University of Windsor

Scholarship at UWindsor

7-11-2015

Road profile recovery using vertical acceleration data

Fei Hu
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

ROAD PROFILE RECOVERY USING VERTICAL ACCELERATION DATA

by
Fei Hu

A Thesis
Submitted to the Faculty of Graduate Studies
through Mechanical Engineering
in Partial Fulfilment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

2015

Road Profile Recovery Using Vertical Acceleration Data

by

Fei Hu

APPROVED BY

_____

Dr. S. Cheng
Department of Civil and Environmental Engineering

_____

Dr. C. Novak
Department of Mechanical, Automotive, & Materials Engineering

_____

Dr. B. Minaker, Advisor
Department of Mechanical, Automotive, & Materials Engineering

May 4, 2015

Author's Declaration of Originality

I hereby declare that the thesis submitted is my own original work. All direct or indirect materials used are stated as references.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I am aware of the University of Windsor Senate Policy on Authorship and I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

For the purpose of studying vehicle stability and suspension performance, road profiles are an essential element that should be considered. Various methods have been developed to measure the road surface. This thesis aims to develop a fast, computationally efficient, and economical method to obtain road profile data only using accelerometers. Two sensors are mounted on a vehicle to measure both wheel and body accelerations in the vertical direction. This approach is based on a dynamic model with an output tracking feedback control. Virtual models are built in both MATLAB® and Altair MotionView®. Four randomly generated ISO class road profiles and one measured road profile are used in this research. Simulation results obtained from the models show that this method has great potential, and can usually recover road inputs with high accuracy using both linear and non-linear properties.

To my parents and my younger brother, for their marvelous love and support.

Acknowledgements

Completing this thesis, and indeed my entire degree has not always been an easy journey. I would like to thank the people who have helped to make this completion possible.

First of all, I would like to thank my supervisor Dr. Bruce Minaker for his guidance and patience. He has not only provided countless advice, but also used his extensive knowledge to assist me to complete this research. Working with Dr. Minaker was a comfortable and educational experience.

This research was supported and funded by Chrysler Canada Inc., the Connect Canada program and the Ontario Centres of Excellence. Without the physical and financial support provided by these organizations, the achievements in this research would be impossible.

I would also like to thank Chrysler engineers Mohammed Malik and Mehrzad Abdollahi, for their supervision on the project and their advice, support and motivation throughout my research.

Moreover, I would like to thank my parents, brother and Ya, for their love, advice and always being so supportive. Last, I take this opportunity to thank all my dear colleagues and friends, for the joyful time with you along this journey. Without this friendship, I would not fulfil my degree and goals.

Contents

List of Tables

List of Figures

List of Abbreviations

All abbreviations used in this work are described in this section.

Abbreviation	Meaning
ARDC	Automotive Research and Development Center
CAE	Computer Aid Engineering
DAE	Differential Algebraic Equation
DOF	Degrees of Freedom
FHWA	Federal Highway Administration
GA	Genetic Algorithm
GMR	General Motor Research
GUI	Graphical User Interface
IRI	International Roughness Index
ISO	International Standards Organization
LPA	Longitudinal Profile Analyser
LTI	Linear Time-Invariant
NM	Nelder-Mead Method
ODE	Ordinary Differential Equation
PID	Proportional-Integral-Derivative
PSD	Power Spectral Density
TF	Transfer Function
WHO	World Health Organization
WNN	Wavelet Neural Network

Nomenclature

The mathematical notation used throughout this work is listed below, separated by the chapter in which it first appears.

Background Review

Label	Description
A	Road amplitude
f	Road excitation frequency in the time domain
G_d	PSD of vertical displacement
L	Wavelength
n	Spatial frequency in cycles/m
n_0	Conventional spatial frequency
s	Road path variable
t	Time
v_0	Constant velocity
z_1	Disturbance
ω	Radian frequency
Ω	Angular spatial frequency
Ω_0	Conventional angular spatial frequency

Overview of Vehicle Dynamics and Control

Label	Description
α, β	Real and imaginary parts of the complex-conjugate pole
$\boldsymbol{u}$	Input signal
$\boldsymbol{z}, \boldsymbol{x}$	State vector
ω	Angular velocity
ω_d	Damped natural frequency
ω_n	Natural frequency
ψ	Yaw angle
τ	Time constant
$\mathbf{M}$	Mass matrix
θ	Pitch angle
φ	Roll angle
ζ	Damping ratio
${}^G R_B$	Rotation matrix between the global coordinate G and body coordinate B
a_0 to a_n	Coefficients of the denominator of the transfer function
a_1 and a_2	Distances of mass center C from front and rear axles
b_0 to b_m	Coefficients of the numerator of the transfer function
b_1 and b_2	Distances of mass center C from the left and right wheels
c_1, c_f	Front suspension damping
c_2, c_r	Rear suspension damping
c_s	Suspension damping coefficient
d	Distance between the locations of the two roots and the origin of the coordinate system in the complex plane
e	Error between the set point and the measured value of the output of a system
F	Force
I_G	Inertia dyadic
I_x	Longitudinal mass moment of inertial
I_y	Lateral mass moment of inertial
k_1, k_f	Front spring stiffness
k_2, k_r	Rear spring stiffness
k_d	Derivative term coefficient
k_i	Integral term coefficient

k_p	Proportional term coefficient
k_R	Torsional stiffness of an anti-roll bar in front or rear
k_s	Suspension spring stiffness
k_t	Tire stiffness
k_{t1}, k_{tf}	Front tires stiffness
k_{t2}, k_{tr}	Rear tires stiffness
M	Torque
m	Body mass
m_1 to m_4	Four wheels masses
m_s	Sprung mass
m_u	Unsprung mass
p	Roll rate, Body position
q	Pitch rate
r	Yaw rate
s	Coordinate frame spin
$s_{1,2}$	Two pairs of complex-conjugate poles
t	Time
v	Body velocity
W	Vehicle track width
w	Body velocity vector
x	Body vertical displacement
x_1 to x_4	Four wheels vertical displacements
y_1 to y_4	Road excitations at four wheels
z_0	Displacement of the road surface
z_1	Position of the sprung mass
z_2	Position of the unsprung mass
C	Damping matrix
F	Force vector
M	Mass matrix
B(Cxyz)	Vehicle body coordinate frame
C	Mass center
G(Oxyz)	Global coordinate frame
P	Point vector
x, y, z	Coordinates of the vehicle body frame

Vehicle Modeling

Label	Description
e	Error between measured accelerations and accelerations from the output
r	Measured accelerations
y	Accelerations from the output, Output vector
N	Numbers of dimension
$\mathbb{A}$	State matrix
$\mathbb{B}$	Input matrix
$\mathbb{C}$	Output matrix
$\mathbb{D}$	Direct transmission matrix
$\mathbf{G}$	Controller gain based on allowed convergence time
$\mathbf{H}$	Controller gain based on quality of fit
n	the number of state variables
m	the number of input variables
l	the number of output variables

Model Implementation

Label	Description
Δ	Penalty function
ω_{body}	Body natural frequency
ω_{wheel}	Wheel natural frequency
f_{desired}	Function of desired parameter
$f_{\text{simulated}}$	Function of simulated parameter

Chapter 1

Introduction

Recently, the dynamic response of a vehicle to the road profile has been studied by many researchers. However, it is often difficult to obtain this dynamic response because the road inputs are not well known. This thesis is an attempt to address this problem by providing a fast and effective means of recovering these inputs.

Three sections are included in this chapter. First of all, the motivation for this research is presented. Secondly, the research objectives are stated. Finally, the rest of this thesis is described in the third section.

1.1   Motivation

A review of accident data reveals that every year, the main reason for most vehicle accidents is loss of vehicle control. According to the World Health Organization (WHO), in 2009 about 2.2% of the vehicle collision death rate is caused by the road profile, because road profile roughness is the main source of vehicle vibration and material fatigue. The road profile is one of the most important parameters that should be considered in order to study vehicle stability and suspension performance, such as ride comfort or structural durability. Generally, when a vehicle is driven over bumps or potholes or other rough terrain, the suspension components can experience large forces and accelerations, which can potentially lead to premature wear or failure of the components. Accordingly, the level of stress on the suspension is highly influenced by the type of road, and the size and number of disturbances. Therefore, it is essential to develop methods to accurately measure road profiles.

This thesis project is a decoupled project between the University of Windsor Vehicle Dynamics and Control Research Group and the Chrysler Canada Automotive Research and Development Centre (ARDC), with the intention of improving overall vehicle quality, particularly durability of the suspension compo-

nents. Currently, the design of the suspension is based on some expectations of the road quality, which is often challenging to gauge accurately. Typically, an expensive and highly specialized instrumented vehicle is driven over a special testing ground to obtain some measurements related to the road and the vehicle responses. Upon the request from Chrysler, a simple and fast road profile data collection method needs to be developed to replace this approach, so that the number and severity of component failures can be reduced.

1.2 Research Objectives

The first objective of this research is to develop and assess computational algorithms to recover the road profile using vertical spindle/body acceleration time history data measured by a small number of convenient on-board accelerometers. Accelerometers are small, cost-efficient sensors commonly used to measure vibrations, typically based on the properties of piezo-electric materials. They are increasingly commonly found in a wide range of equipment, even smart-phones. However, the process of using the measured data from accelerometers to determine the road profile is not a simple one.

The second objective is to demonstrate the viability of the recovery algorithm using a wide variety of vehicles and the road profile history on any road. A dynamic model (quarter car, half car, or full car) will be coded and tested in MATLAB®/Octave, and complex multi-body dynamic models with non-linear parameters will be simulated in Altair MotionView®.

The third and the last objective is that this method can be coded as a software tool to help assess the design of suspension components and vehicle durability, which will in turn reduce the number and severity of component failures, leading to improvements in overall quality and marketplace competitiveness.

1.3 Thesis Structure

The thesis presents a fast, computationally efficient and economical approach to recover the effective road profile. In this section, brief descriptions of each chapter will be given, to help in understanding the structure of this thesis.

The first chapter introduces the research topic and the motivation and objectives behind this topic. The outline of each chapter is also described.

Chapter 2 is the background review of the literature required to achieve the goals of the research. The definition and classification of road profiles will be described. A literature review that shows current

methods is also given, since much research has been done involving road profile estimation. However, some limitations may present in these existing methods.

Chapter 3 is a simple review on vehicle dynamics and control. This chapter shows some fundamental concepts necessary to derive the equations of motion of the dynamic models (quarter car, half car, full car, etc.). Theory regarding control systems is then discussed. Last, frequency response of dynamic systems is reviewed.

Chapter 4 describes the main idea to recover road profiles using only measured acceleration data. A recovery algorithm is developed in this chapter. The proposed recovery algorithm is mainly using a feedback control approach to recover the road profiles.

Chapter 5 can be divided into two sections. The first section is the linear dynamic recovery model implementation in MATLAB®. The process of road profile recovery is completed by programming via MATLAB® codes. The second section introduces four sets of multibody models built in Altair MotionView®, including non-linear models.

Chapter 6 presents the simulation results of both the MATLAB® and Altair MotionView® models. This chapter aims at recovering the road profiles using Altair MotionView® models. These multibody models are simulated with the recovery algorithm running in parallel to recover simulated road profiles. Additionally, a simulation result based on the real measured acceleration data is given in this chapter.

The last chapter summarizes the work that has been done in this research. Recommendations and suggested future work are also included.

Chapter 2

Background Review

2.1 Definition of The Road Profile

A profile of a road is an intrinsic part of the interaction between the pavement and a vehicle suspension, and it influences ride, handling, fatigue, fuel consumption, tire wear, maintenance costs, and vehicle delay costs. It is a two-dimensional cross-sectional shape of the road surface, taken along an imaginary line for the purpose of ride quality analysis. As shown in Figure 2.1, a profile consists of longitudinal and lateral profiles. Profiles taken along the lateral direction show the superelevation and crown of the road design [1], while this research focuses on the longitudinal profile, as it shows the design grade, roughness and texture, and is the primary source of vehicle vibrations.
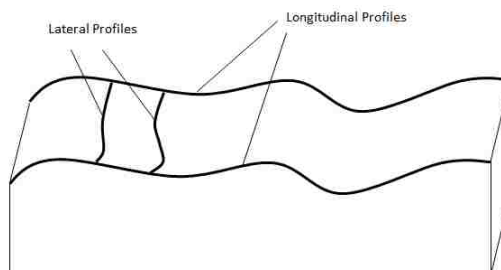
Figure 2.1: Road profile

In physical testing and simulations, the road profile can be classified such as: isolated bump, sinusoidal, fixed waveform, and stochastic roads [2]. An isolated bump is a nominally smooth basic road with a local disturbance, which results in a solitary wave. It includes ramps or true bumps. A ramp is caused by a disturbance with a residual change of road height, while a true bump returns to its original elevation after a disturbance. Figure 2.2 compares the difference between a ramp and a true bump.

(a) Ramp

(b) Bump
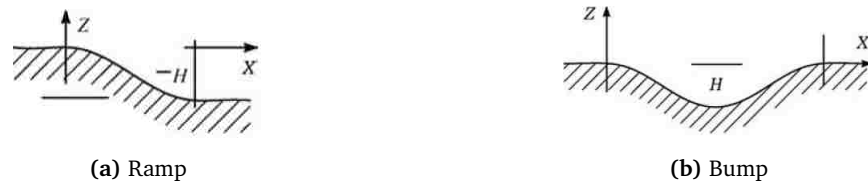
Figure 2.2: Isolated bump (reproduced from Dixon [2])
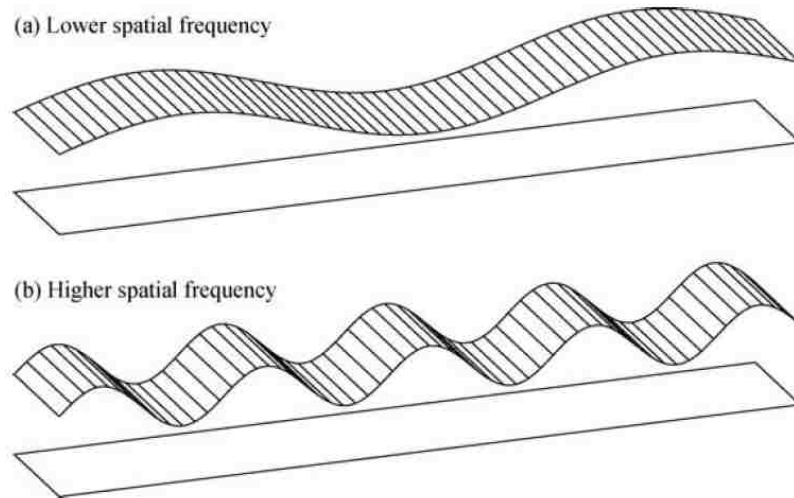
Figure 2.3: Sinusoidal single paths of lower and higher spatial frequencies (reproduced from Dixon [2])
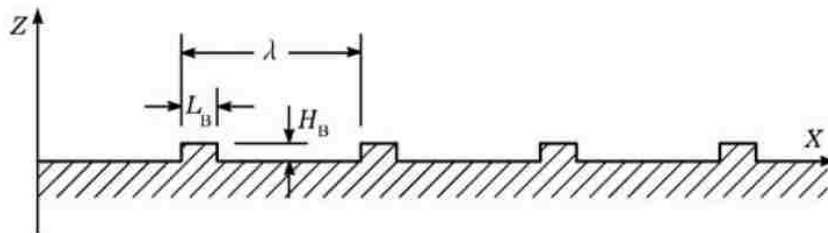
Figure 2.4: Road profile with repeating rectangular bump (reproduced from Dixon [2])

As expected, a sinusoidal road is a road in which the cross sectional shape is a sine wave, and the road has only one spatial frequency of bumps. Figure 2.3 shows two sinusoidal single paths with different spatial frequencies. A periodic disturbance is given by [3]:

$$z_1(s) = A\sin(\Omega s) \tag{2.1}$$

where $s = s(t)$ is the road path variable, which is a vehicle's instantaneous position driven along a given road path. If the vehicle runs at constant velocity $ds/dt = v_0$, the transient position of the vehicle is

$s = v_0 t$, with the assumption that the initial position $s = 0$ at $t = 0$. The amplitude is A and Ω is the wave number. The wavelength is given by:

$$L = \frac{2\pi}{\Omega} \tag{2.2}$$

Then the Ωs term can be written as:

$$\Omega s = \frac{2\pi}{L} s = \frac{2\pi}{L} v_0 t = 2\pi \frac{v_0}{L} t = \omega t \tag{2.3}$$

Hence, the excitation frequency in the time domain is $f = \omega/(2\pi) = v_0/L$.

In ride quality testing, fixed waveforms are sometimes used by engineers. A typical example is the rectangular bump repeating regularly on a road profile, as in Figure 2.4. It is easy to build this kind of profile by putting planks of wood on an otherwise level road surface. This profile can be used to study the influence of road roughness on the maximum cornering ability of vehicles.

2.2 Road Profile Classification

Road surface profiles vary considerably in their quality. Many attempts have been made to classify road profiles. There are two popular methods to classify the roughness of a road profile: the International Roughness Index (IRI) and the International Standards Organization (ISO) classification.

The IRI is the most widely used index in the USA to measure road roughness. It is defined as a numerical value with units of slope (m/km, in/mi, etc.) by accumulating the inches of vertical movement of a vehicle divided by the distance travelled. It is used to measure the bumpiness of the road. Figure 2.5 defines the IRI ranges represented by different classes of road [1]. The limits of each category of the range of IRI are set by the United States Federal Highway Administration (FHWA) for its Highway Performance Monitoring System.

The power spectral density (PSD) and the spatial frequency are introduced by the ISO to classify road profiles. Table 2.1 shows the classification of road profiles according to the ISO 8608 [2, 4]. The table describes the range from a very smooth road to a very rough road, using eight categories labelled from class A to H, by comparing the PSDs of the road profiles. The standard ISO road assumes that everywhere along the road section in a given road, the road surface has the same distribution of shorter and longer periodic bumps with different heights. Class A can be defined as the best quality, as the A class roads have a minor degree of roughness. Conversely, class H roads have high degree of roughness, so they can be described as very rough (see Table 2.1).
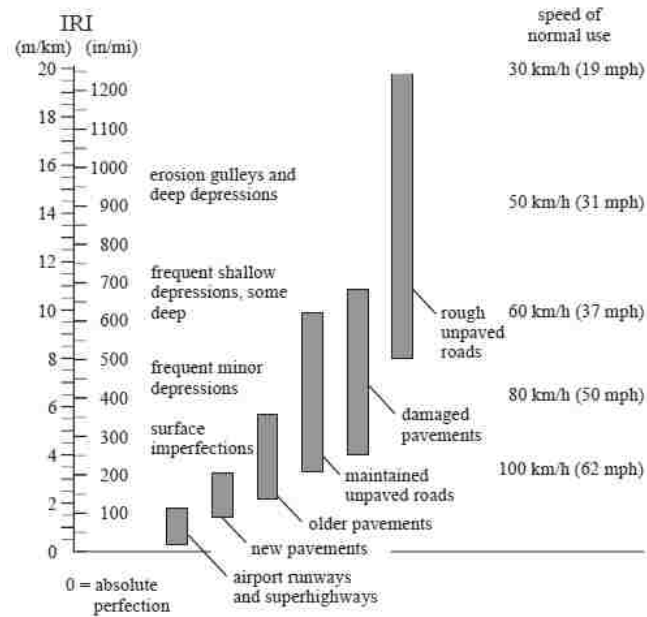
Figure 2.5: IRI classification of different types of road (reproduced from Karamihas and Sayers [1])

Table 2.1: Road profiles classification by ISO 8608, from Agostinacchio, et al. [4]

Road class	$G_d(n_0)(10^{-6})$ m^3		$G_d(\Omega_0)(10^{-6})$ m^3		ISO description
	Lower limit	Upper limit	Lower limit	Upper limit	
A	–	32	–	2	very good
B	32	128	2	8	good
C	128	512	8	32	average
D	512	2048	32	128	poor
E	2048	8192	128	512	very poor
F	8192	32768	512	2048	–
G	32768	131072	2048	8192	–
H	131072	–	8192	–	–
$n_0 = 0.1$ cycles/m			$\Omega_0 = 1$ rad/m		

Two methods are used to generate the road profiles in the ISO 8608 from data measured on site by applying two different procedures. The first one describes the road roughness profiles by calculating the PSDs of vertical displacements G_d, both as a function of spatial frequency n ($n = \Omega/2\pi$ cycles/m) and of angular spatial frequency Ω. Usually, both the function of n and Ω, $G_d(n)$ and $G_d(\Omega)$ are plotted with log-log scale.

The second method calculates the PSDs of the accelerations $G_a(n)$ and $G_a(\Omega)$ of the road profiles in accordance with slope variation of the road surface per unit of covered distance. The relationship

between the two procedures is given in reference [4]:

$$
\begin{aligned}
G_a(n) &= (2\pi n)^4 \cdot G_d(n) \\
G_a(\Omega) &= \Omega^4 \cdot G_d(\Omega)
\end{aligned}
\tag{2.4}
$$

In order to classify different road profiles, conventional values of spatial frequency $n_0 = 0.1$ cycles/m or angular spatial frequency $\Omega_0 = 1$ rad/m are applied as reference frequency to calculate their PSDs by ISO 8608. To identify the class of a real road profile, the PSD of the profile is compared with those in the ISO standard for various classes, in terms of n_0 and Ω_0.

In simulation, these equations can be used to obtain the roughness profile of the road surface in ISO 8608:

$$
\begin{aligned}
G_d(n) &= G_d(n_0) \cdot \left(\frac{n}{n_0}\right)^{-2} \\
G_d(\Omega) &= G_d(\Omega_0) \cdot \left(\frac{\Omega}{\Omega_0}\right)^{-2}
\end{aligned}
\tag{2.5}
$$

where the values of $G_d(n_0)$ and $G_d(\Omega_0)$ must be derived from Table 2.1 on the basis of the considered road class. The PSDs of the standard classes are plotted in Figure 2.6 in the spatial frequency range of 0.01 to 1 cycles/m based on a spatial frequency of $n_0 = 0.1$ cycles/m. Each line is the transition state of two adjacent road profile classes. Below the bottom line is class A, representing very smooth highways. Beyond the top line is class H, which represents very rough roads.
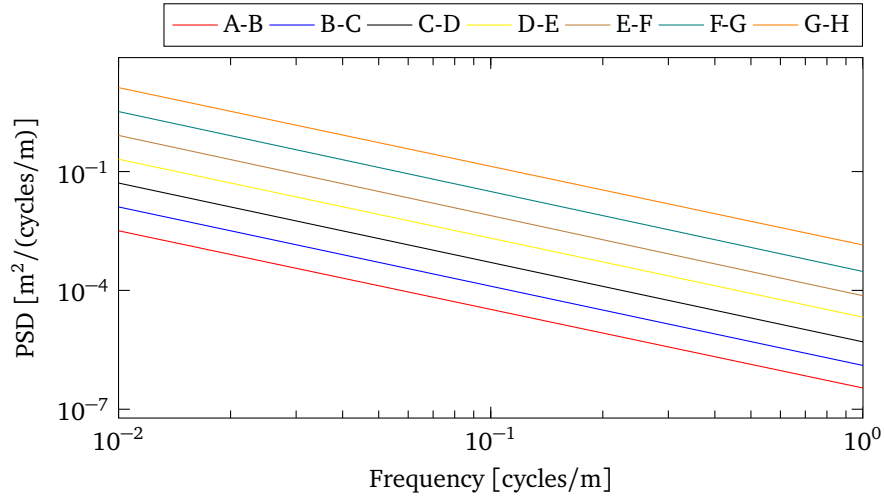
Figure 2.6: Classification of road profiles based on PSD

2.3   Current Methods

Road profile measurements first started in the early 1900s, and they have evolved to measure the road profile with vehicles traveling at normal traffic speed. There are a number of solutions that have been proposed to obtain road profile data, including physical or laser based measurement, and camera based techniques. A procedure that is popular in industry is the use of wheel force transducers to calculate the effective road profile, but these are expensive, and heavy enough to influence suspension performance, and the procedure takes a long time. A more analytical approach is to use a dynamic model and an 'observer', a numerical technique that attempts to force the model states to converge to the real states; see [5–8].

For the purpose of road serviceability, survey and maintenance, many studies have been conducted with various types of profilometers. One of the first is the inertial profiler proposed by the General Motors Research (GMR) Laboratories in 1964 (Figure 2.7) [1, 5]. In this profiler, an accelerometer placed on the body of the test vehicle is used to build an inertial reference. A non-contact light or acoustic sensor such as a laser transducer is used to measure the relative distance between the accelerometer and the ground. The longitudinal distance of this instrument is normally picked up from the vehicle speedometer. An estimated road profile can be obtained from the difference between the height sensor output and the absolute motion of the vehicle body. Current inertial profilers can only measure and record road profile at certain speeds, e.g., between 16 and 112 km/h. In addition, this method depends heavily on the sensor location, because it is difficult to ensure a visual connection between the instrument and the ground. In [9], Pong has also found some drawbacks and limitations of its capabilities.
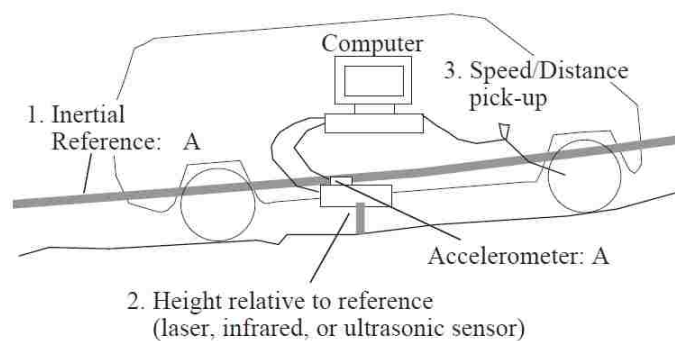
Figure 2.7: GMR profilometer (reproduced from Karamihas and Sayers [1])

Another example is a longitudinal profile analyser (LPA) with a mounted laser sensor, which is developed by the Road and Bridges Central Laboratory in France to measure the road profile, as illustrated in Figure 2.8. This system uses one or two single-wheel trailers towed at constant speed by a car to get measured data. An oscillating beam connecting a feeler wheel that directly contacts the road profile is

supported by the chassis with a suspension and damping system. Vertical movements are measured by an angular displacement transducer associated with the inertial pendulum. Obtained measurements need to be processed to estimate the road profile. The limitation is that this device cannot be integrated in ordinary passenger cars. This instrument sometimes produces biased or corrupted measures as well [5,10].
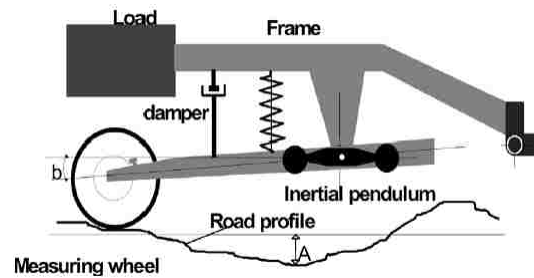
Figure 2.8: Longitudinal profile analyser (reproduced from Imine et.al. [5])

In [10], a road profile estimation approach based on a Kalman-filter using the previously described road profilometer is developed. The proposed method focuses on the vertical dynamic response of an active suspension system, which can be obtained by accelerometers and suspension deflection sensors. In addition, the result of the estimation process is experimentally validated. Though experimental results show the accuracy of the estimation method, this method can be used only at constant speed.

In [11], a method using a wavelet neural network (WNN) is presented to estimate the road profile, as the WNN has strong advantages in dealing with non-linear mapping and online estimates. It has stronger approximating, tolerance and classification capacity than a conventional neural network. Additionally, a WNN combines the time-frequency characteristic of a wavelet transformation with the self-learning of a conventional neural network. A seven degrees of freedom (DOF) vehicle dynamic model is used to obtain acceleration data in MATLAB® Simulink software. Then the WNN is trained and tested using the data to approximate road profiles. This method avoids the use of dynamic models, but only uses virtual data for simulations.

