Approximations for Nonlinear Differential Algebraic Equations to Increase Real-time Simulation Efficiency

Gordon Kwong

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

Master of Science
In
Mechanical Engineering

Steve Southward, Committee Chair
Mehdi Ahmadian, Committee Member
Saied Taheri, Committee Member

May 5, 2010
Blacksburg, Virginia, USA

Keywords: Driving Simulator, Real Time Modeling, Suspension Modeling

Approximations for Nonlinear Differential Algebraic Equations to Increase Real-time Simulation Efficiency

Gordon Kwong

ABSTRACT

Full-motion driving simulators require efficient real-time high fidelity vehicle models in order to provide a more realistic vehicle response. Typically, multi-body models are used to represent the vehicle dynamics, but these have the unfortunate drawback of requiring the solution of a set of coupled differential algebraic equations (DAE). DAE's are not conducive to real-time implementation such as in a driving simulator, without a very expensive processing capability. The primary objective of this thesis is to show that multi-body models constructed from DAE's can be reasonably approximated with linear models using suspension elements that have nonlinear constitutive relationships.

Three models were compared in this research, an experimental quarter-car test rig, a multi-body dynamics differential algebraic equation model, and a linear model with nonlinear suspension elements. Models constructed from differential algebraic equations are computationally expensive to compute and are difficult to realize for real-time simulations. Instead, a linear model with nonlinear elements was proposed for a more computationally efficient solution that would retain the nonlinearities of the suspension. Simplifications were made to the linear model with nonlinear elements to further reduce computation time for real-time simulation.

The development process of each model is fully described in this thesis. Each model was excited with the same input and their outputs were compared. It was found that the linear model with nonlinear elements provides a reasonably good approximation of actual model with the differential algebraic equations.

Acknowledgements

First, I'd like to thank Dr. Steve Southward for providing me the opportunity to come to Virginia Tech to pursue my master's degree. Your guidance throughout these last two years has been invaluable. I'd also like to thank Dr. Mehdi Ahmadian and Dr. Saied Taheri for being on my committee and taking the time to provide comments and feedback on my thesis.

To my family, especially my mom, thanks for all the support throughout the years even though I can be a handful in extremely rare occasions.

I'd like to thank the graduate students at IALR, especially Shawn, who made my time in Danville fun. I enjoyed the laughs and drinks we shared. To my friends at IALR and CVESS, I appreciate the kindness that you showed me for the past two years. I wish nothing but the best for everyone in the future.

To Dr. Valeria La Saponara and Dr. Donald Margolis thank you for providing guidance to me as an undergraduate and encouraging me to pursue my graduate degree.

Finally, I'd like to thank my girlfriend Stephanie who has given me the support to pursue my interests. I would not be here without your encouragement.

Contents

Table of Figures

List of Tables

1 Introduction

In an attempt to reduce costs in the motorsports industry, sanctioning bodies imposed cost cutting measures to help curb spending by race teams. One of these cost cutting measures was to limit the amount of actual track testing which would reduce expenses such as travel and personnel. As a consequence, drivers that compete at an unfamiliar track have a disadvantage. To aid the drivers in learning the track, driving simulators have been used. Vehicle models must also be incorporated into the driving simulator to provide the drivers with a realistic feel for the vehicle. Driving simulators also have the benefit of allowing drivers to push the limits in their vehicles without fear of a costly repair or safety concerns.

Beyond motorsports, driving simulators have other uses including human factors testing. The use of a driving simulator provides an ideal platform to run repeatable tests due to actual testing that could include variability in factors like weather and visibility.

In order to use a driving simulator as an engineering tool, high fidelity vehicle models must be employed. Suspension geometries must be included for an accurate vehicle model. This however, would necessitate solving differential algebraic equations, which would use iterative methods for integration and are computationally expensive algorithms to implement. For a driving simulator, solving differential algebraic equations is not practical because suspension forces must be calculated in real-time in order for the appropriate response to be experienced by the driver.

The objective of this research is to develop a nonlinear approximation to the differential algebraic equations that is suitable for real-time simulation. An experimental test setup was constructed in

order to collect data and later verify the models. This setup was mounted to a quarter car test rig that had a hydraulic wheel loader actuator and two parallel electromagnetic aeroloader actuators. Data was collected after running various setups with filtered white noise as inputs to the actuators and was used to develop and validate two numerical models.

A simplified two-dimensional model of the same suspension was then developed in SimMechanics that included the nonlinear suspension geometry. Parameters for this model were identified using an optimization technique that compared the output response of the model to experimental data.

A linear model with nonlinear constitutive equations was then developed using parameters from the SimMechanics model. The nonlinear elements of the suspension were found by applying a virtual Kinematics and Compliance test in the SimMechanics model and through the derivation of analytic equations.

The objective of this research is to develop an approach for approximating the differential algebraic equations associated with a vehicle suspension model to use in real-time simulation. This analysis was restricted to a quarter car model in order to focus on the nonlinearities due to the suspension geometry. Although the analysis was performed on a single suspension type, the results can easily be extended to apply to a wide range of suspensions.

Three different model types were studied in this analysis. Response data from an experimental model was used to characterize and validate two separate computational models. The first computational model was constructed in SimMechanics, and represented the suspension dynamics with differential algebraic equations. The second computational model was constructed in Matlab as a linear translational model with nonlinear elements.

2 Literature Review

This section is the literature review related to driving simulators and real-time suspension models. Chapter 2.1 discusses current advancements in driving simulators and their uses. Chapter 2.2 discusses various real-time suspension models that are currently used for research and commercial applications.

2.1 Driving Simulators

The use of driving simulators is not a new concept. They serve various purposes from driver behavior research to virtual prototyping. Low level driving simulators have been developed in order to reduce equipment costs compared to a full 6-DOF driving simulator like the one seen in Figure 2.1 at the Virginia Institute for Performance Engineering and Research located at Virginia International Raceway. Haung and Chen developed a low cost 5-DOF driving simulator to use for vehicle dynamics simulations [1]. This driving simulator is missing the longitudinal actuator for acceleration and decelerations to reduce costs. Instead, complex visualizations and a pressure vest were proposed to simulate acceleration and deceleration effects.

The increase in traffic accidents led a Japanese government institute to scientifically determine the causes of traffic accidents [2]. As a result, driving simulators were developed by research institutions in Japan and later by auto manufacturers in an attempt to increase safety.

Driving simulators can be used for conducting driver behavior research. The DLR Institute of Transportation Systems uses a mid level 6-DOF fixed base driving simulator to simulate traffic scenarios to see how drivers would respond [3]. For these experiments to yield relevant results, participants must feel as if they are in a real scenario. The work at the DLR Institute clearly demonstrates the importance of high fidelity models.

Renault and Peugeot operate a high level large amplitude motion based driving simulator for virtual testing [4,5]. This driving simulator includes a moving base to simulate larger longitudinal and lateral accelerations. Systems like driver aids and active safety systems can be tested using this type of driving simulator. Models of systems such as ABS and electronic stability control can be built and tested to validate their use before incorporating them into an actual vehicle. This may help reduce the number of prototypes and physical testing required before a system is finalized.

Figure 2.1: VIPER Service Driving Simulator

Suda et. al. developed a tire evaluation system using a driving simulator [6]. By using a tire test machine in conjunction with a driving simulator, a virtual proving ground for tire tests can be achieved.

Driving simulators can also be used in road design [7,8]. Bella performed a study using a fixed base driving simulator and a sample of participants to determine how people judge the speed in which they enter a curve in the road. The use of a driving simulator in this type of research allows the researcher to reduce the number of variables such as sunlight, the tire to road friction coefficient, and weather that may change how participants would gauge vehicle speeds in turns.

The National Advanced Driving Simulator in Iowa uses their simulator for human factors testing like driver impairment studies that would otherwise be dangerous to test on the road [9]. A high fidelity driving simulator was developed in order to immerse their participants into an environment as realistic as possible. The 9-DOF driving simulator includes a yaw ring that allows the base of the vehicle to rotate 330 degrees as well as an enclosed dome utilizing image projectors for a 360 degree view of simulated terrain.

Each of these studies indicates that replicating actual driving scenarios is very important in order to obtain accurate results. High fidelity vehicle models are required to simulate actual vehicle responses.

2.2 Real-Time Suspension Models

The National Advanced Driving Simulator (NADS) developed their own multi-body dynamics models for use on their driving simulator [9,10]. A recursive formulation of the equations of motion that allowed parallel computing was developed. Simplified DAEs were used in order to accomplish real-time simulation. Suspension kinematics and compliances were physically measured for each vehicle to obtain the vertical suspension force with respect to suspension deflection. Dampers were also tested on a shock dynamometer and used to verify the accuracy of their suspension model. The use of high performance computers utilizing parallel computing is required for this vehicle dynamics model in order to operate in real-time.

Shiiba and Suda proposed a 91-DOF vehicle model to use on their driving simulator at the University of Tokyo [6,11-13]. An approximated multi-body dynamics model was used since high order multi-body vehicle models are difficult to run in real-time because of the computation of differential algebraic equations. Instead, a simplification was proposed by treating the constraints Jacobian matrix as a constant matrix. As a result, DAE's are not solved in real-time and are approximated.

Methods to integrate the Jacobian matrix for real-time simulation were developed by Bae et al. [14]. Instead of integrating the Jacobian matrix at every time step, this method would update the Jacobian only once in many integration steps. This method does not solve DAE's in real-time.

Eichberger and Rulka took the approach to reduce a complex suspension model by using a macro joint approach for real-time simulation modeling [15]. Instead of solving differential algebraic equations or using lookup tables, they chose to simplify the equations of motion down to first order ordinary differential equations that increased the simulation speed by a factor of 15. The paper, however, only focused on reducing simulation time and did not address the accuracy of the model with respect to a DAE model.

Other approaches to real-time vehicle models include a subsystem synthesis approach that separates the equations of motion of the vehicle and the equations of motion of the suspension for a more efficient computation [16-18]. For real-time simulation, approximations were made in the numerical approach to solve the separated equations of motion. This method does not solve DAEs in real-time.

Choi et al. developed a method that used suspension composite joints to reduce computation time [19]. The joints were modeled as massless links to reduce the number of bodies and simplify the equations of motion, but do not solve DAEs.

3 Experimental Model

An experimental model was constructed in order to characterize and validate two numerical suspension models. A suitable experimental suspension platform was selected for testing on the existing quarter-car test rig at the Performance Engineering Research Lab.

3.1 Testbed Construction

A NASCAR chassis was selected as a platform for the quarter-car test rig due to availability. The selected chassis has a double wishbone front suspension and separate pickup point locations for the spring and damper.

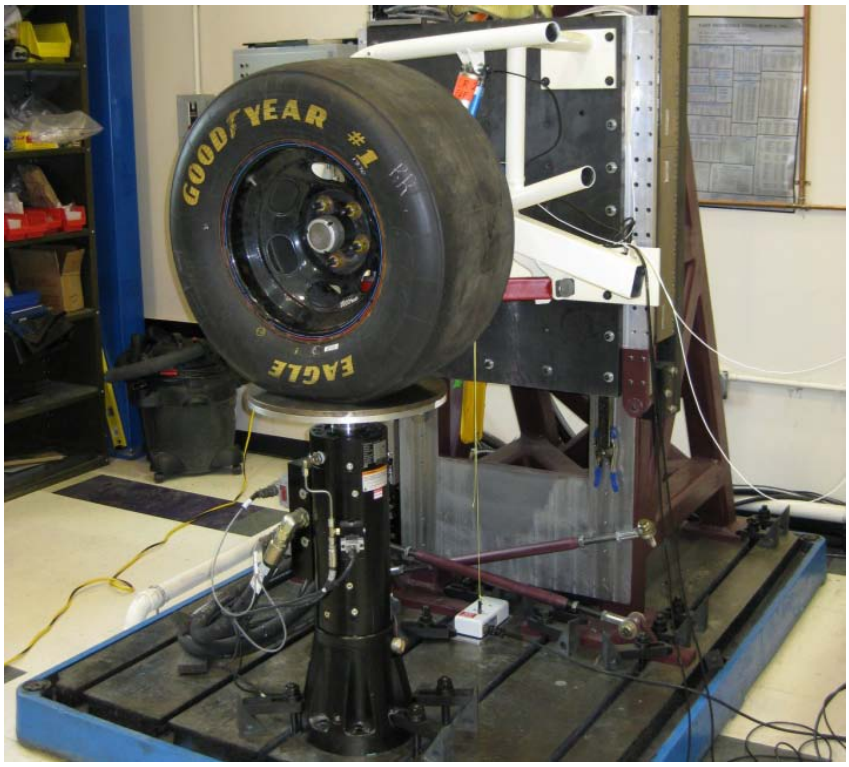

Figure 3.1: Quarter Car Test Rig

The front wheel, suspension components and body work were first removed from the chassis and set aside from the donor vehicle. The front clip was cut from the donor vehicle using a reciprocating saw and cleaned using a sandblaster to remove paint and contaminants and expose bare metal in preparation for welding. To adapt the chassis onto the existing quarter car rig, two plates were fabricated and welded onto the chassis in order to bolt the chassis to the mounting plate of the quarter car test rig. The ability to bolt the chassis onto the quarter car rig allows the rig to retain the capability to upgrade to other suspension systems for future research. Pictures of the final quarter car rig are shown in Figure 3.1 and Figure 3.2.

