Machine Learning from Computer Simulations with Applications in Rail Vehicle Dynamics and System Identification

Mehdi Taheri

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial Fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Mechanical Engineering

Mehdi Ahmadian

Muhammad R. Hajj

Robert L. West

Reza Mirzaeifar

April 27, 2016

Blacksburg, VA, USA

Keywords: Stochastic modeling, meta-models, Surrogate models, Three-piece truck, Global optimization, Latin Hypercube sampling plan

Machine Learning from Computer Simulations with Applications in Rail Vehicle Dynamics and System Identification

Mehdi Taheri

ABSTRACT

The application of stochastic modeling for learning the behavior of multibody dynamics models is investigated. The stochastic modeling technique is also known as Kriging or random function approach. Post-processing data from a simulation run is used to train the stochastic model that estimates the relationship between model inputs, such as the suspension relative displacement and velocity, and the output, for example, sum of suspension forces. Computational efficiency of Multibody Dynamics (MBD) models can be improved by replacing their computationally-intensive subsystems with stochastic predictions. The stochastic modeling technique is able to learn the behavior of a physical system and integrate its behavior in MBS models, resulting in improved real-time simulations and reduced computational effort in models with repeated substructures (for example, modeling a train with a large number of rail vehicles). Since the sampling plan greatly influences the overall accuracy and efficiency of the stochastic predictions, various sampling plans are investigated, and a space-filling Latin Hypercube sampling plan based on the traveling salesman problem (TPS) is suggested for efficiently representing the entire parameter space.

The simulation results confirm the expected increased modeling efficiency, although further research is needed for improving the accuracy of the predictions. The prediction accuracy is expected to improve through employing a sampling strategy that considers the discrete nature of the training data and uses infill criteria that considers the shape of the output function and detects sample spaces with high prediction errors. It is recommended that future efforts consider quantifying the computation efficiency of the proposed learning behavior by overcoming the inefficiencies associated with transferring data between multiple software packages, which proved to be a limiting factor in this study. These limitations can be overcome by using the user subroutine functionality of SIMPACK and adding the stochastic modeling technique to its force library.

Acknowledgement

I would like to express my special appreciation and thanks to my advisor, Dr. Mehdi Ahmadian, who has been a tremendous mentor for me. I would like to thank him for encouraging my research and for allowing me to grow as a researcher. His advice on both research as well as on my career have been priceless. I would also like to thank my committee members, Dr. Muhammad Hajj, Dr. Robert West, and Dr. Reza Mirzaeifar for serving as my committee members.

Special thanks are due to my family, both local and global! Words cannot express how grateful I am to my mother and father for all of the sacrifices that they have made on my behalf. Mom, dad, your prayers were what sustained me throughout my journey. I would also like to thank my friends at CVeSS, Virginia Tech, and other places that my path has crossed for supporting me throughout my graduate education, writing my dissertation, and aiming for my goals.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Computer simulations have been gaining interest as a design tool in the past few decades. Powerful computation resources coupled with accurate numerical techniques have made computer simulations an irreplaceable part of product design and development. In this regard, computer simulations can be divided into two main categories: 1) finite element methods (FEM), and 2) multibody dynamic analysis (MBD). FE analysis is typically used for detailed analysis of single bodies or parts, whereas MBD is used to analyze the behavior of an entire system.

Multibody systems are comprised of a number of interconnected rigid or flexible bodies where each body undergoes translational or rotational displacements in various directions. Connections between various bodies are described using joints, force elements, and constraints. Multibody dynamics studies the movement of mechanical systems under the influence of forces to better understand the interactions between multiple moving parts. Dynamic loads generated by moving parts are often difficult to predict, and MBD provides a reliable solution for the analysis of such systems; the results can be used to evaluate the performance, safety, and comfort of dynamic systems.

Depending on the complexity of the system being modeled and the level of accuracy built into the computer code, simulation run times may vary from a few seconds to hours or even days. While it is always more desirable to perform simulations as fast as possible, in some cases, the plausibility of the study depends on the simulation run time. Examples of scenarios where

computationally inexpensive simulations are required include parametric studies or global optimization routines, where the effectiveness of the study depends on a large number of simulations, hardware in the loop (HIL) analysis where, due to the interaction between the computer code and a physical system, real-time simulations are required, or systems with a large number of degrees of freedom, like train consists with hundreds of railcars, where even the fastest commercially available computers fail to run the simulation in reasonable times (Figure 1.1).

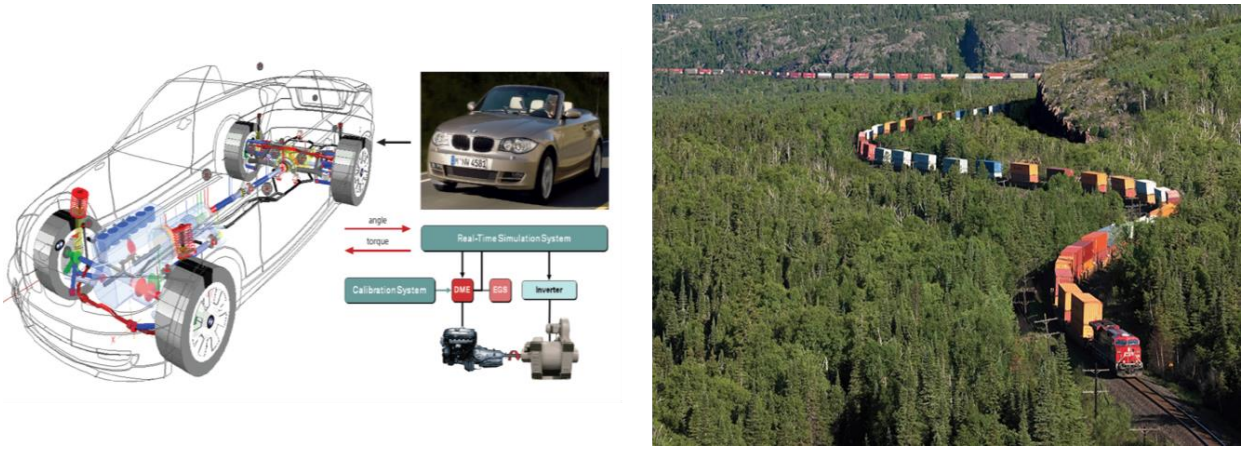

Figure 1.1: Examples of where computationally efficient MBD models are required

The computational costs of an MBD model can be reduced by using simplifying assumptions that would decrease the accuracy of the model. In this study, we have taken a new approach and tried to reduce the simulation time by using a stochastic model to replace parts of the MBD model that are computationally expensive. The stochastic model learns the behavior of a subsystem (in this case, the suspension subsystem) by sampling the outputs of a simulation run to find the relationship between the inputs (i.e. relative displacement and velocity across the suspension) and the outputs (i.e. sum of suspension forces). After the learning process is completed, the stochastic model will replace the deterministic suspension subsystem. The new model can then

be used to perform studies regarding the influence of various elements, like track conditions or curvature on the performance of the rail vehicle.

The accuracy of the stochastic predictions depends on the number of sampled points, the sampling plan chosen, and the infill criteria. The stochastic function approach can also be used to learn the behavior of a physical system, but the uncertainties associated with laboratory measured data have to be considered in the development of the stochastic model. Figure 1.2 and Figure 1.3 provide an overview of the application of the stochastic model to: 1) reduce the computational expense[1] of an MBD model, and 2) to integrate laboratory measured data into MBD models, respectively.

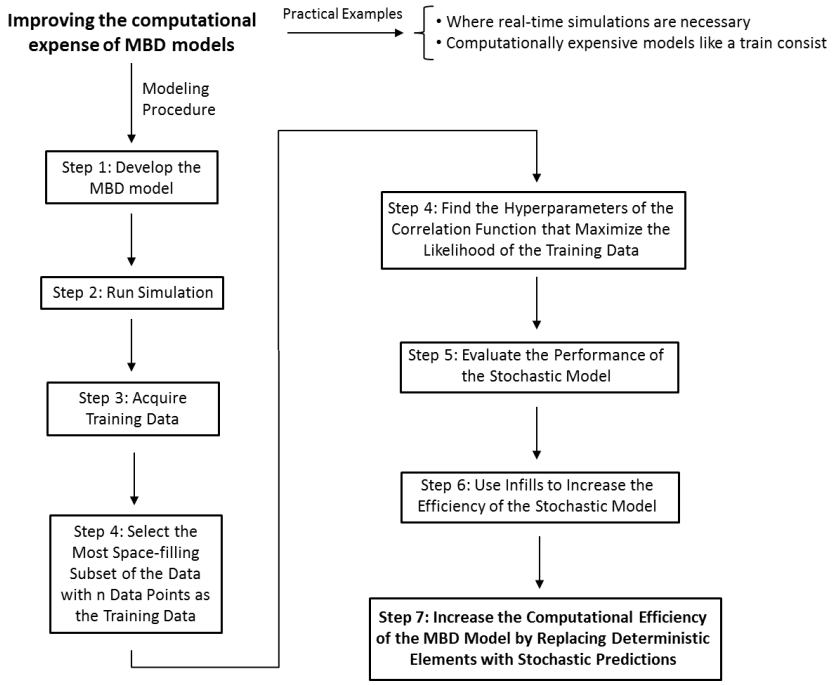

Figure 1.2: Schematic of the modeling procedure for using the stochastic model to reduce the computational expense of MBD models

[1] Computational expense can be quantified with the number of arithmetic operations performed by the CPU, or the time that it takes for a specific computer to complete the process. In this research the latter of the two has been used.

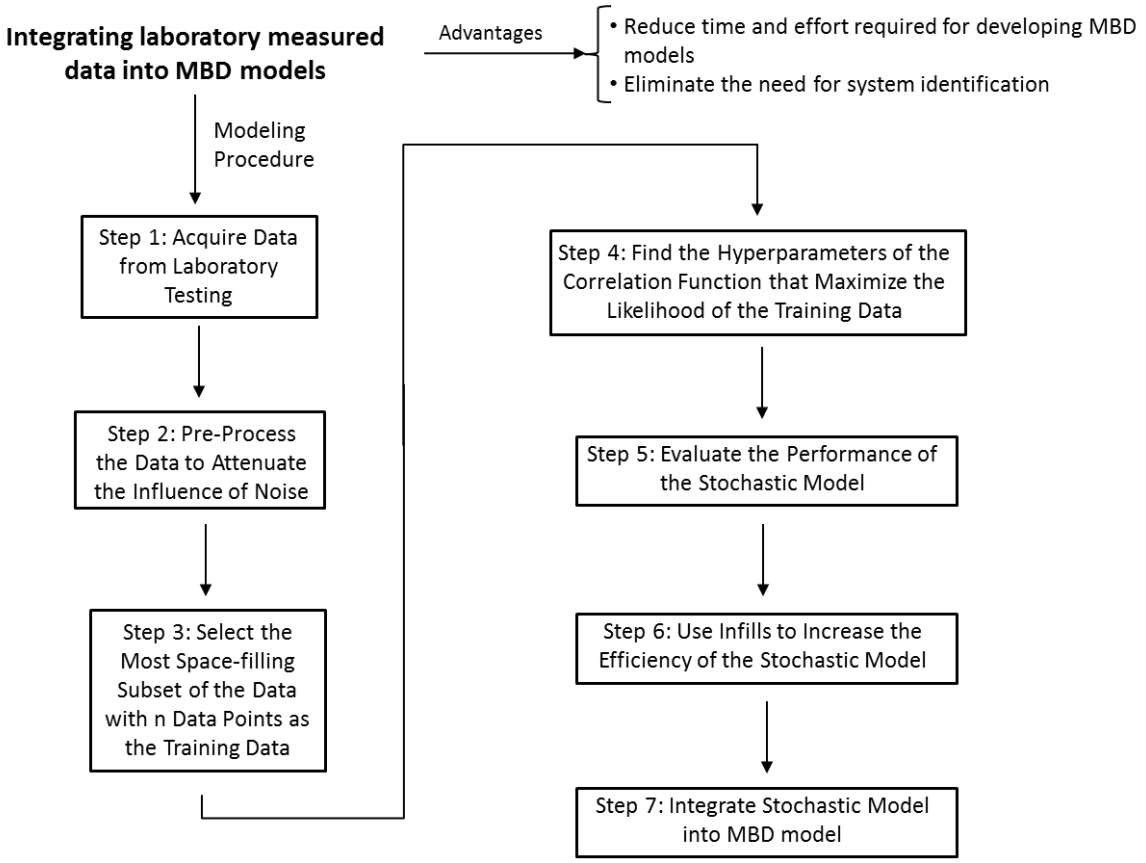

Figure 1.3: Schematics of modeling procedure for using the stochastic model to learn and integrate the behavior of physical system directly into MBD models

1.2 Importance

Computer simulations have gained popularity in the rail industry especially in the past decade. Availability of inexpensive and powerful computation resources have shifted the interest of the industry from laboratory experiments that are usually expensive, time consuming, and require very expensive equipment and human resources, to computer simulations where changing parameters and evaluating designs can be achieved quickly and inexpensively. Simulations can be used to study the influence and the importance of various parameters on the overall dynamics of the rail vehicle. While it is important to run physical experiments to validate the modeling

results, the costs for experiments can be substantially reduced by only performing the experiments that the model predicts will result in significant improvements.

Reducing the computation cost of mathematical models provides the opportunity to perform more detailed studies without compromising on the accuracy of the models. The proposed method for reducing the simulation run time employs a statistical approach to predict the behavior of a computationally expensive dynamic system or subsystem.

Accurate values for system parameters (i.e. stiffness value for springs) are an important factor that influences the accuracy of the results produced by the computer model when compared with the results obtained from actual physical systems. The current approach is to develop the MBD model and then use system identification techniques to estimate the set of parameters that best describe the behavior of the physical system. With minor modifications, the stochastic model can be used to integrate laboratory measured data directly into MBD models. The advantage of the proposed method in comparison with the existing approach is that it can reduce the effort required to develop sophisticated MBD models, and eliminate the need for system identification.

1.3 Research Approach

Our main goal in this study is to develop a stochastic model that can learn the behavior of MBD models or physical systems. The approach taken to fulfil the objectives of this research can be summarized as follows. The first step was to develop a mathematical model for the three-piece truck using SIMPACK. Next we developed a stochastic model that could learn the behavior of a system and predict the underlying relationship between the inputs and the output. The stochastic model was tested on a number of problems to verify its accuracy and efficiency. The stochastic model was then used to learn the behavior of the secondary suspension of the three-piece truck that was previously modeled. Deflection and velocity across the suspension were used as inputs to the stochastic model, and the sum of the secondary suspension's lateral force was the output. The stochastic model replaced the actual suspension elements of the three-piece truck and was able to accurately predict the behavior of the suspension.

The influence of various parameters on the performance of the stochastic model has also been studied. The study suggests that carefully adding additional training points to the model can increase its efficiency, therefore multiple criteria to optimize the selection of infills have been developed.

Our final goal was to use the same concept to learn the behavior of a physical system. This allows us to develop accurate computer models of complex physical systems and integrate laboratory data into MBD models, instead of the usual approach of developing complex MBD models and using system identification to find the set of system parameters that matches the behavior of the

model to the data acquired from laboratory testing. Figure 1.4 shows the flow of data between various sub-algorithms of the stochastic modeling technique.

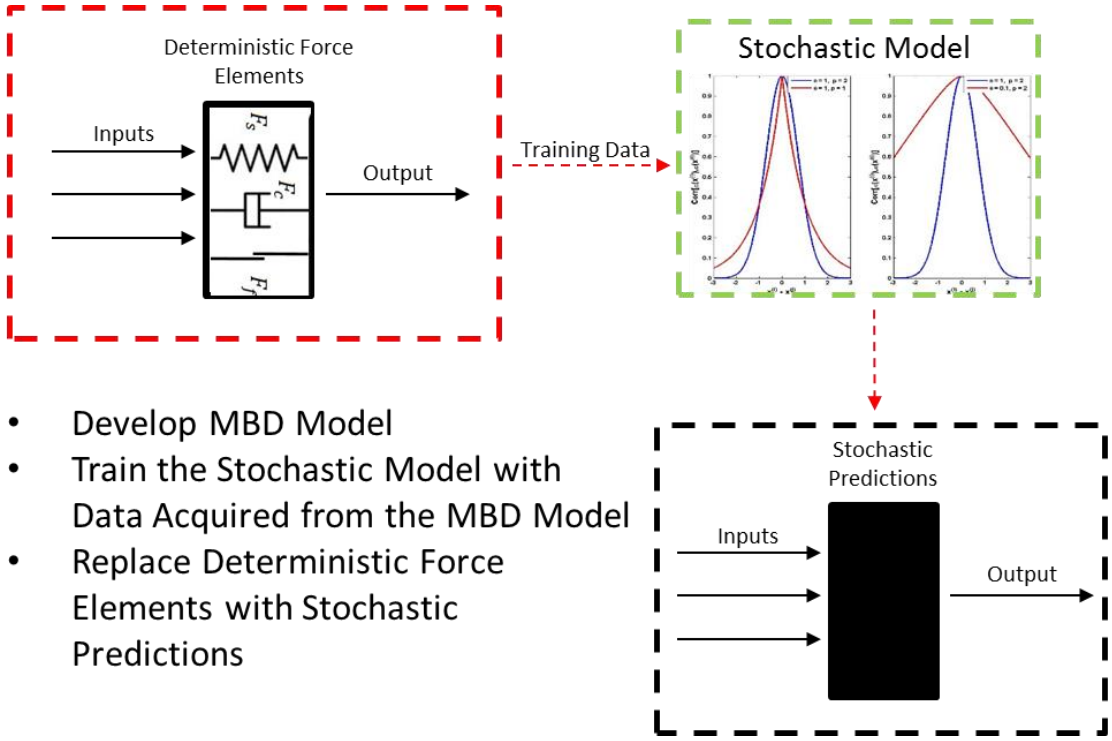

Figure 1.4: Schematic of the approach taken in this research

1.4 Literature Review

To fulfill the objectives of this research, an extensive review of literature has been conducted. In this section, the literature reviewed for this research is discussed. The reviewed literature can be divided into two separate sections: 1) literature regarding the dynamics of the three-piece truck, and 2) literature regarding the development of the stochastic process.

1.4.1 Three-piece Truck

The literature regarding the dynamics of the three-piece truck is mainly focused on two areas: the contact between the wheel and the rail and the resulting creep forces, and the dynamics of

the secondary suspension of the truck and specifically the dynamics of the friction wedge within the secondary suspension.

There are also a number of books on the dynamics of railway vehicles that provide a comprehensive description of all the concepts and theories used for the modeling of passenger and freight railway vehicles. Simon Iwnicki's "Handbook of Railway Vehicle Dynamics" [1] provides an extensive review of the dynamics of rail vehicles and can be used as a starting point in any research regarding the dynamics of railway vehicles.

Vijay K. Garg and Rao V. Dukkipati's "Dynamics of Railway Vehicle Systems" [2] provides all the necessary information for researchers who are interested in deriving and solving the equations of motion for a railway vehicle and analyzing the results.

Wheel/Rail interaction

SIMPACK's rail module provides the necessary force elements for the calculation of the dynamic forces between the wheel and the rail, hence, the reviewed literature is focused on better understanding the dynamics rather than the complicated numerical techniques used to accurately and efficiently estimate these forces.

Kalker's theories are the most popular wheel/rail contact theories and are widely used in multibody dynamics analysis. In [3], J. Kalker reviews a number of the most popular rolling contact theories, including the two-dimensional contact theory developed by Carter, Kalker's linear theory, Kalker's complete theory (CONTACT), the theory of Shen, Hendrick and Elkins, and Kalker's simplified theory or FASTSIM. The paper also reviews the Hertzian contact theory that is

used for the estimation of the normal forces between the wheel and the rail. Except for the CONTACT theory that is not restricted by the assumptions of the Hertzian contact theory, all the aforementioned theories are confined to the assumption of elliptical contact areas. The paper also compares the estimated creepage forces of each theory and makes recommendations on which algorithm is most suitable for the calculation of wheel/rail forces in various vehicle dynamic analysis.

Kalker's FASTSIM algorithm [4] is the most widely used theory for the calculation of wheel/rail creepage forces for multibody dynamics analysis. The algorithm is 15-25 times faster than similar programs, and the relative forces calculated using this algorithm are within 20% of the forces calculated by the exact theory of [5]. The algorithm divides the contact area into horizontal sections, then the algorithm calculates the traction in each slice using the refined form of Coulomb's law.

Three-piece truck's secondary suspension

An extensive literature review has been conducted to fully understand the dynamics of the three-piece truck. An accurate and efficient model of a freight rail vehicle requires a deep understanding of various bodies in the system and their interactions.

In his PhD dissertation, Fuji Xia [6] provides a detailed model of the truck with a state-of-the-art friction wedge model. The concept of friction direction angle is introduced and used to identify the components of the friction force vector in the vertical and lateral planes.

Peter Klauser in [7], [8] develops the model and derives the equations for a simple friction wedge, where the mass of the wedge is neglected and it is modeled as a constraint that transfers the normal loads from the bolster to the side-frame. The model is computationally inexpensive and is used in the rail vehicle dynamic software NUCARS. Gardner and Cusumano [9] take a similar approach in modeling the friction wedge. Their model considers the inertial properties of the friction wedge, but the wedge is kinematically constrained to the bolster and it is assumed that the wedge is always in motion relative to the bolster.

The friction wedge model used in the three-piece truck model developed for the purpose of this research is based on the model developed by Ballew [10]. The friction wedge is modeled as a separate body with four degrees of freedom (vertical, longitudinal, pitch, and yaw). Frictional surfaces of the wedge are assumed to be flat with contact forces on all the corners of the wedge.

1.4.2 Machine Learning

An extensive multidisciplinary review of literature has been conducted to acquire an exhaustive understanding of the concepts that are required for the development of the stochastic model. The model utilizes three sub-algorithms that (1) select the training dataset from the acquired data, (2) find the optimum hyper-parameters for the correlation function, and (3) estimate the output at an untried location.

Optimization algorithms are an essential tool for the selection of the training data, and for training the stochastic model. Venkataraman's "Applied Optimization with MATLAB Programming" [11] provides a detailed review of various local and global optimization techniques. The book is a great starting point for learning and implementing basic optimization concepts.

David Goldberg's "Genetic Algorithms in Search, Optimization, and Machine Learning" [12] provides the necessary mathematical foundations and computer implementation techniques to understand the concepts of finding the optimum using a genetic algorithm and developing the computer algorithm.

The machine learning algorithm developed for the purpose of this research is based on the design-surface concept. Design-surfaces are used to replace computationally expensive computer simulations for the purpose of optimization. Donald R. Jones [13] provides a detailed review of various concepts that are popular in developing surrogate or black-box models of computer simulations. A more practical approach has been used by Forrester et al. [14], where the algorithms for the surrogate modeling technique are explained in detail and examples of their application in the real world are provided.

1.5 Contribution

This research focuses on the application of stochastic modeling techniques to learn the behavior of an MBD model or a physical system. The stochastic model can be used to reduce the simulation time for the MBD model or to incorporate the behavior of a physical system within the MBD model. The main contributions can be summarized as:

1. Modifying the concept of stochastic modeling of a deterministic system to learn the behavior of an MBD model.
2. Incorporating the uncertainty that is associated with laboratory testing in the algorithm that is used to learn the behavior of a physical system.

3. Developing a sampling plan that is the most space-filling sampling plan considering the fact that the acquired data does not fill the entire hypercube of the design space.
4. Developing infill criteria that detect the locations where the stochastic model has difficulty predicting the behavior of the system and improves the efficiency of the stochastic model by including additional sampling points.

1.6 Document Outline

The document is organized as follows. Chapter 2 describes the details of the MBD model for the three-piece truck that is used for the purpose of this study. In chapter 3, we begin with a brief discussion on the statistical concepts that are used to build the stochastic model, and then we will build the stochastic model and provide a detailed description of the process. Chapter 4 contains the case studies of applying the stochastic model to replace parts of an MBD model or to integrate laboratory data into an MBD model; at the end, we will discuss the results obtained and potential improvements. In chapter 5, a summary and conclusion of the work described in this document will be provided. This chapter also includes researcher's suggestions regarding the future work that can be done to further improve the stochastic model.

2 Three-piece Truck's Dynamics and Kinematics

2.1 Introduction

The three-piece truck model built for the purpose of this research is discussed in detail in this chapter. Although the design of the three-piece truck is simplistic, due to the existence of numerous nonlinearities in the system, accurate modeling of the truck can be challenging. System nonlinearities are rooted in numerous frictional surfaces, design and assembly clearances modeled as dead-band springs, wheel/rail contact, and creep forces generated from the contact between a profiled rail and wheel that consequently results in a highly nonlinear, chaotic system[15], [16].

Figure 2.1 shows the schematics of a typical three-piece truck. There are two trucks (bogies) under each car, and the connection is through a pivoted joint between the bolster's center plate and carbody's center pin; side bearings at the two ends of the bolsters provide additional support for the carbody. The weight of the carbody is transferred from the bolster to side-frames via the spring nests that are located at the two ends of the bolster. There are many different configurations for the spring nests, but in general they consist of several inner and outer coil springs; since the coil springs provide little damping to the system, spring-loaded friction wedges are added at each corner. Vibratory energy is dissipated through dry friction when there is relative lateral or vertical movement in the secondary suspension. The connection between the side-frame and the wheelsets is through axle bearings (primary suspension). The components are

held together by gravity, giving the design the advantage of being able to easily disassemble and assemble for maintenance purposes.

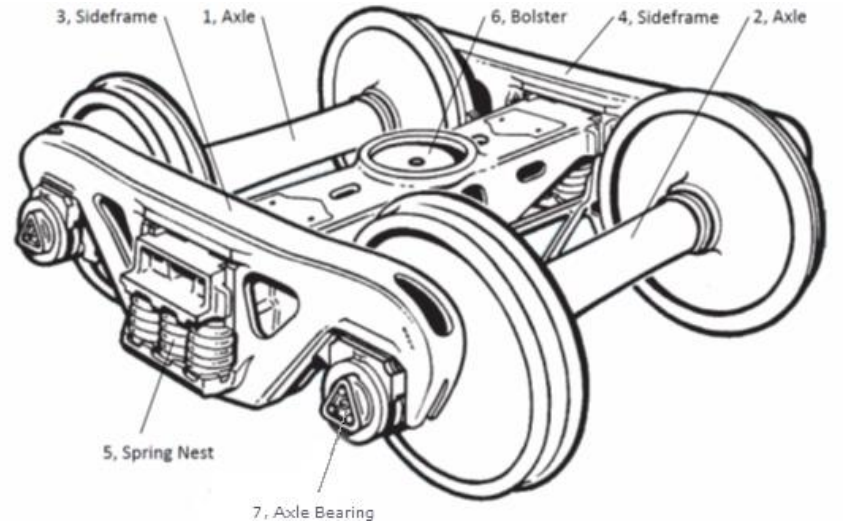

Figure 2.1: Schematics of a three-piece truck [17]

2.1.1 Overview

It is important to create a multibody dynamics model that is accurate in predicting the performance of the vehicle while it utilizes simplifying assumptions that reduce the computational expense of the model. A commercially available multibody dynamics software package (SIMPACK) is used to model the rail vehicle with three-piece trucks. The model consists of two trucks that are duplicated using the substructure functionality of SIMPACK and a carbody. In total, the model consists of 19 bodies, 82 DOF, and 108 linear and nonlinear force elements.

SIMPACK is a general purpose multibody dynamics simulation software that is widely used for analysis and design of mechanical and mechatronic systems. SIMPACK has a modular structure that enables it to generate models easily and exchange components for analyses where different levels of accuracy are required. SIMPACK can be used for different rail vehicle dynamic studies

such as curving, wheel-rail contact force evaluation, passenger comfort, running stability, critical hunting speed, and other types of system analyses [18].

In the following sections of this chapter, kinematic and dynamic relations between various bodies will be discussed in detail. Figure 2.2 (top) shows a 3D overview of the SIMPACK model developed for the purpose of this research. SIMPACK also provides a 2D view of the model (Figure 2.2 (bottom) and Figure 2.3); the connection between bodies and the force elements used in the model can be easily identified in the 2D view. This view is the most useful for debugging once the model is developed.

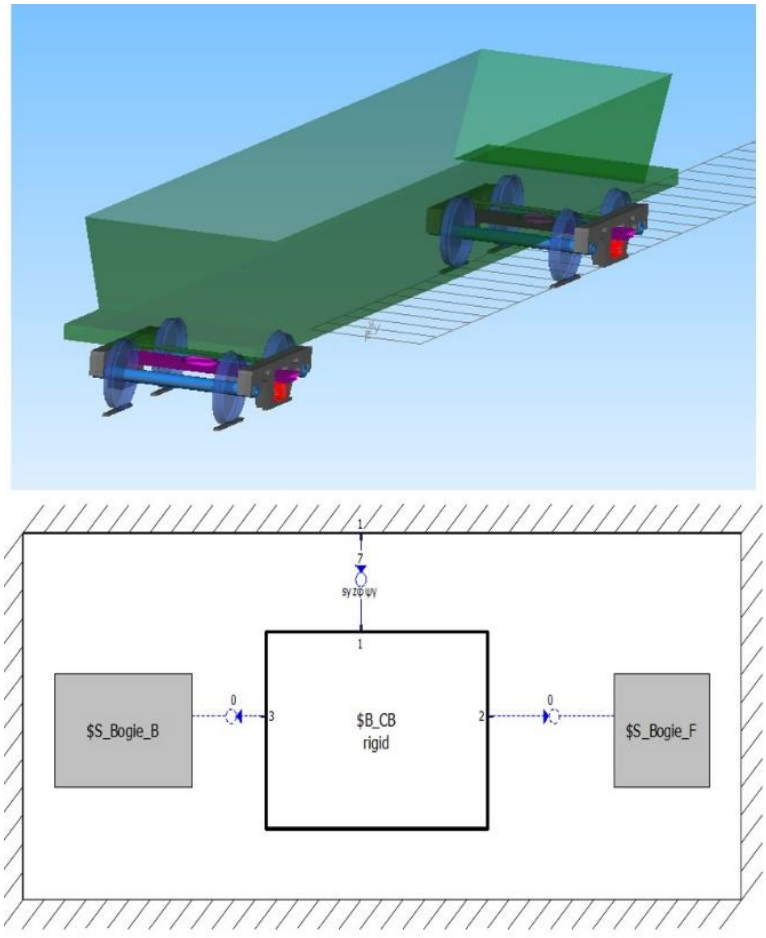

Figure 2.2: Overview of the rail vehicle model developed: 3D view (top), 2D view (bottom)

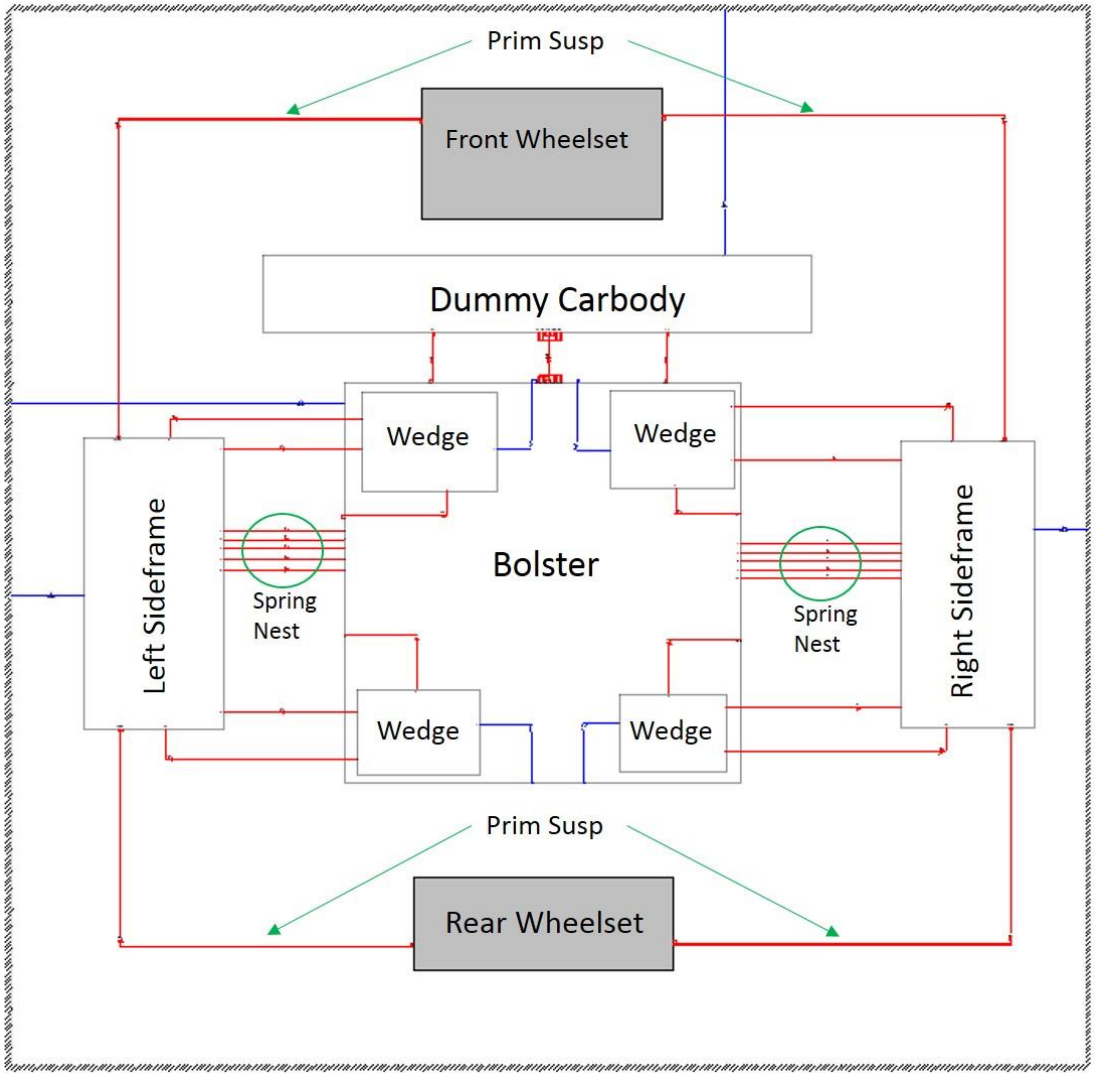

Figure 2.3: 2D view of the bogie substructure model

2.1.2 Model Parameters

The first step in every multibody dynamic modeling attempt is to investigate the system under study to identify various bodies, their interactions, and the resulting degrees of freedom. In 3D space, each rigid body has 6 DOF (3 translational and 3 rotational); to reduce the computational expense of the model, several degrees of freedom are constrained, resulting in 82 DOF instead of $19 \times 6 = 114$ DOF. Table 2.1 shows the allowed degrees of freedom for each body.

Table 2.1: Degrees of Freedom of the system[2]

Body	Longitudinal	Lateral	Vertical	Roll	Yaw	Pitch
Wheelset	✓	✓	✓	✓	✓	✓
Side-frame	✓	✓	✓	✓	✓	×
Bolster	✓	✓	✓	✓	✓	✓
Friction Wedge[3]	✓	×	✓	×	×	×
Carbody	✓	✓	✓	✓	✓	✓

As for any other modeling attempts, the dynamics of the system is highly influenced by the values used for mass and moments of inertia for various bodies in the system. Table 2.2 shows the values used for these parameters. Values were extracted from private conversations with multiple industry partners of Railway Technology Laboratory (RTL) and are also based on what has been used in previous studies [19], [20].

Table 2.2: Mass and moment of inertia for various bodies

Body	Mass $(\frac{lbf.s^2}{in})$	I_{xx} $(lbf.in.s^2)$	I_{yy} $(lbf.in.s^2)$	I_{zz} $(lbf.in.s^2)$	C.G Height (in)
Wheelset	7.73	5,885.74	1,752	5,885.74	18in
Side-frame	2.98	876.22	2,168.43	1,593.13	18in
Bolster	3.78	2,832.23	203.56	2,832.23	18in
Friction Wedge	0.04	-	-	-	-
Carbody (Empty)	114.2	376,156.69	5,310,447.47	5,310,447.47	80in
Carbody (Full)	628.11	1,840,000	16,700,000	16,700,000	83in

2.1.3 Friction Models

As mentioned earlier, the three-piece truck utilizes several frictional components to dissipate the vibratory energy resulted from track irregularities, hence it is important to use a computationally

[2] Degrees of freedom for all bodies are relative to the global coordinate system

[3] The friction wedge's DOFs are relative to the bolster and not the global coordinate system

efficient friction force that can also accurately represent the contact between various parts in the three-piece truck.