In Yang, Yi and Lian's paper [12], a vehicle-mounted system combining laser displacement sensors, accelerometers, and a gyroscope is developed. In this system, the vertical displacement of a test vehicle is measured in a broad vibration frequency range by dual accelerometers and laser sensors. The longitudinal displacement is measured by an incrementally revolving encoder, while the attitude is measured by a gyroscope. An algorithm based on the double integration of acceleration with respect to time is designed to recover road profiles. However, this instrument can only measure road profiles on a large-scale and it is complicated and expensive.

In the paper 'The use of vehicle acceleration measurements to estimate road roughness', Gonzalez et al. proposed a road roughness estimate method based on vehicle accelerometers [13]. This method uses a transfer function between the PSD of the road surface and of the vehicle accelerations to obtain road profiles. However, road profiles are only accurately estimated at a constant vehicle velocity, and a known road profile is required. In addition, the road estimation algorithm depends on a specific frequency. Similarly, Hong, Sohn and Hedrick [14] developed a road estimator based on the Fourier transform of the road. In [15], the road estimator can be used at variable velocity but information about the road is required.

An approach based on sliding mode observers is also quite popular in this area. In [5], a method to obtain road profiles based on a sliding mode observer approach is developed. The profiles estimated from this method are compared with results obtained from the aforementioned profilometers. Similarly, to remove the perturbation, a sliding mode observer is used in the paper 'Second order sliding mode observer for estimation of road profile' [6]. A nominal dynamic model of a vehicle is used in this method for the reconstruction of the unknown inputs of the system. Other methods using sliding mode observers can be also seen in [7, 8]. One of the shortcomings of this kind of approach is that it requires measurement of at least some of the states of the dynamic model, which are often hard to obtain.

Numerous approaches to estimate road profiles are developed in the literature. However, roughly speaking, all these methods have some limitations. Therefore, a simple and low-cost method to estimate a road profile efficiently from different types of vehicles on any surface is still a challenge in vehicle testing.

Chapter 3

Overview of Vehicle Dynamics and Control

The road profile is an essential factor that influences vehicle dynamics. Research in vehicle dynamics has become increasingly popular in recent years. Vehicle dynamics is a portion of engineering mainly based on classical mechanics. For goals such as safety, performance, durability, and efficiency, advanced vehicle modeling with the aid of computer control has been heavily relied on by automotive manufacturers. The foundation of vehicle dynamics and computer control is the development of dynamic models and control systems. Therefore, a brief description of important concepts along with research objectives will follow.

3.1 Equations of Motion

It is quite difficult to capture all the motions of a vehicle in analytical equations. Of course, adding more elements in the model may increase the model's accuracy, but it also substantially increases the complexity and the computation time. This section describes the derivation of different car models considered in this study.

3.1.1 Coordinate System

It is necessary to define a set of coordinate systems in order to derive the equations of motion. This allows a more systematic approach to modeling, which is very important when dealing with more complex models. Figure 3.1 illustrates the vehicle body coordinate frame $B(Cxyz)$, which is attached to the vehicle at the mass center C, and is usually used to express the equations of motion in vehicle dynamics. The x axis is a longitudinal axis passing through C, positive in the forward direction. The y axis corresponds laterally to the left from the driver's viewpoint. Using right-hand triad, the z axis goes vertically to the top. Roll, pitch and yaw are defined as rotations around the x, y and z axes respectively [16].
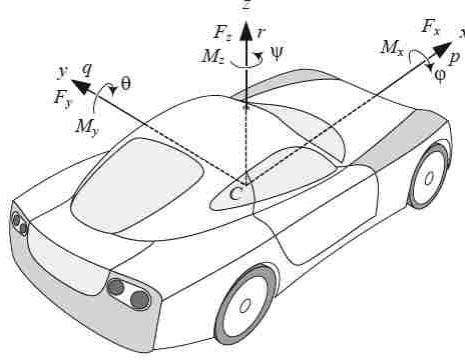
Figure 3.1: Vehicle body coordinate frame *B(Cxyz)* (reproduced from Jazar [16])

Three angles are defined to show the vehicle orientation: roll angle φ about the x axis, pitch angle θ about the y axis, and yaw angle ψ about the z axis. The corresponding rates of the orientation angles are also introduced, as they are important in vehicle dynamics. They are usually called roll rate, pitch rate and yaw rate. The relationship between the angular velocities and the angles is shown in [17]. Here, the approximation of the rotation rates, if at least two angles are small, are given by:

$$p \approx \dot{\varphi} \tag{3.1}$$

$$q \approx \dot{\theta} \tag{3.2}$$

$$r \approx \dot{\psi} \tag{3.3}$$

Let *G(OXYZ)* denote the ground fixed frame called the global coordinate frame. Figure 3.2 shows the global frame *G* and the body frame *B* in a moving vehicle. When two of the rotation angles are zero, the rotation matrices between the global coordinate G and body coordinate B given in [16] are:

$$^{G}R_{B}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi) & sin(\varphi) \\ 0 & -sin(\varphi) & cos(\varphi) \end{bmatrix} \, ^{G}R_{B}(2) = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix} \, ^{G}R_{B}(3) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\tag{3.4}$$

It becomes more complex to express velocities and accelerations involving moving coordinate systems. The reason is that these quantities need to be expressed in inertial frames when used in the equations of motion. Consider a point described by the vector *P* relative to a body-fixed frame. A vector *R* is assumed to be the translation from the origin of an inertial frame to the origin of the body system. Assume the body frame rotates with angular velocity ω relative to the inertial frame. The velocity of the point *P* in
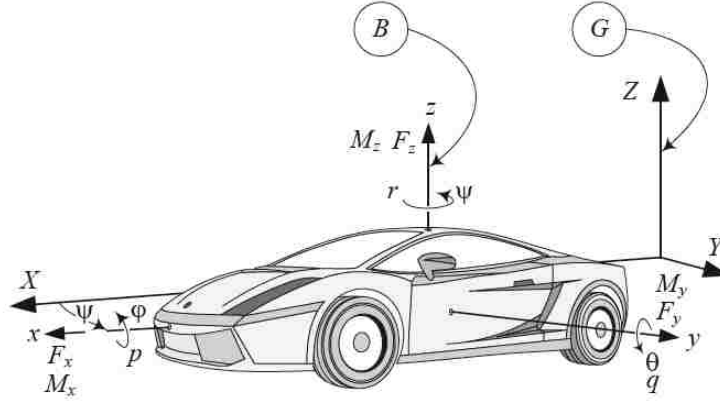
Figure 3.2: Illustration of the body coordinate frame *B* in a global coordinate frame
G, in a moving vehicle (reproduced from Jazar [16])

the inertial frame can be expressed:

$$\frac{dP}{dt}\bigg|_i \;=\; \frac{dR}{dt} + \frac{dP}{dt}\bigg|_b + \omega_i^b \times P \tag{3.5}$$

Similarly, the acceleration of the point *P* in the inertial frame is:

$$\frac{d^2P}{dt^2}\bigg|_i \;=\; \frac{d^2R}{dt^2} + \frac{d^2P}{dt^2}\bigg|_b + \dot{\omega}_i^b \times P + 2\omega_i^b \times \frac{dP}{dt}\bigg|_b + \omega_i^b \times (\omega_i^b \times P) \tag{3.6}$$

3.1.2 Laws of Motion

There are a quantity of approaches to obtain the equations of motion of a multibody system in terms of the generalized coordinates and speeds, such as: Newton-Euler equations, Lagrange's equations, and Hamilton's principle, etc. One of the popular and commonly used methods is the Newton-Euler method.

The first step to generate the equations of motion of a multibody system is to calculate the number of DOF of the system, which is the minimum number of coordinates required to fully describe the given system. The number of DOF can be calculated by subtracting the number of constraints in the system from the number of coordinates of the system. Each body in the system, when floating freely in Cartesian space, has 6 DOF, e.g., three position coordinates and three orientation coordinates. The constraints in the system are what restricts the movement of the bodies in space. Each constraint will decrease the number of DOF by one.

The Newton-Euler equations for dynamic modeling of the system, based on Newton's second law of motion in both the translational and rotational domains, are given in [18] as:

$$\Sigma F \;=\; m\dot{v} + \omega \times mv \tag{3.7}$$

$$\Sigma M_G = I_G \dot{\omega} + \omega \times I_G \omega \tag{3.8}$$

where m is the mass of the body, v is the body velocity, I_G is the inertia dyadic, and ω is the angular velocity of the moving reference frame. As shown in [18], the angular velocity of the body consists of the ω term plus the 'spin' s only if the body is axisymmetric. The equations for rotation and translation can be combined as:

$$M\dot{w} = \Sigma f(p, w) \tag{3.9}$$

where M is the mass matrix, w is the velocity of the body, including v and ω, and p is the position of the body. According to Minaker and Rieveley [18], the force acting on the body can be divided into: inertial forces, passive forces, constraint forces, external forces, and active forces.

3.1.3 Dynamic Models

The use of vehicle dynamics models to study vehicle behaviour and vehicle control system design has evolved steadily over the last few decades. In order to test new control methods, it is necessary to develop a virtual model that precisely represents the vehicle, so control methods can be tested out prior to physical implementation. So far, many different vehicle models have been used for different applications. Common practical models used for vehicle ride, such as the quarter car model, half car model, and full car model, have DOF that range from 2 to 7, e.g. [6–8, 19, 20]. These models mainly focus on vertical motions of vehicle components and rotational axes such as pitch and roll.

Quarter Car Model

For the sake of the vertical dynamics of a vehicle, the well-known 2 DOF quarter car model, shown in Figure 3.3 is widely used. The model only includes a corner of a test vehicle, which means it only has one wheel of the vehicle. The model is composed of a sprung mass and an unsprung mass. The damper and the spring of the suspension system are modeled as c_s and k_s respectively. The tire is modeled as a linear spring without damping, as usually the damping effect of the tire is very small compared to the suspension damping. The tire spring stiffness is k_t.

Using Newton's second law, the differential equations of motion for the sprung mass and unsprung mass are respectively:

$$m_s \ddot{z}_1 + c_s(\dot{z}_1 - \dot{z}_2) + k_s(z_1 - z_2) = 0 \tag{3.10}$$

$$m_u \ddot{z}_2 + c_s(\dot{z}_2 - \dot{z}_1) + k_s(z_2 - z_1) + k_t z_2 = k_t z_0 \tag{3.11}$$
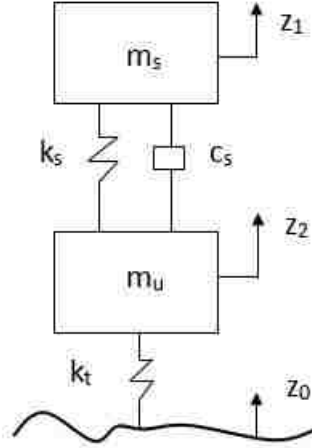
Figure 3.3: A quarter car model

By denoting:

$$z = \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix}, \quad u = z_0 \tag{3.12}$$

the two equations can be combined into a vector equation:

$$\mathbf{M}\ddot{z} + \mathbf{C}\dot{z} + \mathbf{K}z = \mathbf{F}u \tag{3.13}$$

where mass, stiffness, and damping matrices, and the external force vector are written as:

$$\mathbf{M} = \begin{bmatrix} m_s & 0 \\ 0 & m_u \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} c_s & -c_s \\ -c_s & c_s \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} k_s & -k_s \\ -k_s & k_s + k_t \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} 0 \\ k_t \end{bmatrix} \tag{3.14}$$

In this model, m_s and m_u are respectively the mass of the quarter car body and the wheel, k_s is the stiffness of the spring incorporated in the suspension system, k_t is the tire stiffness, c_s is the damping coefficient of the shock absorber, z_1 and z_2 are the position of the sprung mass and unsprung mass respectively, z_0 is the displacement of the road surface.

Half Car Model (Left/Right)

In order to study pitch and other modes of vibration, a half car vibrating model of a vehicle (left/right) is introduced, as illustrated in Figure 3.4. The half car model is a simplified vehicle with two quarter car models, one for the front suspension and one for the rear suspension. The four independent DOF consist of the front and rear axle hops, sprung mass heave, and sprung mass pitch rotation induced by different road profiles underneath the front and rear wheel paths.

Figure 3.4: A four DOF half car model (left/right), from Jazar [16]

The equations of motion for the half car model are given as:

$$m\ddot{x} + c_1(\dot{x} - \dot{x}_1 - a_1\dot{\theta}) + c_2(\dot{x} - \dot{x}_2 - a_2\dot{\theta}) + k_1(x - x_1 - a_1\theta) + k_2(x - x_2 - a_2\theta) = 0 \qquad (3.15)$$

$$m_1\ddot{x}_1 + c_1(\dot{x} - \dot{x}_1 - a_1\dot{\theta}) + k_{t1}(x_1 - y_1) - k_1(x - x_1 + a_1\theta) = 0 \qquad (3.16)$$

$$m_2\ddot{x}_2 + c_2(\dot{x} - \dot{x}_2 - a_2\dot{\theta}) + k_{t2}(x_2 - y_2) - k_2(x - x_2 + a_2\theta) = 0 \qquad (3.17)$$

$$I_y\ddot{\theta} + a_1c_1(\dot{x} - \dot{x}_1 - a_1\dot{\theta}) + a_2c_2(\dot{x} - \dot{x}_2 - a_2\dot{\theta}) - a_1k_1(x - x_1 - a_1\theta) + a_2k_2(x - x_2 - a_2\theta) = 0 \;(3.18)$$

The definition of the parameters used is given in Table 3.1.

Table 3.1: Parameter definition of a half car model (left/right)

Symbol	Meaning	Symbol	Meaning
m	half body mass	I_y	half body lateral mass moment of inertial
m_1	front wheel mass	a_1	distance of mass center C from front axle
m_2	rear wheel mass	a_2	distance of mass center C from rear axle
x	body vertical displacement	k_1	front spring stiffness
x_1	front wheel vertical displacement	k_2	rear spring stiffness
x_2	rear wheel vertical displacement	c_1	front suspension damping
θ	body pitch angle	c_2	rear suspension damping
y_1	road excitation at front wheel	k_{t1}	front tire stiffness
y_2	road excitation at rear wheel	k_{t2}	rear tire stiffness

The equations of motion can be converted into the form of Equation 3.13 by denoting:

$$\boldsymbol{x}(t) \;=\; \begin{Bmatrix} x(t) \\ \theta(t) \\ x_1(t) \\ x_2(t) \end{Bmatrix}, \quad \boldsymbol{u} \;=\; \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} \tag{3.19}$$

The mass, stiffness, and damping matrices, and the external force vector are written as:

$$\mathbf{M} \;=\; \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & I_z & 0 & 0 \\ 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & m_2 \end{bmatrix} \tag{3.20}$$

$$\mathbf{C} \;=\; \begin{bmatrix} c_1 + c_2 & a_2 c_2 - a_1 c_1 & -c_1 & -c_2 \\ a_2 c_2 - a_1 c_1 & c_1 a_1{}^2 + c_2 a_2{}^2 & a_1 c_1 & -a_2 c_2 \\ -c_1 & a_1 c_1 & c_1 & 0 \\ -c_2 & -a_2 c_2 & 0 & c_2 \end{bmatrix} \tag{3.21}$$

$$\mathbf{K} \;=\; \begin{bmatrix} k_1 + k_2 & a_2 k_2 - a_1 k_1 & -k_1 & -k_2 \\ a_2 k_2 - a_1 k_1 & k_1 a_1{}^2 + k_2 a_2{}^2 & a_1 k_1 & -a_2 k_2 \\ -k_1 & a_1 k_1 & k_1 + k_{t1} & 0 \\ -k_2 & -a_2 k_2 & 0 & k_2 + k_{t2} \end{bmatrix} \tag{3.22}$$

$$\mathbf{F} \;=\; \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{t1} & 0 \\ 0 & k_{t2} \end{bmatrix} \tag{3.23}$$

Half Car Model (Front/Rear)

For the purpose of examining and optimizing the roll vibration of a vehicle, the half car vibrating model with only one axle (front or rear) is proposed; see Figure 3.5. Similarly, this model consists of the left and right suspensions of a vehicle. It is common to use same stiffness and damping for the suspension on the left and right wheels. Due to different suspensions and mass distribution, the half car model is different for the front and rear halves. In addition, different anti-roll bars with different torsional stiffness may be used in the front and rear halves [16].
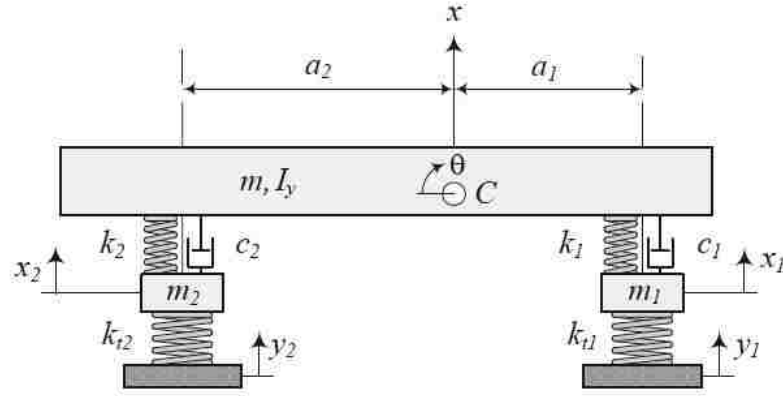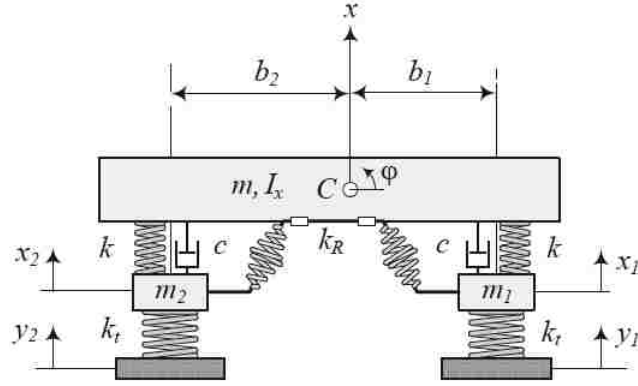
Figure 3.5: A half car vibrating model (front/rear), from Jazar [16]

The equations of motion for the half car model are given as:

$$m\ddot{x} + c(\dot{x} - \dot{x}_1 + b_1\dot{\varphi}) + c(\dot{x} - \dot{x}_2 - b_2\dot{\varphi}) + k(x - x_1 + b_1\varphi) + k(x - x_2 - b_2\varphi) = 0 \qquad (3.24)$$

$$I_x\ddot{\varphi} + b_1 c(\dot{x} - \dot{x}_1 + b_1\dot{\varphi}) - b_2 c(\dot{x} - \dot{x}_2 - b_2\dot{\varphi}) + b_1 k(x - x_1 + b_1\varphi) - b_2 k(x - x_2 - b_2\varphi) + k_R(\varphi - \frac{x_1 - x_2}{W}) = 0 \qquad (3.25)$$

$$m_1\ddot{x}_1 - c(\dot{x} - \dot{x}_1 + b_1\dot{\varphi}) + k_t(x_1 - y_1) - k(x - x_1 + b_1\varphi) - k_R\frac{1}{W}(\varphi - \frac{x_1 - x_2}{W}) = 0 \qquad (3.26)$$

$$m_2\ddot{x}_2 - c(\dot{x} - \dot{x}_2 - b_2\dot{\varphi}) + k_t(x_2 - y_2) - k(x - x_2 - b_2\varphi) + k_R\frac{1}{W}(\varphi - \frac{x_1 - x_2}{W}) = 0 \qquad (3.27)$$

Denoting:

$$x(t) = \begin{Bmatrix} x(t) \\ \varphi(t) \\ x_1(t) \\ x_2(t) \end{Bmatrix}, \quad u = \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} \qquad (3.28)$$

The mass, stiffness, and damping matrices, and the external force vector are written as:

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & I_X & 0 & 0 \\ 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & m_2 \end{bmatrix} \qquad (3.29)$$

$$\mathbf{C} = \begin{bmatrix} 2c & cb_1 - cb_2 & -c & -c \\ cb_1 - cb_2 & cb_1^2 + cb_2^2 & -cb_1 & cb_2 \\ -c & -cb_1 & c & 0 \\ -c & cb_2 & 0 & c \end{bmatrix} \qquad (3.30)$$

$$\mathbf{K} = \begin{bmatrix} 2k & kb_1 - kb_2 & -k & -k \\ kb_1 - kb_2 & kb_1{}^2 + kb_2{}^2 + k_R & -kb_1 - \frac{k_R}{W} & kb_2 + \frac{k_R}{W} \\ -k & -kb_1 - \frac{k_R}{W} & k + k_t + \frac{k_R}{W^2} & -\frac{k_R}{W^2} \\ -k & kb_2 + \frac{k_R}{W} & -\frac{k_R}{W^2} & k + k_t + \frac{k_R}{W^2} \end{bmatrix} \tag{3.31}$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_t & 0 \\ 0 & k_t \end{bmatrix} \tag{3.32}$$

In this model, m is the body mass, which is the front or rear half of the total body mass, I_x is half of the longitudinal mass moment of inertia, m_1 and m_2 are the left and right wheel masses respectively, k_t is the tire stiffness, tire damping is neglected. The suspension of the car has stiffness k and damping c for the left and right wheels. b_1 and b_2 are the distance of mass center C from the left and right wheels of the vehicle. The track width of the vehicle is $W = b_1 + b_2$. The torsional stiffness of the anti-roll bar is k_R. The torque provided by the anti-roll bar is proportional to the roll angle φ.

$$M_R = -k_R(\varphi - \frac{x_1 - x_2}{W}) \tag{3.33}$$

Full Car Model

A general vibrating model of a vehicle that is close to the real vehicle structure is the full car model. The objective using this type of model may be either safety and comfort analysis or design. As shown in Figure 3.6, the model includes a car body, four suspensions and four wheels. The seven DOF of the model are the body bounce x, roll angle φ, pitch angle θ, and the displacements of the wheels x_1, x_2, x_3, x_4. Here, the yaw motion is not considered.
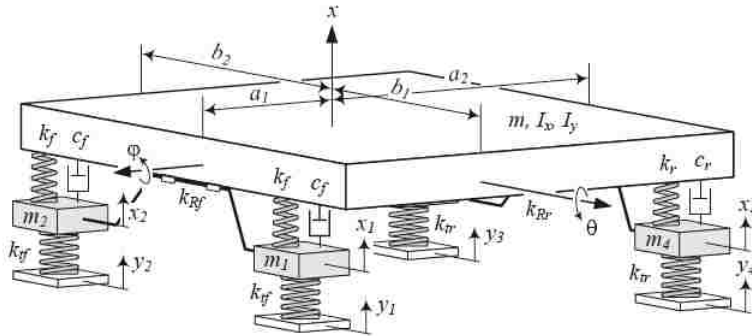

Figure 3.6: A full car vibrating model, from Jazar [16]

The equations of motion of this model are given in [16]:

$$m\ddot{x} + c_f(\dot{x} - \dot{x}_1 + b_1\dot{\varphi} - a_1\dot{\theta}) + c_f(\dot{x} - \dot{x}_2 - b_2\dot{\varphi} - a_1\dot{\theta})$$

$$+c_r(\dot{x} - \dot{x}_3 - b_1\dot{\varphi} + a_2\dot{\theta}) + c_r(\dot{x} - \dot{x}_4 + b_2\dot{\varphi} + a_2\dot{\theta}) + k_f(x - x_1 + b_1\varphi - a_1\theta)$$

$$+k_f(x - x_2 - b_2\varphi - a_1\theta) + k_r(x - x_3 - b_1\varphi + a_2\theta) + k_r(x - x_4 + b_2\varphi + a_2\theta)$$

$$= 0 \qquad (3.34)$$

$$I_x\ddot{\varphi} + b_1c_f(\dot{x} - \dot{x}_1 + b_1\dot{\varphi} - a_1\dot{\theta}) - b_2c_f(\dot{x} - \dot{x}_2 - b_2\dot{\varphi} - a_1\dot{\theta})$$

$$-b_1c_r(\dot{x} - \dot{x}_3 - b_1\dot{\varphi} + a_2\dot{\theta}) + b_2c_r(\dot{x} - \dot{x}_4 + b_2\dot{\varphi} + a_2\dot{\theta}) + b_1k_f(x - x_1 + b_1\varphi - a_1\theta)$$

$$-b_2k_f(x - x_2 - b_2\varphi - a_1\theta) - b_1k_r(x - x_3 - b_1\varphi + a_2\theta) + b_2k_r(x - x_4 + b_2\varphi + a_2\theta)$$

$$+k_R(\varphi - \frac{x_1 - x_2}{W}) = 0 \qquad (3.35)$$

$$I_y\ddot{\theta} - a_1c_f(\dot{x} - \dot{x}_1 + b_1\dot{\varphi} - a_1\dot{\theta}) - a_1c_f(\dot{x} - \dot{x}_2 - b_2\dot{\varphi} - a_1\dot{\theta})$$

$$+a_2c_r(\dot{x} - \dot{x}_3 - b_1\dot{\varphi} + a_2\dot{\theta}) + a_2c_r(\dot{x} - \dot{x}_4 + b_2\dot{\varphi} + a_2\dot{\theta}) - a_1k_f(x - x_1 + b_1\varphi - a_1\theta)$$

$$-a_1k_f(x - x_2 - b_2\varphi - a_1\theta) + a_2k_r(x - x_3 - b_1\varphi + a_2\theta) + a_2k_r(x - x_4 + b_2\varphi + a_2\theta)$$

$$= 0 \qquad (3.36)$$

$$m_1\ddot{x}_1 - c_f(\dot{x} - \dot{x}_1 + b_1\dot{\varphi} - a_1\dot{\theta}) - k_f(x - x_1 + b_1\varphi - a_1\theta)$$

$$-k_R\frac{1}{W}(\varphi - \frac{x_1 - x_2}{W}) + k_{tf}(x_1 - y_1) = 0 \qquad (3.37)$$

$$m_2\ddot{x}_2 - c_f(\dot{x} - \dot{x}_2 - b_2\dot{\varphi} - a_1\dot{\theta}) - k_f(x - x_2 - b_2\varphi - a_1\theta)$$

$$+k_R\frac{1}{W}(\varphi - \frac{x_1 - x_2}{W}) + k_{tf}(x_2 - y_2) = 0 \qquad (3.38)$$

$$m_3\ddot{x}_3 - c_r(\dot{x} - \dot{x}_3 - b_1\dot{\varphi} + a_2\dot{\theta}) - k_r(x - x_3 - b_1\varphi + a_2\theta) + k_{tr}(x_3 - y_3) = 0 \qquad (3.39)$$

$$m_4\ddot{x}_4 - c_r(\dot{x} - \dot{x}_4 + b_2\dot{\varphi} + a_2\dot{\theta}) - k_r(x - x_4 + b_2\varphi + a_2\theta) + k_{tr}(x_4 - y_4) = 0 \qquad (3.40)$$

Denoting:

$$x(t) = \begin{Bmatrix} x(t) \\ \varphi(t) \\ \theta(t) \\ x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{Bmatrix}, \quad u = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} \qquad (3.41)$$

The mass, stiffness, and damping matrices, and the external force vector are written as:

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_4 \end{bmatrix} \qquad (3.42)$$

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & -c_f & -c_f & -c_r & -c_r \\ c_{21} & c_{22} & c_{23} & -b_1 c_f & b_2 c_f & b_1 c_r & -b_2 c_r \\ c_{31} & c_{32} & c_{33} & a_1 c_f & a_1 c_f & -a_2 c_r & -a_2 c_r \\ -c_f & -b_1 c_f & a_1 c_f & c_f & 0 & 0 & 0 \\ -c_f & b_2 c_f & a_1 c_f & 0 & c_f & 0 & 0 \\ -c_r & b_1 c_r & -a_2 c_r & 0 & 0 & c_r & 0 \\ -c_r & -b_2 c_r & -a_2 c_r & 0 & 0 & 0 & c_r \end{bmatrix} \qquad (3.43)$$

where:

$$\begin{aligned}
c_{11} &= 2c_f + 2c_r \\
c_{21} &= c_{12} = b_1 c_f - b_2 c_f - b_1 c_r + b_2 c_r \\
c_{31} &= c_{13} = 2a_2 c_r - 2a_1 c_f \\
c_{22} &= b_1^2 c_f + b_2^2 c_f + b_1^2 c_r + b_2^2 c_r \\
c_{32} &= c_{23} = a_1 b_2 c_f - a_1 b_1 c_f - a_2 b_1 c_r + a_2 b_2 c_r \\
c_{33} &= 2c_f a_1^2 + 2c_r a_2^2
\end{aligned} \qquad (3.44)$$

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12}k_{13} & -k_f & -k_f & -k_r & -k_r \\ k_{21} & k_{22}k_{23} & k_{24} & k_{25} & b_1k_r & -b_2k_r \\ k_{31} & k_{32}k_{33} & a_1k_f & a_1k_f & -a_2k_r & -a_2k_r \\ -k_f & k_{42} & a_1k_f & k_{44} & -\frac{k_R}{W^2} & 0 & 0 \\ -k_f & k_{52} & a_1k_f & -\frac{k_R}{W^2} & k_{55} & 0 & 0 \\ -k_r & b_1k_r & -a_2k_r & 0 & 0 & k_r+k_{tr} & 0 \\ -k_r & -b_2k_r & -a_2k_r & 0 & 0 & 0 & k_r+k_{tr} \end{bmatrix} \tag{3.45}$$

where:

$$\begin{aligned} k_{11} &= 2k_f + 2k_r \\ k_{21} &= k_{12} = b_1k_f - b_2k_f - b_1k_r + b_2k_r \\ k_{31} &= k_{13} = 2a_2k_r - 2a_1k_f \\ k_{22} &= k_R + b_1^2k_f + b_2^2k_f + b_1^2k_r + b_2^2k_r \\ k_{32} &= k_{23} = a_1b_2k_f - a_1b_1k_f - a_2b_1k_r + a_2b_2k_r \\ k_{42} &= k_{24} = -b_1k_f - \tfrac{1}{W}k_R \quad k_{52} = k_{25} = b_2k_f + \tfrac{1}{W}k_R \end{aligned} \tag{3.46}$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_{tf} & 0 & 0 & 0 \\ 0 & k_{tf} & 0 & 0 \\ 0 & 0 & k_{tr} & 0 \\ 0 & 0 & 0 & k_{tr} \end{bmatrix} \tag{3.47}$$

In this model, m is the total body mass, the wheels have a mass m_1, m_2, m_3, and m_4 respectively. The longitudinal and lateral mass moments of inertia are I_x and I_y respectively. The front and rear tires stiffness are k_{tf} and k_{tr} respectively. It is common to make the suspension of the left and right wheels mirror copies. The front and rear suspensions have stiffness k_f and k_r and damping c_f and c_r respectively. The anti-roll bar in the front and in the rear have a torsional stiffness k_{Rf} and k_{Rr}; here only front anti-roll bar is considered, and $k_{Rf} = k_R$. The track width of the vehicle W is equal to $b_1 + b_2$, while the wheelbase of the vehicle is $a_1 + a_2$.

