\mathbf{T_w}$	5.5 s
Sampling frequency $\mathbf{f_s}$	Linear: 250 Hz, Nonlinear: 350 Hz
Window Proportional overlap γ	Maximum
Number of retained latent variables L	$\in [1, L_{\max}]$ with $\min(40, \max(10, \lceil L_{\max}/10 \rceil))$ equally spaced divisions
Ridge regression regularisation constant α	$\in [s_{min} \times 10^{-5}, s_{max}]$ with 30 divisions spaced logarithmically
Windowing method	ADA
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

Table 7.3: Linear case: test MFFE [%] results as a function of varying measurement noise levels $\eta_{\%}$. The probability of the regression method obtaining an MFFE less than the MFFE obtained by SBTR is also given. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

Regression Method	$\eta_{\%} = 0$	$\eta_{\%} = 2$			$\eta_{\%} = 5$			$\eta_{\%} = 20$		
	Mean	Mean	Std Dev.	Prob.	Mean	Std Dev.	Prob.	Mean	Std Dev.	Prob.
Input Displacement										
SBTR	28.38	28.33	0.15	-	28.41	0.50	-	30.61	1.92	-
PCR	29.30	28.43	0.26	0.37	28.48	0.52	0.46	30.34	1.81	0.54
RR	28.88	28.49	0.45	0.36	28.36	0.86	0.52	29.75	1.78	0.63
PLS-SVD	28.38	28.45	0.18	0.30	28.40	0.48	0.50	29.41	1.88	0.67
Sprung Mass Acceleration										
SBTR	0.46	2.03	0.02	-	4.93	0.04	-	18.22	0.27	-
PCR	0.44	2.03	0.02	0.51	4.94	0.05	0.44	18.26	0.16	0.45
RR	0.45	2.04	0.03	0.39	4.98	0.05	0.22	18.43	0.15	0.26
PLS-SVD	0.46	2.03	0.02	0.59	4.92	0.05	0.60	18.18	0.15	0.55
Baseline	-	1.45	0.01	-	3.62	0.02	-	14.50	0.14	-
Spring Displacement										
SBTR	1.29	2.43	0.02	-	5.23	0.04	-	18.67	0.14	-
PCR	1.17	2.43	0.02	0.48	5.23	0.05	0.48	18.68	0.16	0.49
RR	1.23	2.42	0.04	0.53	5.23	0.03	0.47	18.67	0.19	0.50
PLS-SVD	1.29	2.43	0.02	0.46	5.23	0.03	0.46	18.69	0.13	0.46
Baseline	-	1.48	0.01	-	3.70	0.03	-	14.78	0.1	-

Section 7.1 that PLS-SVD requires significantly more latent variables to achieve similar results as compared to the other regression methods. Therefore, it appears that the downsampling has, to the detriment of PLS-SVD, reduced the number of latent variables available. In order to negate the effects of the downsampling, the nonlinear noise experiment will be repeated for all the regression methods but with a sampling frequency of 1000 Hz. Only a single iteration of the experiment will be performed. The results are shown in Table 7.5. It is clear that by sampling at 1000 Hz that PLS-SVD achieves similar results to the other regression methods when no noise is present but quickly deteriorates in performance once noise is introduced. This suggests that the smaller singular values that PLS-SVD requires are dominated by noise. This results in PLS-SVD achieving poor results.

Table 7.4: Nonlinear case: test MFFE results as a function of varying measurement noise levels $\eta_{\%}$. The probability of the regression method obtaining an MFFE less than the MFFE obtained by SBTR is also given. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

Regression Method	$\eta_{\%} = 0$	$\eta_{\%} = 2$			$\eta_{\%} = 5$			$\eta_{\%} = 20$		
	Mean	Mean	Std Dev.	Prob.	Mean	Std Dev.	Prob.	Mean	Std Dev.	Prob.
Input Displacement										
SBTR	27.78	36.16	3.55	-	50.04	4.93	-	70.81	3.63	-
PCR	27.78	33.79	4.38	0.66	47.43	5.23	0.64	69.62	4.54	0.58
RR	27.78	36.63	4.18	0.47	45.34	3.76	0.78	65.57	4.92	0.80
PLS-SVD	124.97	125.59	0.14	0.00	126.03	0.35	0.00	126.27	1.67	0.00
Sprung Mass Acceleration										
SBTR	0.65	4.23	0.35	-	9.16	0.92	-	29.52	2.72	-
PCR	0.65	4.49	0.43	0.32	9.15	0.87	0.50	29.64	2.75	0.49
RR	0.65	4.43	0.29	0.33	9.32	0.83	0.45	28.10	2.50	0.65
PLS-SVD	161.52	161.42	0.09	0.00	160.37	0.29	0.00	152.30	0.99	0.00
Baseline	-	1.66	0.01	-	4.16	0.03	-	16.65	0.13	-
Spring Displacement										
SBTR	0.65	3.16	0.41	-	7.61	1.04	-	26.75	3.22	-
PCR	0.64	3.28	0.45	0.42	7.42	0.78	0.56	26.90	2.92	0.49
RR	0.64	3.25	0.40	0.44	7.49	0.84	0.54	25.04	2.78	0.66
PLS-SVD	111.55	111.52	0.08	0.00	110.81	0.20	0.00	106.09	0.62	0.00
Baseline	-	1.40	0.01	-	3.51	0.03	-	14.02	0.12	-

If we ignore the PLS-SVD results and return to the nonlinear results in Table 7.4, we note that the reconstruction results are similar across the remaining regression methods. RR performed slightly better at the higher noise levels, by obtaining slightly better probabilities of achieving a better MFFE at $\eta_{\%} = 20$. Otherwise, SBTR performed slightly better at the lower noise levels, $\eta_{\%} = 2$. The difference between the baseline MFFE and the approximated MFFE for all the

Table 7.5: Test MFFE results as a function of varying measurement noise levels $\eta_{\%}$. Nonlinear case repeated but at 1000 Hz. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

Regression Method	$\eta_{\%} = 0$	$\eta_{\%} = 2$	$\eta_{\%} = 5$	$\eta_{\%} = 20$
Input Displacement				
SBTR	29.15	38.23	48.14	69.48
PCR	29.13	35.95	45.41	63.35
RR	29.13	31.89	47.72	66.84
PLS-SVD	29.16	58.85	65.81	86.28
Sprung Mass Acceleration				
SBTR	0.97	3.87	7.70	26.23
PCR	0.97	4.48	9.05	30.29
RR	0.97	4.52	9.31	24.20
PLS-SVD	0.97	5.87	13.92	53.52
Spring Displacement				
SBTR	0.99	2.95	5.76	22.79
PCR	1.00	3.06	7.22	26.96
RR	1.00	3.21	7.79	21.00
PLS-SVD	0.99	5.45	14.45	47.54

noise levels appears to be more significant than those achieved by the linear case. It is not immediately clear as to which of the remaining regression methods are more robust in terms of noise.

The optimized numbers of retained latent variables for the different regression methods are shown in Table 7.6. We note the general trend that the number of retained latent variables decreases as the level of noise increases. This indicates that cross validation is working as it should and is properly regularising the problem.

Table 7.6: Number of latent variables retained as a function of noise, $\eta_{\%}$. In the case of RR the constant α is presented.

Regression Method	$\eta_{\%} = 0$	$\eta_{\%} = 2$		$\eta_{\%} = 5$		$\eta_{\%} = 20$	
Linear Case							
	Mean	Mean	Std Dev.	Mean	Std Dev.	Mean	Std Dev.
SBTR	1375	1056.37	204.69	1002.97	228.72	945.30	235.78
PCR	1968	1133.60	312.67	1488.50	702.48	1444.63	751.72
PLS-SVD	1375	1084.57	206.71	954.37	251.42	840.23	324.15
RR	8.06×10^{-6}	1.44	0.99	1.23	1.62	2.72	4.77
Nonlinear Case							
	Mean	Mean	Std Dev.	Mean	Std Dev.	Mean	Std Dev.
SBTR	2800	2621.50	220.76	2629.53	261.68	2353.20	603.81
PCR	3594	4561.40	1084.44	3983.30	1265.39	3797.57	1522.11
PLS-SVD	2799	2731.20	104.21	2671.63	146.27	2345.50	411.00
RR	0.02	0.13	0.29	0.42	0.89	2.83	3.90

7.2.2 Conclusion

After comparing the regression methods across a range of noise levels for both a linear and nonlinear quarter car set-up, it was found that the different regression methods offered little or no benefits in terms of noise robustness when compared to one another. However, it was determined that PLS-SVD performs poorly in terms of noise. PLS-SVD appears to require a large number of latent variables for similar performance which suggests that its smaller singular values are easily dominated by noise.

7.3 Robustness to Model Mismatch

The goal of this section is to investigate the ability of the different regression methods to handle model mismatch. Model mismatch occurs when the identified system does not truly represent the physical test rig. In the case of response reconstruction the misrepresentation occurs when the physical test rig is taken from the real world and recreated and simulated in the laboratory environment. Typically the degrees of freedom are not fully represented in the laboratory or the test rig parameters, such as mass, may vary. It is in this laboratory environment that the process of system identification occurs and therefore we will have mapped a domain that differs from the real world domain. When the real world outputs need to be recreated we may find that the mapped inverse model may be forced to extrapolate into regions in the mapped domain to find a solution. In other words the inverse model has over-fitted to the laboratory domain and generalizes poorly with regards to the real world domain. Regularization can be employed to minimize this error [2]. In the course of this experiment, we will focus on a narrower scope of model mismatch whereby the model parameters of the system are simply scaled from the real world environment to that of the laboratory environment. A broader scope of model mismatch would be to add new dynamics going from one environment to the other. One such dynamic would be to add or remove a discontinuity i.e. a tyre separating from the road surface. In this dissertation, we will focus on the narrower view of model mismatch.

In the next section the issue of model mismatch will be investigated on a linear and nonlinear quarter car set-up. The performance of the different regression methods will also be tested against model mismatch.