There are numerous ways of representing the friction force in a multibody dynamics model [21]. Coulomb's friction force (Equation (2.1)) is the most widely used force law for simplified friction models [22]:

$$F_f = \mu N sign(v) \tag{2.1}$$

where F_f is the friction force, μ is the coefficient of friction, N is the normal force, and v is the relative velocity of contacting bodies. The formulation presented in Equation (2.1) has two major problems: it predicts zero friction force when the relative velocity is equal to zero, and F_f is discontinuous around $v = 0$, which can cause problems during simulation. To overcome the problems associated with the discontinuity of Coulomb's friction, a combination of viscous and Coulomb's friction force is used (Figure 2.4). Equation (2.2) shows the force law for the Coulomb-viscous friction force:

$$F_f = \begin{cases} \mu N sign(v) & v > v_{ref} \text{ or } v < -v_{ref} \\ \dfrac{\mu N}{v_{ref}} v & -v_{ref} \leq v \leq v_{ref} \end{cases} \tag{2.2}$$

where v_{ref} is the velocity where the force law switches between Coulomb and viscous force laws. Another possibility to overcome the discontinuity of the Coulomb friction force is to use a $tanh$ function instead of the $sign$ function in Equation (2.1) (Equation (2.3)) [21].

$$F_f = \mu N tanh\left(\frac{v}{v_{ref}}\right) \tag{2.3}$$

As mentioned earlier, Coulomb's friction law predicts a zero friction force while the object is stationary, but in reality, friction force is equal to the sum of the applied forces on the body. Stick-slip friction models can capture this behavior (Figure 2.4) [23], [24].

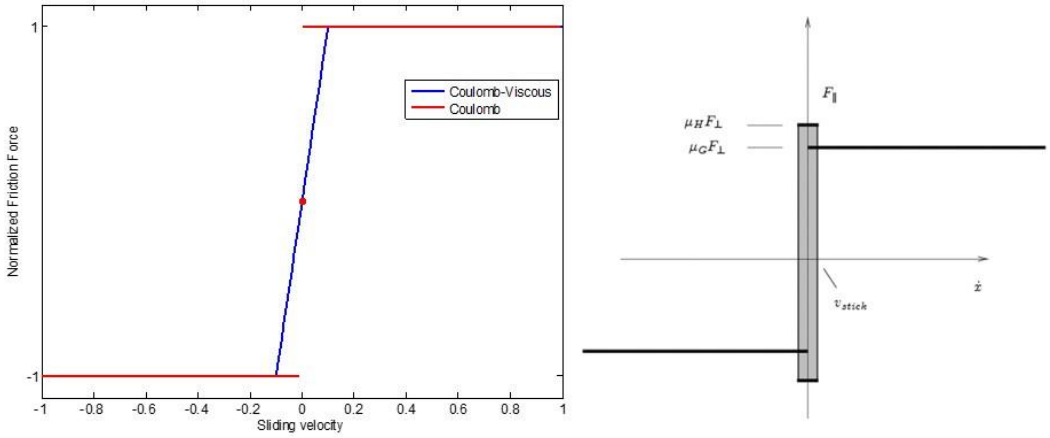

Figure 2.4: Coulomb and Coulomb-viscous friction models (left); discontinuity causes problems during integration. Stick-slip friction model (right)

Stick-slip friction force elements are more accurate but increase the computational expense of the model, especially due to switching from one force law to another; integrators need to reduce the step size to accurately capture the moment when the switching occurs. Xia introduces a stick-slip friction model that uses friction direction angle to model the 2D friction force on the surfaces of the friction wedge. In his model, vibrations in one direction influence the friction force in the perpendicular direction [6]. The stick-slip friction element used for the purpose of this study is a simplified version of the LuGre model [25] that allows for the separation of the contacting bodies.

$$\begin{cases} N = 0 & \delta > 0 \\ N = c_z\delta + d_z\dot{\delta} & \delta < 0 \end{cases} \tag{2.4}$$

$$\begin{cases} F_f = 0 & \delta > 0 \\ F_f = \sigma_1 \Delta x + \sigma_2 v & \delta < 0, |v| \le v_{slip} \\ F_f = \mu_k N sign(v) & \delta < 0, |v| \ge v_{slip} \; or \; |F_f| \ge \mu_s * N \\ \Delta x = \int \Delta v \end{cases} \tag{2.5}$$

where N is the normal force, c_z, d_z are the stiffness and damping coefficients for the elastic contact between the bodies, δ is the penetration length of the contacting bodies, σ_1, σ_2 are stiffness and damping coefficients for the stick state, μ_k is the kinematic coefficient of friction, and μ_s is the static cooefficient of friction. Figure 2.5 provides a schematic of the stick-slip friction element used in the SIMPACK model.

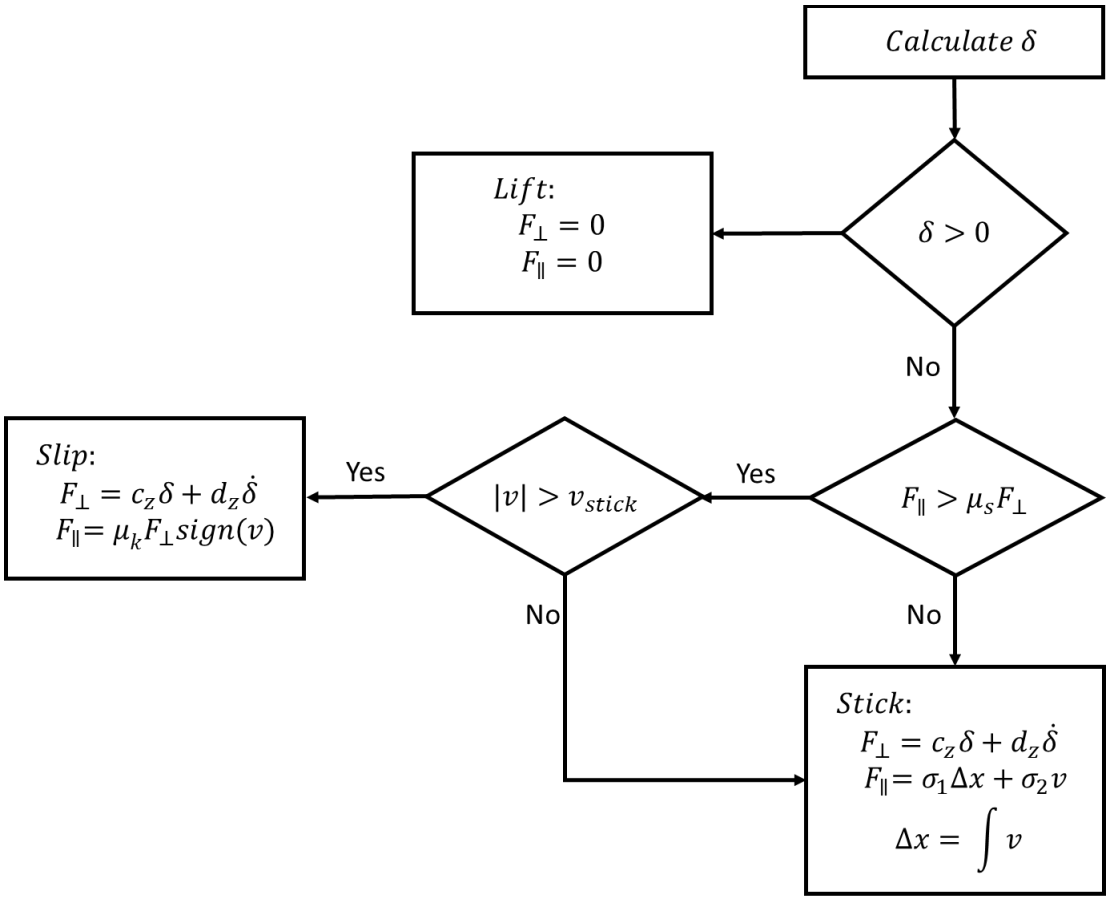

Figure 2.5: Schematics of the stick-slip friction force used

2.2 Modeling of Wheel/Rail Normal and Tangential Forces

2.2.1 Introduction

Accurate estimation of wheel/rail interaction forces has been the subject of numerous research studies [13–17]. The problem can be separated into two categories: 1) normal forces, 2) tangential forces. Normal forces result from the contact between a steel wheel and a steel rail whereas tangential forces are produced when there is relative movement between the wheel and the rail (creepage).

Hertizian contact theory is the most widely used theory for calculating wheel/rail normal forces [29]; finite element methods [30] can also be used to calculate the contact forces but since they are computationally expensive, analytical methods like the Hertzian contact theory are preferred for dynamic analysis. The problem of wheel/rail tangential forces is a rolling friction problem that differs from the sliding friction problem described in section 2.1.3, where contact is assumed to take place in a single point. In rolling friction, the area of adhesion and the area of slip depend on the creepage. Carter's two-dimensional theory [27], and Kalker's linear [31], nonlinear and empirical theories are among the most popular wheel-rail creepage theories. We will discuss a number of popular theories in the following sections. FASTSIM [4] is the most widely used theory for multibody dynamics simulation of rail vehicles. The simplified algorithm is 15-25 times faster than the previous codes and the difference is, at most, 0.2 percent [4].

2.2.2 Hertzian Contact

Hertz's contact theory is based on several assumptions: 1) material properties for the contacting bodies are assumed to be linear, 2) material behavior is limited to the elastic region (no plastic deformation), 3) contacting materials (wheel/rail) are homogeneous and isotropic, 4) contacting surfaces are perfectly smooth and frictionless, 5) the curvature is constant within the contact patch, and 6) radii of the contacting bodies are much larger than radii of the contact ellipse [1], [32]. Figure 2.6 shows the schematics of the contacting bodies. In the case of wheel-rail contact, r_{11} is the effective rolling radius of the wheel, r_{12} is considered to be infinite, r_{21} is the rolling radius of the rail that is considered to be infinite for a tangential rail, and r_{22} is the crown radius of the rail [32].

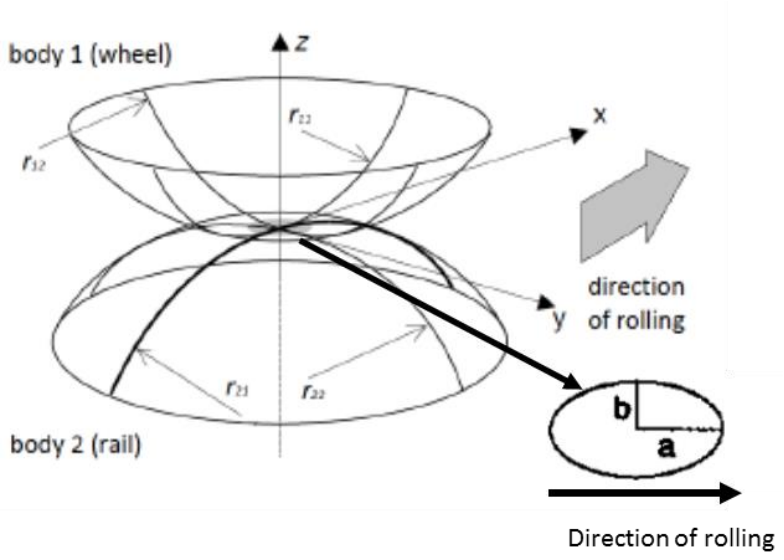

Figure 2.6: Hertzian contact problem

Contact patch dimensions and pressure distribution are functions of the normal force, material properties, and longitudinal and transverse curvature of the contacting bodies [33].

$$a = m\left(\frac{3N(1-\nu^2)}{2E(A+B)}\right)^{\frac{1}{3}}$$
$$b = n\left(\frac{3N(1-\nu^2)}{2E(A+B)}\right)^{\frac{1}{3}} \tag{2.6}$$

where a and b are the longitudinal and transverse radii of the contact ellipse, N is the normal force, A and B are functions of the geometry of the contacting bodies, and m and n are Hertzian coefficients that can be calculated using tables in [3] or can be approximated with the closed-form relationships given by Equation (2.8) [32].

$$A = \frac{1}{2}\left(\frac{1}{r_{11}} + \frac{1}{r_{21}}\right)$$
$$B = \frac{1}{2}\left(\frac{1}{r_{12}} + \frac{1}{r_{22}}\right) \tag{2.7}$$

$$m = \left(\frac{A}{B}\right)^{-0.315}\left(\frac{1+\frac{A}{B}}{2\sqrt{\frac{A}{B}}}\right)^{0.21}$$
$$n = \left(\frac{A}{B}\right)^{0.315}\left(\frac{1+\frac{A}{B}}{2\sqrt{\frac{A}{B}}}\right)^{0.21} \tag{2.8}$$

Figure 2.7 shows the geometric properties of a typical profiled wheel and rail that are commonly used by the freight industry. Wear can significantly change the profiles throughout their service life, but for the purpose of this research, the profiles are assumed to be new.

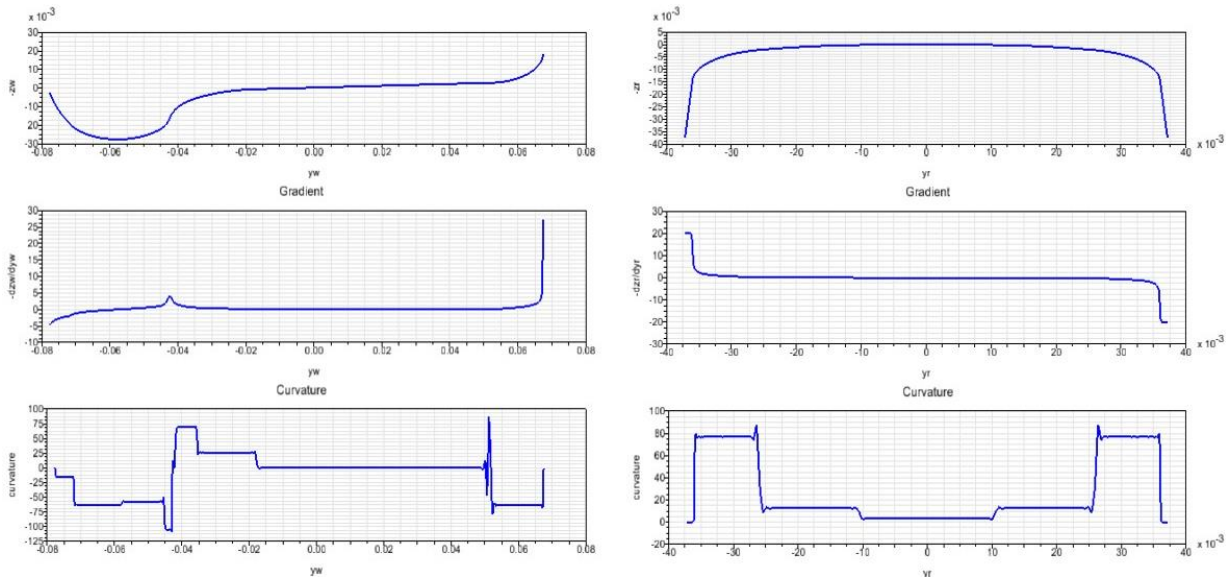

Figure 2.7: Geometric characteristics of wheel with AAR1B profile (left), and rail with UIC60 profile (right)

2.2.3 Creepage

As mentioned earlier, creepage is the difference in the circumferential velocity of the two contacting bodies while rolling over each other. Creepage is defined in longitudinal (x), lateral (y), and spin (ϕ) directions. Creepage is a dimensionless term that is calculated from the projection of the differential velocity of the two contacting bodies normalized by their mean velocities on the respective axis; Equation (2.9) provides the mathematical equations for creep [1].

$$v_x = \frac{proj./x(\overrightarrow{V_0} - \overrightarrow{V_1})}{\frac{1}{2}(\overrightarrow{V_0} + \overrightarrow{V_1})}$$

$$V_y = \frac{proj./y(\overrightarrow{V_0} - \overrightarrow{V_1})}{\frac{1}{2}(\overrightarrow{V_0} + \overrightarrow{V_1})} \tag{2.9}$$

$$\phi = \frac{proj./z(\overrightarrow{\Omega_0} - \overrightarrow{\Omega_1})}{\frac{1}{2}(\overrightarrow{V_0} + \overrightarrow{V_1})}$$

where $\vec{V}_0$ and $\vec{V}_1$ are the absolute velocities of wheel and rail, respectively, and $\vec{\Omega}_0$ and $\vec{\Omega}_1$ are their respective spin velocities. Note that v_x and v_y are dimensionless, but ϕ has the dimension of $\frac{1}{m}$. Mean velocity $\frac{1}{2}\left(\overrightarrow{V_0} + \overrightarrow{V_1}\right)$ is used in the denominator so that the case for pure rolling or pure sliding where V_0 or V_1 is zero can be handled by the algorithm.

It is worth noting that longitudinal, lateral, and spin creep depend on the differential velocity of the wheel and rail, and they are independent from each other, i.e., changing one will not change the other. Furthermore, creepages do not affect contact patch shape, size, or the distribution of the normal pressure within it [32].

2.2.4 Kalker Linear Model

Creepages described in the previous section treat wheel and rail as rigid bodies, whereas in reality, they deform near the contact patch that results in the division of the contact patch into adhesion and slip areas. Kalker's linear theory is based on the assumption that for small creepages, slip area is small and can be neglected compared to the adhesion area. Kalker's linear theory uses Hertzian contact for the calculation of the normal forces. Normalized creep forces (creep forces per unit creepage) are calculated by integrating the tangential stresses over the contact patch area.

Kalker's linear theory proposes the relationship between creepages and creep forces using matrix notation and in the following form:

$$\begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} = -Gab \begin{bmatrix} c_{11} & 0 & 0 \\ 0 & c_{22} & \sqrt{ab}c_{23} \\ 0 & \sqrt{ab} & c_{33} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \phi \end{bmatrix} \tag{2.10}$$

where G is the shear modulus, a and b are radii of the contact patch, and c_{11}, c_{22}, c_{23}, and c_{33} are Kalker's creep coefficients that are calculated by Interpolating Table 2.3 based on the ratio of $\frac{a}{b}$ and the Poisson ratio of the material. Kalker's coefficients can also be calculated using polynomial fits to Table 2.3 (Equation (2.11)).

$$c_{11} = 3.2893 + \frac{0.975}{b/a} + \frac{0.012}{(b/a)^2}$$

$$c_{22} = 2.4014 + \frac{1.3179}{b/a} + \frac{0.02}{(b/a)^2} \tag{2.11}$$

$$c_{23} = 0.4147 + \frac{1.0184}{(b/a)} + \frac{0.0565}{(b/a)^2} - \frac{0.0013}{(b/a)^3}$$

Table 2.3: Kalker's coefficients table [1]

		C_{11}			C_{22}			$C_{23} = -C_{32}$			C_{33}		
	g	$\sigma = 0$	1/4	1/2	$\sigma = 0$	1/4	1/2	$\sigma = 0$	1/4	1/2	$\sigma = 0$	1/4	1/2
	0.0	$\pi^2/4(1-\sigma)$			$\pi^2/4 = 2{,}47$			$\pi\sqrt{g}/3$	—	—	$\pi^2/16(1\text{-}\sigma)g$		
a/b	0.1	2.51	3.31	4.85	2.51	2.52	2.53	0.334	0.473	0.731	6.42	8.28	11.7
	0.2	2.59	3.37	4.81	2.59	2.63	2.66	0.483	0.603	0.809	3.46	4.27	5.66
	0.3	2.68	3.44	4.80	2.68	2.75	2.81	0.607	0.715	0.889	2.49	2.96	3.72
	0.4	2.78	3.53	4.82	2.78	2.88	2.98	0.720	0.823	0.977	2.02	2.32	2.77
	0.5	2.88	3.62	4.83	2.88	3.01	3.14	0.827	0.929	1.07	1.74	1.93	2.22
	0.6	2.98	3.72	4.91	2.98	3.14	3.31	0.930	1.03	1.18	1.56	1.68	1.86
	0.7	3.09	3.81	4.97	3.09	3.28	3.48	1.03	1.14	1.29	1.43	1.50	1.60
	0.8	3.19	3.91	5.05	3.19	3.41	3.65	1.13	1.25	1.40	1.34	1.37	1.42
	0.9	3.29	4.01	5.12	3.29	3.54	3.82	1.23	1.36	1.51	1.27	1.27	1.27
b/a	1.0	3.40	4.12	5.20	3.40	3.67	3.98	1.33	1.47	1.63	1.21	1.19	1.16
	0.9	3.51	4.22	5.30	3.51	3.81	4.16	1.44	1.59	1.77	1.16	1.11	1.06
	0.8	3.65	4.36	5.42	3.65	3.99	4.39	1.58	1.75	1.94	1.10	1.04	0.954
	0.7	3.82	4.54	5.58	3.82	4.21	4.67	1.76	1.95	2.18	1.05	0.965	0.852
	0.6	4.06	4.78	5.80	4.06	4.50	5.04	2.01	2.23	2.50	1.01	0.892	0.751
	0.5	4.37	5.10	6.11	4.37	4.90	5.56	2.35	2.62	2.96	0.958	0.819	0.650
	0.4	4.84	5.57	5.57	4.84	5.48	6.31	2.88	3.24	3.70	0.912	0.747	0.549
	0.3	5.57	6.34	7.34	5.57	6.40	7.51	3.79	4.32	5.01	0.868	0.674	0.446
	0.2	6.96	7.78	8.82	6.96	8.14	9.79	5.72	6.63	7.89	0.828	0.601	0.341
	0.1	10.7	11.7	12.9	10.7	12.8	16.0	12.2	14.6	18.0	0.795	0.526	0.228

2.2.5 Johnson and Vermeulen's Theory

In Johnson and Vermeulen's theory [28] contact patch is unsymmetrically divided into two regions: 1) slip area and 2) adhesion area. As can be seen in Figure 2.8, the adhesion area is assumed to be a small area that starts from the leading edge of the contact patch. The total tangential traction is the difference of the semi-ellipsoidal tangential traction acting on each ellipse. Experimental results showed a maximum of 25% error in the predictions, which was attributed to the assumption of elliptical adhesion region [34]. The disadvantage of Johnson and Vermeulen's theory is that they neglect the effects of spin creepage that are known to be significant, especially near the flange region [2].

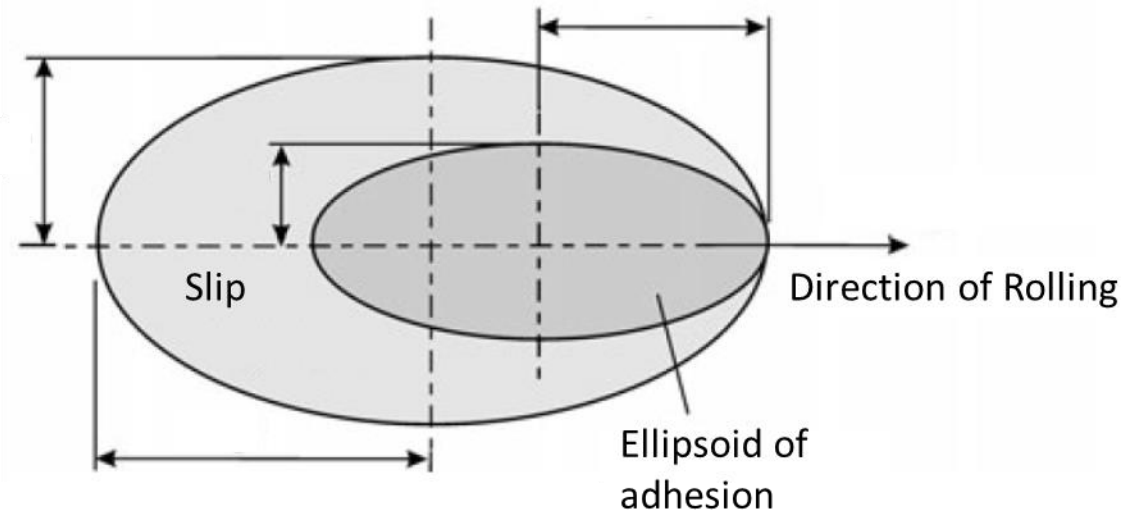

Figure 2.8: Johnson and Vermeulen's contact patch [32]

Since the primary purpose of this work is not the review of various rolling contact theories, only a brief introduction on some of the popular theories are presented and mathematical formulations are not discussed; readers interested in detailed discussions of the theories can use the cited material for further information.

2.2.6 FASTSIM

FASTSIM [3] is the most widely used theory for the study of wheel/rail tangential forces. The contact patch is divided into independent longitudinal parallel strips (Figure 2.9). Whereas creep is calculated at the contact patch in each direction, the contact pressure and creepage forces are computed at the center of the contact patch.

$$p_x(x, y_i) = (\frac{v_x}{L_1} - y_i \frac{\varphi}{L_3})(x - a_i)$$

$$p_y(x, y) = \frac{v_y}{L_2}(x - a_i) + \frac{\varphi}{2L_3}(x^2 - a_i^2) \tag{2.12}$$

where L_1, L_2, L_3 are elasticity coefficients, a_i is the coordinate for the leading edge of the strip, and v_x, v_y, φ are longitudinal, lateral, and spin creepages, respectively [1], [4].

$$L_1 = \frac{8a}{3c_{11}G}, \quad L_2 = \frac{8a}{3c_{22}G}, \quad L_3 = \frac{\pi a\sqrt{a/b}}{4c_{23}G} \tag{2.13}$$

where c_{11}, c_{22}, c_{23} are Kalker's coefficients that can be calculated using tables in [3] based on a/b ratio. Forces are calculated by integrating the surface stresses over the entire contact patch.

$$F_x = -\iint p_x(x)dxdy = -\frac{8a^2bv_x}{3L_1}$$
$$F_y = -\iint p_y(x)dxdy = -\frac{8a^2bv_y}{3L_2} - \frac{\pi a^3 b\varphi}{4L_3} \tag{2.14}$$

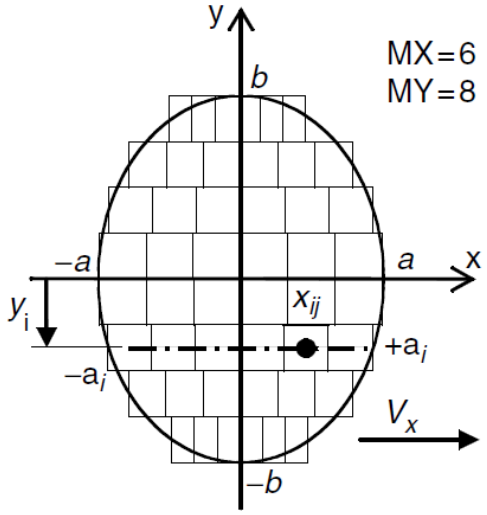

Figure 2.9: Contact patch discretization for FASTSIM [1]

A comparison has been done using a single wheelset to study the difference between the simplified algorithm of FASTSIM and the exact solution given by the CONTACT[4] software [26]. The wheelset is moved in the lateral and yaw directions according to Figure 2.10, and lateral and longitudinal creep forces are compared for the results produced by FASTSIM and CONTACT. Figure 2.11 and Figure 2.12 show the results of the comparison.

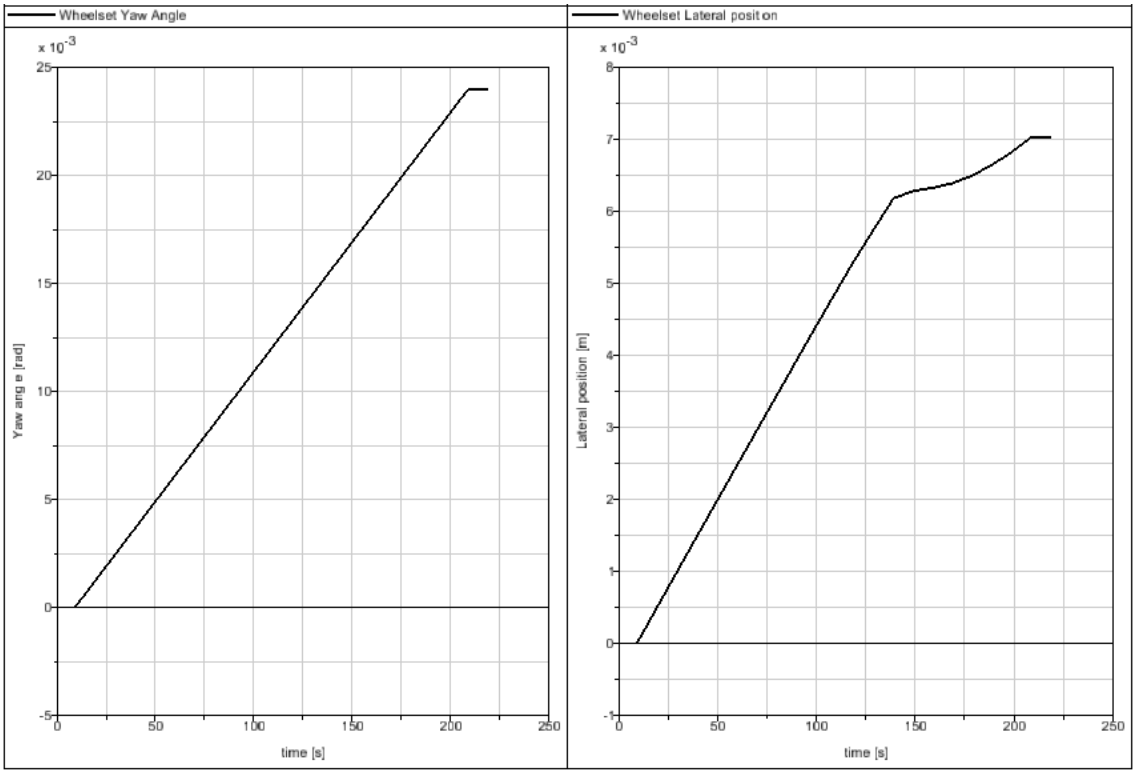

Figure 2.10: Wheelset position in lateral (right), yaw (left) directions

[4] CONTACT is an advanced simulation program for the detailed study of three-dimensional frictional contact

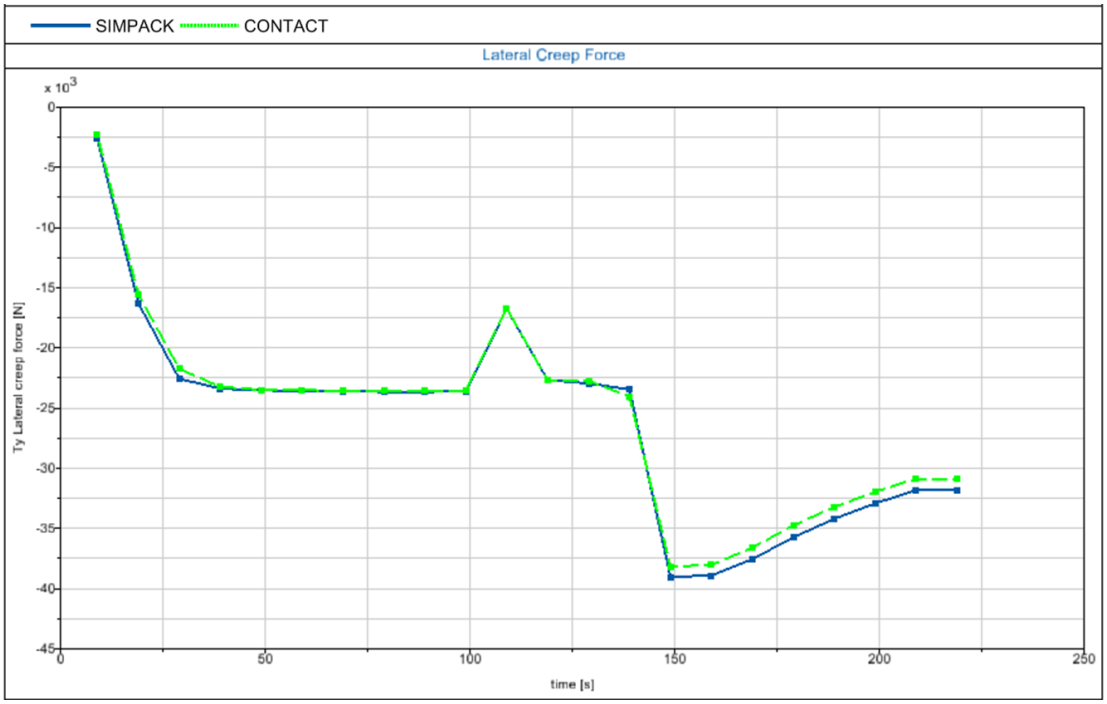

Figure 2.11: Lateral creep forces closely match between FASTSIM and CONTACT

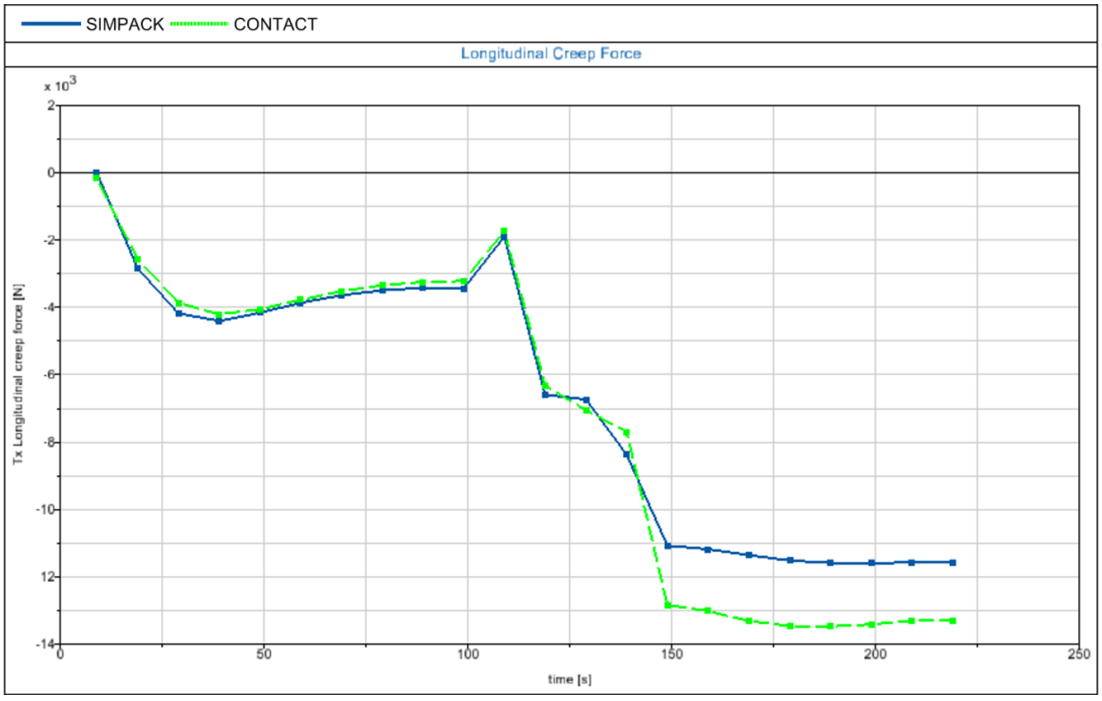

Figure 2.12: Longitudinal creep forces deviate for larger wheelset lateral and yaw displacements

2.3 Primary Suspension

Primary suspension refers to the connection between the wheelsets and the side-frames. The side-frame's pedestal legs sit on top of the roller bearings at the two ends of the wheelset. The term "suspension" is loosely used in the context of the connection between the wheelsets and the side-frame. As mentioned earlier, suspension needs to have two functions, 1) compliance, to absorb the energy, and 2) damping, to dissipate the resulted vibratory energy. Since there is almost no compliance in the connection between the roller bearings and the pedestal legs, the primary suspension of a three-piece truck barely passes the criteria of a suspension element. Although the connection is rigid due to the steel on steel contact, the bodies can move relative to each other in the lateral and longitudinal clearances, dissipating energy through friction. Recently, the railroad industry has moved towards the use of elastomers in the primary suspension to improve the performance of the truck by increasing the compliance of the primary suspension and increasing the coefficient of friction. The primary suspension allows the wheelset to move relative to the side-frames and helps to reduce the transmission of vibrations to the carbody [2].

The primary suspension of the three-piece truck model developed in this research has two spring elements in the vertical direction, with high spring rates that are placed 3 inches apart from each other in the lateral direction. This arrangement can compensate for the very small roll motion of the side-frames. Dead-band springs in the longitudinal and lateral directions represent the clearances between the roller bearings and the pedestal legs. Energy dissipation is done through dry friction in lateral and longitudinal directions. To accurately capture the friction phenomenon,

stick-slip friction elements described in section 2.1.3 are used. Figure 2.13 shows the elements of the primary suspension.