3.1.4   EoM Software

The equations of motion can be produced by hand, or for more complex models, e.g., when including a suspension system, by using the equation of motion generator software EoM. As described in [21], the EoM open source software developed by the University of Windsor Vehicle Dynamics and Control Research Group, is a collection of MATLAB® or Octave codes, and it can be used to generate linear or linearized equations of motion for multibody mechanical systems. By using a simple function file describing

the system as the input of the software, the state space form of the equations can be easily obtained from the output. Moreover, a report with the results of a linear analysis in a .pdf format can be generated. Animations of the mode shapes can be viewed as well.

3.2 Control Theory

A review of some control theory used in this project will be described in this section.

3.2.1 Control Systems

Control systems have become increasingly important for their operation on modern vehicles. A control system is an interconnection of components forming a system to obtain a desired performance of a system, and the process can be divided into open-loop control or closed-loop control, depending on the system architecture and control approach used. Figure 3.7 shows an open-loop control system. In this system, a process is controlled by a controller to obtain desired response, regardless of the actual state of the process. A vehicle performing a predefined sequence of motions without any information and 'with no concern' regarding the results of its motion is an example of an open-loop control system.
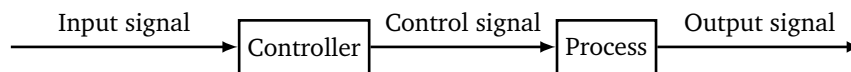
Figure 3.7: Open-loop control system
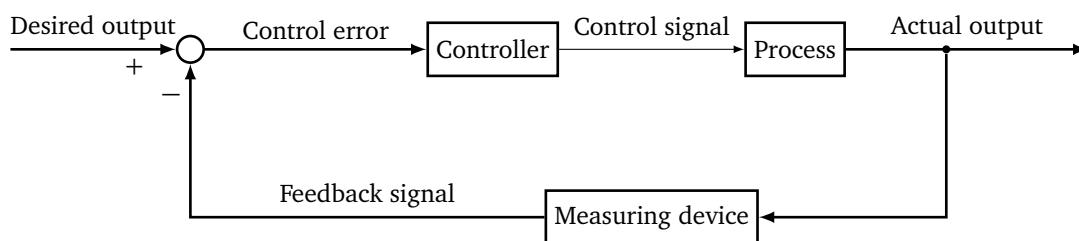
Figure 3.8: Closed-loop control system

A closed-loop feedback control system, also known simply as a feedback control system is shown in Figure 3.8. Compared to an open-loop control system, the closed-loop control system uses an additional measurement of the output and a feedback of this signal to compare it with the desired output response. For most closed-loop control systems, the primary purpose is to drive the output signal as close as possible to the input signal. This system uses a function of a prescribed relationship between the output and

reference input to control the process. The output signal of the controller is based on the difference between the desired and the actual process output. For instance, a person drives a vehicle by looking at the direction of the road and making adjustments. The closed-loop control system has many advantages despite the cost and increased system complexity, described in [22].

3.2.2   Transfer Function

A transfer function (TF) is a representation describing the input-output relation for a linear time-invariant (LTI) system. The transfer function can be obtained by inspection for simple systems, or for more complex systems, it can be calculated from the differential equations that describe the systems. A useful method for dealing with complex systems is to combine transfer functions with block diagrams. For example, consider a simple spring-mass suspension system as shown in Figure 3.9; road displacement u is the input, while output y is the mass displacement.
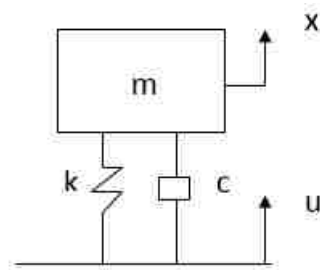
Figure 3.9: One DOF simple suspension model

The equation of motion is:

$$m\ddot{x} + c\dot{x} + kx \;=\; f(t) \tag{3.48}$$

where:

$$f(t) = c\dot{u} + ku \tag{3.49}$$

Using the Laplace transform, the dynamic equation can be rewritten in the domain of the complex variable s:

$$(ms^2 + cs + k)Y(s) \;=\; (cs + k)U(s) \tag{3.50}$$

Therefore, the transfer function from the input to the output can be identified as:

$$H(s) \;=\; \frac{Y(s)}{U(s)} \;=\; \frac{cs + k}{ms^2 + cs + k} \tag{3.51}$$

The transfer function in linear systems can be used to study the frequency response as it is described in the frequency domain. Introducing an n-th order linear system described by:

$$H(s) = \frac{B(s)}{A(s)} = \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0} \tag{3.52}$$

Normally, the order m of the numerator of the transfer function is not higher than the denominator order n. The zeros are the roots of the polynomial $B(s)$, which mean zero responses. The poles are the roots of $A(s)$, which indicate the resonant frequencies. Identifying the poles and zeros of the system transfer function can help in understanding the system behaviour.

3.2.3 PID Controller

A control algorithm or a control law in a closed-loop control system produces a signal to reduce the output signal deviation to zero or to a small value. It is easy to control and tune a system using the proportional-integral-derivative (PID) controller structure. PID controllers are usually used for second and lower order systems because of their flexibility and reliability. For higher order plants with dominant second order behaviour, it can be also applied. The ideal signal produced by a PID controller is given by in [23]:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de(t)}{dt} \tag{3.53}$$

where u is the control variable, $e(t) = r(t) - y(t)$ is the error between the set point and the measured value of the output of a system, k_p, k_i and k_d are the coefficients of the proportional, integral, and derivative terms. Figure 3.10 shows the basic block diagram of a PID controller. Three parts are included in the PID controller: the proportional part $k_p e(t)$ follows the desired set point that is proportional to the error, called a P controller; the integral part $k_i \int_0^t e(\tau)d\tau$ called an I controller removes the steady-state error, and the derivative part $k_d \frac{de(t)}{dt}$ called a D controller accounts for the rate of the change of error in the process. The P, I and D gains k_p, k_i and k_d are responsible for the weights of the controller's actions. The controller can meet specific process requirements by adjusting these gains.

Several other controller structures (PI controllers, PD controllers, etc.) can be used to control processes. The selection of the structure determines the closed-loop behaviour of the plant. It also determines the performance that can be achieved by a well-tuned controller.
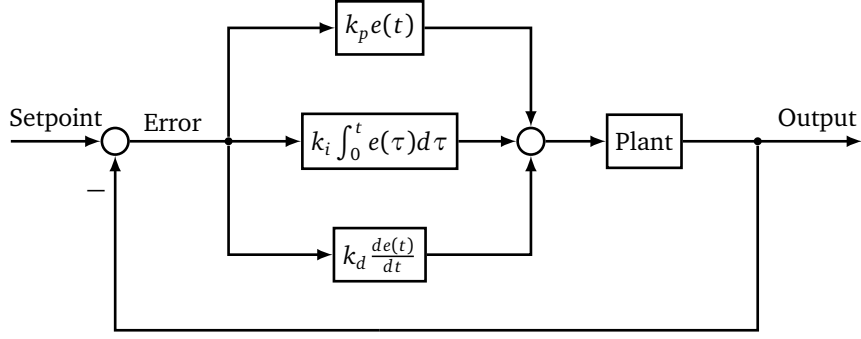
Figure 3.10: Basic block diagram of a PID controller

3.3 Frequency Response

The frequency response function of a system is a measure of the amplitude of an output of the system with respect to its input over a range of frequencies.

3.3.1 Oscillatory Response Parameters

It is possible to study two important parameters widely used in characterizing the responses of dynamic systems, which are the damping ratio ζ and natural frequency ω_n. The damping ratio ζ represents the amount of damping in a system, which describes how rapidly the oscillations decay in a system, whereas the natural frequency ω_n is the frequency of oscillations in an undamped system. For example, referring to [23], a second-order quarter car model, the system's characteristic polynomial is given by:

$$A(s) \;=\; m_s m_u s^4 + (m_s + m_u)c_s s^3 + ((m_s + m_u)k_s + m_s k_t)s^2 + c_s k_t s + k_s k_t \tag{3.54}$$

The roots of the characteristic equation are complex; they are a pair of complex-conjugate poles, i.e.,

$$s_{1,2} \;=\; \alpha \pm j\beta \;=\; -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \tag{3.55}$$

These two roots can be also obtained from the eigenvalues of the matrix $\mathbb{A}$ of the system state space representation. This may be simplified by using the MATLAB® 'pole' or 'eig' functions. The real part of the roots α represents the rate decay of the oscillations, and the imaginary part β determines the frequency of oscillation of the system. A system in which the real parts of all the roots of its characteristic equation are negative is referred to as a stable system. Correspondingly, the undamped natural frequency of the system ω_n is equal to the distance between the locations of the two roots and the origin of the coordinate system in the complex plane in terms of the real and imaginary parts of the two roots:

$$\omega_n \;=\; d \;=\; \sqrt{\alpha^2 + \beta^2} \tag{3.56}$$

The damping ratio here can be expressed as:

$$\zeta = -\frac{\alpha}{\sqrt{\alpha^2 + \beta^2}} \tag{3.57}$$

The damped natural frequency is the frequency of oscillation of an underdamped system ($0 < \zeta < 1$), and is equal to:

$$\omega_d = \omega_n\sqrt{1-\zeta^2} \tag{3.58}$$

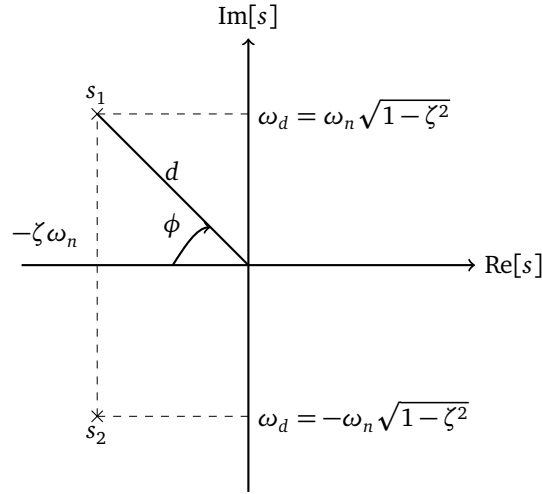**Figure 3.11:** Complex roots and corresponding values of ζ, ω_n and ω_d in the complex plane, from Kulakowski, Gardener and Shearer [23]

The relationship between the two roots of the characteristic equation and ω_n and ζ is shown in Figure 3.11. The damped natural frequency ω_d is equal to the imaginary parts of s_1 and s_2 with a plus or minus sign respectively. Another concept that should be mentioned is the time constant τ. It determines the rate of the exponential decay, and is equal to a negative inverse of the real part of the root:

$$\tau_i = -\frac{1}{\alpha_i}, \quad (i = 1, 2) \tag{3.59}$$

Knowing the roots of the characteristic equation of a linear system, the stability of the system can be defined. If a system is stable, all the roots of the characteristic equations must be real and negative, or complex and have negative real parts. In other words, the system is stable if and only if all poles of the system are located in the left half of the complex plane shown in Figure 3.11. It is important to note that stability of linear systems are not affected by the input signals.

3.3.2 Bode Diagrams

It is very helpful to use Bode diagrams to study stability and dynamic response features of dynamic systems. A Bode diagram is a plot of the frequency response of a linear system. It uses logarithmic charts to plot the magnitude and phase characteristics to describe the dynamic performance of the system. In the Bode diagrams, both magnitude $T(\omega)$ and phase $\phi_T(\omega)$ versus frequency ω are plotted on separate curves. The phase axis is linear and expresses the phase shift, while the magnitude axis is usually quasi-logarithmic (decibels) which is the magnitude of the frequency response. The magnitude in decibels of the system is defined as: $20\log_{10} H(j\omega)$, which is obtained by replacing s with $j\omega$ in the system transfer function. For more details, refer to [23].

3.3.3 High-pass Butterworth Filter

The Butterworth filter is a signal processing filter that aims to have as flat a frequency response as possible in the passband and zero roll off response in the stop band. It is one of the most commonly used digital filters in motion analysis and is easy and fast to use by choosing a cutoff frequency. The Butterworth filter is very useful to simulate the passband of an ideal filter. Because of the design feature of the Butterworth filter, the filter order, defined as the number of poles in the transfer function, can be raised to allow better filtering performance. In order to improve the accuracy of the proposed method, a high-pass Butterworth filter is implemented in this thesis by using the 'butter' function in MATLAB®.

Chapter 4

Vehicle Modeling

The previous chapter presented the derivation of the equations of motion of four common dynamic models and some control theory related to this project. The derivation of a model is very useful for getting understanding of the underlying physics. Knowing the theory of controllers is helpful to build a recovery algorithm to recover the road profile. In this chapter, the vehicle model used in this project will be introduced. In addition, the estimation method will be presented.

4.1   Vehicle Model

For the purpose of simulating vertical dynamics of a test vehicle caused by the road profile, a vibrating system mathematical vehicle model is considered. The suspension analysis is initially based on the conventional quarter car model (for short, called the conventional model), which has been shown in Figure 3.3. Typically, this model consists of a single sprung mass (the vehicle body, powertrain, passengers, cargo, etc.) connected to an unsprung mass (wheel and axle assembly). For simplicity, pitch and roll motion effects are considered negligible. The suspension between the sprung mass and the unsprung mass is modeled as a passive viscous damper and a linear coil spring. The tire is modeled as a simple linear spring without damping. The omission of tire damping is quite common in the quarter car model, as tires are generally surprisingly lightly damped; the tire is more accurately considered as an air spring than a rubber bushing. Despite its simplicity, the conventional model can be used to gain the most fundamental feature of the vehicle without greatly complicating the analysis. The vehicle is traveling at a constant speed, and by assuming that no slip or contact loss occurs between the wheel and the road profile, this results in a vertical vibrating motion of the vehicle with respect to its suspension system. The motion of the vehicle body and the wheel can be obtained using Equations 3.10 and 3.11 respectively.

The concept of the state of a dynamic system has been commonly used in analysis of engineering systems. In a given dynamic system, the state variables are the minimum set of physical variables that can completely describe the motion of the system. In order to use the state variables to describe a geometric space, the concept of state space is introduced in [23]. The equations of motion of the conventional model are derived in Equation 3.13. The vertical accelerations of the sprung mass and unsprung mass are obtained as following:

$$\ddot{z} \;=\; -\mathbf{M}^{-1}\mathbf{C}\dot{z} - \mathbf{M}^{-1}\mathbf{K}z + \mathbf{M}^{-1}\mathbf{F}u \tag{4.1}$$

Equations 3.10 and 3.11 are converted into a first order form, and then combined with Equation 4.1 in the state space form:

$$\begin{Bmatrix} \dot{x} \\ y \end{Bmatrix} \;=\; \begin{bmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{C} & \mathbb{D} \end{bmatrix} \begin{Bmatrix} x \\ u \end{Bmatrix} \tag{4.2}$$

The following definitions are required:

$$x \;=\; \begin{Bmatrix} z \\ \dot{z} \end{Bmatrix}, \quad y \;=\; \ddot{z} \tag{4.3}$$

The result is:

$$\begin{bmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{C} & \mathbb{D} \end{bmatrix} \;=\; \left[\begin{array}{cc|c} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} & \mathbf{M}^{-1}\mathbf{F} \\ \hline -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} & \mathbf{M}^{-1}\mathbf{F} \end{array} \right] \tag{4.4}$$

Generally, $\mathbb{A}$ is an $n \times n$ state matrix, $\mathbb{B}$ is an $n \times m$ input matrix, $\mathbb{C}$ is an $l \times n$ output matrix, $\mathbb{D}$ is an $l \times m$ direct transmission matrix, x is the state vector, u is the input vector, y is the output vector. Here, n, m and l are the number of state variables, input variables and output variables, respectively. In this research, the state variables are the displacements and the velocities of the sprung mass and unsprung mass, the input u is the road profile displacement, and the output y is the sprung mass and unsprung mass accelerations. The parameter values of the vehicle model are given in Table 4.1.

Table 4.1: Parameter values for the conventional model

Parameter	Value	Unit
Sprung mass (m_s)	500	kg
Unsprung mass (m_u)	50	kg
Suspension stiffness (k_s)	18000	N/m
Suspension damping coefficient (c_s)	1500	N·s/m
Tire stiffness (k_t)	150000	N/m

4.2 Control Design

In order to develop a simple method to recover the road profile inputs, a recovery algorithm based on the use of a controller is introduced in this section.

The state space of the conventional model can be shown in a block diagram, as shown in Figure 4.1. In this system, the road displacement is the input of the model, the accelerations of the sprung mass and unsprung mass can be obtained from the output.
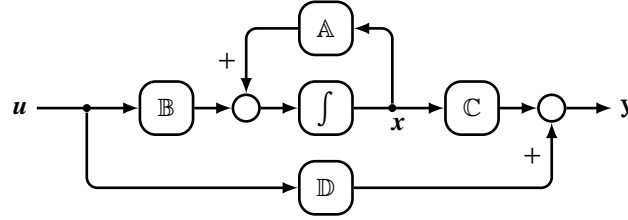

Figure 4.1: Vehicle model block diagram

In order to build a recovery algorithm, the control theory mentioned in Chapter 3 is used. The error e is defined as the difference between the measured accelerations r, and those predicted by the model:

$$e = r - y \tag{4.5}$$

As previously mentioned, in order to drive the error between this acceleration and the measured acceleration to zero, a feedback control is used. The controller is expressed by:

$$\dot{u} = -\mathbf{G}u + \mathbf{H}e \tag{4.6}$$

This equation can be rewritten by inserting the error equation:

$$\dot{u} = -\mathbf{G}u + \mathbf{H}r - \mathbf{H}y \tag{4.7}$$

Here, the selection of controller gains $\mathbf{G}$ and $\mathbf{H}$ are based on allowed convergence time and quality of fit respectively. The vehicle model and the controller can be joined into a combined model. Using the following notations:

$$x' = \begin{Bmatrix} x \\ u \end{Bmatrix}, \ u' = \{r\}, \ y' = \{u\} \tag{4.8}$$

The equations of motion of the combined model in state space form can be derived:

$$
\begin{bmatrix} \mathbb{A}' & \mathbb{B}' \\ \mathbb{C}' & \mathbb{D}' \end{bmatrix} = \left[\begin{array}{cc|c} \mathbb{A} & \mathbb{B} & \mathbf{0} \\ -\mathbf{H}\mathbb{C} & -\mathbf{G}-\mathbf{H}\mathbb{D} & \mathbf{H} \\ \hline \mathbf{0} & \mathbf{I} & \mathbf{0} \end{array} \right]
\tag{4.9}
$$

According to these equations, the combined model can be described using a system block diagram, which is shown in Figure 4.2. Clearly, if **G** and **H** are chosen such that the system is stable and has the appropriate time response, once the measured acceleration data is given as the input of the system, the road displacement can be obtained from the output.
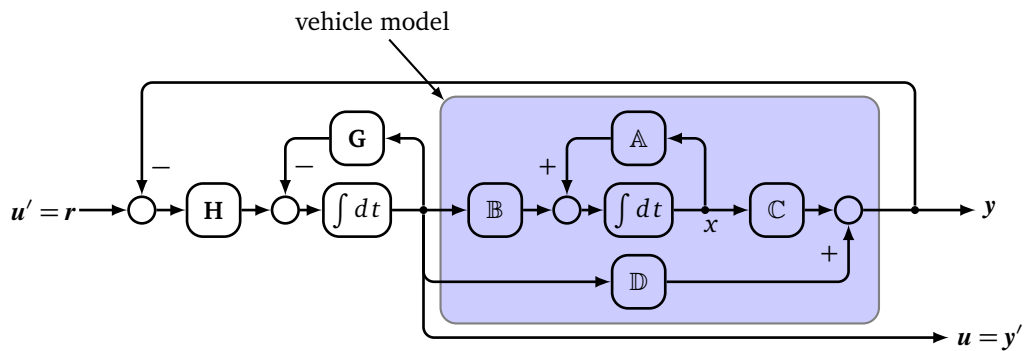
Figure 4.2: System block diagram

As described in the closed-loop system block diagram, a known road profile is used as the input of the simple vehicle model to generate accelerations of the sprung mass and unsprung mass, which is *r*, and then inserting the generated acceleration data *r* into the recovery algorithm to compare with the accelerations *y* generated from the recovered road profile acting negatively, the recovered road profile then can be obtained from the output of this system. Of course, the real suspension will not be modeled perfectly using a linear conventional model, and as a result there will be some discrepancy between the real road input and the recovered road input. However, the model used will ideally be computationally simple and efficient enough that it can run in real time on low cost hardware. There is a tradeoff increased computation effort from model complexity, and the fidelity of the results.

4.3   Optimization Algorithm

In this research, in order to obtain the road profile data, appropriate optimization methods are necessary, such that the **G** and **H** values keep the simulated output as close to the test input as possible. Two optimization methods, the genetic algorithm (GA), and the Nelder-Mead method (NM), are applied to get the best performance of the recovery algorithm. The genetic algorithm is a search method to generate

solutions to optimization and search problems that mimics the progress of natural evolution. The solving process starts with a randomly generated population, which are the candidate solutions to a problem. The fitness of all individuals in the population, which is normally the value of the objective function of the optimization problem to be solved, will be evaluated in an iterative process. In each generation, the selected candidate solutions are then used as parents to generate new individuals, guiding the search to the optimal solution. Usually, the algorithm will terminate after maximum number of iterations is reached, or the optimal solution is found. The optimal solution has been found if fitness ceases to improve.

The genetic algorithm is a good method for the optimization of a system without initial guiding values, as it can initialize with a random set of candidate solutions. If the initial values are known, other methods can be used to refine the system. Also, it is a good choice when parallel computing is an option.

The Nelder-Mead method is a simple method for searching for a local minimum of several variables in a system. It is a frequently used non-linear optimization approach. The method uses a simplex with $N + 1$ vertices in N dimensions to find the minimum of an objective function with N variables. The simplex can be a line segment, a triangle or a tetrahedron. Starting with the given initial points, this method generates a new test point by evaluating the behaviour of the objective function at each test point. The simplex converges to minimize the objective function from the chosen points by comparing the fitness, and finally returns the best result.

Although the algorithm is effective and less complicated, the need in computational resources increases significantly as the number of variables grows. Compared to the majority of classic numerical optimization methods, Nelder-Mead only uses the values of the objective function without any derivative information. The Nelder-Mead method does not benefit significantly from parallel processing.

Chapter 5

Model Implementation

The vehicle model and recovery method have been introduced in Chapter 4. The performance of the recovery model is based on the selection of the values of the controller gains **G** and **H**. In this chapter, in order to verify the robustness of the proposed approach, some computer models that accurately represent the vehicle are developed to estimate the road profile. The following sections will discuss the virtual modeling process. Initially, the conventional model will be coded and tested in MATLAB®/Octave to recover road profiles. Then in order to evaluate the performance of the method, simulations using complex multi-body dynamic models with non-linear parameters will be conducted in Altair MotionView®.

5.1 Modeling in MATLAB®

MATLAB® is a matrix-based technical computing language widely used by scientists and engineers. Due to its numerical computation, analysis, data visualization and programming capabilities, MATLAB® was considered for the initial simulations, as it provides various functions and toolboxes for users to realize the desired functionality through programming. Another reason is that it is easy to build a vehicle model and to determine the behaviour of dynamic loads on a road profile by using the MATLAB® based EoM software.