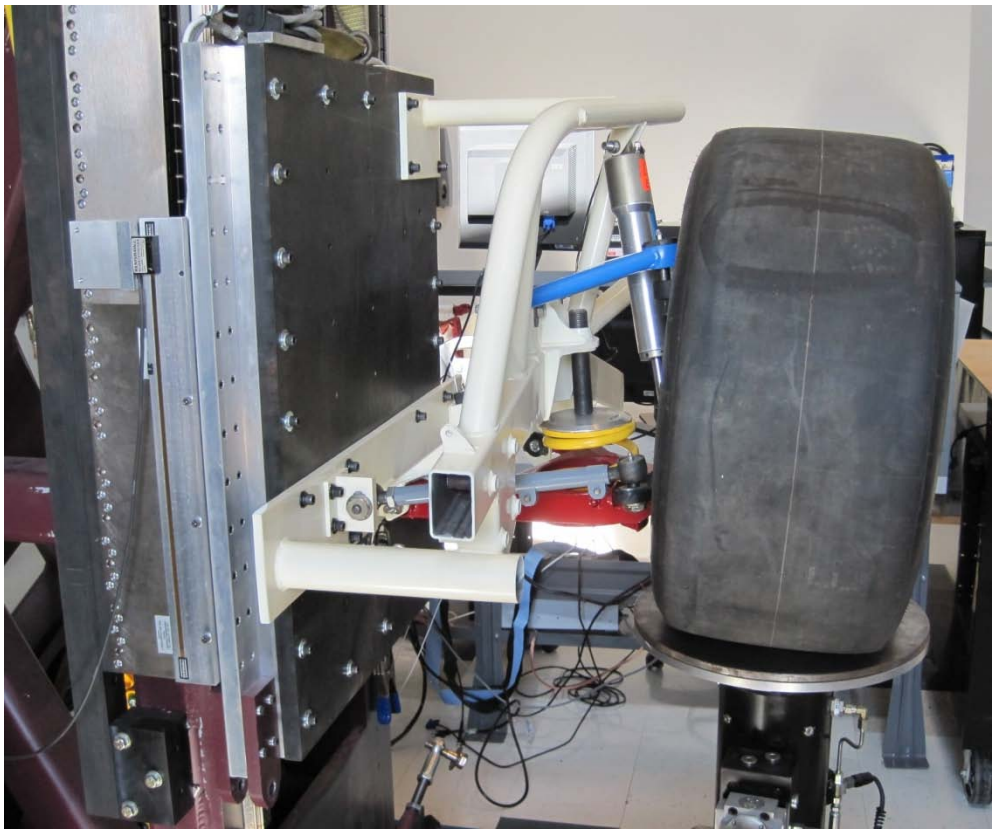

Figure 3.2: Side View of Quarter Car Rig

All components were sprayed using an acid etch primer intended for use on bare metal before the final coat of paint was applied. A color scheme was chosen to easily identify various parts of the

suspension system shown in Table 3.1. This would allow for clear demonstrations during tours and educational programs for local students.

Table 3.1: Color Table for Experimental Setup

Part	**Color**
Chassis	Crème
Upper Control Arm	Blue
Lower Control Arm	Red
Steering Tie Rod	Gray
Spring	Yellow
Spindle	Black

The quarter car test rig has two types of inputs to excite the system. There is a hydraulic actuator that sits underneath the tire to simulate road inputs. Two parallel linear electromagnetic motors apply a force to the mounting plate to simulate aerodynamic loads on the chassis. Electric motors have a linear relationship between current and force. Given a known current, the force will be proportional, eliminating the requirement for closed loop force feedback. Using a transconductance amplifier, an input voltage can be converted into current, becoming a voltage controlled current source. Voltage now has a direct relationship to the force and can be used as an input to provide a known force output for the aeroloaders.

3.2 Experimental Dynamic Tests

The test rig was instrumented with various sensors, however, only the sprung mass displacement sensor and shock displacement sensor were used for this research. The sprung mass displacement was measured using a UniMeasure, Inc VPA-15-DS-L5M with a range of 38 cm. The shock displacement was measured used a Penny and Giles SLS190 series linear displacement sensor with a stroke length of 225 mm. The four signals that were of significance were the output responses and the input excitations; the sprung mass displacement, shock displacement, wheel pan displacement, and the aeroloader control signal. These signals were used to build and

validate the two numerical models. The wheel pan displacement was measured using a linear variable displacement transducer built into the hydraulic actuator.

Six experimental tests were conducted to collect data for various conditions shown in Table 3.1. The random aeroloader and wheel inputs were white noise signals that were filtered at 5Hz and 20 Hz respectively using a 4^{th} order low pass Butterworth filter. The white noise input was developed using Matlab's *randn* function and filtered using the *butter* function. The constant aeroloader force input was obtained by adding a dc offset into the control signal for the aeroloaders.

The first two experimental runs were high energy wheel inputs with and without a dc aero input. The third and fourth runs were low energy wheel inputs with and without dc aero inputs. The fifth and sixth runs were a random aero input with and without dc aero inputs.

Previous tests have shown that for low wheel energy inputs, a linear model is sufficient for predicting output responses [20-22]. The high wheel energy runs were selected to operate the suspension in a more nonlinear region. Two aeroloader conditions were chosen to show the effects of a constant downforce and a time varying downforce.

Table 3.2: Inputs for Experimental Setup

Experimental Run	**High Energy Wheel Input**	**Low Energy Wheel Input**	**Random Aero Input**	**DC Aero Input**
Run 1	X			
Run 2	X			X
Run 3		X		
Run 4		X		X
Run 5			X	
Run 6			X	X

The power spectral density of the inputs is shown in Figure 3.3. The cut off frequencies for each signal correspond to the frequency that the signals were filtered at. Notice that the wheel input plots are very similar, except for the approximately 10 dB magnitude difference.

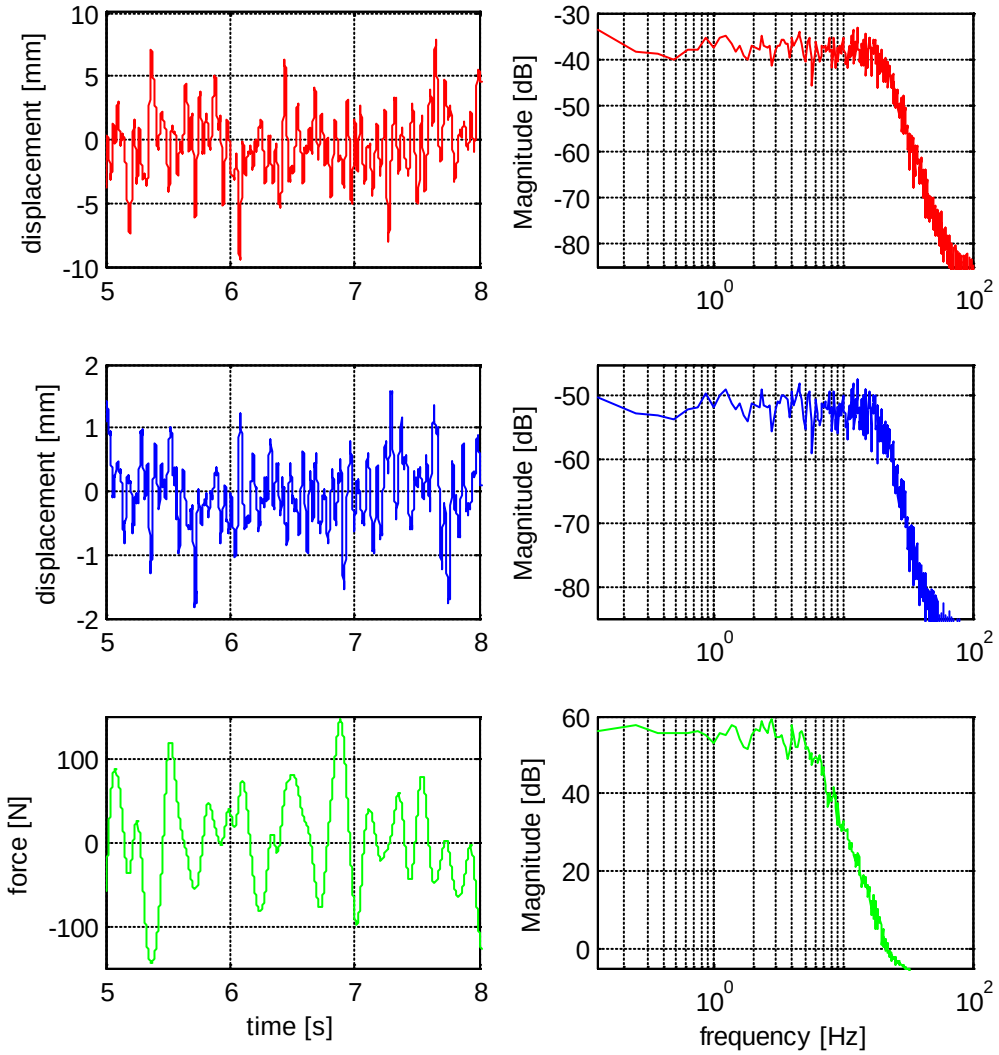

Figure 3.3: Input Signals and their Corresponding Power Spectral Densities

The left column of Figure 3.3 shows the time series plots of the high energy wheel input, low energy input, and random aero input from 5 seconds to 8 seconds. The actual length of the input

signal was 30 seconds but only a small subsection of the input signal was plotted in order to show the difference in the spectral content between the wheel loader inputs and the aeroloader input. The right column shows the corresponding power spectral densities of the input signals. The high energy wheel input has a maximum peak displacement of 9.87 mm and a standard deviation of 2.56 mm. The low energy wheel input has a maximum peak displacement of 1.97 mm and a standard deviation of 0.51 mm. The power spectral densities of the high energy and low energy wheel inputs are very similar since they have the same cutoff frequency. The magnitude of the high energy wheel input is larger by approximately 10 dB due to the larger amplitude in comparison to the low energy wheel input. The random aeroloader input has a maximum peak force of 177.46 N and a standard deviation of 59.24 N. The power spectral density plot rolls off at 5 Hz. At 0 dB, the slope of the magnitude changes and becomes less negative, probably due to round off error. The change in slope however is insignificant because it is over 50 dB smaller than the magnitude of the passband of the input signal.

Two of the time series response plots from the experimental model are shown below in Figure 3.4 and Figure 3.5. Figure 3.4 shows Run 2, the high energy wheel input with a dc aeroloader input. From the figure, the aeroloader control signal maintains a force of approximately -236.5 N. The wheel pan displacement range is approximately 20 mm. Although a displacement of 20 mm may not seem like a high amplitude input, due to the input frequencies, this excitation is very high energy. During the test, the hydraulic fluid in the hydraulic lines that run from the pump to the actuator were beginning to cavitate and vibrations from the hydraulic actuator were felt through the floor. Due to the dc aeroloader input, there is an offset in the shock and sprung mass displacements. The displacement range of the sprung mass is smaller than the displacement range of the wheel pan because the suspension reduces the amplitude of the force that is transmitted to the sprung mass. The shock and sprung mass displacements are at zero when the system is at rest with no inputs applied.

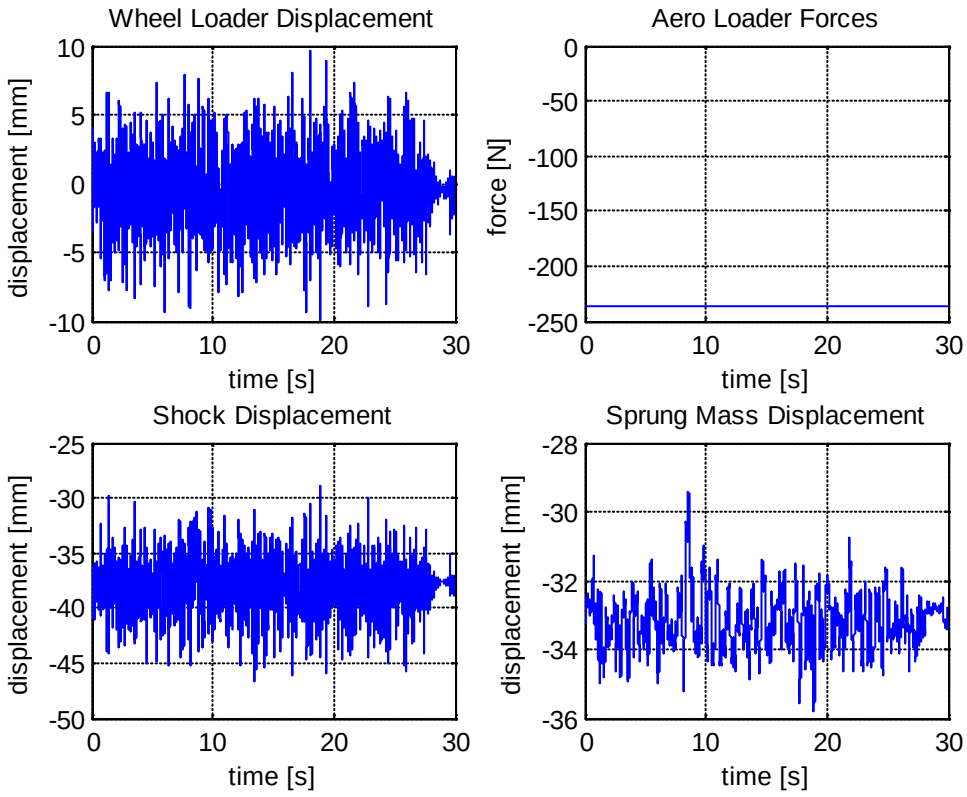

Figure 3.4: Experimental Data from Run 2

In Figure 3.5, the only input was a random aeroloader signal filtered at 5 Hz. A comparison of the spectral content of the outputs shows that the spectral content from Run 5 is significantly lower than the spectral content of the high energy wheel loader input in Run 2. The range of the aeroloader force is approximately ±175 N. The wheel pan displacement is essentially zero and the signal that is seen is most likely from noise. The sprung mass displacement has a range of approximately 35 mm, over 4 times the displacement than from the high energy wheel input run.