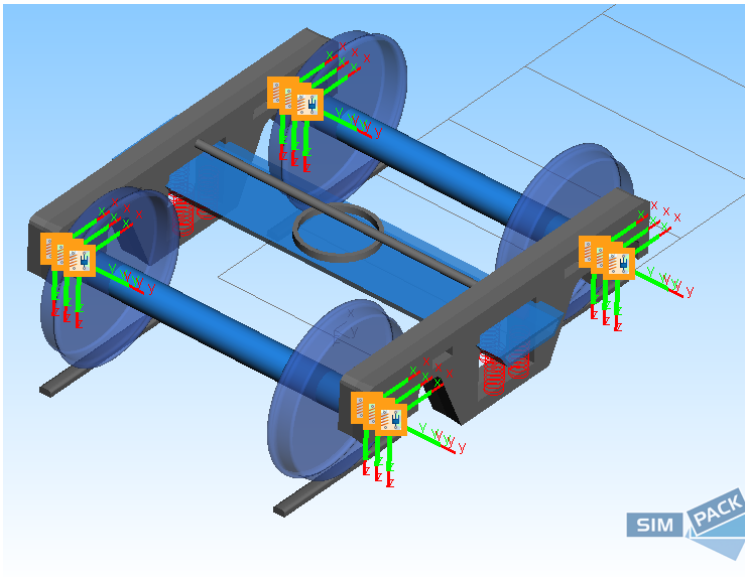

Figure 2.13: Three-piece truck's primary suspension

Figure 2.14 and Figure 2.15 show the behavior of the primary suspension's lateral and longitudinal forces, respectively, during a hunting study. It can be seen from these figures that the behavior of the system, as mentioned earlier, is highly nonlinear.

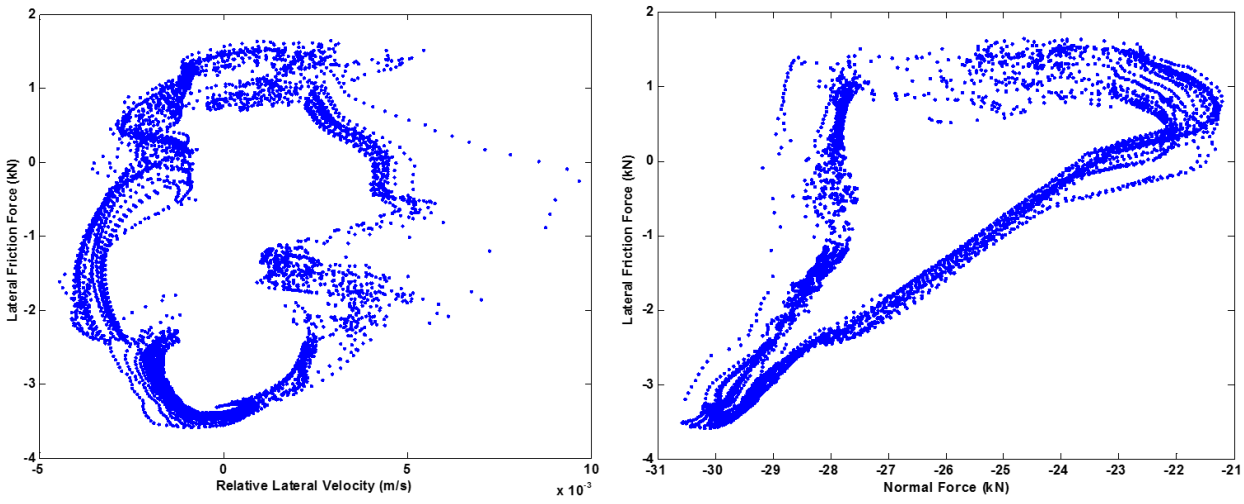

Figure 2.14: Lateral friction force in the primary suspension as a result of relative velocity between the wheelset and the side-frame (left), normal force (right)

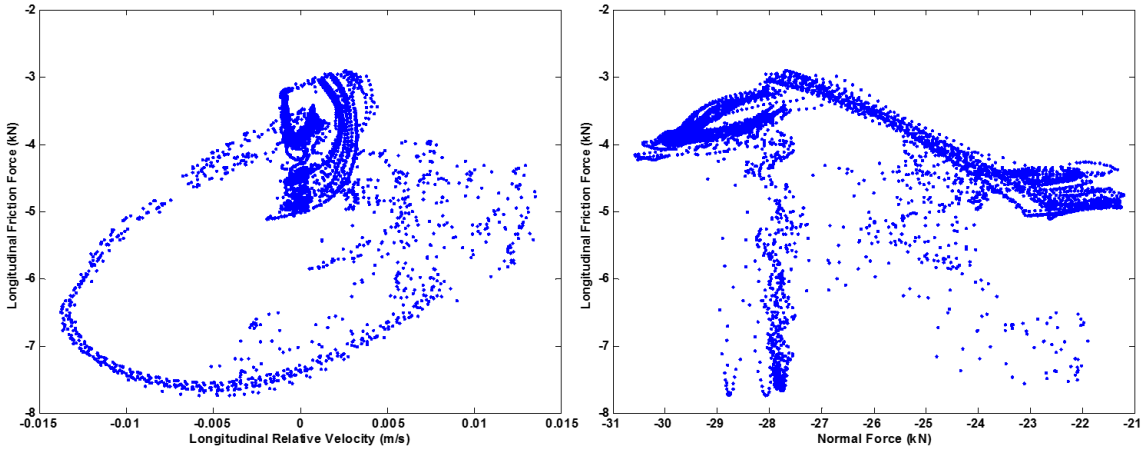

Figure 2.15: Longitudinal friction forces in the primary suspension

2.4 Secondary Suspension

The connection between the bolster and the side-frame is through the secondary suspension of the bogie. The three-piece truck's secondary suspension achieves previously mentioned suspension tasks by the compliance of the coil springs and dry frictional damping of the friction wedges; section 2.4.1 gives a detailed description of the friction wedges and the techniques used for modeling them. In section 2.4.2, different configurations for the coil springs are discussed.

2.4.1 Friction Wedges

Friction wedges were first introduced as a suspension element to the freight rail industry in 1935, and since then, they have been an irreplaceable part of the freight rail vehicle's suspension [35]. Friction wedges are the main source of energy dissipation in the suspension of a three-piece truck, and they play a decisive role in the dynamics of the truck. Popularity of friction wedges is due to their simple manufacturing, low cost, and low maintenance cost [35]; the wedge is spring loaded and located between the side-frame and the bolster. Relative bolster-wedge and side-frame-wedge motions causes friction on the contacting surfaces. Friction wedges are generally

categorized by their damping variability and toe direction [35]. Figure 2.16 shows various configurations for the friction wedge. In the constant damping configuration, the normal force on the surfaces of the wedge is constant and related to the control coil's stiffness and the pre-compression length of the springs; in the variable damping configuration, in addition to control coil stiffness, normal forces on the contacting surfaces are also influenced by the relative vertical distance of the bolster and the side-frame. The toe direction divides friction wedges into three different classes: no-toe, toe-in, and toe-out. They are distinguished by the inclination directions of column faces of the side-frames, as shown in Figure 2.16.

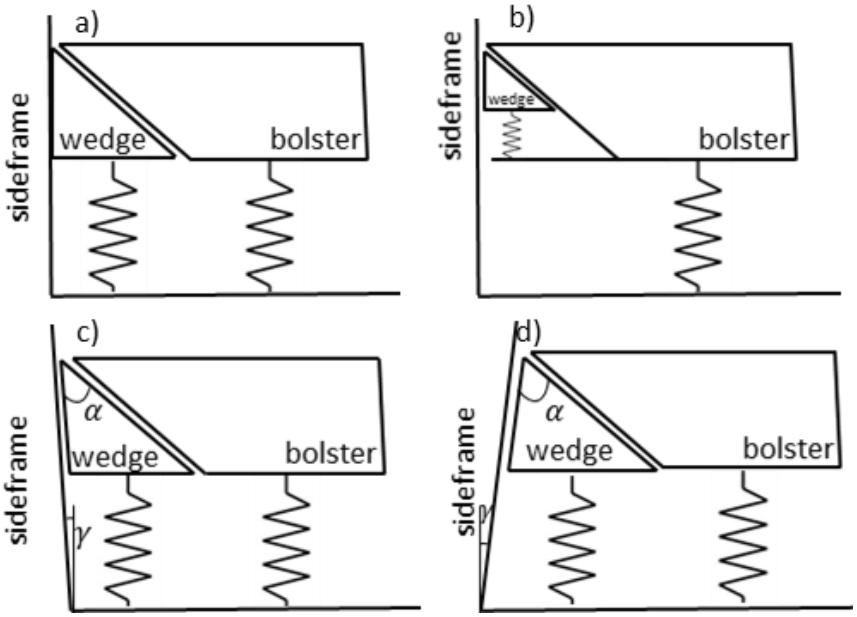

Figure 2.16: Different friction wedge configurations

The split-wedge is a variation of the friction wedge, which uses two wedges on each corner of the bolster. The geometric shape of the bolster has also been modified to have a V-shaped surface, which allows it to contact each split-wedge [20].

Figure 2.17: Comparison between a standard friction wedge and a split wedge [17]

There have been many studies dedicated to the dynamics of the friction wedge within the three-piece truck [7], [9], [15]–[17], [20], [36]–[38]. The problem arises from the complexities associated with modeling friction in general and the fact that the wedge is in contact with the bolster/side-frame on both sides. Peter Klauser introduces a simplified model for the friction wedge, but the model neglects the inertial properties of the wedge and models the wedge as a force transmission element [7]. In [9], only the vertical dynamics of the wedge are considered; the author assumes that the wedge is always in contact with the bolster and side-frame, and that the contacting bodies are always sliding relative to each other. Y. Q. Sun and C. Cole employ FE to model the friction wedge, and their results show significant difference between static modeling and FEM [36]. Vollebregt [26] provides a detailed review of various configurations and modeling techniques used for dynamic modeling of friction wedge suspensions.

Figure 2.18 shows how the friction force on the surface of the wedge varies with the normal force (top left), relative velocity between the wedge and the side-frame (top right), spring force generated by the control coil (bottom right), and the relationship between the control coils and the normal force for the friction element (bottom left). As can be seen, the wedge and consequently the three-piece truck exhibit highly nonlinear dynamics.

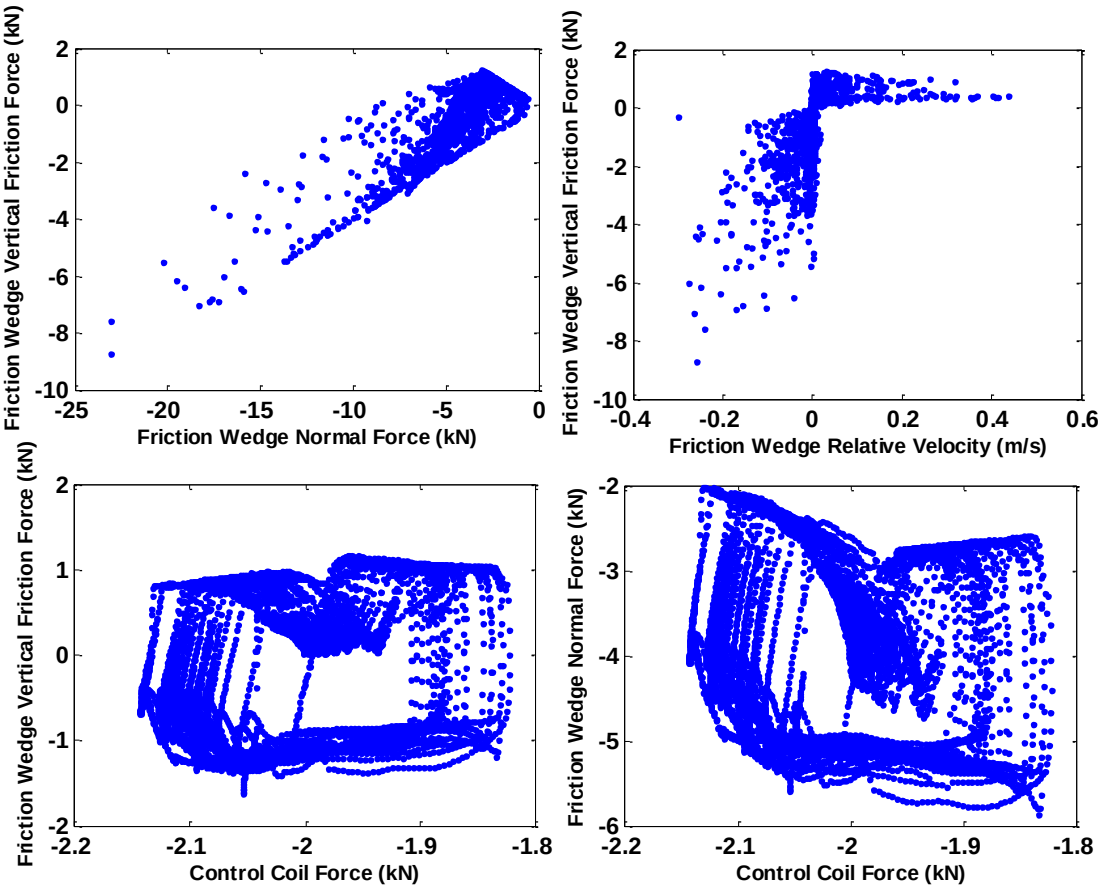

Figure 2.18: Vertical forces on the column side of the friction wedge

2.4.2 Nonlinear Springs

Similar to the friction wedges, coil springs of the secondary suspension have numerous different configurations. Different combinations used for the inner and outer coils provide a large number of suspension configurations that are selected by railroad companies based on the weight of the cargo, type of the product being carried, type of wagon, and numerous other parameters (Figure 2.19). The configuration marked by red in Figure 2.19 is the one used for the modeling of the three-piece truck used for the purpose of this study.

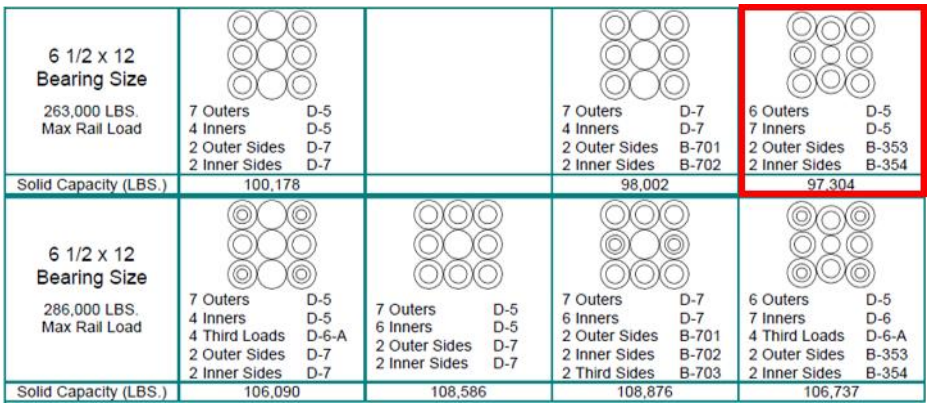

Figure 2.19: Different configurations of the secondary suspension's spring nest [39]

The bolster is seated on top of the coil springs so in case the relative distance between the bolster and the side-frame exceeds the nominal length of the spring, they are detached. To capture this phenomenon and the hard stops at the end of the coil spring's stroke, coil springs are modeled with a nonlinear spring. Figure 2.20 shows the relationship between the relative distance between the bolster and the side-frame and the resulting spring force. Instead of 13 linear springs that are shown in Figure 2.19 to simplify the model, each spring nest consists of 4 spring elements.

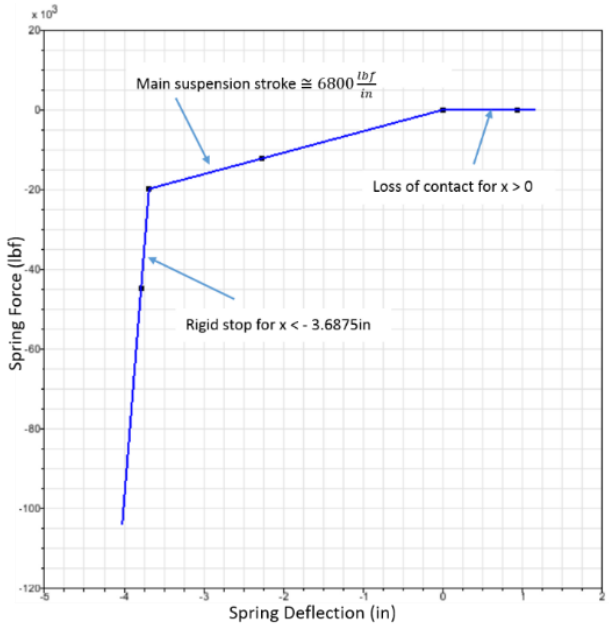

Figure 2.20: Displacement-force relation of the coil springs. Each spring nest consists of 4 spring elements

2.5 Connection between the Bolster and the Carbody

The connection between the bolster and the carbody is through a pivoted joint at the center plate assembly in the center of the bolster and the side-bearings on each side. This assembly plays an important role in the overall dynamics of the system on straight track and while negotiating curves. The main purpose of the assembly is to transfer the vertical loads from the carbody to the bolster, while allowing relative yaw motion between the two bodies; the yaw motion is resisted by friction. The magnitude of the turning resistance moment has a significant effect on the dynamics of the rail vehicle. Higher turning resistance improves the dynamics of the car on straight track, whereas lower turning resistance improves the performance of the truck while negotiating tight curves [40].

2.5.1 Center-Plate Center-Pin Assembly

The center-plate, center-pin assembly is designed to support most of the weight of the carbody. The assembly can be modeled using a spherical joint with friction, restricting the lateral and longitudinal DOFs; in reality the carbody can move in the lateral and longitudinal directions relative to the bolster due to the existence of clearances between the center-plate and center-pin. Here we've used frictional contact elements to model the vertical dynamics of the connection, and lateral and longitudinal frictional dead-band springs to model rim contact between the carbody and the bolster; Figure 2.21 shows the schematics of the connection.

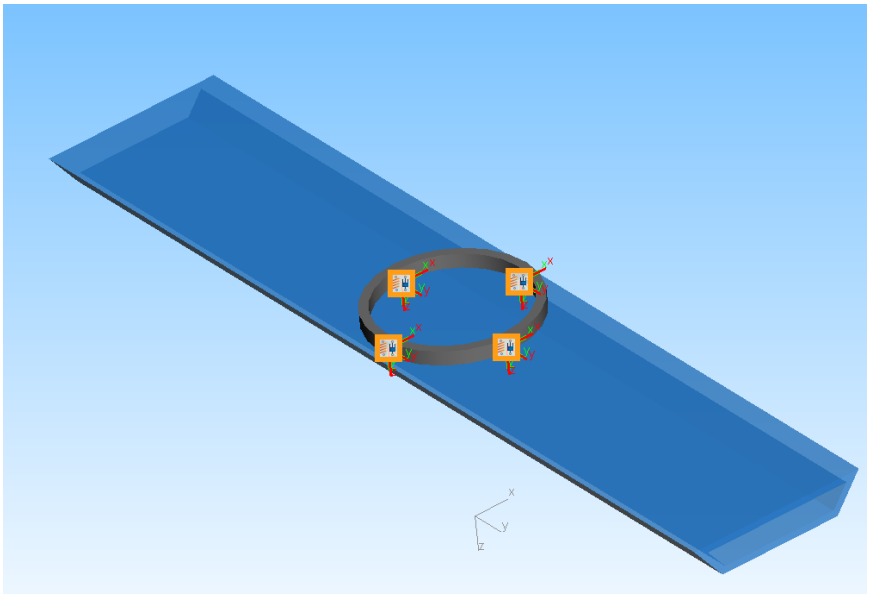

Figure 2.21: Schematics of the center-bowl connection

As mentioned earlier, the frictional turning resistance generated by the contact between the bolster and the carbody greatly influences the dynamics of the rail vehicle. Railway companies have been applying different techniques to modify the friction coefficient based on typical routes on which the car operates. For vehicles that operate on very curvy routes, consistent liquid lubrication to improve bogie steering is a common practice; on the other hand, for routes dominated by straight track, dry contact provides the best lateral stability and reduces hunting [40].

2.5.2 Side-bearings

Side-bearings provide additional support for the connection between the carbody and the bolster, especially during curve negotiation. There are two main types of side-bearings: 1) roller side-bearings (RSB), and 2) constant contact side-bearings (CCSB); Figure 2.22 shows the two configurations. Roller side bearings limit the roll motion of the carbody but allow for the relative yaw movement between the carbody and the bolster that consequently improves curve

negotiation of the truck. CCSB increases the turning resistance of the truck through preloaded rubber springs and friction pads that improve the truck's lateral stability on tangent track (hunting). Increased turning resistance, however, causes reduced curving performance of the car [41].

Figure 2.22: Roller side-bearing (left), and constant contact side-bearing (right)

Side-bearings have a much larger moment arm and if CCSBs are used, they can significantly increase the turning resistance of the truck. The total resistance moment depends on the portion of the total vehicle weight that is carried by the CCSB.

2.6 Model Validation

An important part of any multibody systems (MBS) modeling is to validate the model against experimental data. It is crucial that the behavior of the MBS model and the real vehicle be in good agreement. It is also important to consider that the MBS model is not an exact representation of reality because of the simplifying assumption used in the modeling process. Since [42] does not specify any quantitative limits for a successful rail vehicle model validation, the model described in the previous sections was validated against known characteristics of the three-piece truck, i.e. hunting velocity and frequency.

2.6.1 Hunting

Hunting is the self-excited lateral and yaw instability of the rail vehicle due to the conical profile of the wheel thread. At equilibrium, the two wheels on the axle have the same rolling radii. When the wheelset is displaced to the right due to the conical profile of the wheels, the right wheel will have a larger rolling radius; thus, it travels faster than the left wheel relative to the center of the axle. The difference between the longitudinal velocity of the left and right wheels causes the axle to yaw, which results in a larger rolling radius on the left wheel and, again, the differential velocity introduces a yaw displacement in the opposite direction. Figure 2.23 illustrates hunting of a single wheelset with conical wheels [43].

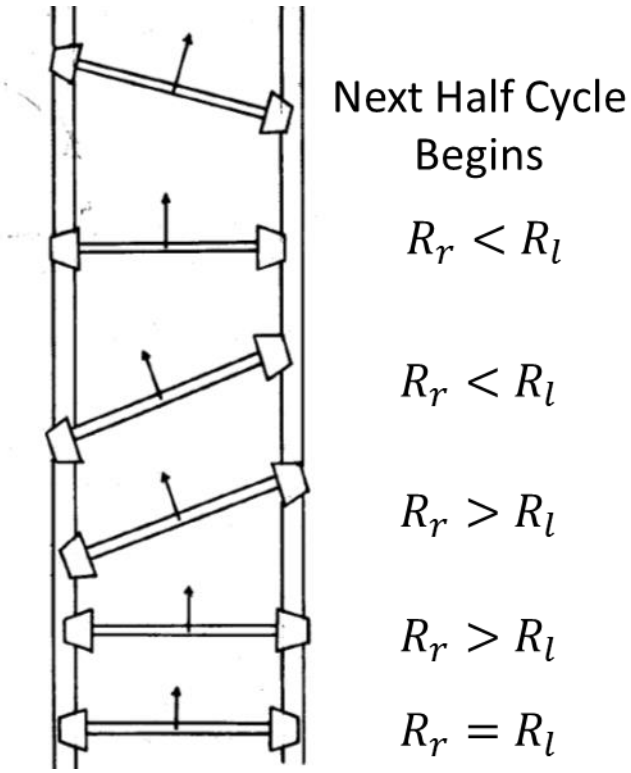

Figure 2.23: Schematics of a single wheelset hunting [43]

Wheel-rail creep forces are functions of the slippage that exists between the wheel and the rail. Creep forces consist of a stabilizing damping force and a negative restoring force. The negative restoring force is a function of displacement and tends to destabilize the system. On the other hand, the damping force stabilizes the system and it is inversely proportional to the forward velocity. At certain velocities, the stabilizing damping force becomes smaller than the destabilizing restoring force, and consequently, the system becomes unstable [43].

Other sources of damping that exist in the rail vehicle, such as the frictional damping of the secondary suspension or the friction forces that exists between the carbody and the bolster, contribute to dissipate the energy introduced to the system by the destabilizing restoring force. Therefore, hunting velocity for the rail vehicle is larger than the hunting velocity of a single wheelset. The frictional stabilizing damping force between the wheel and the rail is proportional to the weight of the rail vehicle; thus, hunting is a much bigger issue for empty rail cars. Empty rail vehicles are known to hunt around 55mph (88.5 km/h) [44].

Hunting is a nonlinear phenomenon, and it can be seen from Figure 2.24 that the predicted hunting velocity is higher when the velocity of the vehicle (V_B) increases. To capture the true hunting velocity of the rail vehicle (V_D), the initial velocity of the model was chosen as 144 km/h, and a constant braking force was applied to the carbody to gradually reduce the velocity of the carbody and observe the velocity where lateral acceleration would be damped; Figure 2.25 shows the results. An eigenvalue analysis of the linearized model was conducted to calculate the hunting frequency of the model, Figure 2.26 shows the results of the eigenvalue study.

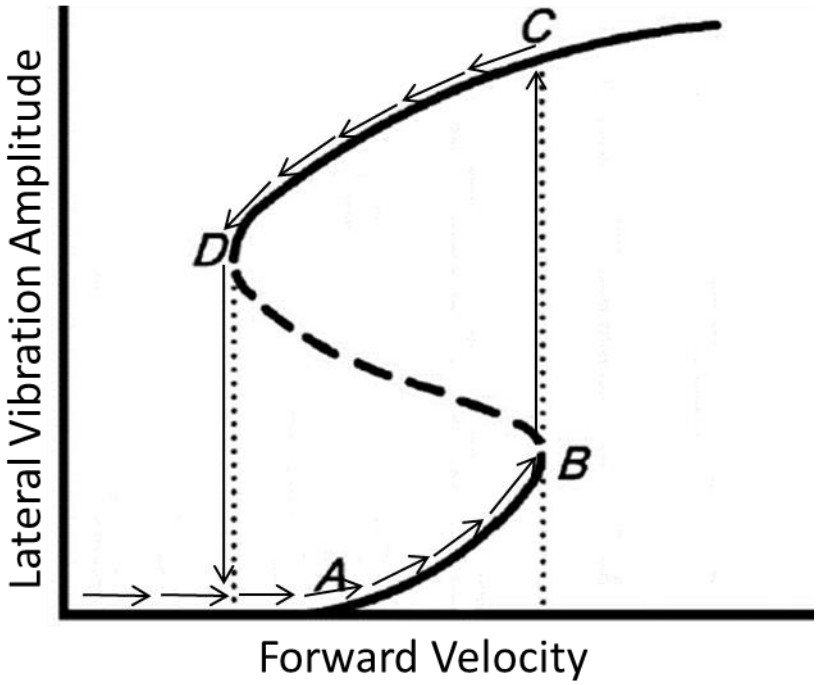

Figure 2.24: Nonlinear hunting velocity of a typical rail vehicle, where the dashed line represents the unstable region [19]

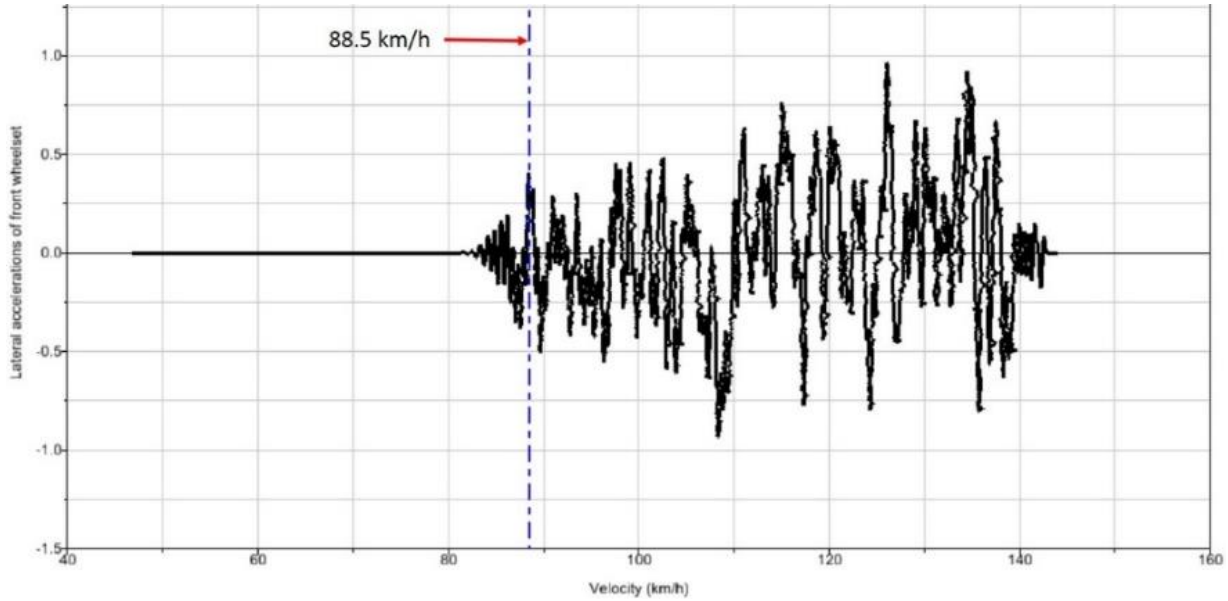

Figure 2.25: Lateral accelerations measured at the C.G of the front axle as a function of carbody longitudinal velocity

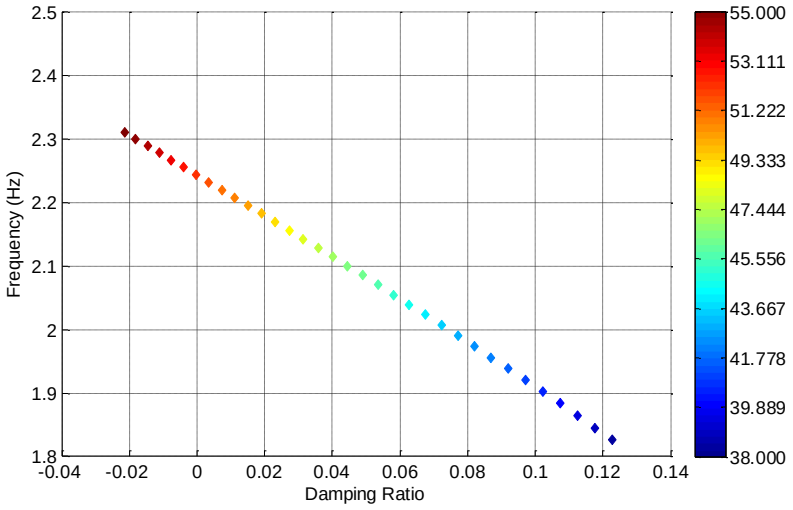

Figure 2.26: Hunting analysis using eigenvalues estimates hunting frequency of 2.2Hz

3 Machine Learning from Computer Simulations

3.1 Introduction

Our main purpose in this chapter is to build a stochastic model that estimates $y = f(\boldsymbol{x})$ where f is a black-box function or a computationally expensive code that converts the input vector $\boldsymbol{x}$ to the output y, i.e. critical speed of a rail vehicle (y) that varies based on the selected parameters for the suspension ($\boldsymbol{x}$). Derivation of the stochastic model requires some basic knowledge of statistics. We will start this chapter by reviewing these concepts and then use them to derive the stochastic model. Once the model is built, its accuracy will be tested on a number of example problems and we will investigate methods to improve the accuracy of the predictions.

The generic solution to the proposed problem is to perform the experiment or run the simulation for a set of input parameters $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}$ and collect the corresponding outputs $y^{(1)}, y^{(2)}, \dots, y^{(n)}$ and then find the best $\hat{f}(\boldsymbol{x})$ that estimates $f(\boldsymbol{x})$ based on the collected data. The technique was developed for the purpose of building cheap surrogates or meta-models for computationally expensive computer simulations. Global optimization methods like genetic algorithms can then be used to optimize a certain objective function, i.e. it can be used to find the set of suspension parameters for a three-piece truck that will result in the highest critical speed. With minor changes to the algorithm, the same method can also be used for parameter estimation, reducing simulation time, or accounting for stochastic processes within the model. The effectiveness of this method is based on the fact that computer simulations have the property of being exactly repeatable, which differentiates them from laboratory experiments. If

the inputs to the simulation algorithm remain the same, multiple runs will give the exact same results; unlike experiments, the output and the parameters are deterministic.

The process of constructing a surrogate can be broken down into four steps: 1) building a sampling plan and running the simulations, 2) choosing a base function for the surrogate and training the model with the data, 3) testing and validating the surrogate, and 4) improving the surrogate by sampling the original function at carefully selected additional points (infills). In the remainder of this chapter, we will discuss each step and then use the model to construct a surrogate for the Branin and the Hartmann's 3-dimensional functions to test the accuracy of the surrogate model. The performance of the stochastic model is also evaluated by using the model to predict system parameters for a single suspension model.

3.2 Review of Basic Statistic Concepts

Statistics, the science of uncertainty, attempts to find order in disorder [45]. In this section, a brief review of statistical concepts required for building the stochastic model is provided. We will start with classical statistics and continue the discussion into Bayesian statistics.

3.2.1 Random Variables and Probability Density Functions

The concept of random variables is the most important concept in studying statistics. The simplest and most intuitive way to define a random variable is to define it as a variable that takes its value by chance. The convention is to use capital letters $X, Y, and\ Z$ for random variables, and lowercase letters ($x, y, and\ z$) for deterministic variables. Random variables are governed by their probability distribution functions (Equation (3.1)) [46].

$$F(x) = Pr\{X \leq x\} \quad -\infty < x < \infty \tag{3.1}$$

where $Pr\{X \leq x\}$ is the probability that the random variable X takes on a value less than or equal to x. Depending on the distribution function associated with X and the value of x, $0 \leq \Pr\{X \leq x\} \leq 1$. Since the probability distribution function is used ambiguously in the literature, we will define the probability density function (PDF) for a continuous random variable X as:

$$\Pr\{\mathrm{a} < X \leq \mathrm{b}\} = \int_a^b f(x)\,dx \qquad for -\infty < a < b < \infty \tag{3.2}$$

where $f(x)$ is the PDF for X. The relationship between PDF and the probability distribution function is:

$$F(x) = \int_{-\infty}^{x} f(\xi)d\xi \qquad -\infty < x < \infty \tag{3.3}$$

The probability density function contains all the information available about a random variable before its value is determined by experiment [46]. We will further clarify the concept with an example: suppose we've collected data on the height h in inches of adult males in the U.S. and created the PDF shown in Figure 3.1. Until we measure their height, the height of any other adult male in the U.S. is a random variable (H) governed by the probability distribution function shown. Before the experiment, we can say that the probability of H being close to 70" is very high or we can make the statement that 95% of times, height of a randomly selected adult male in the U.S. falls between 60 and 80 inches or $\Pr\{60 < H < 80\} = 0.95$.

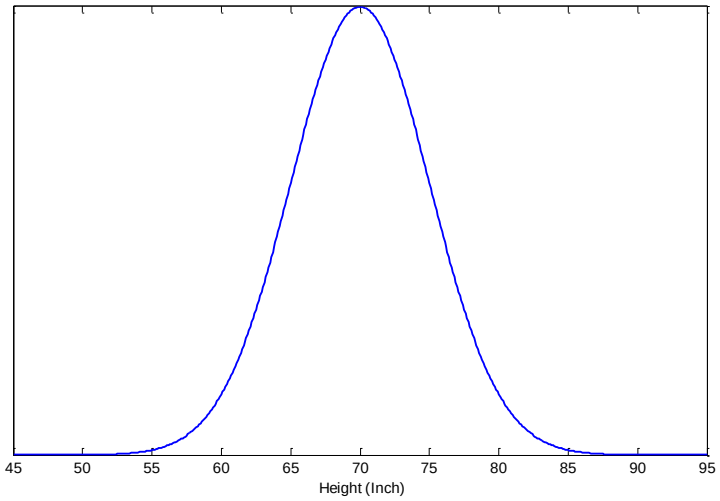

Figure 3.1: Probability density function for the height of adult males in the U.S

The distribution function shown in Figure 3.1 is known as the normal distribution; central limit theorem states that the distribution of the sum (or average) of a large number of independent, identically distributed (iid) variables can be approximated with a normal distribution, regardless of the underlying distribution, which makes the normal distribution the most important probability distribution. Normal distribution is governed by two parameters, μ, σ^2, and the equation for a normal distribution is:

$$\phi(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad -\infty < x < \infty \tag{3.4}$$

where μ is the mean or the expected value and σ^2 is the variance. Standard normal distribution is defined as a normal distribution with $\mu = 0, \sigma^2 = 1$. Figure 3.2 show the influence of μ, σ^2 on the shape of the normal distribution.