As with most MATLAB® programs, the recovery method is coded in the script format (m-file), and includes four main files. The first one is a virtual vehicle model file that has all the information of the model (coordinates, mass, stiffness, etc.). A simulation code structure of the conventional model (Figure 3.3) is built through this script. The second file is one of the optimization files. This script calls the GA function or NM function to address the objective function to get optimized variables. This process can be done by using the 'ga' or 'fminsearch' function in MATLAB®. The coded conventional model is also used in this file. If available, parallel computing functionality can be turned on to boost the optimization

process. A second optimization file is defined as a penalty function. It is a script file that defines the objective function of the optimization problem that will be used in the optimizer script. The last and most important file contains the core functions of the recovery algorithm. This file is used to test the usability of the optimized variables. The 'lsim' function is used to simulate the step response in this executable file. The MATLAB® codes can be seen in [24]. The process is drawn in Figure 5.1.
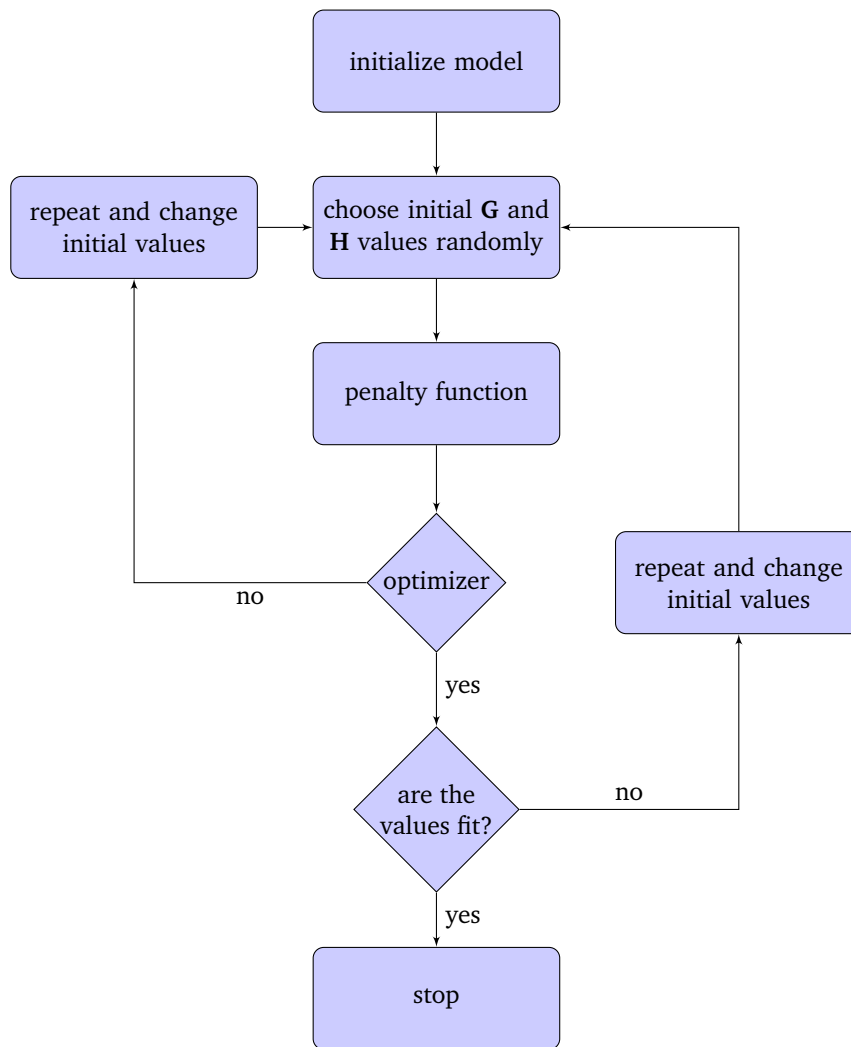
Figure 5.1: Flow chart of the simulation process in MATLAB®

In this project, using the vehicle parameter values from Table 4.1, a value of $\mathbf{G}= 2\text{ s}^{-1}$ is found to work well, through trials to evaluate the system performance, giving the recovery algorithm a time constant of 0.5 seconds. Therefore, the objective function is primarily used to determine the $\mathbf{H}$ values in order to optimize the quality of fit. For the conventional model, the $\mathbf{H}$ matrix only has two elements. Three

methods are used in the penalty function to get optimized values. One is based on the difference between the recovered road profile and the input road profile in the time domain. The second one is based on the stability of the recovery model. The last one is based on a comparison of the output and input in the frequency domain shown in the system Bode plot. The general penalty function is given by:

$$\Delta \; = \; \Sigma(f_{\text{desired}} - f_{\text{simulated}})^2 \tag{5.1}$$

The difference between the desired result and the simulated result is minimized by the optimizer, by selecting the fitted variable values. For more details about the scripts, refer to [24]. It is important to simulate this virtual model because it is a simple but effective means to gain a better understanding of the dynamic behaviour of the vehicle.

5.2   Modeling in Altair MotionView®

Because of its ability to model systems with non-linearities, such as bushings or bodies with large rotations, the standard computer aided engineering (CAE) software Altair MotionView® is considered. In addition, it has the advantage of a user friendly GUI (Graphical User Interface) and it can easily show the simulation animation for a better understanding of vehicle motions. Moreover, Altair MotionView® uses a default solver, called MotionSolve®, that provides many types of simulation, such as: transient, static, linear, etc. In order to verify the result in a more realistic test, corresponding 3D multibody models of the vehicle suspension are built in Altair MotionView®, with the linear data recovery algorithm running in parallel. Based on the structure of the vehicle suspensions, a simple linear suspension, a double wishbone suspension, and a MacPherson strut suspension are mounted in the quarter car model using this software.

For simplicity, the simple linear suspension model (for short, called linear model) is first introduced, as shown in Figure 5.2. This linear model includes a sprung mass, an unsprung mass, and a point mass body that represents the road. Three translational joints are defined in this model; two of them are placed between the three masses. The third joint connects the point mass and the ground. The sprung mass and the unsprung mass are also connected by a suspension system. The tire stiffness here is modeled as a linear coil spring without damping.

To represent the vehicle suspension more realistically, instead of positioning the sprung mass and unsprung mass in the uniform vertical direction, a double wishbone suspension (also called double A arm suspension, for short, called the A arm model) is built. It can be seen in Figure 5.3. This model is using two parallel wishbone shaped arms to connect the chassis of a vehicle with revolute joints. Each arm has one ball joint at the knuckle. A tie rod, part of the steering mechanism, connects to the knuckle and chassis by a ball joint and a constant velocity joint. In reality, a tie rod has ball joints at both ends.
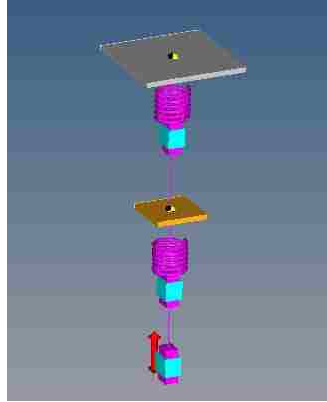
Figure 5.2: One DOF simple suspension model

This allows the tie rod to rotate around its own axis, and increases the DOF of the model by one. For simplicity, the constant velocity joint is used to avoid this effect. The wheel is fixed on the knuckle because only vertical motion is needed. The chassis has a translational joint with the ground to allow vertical movement. A coil spring and a damper connect the lower control arm to the chassis along the vertical axis. A bushing model is used to represent the tire stiffness.
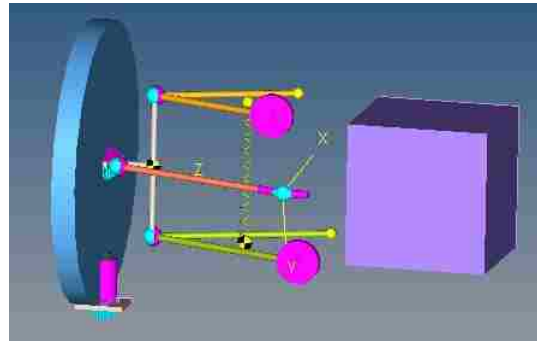
Figure 5.3: Double wishbone suspension model

To further verify the recovery algorithm, a quarter car model with a MacPherson strut suspension (for short, called MacPherson model) is built based on a model obtained from Chrysler. The MacPherson model shown in Figure 5.4 consists of a control arm, a tie rod, a knuckle, the vehicle body, a wheel, and a strut rod and strut tube, including a spring and damper. Normally, the control arm is connected to the vehicle body with a rotational joint. In this case, the control arm is connected to the vehicle body by two bushing models. A spherical joint connects the control arm and the knuckle. The wheel and the strut tube are installed on the knuckle. A cylindrical joint connects the strut rod to the strut tube. The strut rod is connected to the vehicle body by a universal joint with a compliant bushing. The spring and damper are mounted between the strut and the vehicle body along the strut tube to absorb the vibration and shock caused by a bumpy road. A road dummy body connects the wheel to the ground is introduced to define the wheel motion. Similarly to the A arm model, a bushing is used for the tire stiffness.
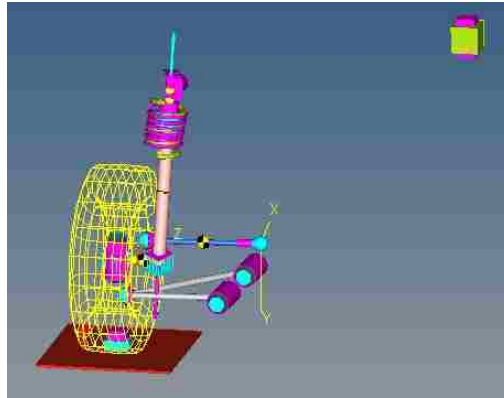
Figure 5.4: MacPherson strut suspension model

Since 3D quarter car models have been built, quarter car models associated with the recovery algorithm must be introduced. As shown in Figure 5.5, two linear models are built, which represent a corner model of the test vehicle and a recovery model respectively. These two models are connected by a differential equation (controller) with respect to the generated accelerations. More specifically, the model of the test vehicle is used to generate the sprung and unsprung mass accelerations. The recovery model, which uses the recovered road profile as the input, runs in parallel. The difference between the two sets of accelerations are inserted in the controller $\dot{u} = -\mathbf{G}u + \mathbf{H}r - \mathbf{H}y$, also running in parallel with the models, to get the recovered road profile. This whole recovery process is clearly shown in the modified system block diagram with two vehicle models running in parallel; see Figure 5.6. The error from the output is used in order to confirm whether the controller works as expected.
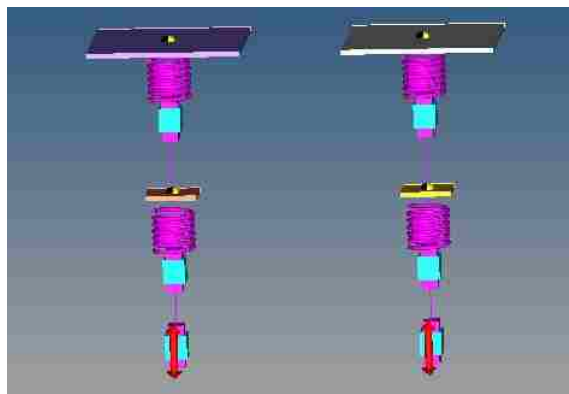
Figure 5.5: Linear model with linear recovery model

In order to study the performance of the proposed method, the model is modified to incorporate non-linear kinematics due to the geometry of the suspension mechanism, and a non-linear damper. Non-linearities in the tire are ignored in current research, but are source of future study. Figure 5.7 shows an A
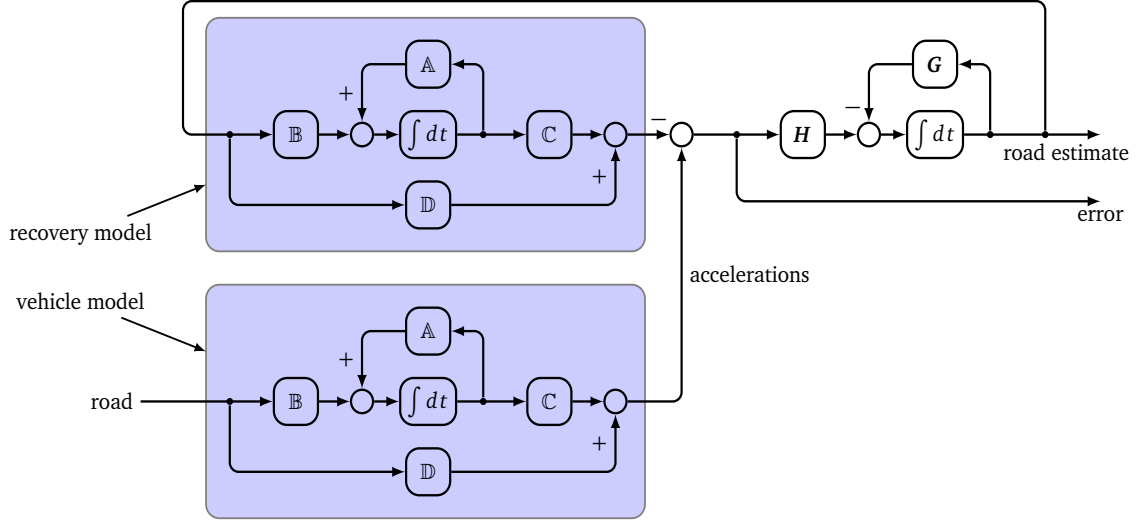
Figure 5.6: System block diagram with two parallel models

arm model with a non-linear damper connected to a simple linear model. The two models are connected by a differential equation using the recovery algorithm. Figure 5.8 shows a baseline force versus velocity curve for the dampers. As shown in the figure, the force provided by a damper is a function of the velocity of the damper movements. The dashed line shows a linear relationship between the damping force and velocity. The other basic curve has the same slope in compression and rebound. However, it is more realistic to have a steeper slope in rebound than in compression. Usually, less damping is needed when over a bump to let the wheel get back in contact with the ground quick enough, while more damping is required when over a hole to ensure proper reaction of the vehicle body.

In order to get an appropriate result, the parameters of the linear model must be set in accordance with the properties of the A arm model shown in Figure 5.7. A method to calculate the equivalent parameters based on the natural frequency of the vehicle body and the wheel is used. The natural frequency of the vehicle body and the wheel of the suspension model can be obtained by doing linear analysis in the MotionView® environment, noted ω_{body} (rad/s) and ω_{wheel} (rad/s). In this case, it is assumed that the tire stiffness does not change. The remaining parameters can be calculated using the following equations:

$$\omega_{\text{body}} \;=\; \frac{1}{2\pi}\sqrt{\frac{k_s k_t}{(k_s + k_t)m_s}} \; Hz \tag{5.2}$$

$$\omega_{\text{wheel}} \;=\; \frac{1}{2\pi}\sqrt{\frac{k_s + k_t}{m_u}} \; Hz \tag{5.3}$$

$$c_s \;=\; 2\zeta\sqrt{\frac{k_s k_t}{k_s + k_t}m_s} \; N \cdot s/m \tag{5.4}$$
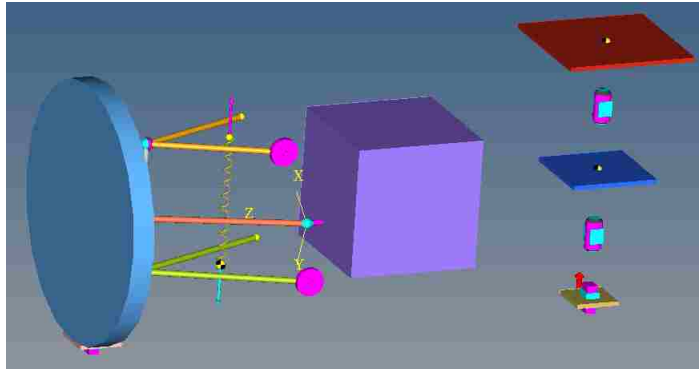
Figure 5.7: Non-linear model with linear recovery model
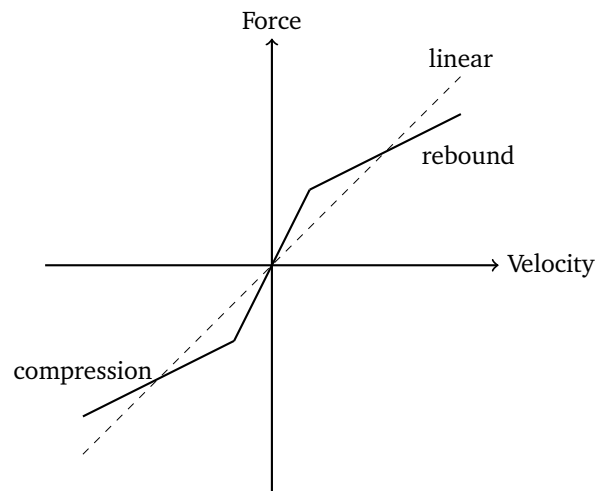
Figure 5.8: Force versus velocity curve for the dampers

It is very easy and fast to implement the numerical calculation in the MATLAB® environment. An optimization script is also developed to verify the fitness of the equivalent parameters. According to the requirement of accuracy, users can choose the number of unknowns to be determined. The scripts are given in Appendix A.

Chapter 6

Simulation Results

The implementation of vehicle models has been discussed in Chapter 5. Two software tools are used to model the dynamics of the vehicle in order to recover the road profile. Both linear and non-linear models are described. Results in this chapter will show the verification of the efficiency of the proposed recovery solution, carried out using MATLAB® and Altair MotionView®. Simulation trials using a half car model and full car model in MATLAB® are shown in Appendix B.

6.1 Simulated Road Profiles

To show the efficiency of the proposed method, simulations are completed by using five types of road profiles in order to examine the performance of a wider range of road roughness classes on the proposed recovery algorithm. They are first used to obtain the body and wheel accelerations in the vertical direction. These data are then used as the input of the recovery system to recover road profiles.

6.1.1 Generated Road Profiles

In order to test the algorithm, the road profile needs to be used as a baseline to compare the recovered road profile. Based on the classification of road profiles, several random profiles are generated from their PSDs. The details of methodology can be found in [4]. Using the theory, a function is developed in MATLAB® that generates a random road profile from a given road profile class. During the simulations, road profiles with a length of 250 metres are chosen. Figure 6.1 shows the realization for the standard road classes classified by ISO 8606. Driving the vehicle model at 10 m/s, ISO classes A, B-C, C-D, and D-E road profiles are generated in the time domain for the convenience of computer simulation. A sampling frequency of 512 Hz is used in all simulations. The road profiles vary from very good (A class, with $h_{max} \approx \pm 15$ mm) to poor (D-E class, with $h_{max} \approx \pm 100$ mm). The profiles contain content that is spread over a span of wavelengths, with the amplitude decreasing with the wavelength. These randomly

generated road profiles are used as the input of the quarter car models introduced in previous chapter. Results will be discussed in the following sections.
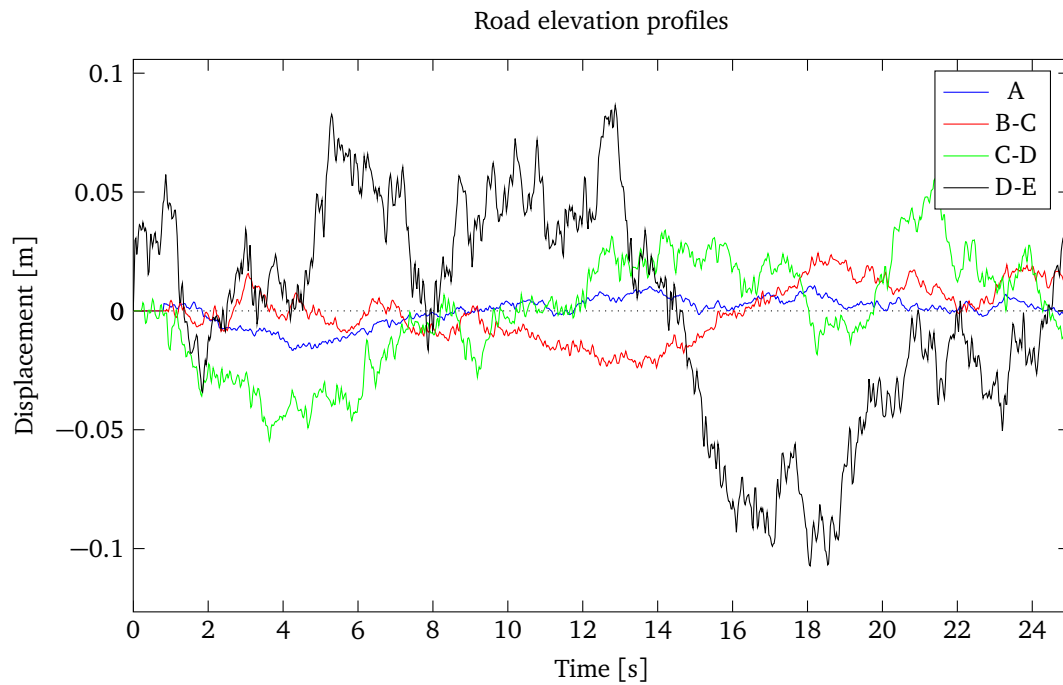
Figure 6.1: Random road profiles realization of ISO road classes

Figure 6.2: Road profile measurement instrument

6.1.2   Measured Road Profile

This thesis project is mainly working on the recovery of a 1.56 kilometre testing road profile provided by Chrysler, called the Chalma road profile. Figure 6.2 shows the device that is used to measure road profile

elevations. Similar to the aforementioned GMR profilometer, this instrument uses an accelerometer to measure vertical vehicle body movement, and a laser sensor for the measurement of the displacement between the vehicle body and the ground. The road profile is equal to the body movement plus the appropriate body-road displacements.
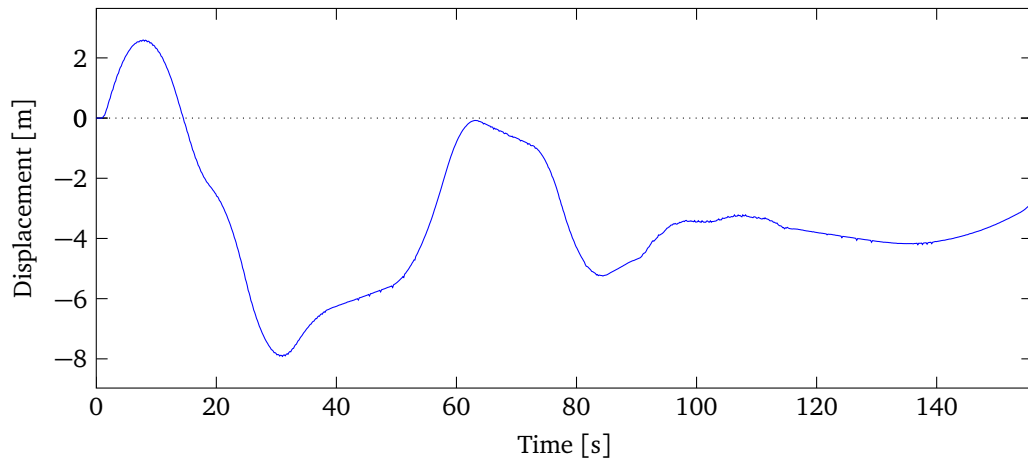
Figure 6.3: Simulated Chalma road profile
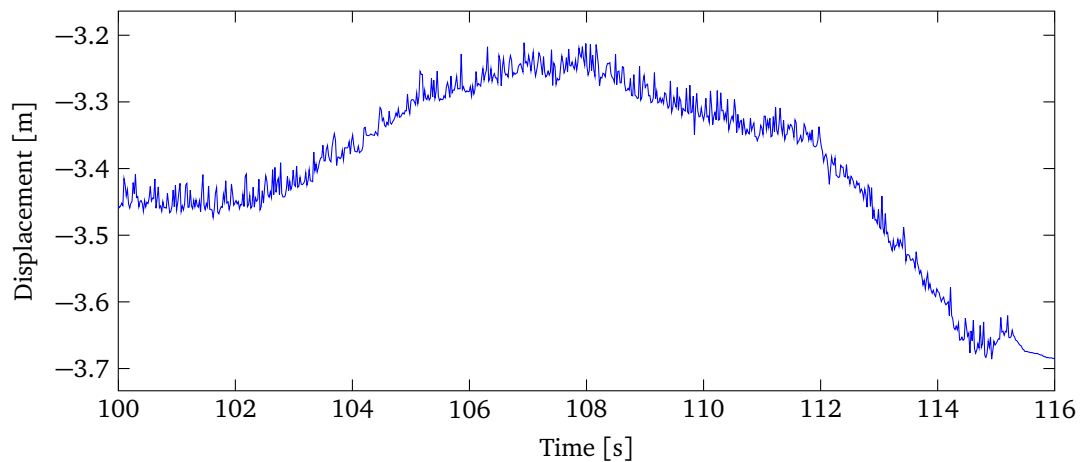
Figure 6.4: Zoomed Chalma road profile

Figure 6.3 shows the Chalma road displacements over 156 seconds with a sampling frequency of 512 Hz; one of the high frequency parts of the Chalma road profile is plotted in Figure 6.4. The roughness of the Chalma road profile and the A class road profile are drawn in Figure 6.5 according to the ISO 8606 classification. It is clear that the Chalma road profile is mostly in the G class range, which is a very aggressive road profile. Also, it is straightforward that the PSDs of the other three random road profiles will be located on their corresponding definition line.
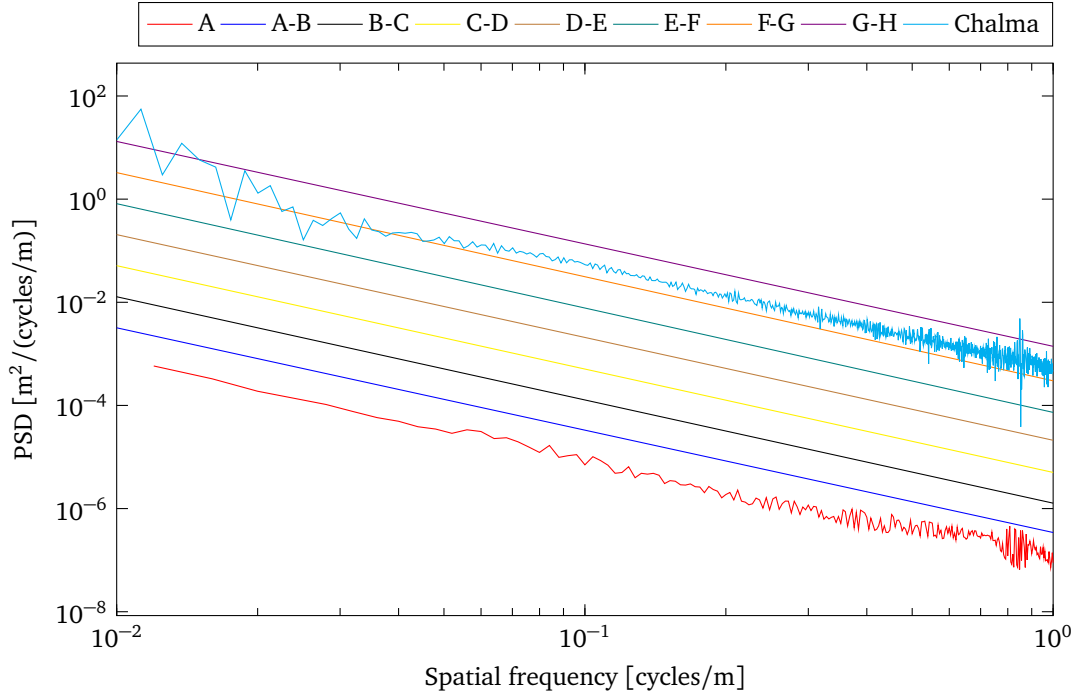
Figure 6.5: PSD classification of road profiles from ISO 8606

6.2 Road Recovery Results

Based on the models used in this research, the simulation results can be divided into two groups. One is the results obtained from a linear recovery model, the other group is the results generated from a nonlinear recovery model. For each of road profiles, the virtual computer vehicle model is run at 10 m/s, and the acceleration responses of the body and wheel are used as the input of the recovery algorithm to recover simulated road profiles.