To test the effects of model mismatch, a series of test rigs representing different levels of model divergence will be used as the laboratory set-up. The default parameters given in Table 2.1 will be modified such that

$$M_{\mathrm{A,mis}} = M_{\mathrm{A}}\left(1-\frac{m_{\%}}{100}\right) \tag{7.1}$$

$$M_{\mathrm{R,mis}} = M_{\mathrm{R}}\left(1+\frac{m_{\%}}{100}\right) \tag{7.2}$$

$$b_{\mathrm{A,mis}} = b_{\mathrm{A}}\left(1-\frac{m_{\%}}{100}\right) \tag{7.3}$$

$$k_{\mathrm{A,mis}} = k_{\mathrm{A}}\left(1+\frac{m_{\%}}{100}\right) \tag{7.4}$$

$$k_{\mathrm{R,mis}} = k_{\mathrm{R}}\left(1-\frac{m_{\%}}{100}\right) \tag{7.5}$$

where $m_{\%}$ is the percentage mismatch. The outputs sampled from the real world rig will be obtained using the default parameters. For the first set of tests, the validation outputs will come from the modified laboratory rig with the test output set obtained from the real world rig. In the second set of tests, the outputs used for the validation will come from a separate validation set obtained from the real world test rig. The idea behind this is to force the cross validation to only retain latent variables that allow the laboratory environment to recreate dynamics that are common to both the real world and the lab environment. An overview of the numerical experiment parameters are given in Table 7.7.

7.3.1 Results

The test results for the linear case with the laboratory and real world cross-validation sets are shown in Table 7.8. In the case of laboratory cross validation we note that PCR and RR have performed poorly with the introduction of model mismatch. However, when we switch to using the outputs from the real world for validation we note that PCR and RR perform almost identically to SBTR and PLS-SVD. To gain insight into this behaviour we can turn to the cross validation latent variable results plotted in Figure 7.6. The cross validation latent variable results are plotted for model mismatch, $m_{\%} = 20\%$. For the case where the real world outputs were used we note that PCR and RR experience a large jump in the cross validation MFFE results when too many of the latent variables are retained. This indicates that the model is overfitting at this point and can not generalize well to the mismatched model. The cross validation results, while using the laboratory created outputs, cannot distinguish that the regression model is overfitting. As to why SBTR and PLS-SVD perform the same regardless

Table 7.7: Experimental design for regression method performance on varying levels of model mismatch. (Variables of interest shown first).

Variable	**Details**
Regression method	SBTR, PLS-SVD, PCR and RR
QC parameters	Perturbed as per Equations (7.1) to (7.5) with $m_{\%} \in [0, 2, 5, 10, 20]$
Sensor Configuration	Sprung mass acceleration + spring displacement
Windowing method	ADA
Nonlinearity constant $\mathbf{k_{NL}}$	Linear: 0, Nonlinear: 1.28×10^{7} $\mathrm{N\,m^{-3}}$
Window length $\mathbf{T_w}$	5.5 s
Sampling frequency $\mathbf{f_s}$	Linear: 250 Hz, Nonlinear: 350 Hz
Window Proportional overlap γ	Maximum
Number of retained latent variables L	$\in [1, L_{\mathrm{max}}]$ with $\min(40, \max(10, \lceil L_{\mathrm{max}}/10 \rceil))$ equally spaced divisions
Ridge regression regularisation constant α	$\in [s_{min} \times 10^{-5}, s_{max}]$ with 30 divisions spaced logarithmically
Noise level $\eta_{\%}$	0 %
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

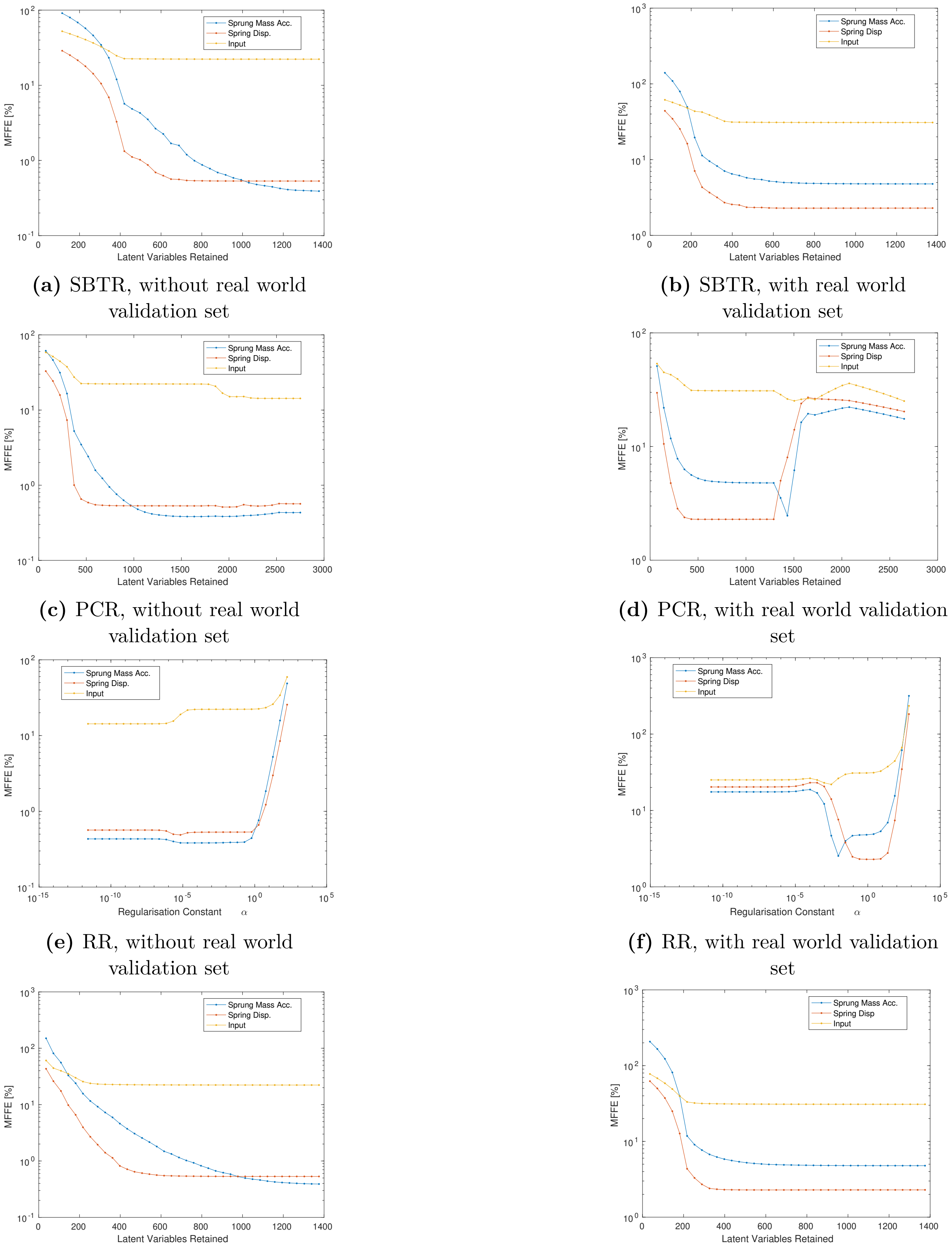
is due to the fact that the number of latent variables they can utilize is limited by the size of the predictor matrix X. The quarter car model has a single input and multiple outputs. This artificially lowers the upper bound on the number of latent variables that they can extract. A sample linear reconstruction for the case of model mismatch, $m_\% = 20\%$ is shown in Figure 7.7.

Table 7.8: Linear case: test MFFE [%] results as a function of varying levels of model mismatch. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

Reg. Method	$m_\% = 0$		$m_\% = 2$		$m_\% = 5$		$m_\% = 10$		$m_\% = 20$	
Input Displacement										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	29.16	29.16	29.84	29.84	30.85	30.85	32.53	32.53	37.65	37.65
PCR	29.54	29.53	1.21×10^5	29.84	2.80×10^5	30.85	5.93×10^5	32.53	1.12×10^6	37.65
RR	29.32	29.28	9.68×10^4	29.84	1.32×10^5	30.86	1.25×10^5	32.55	5.84×10^5	37.69
PLS-SVD	29.16	29.16	229.84	29.84	30.85	30.85	32.53	32.53	37.65	37.65
Sprung Mass Acceleration										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	0.50	0.50	0.94	0.94	2.19	2.19	4.53	4.53	9.83	9.84
PCR	0.49	0.49	4.90×10^4	0.94	1.12×10^5	2.19	2.45×10^5	4.53	4.81×10^5	9.84
RR	0.50	0.50	4.83×10^4	0.94	6.59×10^4	2.19	6.43×10^4	4.53	2.90×10^5	9.84
PLS-SVD	0.50	0.50	0.94	0.94	2.19	2.19	4.53	4.53	9.83	9.84
Spring Displacement										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	1.27	1.27	1.25	1.25	1.25	1.25	1.36	1.36	1.95	1.95
PCR	1.23	1.23	1.05×10^5	1.25	2.41×10^5	1.25	5.31×10^5	1.36	1.02×10^6	1.95
RR	1.25	1.25	9.39×10^4	1.25	1.34×10^5	1.25	1.31×10^5	1.36	5.98×10^5	1.95
PLS-SVD	1.27	1.27	1.25	1.25	1.25	1.25	1.36	1.36	1.95	1.95

The test results for the nonlinear case are given in Table 7.9. In this case we note that using the outputs from the real world or from the modified laboratory set-up, for the validation sets, results in almost identical test performances. A sample nonlinear reconstruction for the case of model mismatch, $m_\% = 20\%$ is shown in Figure 7.8.