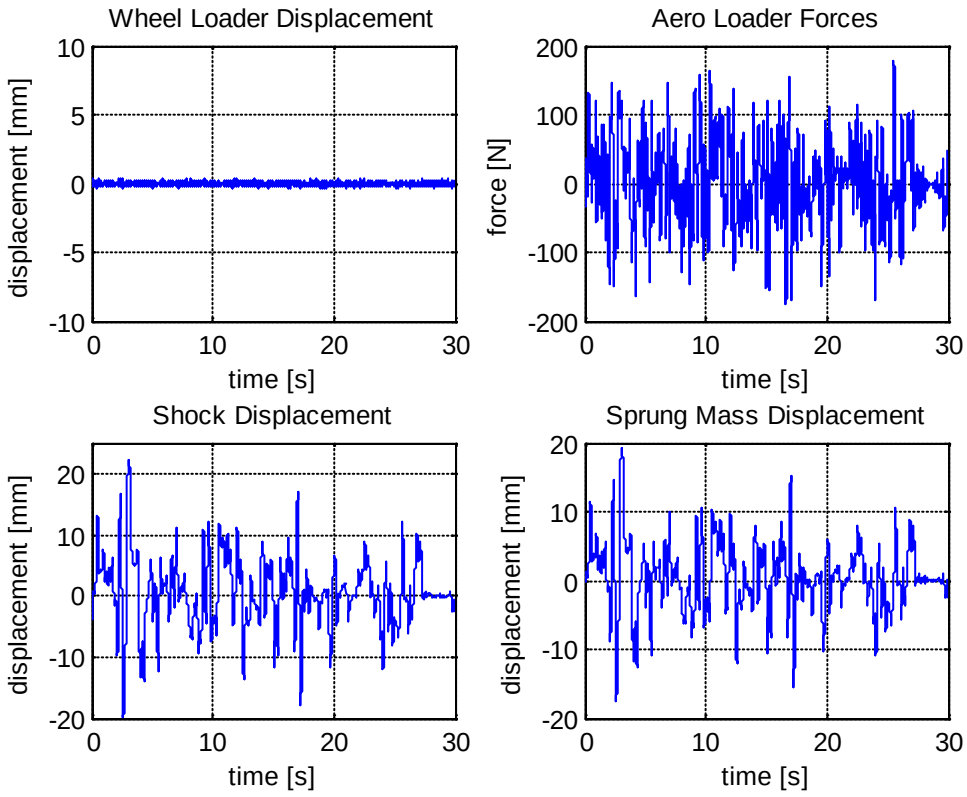

Figure 3.5: Experimental Data from Run 5

The data collected from the experimental test was used to build and validate the numerical models in Chapters 4 and 5. A description of how the experimental data is used is presented in Chapter 4.

4 SimMechanics DAE Model

SimMechanics is a toolbox that works with MATLAB's Simulink software to simplify building multi-body models with the use of a graphical user interface. The Section 4.1 describes how the model was constructed in SimMechanics. The Section 4.2 describes how the unknown parameters were optimized in Matlab. The Section 4.3 shows the results of the optimization.

4.1 Model Construction

A two dimensional model of the experimental test chassis was built in SimMechanics that included the suspension geometry shown in Figure 4.1 and Figure 4.2. To build the model, certain values such as pick up point locations, masses and rotational inertias for each body needed to be specified. The equations of motion were automatically derived and solved using one of Simulink's user specified ODE solvers.

Geometry and pickup point location values are shown in Table 4.1. These values were found by measuring lengths on the experimental test rig. The initial angles for the control arms were measured using a digital level. Center of gravity (CG) locations were approximated since measurements tools were not available. The unsprung mass CG location was chosen to be at the spindle location. The CG location of the upper and lower control arms were chosen to be halfway between the pickup point locations. The sprung mass CG location is approximately at the midpoint of the sprung mass plate.

There were several simplifications that were made when the SimMechanics model was being constructed. The experimental model is a three dimensional model whereas the SimMechanics model is only two dimensional. The assumption was made that the majority of the suspension movement is planar. Each connection was assumed to be frictionless; therefore stiction in the

joints was not accounted for. The spring and damper were also assumed linear. The tire was also modeled as a spring and damper system. The steering tie rod was lumped into the lower control arm system.

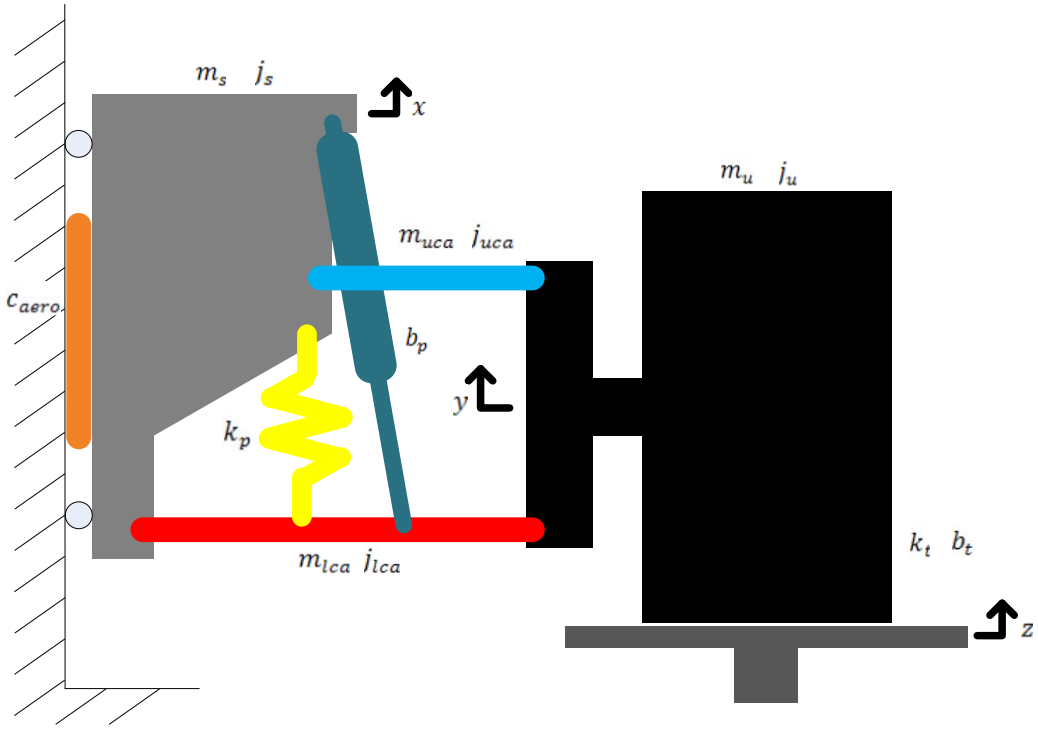

Figure 4.1: Two Dimensional SimMechanics Model of Experimental Setup

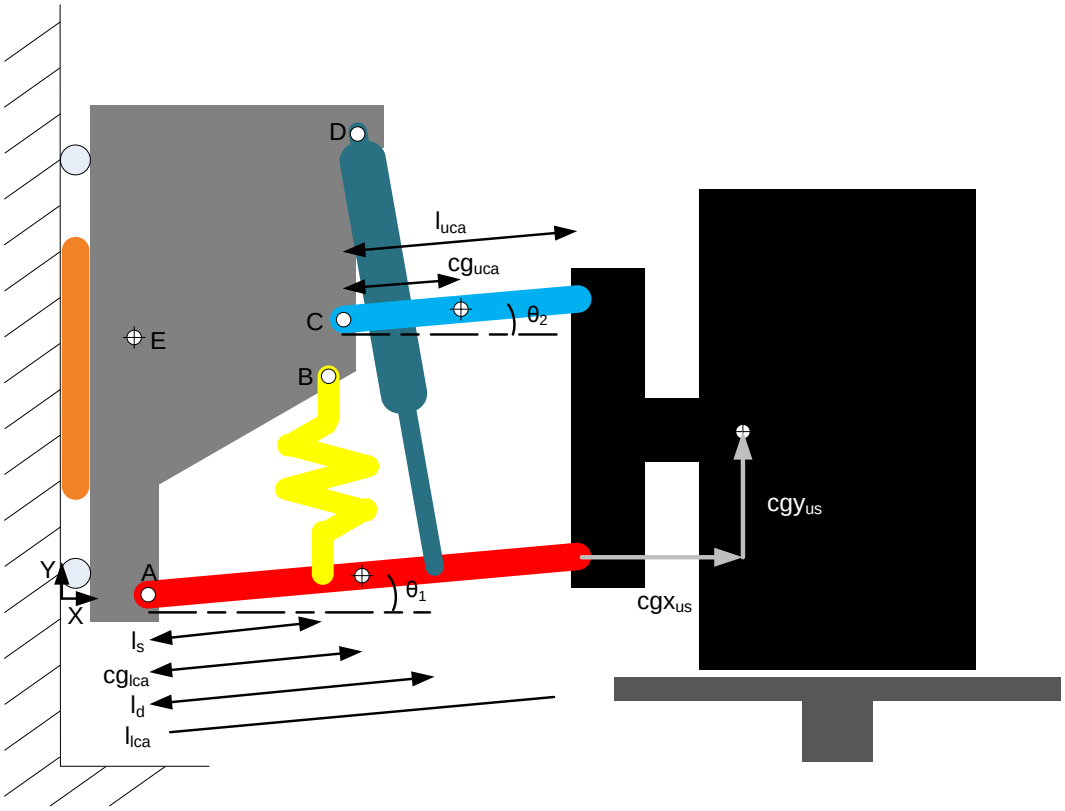

Figure 4.2: Known Parameters for SimMechanics Model

Table 4.1: Parameter Values for SimMechanics Model

Parameter Description	**Variable Name**	**Values**
LCA Pickup Point	A	(0.06, 0) m
Spring Pickup Point	B	(0.32, 0.115) m
UCA Pickup Point	C	(0.216, 0.3) m
Damper Pickup Point	D	(0.345, 0.52) m
Sprung Mass CG	E	(0, 0.5) m
Spring/LCA Pickup Point	l_s	0.285 m
CG of LCA	cg_{lca}	0.2075 m
Damper/LCA Pickup Point	l_d	0.4 m
Length of LCA	l_{lca}	0.415 m
Length of UCA	l_{uca}	0.23 m
CG of UCA	cg_{uca}	0.115 m
X Component of Unsprung Mass CG	cgx_{us}	0.1 m
Y component of Unsprung Mass CG	cgy_{us}	0.2 m
Initial Angle of LCA	θ_1	2.7º
Initial Angle of UCA	θ_2	15.6º

From Figure 4.3, the SimMechanics toolbox essentially looks like Simulink, using a block diagram form to build multi-body models. Like Simulink, subsystems can be created to group blocks and functions together. SimMechanics differs from Simulink in that each block represents a physical component or relationship rather than mathematical operations.

The colored blocks in Figure 4.3 are subsystems that represent physical parts of the quarter car test rig. The orange aeroloader block is connected to the sprung mass block through a prismatic joint that constrains the aeroloader forces to only the vertical direction. The sprung mass is connected to four subsystems, the upper control arm (UCA), the damper, the spring, and the lower control arm (LCA). The upper and lower control arms are directly connected to the unsprung mass, whereas the damper and spring have connections to the lower control arm. The unsprung mass is attached to the tire, which receives an input from the wheel actuator.