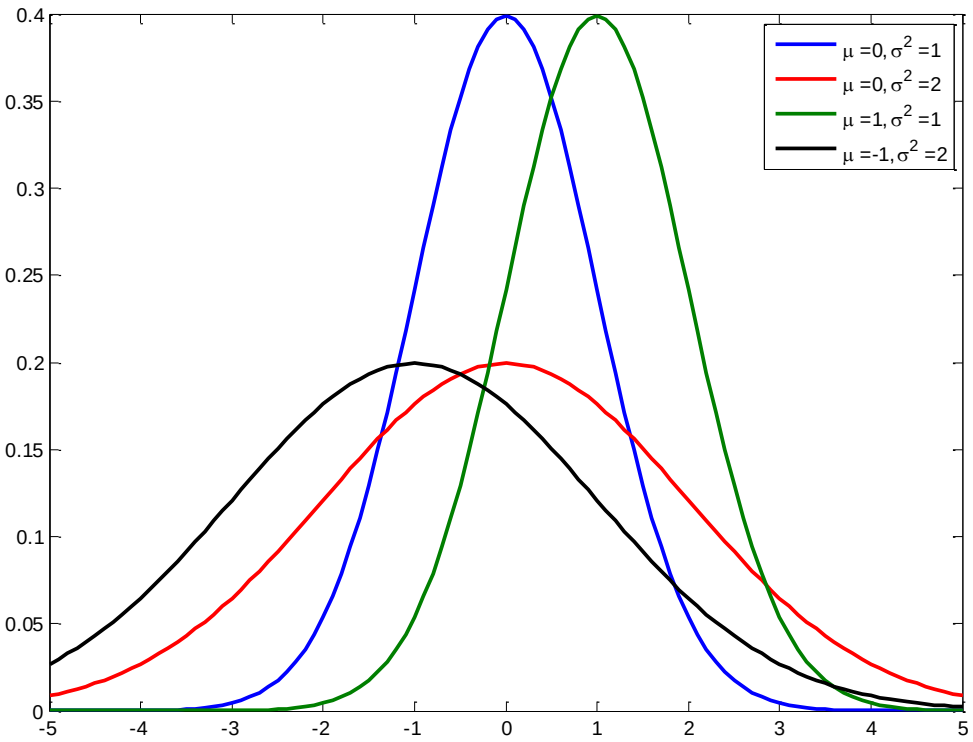

Figure 3.2: Influence of μ, σ^2 on a normal distribution

3.2.2 Expected Value and Variance

Expected value ($E[X]$) and variance ($Var[X]$) are important properties or moments of a random variable. As the name suggests, expected value is the value that we expect the random variable to take; it is also called the mean value for X. Variance is the measure of uncertainty in the data; higher variance means that the probability for the random variable to take a value far from the expected value is higher. If the probability density function is defined by $f(x)$, we have:

$$E[X] = \mu = \int_{-\infty}^{\infty} xf(x)dx \tag{3.5}$$

$$Var[X] = E[(X-\mu)^2] = E[X^2] - \mu^2 \tag{3.6}$$

3.2.3 Joint Distribution Functions and Covariance

Joint distribution functions are defined when working with pairs of random variables (X, Y). If x, y are two real variables, then their joint distribution function (F_{xy}) is given by

$$F_{XY}(x, y) = F(x, y) = \Pr\{X \leq x \; and \; Y \leq y\} \tag{3.7}$$

Similar to the relationship between distribution function and density function of a single random variable, f_{XY} is said to be the joint probability distribution function for random variables X, Y if:

$$F_{XY}(x, y) = \int_{-\infty}^{x} \int_{-\infty}^{y} f_{XY}(\xi, \eta) d\eta d\xi \quad for \; all \; x, y \tag{3.8}$$

Random variables X, Y are independent if their joint probability distribution function is the multiplication of each of their probability distribution functions; in other words: $F_{XY}(x, y) = F_X(x) \times F_Y(y)$. The same concept applies to their joint density function ($f_{XY}(x, y) = f_X(x) f_Y(y)$ for all x, y).

Covariance is a measure of how independent or dependent two random variables are. If X, Y are two jointly distributed random variables with finite variance and μ_x, μ_y as their means, respectively, the covariance of X, Y (σ_{XY} or $Cov[X, Y]$) is given by:

$$\sigma_{XY} = E\big[(X - \mu_x)\big(Y - \mu_y\big)\big] = E[XY] - \mu_x \mu_y \tag{3.9}$$

If the covariance of two random variables is zero, they are uncorrelated; independent random variables with finite variance are uncorrelated, but uncorrelated random variables are not

necessarily independent. Correlation is the normalized covariance given by dividing the covariance by the product of the two standard deviations σ_x, σ_y.

$$Corr[X,Y] = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \quad and \quad -1 \leq Corr[X,Y] \leq 1 \tag{3.10}$$

3.2.4 Maximum Likelihood Estimations

Maximum likelihood estimations (MLE) is one of the most widely used approaches to estimation in all of statistical inferences. MLE was first introduced by R. A. Fisher, a geneticist and statistician, in the 1920s [47]. Since the maximum likelihood estimation maximizes the likelihood function, first we need to define the likelihood function $L(x_1, x_2, \dots, x_n; \theta)$. Suppose $X_1, X_2, \dots, X_n$ are independent random variables taken from the probability distribution function represented by $f(x, \theta)$; the likelihood function is the joint probability distribution of the random variables.

$$L(x_1, x_2, \dots, x_n; \theta) = f(x_1, x_2, \dots, x_n; \theta) = f(x_1; \theta) f(x_2; \theta) \dots . f(x_n; \theta) \tag{3.11}$$

where θ is the function variable. To find θ that maximizes L, we need to take the derivative of L with respect to θ and set it equal to zero and solve for $\hat{\theta}$. Since $ln[g(u)]$ is a strictly increasing function of $g(u)$, finding u that maximizes $ln[g(u)]$ is equivalent to finding u that maximizes $g(u)$. Because it is mathematically more convenient to work $ln[g(u)]$, we use the ln-likelihood function for MLE. To better illustrate the concept, we will derive the MLE for a normal distribution with mean μ and variance σ^2.

$$L(x_1, x_2, \dots, x_n; \boldsymbol{\theta}) = f(x_1, x_2, \dots, x_n; \mu, \sigma^2) = \prod_{i=1}^{n} f(x_i; \mu, \sigma^2)$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} = (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2} \tag{3.12}$$

$$\rightarrow \ln\big(L(\boldsymbol{x}; \boldsymbol{\theta})\big) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

Now we need to take partial derivatives with respect to μ, σ^2, set them equal to zero, and solve the resulting equations simultaneously.

$$\frac{\partial}{\partial\mu}\ln\big(L(\boldsymbol{x}; \mu, \sigma^2)\big) = 0 \rightarrow \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i - n\mu = 0 \rightarrow \sum_{i=1}^{n} x_i - n\mu = 0 \rightarrow$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.13}$$

$$\frac{\partial}{\partial\sigma^2}\ln\big(L(\boldsymbol{x}; \mu, \sigma^2)\big) = 0 \rightarrow \frac{1}{2\sigma^2}\left[\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2 - n\right] = 0 \rightarrow$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2 \tag{3.14}$$

Results are intuitive: the MLE for $\hat{\mu}$ is the sample mean, and the estimation for $\hat{\sigma}^2$ is the sample variance. The same procedure can be used to find MLE for other probability distribution functions. Maximum likelihood estimations are not always as straight forward as illustrated above; in some cases (i.e. Weibull Distribution) where equations resulting from partial derivatives cannot be solved explicitly or taking derivatives of the likelihood function is not feasible, global

optimization methods (i.e. Markov-Chain Monte Carlo) are used to find the optimum parameter values using iterative procedures [45].

Joint probability distribution of dependent variables are more complicated to derive, but for n normally distributed dependent random variables $(x_1, x_2, \cdots, x_n)$ with correlation matrix $\boldsymbol{\Psi}$, the joint probability distribution function is defined as:

$$f(x_1, x_2, \cdots, x_n) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{\det(\boldsymbol{\Psi})}} \exp\left(-\frac{1}{2}[x_1 - \mu_1, \cdots, x_n - \mu_n]\boldsymbol{\Psi}^{-1}[x_1 - \mu_1, \cdots, x_n - \mu_n]^T\right) \quad (3.15)$$

where $(\mu_1, \mu_2, \cdots, \mu_n)$ is the mean value of each variable. In this case, since the likelihood function depends on the parameters of the correlation matrix ($\boldsymbol{\Psi}$), taking partial derivatives is not sufficient for finding the maximum of Equation (3.15), and global optimization techniques are used to find its maximum value.

3.3 Constructing a Sampling Plan

The principal idea behind estimating a function using its surrogate is that the original function is computationally expensive and we cannot afford to run a huge number of simulations; if this was not the case, there was no need for a surrogate to begin with and the original function could have been used to find the global optimum. Since simulations are expensive, we have to develop a sampling plan that represents our design space in the most efficient manner; in other words, we have to make each simulation count.

Our purpose in this section is to find a sampling plan $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\right\}^T, \boldsymbol{x} \in D \subset \Re^k$ where k is the number of variables and D is the design space, such that $\boldsymbol{X}$ is the most space-filling plan among similar plans with n sample points. Space-filling Latin hypercubes [48] are the most popular method of building a sampling plan, but since they do not give unique answers, we use a "space-fillingness" definition to select the best Latin hypercube among similar plans. This method guarantees that each $\boldsymbol{x}^{(i)}$ is represented in a fully stratified manner.

The maximin metric introduced by [49], [50] is the most widely used measure of "space-fillingness" of a sampling plan. Since the approach of [50] provides a more complete and unique sampling plan, we have limited ourselves to the definition of a "maximin" sampling plan introduced by them. The most space-filling sampling plan is the one that minimizes:

$$\Phi_{\mathrm{q}}(\boldsymbol{X}) = \left(\sum_{j=1}^{m} J_j d_j^{-q}\right)^{1/q} \tag{3.16}$$

where d_j is the list of the unique values of inter-site distances between all possible pairs of $\boldsymbol{X}$ obtained using Equation (3.17) in ascending order; J_j is the number of pairs of points in the sampling plan that are separated by the distance d_j. The value used for q is problem dependent but [50] suggests values as big as 20 or 50 for problems with a large number of sample points or variables.

$$d\left(x^{(i_1)}, x^{(i_2)}\right) = \left(\sum_{j=1}^{k} \left|x_j^{(i_1)} - x_j^{(i_2)}\right|\right) \tag{3.17}$$

The problem is now reduced to finding a sampling plan that minimizes Φ_q; global optimization methods like genetic algorithms can be used to find such a plan. But since we have limited computational budget, and in some cases finding the global optimum of Φ_q is computationally expensive, we have to manage the amount of time that we spend and select a sampling that minimizes Φ_q within our budget. Figure 3.3 shows an example of a sampling plan with $n = 9, k = 2$.

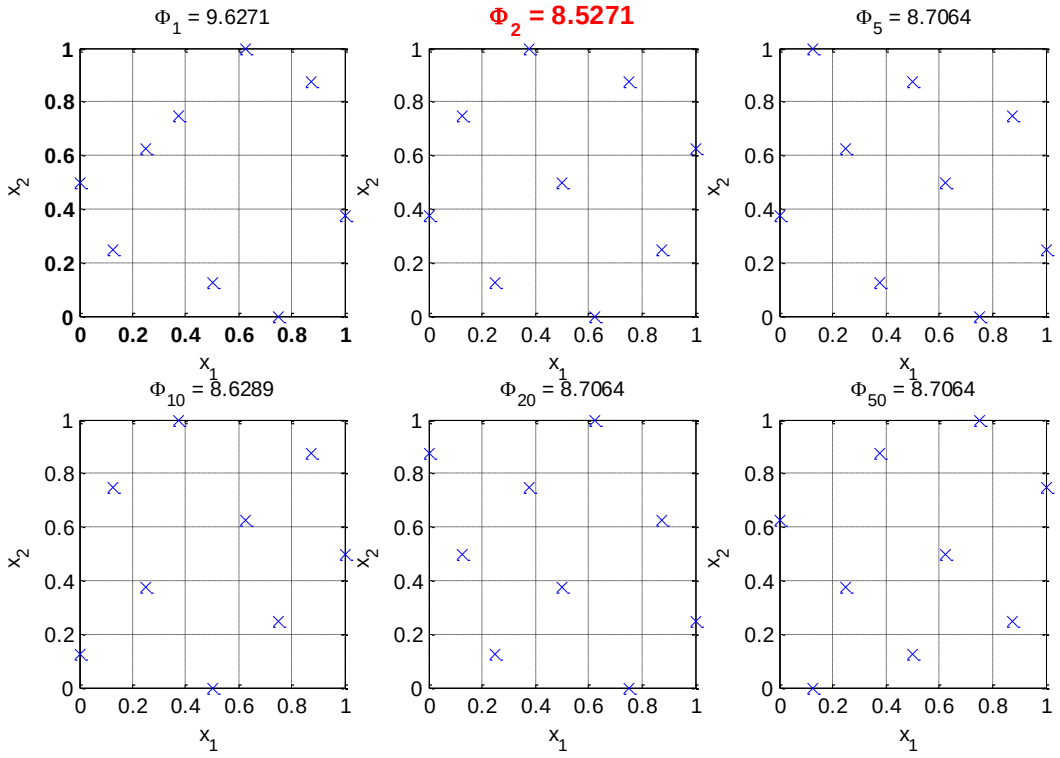

Figure 3.3: Example of a Latin Hypercube sampling with k = 2, n = 9. Φ_2 is the plan with the "best space-fillingness"

It can be seen in Figure 3.3 that the plan with the lowest Φ is the most space-filling, and the sampled data are distributed evenly across the design space.

3.4 Approximation of Deterministic Functions with Stochastic Processes (Kriging, Random Function Approach)

The concept of estimating deterministic functions using stochastic processes has been around for more than half a century. The concept was first introduced in mathematical geology literature by [51] and later developed by [52] as a means to estimate the concentration of valuable minerals based on core sample testing performed at several locations. The approach is widely used in the context of global optimization as a method of developing figures of merit [53]. An advantage associated with the use of stochastic processes is that they provide confidence intervals for the predictions at un-sampled points that can be used to improve the accuracy of the stochastic model by sampling where the predicted error is high; this also has the advantage of providing a stopping criteria for the algorithm.

Computer algorithms generate output that is far more deterministic than laboratory experiments, in which the results are affected by the test environment in addition to the dynamics of the system. A computer algorithm (i.e. solving the equations of motion) can be thought of as a mapping between the input ($\boldsymbol{x}$) and the output $y(\boldsymbol{x})$ vectors. Since y is an expensive function and is not suitable for global optimization routines where a large number of simulations are required for finding the optimum, we are trying to estimate $y(\boldsymbol{x})$ with $\hat{y}(\boldsymbol{x})$ that is cheaper to evaluate. We assume that we have a set of sample data $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}\right\}^T$ with the corresponding set of observed outputs, $\boldsymbol{y} = \left\{y^{(1)}, y^{(2)}, \dots, y^{(n)}\right\}^T$ and use a linear regression line for the predictor function.

Each of the sampled data in $\boldsymbol{X}$ and $\boldsymbol{y}$ provide some information regarding the underlying behavior of the system being studied. As will be shown later in this section, the correlation function and the likelihood function are used to extract as much information as possible from the sample data to minimize our uncertainty regarding the underlying behavior.

$$\hat{y}\left(x^{(i)}\right) = \sum_{h=1}^{k} \beta_h \, x_h^{(i)} + \epsilon^{(i)} \qquad (i = 1,2, \dots, n) \tag{3.18}$$

Computer algorithms are deterministic, making the assumption of independent error terms ($\epsilon^{(i)}$) completely false. Due to the fact that engineering functions are usually smooth and continuous, estimation error ϵ is related or correlated with the distance between input vectors $\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}$.

$$\hat{y}\left(x^{(i)}\right) = \sum_{h=1}^{k} \beta_h \, x_h^{(i)} + \epsilon(\boldsymbol{x}^{(i)}) \qquad (i = 1,2, \dots, n) \tag{3.19}$$

We use a special weighted distance shown below to find correlation between error terms:

$$d\left(x^{(i)}, x^{(j)}\right) = \sum_{h=1}^{k} \theta_h \left| \boldsymbol{x}_h^{(i)} - \boldsymbol{x}_h^{(j)} \right|^{p_h} \tag{3.20}$$

where θ_h, p_h are parameters that define the shape of the correlation function and are estimated using evolutionary algorithms such as the simulated annealing, or the genetic algorithm. Correlation between error terms is given by:

$$Corr[\epsilon\left(\boldsymbol{x}^{(i)}\right), \epsilon\left(\boldsymbol{x}^{(j)}\right)] = \exp\left(-d\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)\right) \tag{3.21}$$

For $\theta_h = c$ and $p_h = 2$ where c is a constant, the correlation function reduces to a Gaussian function.

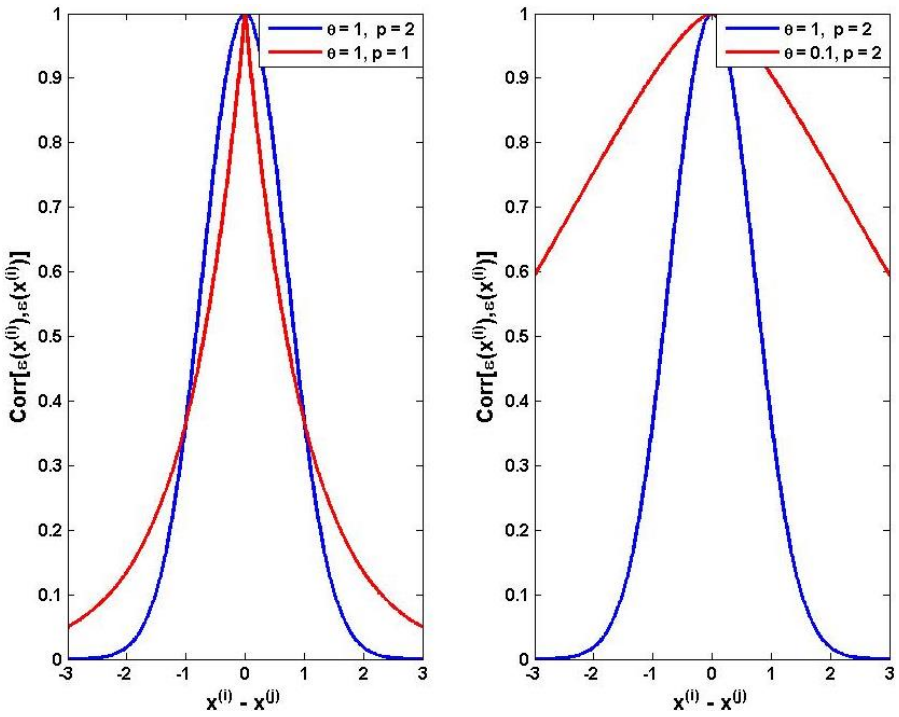

Figure 3.4: Correlation function for various values of θ_h and p_h

The correlation function is intuitive in that if the two points $x^{(i)}$ and $x^{(j)}$ are close to each other they have a higher correlation and as the points move away from each other the correlation reduces. It can be seen from Figure 3.4 that the parameter p affects the "smoothness" of the correlation function; for p = 2, the correlation function has a continuous gradient and as p reduces the correlation initially drops with a larger gradient as the absolute distance between points increases. The influence of $\theta_j's$ are shown in Figure 3.4; a lower θ corresponds to less influential variables, meaning that all points within the sample space have a high correlation. Conversely, larger values of θ indicate that the function is more sensitive to any changes in corresponding variable(s).

Modeling the error terms of Equation (3.19) using the stochastic process modeling technique described is so powerful that we can replace the first term ($\sum_{h=1}^{k} \beta_h \, x_h^{(i)}$) with a constant $\hat{\boldsymbol{\mu}}$, reducing Equation (3.19) to:

$$\hat{y}\left(x^{(i)}\right) = \hat{\boldsymbol{\mu}} + \epsilon(\boldsymbol{x}^{(i)}) \qquad (i = 1,2,\dots,n) \tag{3.22}$$

where $\hat{\boldsymbol{\mu}}$ is the mean of the stochastic process ($n \times 1$ column vector) and $\epsilon(\boldsymbol{x}^{(i)})$ is a normally distributed stochastic process with a mean of 0 and a standard deviation of σ^2. We use the concept of maximum likelihood estimation described in section 3.2.4 to estimate $\hat{\boldsymbol{\mu}}, \sigma^2$ that maximizes the likelihood of the sample data. The likelihood function of a normal distribution can be expressed in terms of outputs (y) and the correlation matrix ($\boldsymbol{\Psi}$) as:

$$L = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}|\boldsymbol{\Psi}|^{1/2}} \exp\left[-\frac{(y-\boldsymbol{\mu})^T\boldsymbol{\Psi}^{-1}(y-\boldsymbol{\mu})}{2\sigma^2}\right] \tag{3.23}$$

where $\boldsymbol{\Psi}$ is given by

$$\boldsymbol{\Psi} = \begin{bmatrix} corr\left[\epsilon\left(\boldsymbol{x}^{(1)}\right), \epsilon\left(\boldsymbol{x}^{(1)}\right)\right] & \cdots & corr\left[\epsilon\left(\boldsymbol{x}^{(1)}\right), \epsilon\left(\boldsymbol{x}^{(n)}\right)\right] \\ \vdots & \ddots & \vdots \\ corr\left[\epsilon\left(\boldsymbol{x}^{(n)}\right), \epsilon\left(\boldsymbol{x}^{(1)}\right)\right] & \cdots & corr\left[\epsilon\left(\boldsymbol{x}^{(n)}\right), \epsilon\left(\boldsymbol{x}^{(n)}\right)\right] \end{bmatrix} \tag{3.24}$$

Since it is more convenient to work with the natural logarithm of the likelihood function, we take the natural logarithm of both sides of Equation (3.23):

$$\ln(L) = \ln\left(\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}|\boldsymbol{\Psi}|^{\frac{1}{2}}} exp\left[-\frac{(y-\boldsymbol{\mu})^T\boldsymbol{\Psi}^{-1}(y-\boldsymbol{\mu})}{2\sigma^2}\right]\right)$$

$$\rightarrow \ln(L) = \ln\left(\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}|\boldsymbol{\Psi}|^{\frac{1}{2}}}\right) + \ln\left(exp\left[-\frac{(y-\boldsymbol{\mu})^T\boldsymbol{\Psi}^{-1}(y-\boldsymbol{\mu})}{2\sigma^2}\right]\right) \tag{3.25}$$

$$\rightarrow \ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|\boldsymbol{\Psi}|) - \frac{(y-\boldsymbol{\mu})^T\boldsymbol{\Psi}^{-1}(y-\boldsymbol{\mu})}{2\sigma^2}$$

We find the mean and the standard deviation that maximize Equation (3.25) by taking partial derivatives with respect to $\boldsymbol{\mu}$ and σ^2 and setting them equal to zero:

$$\frac{\partial(\ln(L))}{\partial \boldsymbol{\mu}} = -\mathbf{1}^T \boldsymbol{\Psi}^{-1} y + \mathbf{1}^T \boldsymbol{\Psi}^{-1} \mathbf{1} = 0 \rightarrow$$
$$\hat{\boldsymbol{\mu}} = \frac{\mathbf{1}^T \boldsymbol{\Psi}^{-1} y}{\mathbf{1}^T \boldsymbol{\Psi}^{-1} \mathbf{1}} \tag{3.26}$$

$$\frac{\partial(\ln(L))}{\partial \sigma^2} = \frac{1}{2\sigma^2}\left[\frac{1}{\sigma^2}(y-\boldsymbol{\mu})^T \boldsymbol{\Psi}^{-1}(y-\boldsymbol{\mu}) - n\right] = 0 \rightarrow$$
$$\hat{\sigma}^2 = \frac{(y-\boldsymbol{\mu})^T \boldsymbol{\Psi}^{-1}(y-\boldsymbol{\mu})}{n} \tag{3.27}$$

where $\mathbf{1}$ is a $n \times 1$ column vector of ones. Substituting these parameters into the original ln-likelihood function (Equation (3.25)) and eliminating the constant terms gives us the "concentrated ln-likelihood function"

$$\ln(L) \approx -\frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln(|\boldsymbol{\Psi}|) \tag{3.28}$$

Equation (3.28) solely depends on the parameters of the correlation function (θ, p). Since we cannot take derivatives of Equation (3.23) with respect to these parameters, we need to use numerical optimization techniques to find the values of θ, p that correspond to the global optimum of the concentrated ln-likelihood function.

To estimate the value of the function at a new point $y^* = f(\boldsymbol{x}^*)$, we assume that the set $(\boldsymbol{x}^*, y^*)$ belongs to the same dataset as the previously observed data $(\boldsymbol{X}, y)$ and create an augmented

dataset. In the previous section, we used the observed data and estimated the parameters that maximize the ln-likelihood function $(\hat{\boldsymbol{\mu}}, \hat{\sigma}^2, \theta, p)$. This process is what has been referred to as the learning procedure. In this section, we use these fixed parameters and maximize the ln-likelihood of the augmented dataset. With all the parameters fixed, the new ln-likelihood function only depends on y^*.

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln\left(|\tilde{\boldsymbol{\Psi}}|\right) - \frac{(\tilde{y} - \hat{\boldsymbol{\mu}})^T \tilde{\boldsymbol{\Psi}}^{-1} (\tilde{y} - \hat{\boldsymbol{\mu}})}{2\hat{\sigma}^2} \tag{3.29}$$

where $\tilde{y} = [y^T \; y^*]^T$ is the augmented vector of function values and $\tilde{\boldsymbol{\Psi}}$ is the augmented correlation matrix.

$$\tilde{\boldsymbol{\Psi}} = \begin{bmatrix} \boldsymbol{\Psi} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^T & 1 \end{bmatrix} \tag{3.30}$$

$$\boldsymbol{\psi} = \begin{bmatrix} Corr\left[\epsilon\left(x^{(1)}\right), \epsilon(x^*)\right] \\ \vdots \\ Corr\left[\epsilon\left(x^{(n)}\right), \epsilon(x^*)\right] \end{bmatrix} \tag{3.31}$$

Since only the last term in Equation (3.29) depends on y^*, we eliminate the rest of the terms and expand Equation (3.29) using the partitioned inverse method (Equation (3.33)) [54].

$$\ln(L) \approx \frac{-\begin{bmatrix} (y - \hat{\boldsymbol{\mu}}) \\ (y^* - \mu) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Psi} & \boldsymbol{\psi}^T \\ \boldsymbol{\psi}^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} (y - \hat{\boldsymbol{\mu}}) \\ (y^* - \mu) \end{bmatrix}}{2\hat{\sigma}^2} \tag{3.32}$$

$$\begin{bmatrix} \boldsymbol{\Psi} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{\Psi}^{-1} + \boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(1 - \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1}\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1} & -\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(1 - \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1} \\ -(1 - \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1}\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1} & (1 - \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})^{-1} \end{bmatrix} \tag{3.33}$$

Substituting Equation (3.33) into Equation (3.32) and eliminating the terms without y^*, the ln-likelihood function simplifies to:

$$\ln(L) = \left(-\frac{1}{2\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(y^*-\hat{\mu})^2 + \left(\frac{\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}(y-\hat{\mu})}{\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(y^*-\hat{\mu}) \quad (3.34)$$

The only variable in Equation (3.34) is the function estimation at the new point y^*; to maximize the likelihood function, we take its derivative with respect to y^* and set it equal to zero:

$$\frac{d(\ln(L))}{dy^*} = \left(-\frac{1}{\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right)(y^*-\hat{\mu}) + \left(\frac{\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}(y-\hat{\mu})}{\hat{\sigma}^2(1-\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi})}\right) = 0 \quad (3.35)$$

Equation (3.35) estimates the value for y^* to be

$$\hat{y}(x) = \hat{\mu} + \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}(y-\hat{\mu}) \quad (3.36)$$

Equation (3.36) is the function that approximates y. The model is built such that the prediction passes through all the sample data (interpolator). If $\boldsymbol{x} = \boldsymbol{x}^{(i)}$, $\boldsymbol{\psi}$ is the i^{th} column of $\boldsymbol{\Psi}$, which means $\boldsymbol{\psi}\boldsymbol{\Psi}^{-1} = \mathbf{1}$; using Equation (3.36), the predicted value is $\hat{y}(\boldsymbol{x}) = \hat{\boldsymbol{\mu}} + y^{(i)} - \hat{\boldsymbol{\mu}} = y^{(i)}$.

One of the advantages of using the stochastic process model is that it allows for the calculation of the estimated error at untried locations. We can use this feature to improve the predictor by sampling at locations where the estimated error is the highest.

$$\hat{s}^2(\boldsymbol{x}) = \sigma^2\left[1 - \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi} + \frac{1-\mathbf{1}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}}{\mathbf{1}^T\boldsymbol{\Psi}^{-1}\mathbf{1}}\right] \quad (3.37)$$

To illustrate the functionality of the developed algorithm, in Appendix A we will use the stochastic model to find the global optimum of some example problems.

3.4.1 Stochastic Prediction with Noisy Data

As mentioned earlier, the stochastic predictor in section 3.4 is an interpolator, it passes through all the sample data, which is an accurate assumption for deterministic computer codes. In this section, we will discuss the necessary changes required when dealing with noisy sample data. If we assume that the variance of the noise is τ^2 and we reconstruct the correlation matrix by adding τ^2 to the main diagonal of $\boldsymbol{\Psi}$, then the new correlation matrix becomes:

$$\tilde{\boldsymbol{\Psi}} = \boldsymbol{\Psi} + \tau^2 \mathbf{I} \tag{3.38}$$

where $\mathbf{I}$ is an $n \times n$ identity matrix. Now $\boldsymbol{\psi}$ is never a column of $\boldsymbol{\Psi}$ and the predicted value at a sample data is not equal to the observed data [14]. In the previous section, we used Equation (3.27) to calculate $\hat{\sigma}^2$ that minimizes the ln-likelihood function; the assumption of noisy data prevents us from estimating the value of $\hat{\sigma}^2$ using partial derivatives and we have to add it to the list of parameters that needs to be estimated by the global optimization routine. Using the same process as for the noise-free case, we can derive the predictor function as:

$$\hat{y}(x) = \hat{\mu}_r + \boldsymbol{\psi}\tilde{\boldsymbol{\Psi}}^{-1}(y - 1\hat{\mu}_r) \tag{3.39}$$

where

$$\hat{\mu}_r = \frac{1^T \tilde{\boldsymbol{\Psi}}^{-1} y}{1^T \tilde{\boldsymbol{\Psi}}^{-1} 1} \tag{3.40}$$

3.5 Parameter Estimation Using the Stochastic Predictor

Now that we have tested the efficiency and the accuracy of the stochastic predictor, in this section we will test the algorithm on a simple engineering application. The purpose of this test is to estimate the suspension parameters of a linear single suspension model. Figure 3.5 shows the schematics of the single suspension model; in the first example, we are trying to estimate k_s, c_s, and for the second example, the number of parameters to be estimated are increased to four (k_s, c_s, m_s, m_{us}). As the number of the parameters is increased, we need more training data to achieve the same level of prediction accuracy. We will run the model once with the parameters shown in Table 3.1 and save the data for later comparisons; we will call this data the 'simulated data'. We will run additional simulations using Latin Hypercube sampling plan discussed earlier for each parameter in their respective upper and lower limits.

For the first example, we will start with $n = 10$ sample data and use an infill criteria that balances improving the global minimum prediction by sampling at the location of predicted global minimum, and improving the overall accuracy of the predictor by sampling where the estimated error is the highest to select additional sampling points if needed.

Table 3.1: Parameters for the single suspension model [55]

Parameter	Value	Lower Limit	Upper Limit	Symbol
Sprung Mass	236.12 kg			m_s
Unsprung Mass	23.61 kg			m_{us}
Suspension Stiffness	12394 N/m	8000 N/m	15000 N/m	k_s
Suspension Damping	1385.4 N-s/m	500 N-s/m	1500 N-s/m	c_s
Tire Stiffness	181818.88 N/m			k_t
Tire Damping	1531 N-s/m			c_t

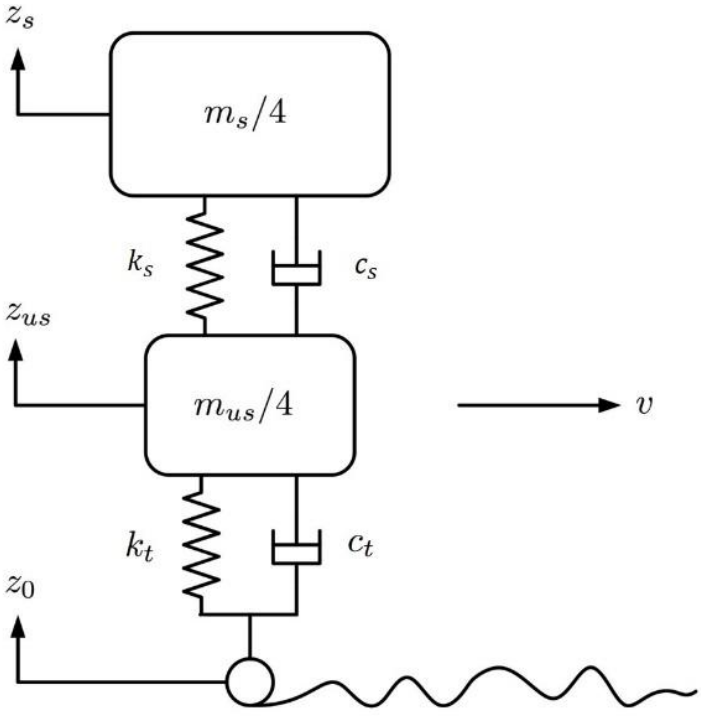

Figure 3.5: Schematics of a single suspension model with ground excitation

Figure 3.6 shows the contour plot for the mean squared error (MSE) between the predicted acceleration signal and the simulated acceleration signal. Figure 3.7 shows the contour plots of the system generated by running the original model more than 2500 times.

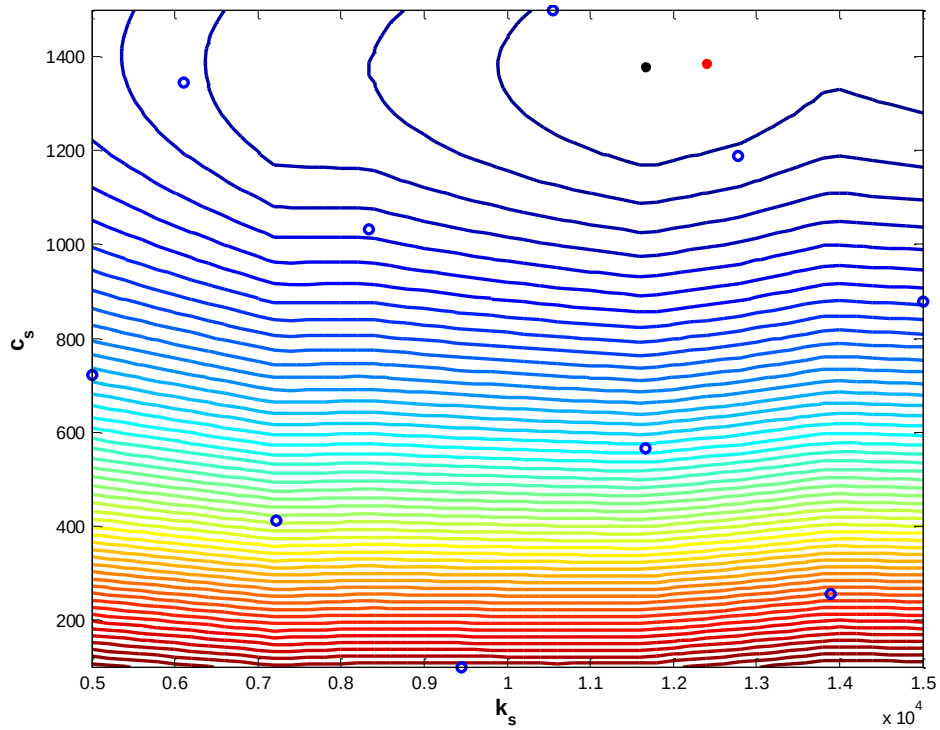

Figure 3.6: Stochastic prediction of quarter car parameters (red dot is the true parameter values, black dot is the estimated parameter values, and blue circles are 10 sampled points)

Contour plots from Figure 3.6 and Figure 3.7 ilustrate the accuracy of the stochastic model in predicting the underlying shape of the function with training data as few as 10.