The output reconstruction results are more sensitive to model mismatch for the nonlinear case as compared to the linear case. The output reconstruction scores become significantly degraded



(a) SBTR, without real world validation set

(b) SBTR, with real world validation set

(c) PCR, without real world validation set

(d) PCR, with real world validation set

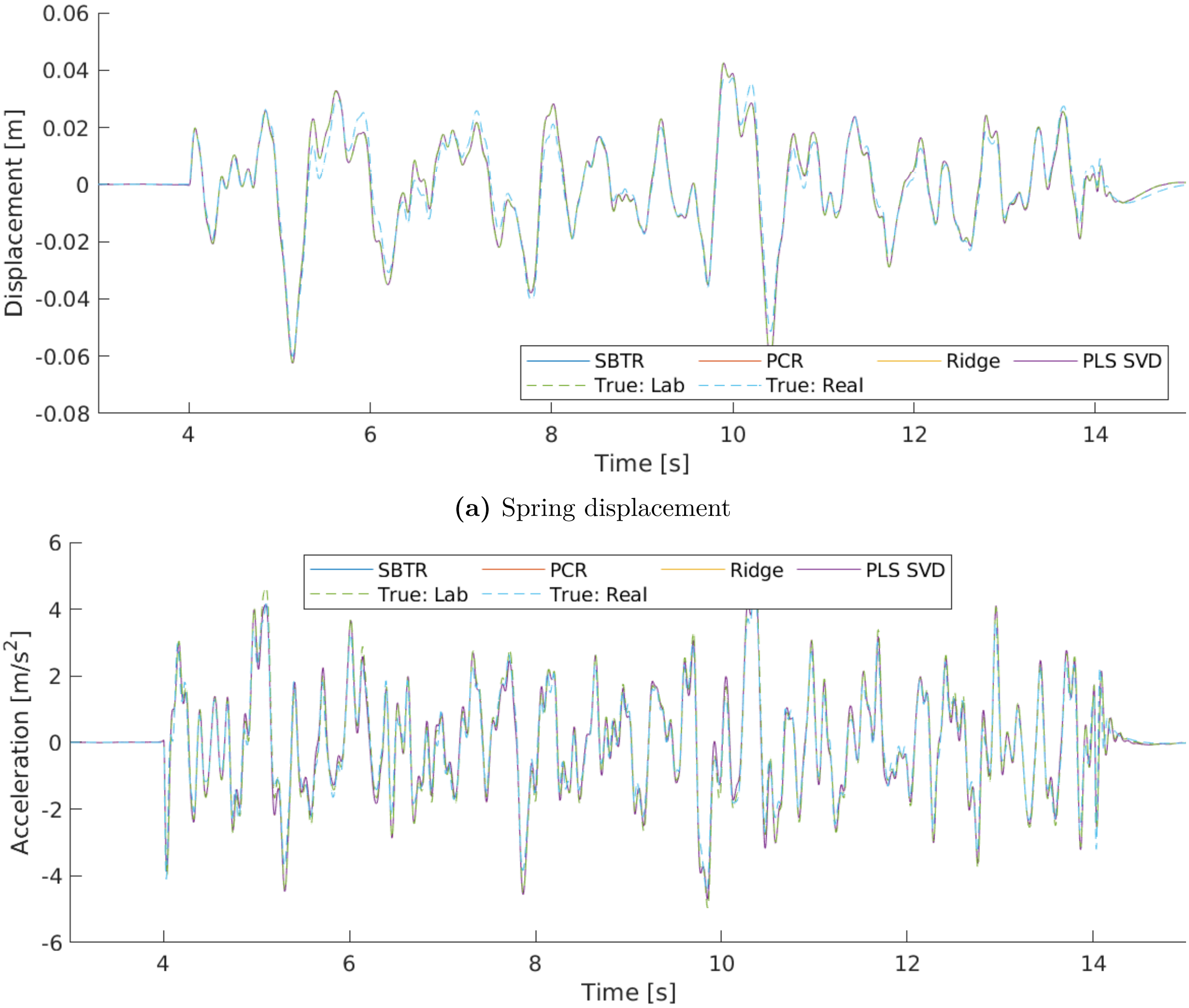
(e) RR, without real world validation set

(f) RR, with real world validation set

(g) PLS-SVD, without real world validation set

(h) PLS-SVD, with real world validation set

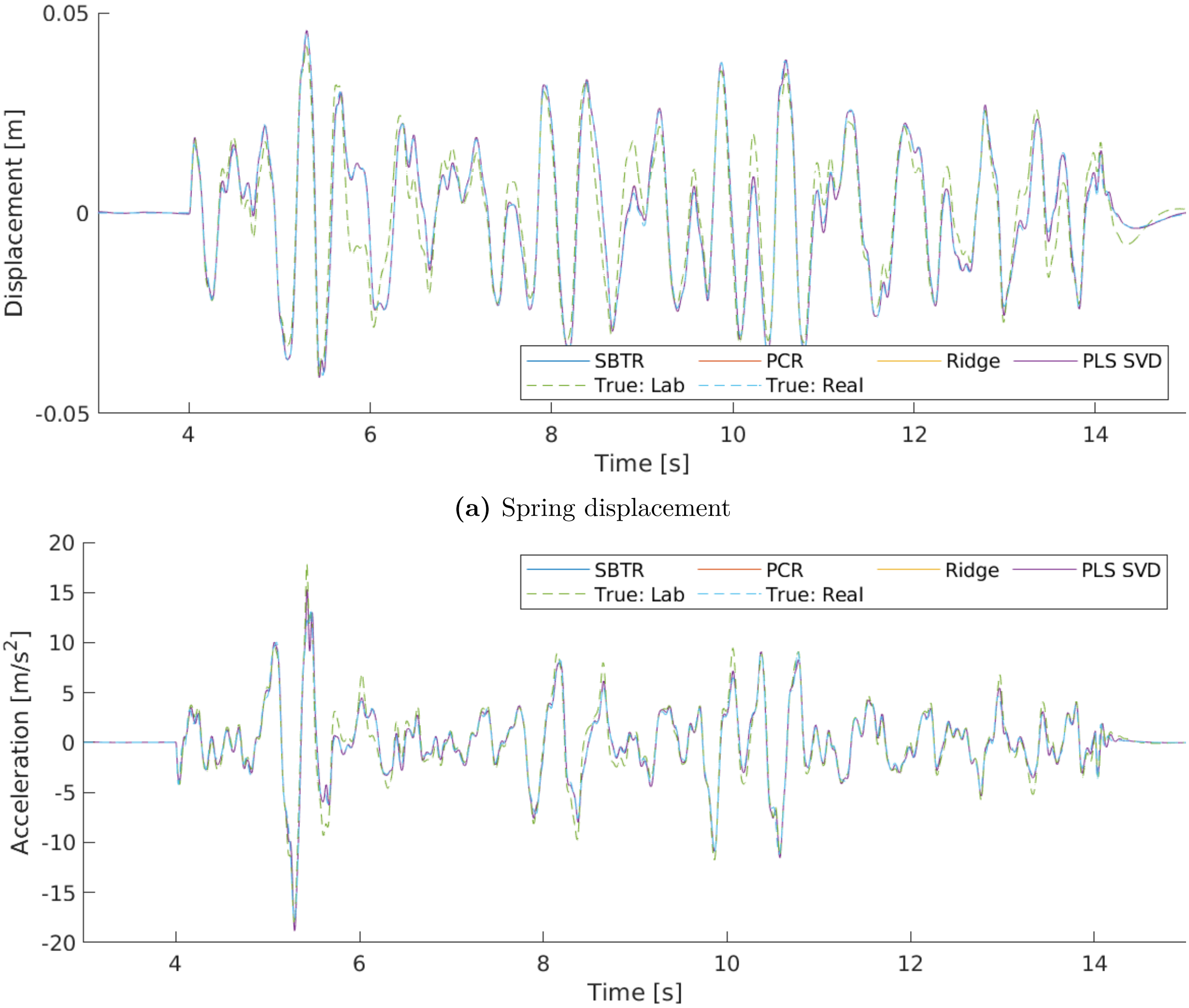
Figure 7.6: Cross Validation results using validation outputs from either the laboratory set-up or from the real world set-up. Linear case with model mismatch $m_{\%} = 20\%$



(a) Spring displacement

(b) Sprung mass acceleration

Figure 7.7: Response reconstructions for the case of model mismatch, $m_{\%} = 20\%$. Linear case.



(a) Spring displacement

(b) Sprung mass acceleration

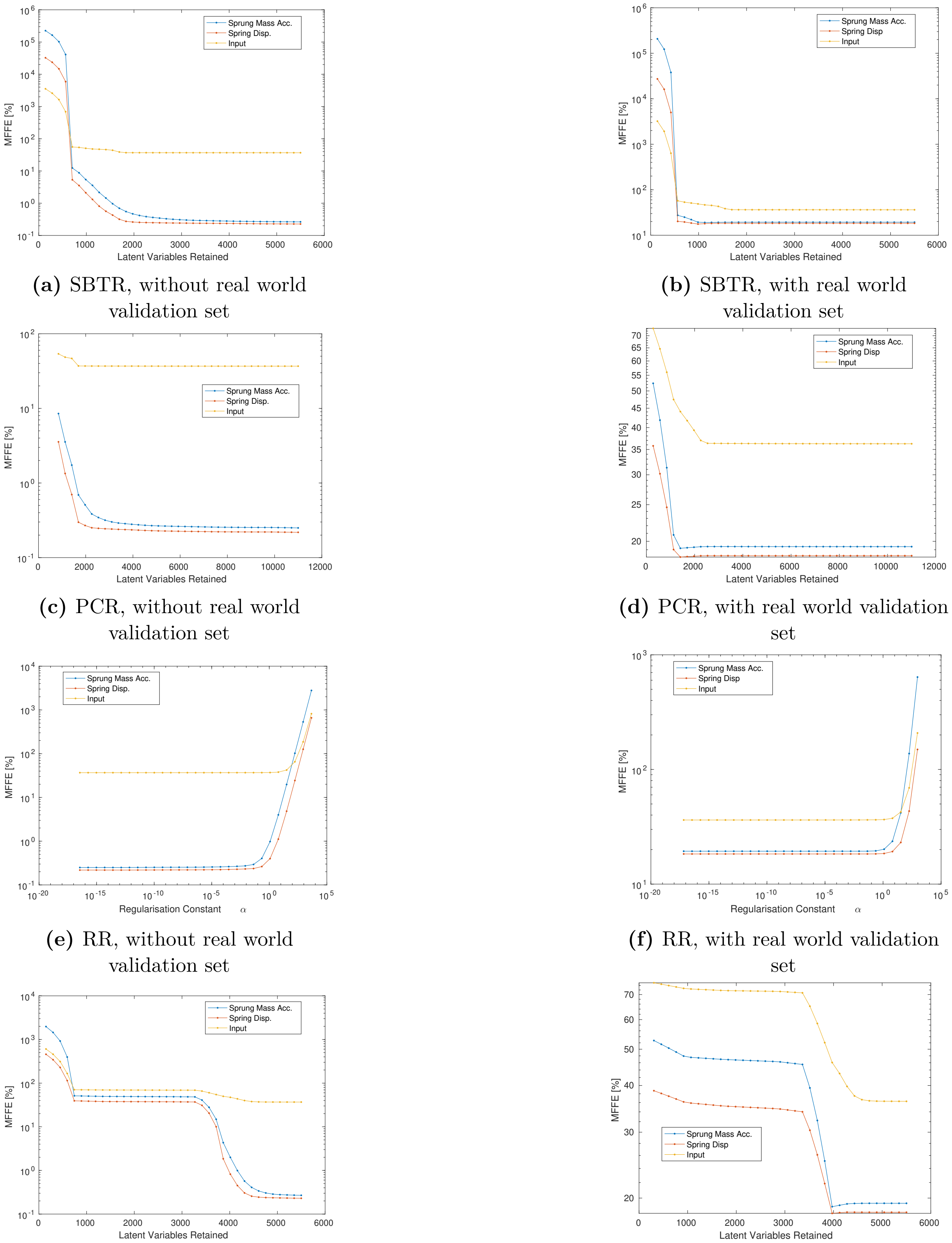
Figure 7.8: Response reconstructions for the case of model mismatch, $m_{\%} = 20\%$. Nonlinear case.