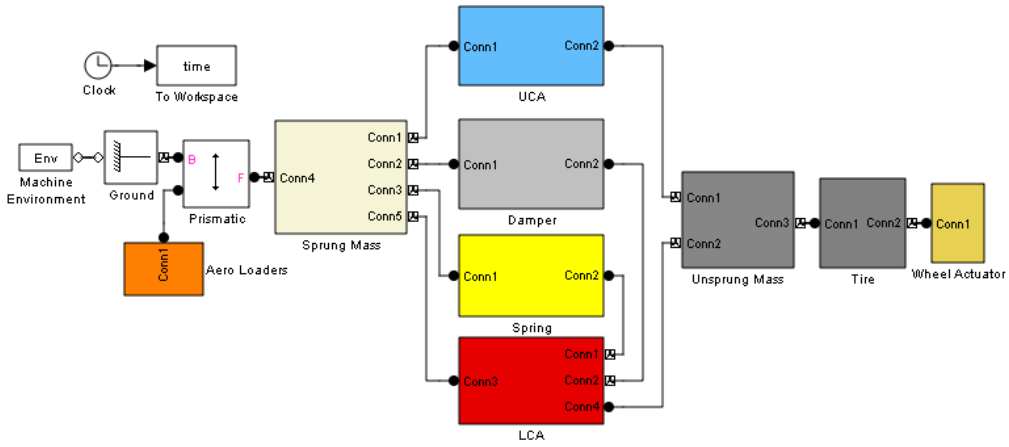

Figure 4.3: Overview of SimMechanics Model

Figure 4.4 shows the subsystem of the sprung mass block in Figure 4.3. The sprung mass is a block that contains all the information required to simulate the rigid body. Each connection is a user specified pickup point or joint that represents a physical connection like a rotational pivot point. For example, a body sensor is attached to the CG location of the sprung mass, outputting

position and acceleration data. Each of the connections on the sprung mass represents a pickup point location. On the left of the sprung mass, Conn4 is attached to the prismatic joint and is also where the aeroloader forces are applied. On the right of the sprung mass are the pickup point locations for the upper control arm, damper, spring, and lower control arm.

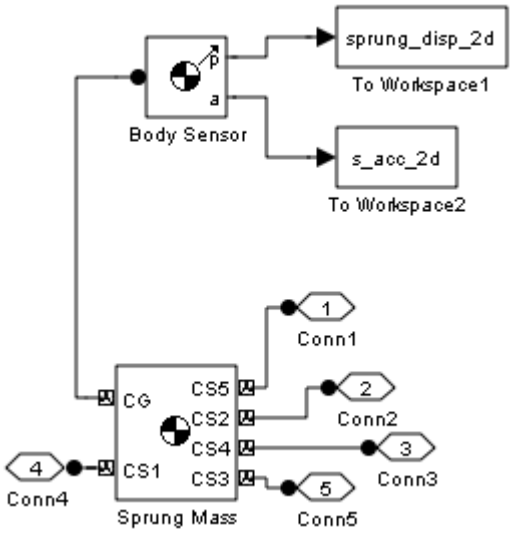

Figure 4.4: Sprung Mass Block in SimMechanics

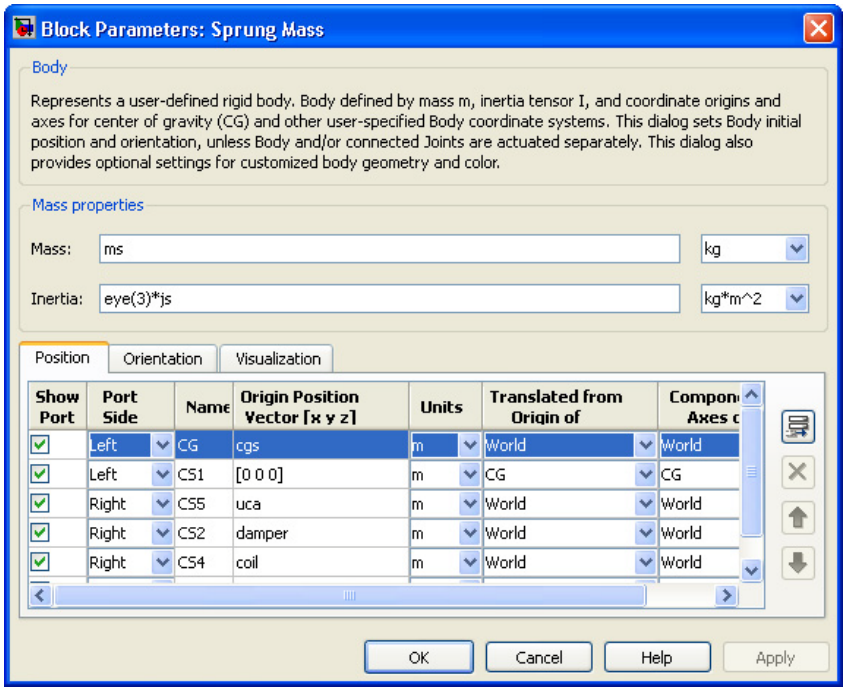

Figure 4.5: Sprung Mass Parameter Window

Figure 4.5 shows the block parameters for the sprung mass. This is where the user can specify the location of the pickup points, the axes in which the coordinate system shall lie, and the orientation. The body requires a mass value and inertia matrix to be specified.

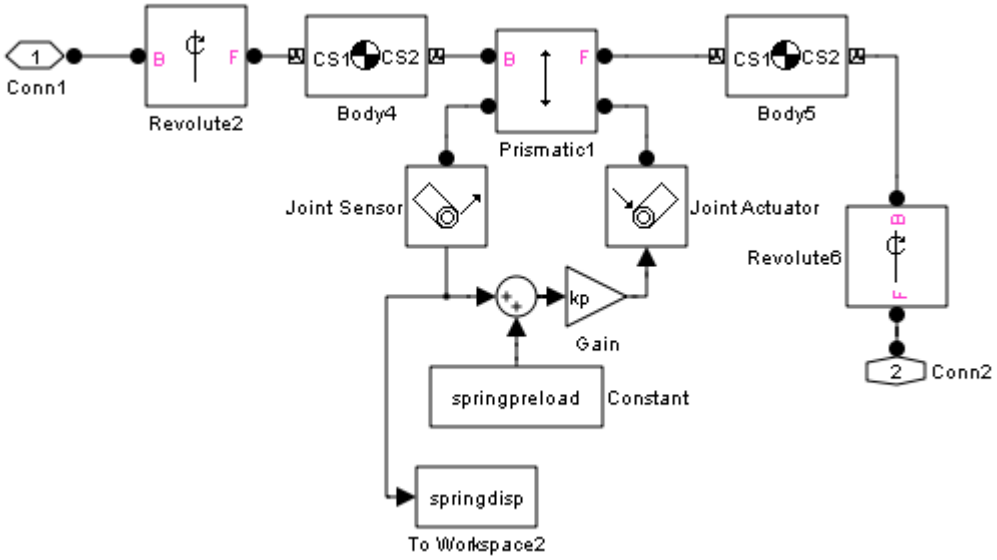

Figure 4.6: Spring Block in SimMechanics

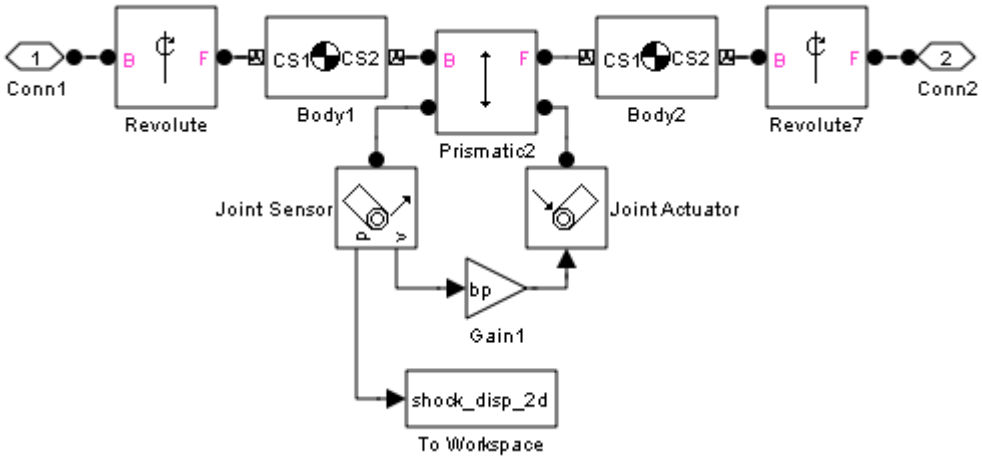

Figure 4.7: Damper Block in SimMechanics

The spring and damper in Figure 4.6 and Figure 4.7 are built using a joint sensor and actuator that acts on the prismatic axis of the spring and damper. The spring force is dependent on the spring displacement along the prismatic axis and is multiplied by a factor kp, the linear spring constant. The damper force on the other hand, is dependent on the velocity along the prismatic axis of the

damper, and is multiplied by a factor bp, the damping constant. The joint sensor detects the displacement or velocity whereas the joint actuator imposes a force on the system.

The tire is modeled as both a spring and damper, seen in Figure 4.8. The joint sensor outputs both position and velocity. The force from the spring and damper are added before going to the joint actuator.

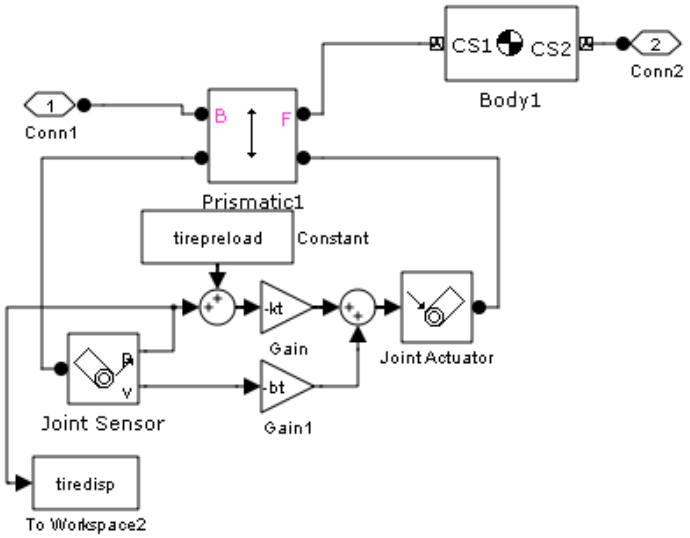

Figure 4.8: Tire Block in SimMechanics

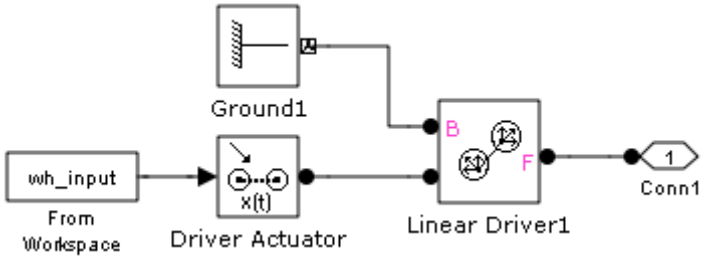

Figure 4.9: Wheel Actuator Block in SimMechanics

Wheel actuation is accomplished using an input signal that is developed in MATLAB. A matrix that includes both a time vector and a displacement vector defines the position with respect to time of the wheel actuator. The actuator is then constrained to move only in the vertical direction.

4.2 Parameter Identification

Although certain parameters, such as geometry and pickup point coordinates, are easily measured from the experimental model, the challenge for the SimMechanics DAE model was to identify the unknown parameters. The approach adopted for this study was to estimate parameters for the two dimensional SimMechanics model using an optimization method to match the response of the sprung mass displacement and the shock pot displacements from the experimental test runs. A vector P contained the unknown variables in the SimMechanics model that were to be identified. The function J(P) is a cost function that compares the sum of the squares of the error between the simulated response and the experimental data of the sprung and shock displacements. These errors were scaled in order to eliminate the optimization bias towards the shock or the sprung mass due to the range difference in the displacement data. The optimum parameter vector would be the argmin of J(P), which was estimated here using the MATLAB function *fmincon*.

$$P = \left[m_s, j_s, m_u, j_u, m_{uca}, j_{uca}, m_{lca}, j_{lca}, k_p, b_p, k_t, b_t, c_{aero}\right] \tag{4.1}$$

$$J(P) = \alpha \sum\left(\delta_{shock}^{sim} - \delta_{shock}^{exp}\right)^2 + \beta \sum\left(\delta_{sprung}^{sim} - \delta_{sprung}^{exp}\right)^2 \tag{4.2}$$

$$P_{opt} = argmin\, J(P) \rightarrow FMINCON \tag{4.3}$$

$\delta_{shock}^{sim} = shock\ displacement\ from\ SimMechanics\ model$
$\delta_{shock}^{exp} = shock\ displacement\ from\ experimental\ testing$
$\delta_{sprung}^{sim} = sprung\ mass\ displacement\ from\ SimMechanics\ model$
$\delta_{sprung}^{exp} = sprung\ mass\ displacement\ from\ experimental\ testing$
$\alpha = scale\ factor\ for\ shock\ displacement$
$\beta = scale\ factor\ for\ sprung\ mass\ diplacement$

Limits were placed on each variable in the vector P during the optimization in order to constrain solutions to known ranges. As a result, initial optimization iterations would saturate at the limits.