6.2.1 Results from Linear Recovery Models

Three different recovery cases are defined to recover road profiles. First, a virtual conventional model based on the system block diagram shown in Figure 4.2 is simulated using MATLAB® codes. Based on the vehicle model parameter values shown in Table 4.1, controller gains values can be optimized by running the optimization code, using the first type of penalty function described in Chapter 5, which is $\mathbf{H} = [8.998 \cdot 10^6 \ 1.075 \cdot 10^6] \ \text{s}^{-1}$. Applying the randomly generated road profiles as the input of the conventional model, results indicate that the recovered road profiles are very close to the simulated road profiles, see Figure 6.6 to Figure 6.9. The differences between the recovered and simulated road profiles are also shown in the figures. Results show similar high levels of accuracy from the difference plots.
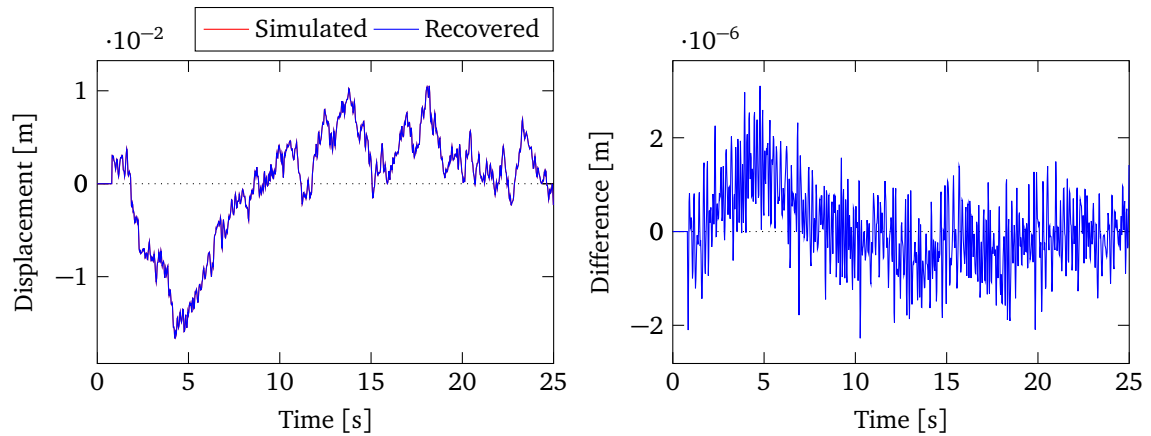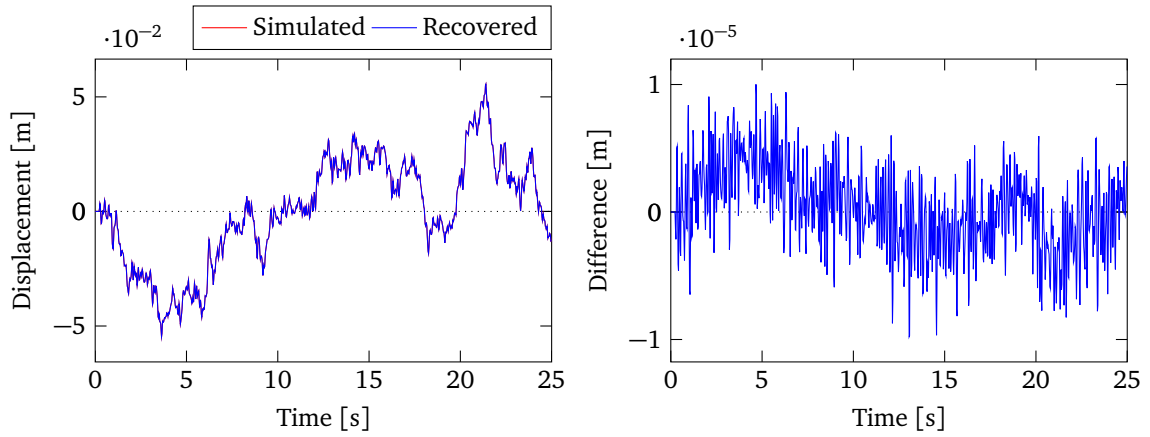
Figure 6.6: Recovered and simulated ISO A class road profiles and the difference in MATLAB®
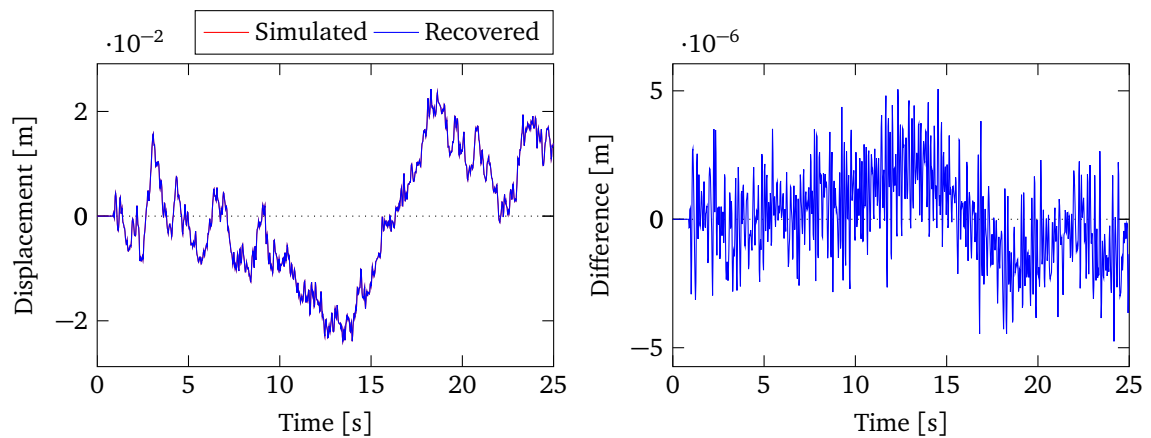
**Figure 6.7:** Recovered and simulated ISO B-C class road profiles and the difference in MATLAB®

When it comes to the Chalma road profile recovery, vertical accelerations of the sprung mass and unsprung mass obtained from the conventional model using the Chalma road as the input signal are shown in Figure 6.10. Employing the accelerations as the input of the recovery system, using the same **H** values, the recovered road profile is almost the same as the original road profile, as shown in Figure 6.11. The corresponding PSDs of the two profiles are shown in Figure 6.12; the curves are superimposed.

Figure 6.8: Recovered and simulated ISO C-D class road profiles and the difference in MATLAB®
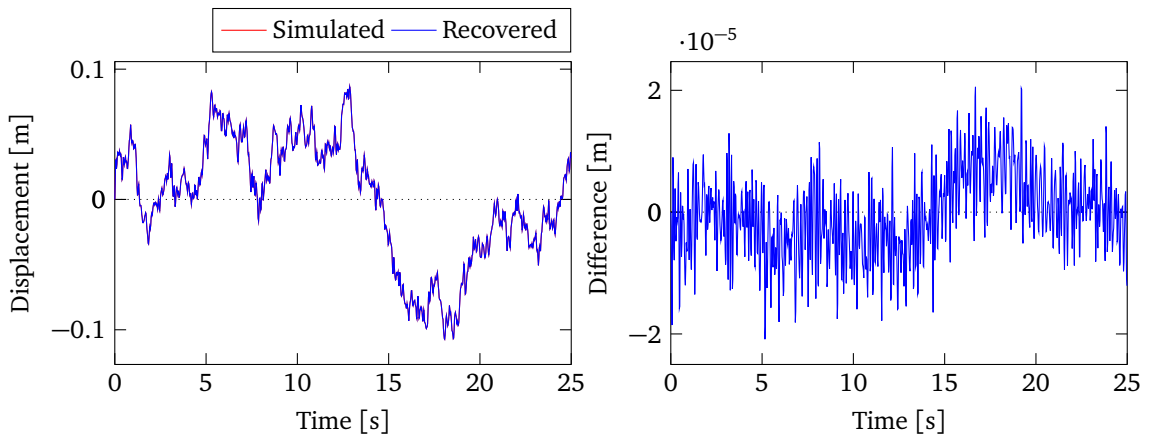
Figure 6.9: Recovered and simulated ISO D-E class road profiles and the difference in MATLAB®
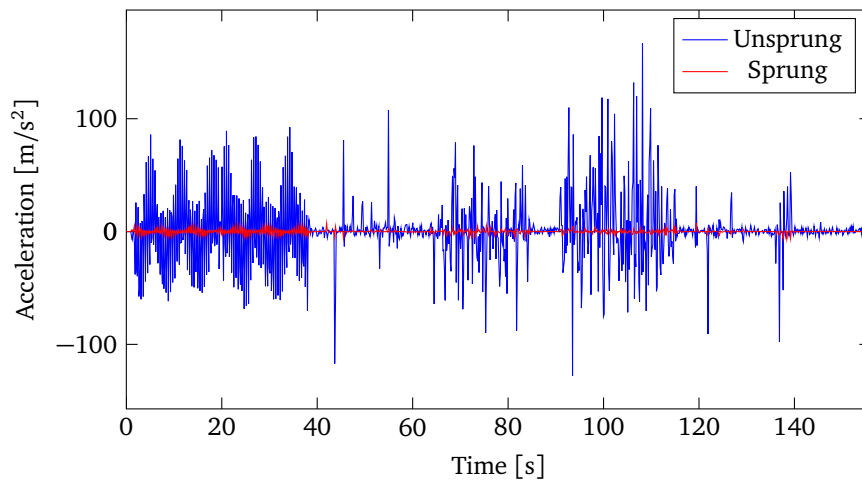
Figure 6.10: Accelerations of the sprung mass and unsprung mass from Chalma in MATLAB®
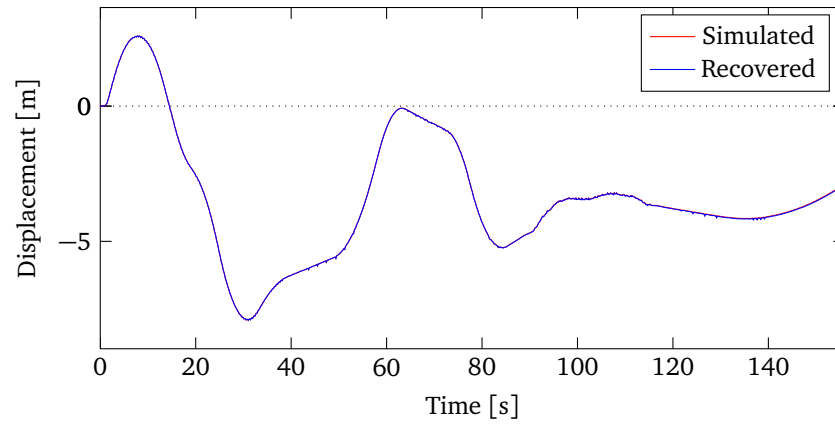
Figure 6.11: Recovered and simulated Chalma road profiles in MATLAB®
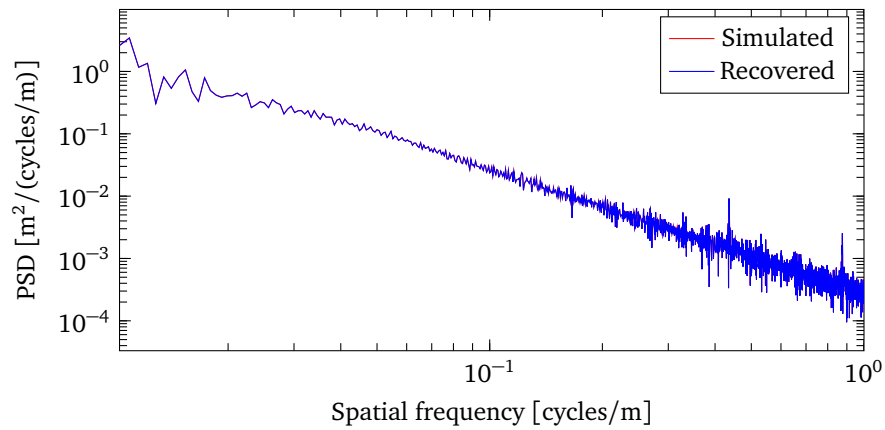
Figure 6.12: PSDs of recovered and simulated Chalma road profiles using MATLAB® model
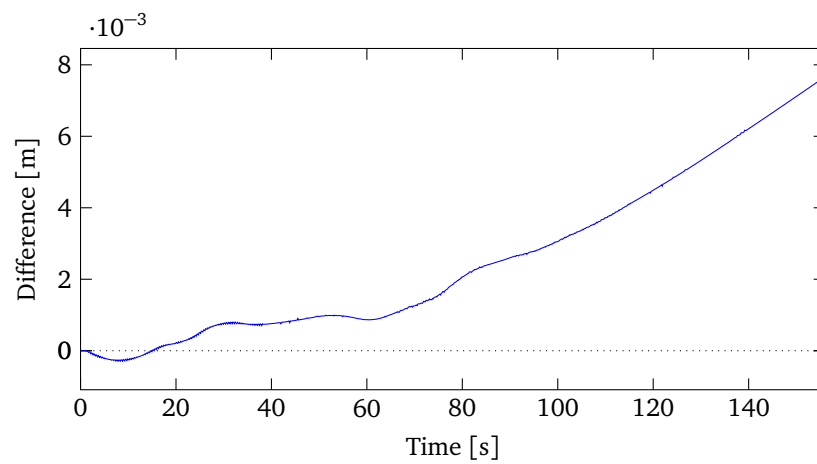
Figure 6.13: Difference between recovered and simulated Chalma road profiles in MATLAB®

Figure 6.13 shows the difference between the recovered and simulated road profile. The maximum value of the difference is below $8 \cdot 10^{-3}$ m. When compared to the road profile elevations, this result shows excellent accuracy. In order to show the recovered plot clearly, Figure 6.14 shows four zoomed sections of the recovery result, which still show very good agreement on the two profiles.
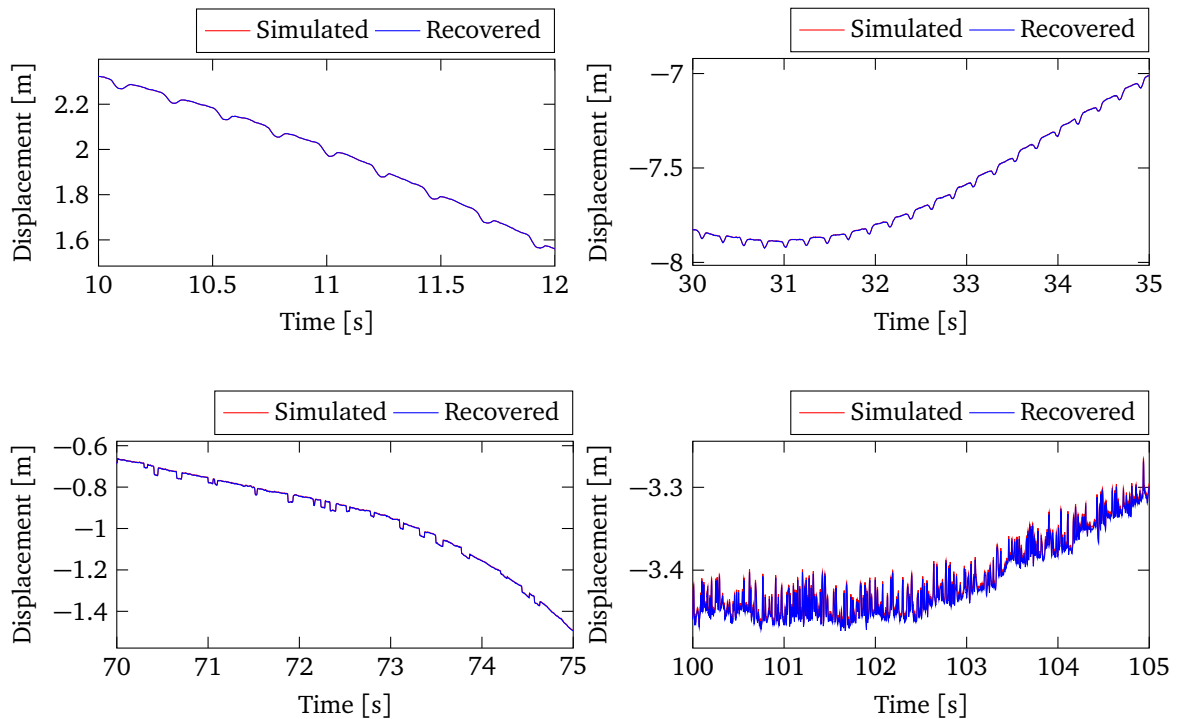
Figure 6.14: Zoomed sections of the recovered and simulated Chalma road profiles

Figure 6.15 shows the Bode diagram of the recovery system at frequencies from $0.01 \sim 100$ Hz. When the frequency is at 0.015 Hz, the magnitude is approximately 0.01 dB. Correspondingly, the transfer function of the system is $H(j\omega) = 1.0011$, which means the output is almost the same as the input. Therefore, the frequency response shows that above 0.015 Hz, the recovered road matches the input road very well (as 0 dB means the input is equal to the output).

Of course, when using the same linear model to both generate the accelerations, and recover the displacement input, one would expect very good agreement. In order to validate the simulation results, a corresponding multibody model representing the virtual conventional model is then built in Altair MotionView®. As shown in Figure 5.5, two linear models are simulated, with the linear data recovery algorithm running in parallel. The methodology of the road profile recovery algorithm presented in Chapter 4 allows for the simultaneous estimation of the road profile displacement. Simulation results of the random road profiles are plotted from Figure 6.16 to Figure 6.19. Results show that the recovered road profiles are also very close to the simulated ones.
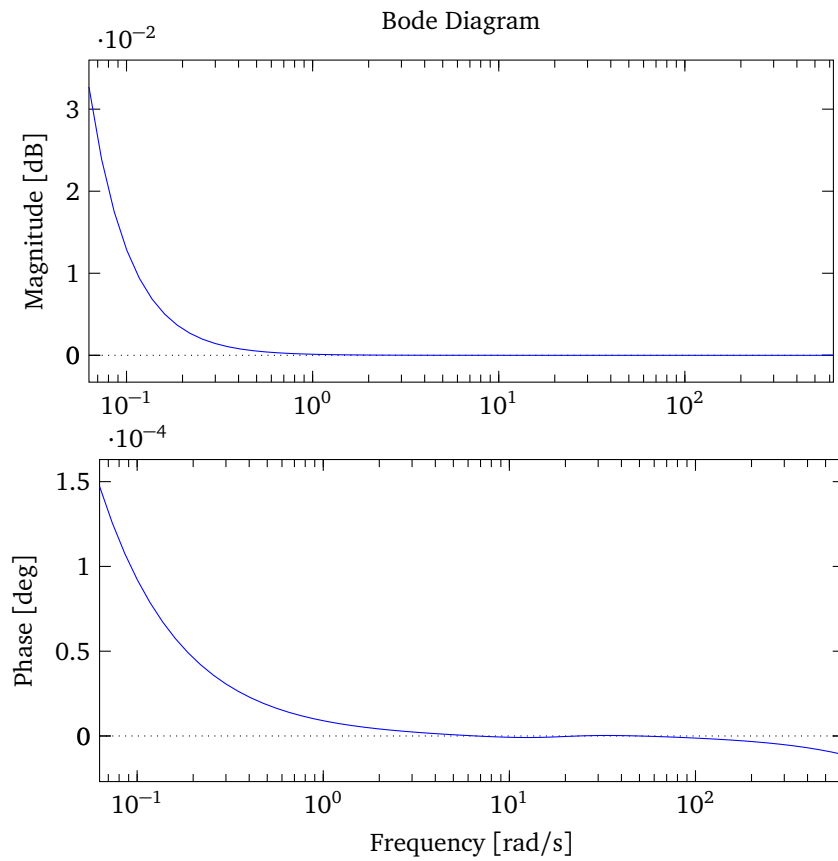
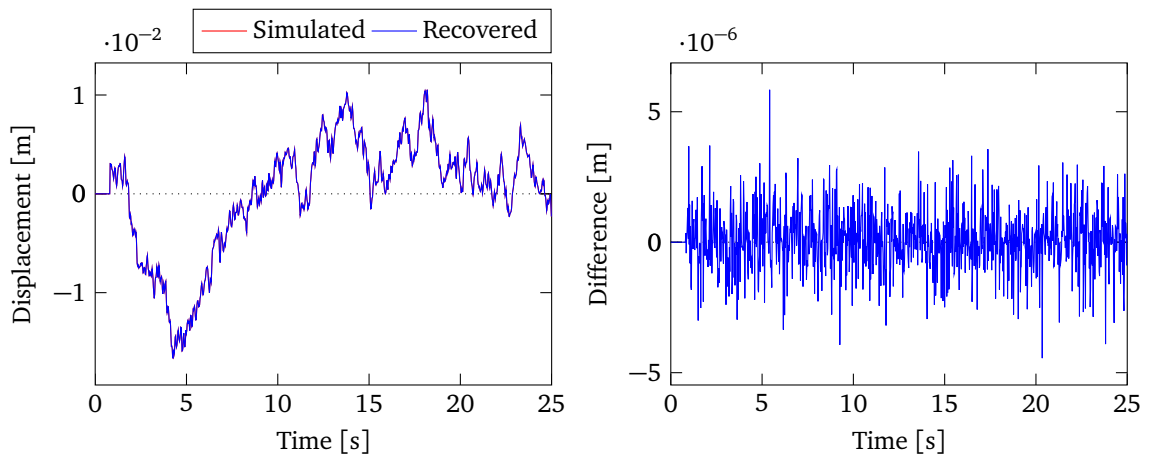**Figure 6.15:** Bode diagram of the recovery system

Figure 6.16: Recovered and simulated ISO A class road profiles and the difference using two linear models in Altair MotionView®
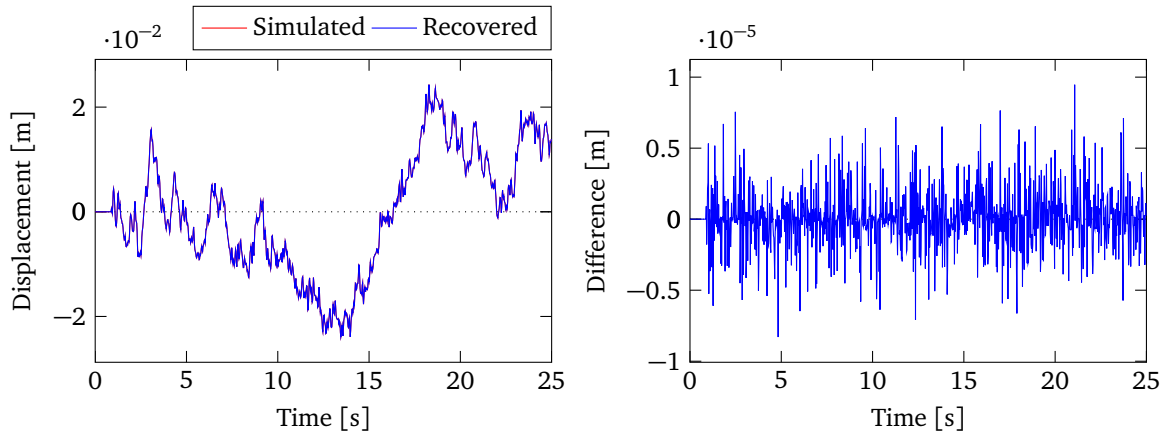
Figure 6.17: Recovered and simulated ISO B-C class road profiles and the difference using two linear models in Altair MotionView®
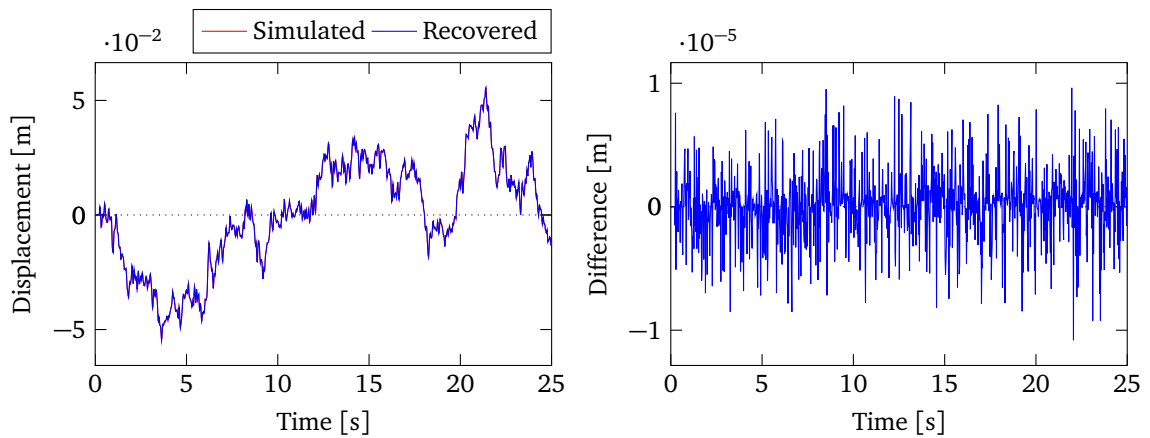

Figure 6.18: Recovered and simulated ISO C-D class road profiles and the difference using two linear models in Altair MotionView®
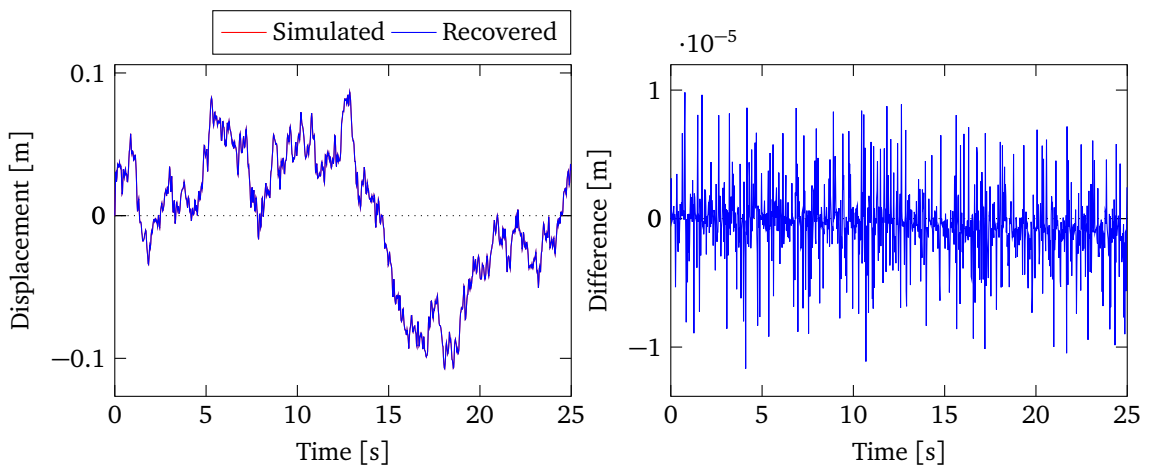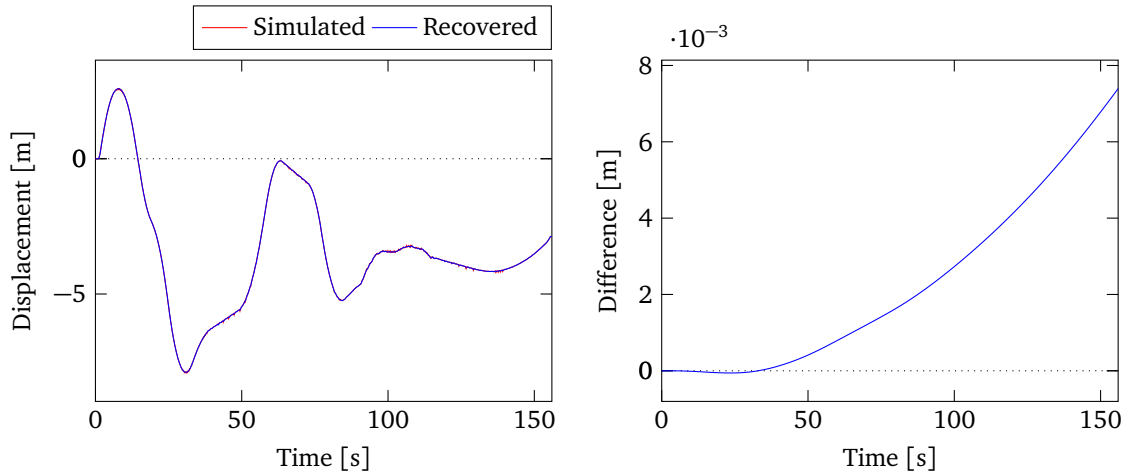

Figure 6.19: Recovered and simulated ISO D-E class road profiles and the difference using two linear models in Altair MotionView®

Figure 6.20 shows the simulation results of the Chalma road profile. The left side figure of Figure 6.20 shows the recovered road profile is the same as the simulated road profile. From the right side figure, when compared to the simulated road profile, the result is as good as the result in MATLAB®, in which the maximum difference is 8 mm. As can be seen in Figure 6.21, the PSD of the recovered road profile matches the original one almost exactly. With two of the same linear models running together, no doubt the results will be very good if the **H** values are appropriate.

Figure 6.20: Recovered and simulated Chalma road profiles and the difference using two linear models in Altair MotionView®
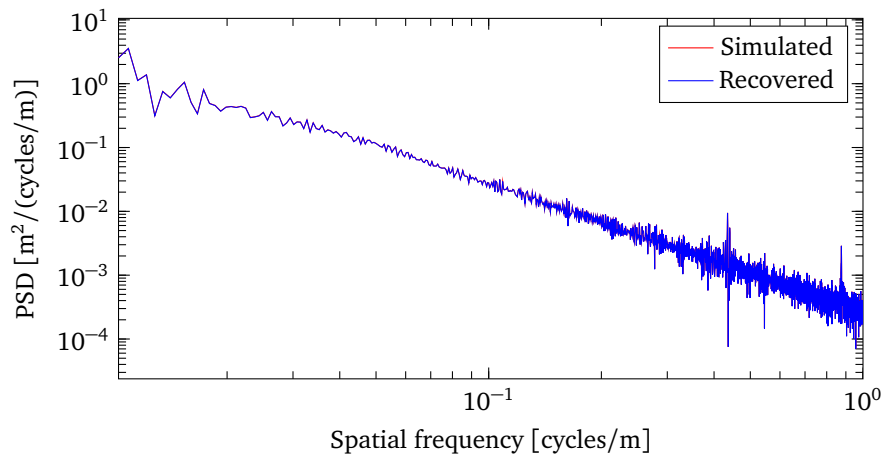
Figure 6.21: PSDs of recovered and simulated Chalma road profiles using two linear models in Altair MotionView®

Last, an A arm model with a linear recovery model running together is utilized to verify the results in a more realistic simulation, see Figure 5.7. As mentioned in Chapter 5, the parameters of the linear recovery model must match the A arm model properties. Therefore, after analysing the natural frequency

of the suspension, equivalent parameters are given in Table 6.1. It is noted that the **H** values are modified when adapting to the non-linear model. The damper force-velocity curve used in this model is shown in [24].