as the level of the model mismatch increases. The cross validation results for the nonlinear case for the different cross validation set-ups are shown in Figure 7.9. The cross validation results show less of a pronounced minimum as compared to the linear case when using the real world outputs for cross validation. The minimums do shift to a lower number of retained variables when using the real world data set as a validation set, as demonstrated in Table 7.10. However, the difference in validation scores between using the minimum or all of the latent variables are small. This effect is noticeable in Figures 7.9d and 7.9h where there are distinct minimums but the differences in MFFE at the minimums and using all of the latent variables are small. One possible reason for this is that the linear regression methods have difficulty fitting the nonlinear system. In other words the regression methods are underfitting and the problem is therefore regularised in a sense not by the truncated number of latent variables but rather by the form of the linear regression.

Table 7.9: Nonlinear case: test MFFE [%] results as a function of varying levels of model mismatch. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

Regression Method	$m_{\%} = 0$		$m_{\%} = 2$		$m_{\%} = 5$		$m_{\%} = 10$		$m_{\%} = 20$	
Input Displacement										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	49.56	49.56	47.48	47.48	44.19	44.19	38.53	38.53	29.86	29.86
PCR	49.53	49.53	47.46	47.46	44.17	44.17	38.51	38.51	29.85	29.85
RR	49.53	49.53	47.46	47.46	44.17	44.17	38.51	38.51	29.85	29.85
PLS-SVD	49.57	49.57	47.49	47.49	44.20	44.20	38.54	38.54	29.86	29.86
Sprung Mass Acceleration										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	1.19	1.19	2.57	2.57	5.66	5.66	10.99	10.99	21.49	21.49
PCR	1.18	1.18	2.55	2.55	5.64	5.64	10.97	10.97	21.48	21.48
RR	1.18	1.18	2.55	2.55	5.64	5.64	10.97	10.97	21.48	21.48
PLS-SVD	1.19	1.19	2.58	2.58	5.67	5.67	11.00	11.00	21.50	21.50
Spring Displacement										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	1.19	1.19	3.36	3.36	7.80	7.80	15.83	15.83	31.06	31.06
PCR	1.18	1.18	3.33	3.33	7.77	7.77	15.80	15.80	31.04	31.04
RR	1.18	1.18	3.33	3.33	7.77	7.77	15.80	15.80	31.04	31.04
PLS-SVD	1.19	1.19	3.37	3.37	7.81	7.81	15.84	15.84	31.07	31.07

In terms of the performance differences between the regression methods themselves, the differences are negligible with no apparent benefit in terms of model mismatch compared to one another. The differences come down to a second decimal place in terms of percentages when a suitable cross validation method is in place.



(a) SBTR, without real world validation set

(b) SBTR, with real world validation set

(c) PCR, without real world validation set

(d) PCR, with real world validation set

(e) RR, without real world validation set

(f) RR, with real world validation set

(g) PLS-SVD, without real world validation set

(h) PLS-SVD, with real world validation set

Figure 7.9: Cross Validation results using validation outputs from either the laboratory set-up or from the real world set-up.. Nonlinear case with model mismatch $m_{\%} = 20\%$

Table 7.10: Number of latent variables retained as a function of model mismatch, $m_{\%}$. In the case of RR the constant α is presented.

Regression Method	$m_{\%} = 0$		$m_{\%} = 2$		$m_{\%} = 5$		$m_{\%} = 10$		$m_{\%} = 20$	
Linear Case										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	1375	1375	1375	1375	1375	1173	1375	780	1375	760
PCR	1974	1954	2081	1303	2099	1158	2026	743	2007	573
PLS-SVD	1375	1375	1375	1375	1375	1340	1375	780	1375	579
RR	6.6×10^{-6}	8.7×10^{-6}	5.4×10^{-6}	1.9×10^{-1}	9.5×10^{-6}	5.6×10^{-1}	6.3×10^{-6}	7.5×10^{-1}	6.8×10^{-6}	2.51
Nonlinear Case										
	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.	Lab.	Real.
SBTR	5500	5500	5500	5500	5500	1158	5500	1189	5499	987
PCR	10999	11000	11000	11000	10998	1189	10999	1128	10998	1410
PLS-SVD	5500	5500	5500	5500	5500	4086	5500	4014	5499	3972
RR	1.8×10^{-16}	1.3×10^{-13}	1.7×10^{-16}	1.8×10^{-16}	6.5×10^{-13}	4.8×10^{-15}	5.0×10^{-15}	1.27	3.5×10^{-17}	6.7×10^{-18}

7.3.2 Conclusion

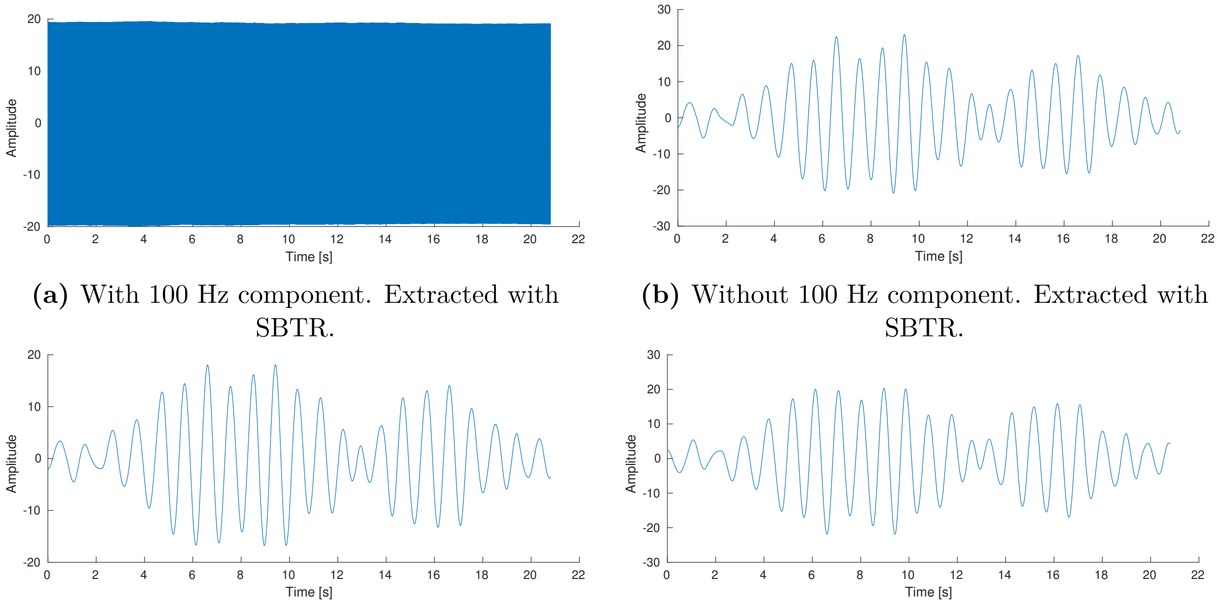
In this chapter it was demonstrated that regularisation can be employed to mitigate the effects of model mismatch. The use of ADA combined with regularisation is effective when applied to the linear quarter car model but offers no benefit when applied to the nonlinear quarter car system. There are no apparent advantages in choosing one regression method over another in terms of mitigating the effects of model mismatch provided a sensible cross validation method is used.

7.4 Analysis of SBTR in terms of Direct Inverse Regression for Response Reconstruction

In the previous chapters we demonstrated that the results obtained from using SBTR as a regression method performed similarly to PCR and RR. In this section we will demonstrate as to why we would expect SBTR to perform similarly to PCR in terms of response reconstruction through the use of direct inverse regression.

To demonstrate this we employ the default linear quarter car set-up. The arguments given are based on the results discussed in Section 2.3. The linear quarter car set-up will be chosen since we know that the frequency content of the output of a system can only contain the same frequency content of the input. The absolute displacement sensor set-up will be used. The magnitude Bode plot of the quarter car system is shown in Figure 2.7. The Bode plot indicates the gain of the output of the system for a given frequency in the input. We note that the higher frequencies of the system are significantly attenuated. In other words if we added a high frequency component to the input whose frequency is well beyond the cut-off frequency we would expect the corresponding output of the system to be unaffected by the high frequency component. This is demonstrated in Figure 2.8 where two inputs are passed through the linear quarter car. The input signals are identical except the one has a 100 Hz frequency component added to it with an amplitude that engulfs the signal. However, the corresponding outputs to different inputs are identical.

Recall that the target filtering offered by SBTR is done independently of the predictor. Therefore, it has no insight if it has any effect on the predictor. The target filtering determines the most significant scores by how much variance it contributes to the total variance of the target matrix. In this case we added a large amplitude high frequency component to the input that clearly dominates the variance of the signal. Therefore, we would expect the target filtering to retain this high frequency content. This is exactly what it does as we can see by the first score of the target matrix plotted for the different input signals in Figure 7.10. The first scores extracted by PLS-SVD are also shown. Note that they do not retain the high frequency component.