Multiple iterations were performed in order to ensure each parameter was not at one of the maximum or minimum limits.

The optimized parameters are shown in Table 4.2. Due to the amount of time it takes to complete an optimization, the bulk of the optimizations were focused on Run 1 and Run 3 to make sure that the parameters were not at a maximum or minimum. The rotational inertia of the sprung mass j_s does not change for any of the runs because the system was constrained such that there was no rotation between the ground and sprung mass.

Table 4.2: Optimized Parameters of the SimMechanics Model

Parameters	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Run 6**
$\boldsymbol{m_s}$ [kg]	173	170	181	170	180	170
$\boldsymbol{j_s}$[kg m^2]	20	20	20	20	20	20
$\boldsymbol{m_u}$ [kg]	14.930	7.315	6.708	8.050	6.283	1.216
$\boldsymbol{j_u}$ [kg m^2]	11.262	9.460	4.781	5.738	7.047	12.410
$\boldsymbol{m_{uca}}$ [kg]	3.814	1.868	3.474	4.169	3.251	2.291
$\boldsymbol{j_{uca}}$ [kg m^2]	0.504	0.247	0.272	0.190	0.328	0.257
$\boldsymbol{m_{lca}}$ [kg]	5.025	3.852	4.476	3.139	3.591	3.442
$\boldsymbol{j_{lca}}$ [kg m^2]	0.191	0.093	0.299	0.216	0.3581	0.2310
$\boldsymbol{k_p}$ [N/m]	29235	24558	28033	19623	32486	33770
$\boldsymbol{b_p}$ [N/m/s^2]	2187	3149	3000	2100	2202	2667
$\boldsymbol{k_t}$ [N/m]	301670	253403	156148	109304	323790	1464242
$\boldsymbol{b_t}$ [N/m/s^2]	476	854	949	685	926	720
$\boldsymbol{c_{aero}}$	0	0.9	0	1.1	1	0.9

The difference between parameter values for each test case can be due to several reasons. All parameters do not have an equal weight on the effect on the output response. These parameters may have a larger fluctuation as the optimization routine adjusts parameters in order to minimize the cost function. As a result, some parameters are similar while others vary significantly. An optimization can be made to average all the test cases and come up with one set of parameter values, however, the decision was made to separate the optimizations for each test case.

4.3 Simulated Dynamic Tests

Figure 4.10 and Figure 4.11 shows the experimental data compared to the simulated output using the optimized parameters in Table 4.2 for Run 3 and Run 5 respectively. The error between the experimental model and the SimMechanics model is shown in red. Although the SimMechanics model uses optimized parameters that minimize the error, the results do not match perfectly. This was expected since there were several simplifications that were made when the model was developed. Some of the error that is shown is due to a phase shift between the two data sets. This effect can be seen more clearly in Figure 4.11 since the spectral content of the response is much lower than the response in Figure 4.10.

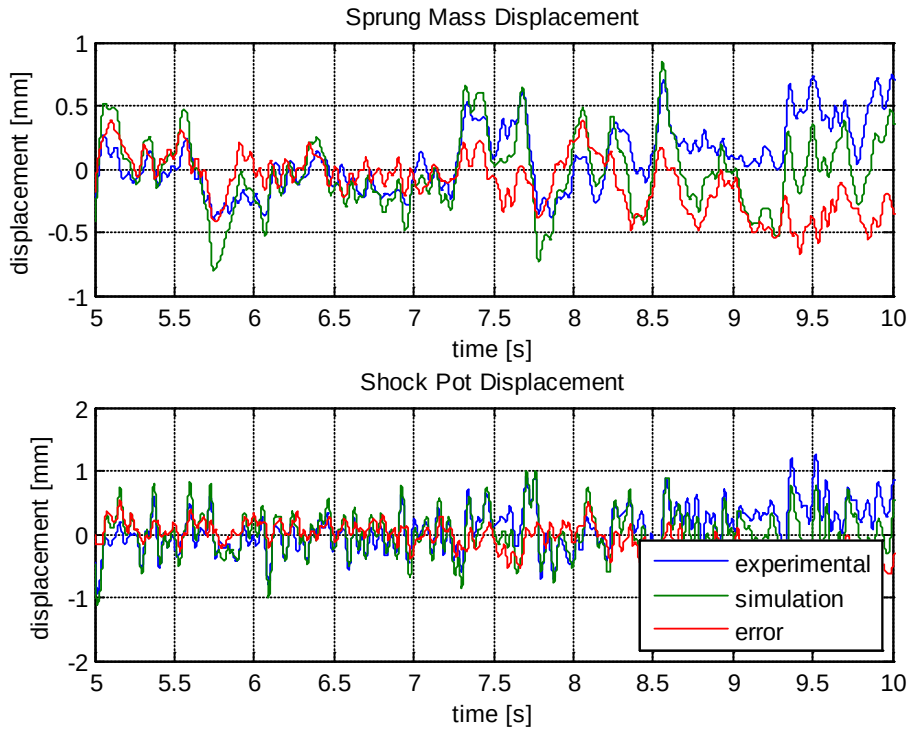

Figure 4.10: Response Comparison of Optimized Parameter Simulation and Experimental Data for Run 3

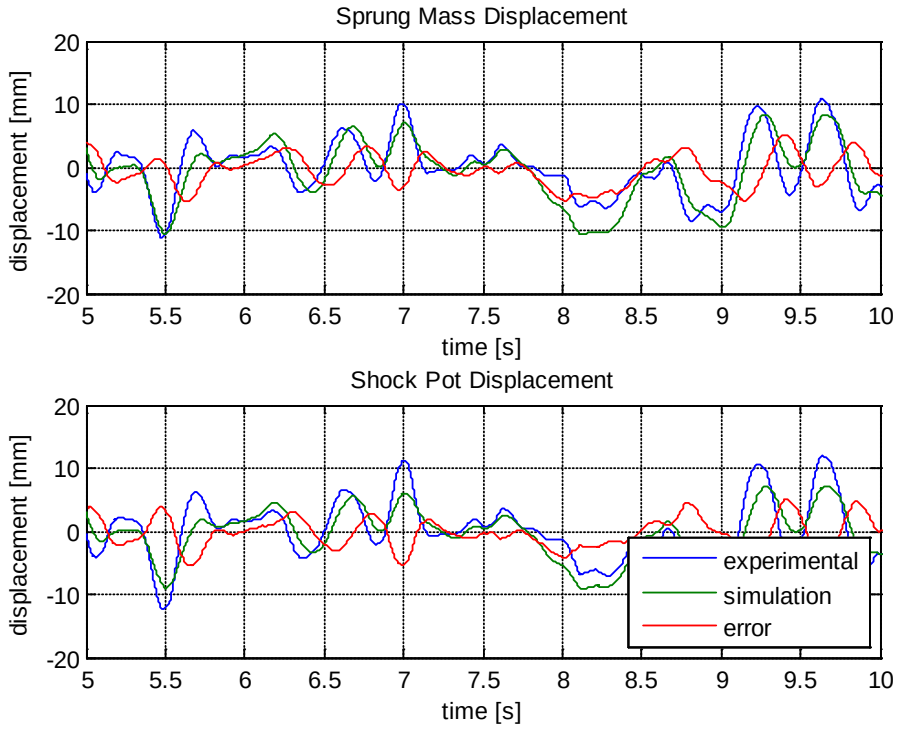

Figure 4.11: Response Comparison of Optimized Parameter Simulation and Experimental Data for Run 5

Values for the unknown parameters were identified for each test run. These parameters are used to develop the linear model with nonlinear suspension elements in Chapter 5.

5 Linear Model w/ Nonlinear Elements

This chapter first describes how the linear model with nonlinear elements was constructed and how the equations of motion for this system were derived. The following sections then describe how the nonlinear suspension elements were created and implemented into the linear model.

In order for the model to run in real-time, simplifications were necessary to reduce computation time. The first simplification was to separate the linear model into two parts, a linear portion and a nonlinear portion. Figure 5.1 is a diagram of the generalized Lur'e problem and shows how the linear and nonlinear portions of the suspension will interact to produce the output signal. The nonlinear portion is a memoryless nonlinearity. The second simplification was to discretize the model which approximated the continuous time model.

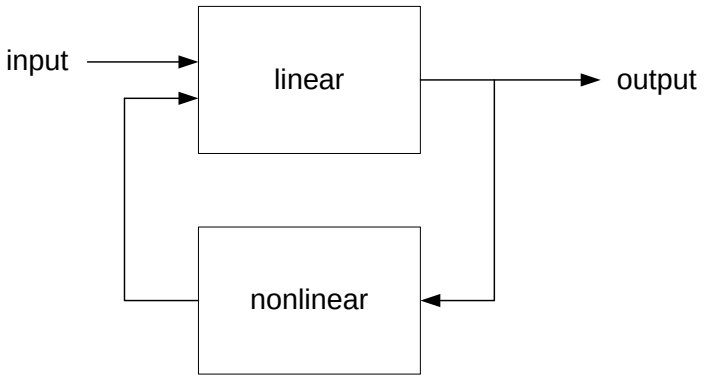

Figure 5.1: Generalized Lur'e Problem

5.1 Model Construction

Several ways to represent the nonlinear suspension elements were investigated. The first was by using the displacement to force relationship directly by using displacement δ as an input, as seen in Figure 5.2. The $f(\delta)$ block would be a lookup table that maps the force versus displacement

plot of the spring. The second model in Figure 5.3 uses a lookup table to find the nonlinear spring constant and multiplies the spring constant by the displacement obtain the force. The third model in Figure 5.4 separates the spring into two parts, a linear part and a nonlinear part. The nonlinear subsystem in this third model is required to be a memoryless nonlinearity. Note that all three nonlinear model representations are mathematically equivalent; however, only the third one is useful for real-time implementation as will be shown.

Figure 5.2: Direct Displacement to Force Relationship Model for a Nonlinear Spring.

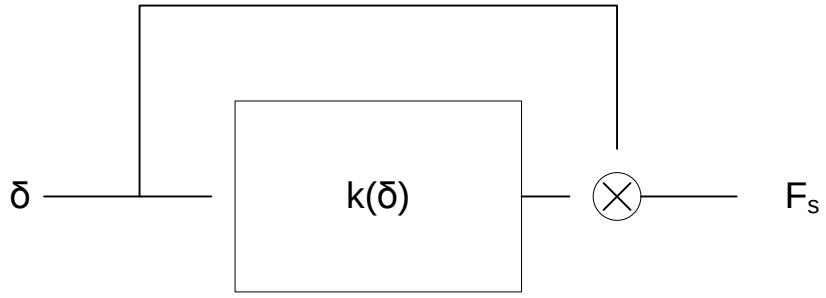

Figure 5.3: Nonlinear Spring Constant Model

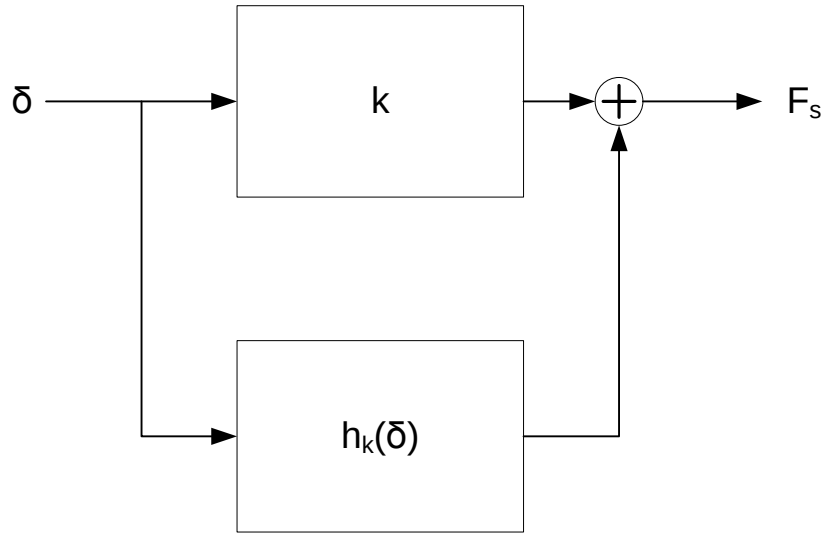

Figure 5.4: Separated Linear and Nonlinear Spring Model

The model for the damper is an exact analog of the spring, except that instead of only being a function of displacement, it is a function of both displacement and velocity as shown in Figure 5.5.