Table 6.1: Equivalent parameter values for the linear recovery model

Parameter	Symbol	Value	Unit
Body natural frequency	ω_{body}	0.853	Hz
Wheel natural frequency	ω_{wheel}	8.209	Hz
Sprung mass	m_s	450	kg
Unsprung mass	m_u	61.8	kg
Suspension stiffness	k_s	14130	N/m
Suspension damping coefficient	c_s	892	N·s/m
Tire stiffness	k_t	150000	N/m
Controller gain	**H**	$[1.075 \cdot 10^7 \ 0.094 \cdot 10^7]$	s^{-1}

Figure 6.22 to Figure 6.25 display the random road profiles recovery plots and the difference plots. With non-linearities involved, as the roughness of the road profile is getting worse, the accuracy is decreasing.
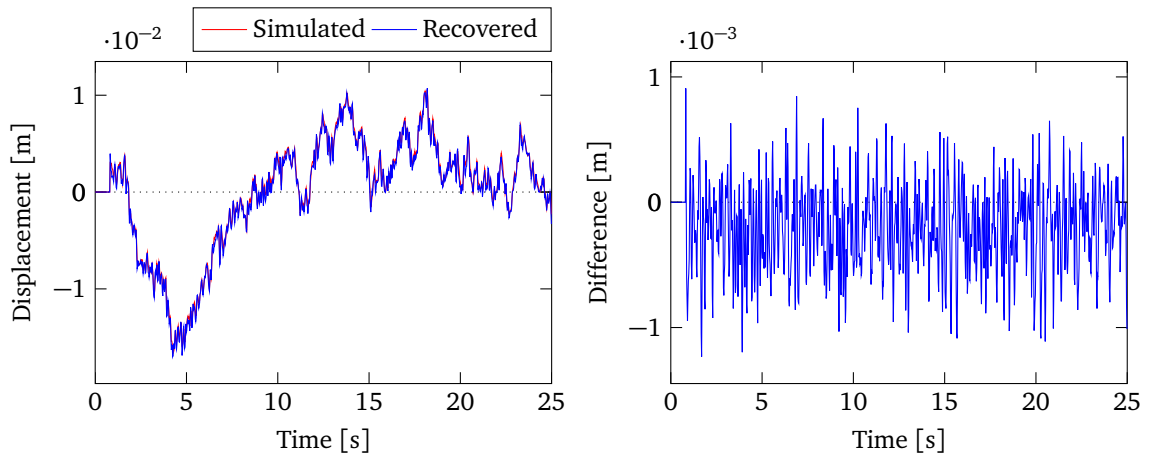

Figure 6.22: Recovered and simulated ISO A class road profiles and the difference using a non-linear model with linear recovery model in Altair MotionView®
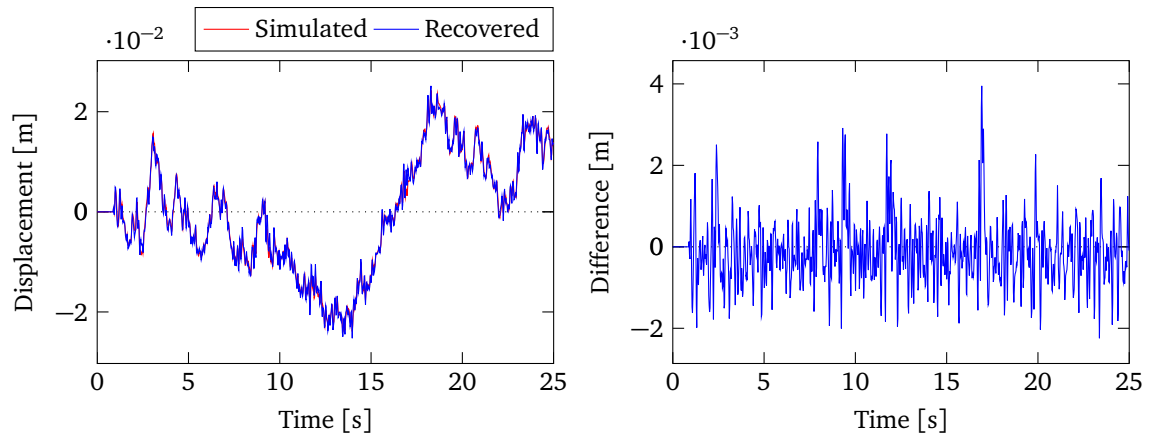
Figure 6.23: Recovered and simulated ISO B-C class road profiles and the difference using a non-linear model with linear recovery model in Altair MotionView®
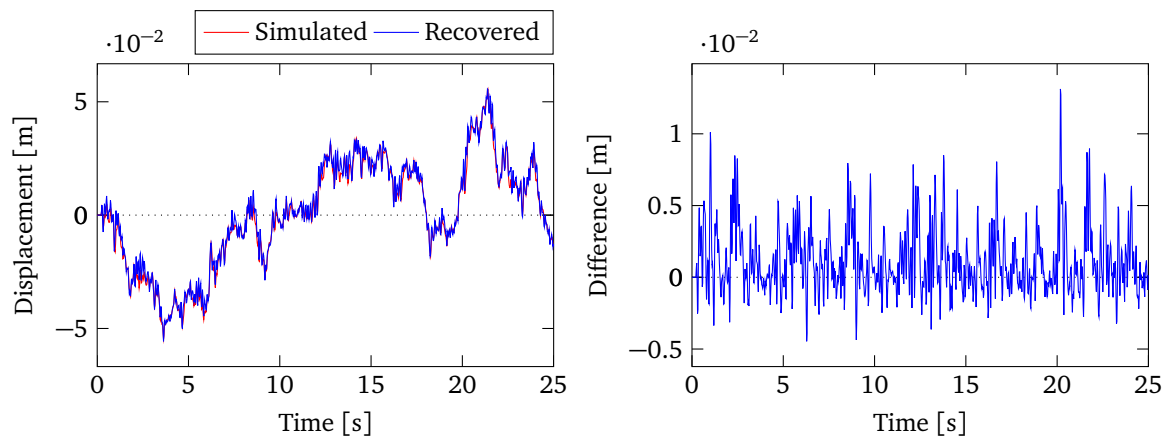

Figure 6.24: Recovered and simulated ISO C-D class road profiles and the difference using a non-linear model with linear recovery model in Altair MotionView®
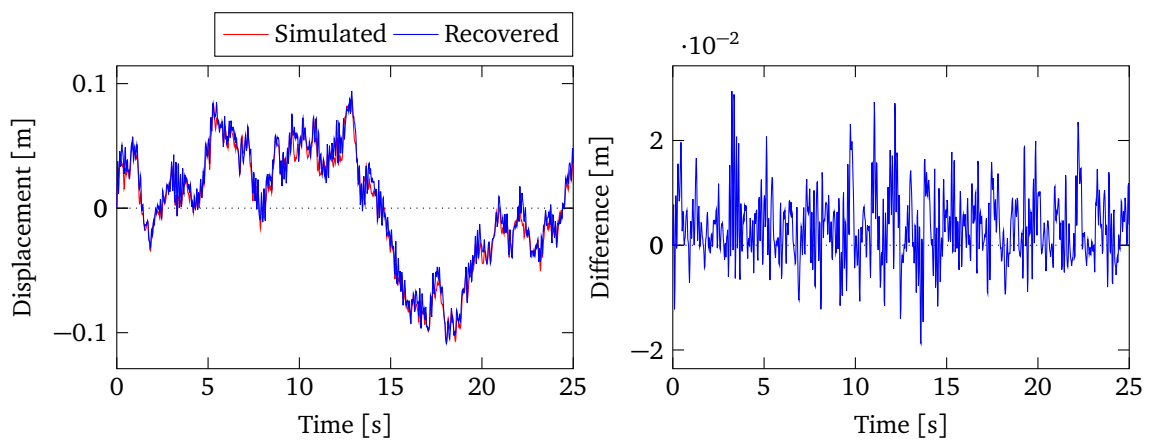

Figure 6.25: Recovered and simulated ISO D-E class road profiles and the difference using a non-linear model with linear recovery model in Altair MotionView®
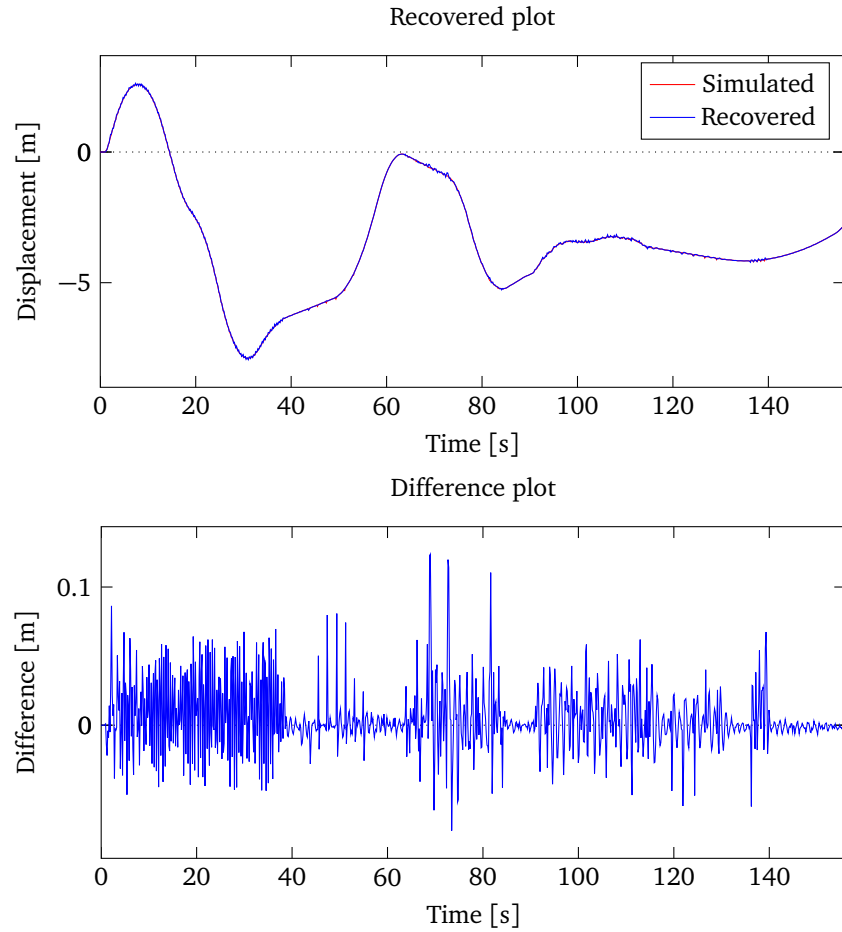
Figure 6.26: Recovered and simulated Chalma road profiles and the difference using a non-linear model with linear recovery model in Altair MotionView®
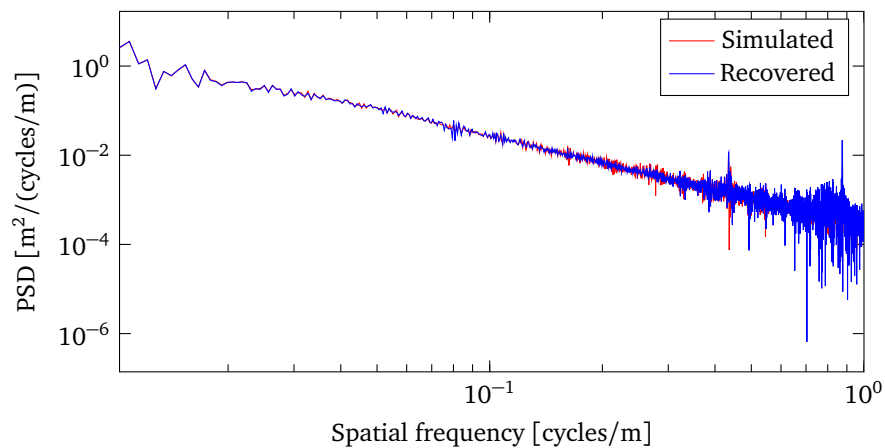

Figure 6.27: PSDs of recovered and simulated Chalma road profiles using a non-linear model with linear recovery model in Altair MotionView®

Simulation results of the Chalma road recovery are shown in Figure 6.26 and Figure 6.27. It can be seen that the non-linearities cause a deterioration of accuracy, which is not as good as the result from using two linear models. It is clear from the PSDs of the two profiles in Figure 6.27 that some of the higher frequency profiles are not recovered. As expected, it is very difficult to use a linear model to regenerate the motion of a non-linear model, even though the two models are similar.

It is worth noting that the two software tools use two different solvers. The default solver in Altair MotionView®– MotionSolve®– uses differential algebraic equations (DAEs) to solve dynamic problems, while the MATLAB® standard solver is developed for solving ordinary differential equations (ODEs). A discrete time matrix exponential method is implemented for efficient computation in MATLAB®. In addition, the integrator used by MotionSolve® for this research is DSTIFF. The parameters used for the integrator need to be chosen to ensure minimum error in each step. The values and meanings of the parameters used are given in Table 6.2.

Table 6.2: MotionSolve® integrator parameter values and meanings

Parameter	Value	Meaning
Integration tolerance	$1 \cdot 10^{-4}$	maximum error allowed
Max stepsize	$1/512$	maximum sampling step size
Min stepsize	$1 \cdot 10^{-6}$	minimum sampling step size
DAE index	3	choose constraint from the index of the solver
DAE constraint tolerance	$1 \cdot 10^{-5}$	tolerance of the differential algebraic equations
DAE max iterations	4	maximum numbers of iterations allowed
DAE min iterations	1	minimum numbers of iterations allowed

6.2.2 Results from Non-linear Recovery Models

Often, the accuracy of simulation results is improved by adopting a more sophisticated and realistic model because of the improved similarity to reality. Instead of employing a linear recovery model to recover road profiles, two non-linear recovery models, connected through the differential equation of the proposed controller, are built.

As displayed in Figure 6.28, the non-linear recovery model is exactly same as the model that represents the test vehicle. The two models are expected to have the same motion if they are given the same input road profile. As before, the left model is used to generate acceleration data, then the accelerations are compared with the accelerations generated from the right model, which are driven by the road profiles recovered by running the recovery algorithm in parallel, using the **H** values from Table 6.1. Simulation

results of the random road profiles are shown in the following plots from Figure 6.29 to Figure 6.32. Because the two models have the same properties, the results show good accuracy as expected on the random road profiles recovery.
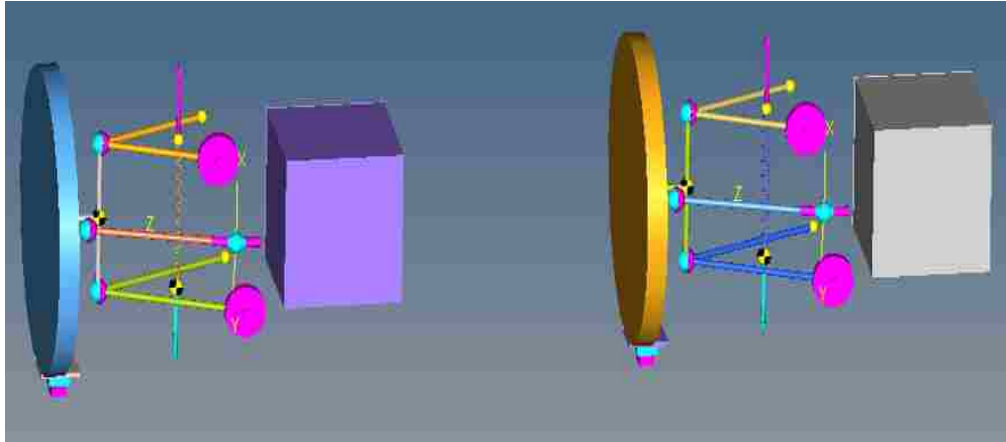

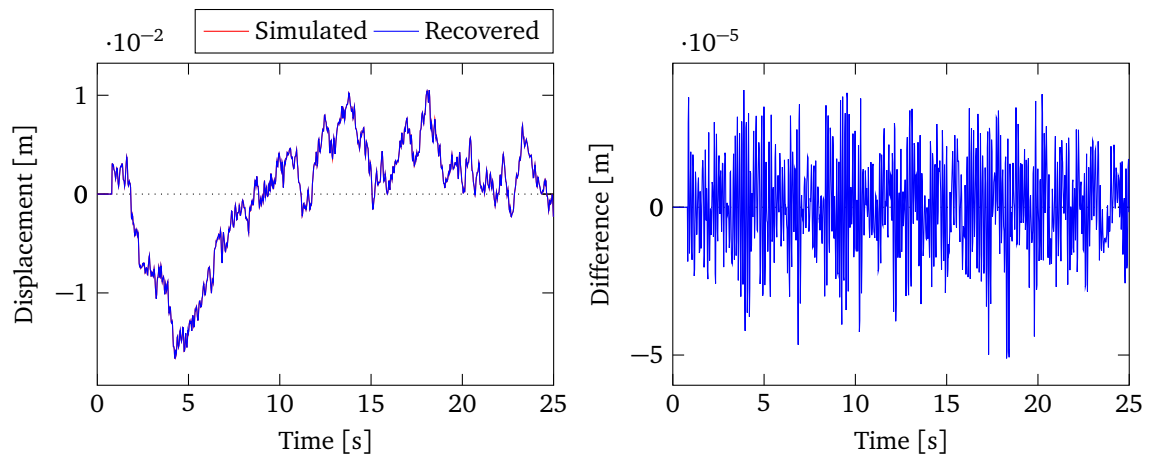**Figure 6.28:** Two A arm models running together in Altair MotionView®

Figure 6.29: Recovered and simulated ISO A class road profiles and the difference using two A arm models in Altair MotionView®
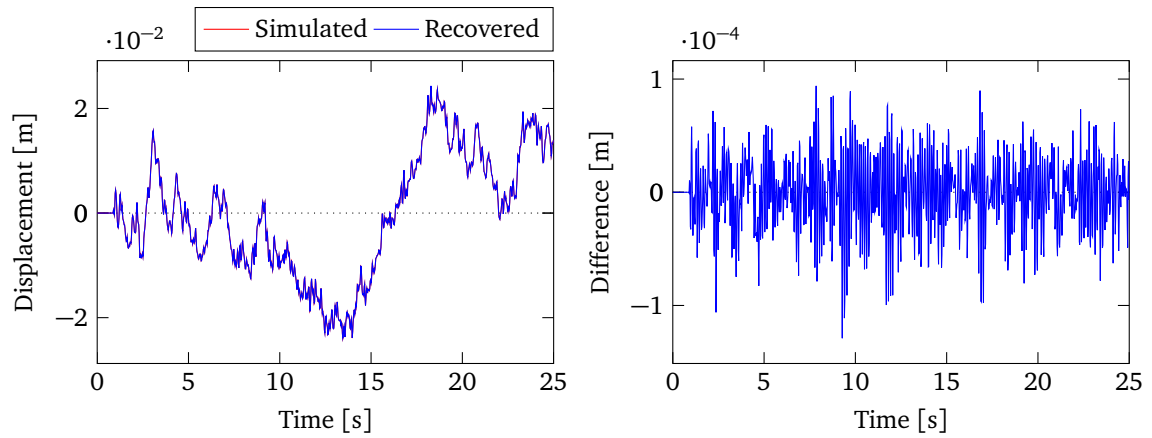
Figure 6.30: Recovered and simulated ISO B-C class road profiles and the difference
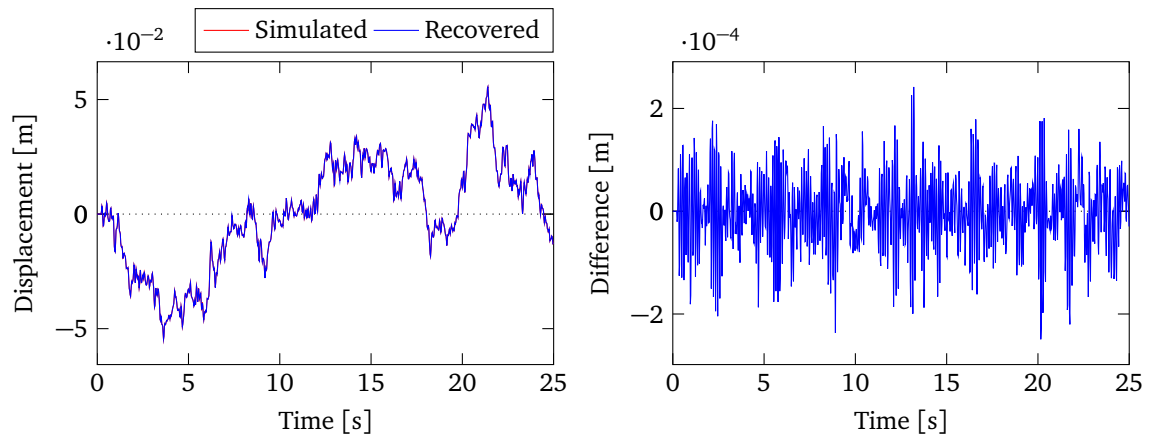using two A arm models in Altair MotionView®

Figure 6.31: Recovered and simulated ISO C-D class road profiles and the difference
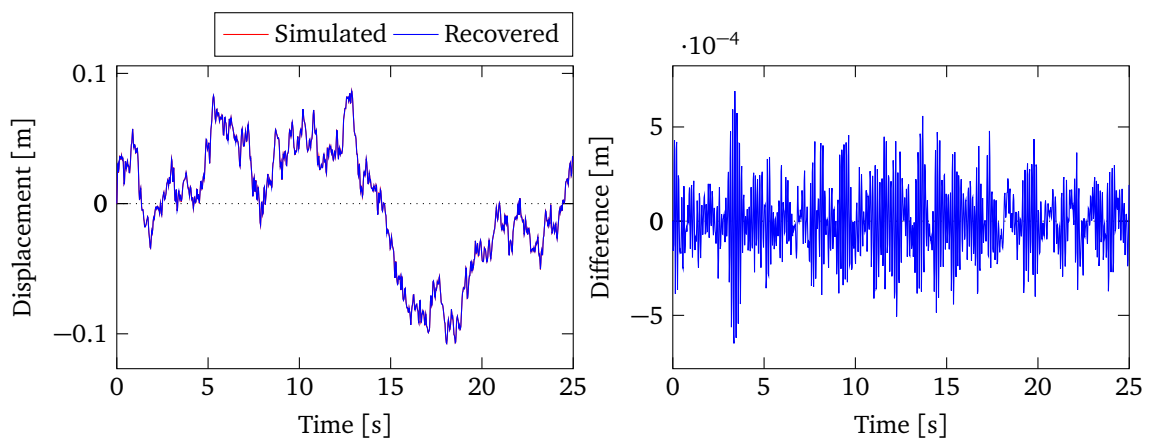using two A arm models in Altair MotionView®

Figure 6.32: Recovered and simulated ISO D-E class road profiles and the difference
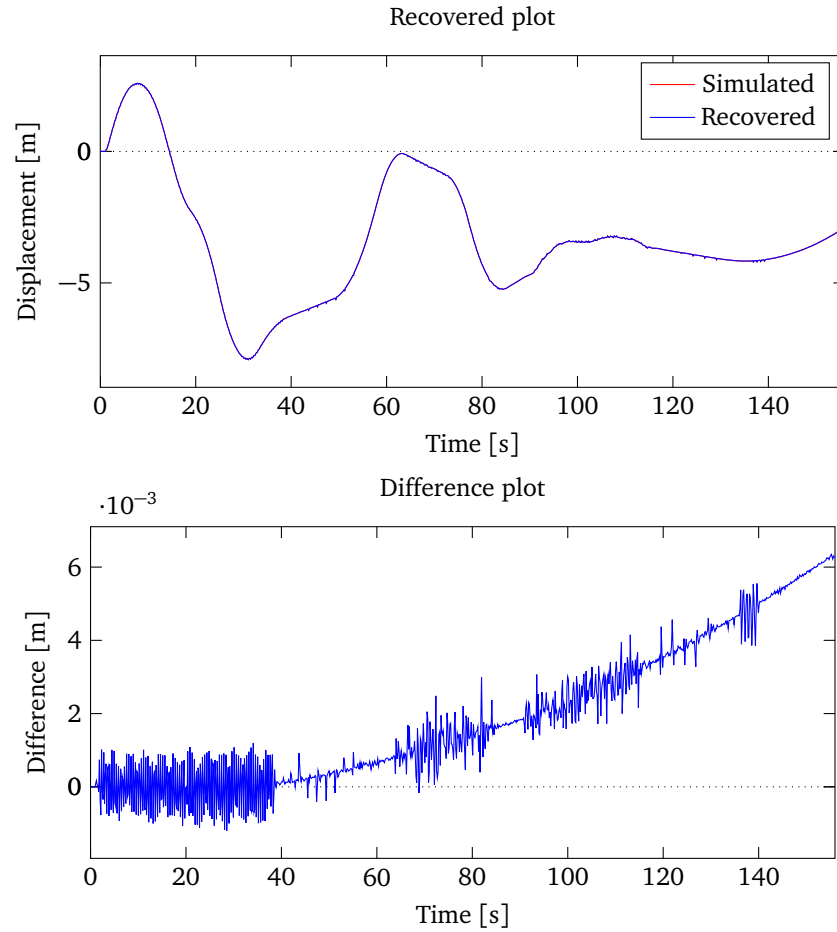using two A arm models in Altair MotionView®

Figure 6.33: Recovered and simulated Chalma road profiles and the difference using two A arm models in Altair MotionView®
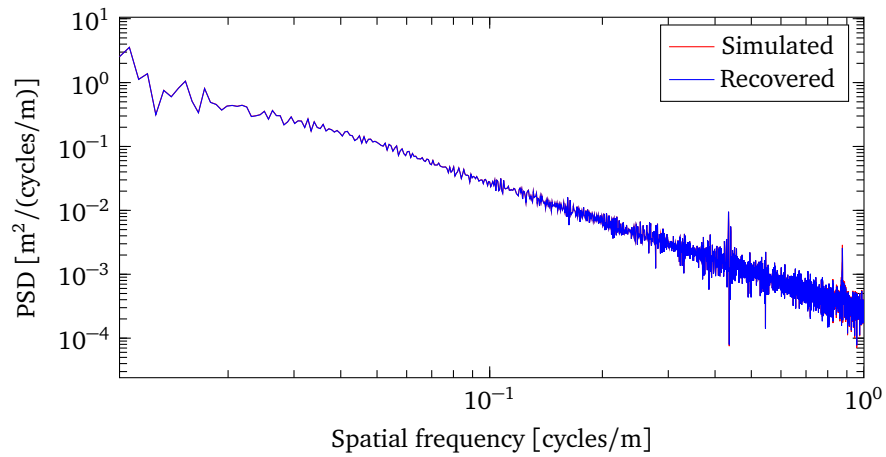

Figure 6.34: PSDs of recovered and simulated Chalma road profiles using two A arm models in Altair MotionView®

Simulation results of the Chalma road profile are shown in Figure 6.33 and Figure 6.34. According to the plots shown in Figure 6.33, results show that the difference between the recovered Chalma road displacement and simulated displacement is below 7 mm. It can be observed that the first quarter of the simulated Chalma road profile can be accurately predicted. However, starting at 40 s, there is a gradual drifting effect in the recovered road profile, increasing with distance from the zero axis as shown in the difference plot. This effect cannot be avoided when dealing with errors in the estimation of the measured accelerations. To explain this phenomenon, the poles of the test model and the recovery system are given in Table 6.3 and Table 6.4.