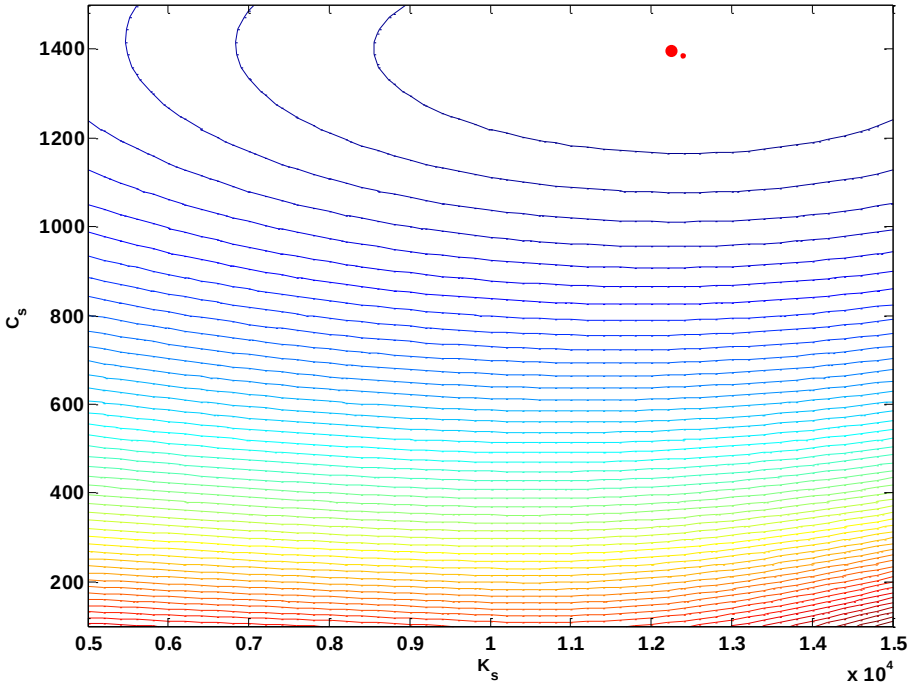

Figure 3.7: Contour plot for the original function with more than 2600 simulation runs

For the second example, we introduce two more unknown parameters (m_s, m_{us}), where

$$(50 \text{ kg} < m_s < 500\ kg) \text{ and } (5kg < m\_us < 100kg)$$

To achieve the same level of accuracy in the predicted parameters the number of training data has been increased to $n = 51$. Figure 3.8 shows the output for the stochastic model. Table 3.2 shows the results of estimating system parameters for the single suspension model.

Table 3.2: Results of estimating MBD model parameters for the single suspension model with 4 unknown parameters

Parameter	Original Value	Estimated Value	% Error
Suspension Stiffness (N/m)	12394 N/m	1236.52	0%
Suspension Damping (Ns/m)	1385.4 N-s/m	1441.66	4%
Sprung Mass (kg)	236.12 kg	228.33	3%
Unsprung Mass (kg)	23.61 kg	31.65	34%

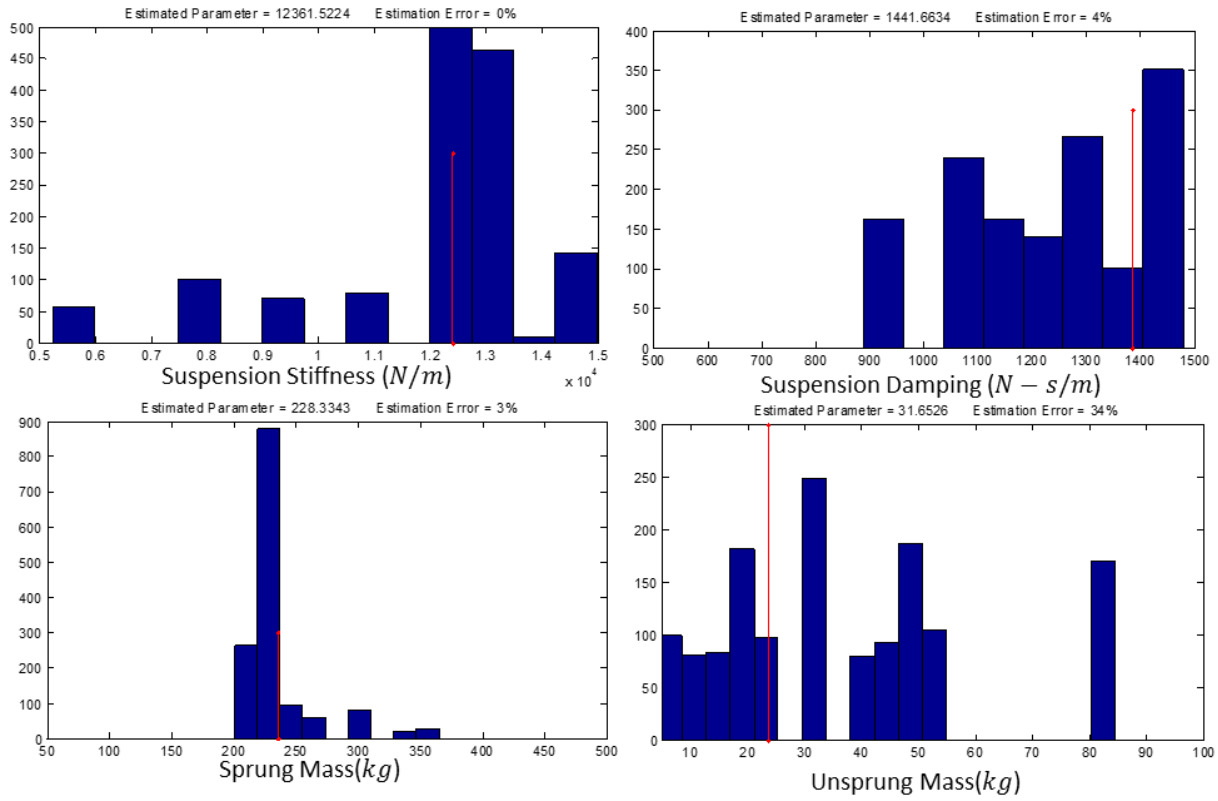

Figure 3.8: Estimated parameters for the single suspension system with 4 unknown parameters (red line indicates the actual value of the parameter)

4 Case Studies

4.1 Introduction

In this chapter, the performance of the stochastic modeling technique developed earlier is evaluated by using two dynamic systems as case studies. Once a baseline for the accuracy of the predictions is established, various techniques to improve the accuracy and efficiency of the algorithm are investigated. For the stochastic model to be able to accurately and efficiently learn the behavior of a physical system instead of a computer simulation, the algorithm is modified to compensate for the noisy measurements; various techniques to attenuate the influence of noisy data on the performance of the stochastic model are investigated.

Two nonlinear multibody dynamic systems are used as case studies. Although the systems investigated here are suspension systems, the stochastic model has been developed with the provision of working for a broad range of applications. The first MBD model is a single suspension system that utilizes nonlinear spring and damper elements along with a variable damping frictional element (similar to the vertical dynamics of the three-piece truck's secondary suspension). For the second case study, the stochastic modeling technique is used to learn the behavior of the three-piece truck's secondary suspension.

The procedure used can be summarized with the following steps: 1) run the original model and acquire the training data, 2) use the stochastic method and build a model that gets the inputs to the suspension (relative displacement and velocity across the suspension) and predicts the output (sum of suspension forces), 3) create a similar MBD model where deterministic

suspension forces are replaced with stochastic predictions, and 4) improve the performance of the stochastic model by adding carefully selected additional sample points.

SIMULINK is used as the link between the SIMPACK model and the prediction algorithm. There is some overhead associated with coupling SIMPACK with SIMULINK that can be avoided by using user-defined subroutines within SIMPACK instead of transferring the data between SIMULINK and SIMPACK. Figure 4.1 shows the SIMULINK model used.

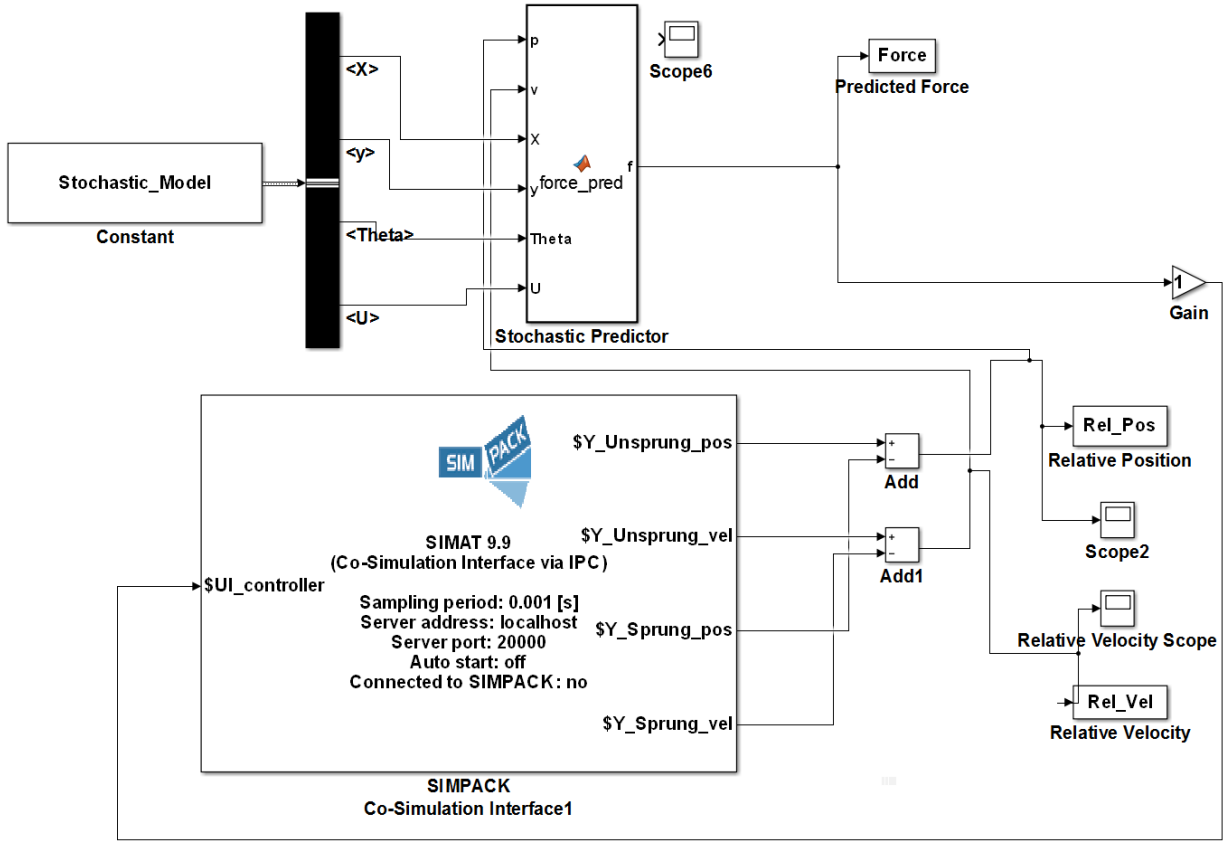

Figure 4.1: SIMULINK model that replaces the actual force elements in the MBD model

Preliminary results indicate that the accuracy of the predictions can be improved by modifying various sub-algorithms within the learning algorithm. The influence of the following parameters on the performance of the stochastic model has been investigated:

- Sampling plans
- Cost functions for identification of the optimum hyper-parameters
- Infill criteria
- Conditioning input/output signal

Normalized mean square error (NMSE), also known as Nash-Sutcliffe coefficient [56], is used to provide a measure for the accuracy of the predictions that is independent of the dynamic system being modeled. The criteria chosen has a range between $-\infty$ and 1. A value of 1 represents a perfect match between the simulated values and the predicted values. An efficiency of 0 corresponds to a model that predicts the mean of simulated values for all the inputs. Values close to 1 are desirable, but any model with an efficiency of less than 0 corresponds to a model that has a worse performance than the simple mean of the simulated values.

$$NMSE = 1 - \frac{\sum_1^n \left(f_{pred} - f_{meas}\right)^2}{\sum_1^n \left(f_{meas} - \overline{f_{meas}}\right)^2} \tag{4.1}$$

4.2 Single Suspension Model

The single suspension model (Figure 4.2) used for the purpose of this study employs a nonlinear spring, a nonlinear viscous damper, and a variable damping friction element where the normal forces depend on the deflection of the spring. Figure 4.3 shows the nonlinear relationships between the suspension's relative deflection/velocity and the resulting spring/damper forces. Relative distance and velocity across the suspension are inputs for the stochastic model, and the

sum of suspension forces ($F_{tot} = F_s + F_c + F_f$) is the output. A random base excitation is used to excite the MBD model for the acquisition of the training data.

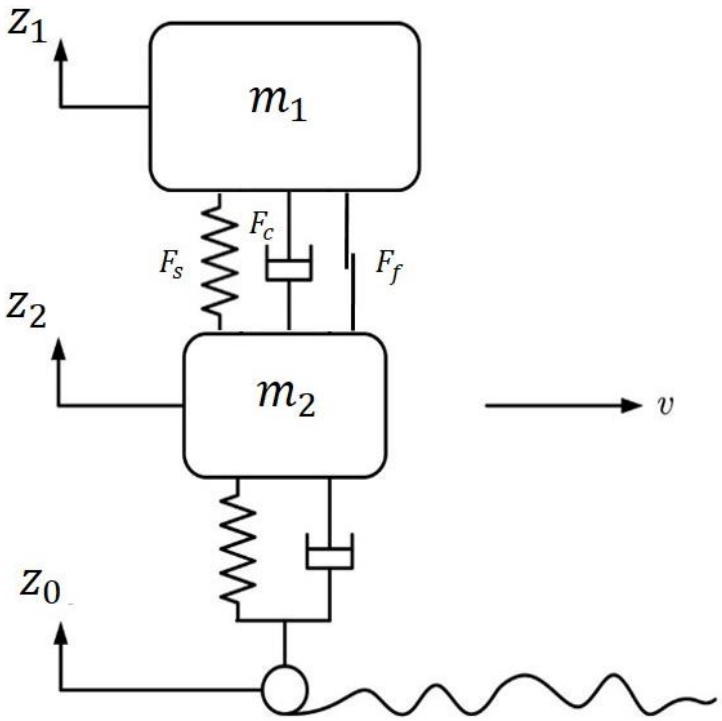

Figure 4.2: Schematics of the single suspension model used for the first case study

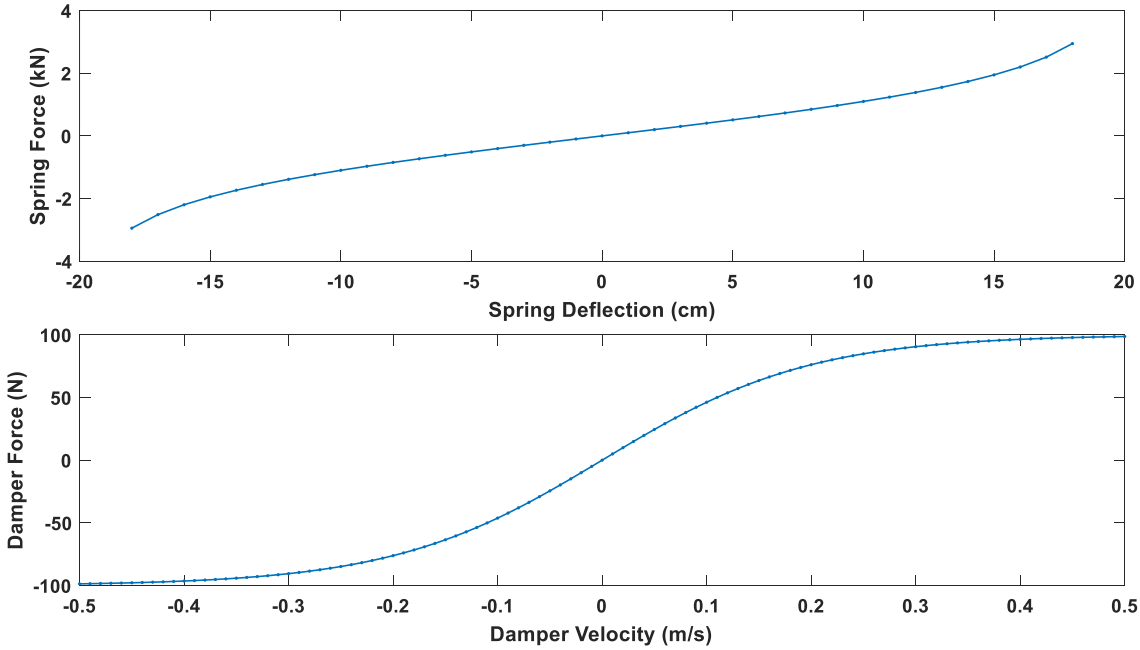

Figure 4.3: Nonlinear relationships for spring and damper elements

Figures 4.4 - 4.7 show the data acquired from running the deterministic model; a subset of this data will be used as the training data for the stochastic model.

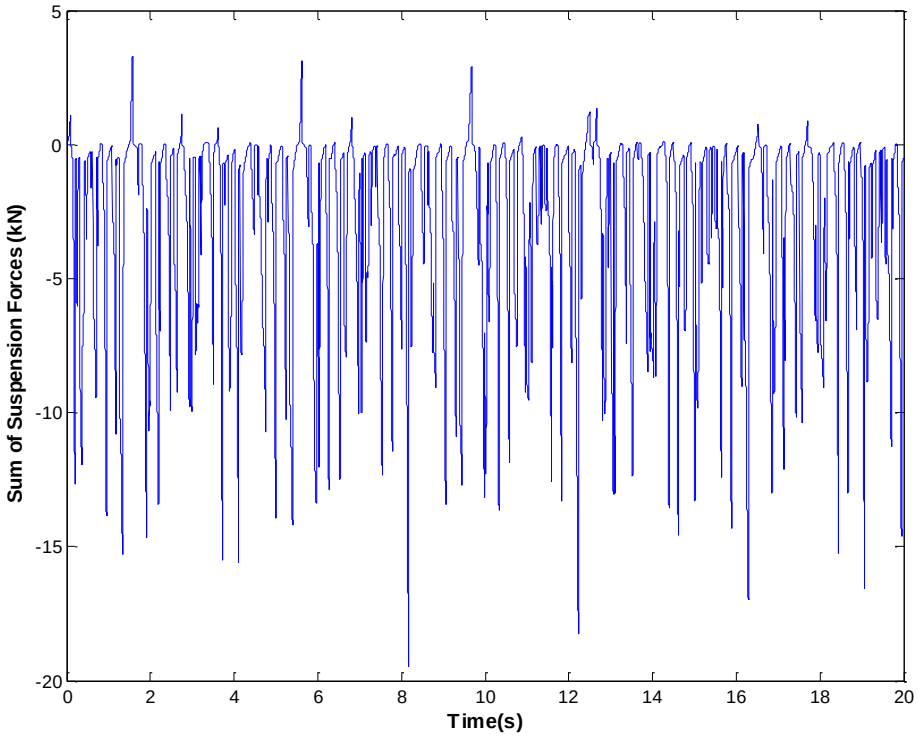

Figure 4.4: Sum of suspension forces (model output)

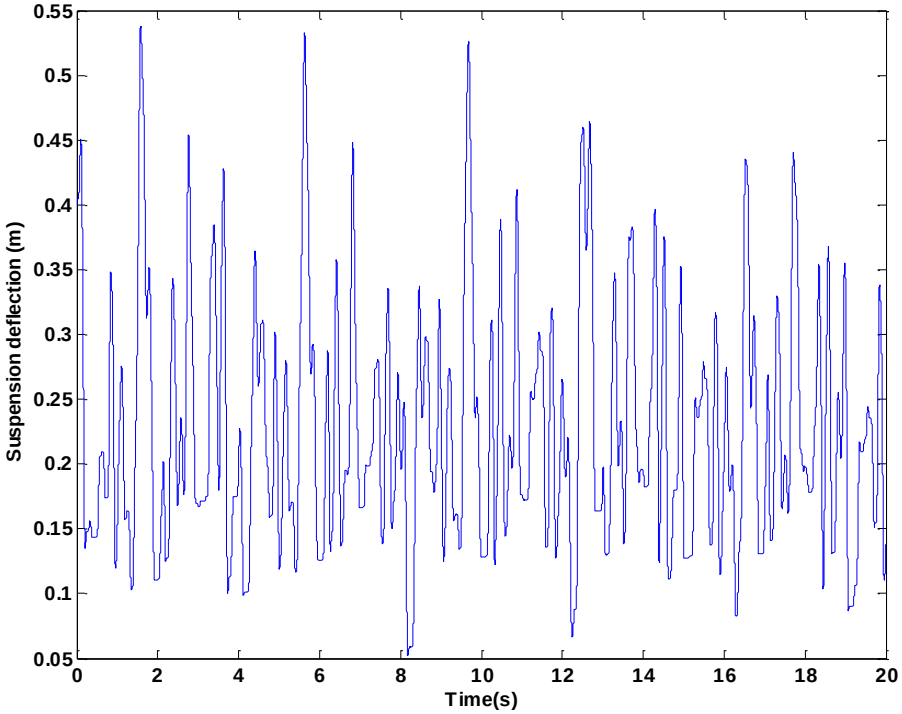

Figure 4.5: Relative deflection across the suspension (model input)

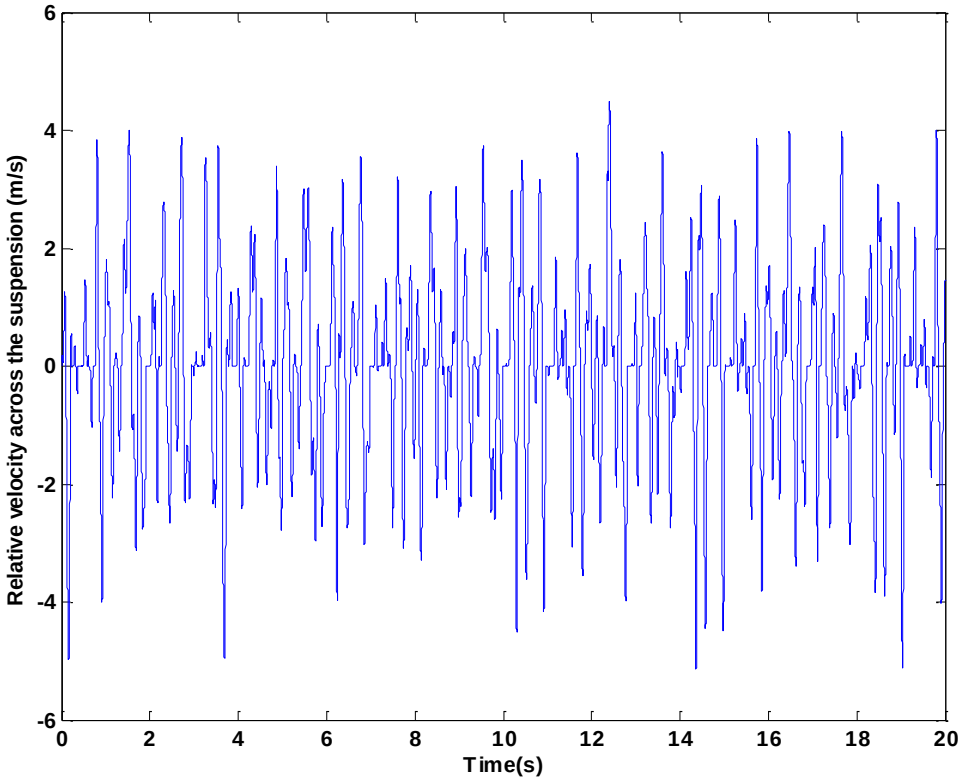

Figure 4.6: Relative velocity across the suspension (model input)

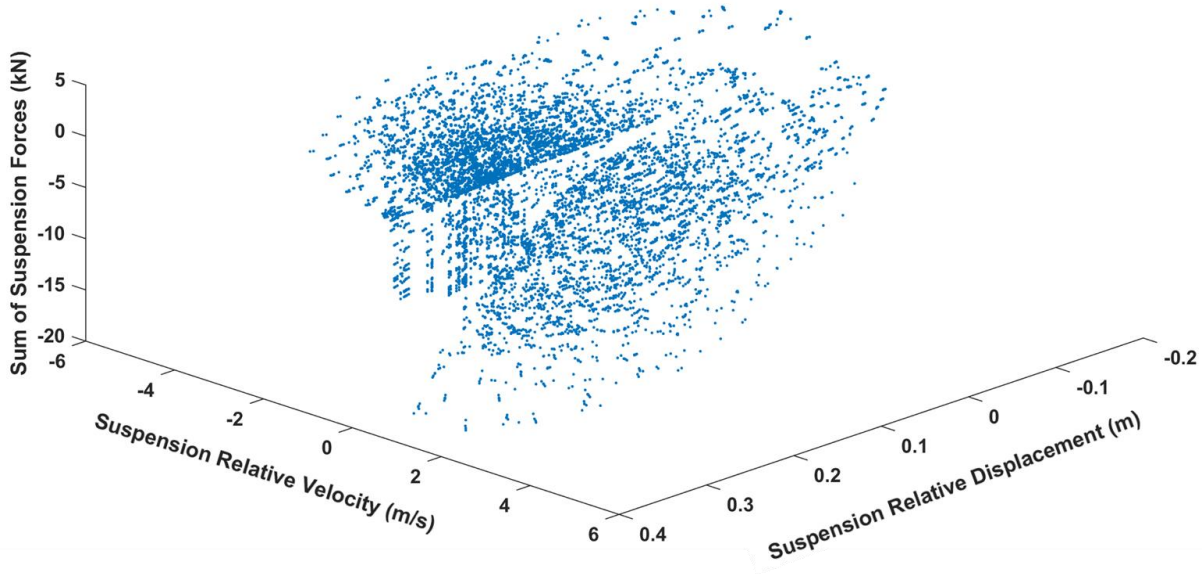

Figure 4.7: Input-output space for the single suspension model

An important part of this method is to select the training data that can represent the entire dynamic range of the suspension; concepts described in section 3.3 (constructing a sampling

plan) are used to choose the best (within the allocated computational budget) collection of velocity-deflection pairs that can represent the behavior of the suspension given a specific total number of training data points. Since all the velocity-deflection pairs produced by the Latin Hypercube algorithm are not available in the input signal, the algorithm has been modified to use the nearest point instead. Figure 4.8 shows an example of the data points selected for the training of the stochastic model using the modified Latin Hypercube algorithm.

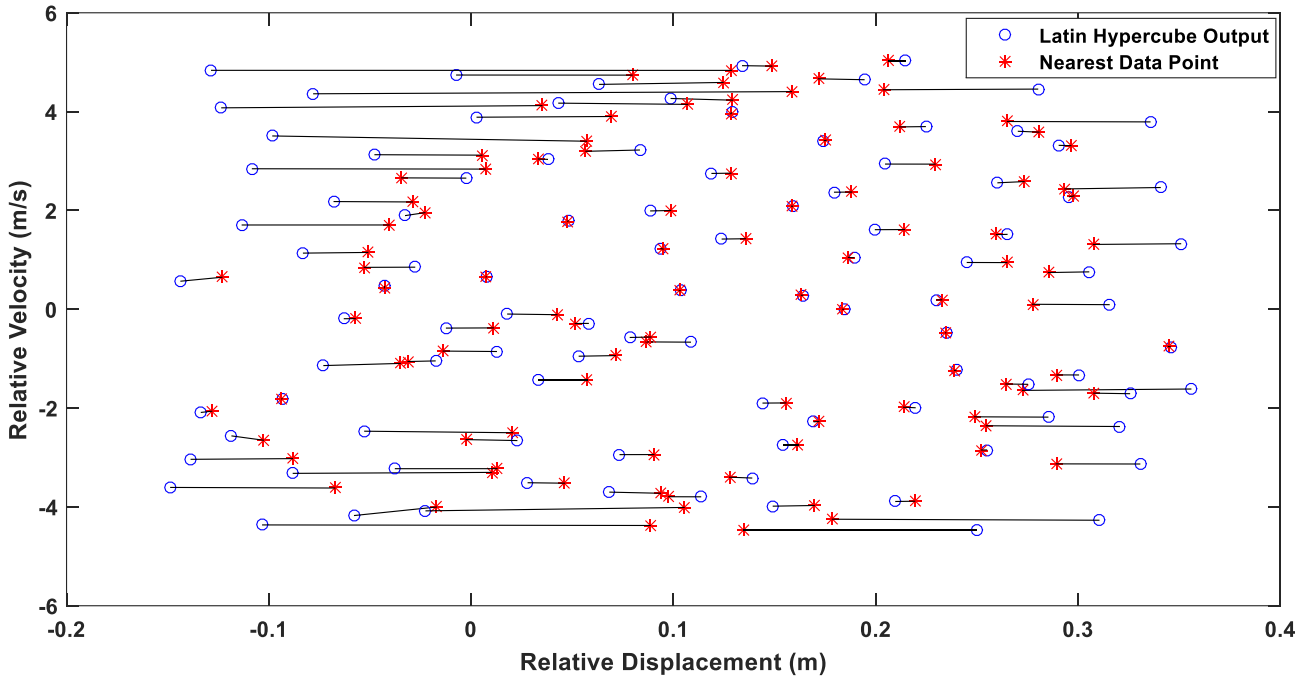

Figure 4.8: Training points selected from the input signal. The algorithm has been modified to select the nearest point in the signal.

The number of data points used for the training of the stochastic model greatly influences the accuracy of its predictions. To illustrate this point, we have trained the model with different numbers of data points and compared the results. It should also be noted that as can be seen in Figure 4.9, the computational expense of the training algorithm and the prediction algorithm

increase with the size of the training data, highlighting the importance of an efficient sampling plan.

The accuracy of global optimization techniques highly depends on the number of iterations used. Since the number of iterations for each of the sub-algorithms are different, data illustrated in Figure 4.9 is normalized with the number of iterations to provide an independent measure for the computational expense of each of the sub-algorithms. The numeric values shown in Figure 4.9 are also dependent on the computational power of the workstation, but the trends will remain the same.

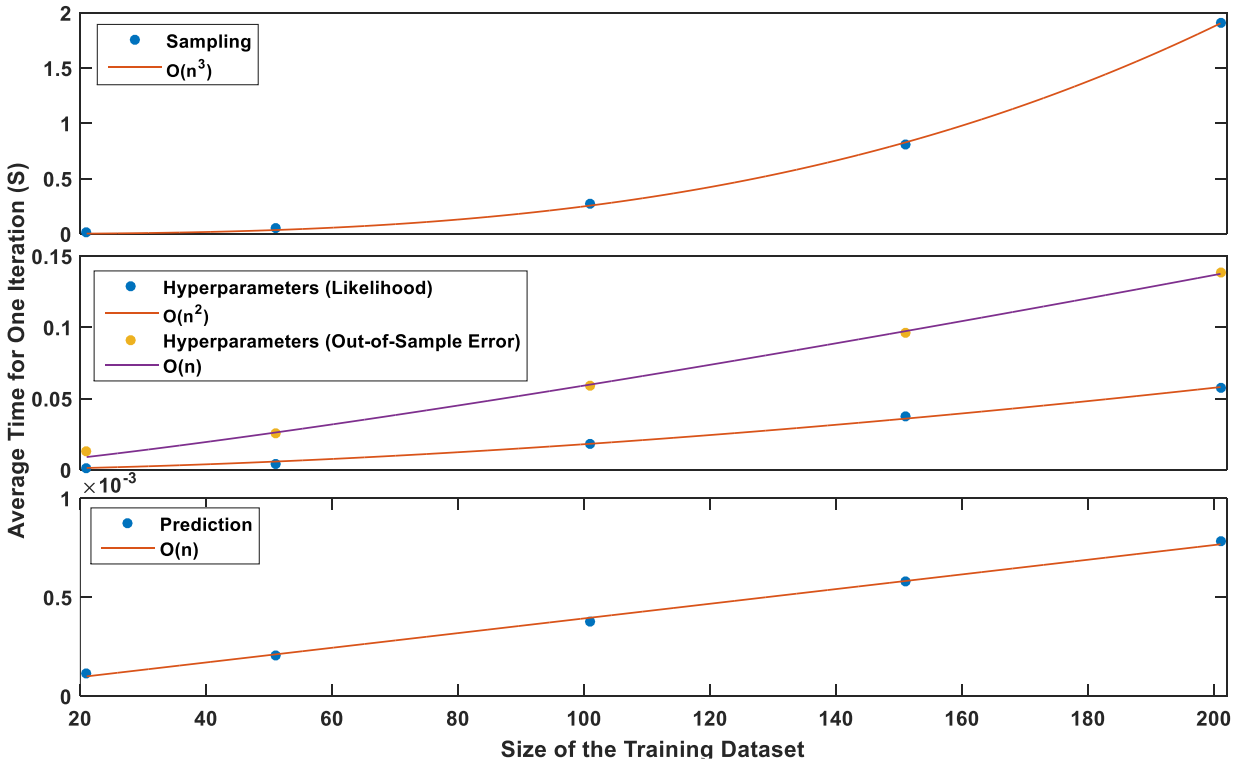

Figure 4.9: Influence of size of the training dataset on the computational expense of the algorithm

As can be seen in Figure 4.10, there is some variation in the performance of the stochastic models with the same number of training data points. These variations are caused by the fact that global

optimization techniques used in various sub-algorithms do not guarantee convergence to the global optimum, and their performance depends on the number of iterations. To attenuate the influence of these variations, multiple runs for each size of the training data are performed and the average value is used to compare the performance of stochastic models.

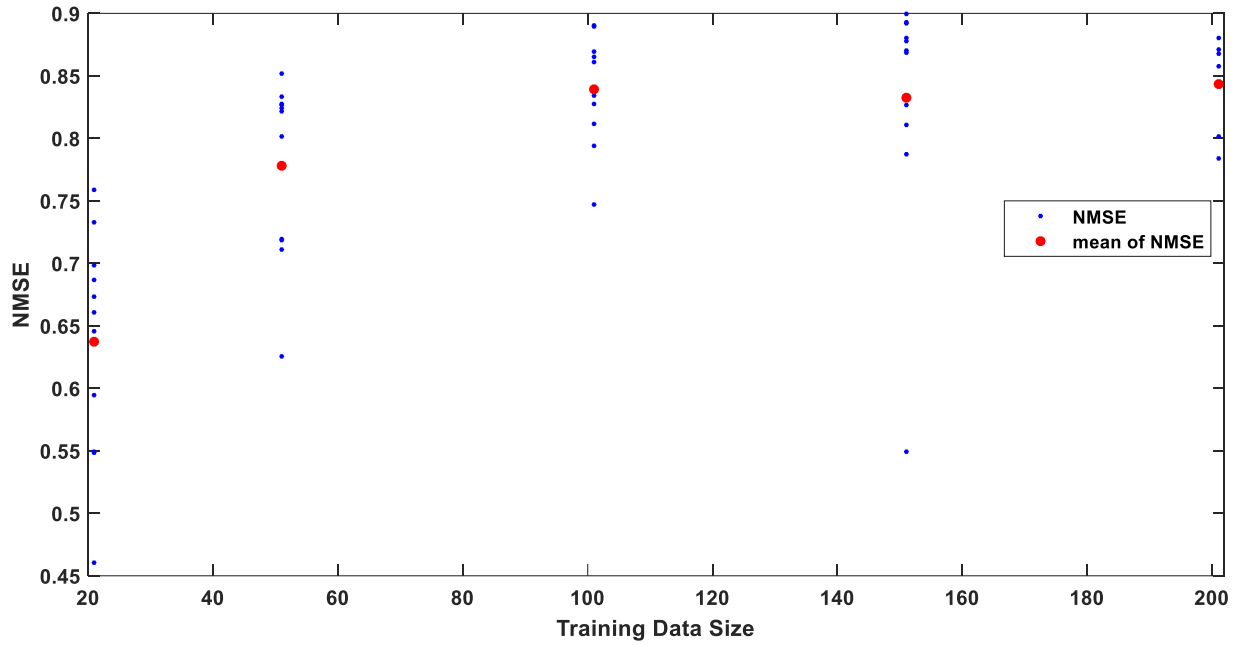

Figure 4.10: Increasing the size of the training data increases the accuracy of its predictions.

It can be seen in Figure 4.11 that while increasing the size of the training data improves the overall accuracy of the model, even the model with the highest number of training data has difficulty predicting the output in some parts of the sample space. The most efficient method of addressing this problem is to develop an algorithm that can identify these problematic locations and carefully select additional training data that can improve the accuracy of the predictions in the problematic regions mentioned above, hence improving the overall performance of the stochastic model. Figure 4.12 shows the areas of the input space for which the stochastic model has difficulty making accurate predictions.