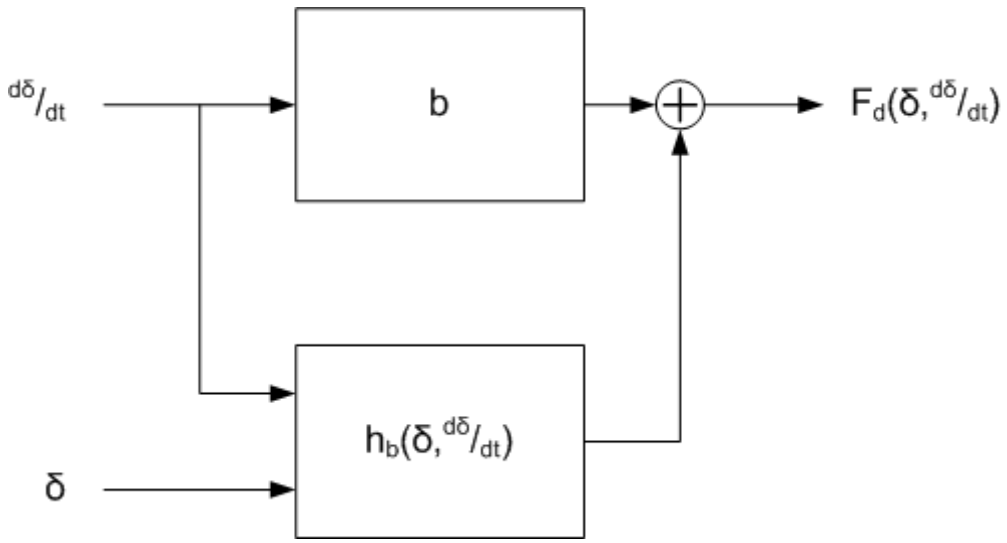

Figure 5.5: Separated Linear and Nonlinear Damper Model

The proposed linear translational model incorporates suspension elements with nonlinear constitutive equations to retain the nonlinearities from the suspension geometry. These nonlinear

suspension elements were characterized by running a virtual K&C test using the two dimensional SimMechanics model and analytically deriving equations from the response data. The nonlinear portions of the suspension were separated from the linear portion and put into a generalized Lur'e formation. To decrease computation time, the nonlinear portions were implemented as lookup tables and the linear portion was integrated into the complete linear translational model so that it could be discretized using a Tustin approximation in Matlab.

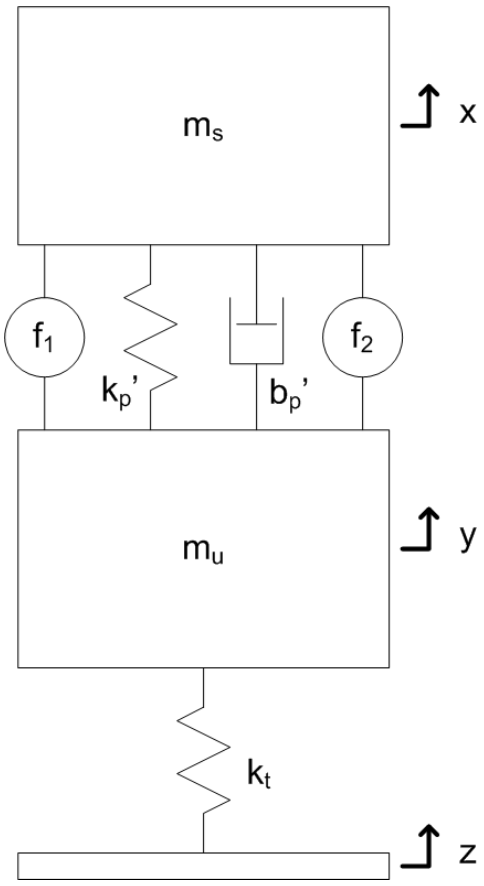

Figure 5.6: Proposed Linear Translational Model with Nonlinear Suspension Elements

$$m_s = sprung\ mass$$
$$m_u = unsprung\ mass$$
$$k_p = linear\ spring\ constant$$
$$b_p = linear\ damper\ constant$$
$$f_1 = nonlinear\ component\ of\ spring\ force$$
$$f_2 = nonlinear\ component\ of\ damping\ force$$

Figure 5.6 shows the linear model with nonlinear suspension elements. f_1 and f_2 are force sources that add to the linear forces from k_p' and b_p'. They were implemented using lookup tables that maps out the nonlinear forces of the spring and damper. The linear model only incorporates a spring for the tire and neglects tire damping. Tire damping was neglected in this model since the contributions of forces from the tire are relatively small. A comparison of the maximum spring and damper tire forces in the SimMechanics model yields peak force values of approximately 4004 N and 586 N respectively, which indicates that the damping in the tire contributes less than 13% of the total tire forces. There are only two masses for the linear model, a sprung mass and an unsprung mass. To augment the mass difference between the DAE model and the linear model, the masses of the upper and lower control arms in the DAE were split evenly into the sprung and unsprung masses of the linear model. This, and the fact that rotational inertias contribute to the linear translational masses, represent an approximation associated with the proposed method.

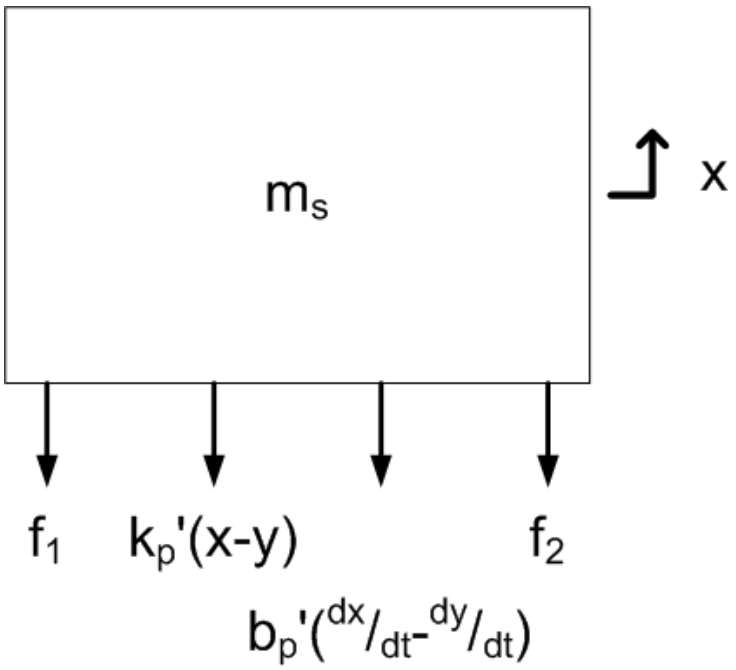

Figure 5.7: Free Body Diagram of Sprung Mass

$$m_s\ddot{x} = -f_1 - f_2 - k_p'(x-y) - b_p'r(\dot{x}-\dot{y}) \tag{5.1}$$

$$\ddot{x} = -\frac{b_p'r}{m_s}\dot{x} - \frac{k_p'}{m_s}x + \frac{b_p'r}{m_s}\dot{y} + \frac{k_p'}{m_s}y - \frac{1}{m_s}f_1 - \frac{1}{m_s}f_2 \tag{5.2}$$

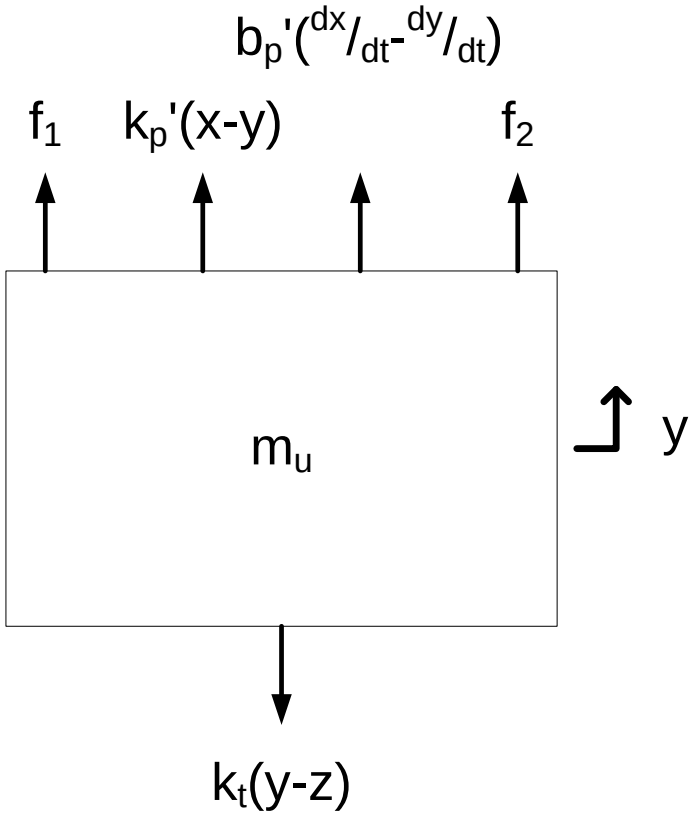

Figure 5.8: Free Body Diagram of Unsprung Mass

$$m_u\ddot{y} = f_1 + f_2 + k_p'(x-y) + b_p'r(\dot{x}-\dot{y}) - k_t(y-z) \tag{5.3}$$

$$\ddot{y} = +\frac{b_p'r}{m_u}\dot{x} + \frac{k_p'}{m_u}x - \left(\frac{b_p'r}{m_u}\right)\dot{y} - \left(\frac{k_p'+k_t}{m_u}\right)y + \frac{1}{m_u}f_1 + \frac{1}{m_u}f_2 + \frac{k_t}{m_u}z \tag{5.4}$$

Figure 5.7 and Figure 5.8 shows the free body diagrams of the sprung and unsprung masses. Using these free body diagrams, the equations of motion for the linear translational dynamic system were derived in Equations 5.1-5.4. Equation 5.5 and 5.6 is the state space representation of the equations of motion.

$$\begin{bmatrix} \ddot{x} \\ \dot{x} \\ \dot{y} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{rb_p{}'}{m_s} & -\frac{k_p{}'}{m_s} & \frac{rb_p{}'}{m_s} & \frac{k_p{}'}{m_s} \\ 1 & 0 & 0 & 0 \\ \frac{rb_p{}'}{m_u} & \frac{k_p{}'}{m_u} & -\left(\frac{rb_p{}'}{m_u}\right) & -\left(\frac{k_p{}'+k_t}{m_u}\right) \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ x \\ \dot{y} \\ y \end{bmatrix} + \begin{bmatrix} -\frac{1}{m_s} & -\frac{1}{m_s} & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_u} & \frac{1}{m_u} & \frac{k_t}{m_u} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ z \end{bmatrix} \quad (5.5)$$

$$\begin{bmatrix} \dot{x} \\ x \\ \dot{y} \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ x \\ \dot{y} \\ y \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ z \end{bmatrix} \quad (5.6)$$

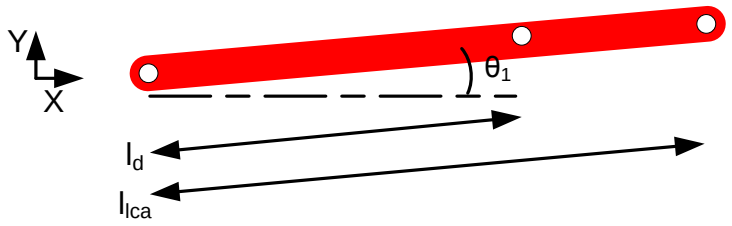

Figure 5.9: Damper and Unsprung Mass Locations on LCA

$$r = \frac{l_d}{l_{lca}} \quad (5.7)$$

Notice that the damping in the Equation 5.5 is multiplied by a factor r. This factor r is a ratio to scale the vertical displacement and vertical velocities of the damper from the damper pickup point on the lower control arm to the unsprung mass. Figure 5.9 shows the distance from the pivot point of the lower control arm to the damper and unsprung mass. These lengths are used to define the ratio r in Equation 5.7.