(a) With 100 Hz component. Extracted with SBTR.

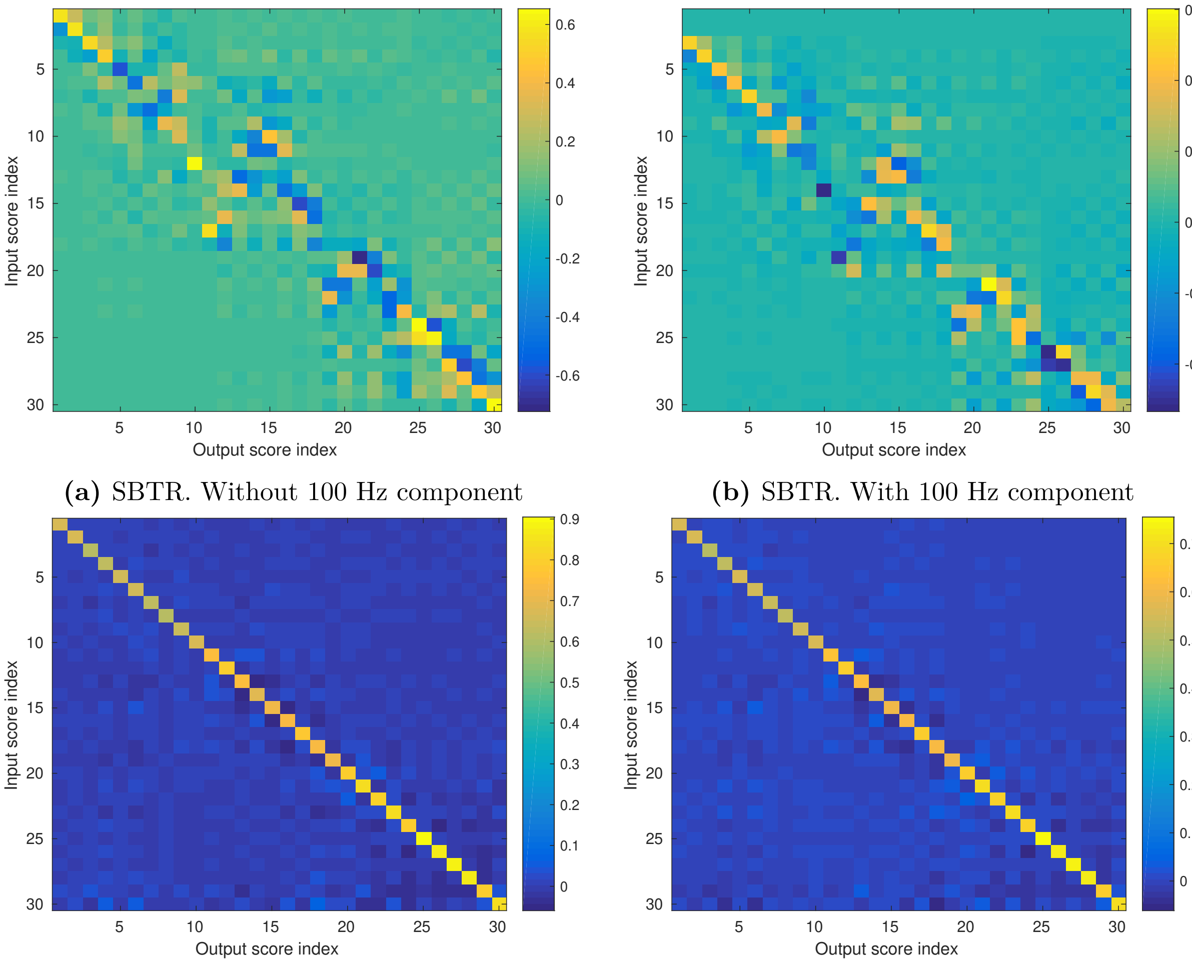
(b) Without 100 Hz component. Extracted with SBTR.

(c) With 100 Hz component. Extracted with PLS-SVD.

(d) Without 100 Hz component. Extracted with PLS-SVD.

Figure 7.10: First scores for the test inputs where the one signal has a large amplitude 100 Hz signal added to it.

As a consequence of the linear system we would expect the frequency components in the outputs to only covary with the corresponding frequency components in the inputs. If we make the assumption that certain scores are closely related to certain frequency components, then we might expect that some input score would only vary with a select few output scores. So for the case of SBTR with the spurious high frequency in the input we would expect the input score associated with the spurious signal to covary only with the output score associated with the same frequency in the output. To demonstrate this the β weights which map the output scores to the input scores are plotted in Figure 7.11, i.e. using Equation (3.33) for SBTR and Equation (3.42) for PLS-SVD. The weights for the first 30 scores are shown. The weights obtained by PLS-SVD are almost identical and show no signs of being influenced by the high frequency component. In the case of SBTR, the weights are similar except that the weights associated for the spurious high frequency are close to zero for the first 30 output scores. This indicates that the high frequency component in the input does not covary with any of the first significant output scores. The weights only become significant with the much later output scores (not shown). However, the later output scores are normally truncated since it is the singular values associated with these predictor score components that cause instability when we invert the system. So if we included the spurious components or not the truncation of the singular values in the outputs would remove the influence of the spurious component. This negates the need for the target filtering that SBTR offers. In other words PCR would achieve the same result.



(a) SBTR. Without 100 Hz component

(b) SBTR. With 100 Hz component

(c) PLS-SVD. Without 100 Hz component

(d) PLS-SVD. With 100 Hz component.

Figure 7.11: Comparison of the first 30 β coefficients which map between the input and output scores.

7.5 Comparison against Finite Impulse Response (FIR) models

In FIR models the current output of the system is a function of past inputs such that

$$z(k) = f(u(k-1), \dots, u(k-s_w)). \tag{7.6}$$

This is in contrast to other models such as Autoregressive eXogenous (ARX) which include output feedback as well, i.e.

$$z(k) = f(u(k-1), \dots, u(k-s_w), z(k-1), \dots, z(k-s_w)). \tag{7.7}$$

In this chapter we will be focusing on non-causal inverse implementations of FIR models where the current input is a function of both past and future outputs, written as

$$u(k) = f(z(k-s_w/2), \dots, z(k+s_w/2)). \tag{7.8}$$

The non-causal form of FIR models will be used and compared against ADA since it was highlighted in Section 5.2 that using the middle section of the window for window merging produces the most accurate output reconstruction as compared to using the last or first section of the window. The middle windowing method is a non-causal implementation since it uses both future and past outputs to infer the current input. By using the FIR model the predictor matrix X takes on the form

$$X = \left[\begin{array}{cc} \overbrace{\begin{matrix} z_1(1) & z_1(2) & \cdots & z_1(s_w-1) & z_1(s_w) \\ z_1(2) & z_1(3) & \cdots & z_1(s_w) & z_1(s_w+1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_1(m-s_w-1) & z_1(m-s_w) & \cdots & z_1(m+s_w-2) & z_1(m+s_w-1) \\ z_1(m-s_w) & z_1(m-s_w+1) & \cdots & z_1(m+s_w-1) & z_1(m+s_w) \end{matrix}}^{\text{channel 1}} & \overbrace{\begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix}}^{\text{channel 2 etc.}} \end{array}\right], \tag{7.9}$$

with the corresponding target matrix Y written as

$$Y = \begin{bmatrix} u_1(s_w/2) & \cdots & u_{q-1}(s_w/2) & u_q(s_w/2) \\ u_1(s_w/2+1) & \cdots & u_{q-1}(s_w/2+1) & u_q(s_w/2+1) \\ \vdots & \ddots & \vdots & \vdots \\ u_1(m-s_w/2-1) & \cdots & u_{q-1}(m-s_w/2-1) & u_q(m-s_w/2-1) \\ u_1(m-s_w/2) & \cdots & u_{q-1}(m-s_w/2) & u_q(m-s_w/2) \end{bmatrix}. \tag{7.10}$$

It is worth noting that we lose the first and last $s_w/2$ samples of the target matrix Y since we shifted the inputs to make the system non-causal.

The lack of feedback means that the FIR is inherently stable. This is suitable and sometimes sought after if the system under consideration is stable, however if the system is unstable it will only be able to approximate the instability for a short period of time before diverging [32]. FIR models come with the cost of needing significantly more terms than what output feedback models need to map the same system [32]. In Equation (5.23) we saw that the FIR model can be structured with predefined weights. A more refined method of achieving this is through the use of Orthonormal Basis Functions (OBF) [32, 24]. These functions weight each input using filters that are designed using prior knowledge of the system. Laguerre Filters can be implemented if the system is lightly damped and an estimate of the dominant pole is known, whereas Kautz Filters are implemented if the system is highly oscillatory and an estimate of a complex pair of poles are known. More generalized filters can be implemented by combining the two types of filters. These filters have the benefit of reducing the number of terms needed to model the system with FIR if a rough estimate of the system dynamics is known. The use of OBF significantly reduces the number of terms that need to be incorporated. The use of OBF will not be explored further in this dissertation.

7.5.1 Numerical Investigation

The goal of this section is to benchmark ADA against FIR in terms of response reconstruction since ADA can be seen as a subset of FIR. The idea behind this benchmark is to ensure that ADA is not an indirect method of achieving an FIR implementation. If so, it needs to be determined whether ADA offers any substantial benefits over using FIR. In Section 5.3 we saw that ADA acts in such a way that it weights the coefficients of the outputs closest to the input in time and de-emphasizes the weights further away in time. A similar approach can be achieved through the use of FIR models combined with Tikhonov regularisation. By using Tikhonov regularisation the β coefficients can be penalized and thus shaped by the choice of the Γ matrix in Equation (3.21). To this end 3 choices of the Γ matrix will be implemented, namely: Finite Impulse Response with Triangular Weighting (FIR-T), Finite Impulse Response with Difference Smoothing and Triangular Weighting (FIR-DT) and Finite Impulse Response with Ridge Regression (FIR-RR).