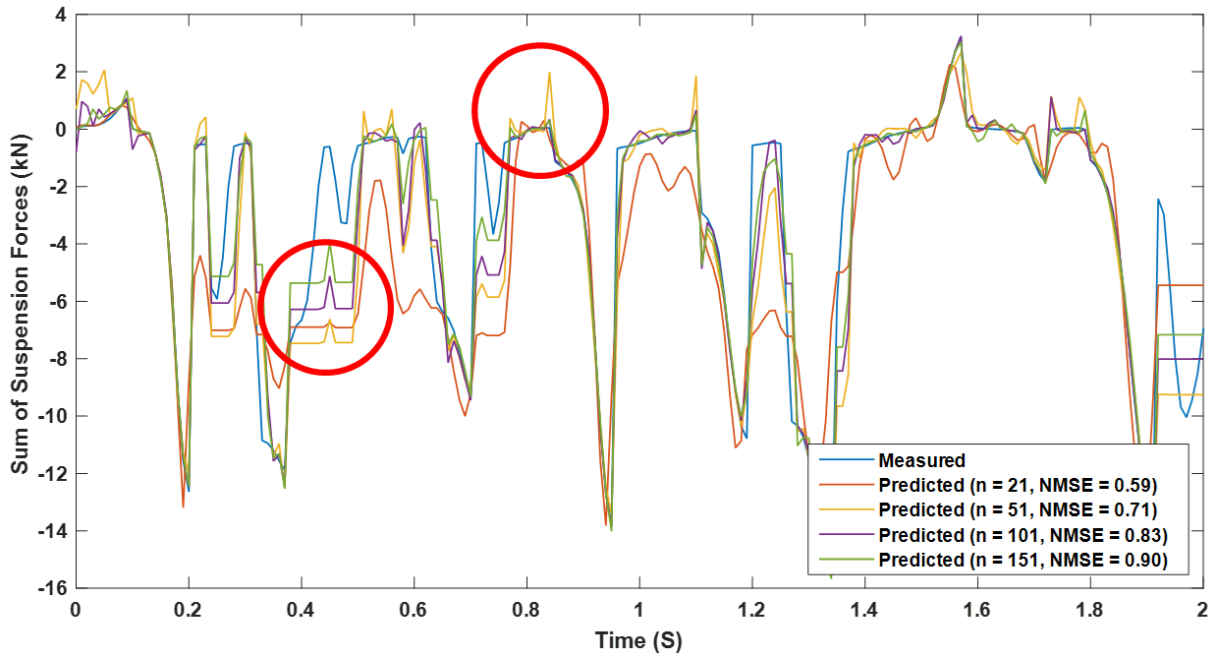

Figure 4.11: Comparison between the calculated suspension forces and the stochastic predictions. Stochastic model has difficulty predicting rapid changes in the output force

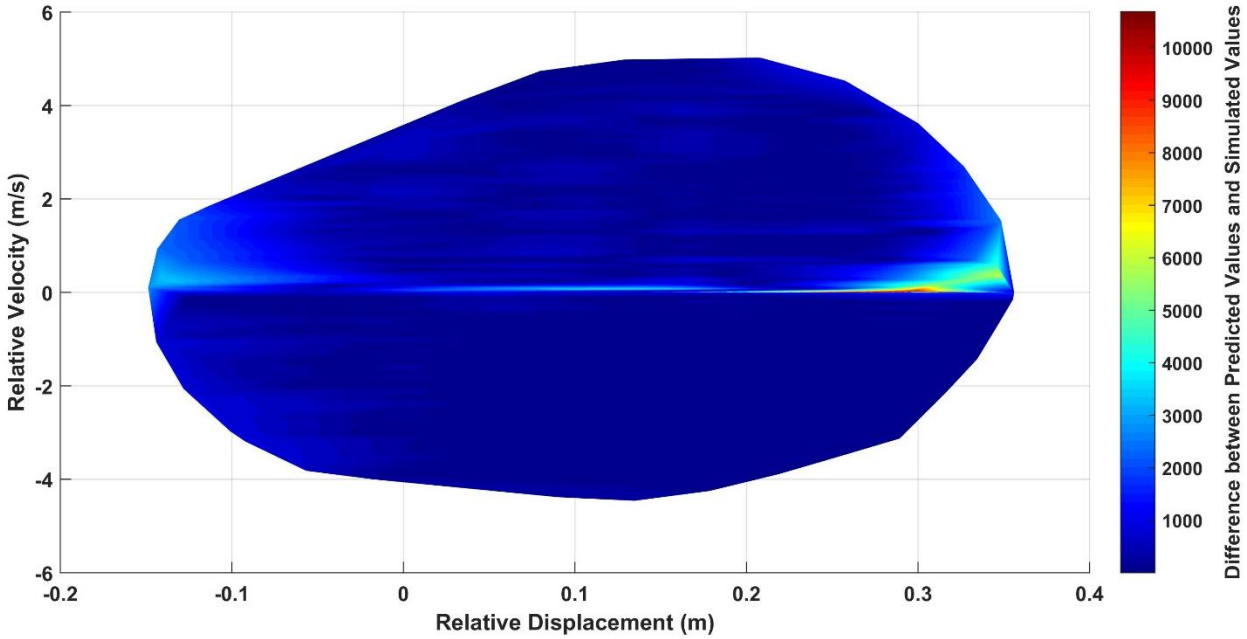

Figure 4.12: Color map for the difference between the predicted forces and the simulated forces shows the areas for which the model has difficulty predicting the output

4.3 Three-Piece Truck's Lateral Suspension Model

The next case study is the lateral secondary suspension forces of the three-piece truck model described in Chapter 2. To simplify the modeling approach and since studies on the dynamics of the three-piece truck indicated that the behavior of the truck is dominated by its motion in the vertical and lateral directions, all the degrees of freedom for the bolster except for its lateral and vertical DOFs are constrained.

The three-piece truck's lateral dynamics are coupled with its vertical dynamics through the variably damped friction wedges. Normal forces for the friction in the lateral direction are functions of the relative displacement between the bolster and the side-frame in the vertical direction, which makes the stochastic modeling approach more complex (4 inputs). The inputs to the stochastic model are the relative displacement and velocity across the suspension in the vertical and lateral directions, and the output, similar to the previous case, is the sum of the suspension forces in the lateral direction. As described in Chapter 2, the lateral dynamics of the bolster are governed by the compliance of the springs and the lateral frictional forces of the wedge. Figure 4.13 shows the truck model used to acquire the training data. Pseudo-random signals shown in Figure 4.14 and Figure 4.15 are used to excite the model in the lateral and vertical directions.

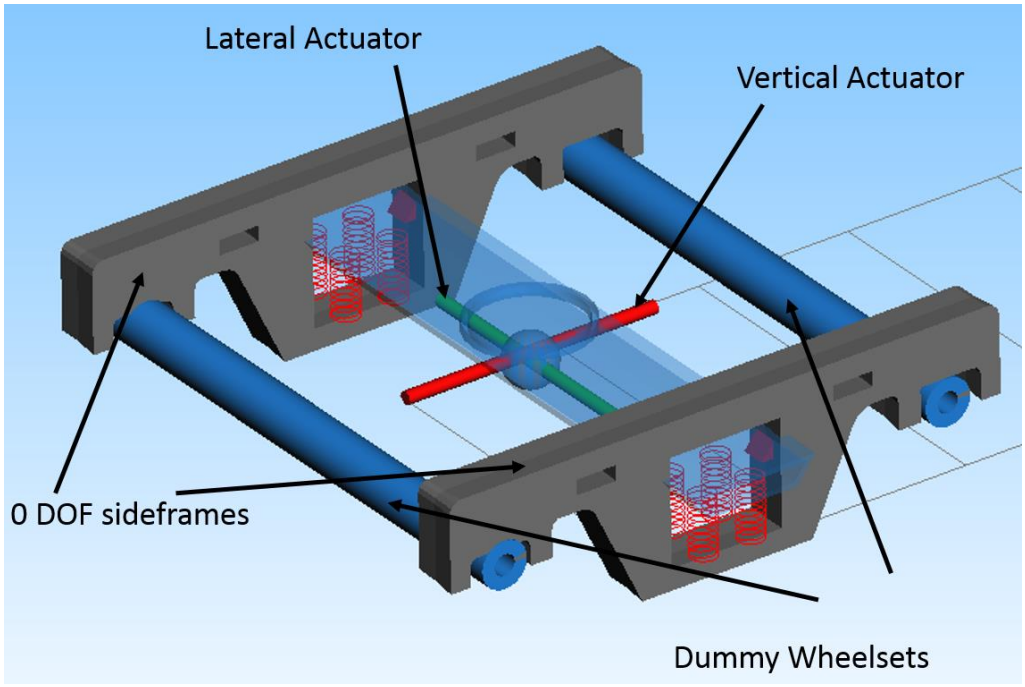

Figure 4.13: The three-piece truck has been modified to acquire the training data

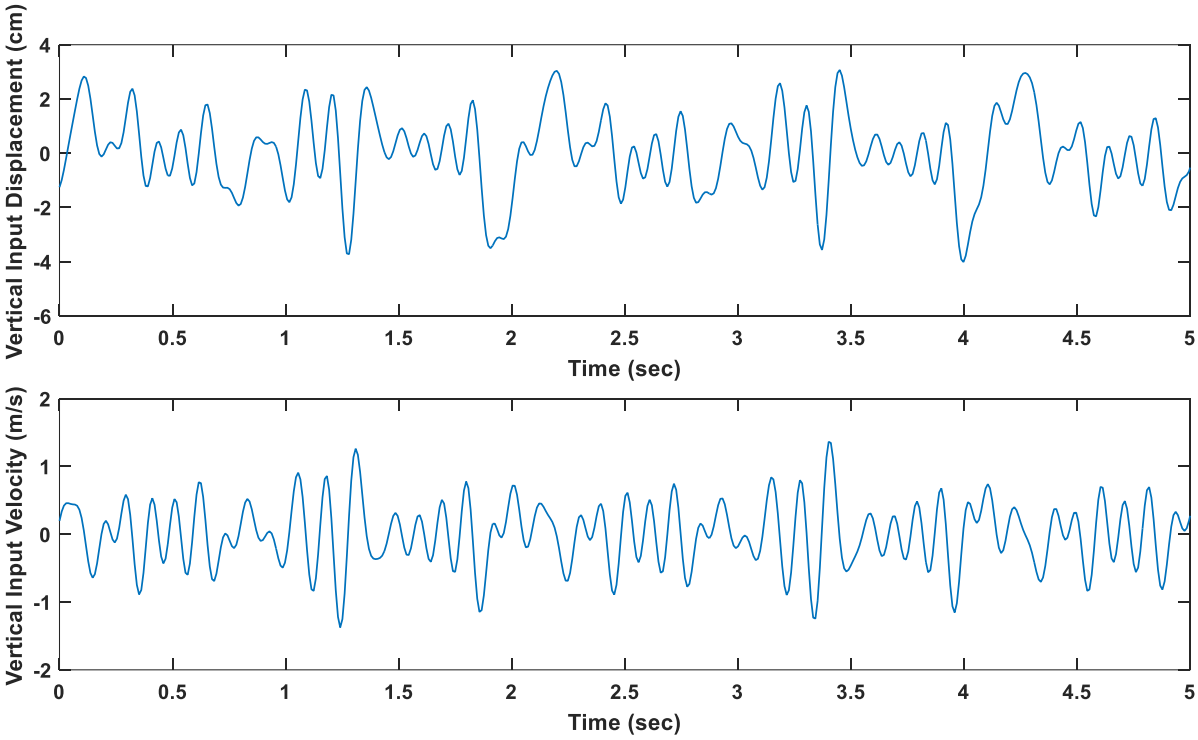

Figure 4.14: Pseudo-random input to the model in the vertical directions

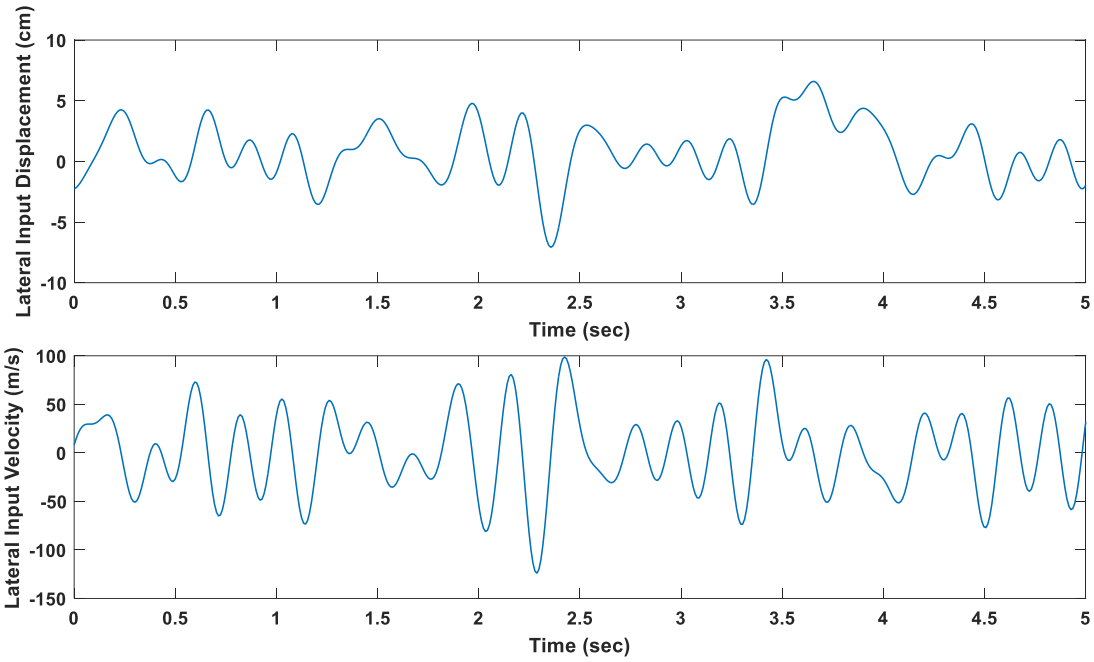

Figure 4.15: Pseudo-random input to the model in the lateral directions

Similar to the previous example, we have trained the stochastic model with various numbers of training points and investigated the influence of the number of training points on the accuracy of the predictions and the computational expense of the model. Figure 4.16 shows the influence of the number of training points on the accuracy of the predicted lateral forces.

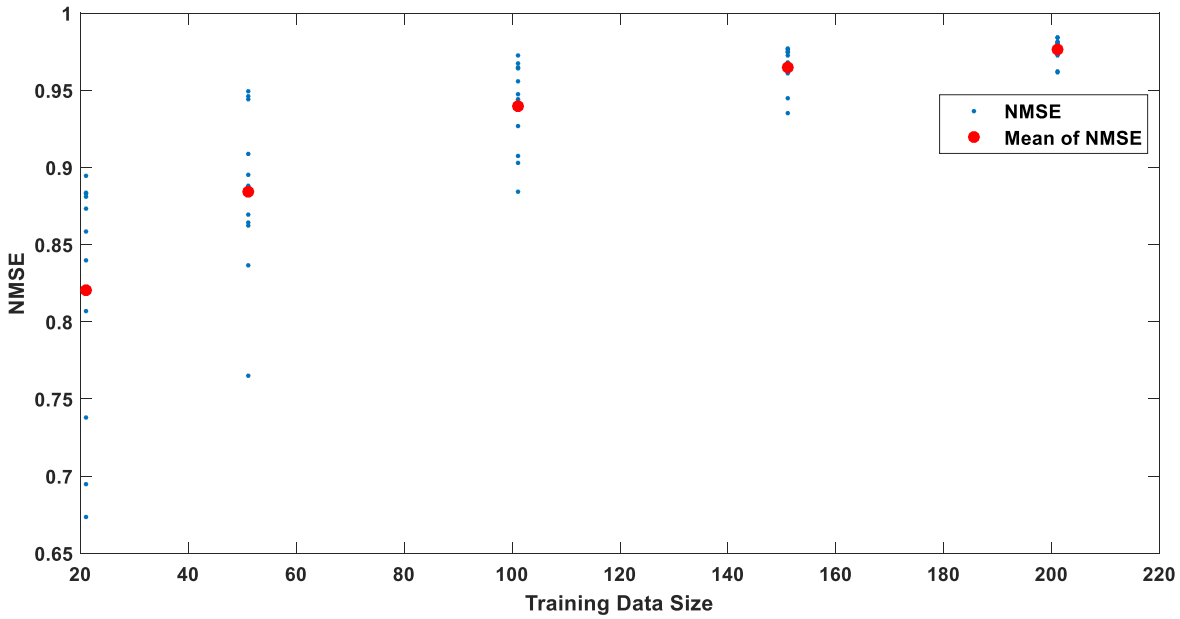

Figure 4.16: Increasing the size of the training data increases the accuracy of its predictions

Predictions follow the same trend as for the single suspension model (Figure 4.10): increasing the number of training data increases the accuracy of the predictions, but since the computational expense of the training and the prediction algorithm increase nonlinearly with the increase in the size of the training dataset, the advantage of using predicted forces instead of simulated force reduces. Figure 4.17 shows the average time required to complete one iteration of various sub-algorithms within the stochastic model.

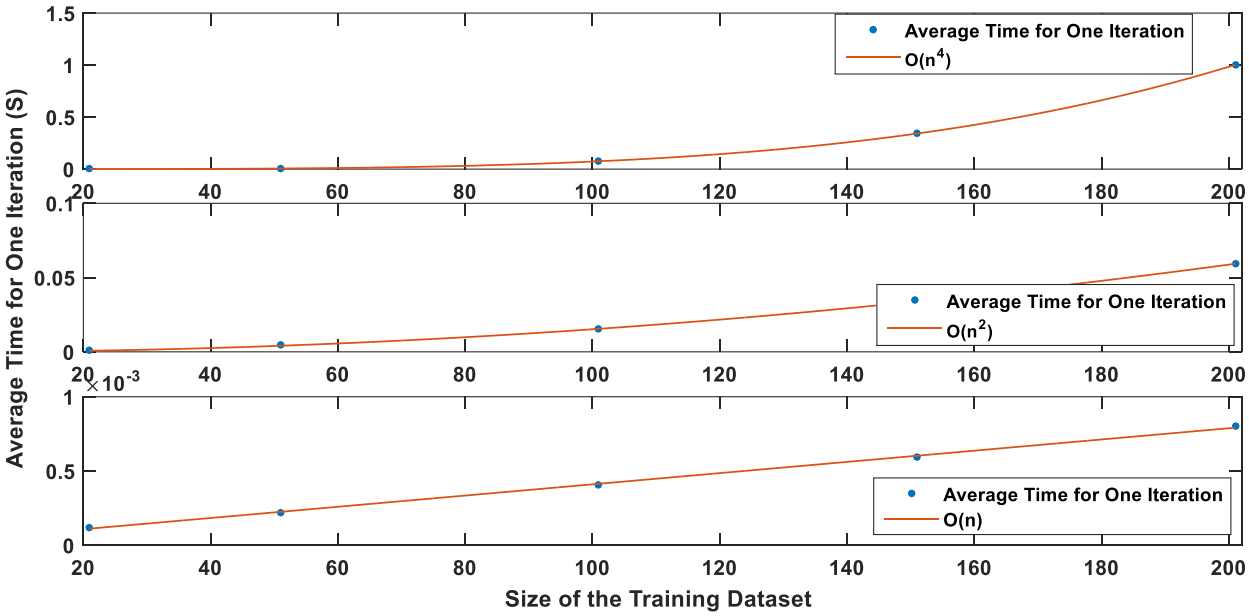

Figure 4.17: Influence of the size of training dataset on the computational expense of running the stochastic model, top to bottom: selection of the training data, finding the optimum hyper-parameters, and prediction

The stochastic predictor has difficulty predicting the sharp spikes in the suspension force, so using a low-pass filter on the training data can increase the accuracy of the predictions by eliminating the high frequency content. Figure 4.18 shows the influence of applying a 4^{th} order Butterworth filter with cut-off frequency of 15 Hz on the accuracy of the predictions. It should also be noted that although the application of a low-pass filter increases the accuracy of the predictions, it comes at the cost of eliminating some of the higher frequency dynamics. In the case of the three-

piece truck, application of a low-pass filter leads to the exclusion of the chaotic dynamics of the truck.

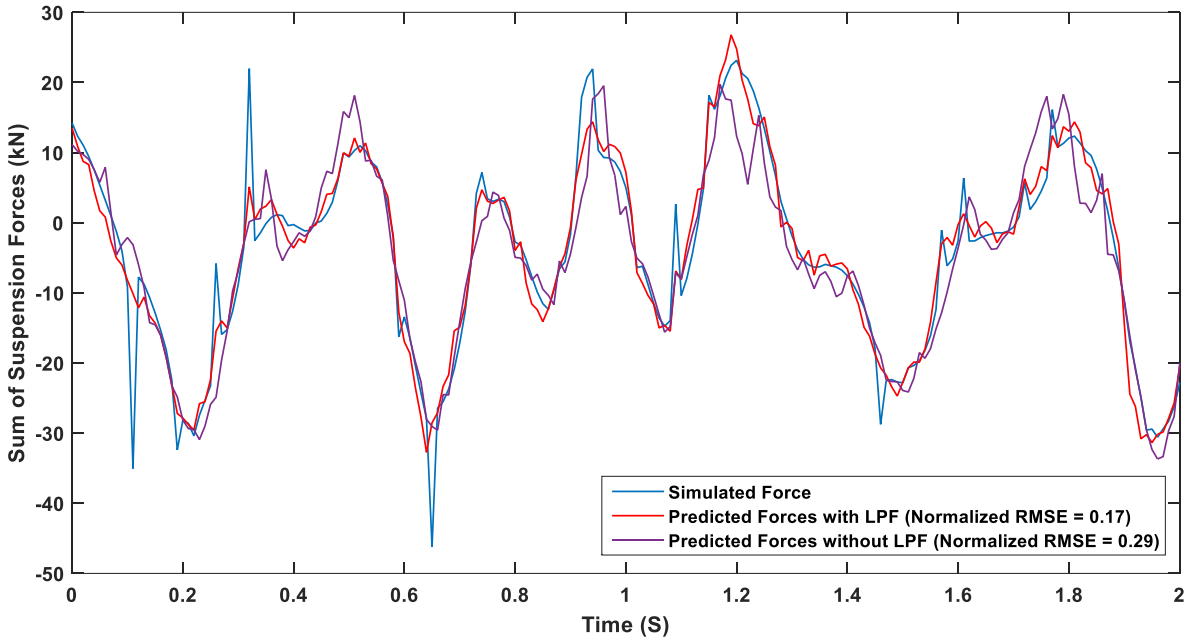

Figure 4.18: Applying a low-pass filter with cut-off frequency of 15 Hz increases the accuracy of the stochastic predictions

4.4 Improving the Accuracy and Efficiency of the Algorithm

Figure 4.9 and Figure 4.17 show the relationship between the size of the training data and the average time that is required to run a single iteration of each of the sub-algorithms. Since the computational expense of running the training algorithm and the prediction algorithm increase with the size of the training data, the goal is to increase the accuracy of the stochastic model without increasing the size of the training data.

In the previous sections of this chapter, a baseline for the performance of the stochastic model has been established. In the following sections, several techniques to increase the accuracy and

the efficiency of the stochastic model will be explored. The parameters that their influence has been investigated are as follows:

- Various methods of selecting the training data from the acquired data
- Different cost functions that can be used to find the optimum hyper-parameters for the stochastic model
- Various techniques for selecting additional new points to be added to the training data
- Using signal processing techniques

4.4.1 Selection of the Training Data

As mentioned earlier, the subset of the acquired data that is used for the training of the stochastic model has a significant effect on the overall accuracy of the stochastic model. In section 4.3, we built the most space-filling sampling plan using the Latin Hypercube algorithm. The advantage of using such a sampling plan is that it can be used for all other models with the same number of parameters and the same number of training points, thereby reducing the computational cost of the process. The downside of this method is that we are neglecting the fact that unlike most Latin Hypercube applications, here the data are acquired prior to building the sampling plan. It can be seen in Figure 4.19 that since the points in the Latin Hypercube sampling plan are replaced with the nearest point in the acquired signal, as the size of the sampling plan increases, a bigger portion of the training points are sampled at the outer edge of the sample space, which in turn reduces the efficiency of the stochastic model. To overcome this problem, a sampling plan based on the Traveling Salesperson (TSP) algorithm was built, and the influence of using such a sampling plan is studied. A pseudo-code for the algorithm is shown in Figure 4.20.

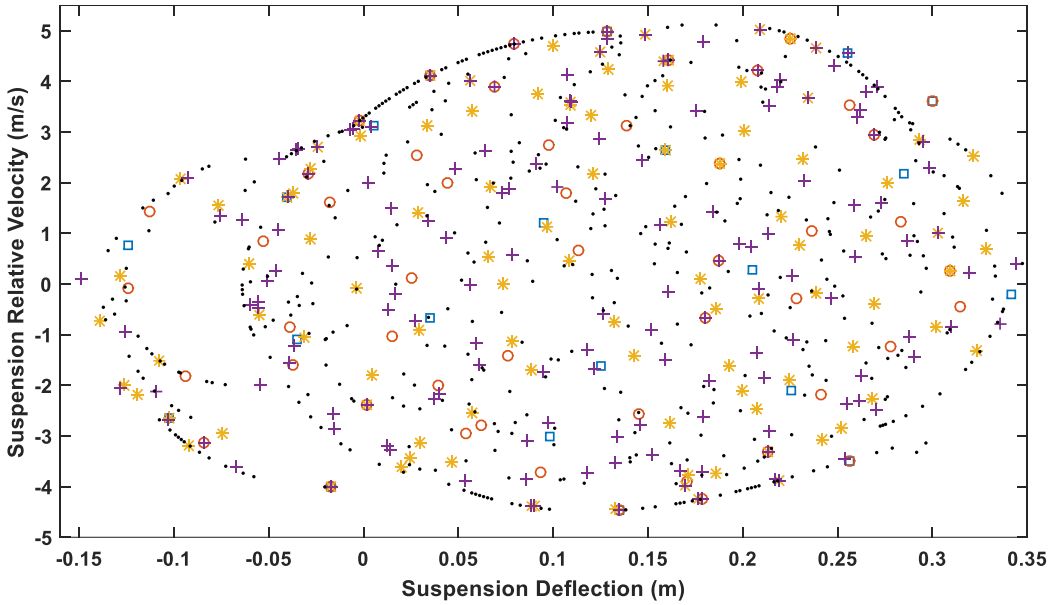

Figure 4.19: Sampling plan based on the Latin Hypercube method loses efficiency as the size of the training data increases

Figure 4.21 shows the comparison of the accuracy of each of the previously mentioned sampling plans. It can be seen that the TSP algorithm has significant advantage over the LH algorithm.

function *SamplingPlan*
Inputs:
acquired dataset: D
Number of desired training points: n
Number of iterations: iter
Outputs:
Most space-filling Sampling Plan: S

S: Random subset of the acquired data
Φ_{best} = Calculate the space-fillingness using Equation 5.24
For I = 1 to iter
S_{new} = Replace two point in S with points from D
Φ_{new} = Calculate the space-fillingness
if ($\Phi_{new} < \Phi_{best}$)
Accept Step:
$S = S\_new$
$\Phi_{best} = \Phi_{new}$
else
Reject Step
end if
end for
Return S
End function

Figure 4.20: Pseudo-code for the TSP sampling plan

Figure 4.21: Influence of using TSP algorithm for the selection of training data

4.4.2 Cost Functions for the Selection of Hyper-parameters

There are various cost functions that can be used for the selection of optimum hyper-parameters of the stochastic model [57]. In this section, two of the most popular functions are compared: 1) the likelihood function that has been discussed in detail in the previous chapters, and 2) the out-of-sample error (cross-validation). The performance of each method is evaluated in two areas: 1) accuracy of the stochastic model produced represented by the normalized NMSE, and 2) the computational expense of running the algorithm.

To measure the out-of-sample error, the model is cross-validated against a subset of the data that has not been used in the training stage, and a global optimization algorithm is used to find the optimum hyper-parameters. As can be seen in Figure 4.22 and Table 4.1, the increased

accuracy of using the out-of-sample error instead of the likelihood function shrinks as the size of the training dataset increases; on the other hand, the computational expense of calculating the out-of-sample error is much higher than the computational expense of calculating the likelihood function.

Figure 4.22: The improvements of using the out-of-sample error as the cost function reduces with increase in the size of the training data

Table 4.1: Influence of various cost functions on the efficiency of the stochastic model

Training Data Size	NMSE		% Improvement	% Increase in Computational Expense
	Likelihood Function	Out-of-sample Error		
21	0.64	0.73	12.28	91.84
51	0.78	0.81	4.36	84.39
101	0.84	0.84	0.53	68.19
151	0.83	0.86	3.37	61.52

4.4.3 Infill Criteria

As mentioned earlier, the accuracy of the predictions can be improved by adding carefully selected additional data points (infills) to the training data. In order to find the criteria for selecting infills, 5000 points were randomly selected from the acquired data, individually added to the training set, and without changing the hyper-parameters for the stochastic model, the normalized mean squared error (NMSE) between the predicted forces and the measured forces was calculated. Based on the results of this experiment (illustrated in Figure 4.23), and using various parameters that seemed the most influential in the overall performance of the stochastic model, several infill criteria have been developed. Criteria that have been investigated in this research are as follows:

- Brute force
- Difference between the predicted and measured force values
- Number of data points within a specific distance of each data point (point clusters)
- $\hat{s}^2$ parameter defined by Equation (3.37)

In the brute force method, 1000 points in the simulated signal are individually added to the training set and, without changing the hyper-parameters, the resulting NMSE is calculated; the point that yielded the lowest NMSE is then added to the training set. The same procedure is repeated 50 times. For each of the other parameters mentioned above, a threshold of mean plus two standard deviations is defined, and the signal is filtered to those values that are bigger than the threshold. A subset of 50 points that would yield the lowest NMSE was then selected using a global optimization algorithm to be added to the base model with 101 training points, yielding a

model with the total of 151 data points. Table 4.2 shows the results of employing various infill criteria mentioned above for the selection of additional points.

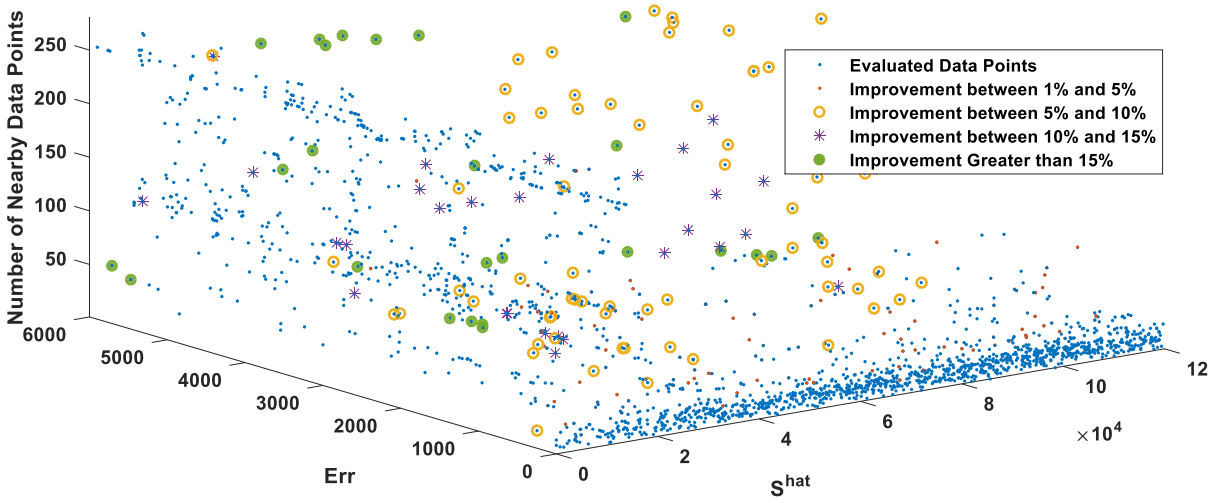

Figure 4.23: Evaluating the sample space shows that points with high $\hat{S}$, Err and high number of nearby points are good locations for infills

Results indicate that the accuracy of the stochastic model can be greatly improved by using a limited number of carefully selected additional data points. To illustrate the efficiency of using infills instead of initial sample points, Figure 4.24 shows the output of the stochastic model for 201 initially selected data points in comparison with 101 initially selected data points with 50 infills that would give a total of 151 training points. The "Difference between Measured Values and Estimated Values" infill criterion is used to select the additional training points.

Table 4.2: Performance improvements of various infill criteria

Infill Criteria	NMSE for		% improvement[5]	
	SS	TPT	SS	TPT
Brute Force	0.9856	0.9751	15.552	43.01
$\|f_{pred} - f\_simu\|$	0.9925	0.9745	16.139	26.68
Number of Neighboring Points	0.9918	0.9781	16.080	42.86
Combination of Methods 2 and 3	0.9866		15.637	
s^2	0.8639	0.9753	3.655	43.94

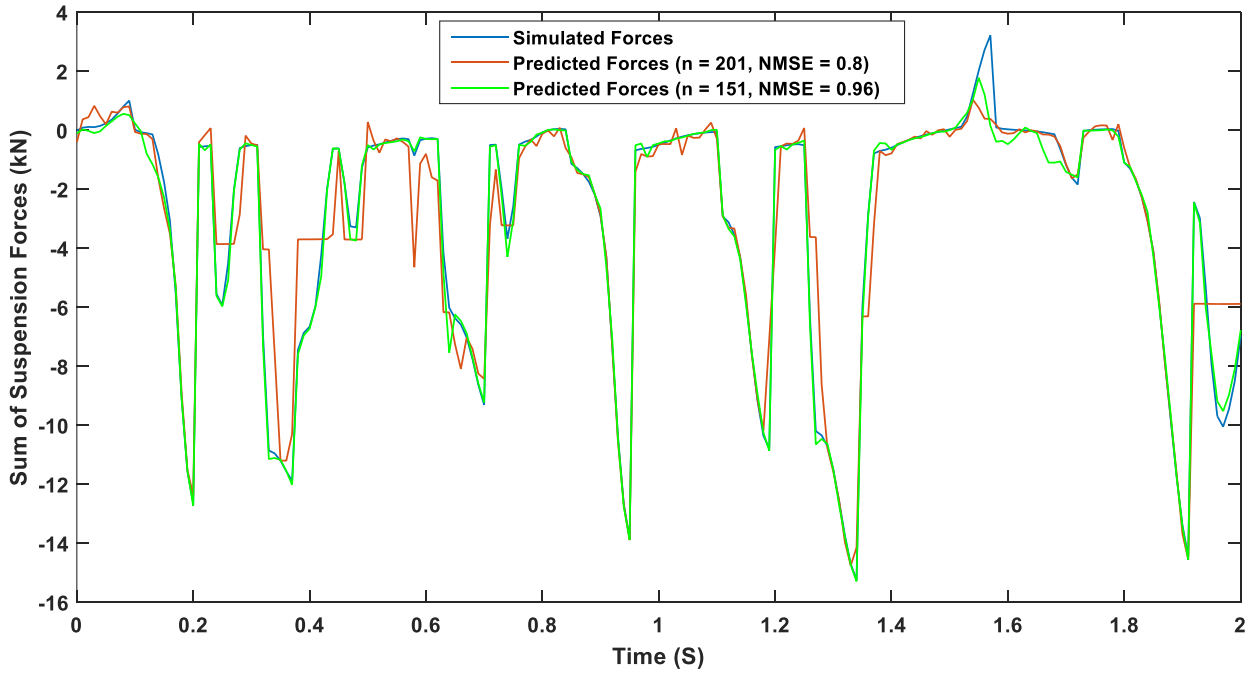

Figure 4.24: Predicted forces and simulated forces for the single suspension model; using carefully selected additional training points can improve the efficiency of the model

The improved accuracy caused by the infills is due to the fact that the criteria developed can identify the parts of the sample space where the stochastic model has difficulty making accurate predictions. Once these areas have been identified additional, sample data can be selected from them, which would improve the stochastic model's understanding of the underlying behavior of

[5] Based on the mean NMSE value of 11 stochastic models with 151 training data points.

the system in its entire design space. Mathematically this can be explained using Equation (3.35). The term $1 - \boldsymbol{\psi}^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}$ is a measure of uncertainty regarding our predictions, and the techniques explored decrease the mentioned term, which in turn will increase the likelihood of the predicted value calculated using Equation (3.35).

4.5 Integration of Test Data in a Multibody Dynamic Model Using the Stochastic Model

As mentioned earlier, accuracy of a multibody dynamic model highly depends on the system parameters used to develop the model. As outlined in [58], system identification techniques can be used to solve the problem. System identification procedures can be summarized as follows:

- Develop a mathematical model
- Choose the initial system parameters based on direct measurement, calculation, or estimation
- Develop a metric that shows the goodness of simulation results relative to test data as a function of system parameters
- Use optimization techniques to choose system parameters that minimize the previously developed cost function

Although this can be an efficient approach for systems with a limited number of parameters as evident in [58], the computational expense of estimating system parameters rapidly increases with the number of parameters. In this section, the stochastic model is used to learn the behavior

of a physical system and directly integrate the laboratory acquired data into the multibody dynamic model.

For the stochastic model to be able to learn the behavior of a physical system, the algorithm has to be modified to compensate for the noise in the acquired data. This gives the stochastic model the advantage of being able to accurately and efficiently learn and predict the behavior of physical systems. Various techniques to compensate for the noisy data include, adding the variance of the noise (τ^2) to the diagonal elements of $\boldsymbol{\Psi}$, averaging the values of the nearby points, and using signal processing techniques, have been investigated.

While using the stochastic model to replace parts of a mathematical model can reduce the computational expense of MBD analysis, incorporating the behavior of physical systems within the MBD model provides the added advantage of reducing the complexity of developing such models, and eliminating the need for system identification. Data acquired from laboratory experiments can be used directly in the MBD model, eliminating the need for detailed mathematical representation of various elements in the system.