5.2 Nonlinear Constitutive Equations for Suspension Spring

5.2.1 Analytic Solution

The analytical solution for the nonlinear spring constitutive equations was derived by isolating the problem to a simple geometry case. The spring is assumed to be linear where the force is found by the Equation 5.8. Equations 5.9-5.14 are the analytic equations for the vertical spring force F_{SY}.

$$F_s = k_p \Delta x \tag{5.8}$$

$F_s = spring\ force\ along\ the\ direction\ of\ the\ spring\ axis$
$k_p = spring\ constant$
$\Delta x = change\ in\ spring\ displacement\ along\ the\ spring\ axis$

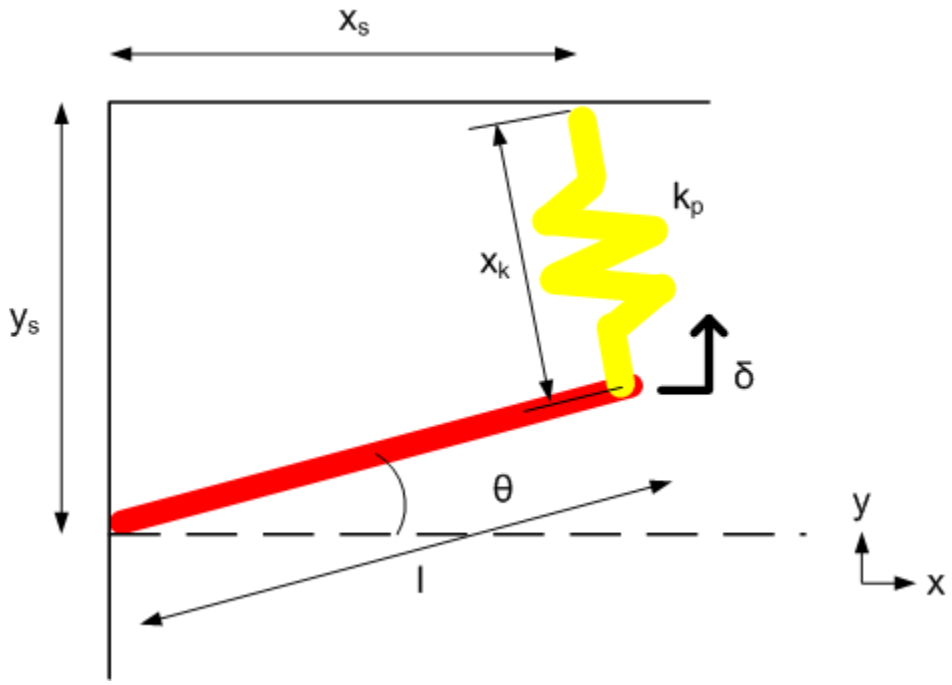

Figure 5.10: Nonlinear Spring Geometry

$x_s = distance\ to\ spring\ pickup\ point\ in\ the\ x\ direction$
$y_s = distance\ to\ spring\ pickup\ point\ in\ the\ y\ direction$
$l = length\ of\ lower\ control\ arm$
$x_k = spring\ length\ along\ the\ spring\ axis$
$k_p = spring\ constant$

$$\delta = vertical\ lower\ arm\ displacement\ in\ the\ y\ direction$$
$$\theta = angle\ of\ lower\ control\ arm\ with\ respect\ to\ horizontal$$

$$\sin\theta(t) = \frac{\delta(t)}{l} \tag{5.9}$$

$$\theta(t) = \sin^{-1}\frac{\delta(t)}{l} \tag{5.10}$$

$$x_k(t) = \sqrt{(y_s - \delta(t))^2 + (l\cos\theta(t) - x_s)^2} \tag{5.11}$$

$$x_0 = \sqrt{(y_s - \delta_0)^2 + \left(l\cos(\sin^{-1}\frac{\delta_0}{l}) - x_s\right)^2} \tag{5.12}$$

$$F_{SY} = F_S\left(\frac{y_s - \delta(t)}{x(t)}\right) = k(x_0 - x)\left(\frac{y_s - \delta(t)}{x(t)}\right) \tag{5.13}$$

$$F_{SY} = k_p\sqrt{(y_s - \delta_0)^2 + \left(l\cos(\sin^{-1}\frac{\delta_0}{l}) - x_s\right)^2} * \left(\frac{y_s - \delta(t)}{\sqrt{(y_s - \delta(t))^2 + (l\cos\theta(t) - x_s)^2}}\right) - k_p\left(y_s - \delta(t)\right) \tag{5.14}$$

First, the angle of the lower control arm θ is defined in terms of the lower control arm displacement in the y direction δ using the trig relation sine in Equation 5.9. The lower control arm displacement in the y direction δ is defined as the distance from the dashed horizontal line to end of the lower control arm. At δ=0, the lower control arm lies along the dashed horizontal line. The length of spring along the spring axis x_k was found using Pythagorean's theorem. The original spring length x_0 is also defined in order to add preload to the spring. The vertical spring force in the y direction F_{SY} in Equation 5.13 was found by taking the y component of the spring force F_s in Equation 5.8. The final vertical spring force F_{SY} is shown in Equation 5.14.

Table 5.1: Parameters for Nonlinear Spring Constitutive Equation

Parameters	Parameter Values
k_p [N/m]	29235
l [m]	0.285
x_s [m]	0.26
y_s [m]	0.115

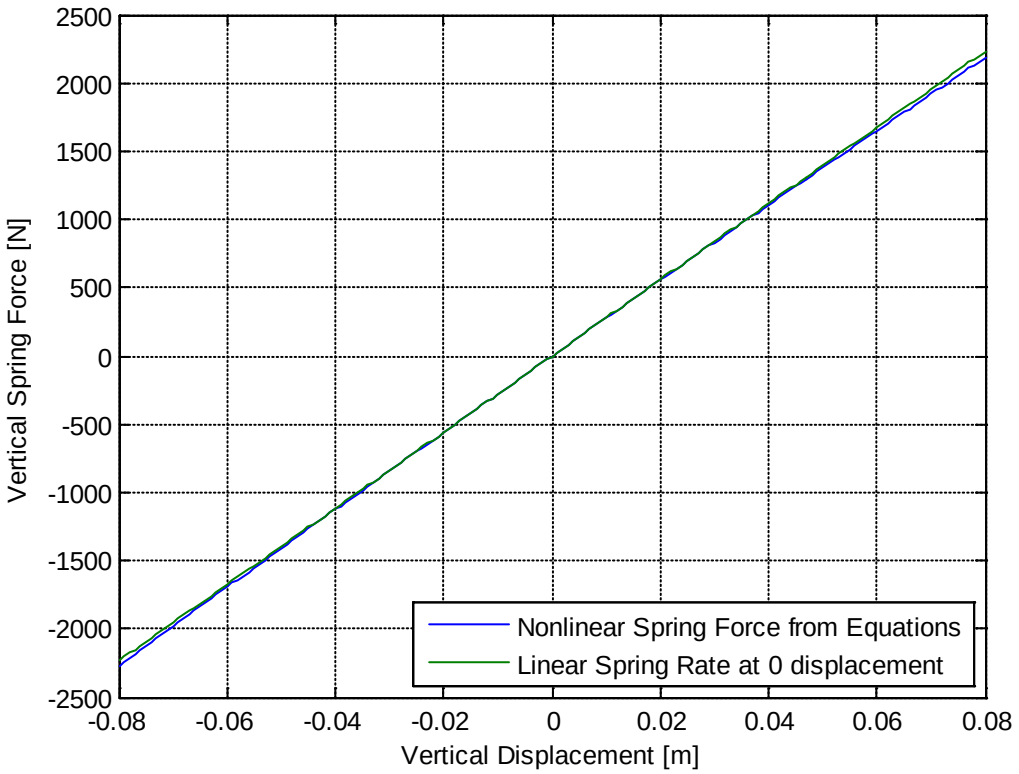

Figure 5.11: Nonlinear Constitutive Equation

A plot of Equation 5.14 is shown in Figure 5.11 using the parameters in Table 5.1. The slope of the nonlinear spring force at a vertical displacement of 0 m was found using the *gradient* function in MATLAB and plotted in green to show the nonlinearity in the spring force. Note that the vertical displacement in the plot is actually the vertical displacement of the spring at its' pickup point location on the lower control arm.

5.2.2 Numerical Solution

A simulated kinematics and compliance (K&C) test was performed in SimMechanics to find the numerical solution to the nonlinear spring force. Essentially, a K&C test locks the sprung mass in place so it has no movement. Then, the wheel actuator moves quasi-statically up and down while measuring force and displacement in the y-direction as well as along the axis of the spring. This test was performed in SimMechanics by replacing the prismatic axis on the sprung mass with a weld joint that holds the sprung mass in place shown in Figure 5.12. The plot of vertical spring force with respect to displacement is shown in Figure 5.13. The vertical displacement data was taken at the joint between the unsprung mass and lower control arm. The spring rate is less in Figure 5.13 than in Figure 5.11 because the vertical displacements on the x axis are measured at a different location. In Figure 5.11, the vertical displacement is measured at the spring's pickup point on the lower control arm, whereas in Figure 5.13, the vertical displacement is taken at the end of the lower control arm at the unsprung mass connection.

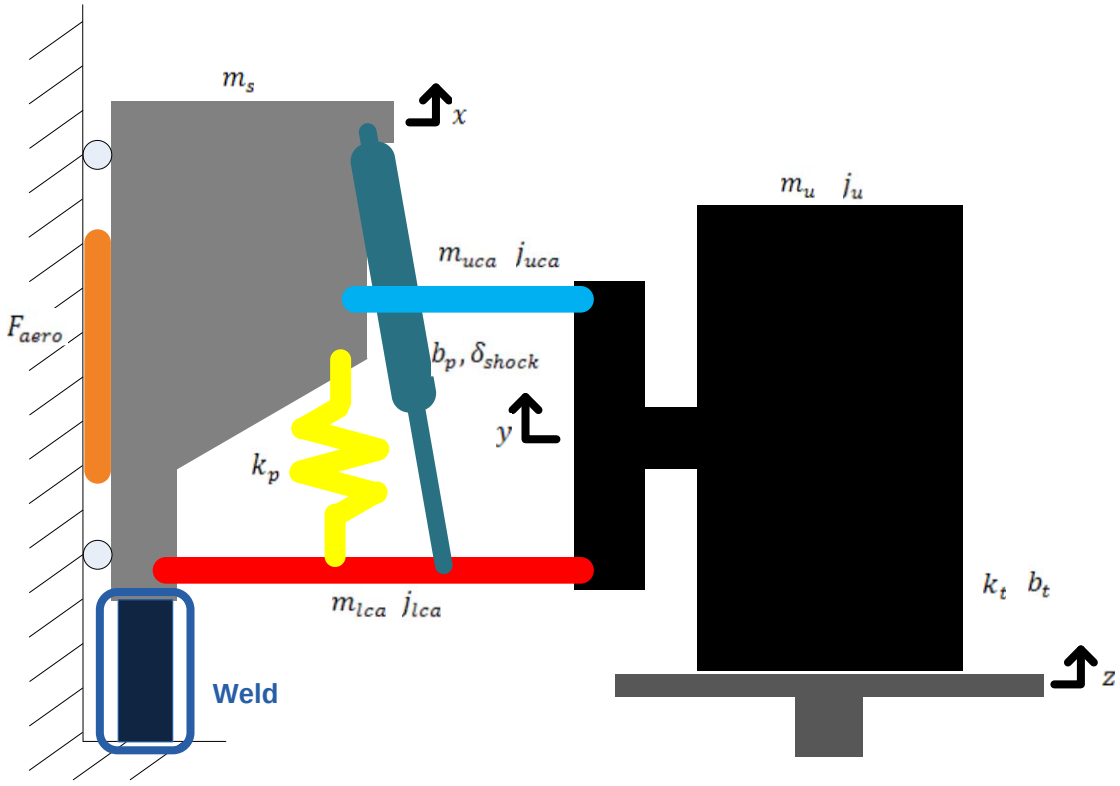

Figure 5.12: K&C Test in SimMechanics

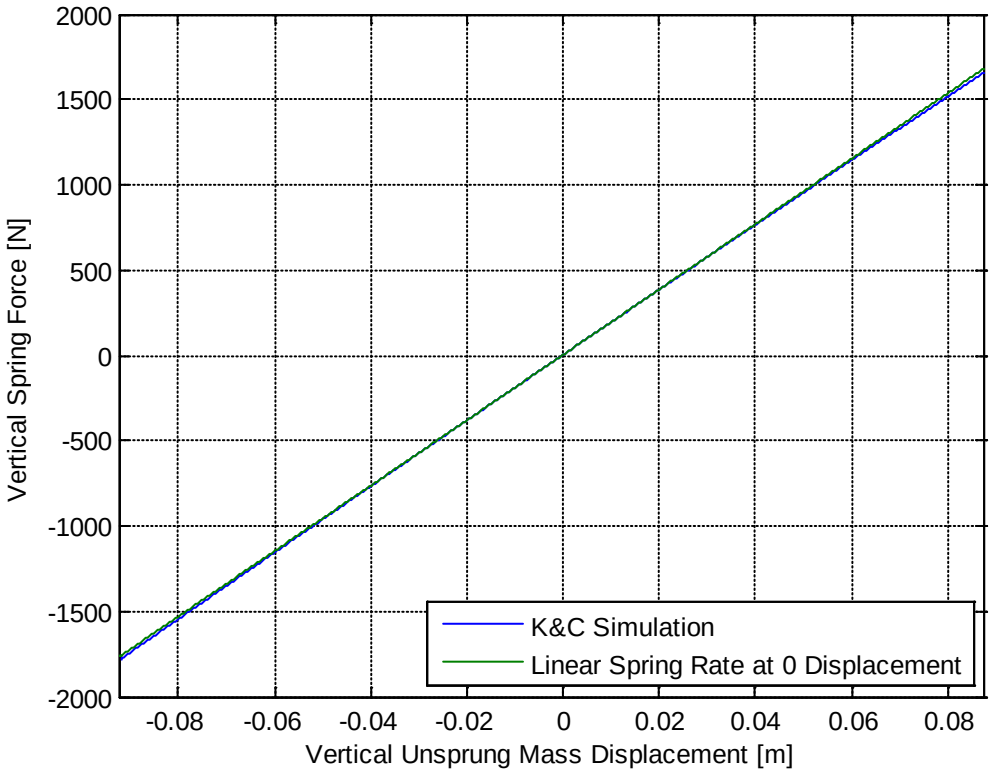

Figure 5.13: Spring Force from Simulated K&C Test in SimMechanics

5.2.3 Table Lookup Implementation

To put this data into a table lookup, the linear and nonlinear portions needed to be separated first. There are many possible solutions for this separation; however for this analysis, the linear portion of the spring rate was determined to be 19175.4 N/m by finding the derivative of K&C simulation data at zero displacement by using the *gradient* function in MATLAB. The linear spring force was then plotted against the K&C simulation data in order to visually distinguish the nonlinearity in the simulation data in Figure 5.13. The nonlinear portion of the spring force shown in Figure 5.14 was found by determining the error between the K&C Simulation data and the linear spring rate at 0 displacement line in Figure 5.13.