In FIR-T the coefficients relating to the outputs further away from the required input (both forwards and backwards in time) are penalised. This is achieved by setting

$$\Gamma'\Gamma = \alpha W \tag{7.11}$$

where W is an inverted triangular set of penalty weights, given as

$$W = diag\left(\begin{bmatrix} s_{\mathrm{w}}/2 & s_{\mathrm{w}}/2-1 & \cdots & 2 & 1 & 2 & \cdots & s_{\mathrm{w}}/2-1 & s_{\mathrm{w}}/2 \end{bmatrix}\right). \tag{7.12}$$

and α scales the amount of regularisation we wish to impose. This should ideally mimic the weighting function achieved by ADA in Equation (5.24). FIR-DT further modifies the triangular weighting matrix through the use of a first difference matrix A, given as

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix}. \tag{7.13}$$

The first difference matrix ensures that the difference between each successive β coefficient is small [13]. The difference matrix is then combined with the weighting matrix W to obtain the final form of the regularisation matrix such that

$$\Gamma'\Gamma = \alpha A'WA. \tag{7.14}$$

This weighting scheme was originally implemented and developed for a causal FIR system where the penalty weights increased linearly further back in time [13]. Finally, the last choice of penalty matrix Γ is that of FIR-RR, i.e.

$$\Gamma = \alpha I \tag{7.15}$$

which is the case where we do not assume any form of β except that we wish to limit the magnitude of the weights. This acts as a reference in order to determine whether the regularisation in terms of the form of the β does in fact contribute to the accuracy of the response reconstruction.

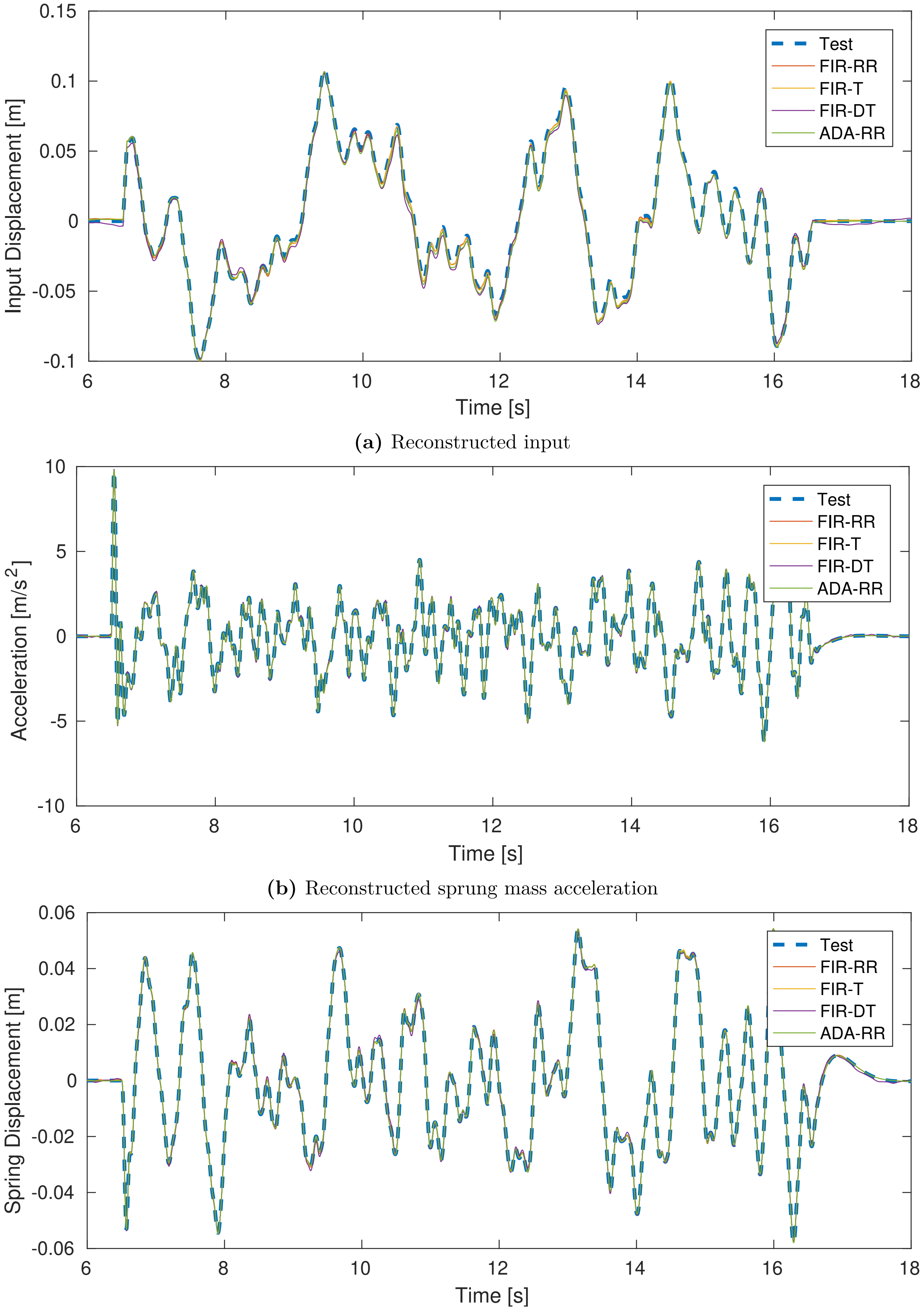
7.5.2 Procedure

The three different regularised FIR implementations will be compared against the ADA combined with RR for two different test cases. The first being the linear system and the second being the default nonlinear system. The inputs and responses will be sampled at 250 Hz and 350 Hz for the linear and nonlinear system respectively. The window lengths will be determined via grid search cross validation with the window lengths being sampled from $T_w \in [0.1, 12]$ s with a grid of 50 equally spaced intervals. The window search is made more extensive in this investigation since no previous window parameter search were performed on the FIR methods. An overview of the numerical experiment parameters is given in Table 7.11.

7.5.3 Results

The reconstructed inputs and outputs for the linear and nonlinear systems are shown in Figures 7.12 and 7.13 respectively.

The response reconstruction results for the linear and nonlinear systems are shown in Table 7.12. For the linear case we note that ADA outperforms the FIR methods in terms of response reconstruction while achieving a similar input reconstruction score. The results achieved by FIR-RR and FIR-T are similar whereas FIR-DT performs the worst.

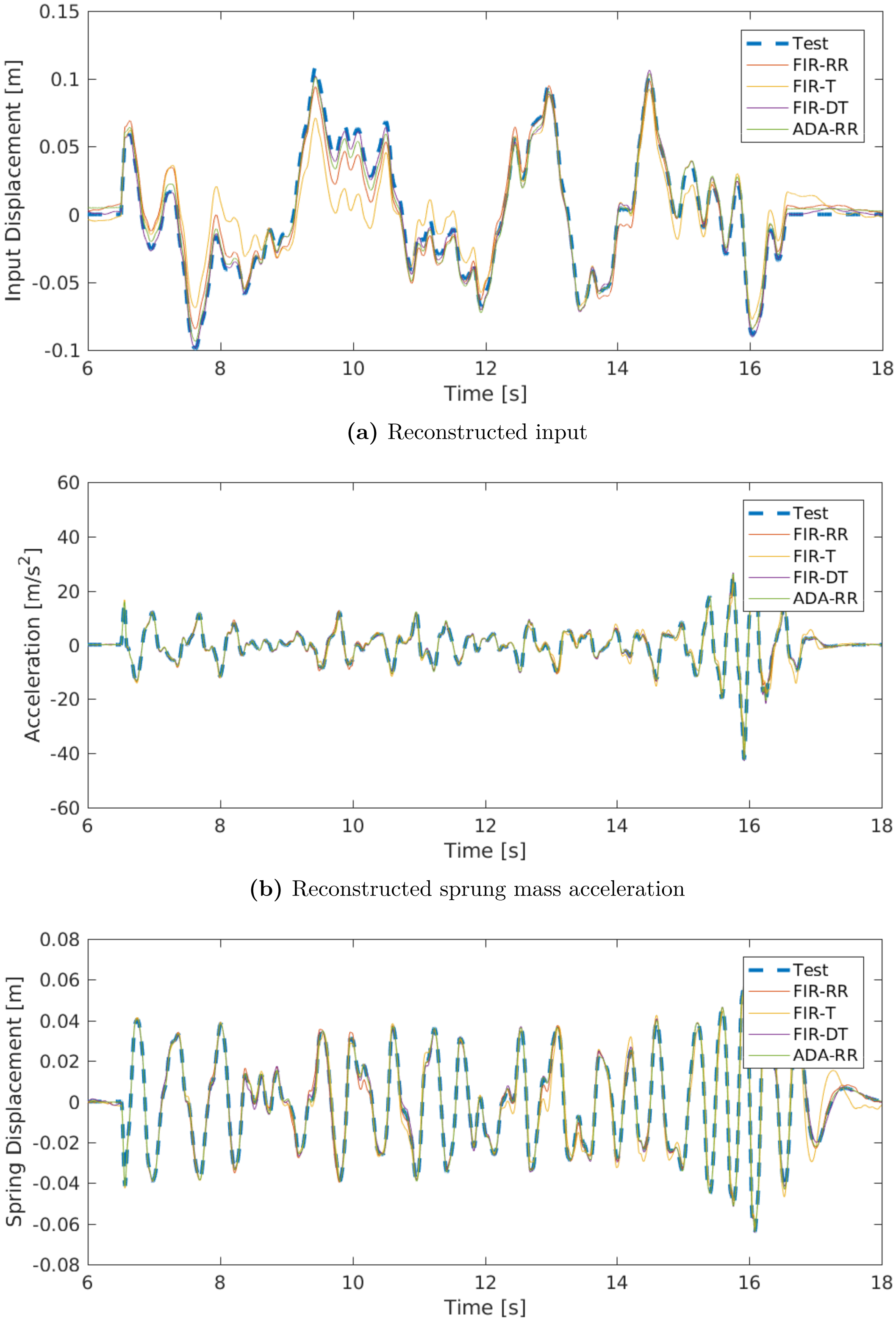


(a) Reconstructed input

(b) Reconstructed sprung mass acceleration

(c) Reconstructed spring displacement

Figure 7.12: Comparison of recreated input and output results using FIR methods against ADA. Linear System.



(a) Reconstructed input

(b) Reconstructed sprung mass acceleration

(c) Reconstructed spring displacement

Figure 7.13: Comparison of recreated input and output results using FIR methods against ADA. Nonlinear System.