To quantify the influence of noise on the accuracy of the predictions, different levels of noise are artificially added to the simulation data for the single suspension and the three-piece truck models, and the results of each of the above-mentioned methods are compared with the base scenario where there was no added noise. Based on the results from previous analysis, the stochastic model with 101 training data points is the most efficient model, hence, to study the influence of noise, only the case with 101 training data points is used.

As mentioned in section 3.4.1, adding the variance of the noise (τ^2) to the diagonal elements of $\boldsymbol{\Psi}$ causes the stochastic predictor to no longer be an interpolator, as the predictions will no longer pass through the sampled data, which in turn will prevent the algorithm from fitting the noise. The disadvantage of this method is that since τ^2 is a parameter of the stochastic model that needs to be optimized, it is added to the list of parameters whose values are determined by the global optimization routine, which will increase the complexity of the model.

Various digital signal processing (DSP) techniques can be used to improve the signal to noise ratio (SNR) of the acquired data. The influence of two of the most popular signal processing methods (moving average, low-pass filter) on the accuracy of the stochastic model's predictions is investigated in this section. The acquired data are pre-processed before being used by the learning algorithm. Figure 4.25 shows the influence of the aforementioned techniques on the acquired data.

A subset of the data shown in Figure 4.25 will be used to train the stochastic model. Based on the nomenclature used in Chapter 3, elements of $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}\right\}^T$ are chosen from the top two subplots of Figure 4.25, and elements of $\boldsymbol{y} = \left\{y^{(1)}, y^{(2)}, \dots, y^{(n)}\right\}^T$ are chosen from the third subplot.

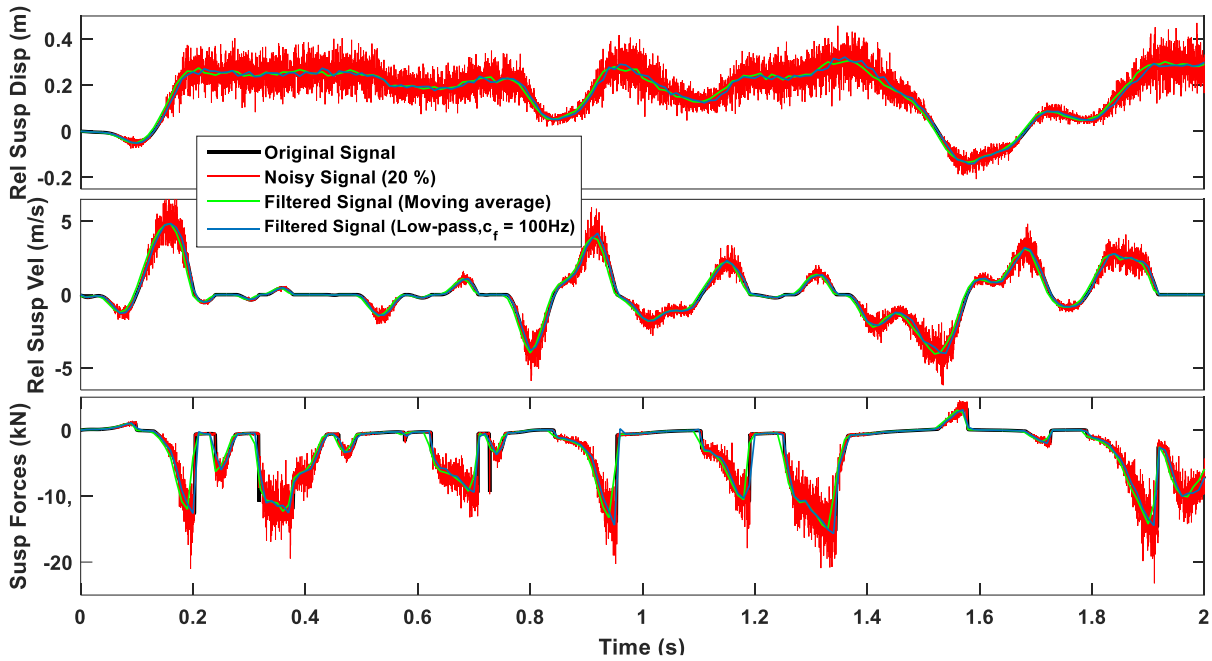

Figure 4.25: Influence of digital signal processing on the noise; the pre-processed signal is inputted to the learning algorithm instead of the original signal to reduce the influence of noise on the accuracy of the predictions

As can be seen in Figure 4.25, the influence of applying the moving average filter on the noisy data is such that the filtered noisy data and the original signal lie of top on each other. Applying a low-pass filter shifts the signals in time, but using Matlab's *filtfilt* function eliminates the phase shift caused by applying LPF. As illustrated in Figure 4.26, the single suspension system amplifies the noise in the input. Since the assumption is that the noise in the acquired data is from the sensor and not from the dynamics of the system, and because once the data is filtered the difference in the signal to noise ratio between the signal with artificially added noise and the signal with noisy input is less than 20%, the assumption of adding the same level of noise to the input and output signals is an accurate assumption. Table 4.3 shows the results of applying various signal processing methods to the noisy data and then running the stochastic modeling algorithm.

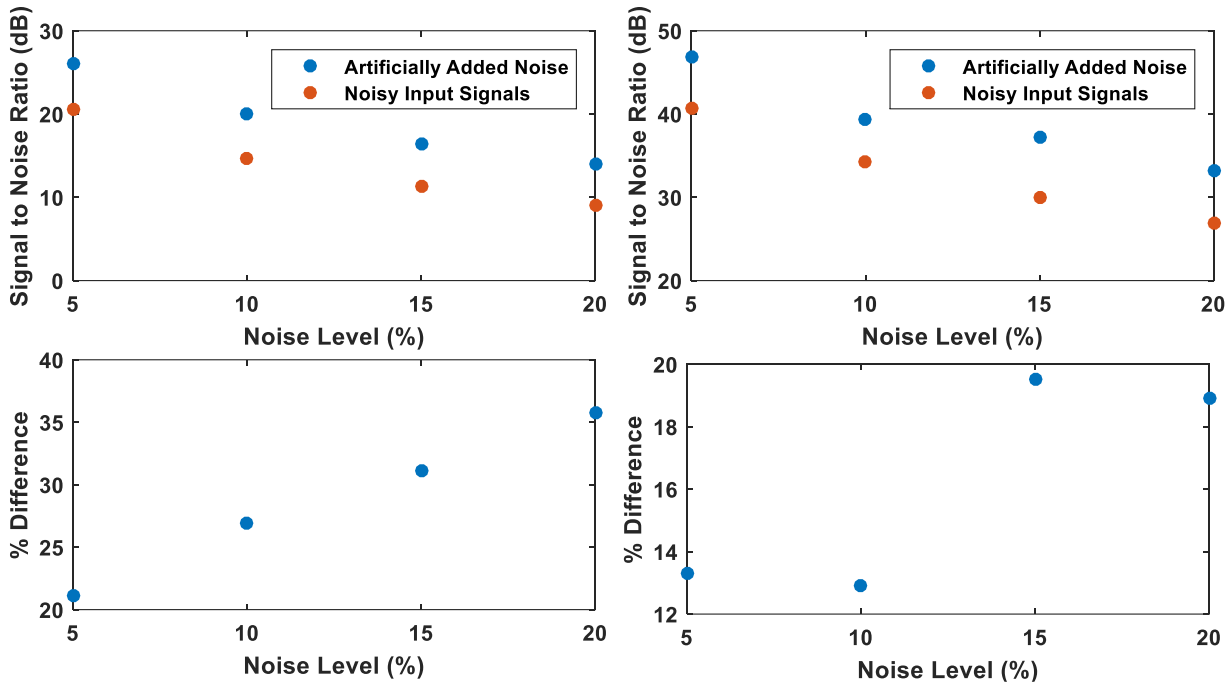

Figure 4.26: The single suspension system amplifies the noise in the input signal

Table 4.3: Investigating the techniques for reducing the influence of noise on the stochastic predictions for the single suspension model

Noise Level	NMSE				% Improvement		
	Original	Tau	Moving Average (MA)	Low-pass Filter (LPF)	Tau	MA	LPF
0	0.77	0.77	0.78	0.78	0.44	4.28	4.35
5	0.68	0.68	0.73	0.72	1.44	16.98	12.50
10	0.69	0.7	0.72	0.7	3.41	7.78	3.23
15	0.47	0.48	0.65	0.56	3.12	31.83	16.98
20	0.44	0.46	0.57	0.5	3.56	20.62	10.71

Since adding the power of noise to the diagonal elements of Ψ did not result in much improvement in the accuracy of the predictions, it was not considered for the study of noise in the three-piece truck model. Table 4.4 shows the results of using the stochastic modeling approach to learn the behavior of the three-piece truck's secondary suspension from noisy data. Although, the accuracy of the prediction have decreased in comparison with noise-free case, the method is proved to be an accurate replacement for system identification methods.

Table 4.4: Investigating the techniques for reducing the influence of noise on the stochastic predictions for the three-piece truck model

Noise Level	NMSE			% Improvement	
	Original	Moving Average (MA)	Low-pass Filter (LPF)	MA	LPF
0	0.82	0.88	0.93	32.72	61.11
5	0.81	0.87	0.91	28.98	52.63
10	0.81	0.84	0.91	17.74	52.63
15	0.69	0.8	0.9	34.73	67.74
20	0.67	0.76	0.89	27.01	66.67

Applying a low-pass filter to the data, as seen in the previous section, not only increases the accuracy of the predictions but can also reduce the influence of noise on the output of the stochastic model. Figure 4.27 shows the results of using the stochastic modeling technique to estimate the single suspension model's sum of suspension forces. The input signals are pre-processed to attenuate the influence of noise on the model's predictions.

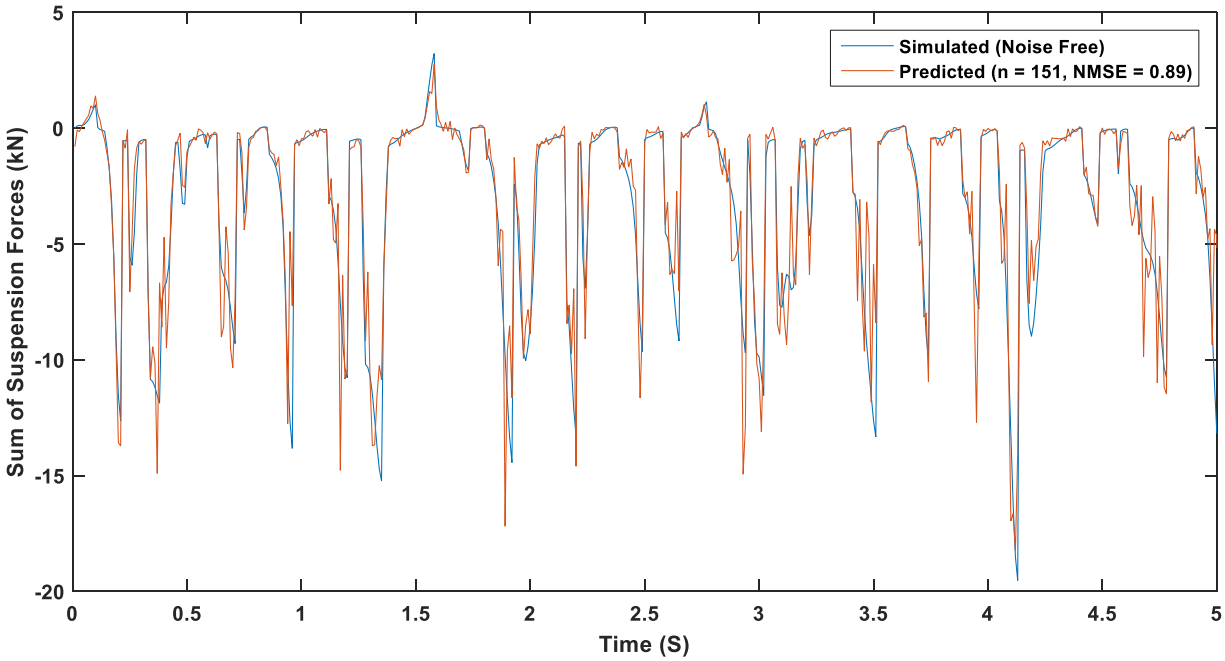

Figure 4.27: The stochastic model can closely predict the underlying behavior of the noisy measured system. Input signals shown in Figure 4.25 are used to train the stochastic model

5 Conclusions and Future Work

5.1 Introduction

In this chapter, we will provide a brief summary of the research that has been presented in this document, make conclusions regarding the findings, and provide suggestions for areas that can be further improved.

To summarize, a detailed multibody dynamic model of the three-piece truck commonly used by the freight railroad industry was developed. The development of the model brought three main problems with efficiently executing such tasks to the researcher's view: 1) developing detailed MBD models is difficult and time consuming, 2) finding the correct model parameters proved to be challenging as well, and 3) the resulting model is computationally expensive.

This research focused on addressing these problems using a stochastic modeling technique. In scenarios where laboratory measured data is available, the data can be directly integrated into the MBD model, thus eliminating the need for developing detailed MBD models and finding model parameters. In cases where such data does not exist, parts of the MBD model can be replaced with the stochastic model to reduce the computational expense of the MBD model.

Since the systems used for the purpose of this research were simplified, the improved computational efficiency gained by replacing the deterministic force elements with stochastic predictions were not significant. The significance of using such techniques will be magnified when dealing with much more computationally intensive models such as a train consist with hundreds of railcars.

In practice, the stochastic model can be used, along with simulation data from deterministic MBD models, to address the third problem mentioned above and reduce the computational expense of MBD models or it can be used with laboratory acquired data to address the first two problems. Using the laboratory measured data, the process of developing accurate MBD models of complex dynamic systems can be significantly simplified.

5.2 Conclusions

In conclusion, the stochastic modeling technique proved to be an efficient method for learning the behavior of dynamic systems. It can be used to replace computationally expensive subsystems of deterministic MBD models to increase the efficiency of the models without reducing their accuracy. Even though the reduction in the computational expense for the simple case studies investigated here are not significant, for more complex dynamic systems like a train consist the difference is expected to be substantial. The algorithm, along with signal processing techniques, was able to learn the behavior of physical systems and integrate measured data within MBD models. This proved to be an accurate alternative to system identification, especially in cases where the dynamic system shows strong nonlinear behavior.

The algorithm was also used to estimate system parameters for a single suspension model. Results indicated that using response surface methods is an accurate alternative to traditional system identification techniques used for estimating parameter values of MBD models.

The number of training points and the selection of the most space-filling sampling plan are the most important parts of the stochastic modeling technique. Latin hypercube sampling plans

developed in the previous chapter were shown to be inefficient due to the nature of the problem being investigated. A new sampling plan with modified feasible region has been developed to address this problem. Additional number of sample points improved the accuracy of the predictions, and multiple criteria have been developed to improve the efficiency of selecting additional training data.

The accuracy of the predictions can also be improved by using a low-pass filter to eliminate the high frequency content of the output signal. The only downside of using the LPF is that it would eliminate some of the dynamics of the system, and it should not be used in studies where such dynamic behaviors are an important part of the analysis.

5.3 Future Work

Although the results presented here have verified the effectiveness of the stochastic modeling approach, the technique can be further improved in a number of ways:

1. The sampling used for the selection of the training data had a significant influence on the efficiency of the model, further research can be directed at developing more efficient sampling plans.
2. As presented in Figure 4.10 and Figure 4.16, there is some variation associated with stochastic models with the same number of training data. Future work should focus on the underlying causes of these variations, thereby improving the performance of the model.

3. The researcher's experience has indicated that with the current state of the algorithm, it is computationally impractical to use training sets with more than 500 data points. While this is sufficient for the dynamic system studied here, future research should be directed at reducing the computational expense of running the algorithm with a large number of training data.
4. A Kriging correlation function was used for the purpose of this study, but the influence of other Radial Base Functions (RBF) should be studied.
5. Infill criteria developed here are effective in improving the efficiency of the stochastic model, and further research can be focused on developing more efficient infill criteria.
6. Using the stochastic model to integrate laboratory measured data into MBD models proved to be an effective alternative for system identification. Further research should focus on improving the performance of the stochastic model in the presence of noise.
7. The algorithm was used to estimate model parameters for the single suspension case study, and further research should focus on improving the stochastic model to be used as an alternative to system identification.
8. Developing a method for quantifying the gained computational efficiency of using the stochastic modeling technique.

6 Bibliography

[1] S. Iwnicki, *Handbook of Railway Vehicle Dynamics*. CRC Press, 2006.

[2] V. K. Garg, *Dynamics of Railway Vehicle Systems*. 1984.

[3] J. Kalker, "Wheel-Rail Rolling Contact Theory," *Wear*, 1991.

[4] J. Kalker, "A Fast Algorithm for the Simplified Theory of Rolling Contact," *Veh. Syst. Dyn.*, vol. 11, no. 1, 1982.

[5] J. J. Kalker, *Three-Dimensional Elastic Bodies in Rolling Contact*. Springer Netherlands, 1990.

[6] F. Xia, "The dynamics of the three-piece-freight truck," Technical University of Denmark, 2002.

[7] P. E. Klauser, "Modeling Friction Wedges, Part 1: An Improved Model," in *ASME International Mechanical Engineering Congress and Exposition*, 2004.

[8] P. E. Klauser, "MODELING FRICTION WEDGES, PART II: AN IMPROVED MODEL," in *ASME International Mechanical Engineering Congress and Exposition*, 2004.

[9] J. F. Gardner and J. P. Cusumano, "Dynamic models of friction wedge dampers," pp. 65–69, 1984.

[10] B. Ballew, B. J. Chan, and C. Sandu, "JRC2008-63055," pp. 1–11, 2008.

[11] P. Venkataraman, *Applied Optimization with MATLAB Programming*. Wiley, 2009.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[13] D. R. Jones, "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *J. Glob. Optim.*, vol. 21, no. 4, pp. 345–383, 2001.

[14] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008.

[15] F. Xia and H. True, "ON THE DYNAMICS OF THE THREE-PIECE-FREIGHT TRUCK," in *IEEE/ASME Joint Rail Conference*, 2003, pp. 149–159.

[16] N. K. Chandiramani, K. Srinivasan, and J. Nagendra, "Experimental study of stick-slip dynamics in a friction wedge damper," *J. Sound Vib.*, vol. 291, no. 1–2, pp. 1–18, Mar. 2006.

[17] A. B. Kaiser, J. P. Cusumano, and J. F. Gardner, "Modeling and Dynamics of Friction Wedge Dampers in Railroad Freight Trucks," *Veh. Syst. Dyn.*, vol. 38, no. 1, pp. 55–82, Jul. 2002.

[18] "Industrial Sector: Rail." [Online]. Available: http://www.simpack.com/industrial_sectors_rail.html.

[19] W. Zhai and K. Wang, "Lateral Hunting Stability of Railway Vehicles Running on Elastic Track Structures," *J. Comput. Nonlinear Dyn.*, vol. 5, no. 4, p. 041009, 2010.

[20] B. J. Sperry and R. West, "Complex Bogie Modeling Incorporating Advanced Friction

Wedge Components By," 2009.

[21] S. Andersson, A. Soderberg, and S. Bjorklund, "Friction models for sliding dry, boundary and mixed lubricated contacts," *Tribol. Int.*, vol. 40, pp. 580–587, 2007.

[22] H.-K. Hong and C.-S. Liu, "Coulomb Friction Oscillator: Modelling and Responses To Harmonic Loads and Base Excitations," *J. Sound Vib.*, vol. 229, no. 5, pp. 1171–1192, Feb. 2000.

[23] A. J. McMillan, "A NON-LINEAR FRICTION MODEL FOR SELF-EXCITED VIBRATIONS," *J. Sound Vib.*, vol. 205, no. 3, pp. 323–335, 1997.

[24] Nguyen B. Do, "MODELING OF FRICTIONAL CONTACT CONDITIONS IN STRUCTURES," 2005.

[25] T. Piatkowski, "Dahl and LuGre dynamic friction models — The analysis of selected properties," *Mech. Mach. Theory*, vol. 73, pp. 91–100, Mar. 2014.

[26] E. A. H. Vollebregt, "User guide for CONTACT, Vollebregt & Kalker's rolling and sliding contact model," 2013.

[27] F. Carter, "On the Action of a Locomotive Driving Wheel," *Proc. R. Soc. London*, vol. 112, no. 760, pp. 151–157, 1926.

[28] J. P. Vermeulen and L. K. Johnson, "Contact of Nonspherical Elastic Bodies Transmitting Tangential Forces," *Appl. Mech*, pp. 338–340, 1964.

[29] H. Hertz, "Über die Berührung fester, elastischer Körper," *J. für die reine und Angew. Math.*, pp. 156–171, 1881.

[30] L. P. Sun, J. Zhang, and J. Zhang, "Analyses on Wheel-Rail Contact Using Finite Element Method during Passing through Curved Track."

[31] J. J. Kalker, "On the rolling contact of two elastic bodies in the presence of dry friction," Delft, 1967.

[32] A. Keylin, "ANALYTICAL EVALUATION OF THE ACCURACY OF ROLLER RIG DATA FOR STUDYING CREEPAGE IN RAIL VEHICLES," 2012.

[33] J. Ayasse and H. Chollet, "Wheel-Rail Contact," in *Handbook of Railway Vehicle Dynamics*, 2006.

[34] A. Shabana, K. Zaazaa, and H. Sugiyama, *Railroad Vehicle Dynamics: A Computational Approach*. CRC Press, 2007.

[35] Q. Wu, C. Cole, M. Spiryagin, and Y. Q. Sun, "A review of dynamics modelling of friction wedge suspensions," *Veh. Syst. Dyn.*, vol. 52, no. 11, pp. 1389–1415, Aug. 2014.

[36] Y. Q. Sun and C. Cole, "Finite Element Modeling and Analysis of Friction Wedge Damping During Suspension Bounce Modes," *J. Vib. Acoust.*, vol. 131, no. 5, p. 054504, 2009.

[37] L. Baruffaldi and A. A. dos S. Jr., "EFFECTS OF NONLINEAR FRICTION WEDGE DAMPING ON FREIGHT TRAIN DYNAMICS," *Int. Mech. Eng. Congr. Expo.*, pp. 1–8, 2010.

[38] Q. Wu, C. Cole, M. Spiryagin, and Y. Q. Sun, "A review of dynamics modelling of friction wedge suspensions," *Veh. Syst. Dyn.*, no. August 2014, pp. 1–27, Aug. 2014.

[39] "Standard Car Truck Company Manuals - Barber M-976 AAR Spring Groups." [Online]. Available: http://www.sctco.com/pdf/Section3.pdf.

[40] H. Wu and J. Robeda, "EFFECTS OF BOGIE CENTER PLATE LUBRICATION ON VEHICLE CURVING AND LATERAL STABILITY," *Dyn. Veh. Roads Tracks Proc. 18th IAVSD Symp. held Kanagawa, Japan August 24-30, 2003 Suppl. to Veh. Syst. Dyn.*, vol. 41, pp. 292–301, 2003.

[41] W. P. O. Donnell, M. Enterprises, and P. B. Aspengren, "Constant contact side bearing set-up height adjustment and the need for level track," pp. 19–23, 2006.

[42] U. C. 518:2009, "Testing and approval of railway vehicles from the point of view of their dynamic behaviour - Safety - Track fatigue - Ride quality," *Paris Int. Union Railw.*, 2009.

[43] Y. H. Tse, V. K. Garg, S. P. SINGH, and L. Allen, *VALIDATION OF FREIGHT CAR HUNTING (NONLINEAR/LINEAR) MODEL*. 1979.

[44] P. Shahidi, D. Maraini, B. Hopkins, and A. Seidel, "Estimation of Bogie Performance Criteria Through On-Board Condition Monitoring," *Int. J. Progn. Heal. Manag.*, pp. 1–10, 2014.

[45] N. A. C. Cressie, *Statistics for spatial data*, 2, revised. J. Wiley, 1993.

[46] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*. Academic Press, 1998.

[47] J. Aldrich, "R.A. Fisher and the making of maximum likelihood 1912-1922," *Stat. Sci.*, vol. 12, no. 3, 1997.

[48] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[49] M. Johnson, L. Moore, and Ylvisaker D., "Minimax and maximin distance designs," *J. Stat. Plan. Inference*, vol. 26, pp. 131–148, 1990.

[50] M. D. Morris and T. J. Mitchell, "Exploratory designs for computational experiments," *J. Stat. Plan. Inference*, vol. 43, pp. 381–402, 1995.

[51] D. G. Krige, "A statistical approach to some basic mine valuation problems on the Witwatersrand," *J. Chem. Metall. Min. Soc. South Africa*, 1951.

[52] G. Matheron, "Principles of geostatistics," *Econ. Geol.*, vol. 58, no. 8, pp. 1246–1266, 1963.

[53] D. R. Jones, M. Schonlau, and J. William, "Efficient Global Optimization of Expensive Black-Box Functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[54] H. Theil, *Principles of Econometrics*. Wiley, 1971.

[55] P. S. P. Chavan, P. S. H. Sawant, and J. A. Tamboli, "Experimental Verification of Passive Quarter Car Vehicle Dynamic System Subjected to Harmonic Road Excitation with Nonlinear Parameters .," *IOSR J. Mech. Civ. Eng.*, pp. 39–45.

[56] J. E. Nash and J. V Sutcliffe, "River flow forecasting through conceptual models part I — A discussion of principles," *J. Hydrol.*, vol. 10, no. 3, pp. 282–290, 1970.

[57] C. E. Rasmussen, C. K. I. Williams, and NetLibrary Inc., "Gaussian processes for machine learning," *Adapt. Comput. Mach. Learn.*, p. xviii, 248 p., 2006.

[58] E. R. Andersen, C. Sandu, and M. Kasarda, “Multibody Dynamics Modeling and System Identification for a Quarter-Car Test Rig with McPherson Strut Suspension Multibody Dynamics Modeling and System Identification for a Quarter-Car Test Rig with McPherson Strut Suspension,” Virginia Tech, 2007.

[59] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*. Springer, 1996.

[60] R. Kan and T. G.T., “Stochastic global optimization methods part I: Clustering methods,” *Math. Program.*, vol. 39, no. 1, pp. 27–56, 1987.

[61] A. Törn and A. Zilinskas, *Global Optimization (Lecture Notes in Computer Science)*. Springer, 1989.

[62] C. A. Coello, “A Survey of Constraint Handling Techniques used with Evolutionary Algorithms.”

[63] B. G. . Caraenen, A. . Eiben, and E. Marchiori, “How to Handle Constraints with Evolutionary Algorithms.”

[64] P. Gray, W. Hart, L. Painton, C. Phillips, M. Trahan, and J. Wagner, “A Survey of Global Optimization Methods,” *Sandia National Laboratories*. [Online]. Available: http://www.cs.sandia.gov/opt/survey/.

[65] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” *J. Chem. Phys.*, vol. 21, no. 6, p. 1087, 1953.

[66] B. D. Hughes, *Random Walks and Random Environments: Random walks*, no. v. 1. Clarendon Press, 1995.

[67] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.

[68] I. Rechenberg, *Cybernetic solution path of an experimental problem, Translation 1122*. Royal Aircraft Establishment Library.

[69] G. Mitsuo, “Genetic Algorithms and Their Applications,” in *Springer Handbook of Engineering Statistics*, P. Hoang, Ed. Springer London, 2006, pp. 749–773.

[70] L. C. W. Dixon and G. P. Szegö, *Towards global optimisation 2*. North-Holland Pub. Co., 1978.

[71] A. Corana, M. Marchesi, C. Martini, and S. Ridella, “Minimizing Multimodal Functions of Continuous Variables with the ‘Simulated Annealing’ algorithm,” *ACM Trans. Math. Softw.*, vol. 13, no. 3, pp. 262–280, Sep. 1987.

[72] M. Molga and C. Smutnicki, “Test functions for optimization needs,” 2005.

Appendices

Appendix A: Stochastic predictor example problems

In this section, we will test the algorithm that was built based on the mathematical concepts described in section 3.4 to estimate the value of well-known functions used to test optimization algorithms. We will compare the results for the location and the function value of the optimum point ($\boldsymbol{x}^{\dagger}, y^{\dagger}$), and also compare the overall landscape of the functions when possible.

A.1 Estimation of the Branin Function

The Branin function is a two-dimensional function with three global minima; the function is widely used to test global optimization algorithms. Equation (A.1) shows the description of the function and the domain in which it is evaluated. Figure A.1 shows the overall landscape of the function.

$$f(\boldsymbol{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)\cos(x_1) + s \quad , x_1 \in [-5,10], x_2 \in [0,15] \tag{A.1}$$

where the typical values for its parameters are $a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, r = 6, s = 10, \text{and } t = \frac{1}{8\pi}$. The global minima of the function are located at $(-\pi, 12.275), (\pi, 2.275), and\ (9.42478, 2.475)$ with $f(\boldsymbol{x}^{\dagger}) = 0.397887$.

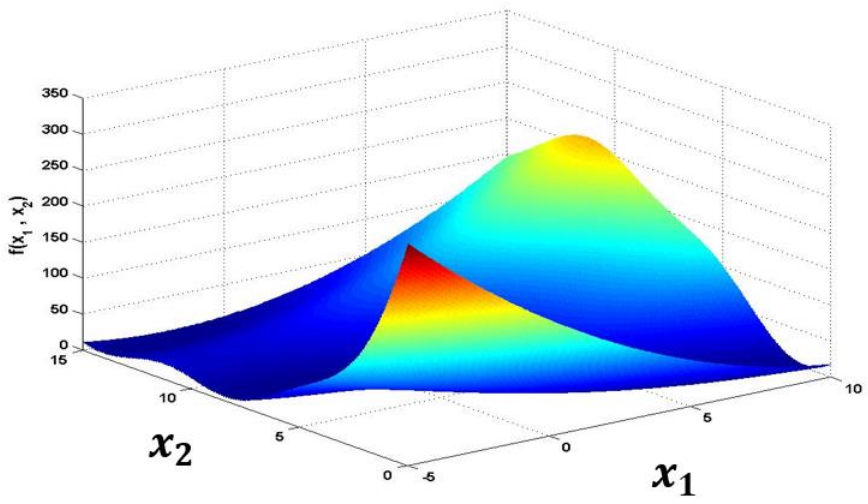

Figure A.1: Branin function

We will sample the function at 21 points using the Latin hypercube sampling plan and build the estimator function. Figure A.2 shows the comparison between the contour plot of the original function and the contour plot of the estimated function; the difference between the two plots can hardly be noticed.

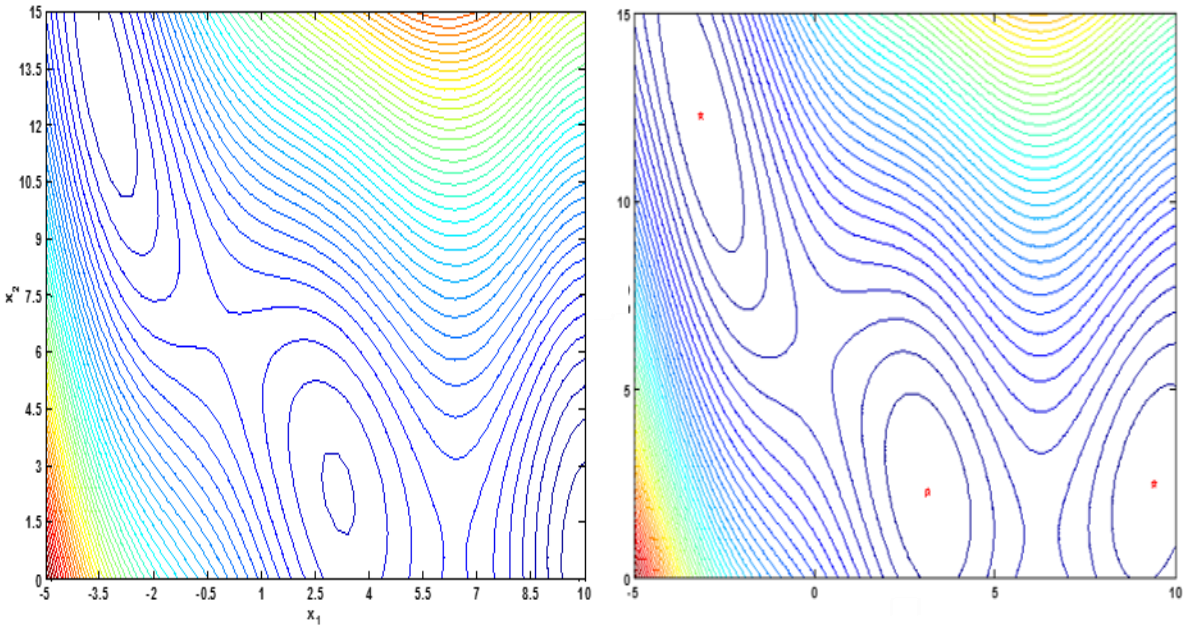

Figure A.2: Comparison between the original function (right) and the estimated function (left) for (n = 21)

Even with samples as few as 21, the two plots are almost identical; using genetic algorithm to find the global minimum of the predicted function yields $\hat{f}(x^{\dagger}) = -3.333$. Since the difference between the estimated minimum and the original function's minimum is bigger than our tolerance, we add this point to our sample data and re-run the estimator and continue this step until the difference is within our tolerance. Figure A.3 shows the convergence of the estimated function's global minimum to the function's global minimum after five additional data points are evaluated.

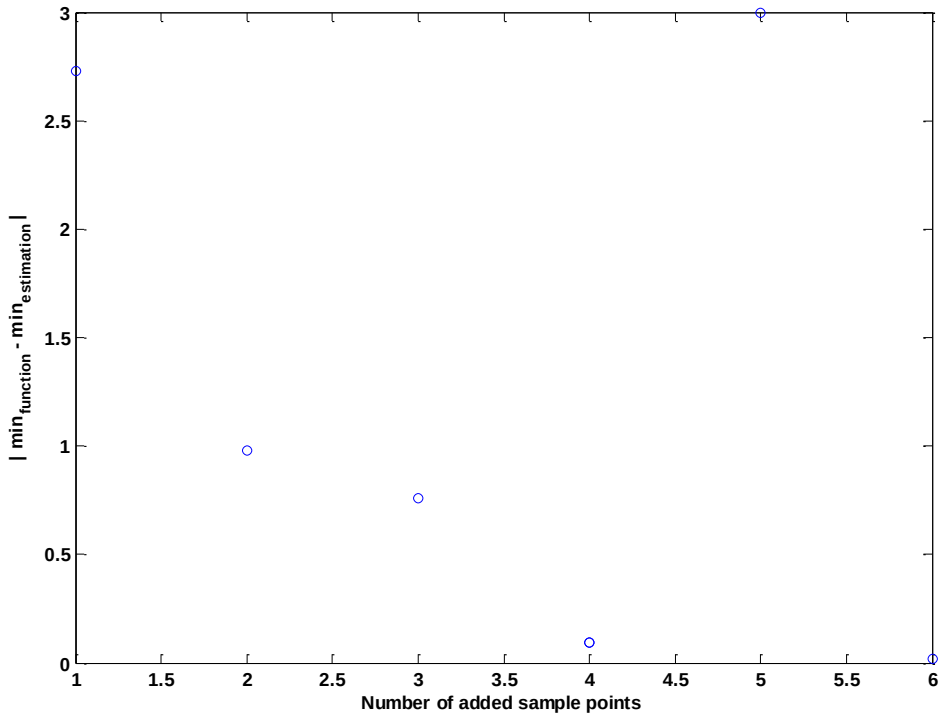

Figure A.3: Improving function estimations by adding points at the location of predicted global minimum

With the total of 26 function evaluations, the global minimum is estimated to be 0.4172 with $x^{*} = [\,9.4435\ 2.5785]$ which is within 5% of the original function's global minimum.

A.2 Hartmann 3-Dimensional Function

Hartmann's three-dimensional function is a more complicated function than the Branin function with $k = 3$. Since the function has 3 variables, we cannot use contour plots to qualitatively compare the results of the estimator function. Equation (A.2) shows the mathematical formulation of the problem.

$$f(x) = -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{3} A_{ij}\left(x_j - P_{ij}\right)^2\right) \tag{A.2}$$

where

$$A = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, P = 10^{-4}\begin{bmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix}, \alpha = [1.0\ 1.2\ 3.0\ 3.2]^T$$

The function is usually evaluated on the unit hypercube domain $x \in (0,1)^3$. The function has 4 local minima and one global minimum located at $(f(x^\dagger) = -3.86278, x^\dagger = [0.114614\ 0.555649\ 0.852547])$. We have evaluated the function at 10 sample points defined by the Latin Hypercube algorithm. Figure A.4 shows the convergence of the estimated global minimum to the original global minimum after only 3 more function evaluations with $(\hat{f}(\hat{x}^\dagger) = -3.8323, \hat{x}^\dagger = [0.1147\ \ 0\ 0.5469\ 0.8629])$.