Table 6.3: Poles of the recovery system using two A arm models

Pole	Value
s_1	$-2.29 \cdot 10^9$
$s_{2,3}$	$-12.28 \pm 15.44i$
$s_{4,5}$	$-0.0040 \pm 0i$
$s_{6,7}$	$-0.083 \pm 5.31i$
$s_{8,9}$	$-0.74 \pm 50.79i$

Table 6.4: Poles of the test model

Pole	Value
s_1	$-2.29 \cdot 10^9$
$s_{2,3}$	$-12.28 \pm 15.44i$
$s_{4,5}$	$-0.0064 \pm 0i$

As it can be observed, both the test model and the recovery system have a pair of poles that are close to 0, which will affect the system stability. Since greater **H** values are used in this model, the result is less sensitivity of the closed-loop system to disturbances. Usually, the integral control action is added to improve the system steady-state performance. However, it will worsen the stability problem, and it may cause oscillatory response. As a result, the difference plot is offset from the zero axis.

According to the results, from the difference plot, at low frequency (long wavelength components of the Chalma road profile), the recovery accuracy is very good. For getting high accuracy at high frequency, i.e., the short wavelength parts of the road profile, a 9-th order high pass Butterworth filter with a cutoff frequency of 0.3 Hz is designed to filter the input and the output profiles, because the low frequency parts are a manifestation of the drifting effect. The transfer function of the filter is given by:

$$H(s) = \frac{s^9}{s^9 + 10.86s^8 + 58.92s^7 + 208.7s^6 + 530s^5 + 999.1s^4 + 1398s^3 + 1402s^2 + 917.8s + 300.4} \quad (6.1)$$

By using the two profiles as the input of the transfer function, the filtered results are shown in Figure 6.35. After filtering, only road excitation at the range of ±7 cm are shown in the plot, and the difference is in the range of ±1.5 mm, which means the accuracy of high frequency parts of the road profile is improved.
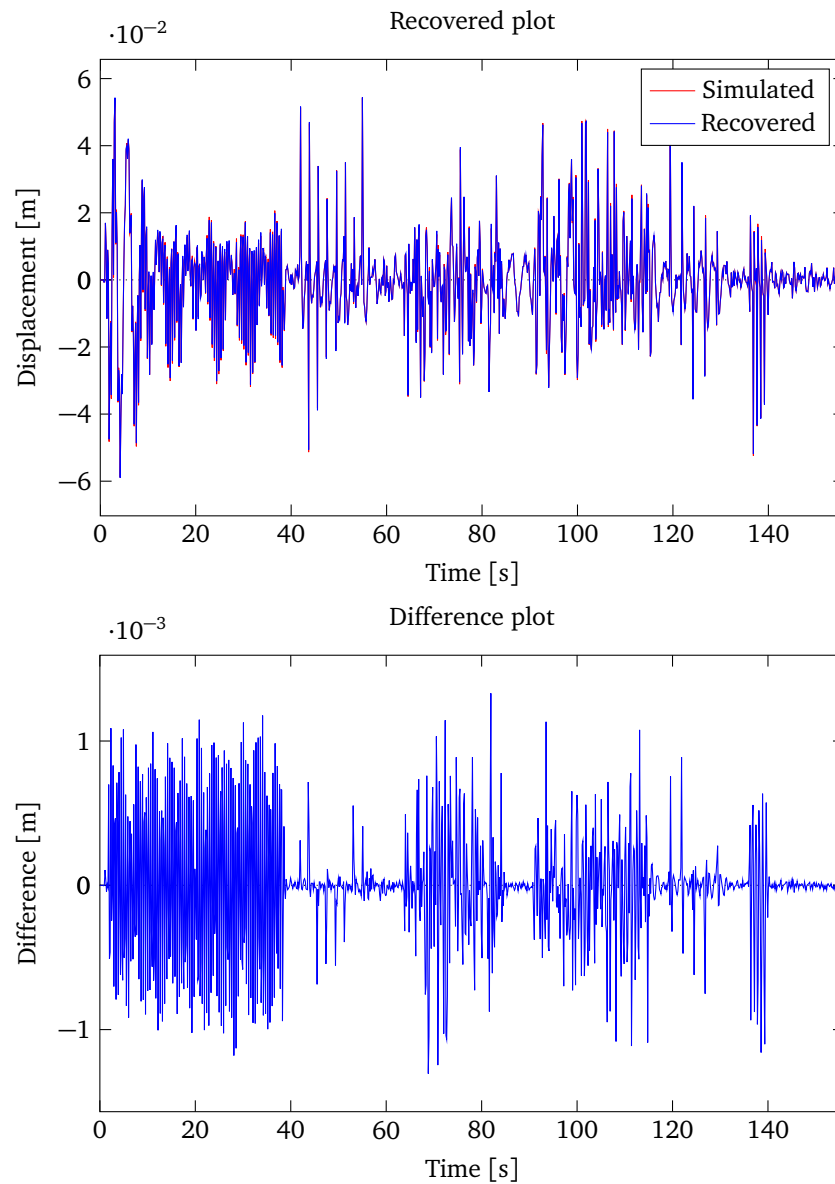

Figure 6.35: Filtered recovered and simulated Chalma road profiles and the difference using two A arm models in Altair MotionView®

To extend the ability of the recovery algorithm, as shown in Figure 6.36, two of the same non-linear MacPherson models are applied. The connections of the two models are through the same controller as described in the recovery model using two A arm models. Notice that the non-linear damper used in this MacPherson model is different than with the A arm model. While using the same controller gain **H**, the simulation results are given in Figure 6.37 to Figure 6.40.
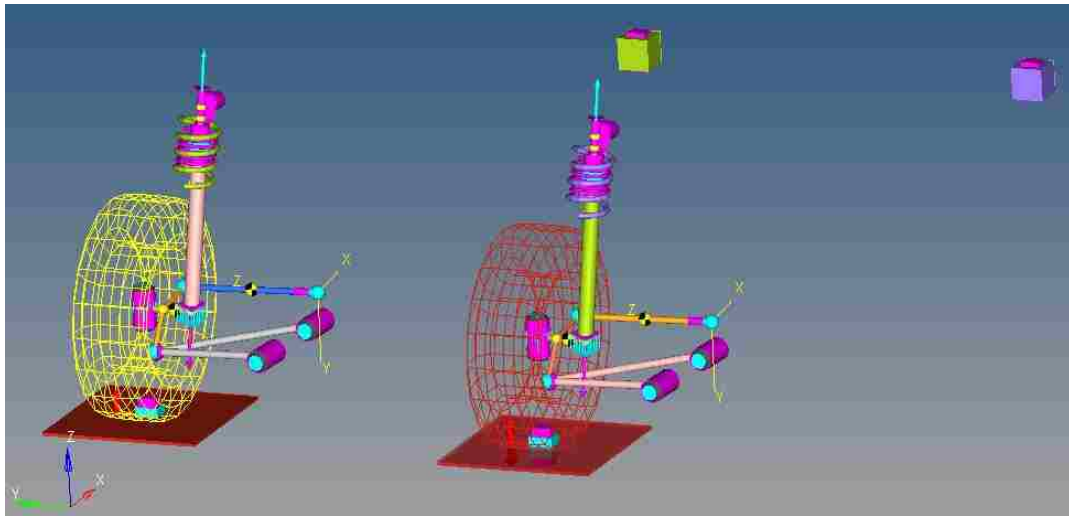

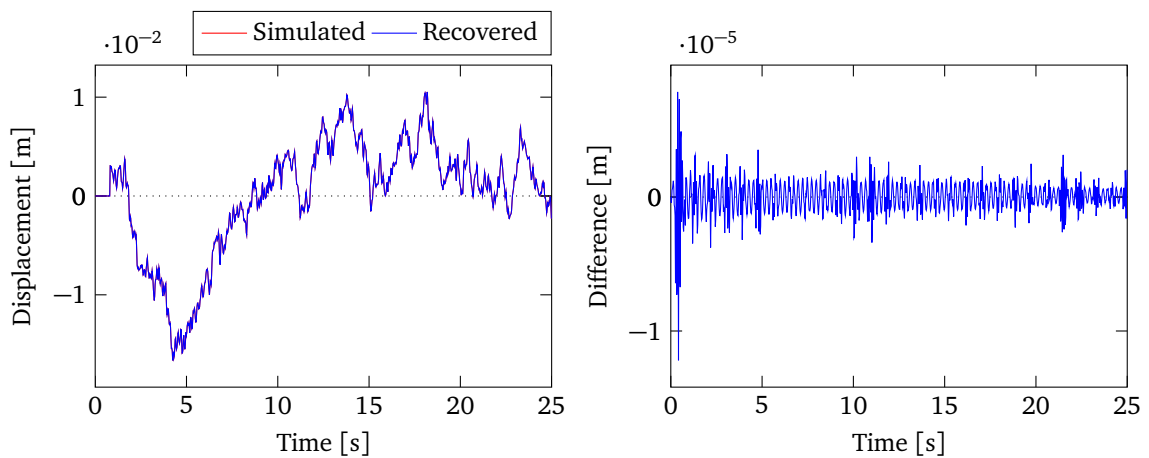**Figure 6.36:** Two non-linear MacPherson models running together

Figure 6.37: Recovered and simulated ISO A class road profiles and the difference
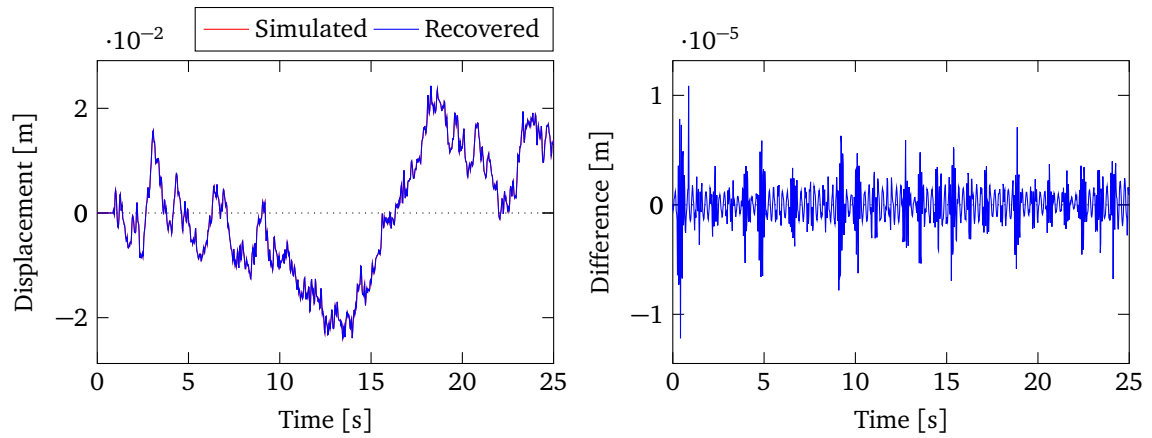using two MacPherson models in Altair MotionView®

Figure 6.38: Recovered and simulated ISO B-C class road profiles and the difference
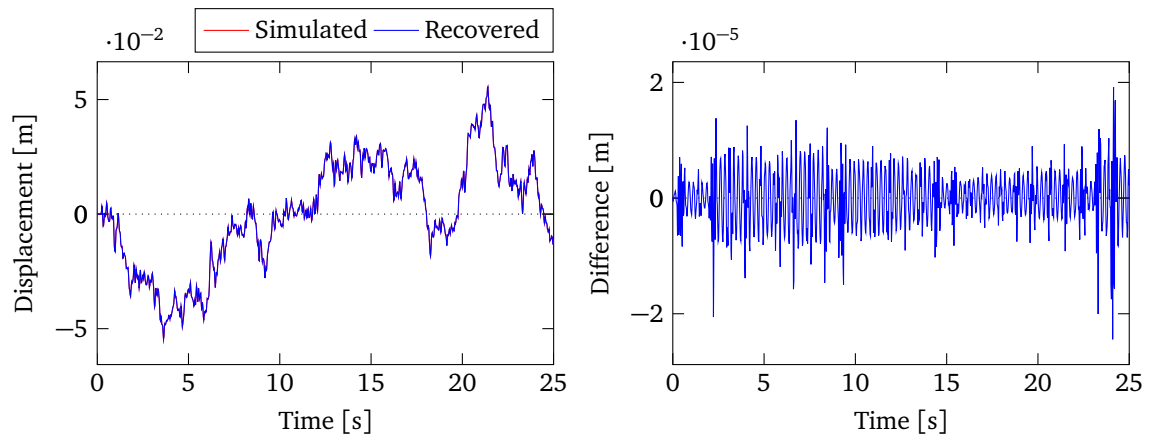using two MacPherson models in Altair MotionView®

Figure 6.39: Recovered and simulated ISO C-D class road profiles and the difference
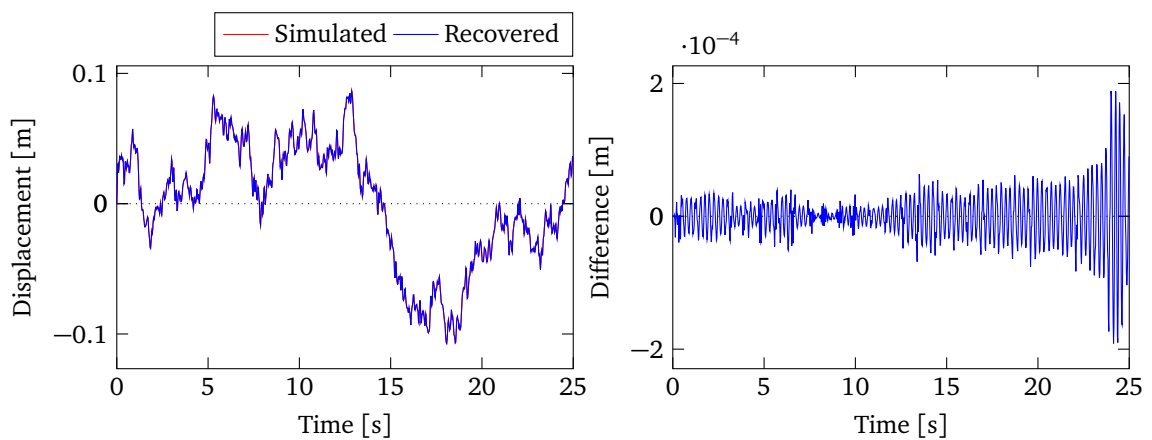using two MacPherson models in Altair MotionView®

Figure 6.40: Recovered and simulated ISO D-E class road profiles and the difference
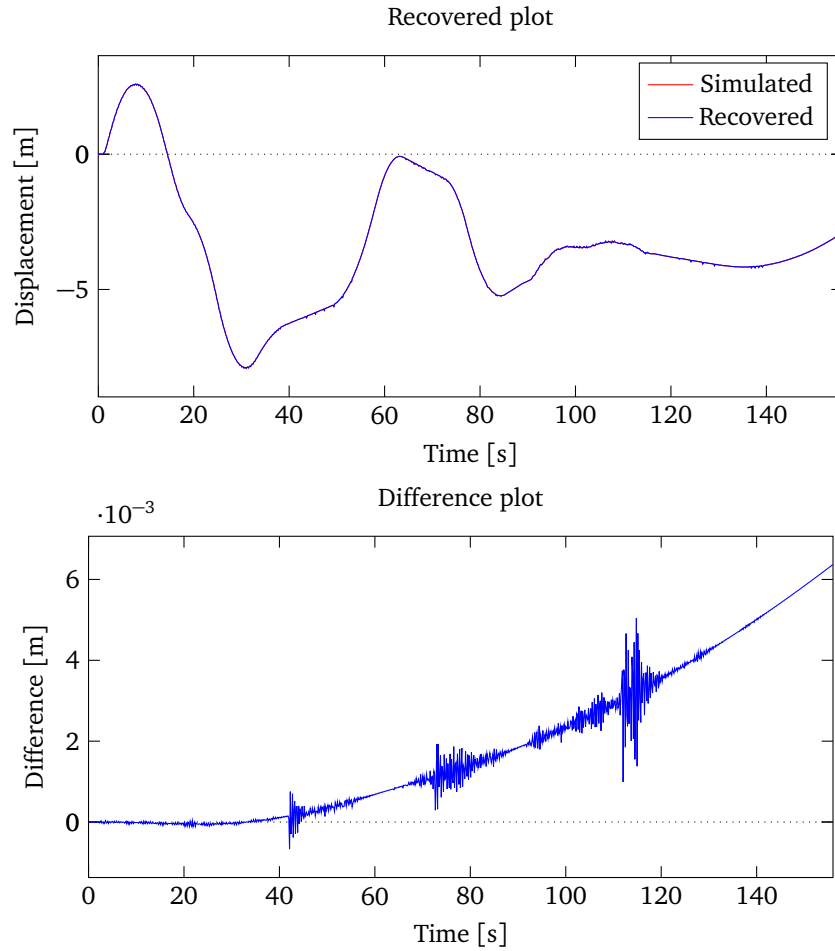using two MacPherson models in Altair MotionView®

Figure 6.41: Recovered and simulated Chalma road profiles and the difference using two MacPherson models in Altair MotionView®
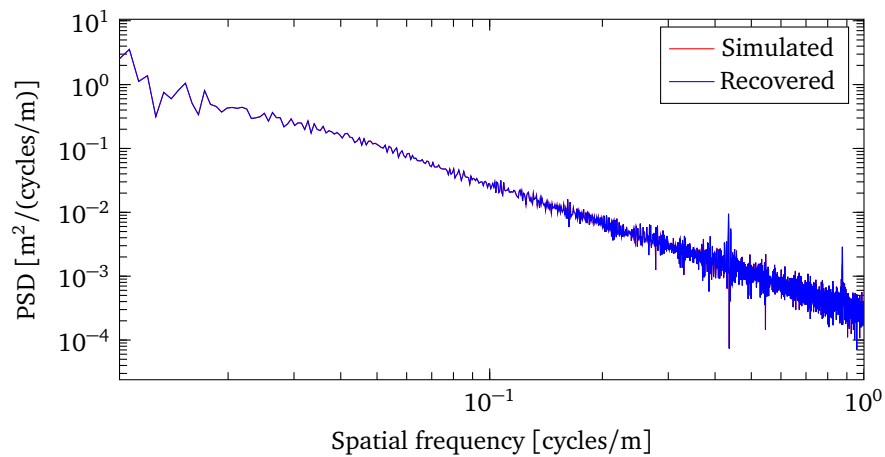

Figure 6.42: PSDs of recovered and simulated Chalma road profiles using two MacPherson models in Altair MotionView®

Results of the Chalma road recovery using this two MacPherson models are displayed in Figure 6.41, and Figure 6.42 shows the PSDs of the recovered and simulated Chalma road profiles. The difference plot shows clearly that the recovered road profile converges to the simulated road profile very well except that the drift effect happens. In this case, using the same high-pass Butterworth filter, the result of the filtered road profiles are presented in Figure 6.43. The recovery error stay less than 2.5 mm anyway; a good filtering enables high recovery accuracy.
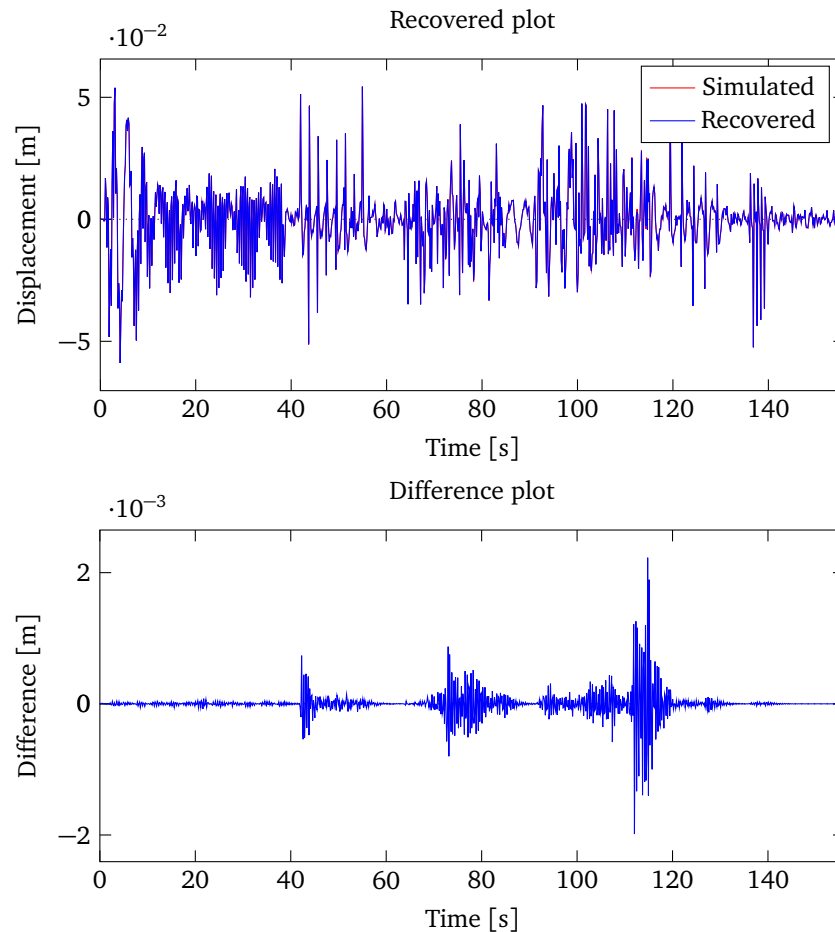

Figure 6.43: Filtered recovered and simulated Chalma road profiles and the difference using two MacPherson models in Altair MotionView®

In order to simulate the proposed method using only acceleration data, body and wheel accelerations are pre-calculated and exported by simulating the virtual MacPherson model driving along the Chalma road profile. Assuming that these are real data measured from a test vehicle, these data are then used as the input of the same MacPherson model with the recovery algorithm, to recover the road profile. The pre-calculated acceleration data are plotted in Figure 6.44. When compared to the wheel acceleration, the body acceleration is too small to see the details in the plot.
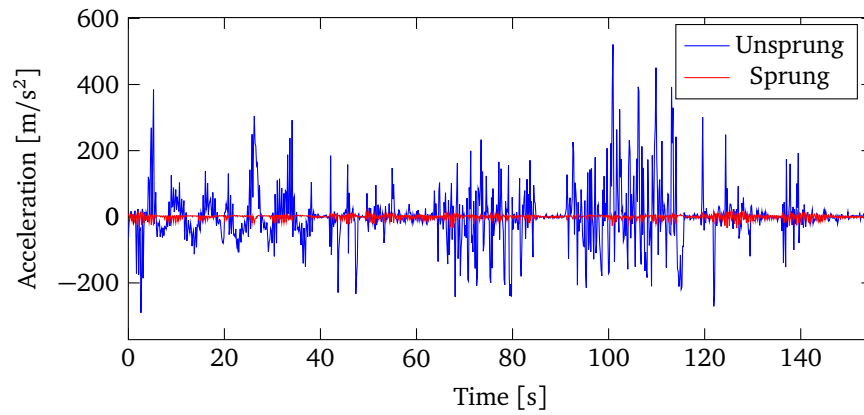
Figure 6.44: Accelerations of the sprung mass and unsprung mass from MacPherson model in Altair MotionView®
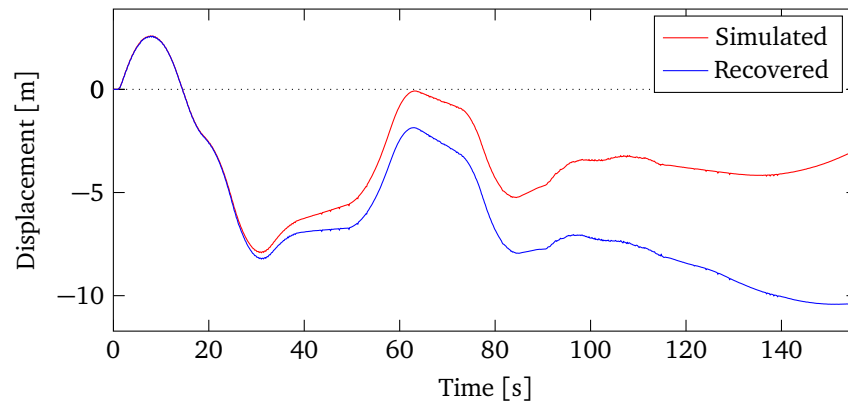

Figure 6.45: Recovered and simulated Chalma road profiles from pre-calculated acceleration data in Altair MotionView®
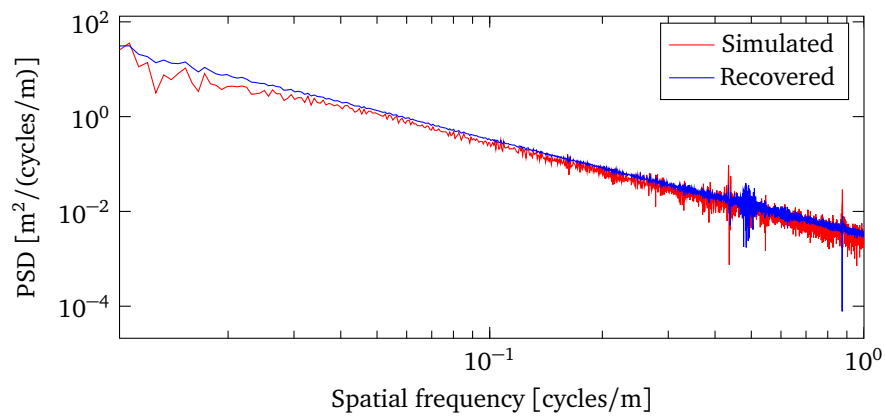

Figure 6.46: PSDs of recovered and simulated Chalma road profiles from pre-calculated acceleration data in Altair MotionView®

Figure 6.45 shows the recovery result using the pre-calculated acceleration data as the input of the recovery algorithm. Because of some data storage issues, the result from Figure 6.45 shows that the recovery model seems to be not predicting the Chalma road profile very well. Only a part of the recovered road profile is recovered accurately at the first 30 sec, while after that the recovered road profile is drifting apart from the original road profile. However, as shown in Figure 6.46, the recovered road profile has similar PSD with the original road profile, even though the recovered road profile looks different. It can be inferred that the short wavelength, high frequency components are captured. A 9-th order high pass Butterworth filter with a cutoff frequency of 0.1 Hz is introduced to filter the recovered and the original Chalma road profiles. The transfer function of the filter is given by:

$$H(s) = \frac{s^9}{s^9 + 3.618s^8 + 6.546s^7 + 7.73s^6 + 6.544s^5 + 4.112s^4 + 1.917s^3 + 0.641s^2 + 0.1399s + 0.01526}$$
(6.2)

The filtered results are plotted in Figure 6.47. As it is shown in the plots, after filtering, the road trajectory of the recovery model replicates the original road profile more accurately. The maximum recovery error is less than 3 mm. This shows that the recovery algorithm is sensitive to numerical error.