Table 7.11: Experimental design benchmarking ADA against different forms of FIR models. (Variables of interest shown first).

Variable	Details
Reg. method	FIR-T, FIR-DT, FIR-RR and ADA-RR
$\mathbf{k_{NL}}$	Linear: 0, Nonlinear: 1.28×10^{7} N m^{-3}
Sensor config.	Sprung mass acceleration + spring displacement
Window length $\mathbf{T_w}$	$\in [0.1, 12]$ s with a grid of 50 equally spaced intervals
Sampling frequency $\mathbf{f_s}$	Linear: 250 Hz, Nonlinear: 350 Hz
Window proportional overlap γ	Maximum
Ridge regression regularisation constant α	$\in [10^{-16}, 10^{5}]$ with 30 divisions spaced logarithmically
QC parameters	Default values; Table 2.1
Noise level $\eta_{\%}$	0 %
Train. Set	APRBS; Table 5.1
Val. Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

For the nonlinear case the results obtained by FIR-DT are significantly better than those obtained by FIR-RR and FIR-T which indicates that the difference smoothing may be a suitable fit for the nonlinear system. For the nonlinear case the triangular weighting performed poorly overall. The output responses offered by ADA significantly outperform the results obtained by the FIR implementations. The optimized hyper-parameters for the numerical experiment are shown in Table 7.13.

7.5.4 Illustrative Use Case

In this section, we will create a scenario whereby all the challenges to response reconstruction, and which are covered in this dissertation, are introduced. These are noise, model mismatch and nonlinearity. The investigation is not exhaustive but rather proposed to give an illustrative sense of the performance of the regression methods on a challenging response reconstruction problem. To this the end, the numerical experiment will be performed with the FIR and ADA regression methods with noise $\eta_{\%}$ set to 5 % , model mismatch $m_{\%}$ set to 10 % and the non-linearity term k_{NL} set to 1.28×10^{7} N m^{-3}. In the case of model mismatch, the validation response set will come from a “real world” recording as opposed to a laboratory recording. An overview of the numerical procedure is given in Table 7.14.

Results

The response reconstruction results are shown in Table 7.15 with the corresponding reconstructed inputs and outputs shown in Figure 7.14. By referring to the results in Table 7.15 we see that ADA and FIR-DT perform similarly well for the reconstructed test results. These results are achieved within close enough margin to each other that it probably falls within the uncertainty introduced by noise. We see that FIR-T performs poorly for the problem at hand. If we compare the results to the benchmark regression method, FIR-RR, we note that the imposed structure on the β parameters and the smoothing that both FIR-DT and ADA offer could be key factors

Table 7.12: MFFE scores for the approximated input and output signals using different FIR methods. The discrepancy between input and output reconstruction accuracies is due to functional reproducibility, as discussed in Section 2.3.

	Training			**Validation**			**Test**		
	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta}_{\mathbf{z}}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta}_{\mathbf{z}}$
Linear Case									
FIR-RR	12.40	1.26	0.26	12.71	1.63	0.35	3.50	0.91	1.13
FIR-T	12.33	1.38	0.34	12.41	0.46	0.25	3.80	0.90	1.15
FIR-DT	10.57	4.39	2.01	17.98	1.48	1.81	6.98	3.08	3.16
ADA-RR	11.14	0.71	0.16	10.48	0.38	0.23	5.49	0.16	0.35
Nonlinear Case									
FIR-RR	25.66	11.33	1.99	43.72	5.03	2.82	20.76	8.13	7.25
FIR-T	38.86	5.92	4.32	59.83	2.34	1.60	38.90	17.56	13.44
FIR-DT	5.22	8.54	4.70	35.94	4.71	3.60	7.50	5.41	4.89
ADA-RR	5.91	0.55	0.13	36.60	0.21	0.16	12.55	0.39	0.40

Table 7.13: Optimized hyper-parameter results for the numerical demonstration using different FIR methods for an illustrative use case.

	α	$\mathbf{T}_{\mathbf{w}}[\mathbf{s}]$
Linear Case		
FIR-RR	11.80	12.00
FIR-T	4.74×10^{-2}	11.51
FIR-DT	1.04×10^{-9}	8.11
ADA-RR	8.52×10^{-3}	10.54
Nonlinear Case		
FIR-RR	4.27	3.500
FIR-T	4.80×10^{-1}	4.471
FIR-DT	4.79×10^{-2}	7.629
ADA-RR	8.52×10^{-3}	10.543

Table 7.14: Experimental design comparing ADA against different forms of FIR models for an illustrative use case. (Variables of interest shown first).

Variable	Details
Regression method	FIR-T, FIR-DT, FIR-RR and ADA-RR
Nonlinearity constant $\mathbf{k_{NL}}$	Nonlinear: 1.28×10^7 N m^{-3}
Model mismatch	Perturbed as per Equations (7.1) to (7.5) with $m_{\%} = 10$
Noise level $\eta_{\%}$	5 %
Sensor configuration	Sprung mass acceleration + spring displacement
Window lengths $\mathbf{T_w}$	$\in [0.1, 12]$ s with a grid of 50 equally spaced intervals
Sampling frequency $\mathbf{f_s}$	350 Hz
Window proportional overlap γ	Maximum
Ridge regression regularisation constant α	$\in [10^{-16}, 10^5]$ with 30 divisions spaced logarithmically
QC parameters	Default values; Table 2.1
Training Set	APRBS; Table 5.1
Validation Set	APRBS; Table 5.1
Test Set	Road profile; Table 5.2

to their better performances.

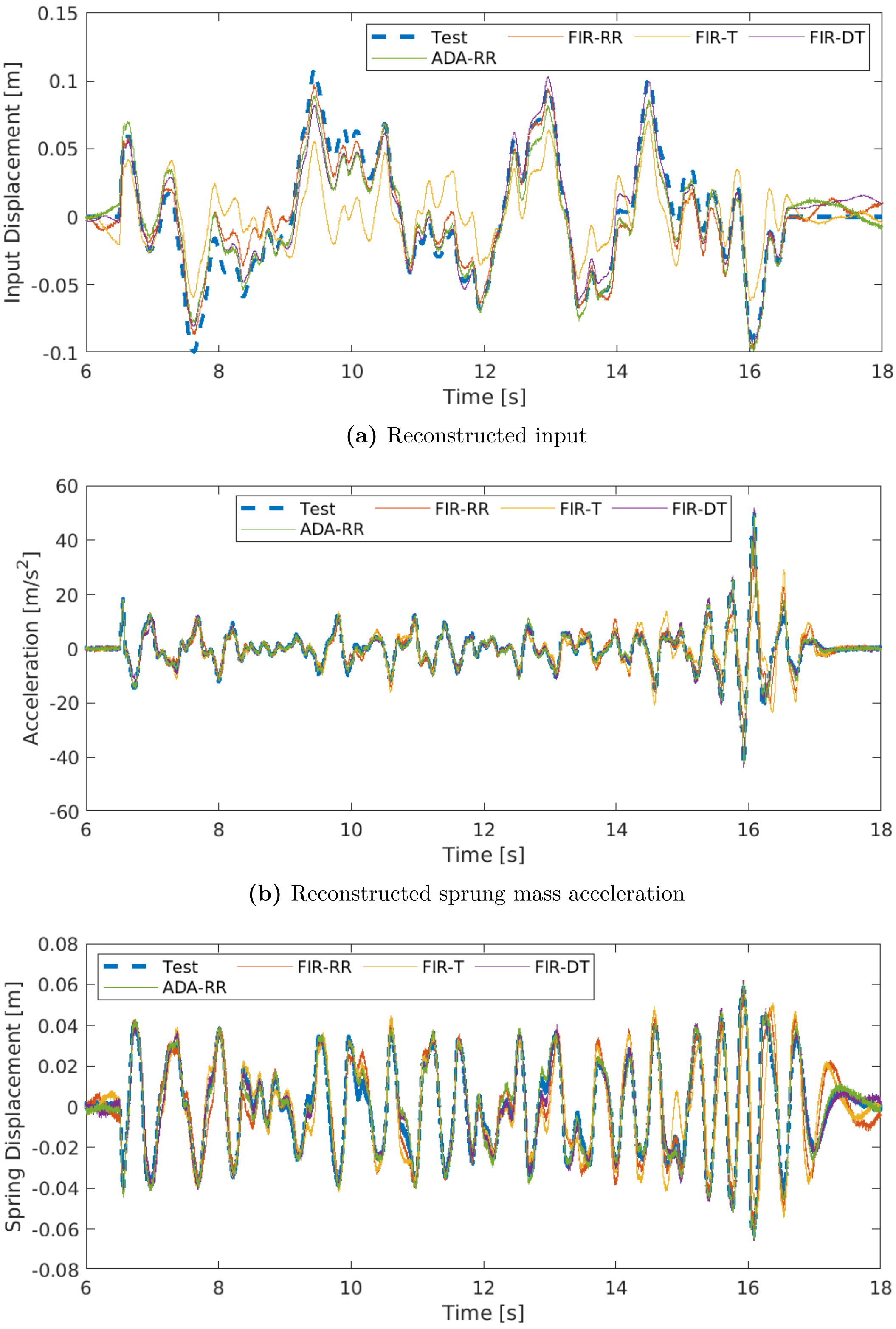
Table 7.15: MFFE [%] scores for the approximated input and output signals using different FIR methods for an illustrative use case.

	Training			**Validation**			**Test**		
	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta_z}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta_z}$	$\mathbf{u}_{\text{road}}$	$\ddot{\mathbf{z}}_{\mathbf{A}}$	$\mathbf{\Delta_z}$
FIR-RR	15.37	8.78	6.26	46.52	16.26	12.66	24.45	31.56	25.17
FIR-T	56.34	16.12	6.71	62.47	13.79	11.62	71.46	46.49	35.09
FIR-DT	25.72	6.59	11.84	46.65	14.91	11.58	22.20	15.53	14.41
ADA-RR	18.82	6.64	4.68	29.90	16.01	13.85	24.69	14.75	14.23

The hyper-parameters that were optimized for this numerical experiment are shown in Table 7.16. Here we note that the different regression methods use similar window lengths, save for FIR-T which used a significantly shorter window length. Here we also note that the regularization constants α are different orders of magnitude between the various regression methods.