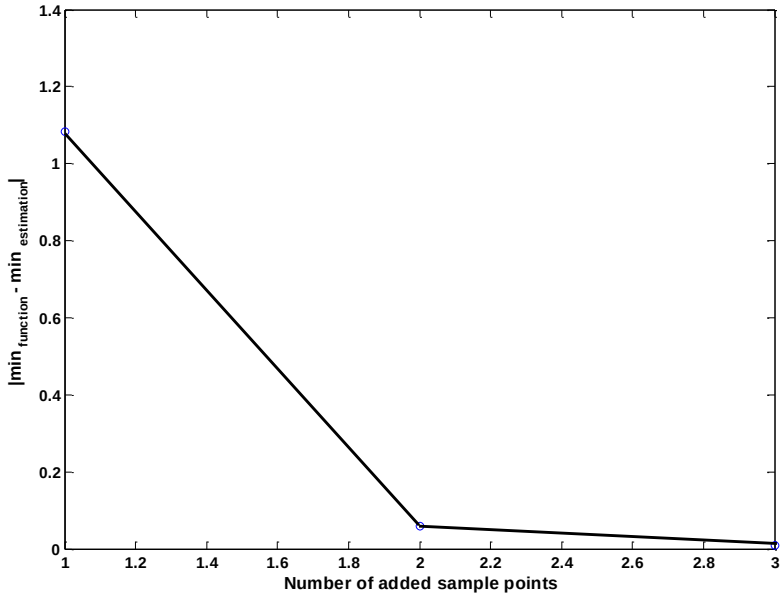

Figure A.4: Convergence of the estimator global minimum to the original function's global minimum

Appendix B: Global Optimization

B.1 Introduction

We use optimization in everyday tasks such as selecting the optimum route to work, or choosing the line to stand in at the grocery store. Engineering is filled with optimization problems, from designing a soda can that can hold a certain amount of liquid (constraint) while using the minimum amount of aluminum sheet (objective function) to finding the optimum ratio between air and fuel for an internal combustion engine that meets certain performance criteria while using the minimum amount of fuel. Optimization is a branch of applied mathematics that deals with finding the optimum value of an objective function when operating within a set of constraints that define the domain of the design variables. Equation (B.1) shows the standard definition of an optimization problem.

$$\begin{aligned} &minimize && f(\boldsymbol{x}) \\ &&& over\ \boldsymbol{x} \in \mathbb{R}^k \\ &subject\ to: \\ &&& \boldsymbol{h_i}(\boldsymbol{x}) = \boldsymbol{0} \quad i = 1 \cdots n \\ &&& \boldsymbol{g_i}(\boldsymbol{x}) \leq \boldsymbol{0} \quad i = 1 \cdots n \\ &&& \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b} \end{aligned} \tag{B.1}$$

where $f(\boldsymbol{x})$ is the objective function that we are trying to minimize, $\boldsymbol{x}$ is the vector of optimization variables, $\boldsymbol{h}'s$ are the equality constraints, $\boldsymbol{g}'s$ are the inequality constraints, and $\boldsymbol{a}, \boldsymbol{b}$ are the vectors for the lower and upper bounds of $\boldsymbol{x}$. Optimization problems can be divided into two main categories: 1) local optimization, and 2) global optimization.

Local optimization deals with finding a local minimum of the objective function. For complicated functions with multiple local minima, various starting points will result in various local optima that are not necessarily the minimum value that the objective function can take within its domain (global optimum). Some of the most popular local optimization algorithms are simplex [11] that deals with linear problems (LP) and Sequential quadratic Programing (SQP) [11] that deals with nonlinear programing (NLP). Global optimization algorithms are usually computationally expensive and require iterative procedures for finding the global minimum. Recent advancements in the computation power of personal computers have increased the popularity of global optimization to a point where the research done in the field of optimization is dominated by methods and applications of global optimization. Popular methods for global optimization includes Branch and bound [59], clustering methods[40][41], evolutionary algorithms [11], and simulated annealing [11]. In this section, we will discuss two of the most popular techniques for global optimization, Genetic algorithms (GA), and Simulated annealing (SA).

A.2 A Review of Global Optimization Techniques

A brief review of some of the most popular global optimization techniques is provided in this section. Readers interested in more information regarding each method are encouraged to use the cited material.

Branch and Bound is a general search technique. Consider the standard form of an optimization problem introduced by Equation (B.1), Branch and bound method requires the computation of a lower bound and an upper bound (not required for all of the sub-problems) on the original

problem or any of its sub-problems, and a procedure for dividing the feasible region to produce smaller sub-problems. We start by considering the root problem that covers the entire feasible region and calculate its lower and upper bounds. If the lower and upper bounds match or are within a specified range, the optimal solution has been found and we terminate the algorithm. Otherwise, two or more sub-problems are generated by dividing the feasible region into two or more subsystems that represent the entire feasible region. The algorithm iterates and generates a tree of sub-problems. The optimal solution of each sub-problem is a solution to the root problem but not necessarily the global optimum. The current best feasible solution computed previously can be used to prune branches where the lower bound is higher than the computed best solution. Iterations continue until all the branches have been solved or pruned, or a certain threshold for the number of iterations or the current best solution it met [59].

As mentioned earlier, the ability to converge to a global solution (as opposed to a local one) depends on the assumed initial condition. The simplest procedure for finding the global solution is to perform local optimization from a series of initial points distributed over the feasible region (multi-start). **Clustering methods** for global optimization are a type of multi-start method that carefully select the matrix of starting points to avoid identifying the same local minima, hence, increasing the efficiency of multi-start methods. The algorithm for the clustering methods can be explained by three main steps: 1) sample points in the feasible region, 2) transform the sample points to groups around each local minima, and 3) identify the groups that represent neighborhoods of local minima by applying a clustering technique. Clustering methods are most efficient when the design variables are less than a few hundred.

Evolutionary algorithms (EA) are nature-inspired search methods. They are based on the natural selection and survival of the fittest. Evolutionary algorithms search from a population of points instead of using only one point. The initial population is created using random selection from the feasible region. Evolutionary methods usually only handle bound constraints on the variables. In each generation (iteration) weak genes die out and the next generation is a combination of the "fittest" genes, this is usually achieved by swapping parts of the genes. Genes can also be "mutated", by making random changes to an existing solution or adding a random gene to the population. The percentage of mutated population and "recombined" population is varied so in the beginning there are more muted genes to better sample the entire design space and as the number of generations increase, the percentage of muted population decreases. Mutation and recombination of the population genes biases the solution towards regions of the design space that the probability of finding the global optimum is higher. Figure B.1 shows a simplified flowchart for an evolutionary optimization algorithm.

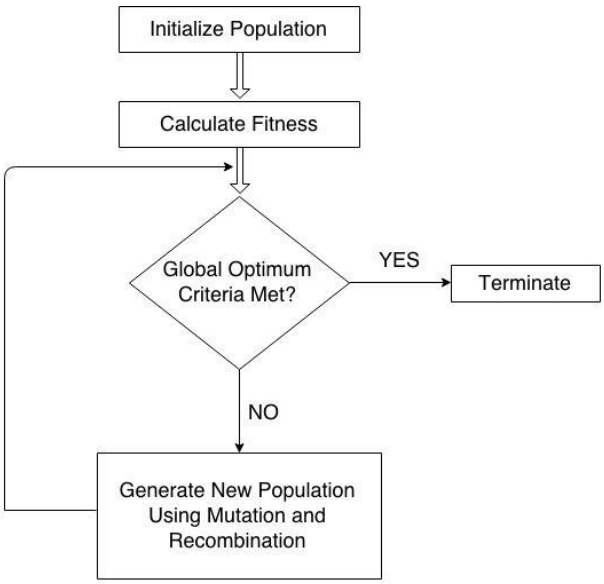

Figure B.1: Flowchart for a typical Evolutionary Algorithm (EA) global optimizer

There are several different type of evolutionary global optimizers that differ in their strategy for generating the new population based on the results from previous iterations. These include: 1) genetic programming (GP), that evolve programs, which makes them suitable for problems where determination of function is required 2) evolutionary programming (EP), that emphasizes on optimizing continuous functions without recombination, 3) evolutionary strategies (ES), that optimizes continuous functions with recombination (ES and EPs are best for optimizing continuous functions), and 4) genetic algorithms (GAs) that uses general combinatorial problems for optimizing. Almost all EAs are developed for unconstrained optimization problems, some methods for handling constraints have been proposed [62],[63]. In the next section we will discuss Genetic Algorithms (GAs) in detail. Genetic Algorithms are the most popular method among evolutionary algorithms and are suitable for optimizing combinatorial and continuous problems. The benefit of using evolutionary search methods for global optimization problems is that they can handle noisy functions with multiple local optima without getting stuck in the local optima and often find the global optimum of the function. The downside of using EAs is that the convergence to the global optimum is not guaranteed and as the number of iterations increase the probability of the current solution being the global optimum increases [64].

Since EAs rely on recombination operations, a challenging aspect of applying them is to represent the problem in a manner that interesting solutions are likely to be generated from the combination of two solutions. Evolutionary Algorithms have been successfully applied to optimization problems in several areas of science and engineering: engineering design, parameter fitting, knapsack problems, transportation problem, image processing, traveling

salesman, scheduling, and so on. As can be inferred from the list, a significant number of problems are from the area of discrete or combinatorial programming.

Simulated annealing gets its name from the process of controlled cooling of a material to change the size of its crystals (annealing). It is a generalization of a Monte Carlo method [65] and it is suggestive of a biased random walk [11]. The concept of slow cooling resembles the slow decrease in the probability of accepting worse solutions as the number of iterations increase. Allowing the algorithm to accept worse solution is a key feature of SAs that permits a more extensive search of the design space, by escaping local minima.

The original method developed by [65] choose an initial state for the thermodynamic system (E, T); by holding temperature (T) constant and perturbing the initial configuration, a new energy (E) for the system is calculate. If ΔE is negative the change is accepted and if it is positive, the change is accepted with the probability given by Boltzmann factor (Equation (B.2)).

$$P = e^{-dE/T} \tag{B.2}$$

where P is the probability of accepting a worse solution. This step is repeated until a solution is accepted, then the temperature is reduced and everything is repeated for the new temperature until frozen state is achieved $(T = 0)$. As mentioned earlier, the possibility of accepting a worse solution allows the algorithm to escape local minima but as the number of iterations increase this possibility should be reduced so that the solution does not wonder all over the design space and cancel previously achieved improvements [11][66]. Applying this procedure to optimization problems are straight forward. The current state of the thermodynamic system (E, T) is analogous to the current solution of the optimization problem where energy (E) represents the

objective function value. The biggest problem with implementing SA is how fast or how slow should the possibility of accepting a worse solution decrease, which is known as the annealing schedule and it differs based on the problem. Constructing an algorithm that does not depend on these user specified control parameters is an ongoing research topic. Section B.4 contains detailed description of a simulated annealing algorithm.

Random walk (RW) also known as drunkard's walk, are a special case of undirected, local search methods where the next candidate solution to be evaluated is generated randomly from the current solution [66]. RWs can be seen as EAs with initial population size of one and in each step one offspring is reproduced (i.e. by mutation). The new offspring always replaces the current solution regardless of improvement or deterioration.

B.3 Genetic Algorithms

The term genetic algorithm was first introduced by John Holland, in his "Adaptation in Natural and Artificial Systems" book published in 1975 [67]. Work on evolutionary computation goes even further back to 1960s [68]. Recent advancements in computation power and cost of computers has increased the interest in genetic algorithms and expanded their applications to various fields from engineering design to scheduling problem in the transportation industry[69]. In this section we will describe the GA used for the purpose of this study in more details and test the accuracy of our algorithm on various global optimization test problems[70]. As mentioned earlier the concept is based on the Darwinian theory of natural selection and survival of the fittest. The natural process takes place through constant mutation and recombination of the

chromosomes in the population to yield a better gene structure. All of these terms appear in the discussion related to evolutionary computation.

GAs use stochastic information in their algorithm and hence are exceptionally useful for global optimization problems especially ill-behaved, discontinuous, and non-differentiable problems. GAs have difficulty handling continuous problems due to the fact that they use population of solutions or a design vector (X); the problem can be clarified with an example, for a genetic algorithm values like 2.1 and 2.101 are distinct and the algorithm cannot reject small changes like this in the design vector and recognize that they are the same solution with different degrees of precision. We will discuss the nature inspired terminology and then create a pseudo-algorithm.

Chromosomes can be related to the design vector (X), some algorithms work with the actual design vector and others use a mapping, for instance a binary mapping to encode X. Chromosomes are made of smaller pieces called allele that are in fact different members of the design vector X.

Fitness is related to the objective function $f(X)$, where the population is ranked/sorted based on their objective function value. Selection relates to choosing a portion of the ranked population as parents to generate the next generation based on their chromosomes. There are various ways for selecting parents for the next generation. In one scheme known as the tournament scheme, a fraction of the high-ranking individuals are chosen for reproduction and the rest of the population is filled with new immigrants. A modified version of the above scheme suggests assigning probability to selection of the parent genes, where the higher ranking individuals have

a higher probability of being selected; some of the popular probability functions used for selecting parents are roulette wheel, elitist selection, linear, or geometric ranking [12].

Genetic operations, relates to the ways of defining the new population based on the genes of the selected individuals from the previous generation. This is what causes the genes to evolve and form new and superior solutions that would eventually converge to the global optimum (fittest individual). Two main types of genetic operations exist, crossover or recombination, and mutation. Crossover operation has many different types; a simple crossover is where a portion of each parent's chromosomes are swapped to form children. Location for the partitioning is also defined randomly. Figure B.2 shows a simple crossover operation. Other types of crossover include arithmetic crossover and a direction-based crossover. Equations (B.4, B.5) shows the mathematical definition for different types of crossover operations.

$$u_i = \begin{cases} x_i & i < r \\ y_i & else \end{cases}$$
$$v_i = \begin{cases} y_i & i < r \\ x_i & else \end{cases} \tag{B.3}$$

where r is a random number that specifies the location for swapping, x_i, y_i are members of the parent design vectors (X, Y), and u_i, v_i are members of the children design vectors (U, V).

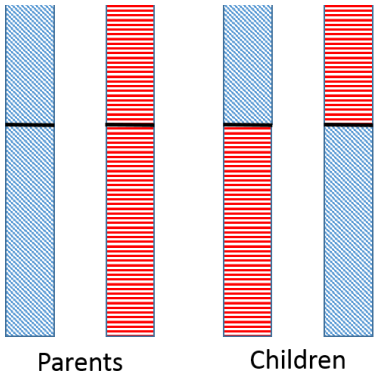

Figure B.2: Schematics of a simple crossover operation, the black line indicates the random location defined by r (Equation (B.3))

Arithmetic crossover uses a linear combination of parent genes to create children.

$$C_1 = \lambda_1 X + \lambda_2 Y$$
$$C_2 = \lambda_2 X + \lambda_1 Y \tag{B.4}$$

where λ_1, λ_2 are random numbers. A direction-based crossover uses the objective function value of parent genes to direct the search towards the superior parent. If the objective function values of design vectors X, Y are $f(X), F(Y)$ and $F(Y) < F(X)$, and r^* is a randomly generated number, then children are obtained using Equation (B.5).

$$C = r^*(Y - X) + Y \tag{B.5}$$

Mutation is the replacement of a single element of individuals with a random number, the element is also randomly chosen. If X is the design vector for an individual and $k\ and\ x_k$ are random numbers then Equation (B.6) describes the mutation of a gene:

$$C = \begin{cases} x_i & i \neq k \\ x_k & i = k \end{cases} \tag{B.6}$$

An important issue when implementing GA is the number of initial population, which usually remains the same for all generations. It can be difficult to establish how many members must be considered to represent the population in every generation. As far as the initial generation is concerned, using normal random set of the design space is recommended. Some versions of GA dope the initial population by adding known good solutions (if they exist).

The process of searching for the global optimum can be broadly divided into exploration and exploitation. Exploitation refers to using the characteristics of the current population to focus on the portion of the design space where the probability of finding the global optimum is higher, techniques described earlier all fall into the category of exploitation. Exploration refers to searching across the design space by adding unbiased random population from the entire design space to the current generation. Immigration is a technique that handles the exploration aspect of GAs. In some versions of GA immigrants are allowed to breed with the current population before the evaluation section of the code.

B.4 Simulated Annealing

As mentioned earlier, simulated annealing is a variation of the random walk method [11]. The basic idea behind SA that differentiates it from random walk is the change in the probability of accepting a worse solution as the number of iterations increase. There are many different versions of the algorithm that are different in the way they assign the probability of accepting a worse solution [11], [71]. Equation (B.7) is the function for the probability of accepting a worse solution.

$$p = e^{-\beta \Delta f}$$
$$\beta = -k/T \tag{B.7}$$

where β is related to the Boltzmann probability distribution, Δf is the change in the objective function value from the previous step, k is Boltzmann's constant, and T is the annealing temperature. k, T_0 are user specified parameters that control the annealing schedule. Development of SA algorithms that are independent of user specified parameters is an ongoing research [11]. Corana et al., updates the value for β in each time step to achieve a 1:1 ratio between rejected and accepted trials [71]. Venkataraman suggests a β that yields a probability of $p = 0.7$ for a Δf equal to half of f_0 [11].

$$\beta = -\frac{\log(0.7)}{0.5 f(x_0)} \tag{B.8}$$

In this section a pseudo-code for the method is presented.

Step 0:

Choose initial point x_0, and calculate $f_0 = f(x_0)$

Step 1:

Choose a random point on the a unit n-dimensional hyper-sphere to calculate the search direction (S)

Step 2:

Calculate $f_1 = f(x_0 + \alpha S)$ and $\Delta f = f_1 - f_0$, where α is the step size

Step 3:

$if\ \Delta f \leq 0\ then\ p = 1$
$Else\ p = e^{-\beta \Delta f}$

Step 4:

Generate random number r where $0 \leq r < 1$
$if\ r \leq p$ accept the step and update the optimum design
$else$ reject the step
$Go\ to$ step 1

With reasonably large number of iteration the algorithm converges to the global optimum. A glance at the algorithm indicates that the step size (α) is also an important parameter that can influence the convergence rate for the algorithm; golden-section method [11] is used for the calculation of α.

B.5 Example Problems

In this section we will test the developed algorithms for SA and GA on a number of global optimization problems with multiple local minima to compare the accuracy and efficiency of each algorithm. The literature is reach with global optimization sample problems [72], we have selected a few 2D problems for better visualization of the solution.

The first problem is a simple function defined by Equation (B.9) and illustrated in Figure B.3. As can be seen in Figure B.3 the function has multiple local minima and a strong global minimum located at $x_1^* = 4, x_2^* = 4$ with $f(x_1^*, x_2^*) = -19.9667$.

$$f(x_1, x_2) = -20\frac{\sin\left(0.1 + \sqrt{(x_1 - 4)^2 + (x_2 - 4)^2}\right)}{0.1 + \sqrt{(x_1 - 4)^2 + (x_2 - 4)^2}} \tag{B.9}$$

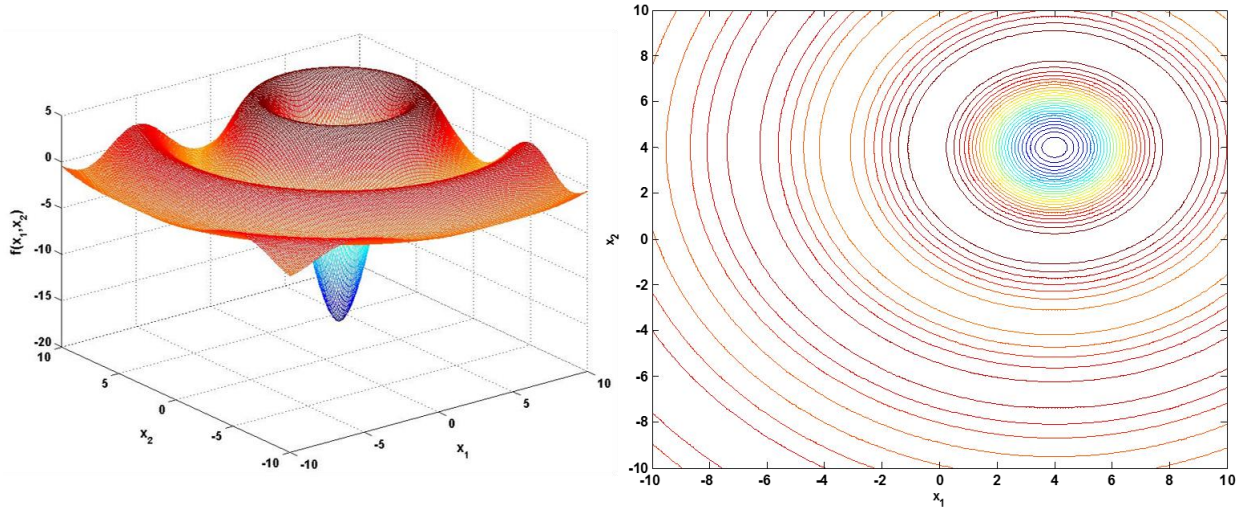

Figure B.3: The function contains multiple local minima with the global minimum at $x_1 = 4$, $x_2 = 4$

Figure B.4 shows the solution using the SA algorithm. The solution wonders around at the beginning when the temperature (T) is high and as the temperature gradually reduces allowing less worst solutions to be accepted, solution stays in the region of the global minimum until the stopping criteria is met. As mentioned earlier β, T_0 are important parameters that define the annealing schedule. Table B.1 shows the effect of β, T_0 on the efficiency of the SA method.

Figure B.5 shows the convergence of the solution to the global minimum using the genetic algorithm, population size is an important parameter that can influence the efficiency of GA.

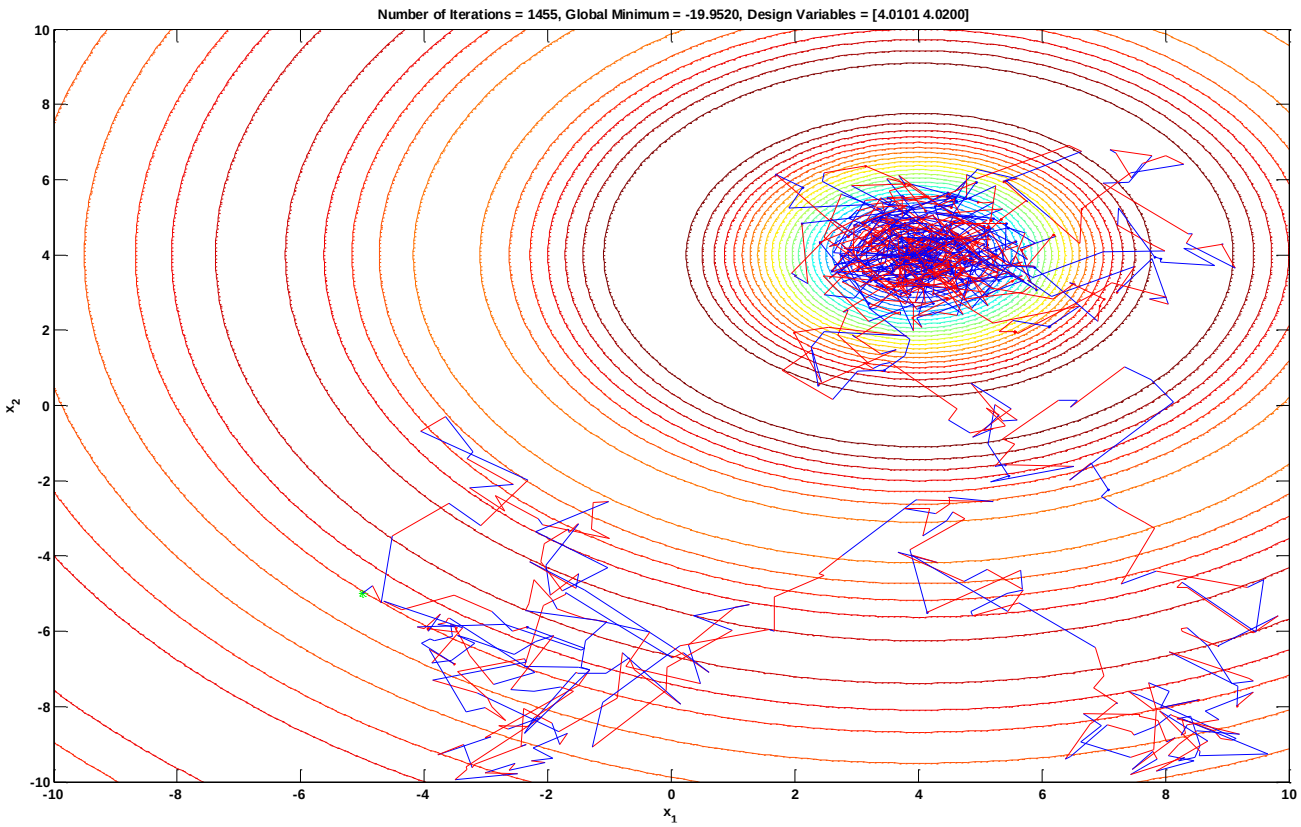

Figure B.4: Solution found using simulated annealing algorithm after 1455 iterations

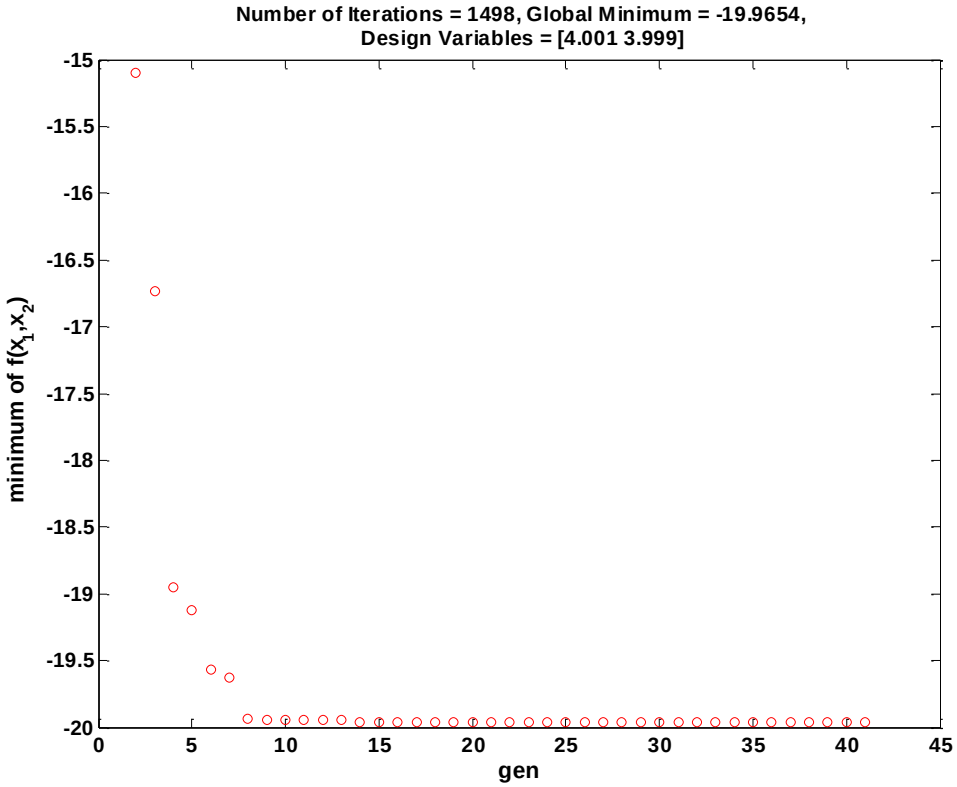

Figure B.5: Solution to example 1 using genetic algorithm

Table B.1: Comparison of the effect of user defined parameters on the efficiency of the simulated annealing algorithm

$\boldsymbol{\beta}$	$\boldsymbol{T_0}$	**Number of iterations (n)**	$\boldsymbol{x_1^*}$	$\boldsymbol{x_2^*}$	$\boldsymbol{f(x_1^*, x_2^*)}$
-70	1000	3215	3.9767	4.0156	-19.9454
-80	1000	5545	3.9985	3.9941	-19.9625
-90	1000	Did not converge to the global minimum			
-70	2000	2630	3.9971	3.9765	-19.9491
-80	2000	2657	3.9982	3.9873	-19.9576
-90	2000	4617	4.0050	4.0048	-19.9619

For the second example we have chosen Matlab's peak function (Equation (B.10)), as can be seen in Figure B.6 the function contains multiple local minima with a global minimum located at $x_1^* = 0.228, x_2^* = -1.626$ with $f(x_1^*, x_2^*) = -6.5511$. Figure B.7 and Figure B.8 show the convergence history of the SA and the genetic algorithm respectively.

$$f(x_1, x_2) = 3(1-x_1)^2 e^{-((x_2+1)^2+x_1^2)} 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-(x_1^2+x_2^2)} - \frac{1}{3} e^{-((x_1+1)^2+x_2^2)} \tag{B.10}$$

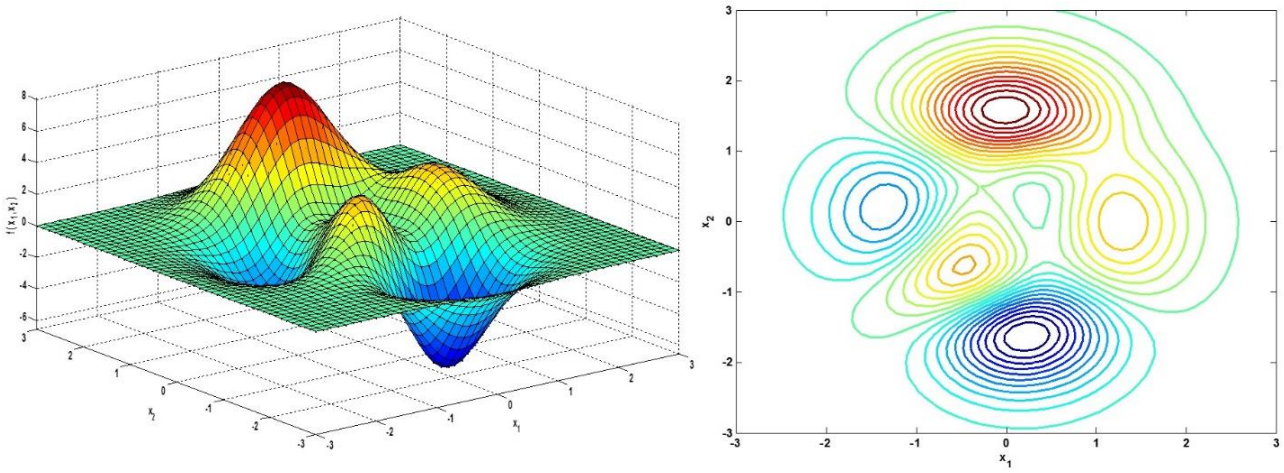

Figure B.6: Peaks function, global minimum located at $x_1^* = 0.228, x_2^* = -1.626$ *with* $f(x_1^*, x_2^*) = -6.5511$

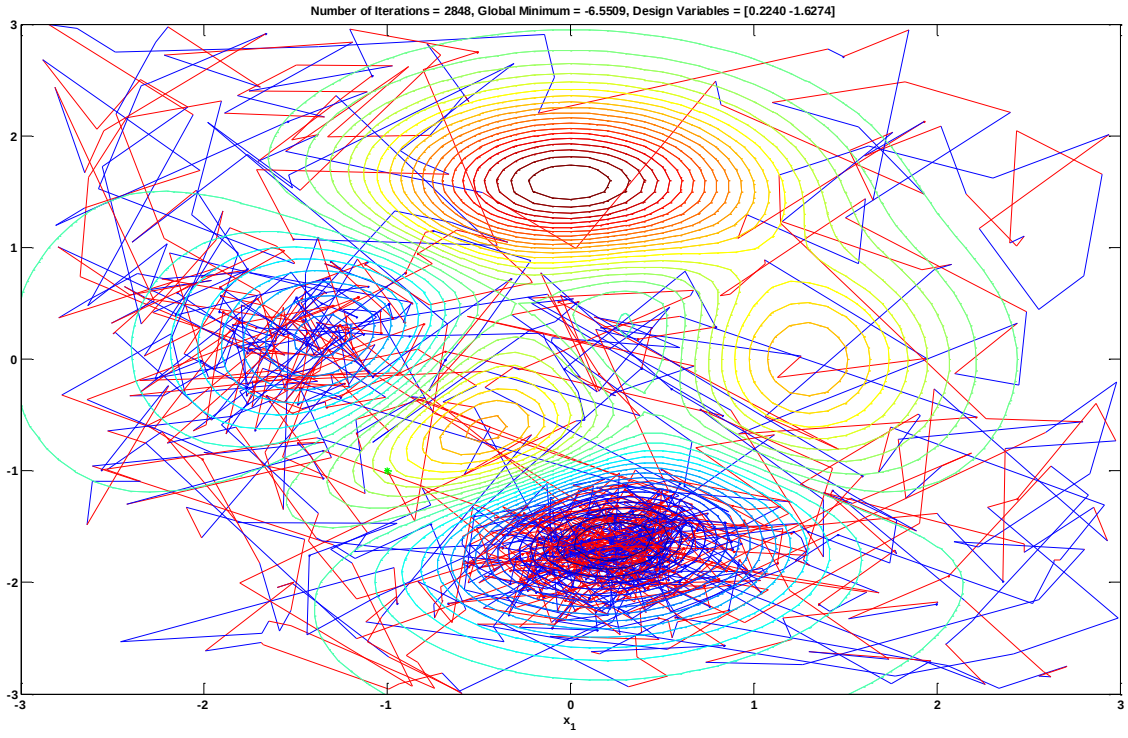

Figure B.7: Simulated annealing, global minimum found after 2843 iterations at $x_1^* = 0.224, x_2^* = -1.6274$ *with* $f(x_1^*, x_2^*) = -6.5509$

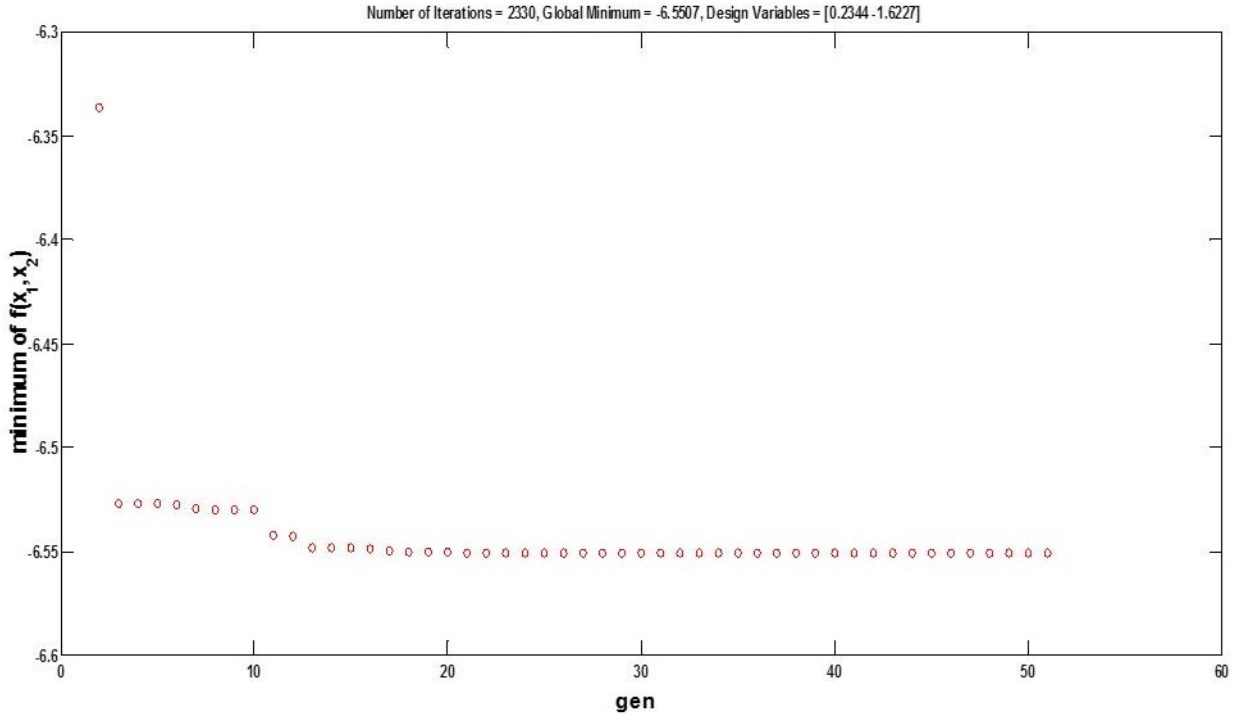

Figure B.8: Genetic Algorithm, global minimum found after 2330 iterations at $x_1^* = 0.234, x_2^* = -1.6227$ *with* $f(x_1^*, x_2^*) = -6.5507$