An effectiveness test of the proposed recovery approach is also carried out with unknown road profile data. Starting with measured body and wheel acceleration data picked up from vehicle mounted accelerometers, a MacPherson model is then used to estimate the road profile. The measured acceleration data are plotted in Figure 6.48. The recovered body and wheel accelerations and the difference between the measured and the recovered acceleration data are drawn in Figure 6.49 and Figure 6.50. As shown in Figure 6.49, the recovered body acceleration is quite different from the measured data. However, as it can be seen in Figure 6.50, the wheel acceleration is recovered very well. It can be inferred that the measured data contains some noise, or the vehicle model does not fully represent the test vehicle. The estimated road profile with a zoomed section is plotted in Figure 6.51. Because of lacking of information about the original road profile and the test vehicle, this result cannot be verified.
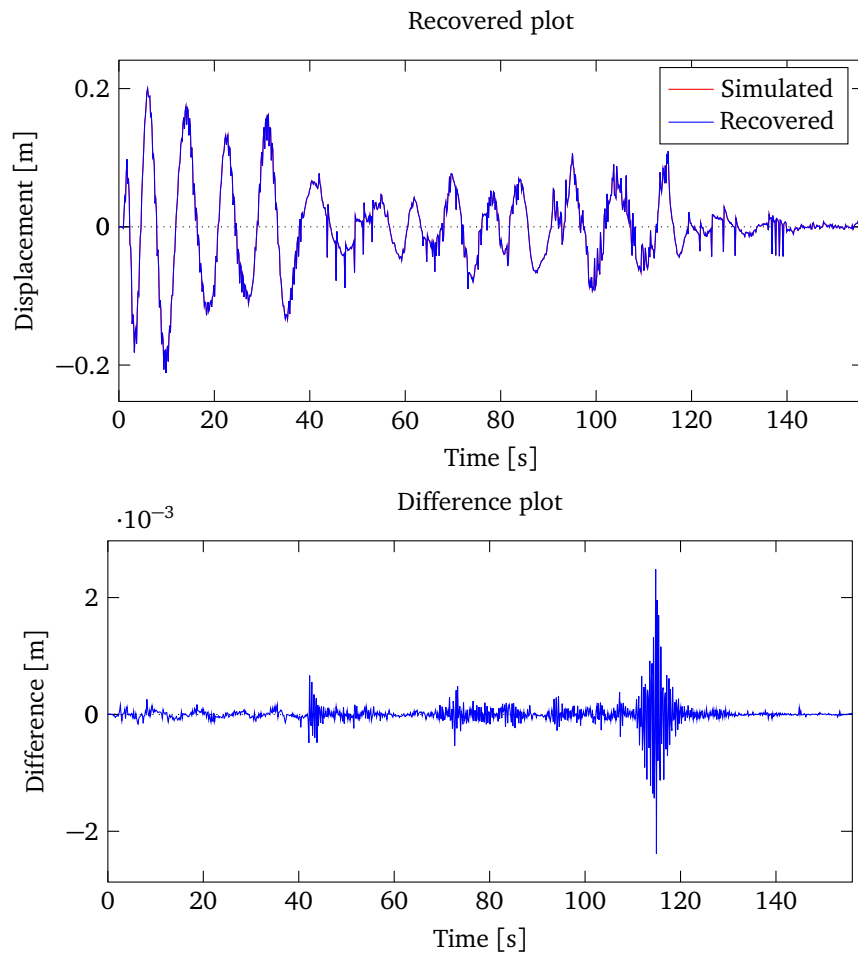
Figure 6.47: Filtered recovered and simulated Chalma road profiles and the difference from pre-calculated acceleration data in Altair MotionView®
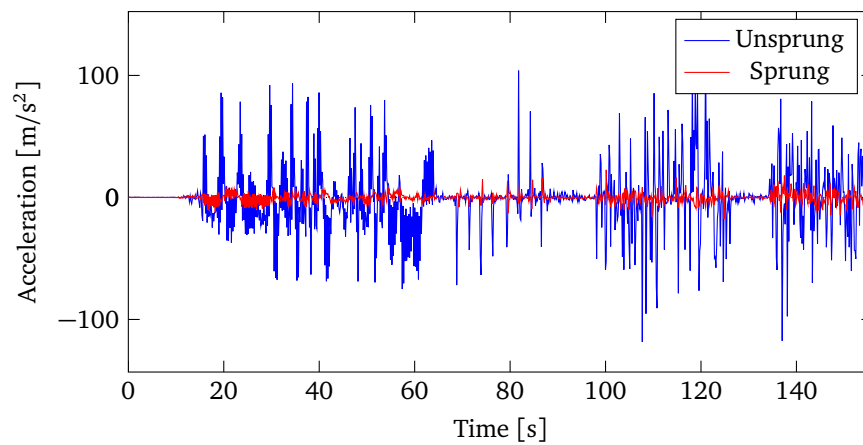

Figure 6.48: Real measured body and wheel accelerations
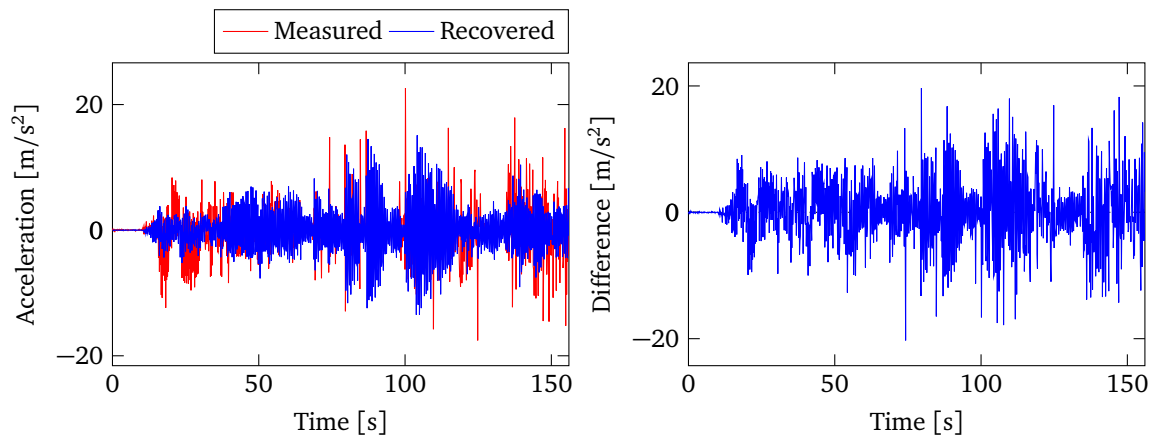
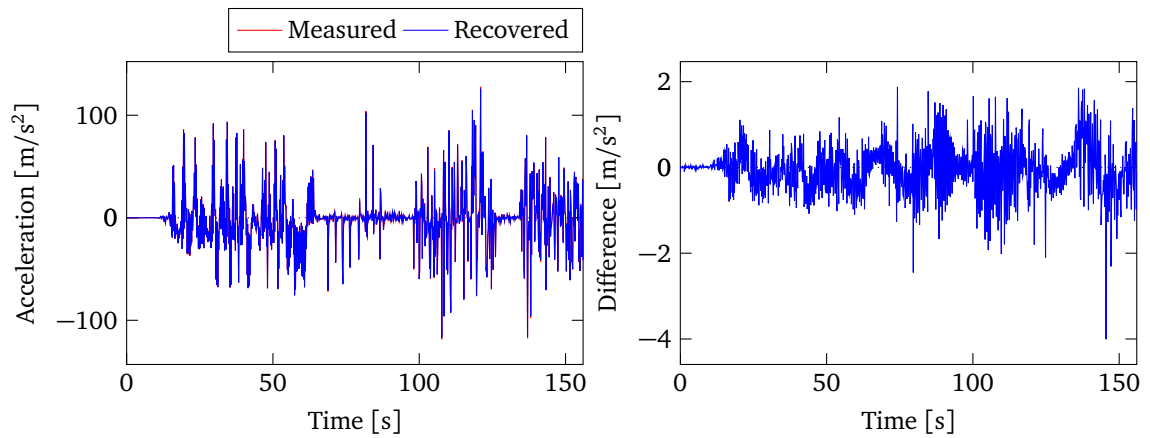**Figure 6.49:** Measured and recovered body accelerations and the difference

Figure 6.50: Measured and recovered wheel accelerations and the difference
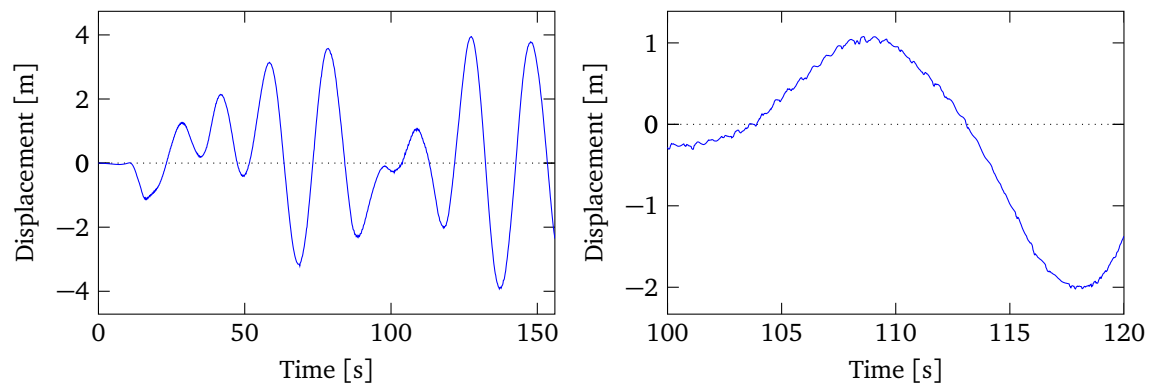

Figure 6.51: Estimated road profile and a zoomed section

6.3 Summary

Two kinds of simulation results have been presented in this chapter. All the four ISO class road profile recovery results show that the accuracy of the proposed method does not appear to be influenced by the road roughness, when the recovery model is a good representation of the test vehicle. Even when using a linear recovery model to predict the motion of the non-linear model, the method is able to recover the randomly generated road profiles. However, in this case, the accuracy of the result decreases with increasing road roughness.

In the numerical simulations, it has been found that even for rougher road, like the Chalma road profile, the recovery algorithm still has the ability to recover the input road profile. When dealing with non-linearities, the linear recovery model is less accurate in the recovery of high frequency components. However, the discrepancy between the real road input and the recovered road input is unavoidable as the real suspension response cannot be estimated by a linear quarter car model.

A non-linear recovery model is introduced for the purpose of improving the modeling of the real suspension. As expected, simulation results indicate that this approach recovers road profiles with higher accuracy.

Because there is no related information about the test data, simulation with real measured acceleration data is discussed without verification.

Chapter 7

Conclusions and Future Work

7.1   Conclusions

The road profile data plays a key role in assessing vehicle suspension performance. It becomes increasingly important to develop methods to collect road profiles precisely. The purpose of this project is to develop a simple and economical method of recovering road profile effectively. The development of the method to recover road profile is stated throughout this thesis.

Nowadays, virtual simulation is very helpful for research in the area of vehicle dynamics and control. A data recovery algorithm using vertical acceleration data from the body and wheel of a dynamic model is presented in this thesis in order to achieve the project objectives. To verify the proposed approach, two computer software tools are used to develop virtual dynamic models to do the simulation. Numerical simulations are then carried out using these models to obtain acceleration data as the input signal of the proposed recovery algorithm. One of the key features of this research is that only acceleration data from the body and wheel of a vehicle are needed.

To verify the effectiveness of the proposed method, tests were implemented in both MATLAB® and Altair MotionView®. Once the virtual models were built, road profiles regardless of the roughness could be recovered with proper controller gains. Two groups of models were built; a linear recovery model and a non-linear recovery model. Four different ISO class road profiles and one testing track are tested using these virtual models.

Results show that this approach can recover the road profile with either linear or non-linear vehicle parameters. It can be pointed out that if the recovery model best represents real non-linear vehicle dynamics, the road profile can be recovered with higher accuracy. Although the drift effect happens sometimes, a similar level of recovery accuracy can still be obtained via high pass Butterworth filters.

Overall this thesis has presented the potential of this approach to be developed and implemented as a low-cost tool for estimating the road profile over which a vehicle has been driven.

7.2   Future Work

Numerous uncertainties still exist and are avenues of future study. This thesis shows that it is possible to use only vertical accelerations from a vehicle to obtain road profile data accurately. This research only uses quarter car models to achieve the goal; the extension of the recovery model to half car or full car is a natural direction of future work. Also, methods for determination of the controller gains $\mathbf{G}$ and $\mathbf{H}$ values independently of the road profile are still challenging. The potential errors associated with modeling approaches (e.g. the assumption of linearity or mis-estimation of model parameters) or the cumulative drift effect when comparing with measured acceleration data could lead to a reduction in accuracy in practice.

In addition, when using experimental data to validate the developed method, more detailed vehicle models must be considered, particularly in the unpredictable non-linear region, such as non-linear tires, which is a difficult task. Additionally, the effects of measurement noise on the results should also be investigated.

References

[1] S. Karamihas and M. Sayers, *The little book of profiling*. Ann Arbor, Michigan, USA: Transportation Research Institute, University of Michigan, 1998.

[2] J. Dixon, *Suspension geometry and computation*. Chichester, West Sussex, United Kingdom: John Wiley and Sons Ltd, 2009.

[3] G. Rill, "Vehicle dynamics," University Lecture Notes, 2009.

[4] M. Agostinacchio, D. Ciampa, and S. Olita, "The vibrations induced by surface irregularities in road pavements – a Matlab approach," *Eur. Transp. Res. Rev.*, vol. 6, pp. 267–275, 2014.

[5] H. Imine, Y. Delanne, and N. M'Sirdi, "Road profile input estimation in vehicle dynamics simulation," *Vehicle System Dynamics*, vol. 44, no. 4, pp. 285–303, 2006.

[6] A. Rabhi *et al.*, "Second order sliding mode observer for estimation of road profile," in *Proc of Workshop on Variable Structure Systems (VSS)*. IEEE, 2006.

[7] R. McCann and S. Nguyen, "System identification for a model-based observer of a road roughness profiler," in *Proc of Region 5 Technical Conference*. IEEE, 2007.

[8] H. Imine and Y. Delanne, "Triangular observers for road profiles inputs estimation and vehicle dynamics analysis," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2005.

[9] M. Pong, "The development of an extensive-range dynamic road profile and roughness measuring system," Ph.D. dissertation, Pennsylvania State University, 1992.

[10] M. Doumiati *et al.*, "Estimation of road profile for vehicle dynamics motion: experimental validation," in *American Control Conference (ACC)*. IEEE, 2011.

[11] A. Solhmirzaei, S. Azadi, and R. Kazemi, "Road profile estimation using wavelet neural network and 7-DOF vehicle dynamic systems," *Journal of Mechanical Science and Technology*, vol. 26, no. 10, pp. 3029–3036, 2012.

[12] D. Yang, H. Yi, and X. Lian, "Research on rapid measurement of medium short wave longitudinal road profiles," in *International Conference on Electrical and Control Engineering (ICECE)*. IEEE, 2010.

[13] A. Gonzalez *et al.*, "The use of vehicle acceleration measurements to estimate road roughness," *Vehicle System Dynamics*, vol. 46, no. 6, pp. 483–499, 2008.

[14] K. Hong, H. Sohn, and J. Hedrick, "Modified skyhook control of semi-active suspensions: A new model, gain scheduling, and hardware-in-the-loop tuning," *Journal of Dynamic Systems, Measurement ,and Control*, vol. 124, no. 1, pp. 158–167, 2002.

[15] T. Heyns, P. Heyns, and J. Villiers, "A method for real-time condition monitoring of haul roads based on Bayesian parameter estimation," *Journal of Terramechanics*, vol. 49, pp. 103–113, 2012.

[16] R. Jazar, *Vehicle dynamics: theory and application*. New York, NY, USA: Springer Science+ Business Media, LLC, 2008.

[17] H. Burah, *Analytical dynamics*. Boston, MA, USA: WCB/McGraw-Hill, 1999.

[18] B. Minaker and R. Rieveley, "Automatic generation of the nonholonomic equations of motion for vehicle stability analysis," *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, vol. 48, no. 9, pp. 1043–1063, 2010.

[19] N. Harris, A. Gonzalez, and E. O'Brien, "Characterisation of pavement profile heights using accelerometer readings and a combinatorial optimisation technique," *Journal of Sound and Vibration*, vol. 329, no. 5, pp. 497–508, 2010.

[20] P. McGetrick *et al.*, "Dynamic axle force and road profile identification using a moving vehicle," *International Journal of Architecture, Engineering and Construction*, vol. 2, no. 1, pp. 1–16, 2013.

[21] B. Minaker, X. Yang, and S. Li, "Design optimization of an SAE Baja vehicle using the EoM open source multibody dynamics code," in *Proceedings of The Canadian Society for Mechanical Engineering International Congress*, 2014.

[22] R. Dorf and R. Bishop, *Modern control systems*. Upper Saddle River, NJ, USA: Pearson Education, 2011.

[23] B. Kulakowski, J. Gardner, and J. Shearer, *Dynamic modeling and control of engineering systems*. New York, NY, USA: Cambridge University Press, 2007.

[24] F. Hu, "Road profile recovery user manual," ARDC Chrysler Canada, 2015.

Appendix A

M<small>ATLAB</small>® Codes

Penalty Function

```matlab
function penalty = equivalent_penalty(x)
global e1 e2;
% Body and wheel natural frequencies

params.ks=x(1);
params.mu=x(2);
params.cs=x(3);
params.ms=x(4);
% Set unknown parameters

params.kt=308570;
% Set the tire stiffness as known parameter

sys1=run_eom('input_ex_quarter_car',params,'quiet');
% Use EoM software to do the simulation

evals=eig(sys1.A,sys1.E)/(2*pi);
% Calculate the eigenvalues from the model

evals=sort(evals);
% Sort the eigenvalues

a1=e1-evals(2);
a2=e2-evals(4);
% Compare the natural frequencies with the eigenvalues
penalty=sum(a1^2+a2^2)
```

Optimizer file

```matlab
global e1 e2;

e1=-1.2225104E-01+ 8.4442185E-01i;
e2=-1.0712943E+00+ 8.1388394E+00i;
% Body and wheel natural frequencies calculated from MotionView

fminsearch(@equivalent_penalty,[15000,50,1000,500])
% Call the penalty function, search the optimal values using the initial values
```

Appendix B

Other Simulation Results

Figure B.1: Recovered and simulated Chalma road profiles using half car model in MATLAB®, set rear input equal to front, but delayed by wheelbase, at 10 m/s
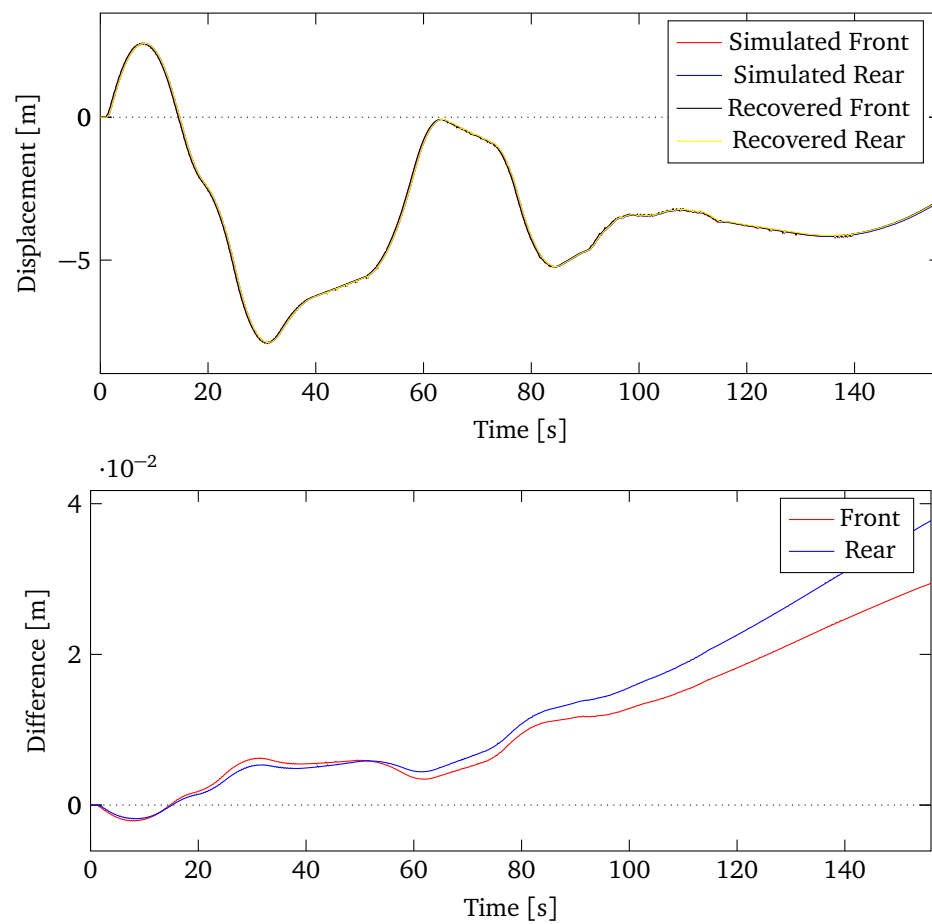
Figure B.2: Filtered recovered and simulated Chalma road profiles and the difference using full car model in MATLAB®, set rear input equal to front, but delayed by wheelbase, at 10 m/s
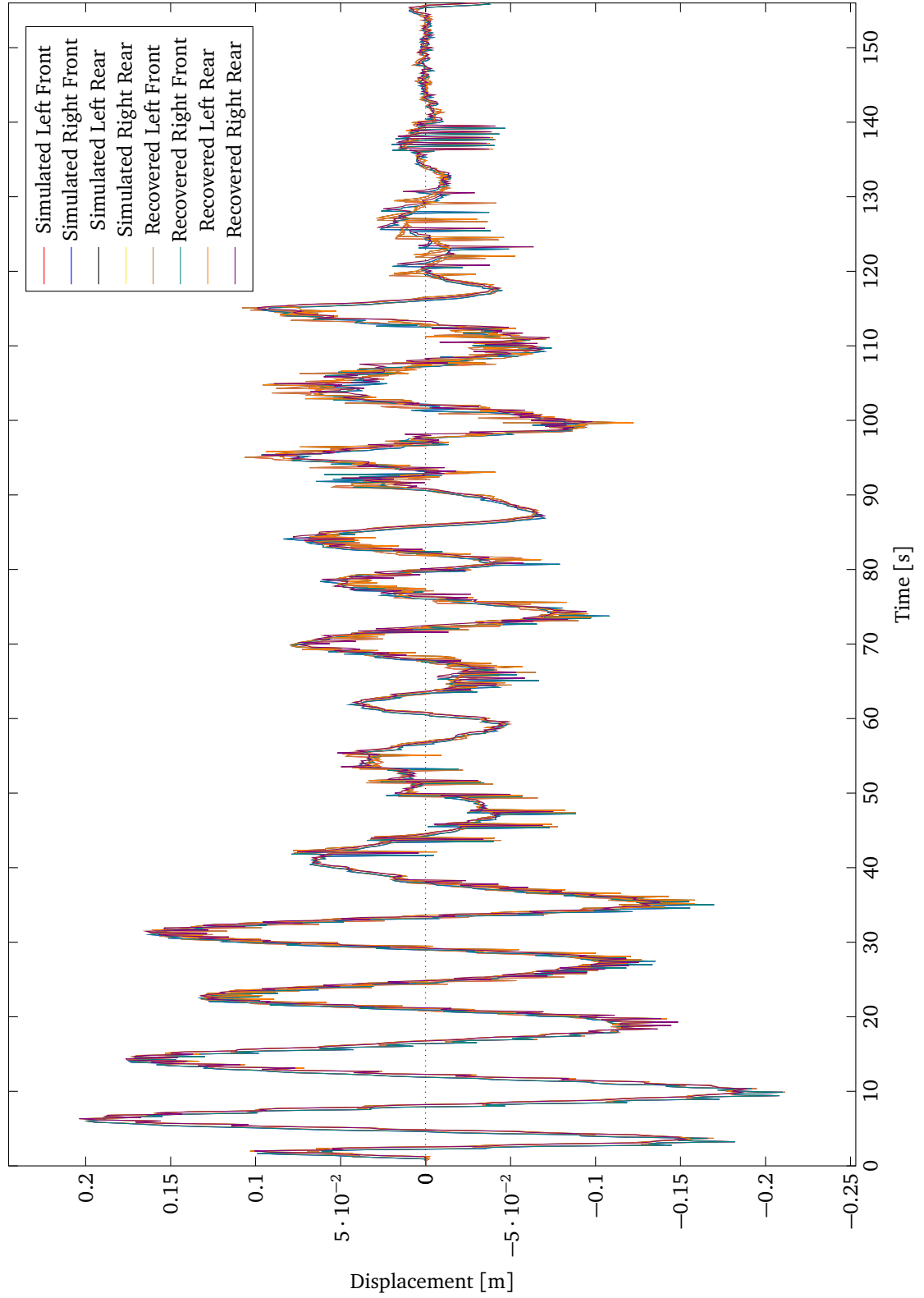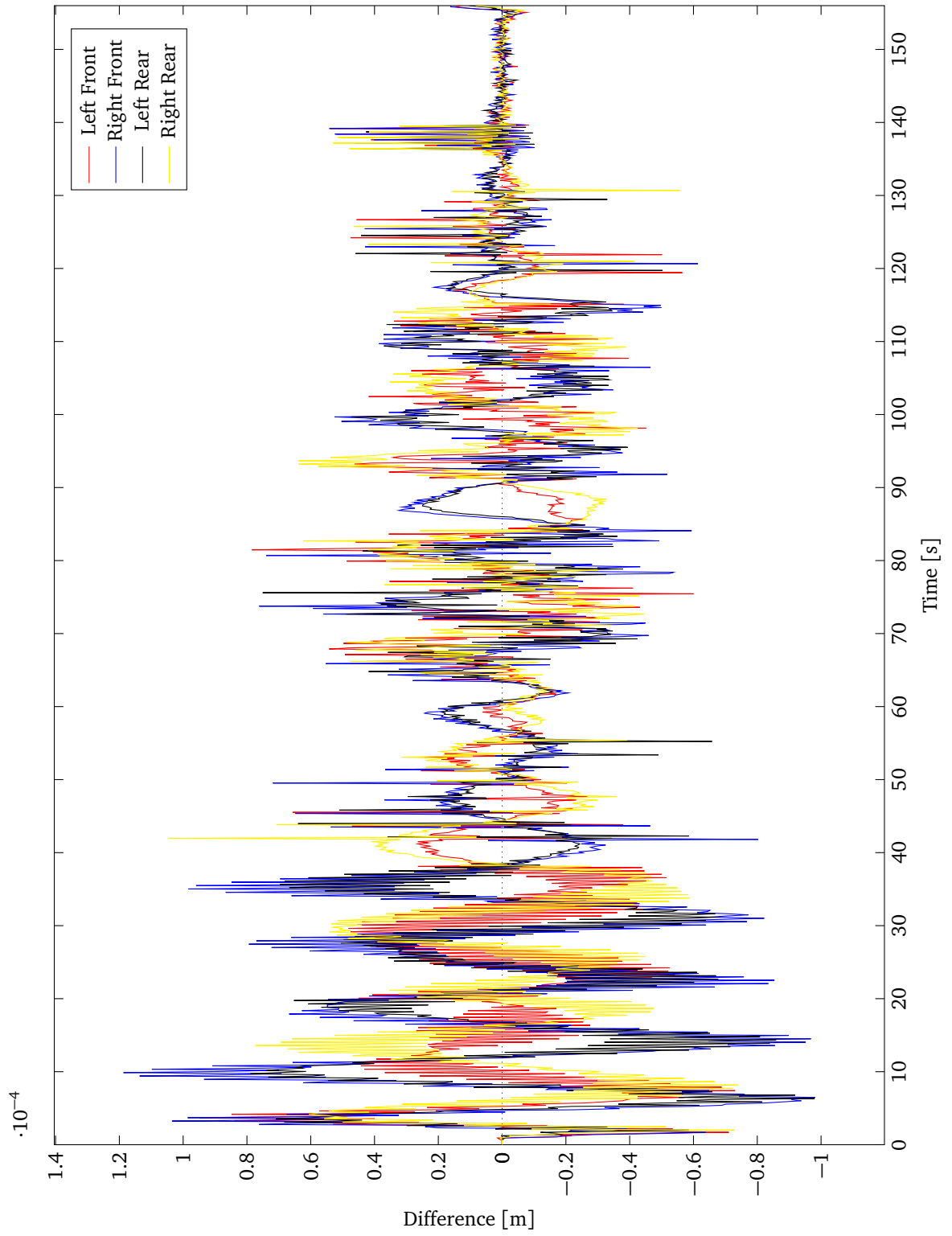
Figure B.3: Difference between the filtered recovered and simulated Chalma road profiles using full car model in MATLAB®

VITA AUCTORIS

Fei Hu was born in Huanggang, Hubei, China in 1989. He graduated from Hubei provincial No. 1 High School of Qichun County in 2007. He then attended the Northeast Dianli University, where he earned Bachelor's Degree of Engineering in Thermal Energy and Power Engineering in 2011. He is presently a candidate for the Master of Applied Science in Mechanical Engineering degree at the University of Windsor, and is scheduled to graduate in June 2015.