7.5.5 Conclusion

In this section, we compared the performance of ADA to related FIR regression methods. Although the experiments were not exhaustive, the results did indicate that the performance of ADA is better than the related FIR methods in terms of response reconstruction accuracy. By repeating the experiment with challenges that require better regularisation we gained insights



(a) Reconstructed input

(b) Reconstructed sprung mass acceleration

(c) Reconstructed spring displacement

Figure 7.14: Comparison of recreated input and output results using FIR methods against ADA for an illustrative use case.

Table 7.16: Optimized hyper-parameter results for the numerical demonstration using different FIR methods for an illustrative use case.

	α	$\mathbf{T_w[s]}$
FIR-RR	0.64	8.03
FIR-T	7.23	2.08
FIR-DT	469.42	9.52
ADA-RR	6.81×10^{-5}	7.04

into how ADA may be performing regularisation. It may be implicitly imposing a shape and smoothness penalty on the β regression coefficients.

Chapter 8

Conclusion

The aim of this dissertation was to investigate the use of linear regression techniques in the problem of response reconstruction. The method was broken down into two parts: the representation of the time signals and the regression method used to map between the output and input signals.

In terms of the representation of the time signals, it was highlighted in Section 1.7 that the non-overlapping windowing method can only represent sensor set-ups that require differentiation in order to obtain the internal states of the system. This is a substantial shortcoming. The sensors most commonly used in response reconstruction, such as strain gauges and accelerometers, require integration to implicitly estimate the states of the system. To remedy this limitation, overlapping windows were introduced and investigated in Sections 5.2 and 5.3. A windowing method developed through the course of this dissertation that averaged the predictions over multiple overlapping predictions, called ADA, was found to produce the most accurate response reconstructions. The introduction of overlapping windows greatly increased the computational complexity of the problem as demonstrated in Section 5.4, however, methods such as varying the stride and choosing a more suitable sampling frequency were shown to mitigate this computational overhead.

The ADA method was then compared against closely related FIR models in Section 7.5 and was shown to produce significantly better response reconstructions. Methods that could provide rough estimates of the parameters needed to define the ADA implementation such as sampling frequency and window length were explored in Sections 6.1 and 6.2.

Regarding the regression methods employed to map between the outputs and the inputs, four related regression methods were compared, namely: SBTR, PCR, RR and PLS-SVD. The regression methods were benchmarked using ADA in terms of handling nonlinearities of the system, robustness to noise and robustness to model mismatch. In all 3 test cases the difference in terms of response reconstruction were insignificant. The exception to this was PLS-SVD, which was found to require a larger number of latent variables and was negatively affected by the introduction of measurement noise. An explanation as to why SBTR performed no better or worse than PCR or RR was given in Section 7.4. The only feature that separates SBTR from PCR, namely its target filtering, can be easily swayed by spurious signals. The target filtering also introduces significant computational cost that RR or PCR do not incur. The default recommendation would be to use RR with ADA.

In summary it is shown that ADA combined with an appropriate linear regression is a suitable black-box method of reconstructing responses in dynamic systems. It has wider application in response reconstruction in that it can be readily applied to non-linear systems, a broader range of sensor configuration and non-minimum phase systems.

8.1 Future Work

The concepts of model mismatch and measurement noise are covered in this dissertation. An interesting proposal, which lends itself to a more experimental approach, would be to quantify the contributions of each to the overall noise of the system.

In this dissertation, a narrow view of model mismatch is treated. In this view of model mismatch, the mismatch is seen as scaled versions of the model parameters. A more in-depth investigation should be performed on model mismatch whereby the class of the dynamics is changed from the real-world environment to that of the laboratory environment. For example, certain degrees-of-freedom may be introduced or removed when moving to the laboratory environment. Likewise in the studies of nonlinearity of the system, the non-linearities are simply scaled up. More insights could be gained if instead different classes of nonlinearities are introduced such as discontinuities or saturations.

The current implementation of ADA can be seen as a post-processing smoothing step that occurs after a linear regression prediction. An interesting avenue to explore would be to replace the linear regression with a non-linear regression method such as a neural network.

In this dissertation the means of reducing the computational complexity of ADA such as downsampling and using differing strides have been done independently of one another. For convenience sake, the stride was set to 1 for most of the numerical experiments with ADA. It would be worthwhile to determine a more rigorous approach to choosing the stride a priori as is the case with sampling frequency and window length.

ADA has been tested on a single numerical model and has shown promising results but to ensure that it has wide applicability, ADA needs to be tested on a variety of inverse problems. Furthermore, ADA needs to be tested on a physical testbed to understand its practical strengths and shortcomings.

Bibliography

[1] M. S. Allen and T. G. Carne. Delayed, multi-step inverse structural filter for robust force identification. *Mechanical Systems and Signal Processing*, 22(5):1036–1054, 2008.

[2] E. Asaadi, D. N. Wilke, P. S. Heyns, and S. Kok. The use of direct inverse maps to solve material identification problems: pitfalls and solutions. *Structural and multidisciplinary optimization*, 55(2):613–632, 2017.

[3] M. Aucejo and O. De Smet. On a full bayesian inference for force reconstruction problems. *Mechanical Systems and Signal Processing*, 104:36 – 59, 2018.

[4] S. Billings and L. A. Aguirre. Effects of the sampling time on the dynamics and identification of nonlinear models. *International journal of Bifurcation and Chaos*, 5(06):1541–1556, 1995.

[5] S. A. Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains.* John Wiley & Sons, 2013.

[6] C. Bishop. Springerlink. *Pattern recognition and machine learning*, 4, 2006.

[7] F. L. Bookstein, P. D. Sampson, A. P. Streissguth, and H. M. Barr. Exploiting redundant measurement of dose and developmental outcome: New methods from the behavioral teratology of alcohol. *Developmental Psychology*, 32(3):404, 1996.

[8] R. L. Burden and J. D. Faires. Numerical analysis. *PWS, Boston*, 1993.

[9] C. R. Cater. Advances in dynamic response reconstruction using non-linear time domain system identification. Master's thesis, 1997.

[10] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, pages 808–817, 2000.

[11] J. Crous, S. Kok, D. Wilke, and P. Heyns. A rotation-scaling based approach to multi-target regression in high dimensional spaces as an alternative to the standard projection based approach. 2016. Unpublished.

[12] J. Crous, D. N. Wilke, S. Kok, D.-G. D. Chen, and S. Heyns. On system identification for accelerated destructive degradation testing of nonlinear dynamic systems. In *Statistical Modeling for Degradation Data*, pages 335–364. Springer, 2017.

[13] B. S. Dayal and J. F. MacGregor. Identification of finite impulse response models: methods and robustness issues. *Industrial & engineering chemistry research*, 35(11):4078–4090, 1996.

[14] S. De Jong. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.

[15] M. Deflorian, F. Klöpper, and J. Rückert. Online dynamic black box modelling and adaptive experiment design in combustion engine calibration. *IFAC Proceedings Volumes*, 43(7):703–708, 2010.

[16] M. Deflorian and S. Zaglauer. Design of experiments for nonlinear dynamic system identification. *IFAC Proceedings Volumes*, 44(1):13179–13184, 2011.

[17] J. J. A. Eksteen. *Advances in iterative learning control with application to structural dynamic response reconstruction.* PhD thesis, University of Pretoria, 2014.

[18] J. B. Elsner and A. A. Tsonis. *Singular spectrum analysis: a new tool in time series analysis.*

Springer Science & Business Media, 2013.

[19] I. O. for Standardization. *ISO 8608 : Mechanical Vibration, Road Surface Profiles, Reporting of Measured Data: International Standard.* International Organization for Standardization, ISO, 1995.

[20] M. French. An introduction to road simulation testing. *Experimental techniques*, 24(3):37–38, 2000.

[21] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[22] P. Geladi and B. R. Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[23] H. Hassani. Singular spectrum analysis: methodology and comparison. 2007.

[24] P. S. Heuberger, P. M. van den Hof, and B. Wahlberg. *Modelling and identification with rational orthogonal basis functions.* Springer Science & Business Media, 2005.

[25] P. Johannesson and I. Rychlik. Modelling of road profiles using roughness indicators. *International Journal of Vehicle Design*, 66(4):317–346, 2014.

[26] M. S. Kumar and S. Vijayarangan. Analytical and experimental studies on fatigue life prediction of steel and composite multi-leaf spring for light passenger vehicles using life data analysis. *Materials science*, 13(2):141–146, 2007.

[27] Q. Li and Q. Lu. A hierarchical bayesian method for vibration-based time domain force reconstruction problems. *Journal of Sound and Vibration*, 421:190 – 204, 2018.

[28] L. Ljung. *System identification: theory for the user.* PTR Prentice Hall, Upper Saddle River, NJ, 1999.

[29] P. Mainçon and C. Barnardo-Viljoen. An inverse finite element method for the analysis of viv data. *Marine Structures*, 33:143–159, 2013.

[30] Y. Miyashita, T. Itozawa, H. Katsumi, and S.-I. Sasaki. Comments on the nipals algorithm. *Journal of chemometrics*, 4(1):97–100, 1990.

[31] P. Moylan. Stable inversion of linear systems. *IEEE Transactions on Automatic Control*, 22(1):74–78, 1977.

[32] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models.* Springer Science & Business Media, 2013.

[33] J. P. Noël and G. Kerschen. Nonlinear system identification in structural dynamics: 10 more years of progress. *Mechanical Systems and Signal Processing*, 83:2–35, 2017.

[34] A. D. Raath. *Structural dynamic response reconstruction in the time domain.* PhD thesis, 1993.

[35] K. K. Stevens. Force identification problems—an overview. In *Proceedings of the 1987 SEM Spring Conference on Experimental Mechanics*, pages 14–19. Houston, TX, 1987.

[36] A. Tishler and S. Lipovetsky. Modelling and forecasting with robust canonical analysis: method and application. *Computers & Operations Research*, 27(3):217–232, 2000.

[37] T. Uhl. The inverse identification problem and its technical application. *Archive of Applied Mechanics*, 77(5):325–337, 2007.

[38] J. A. Wegelin. A survey of partial least squares (pls) methods, with emphasis on the two-block case. *University of Washington, Tech. Rep*, 2000.