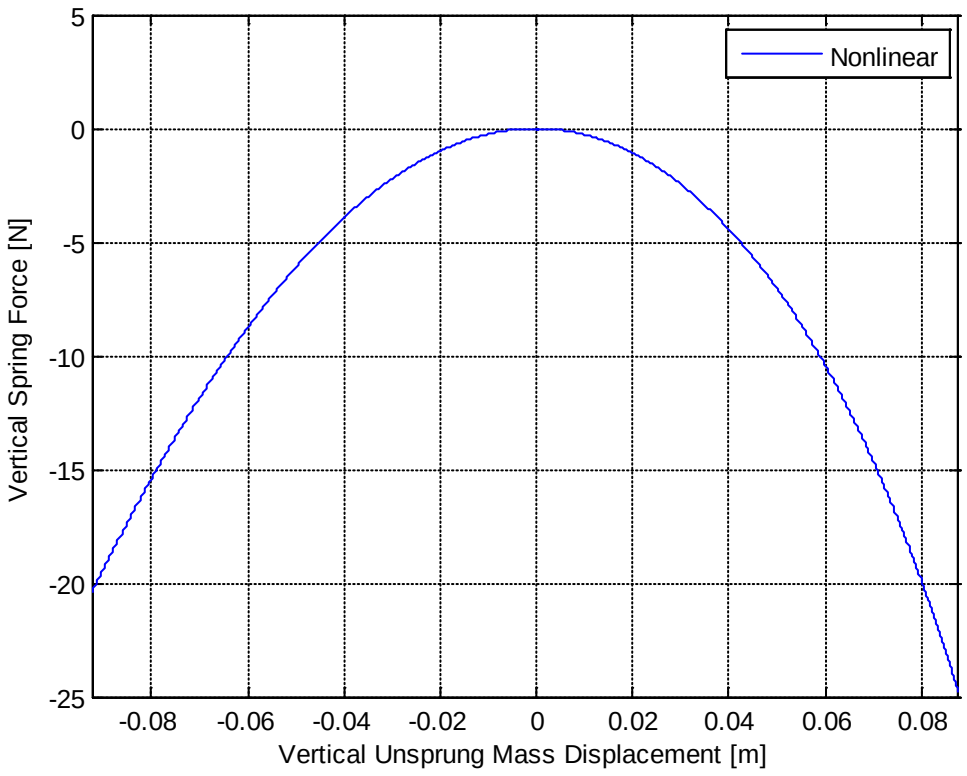

Figure 5.14: Nonlinear Spring Force

The nonlinear data was then put into a table lookup for implementation in the linear model with nonlinear elements. For displacement points that lie in between the discrete points of the table

lookup, the force data is linearly interpolated. By increasing the number of discrete points, the error from linear interpolation can be minimized. It should also be noted that the relative magnitude of the nonlinear portion of the spring force compared to the linear portion of the spring force is very small. A comparison of the maximum values of the nonlinear spring force and the linear spring force yields less than a 2% difference.

5.3 Nonlinear Constitutive Equations for Suspension Damper

5.3.1 Analytic Solution

To develop the nonlinear constitutive equations for the damper, an analytic solution was also derived. The vertical damping force was a function of both displacement and velocity.
A linear damper that follows the Equation 5.15 was assumed for simplicity. The nonlinearities are a result of the suspension geometry. The vertical damping force equations were derived using Figure 5.15. Equations 5.16-5.22 are the resultant equations.

$$F_D = b_p \dot{x} \tag{5.15}$$

$F_D = damper\ force\ along\ the\ direction\ of\ the\ damper\ axis$
$b_p = damper\ constant$
$\dot{x} = change\ in\ spring\ displacement\ along\ the\ damper\ axis$

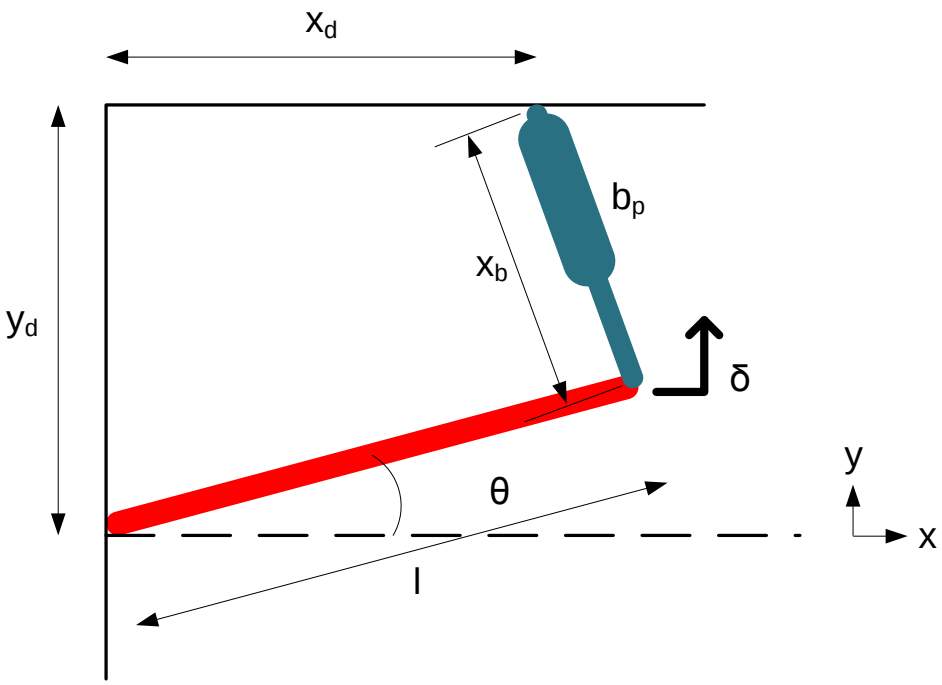

Figure 5.15: Nonlinear Damper Geometry

$$x_d = distance\ to\ damper\ pickup\ point\ in\ the\ x\ direction$$
$$y_d = distance\ to\ damper\ pickup\ point\ in\ the\ y\ direction$$
$$l = length\ of\ lower\ control\ arm$$
$$x_b = spring\ length\ along\ the\ damper\ axis$$
$$b_p = damper\ constant$$
$$\delta = vertical\ lower\ arm\ displacement\ in\ the\ y\ direction$$
$$\theta = angle\ of\ lower\ control\ arm\ with\ respect\ to\ horizontal$$

$$\sin\theta(t) = \frac{\delta(t)}{l} \tag{5.16}$$

$$\theta(t) = \sin^{-1}\frac{\delta(t)}{l} \tag{5.17}$$

$$\dot{\theta}(t) = \frac{\dot{\delta}(t)}{l\cos\theta\,(t)} \tag{5.18}$$

$$x_b = \sqrt{(y_d - \delta(t))^2 + (l\cos\theta(t) - x_d)^2} \tag{5.19}$$

$$\dot{x}_b = -\frac{(y_d - \delta(t))\dot{\delta}(t) + (l\cos\theta(t) - x_d)l\sin\theta(t)\,\dot{\theta}(t)}{\sqrt{(y_d - \delta(t))^2 + (l\cos\theta(t) - x_d)^2}} \tag{5.20}$$

$$F_{DY} = F_D\left(\frac{y_s - \delta(t)}{x(t)}\right) = b\dot{x}(t)\left(\frac{y_s - \delta(t)}{x(t)}\right) \tag{5.21}$$

$$F_{DY} = \frac{b_p(y_d - \delta(t))\left((\delta(t) - y_d)\dot{\delta}(t) + \left(x_d - l\cos\theta\,(t)\right)l\sin\theta(t)\,\dot{\theta}(t)\right)}{(y_d - \delta(t))^2 + (l\cos\theta(t) - x_d)^2} \tag{5.22}$$

Using the trig relation of sine in Equation 5.16, θ was found in terms of vertical lower control arm displacement in Equation 5.17. Equation 5.18 is the derivative of Equation 5.17. The damper length in Equation 5.19 was found using Pythagorean's Theorem. Equation 5.20 is the derivative Equation 5.19. F_{DY} in Equation 5.21 is the vertical force of the damper found by taking the y component of the damper force in Equation 5.15. The final vertical damper force is shown in Equation 5.22.

5.3.2 Numerical Solution

The analytic equations were simulated in MATLAB by generating a matrix of displacement and velocity values as the independent variables and solving for force in the z direction. Figure 5.16 shows the vertical damper force of the damper with respect to the two independent variables, vertical displacement and vertical velocity using parameters from Table 5.2. The force variation is shown using color defined with the colorbar to the right of the plot. There is a slight nonlinearity that can be seen from the plot where the top of the plot, the color lines have a slight positive slope and at the bottom of the plot the color lines has a slight negative slope. To illustrate the curvature in Figure 5.16, a plot of the force versus displacement at deltadot values of +4 m/s and -4 m/s is shown in Figure 5.17. The slope of the force versus displacement plot for deltadot = 4 m/s is positive, and negative for deltadot = -4 m/s. This plot shows that the damper force has a displacement dependence and that dependence is nonlinear since the lines are curved.

The force variation for the displacement shown is approximately 250 N. The nonlinearities can be easily seen in Figure 5.18 by plotting only the nonlinear portion of the damping force.

Table 5.2: Parameters for Nonlinear Damper Constitutive Equations

Parameters	Parameter Values
b_p [N/(m/s)]	2187
l [m]	0.4
x_d [m]	0.285
y_d [m]	0.52

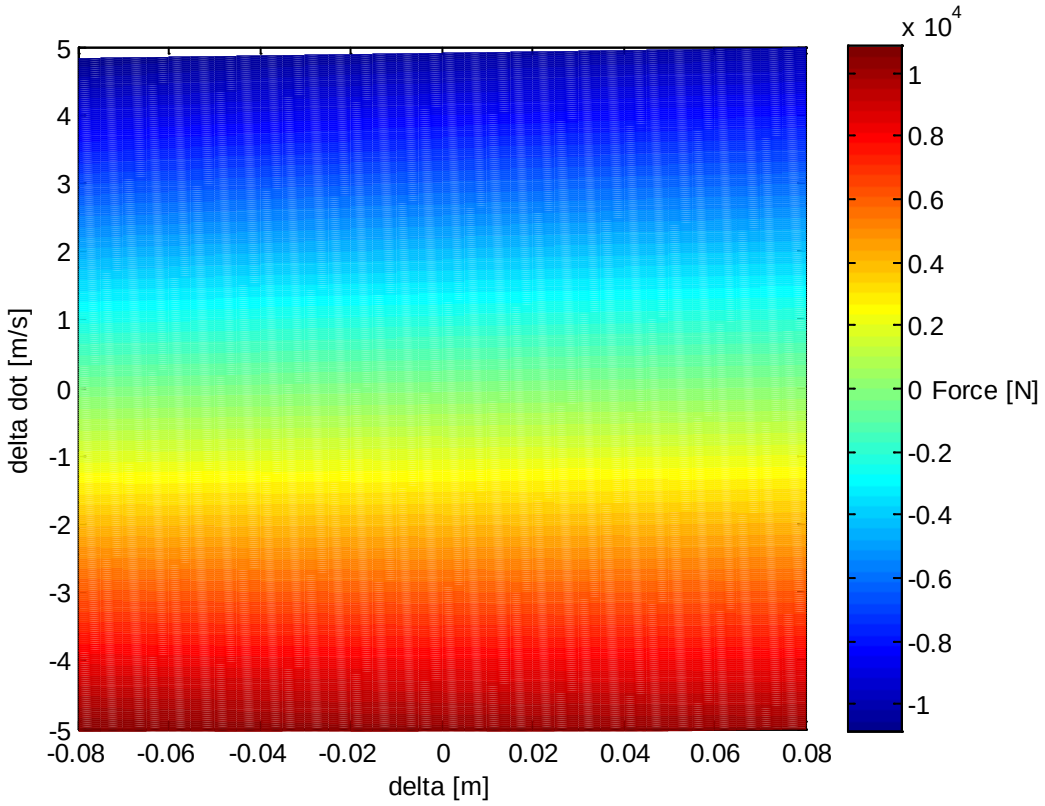

Figure 5.16: Vertical Damper Force with respect to displacement and velocity

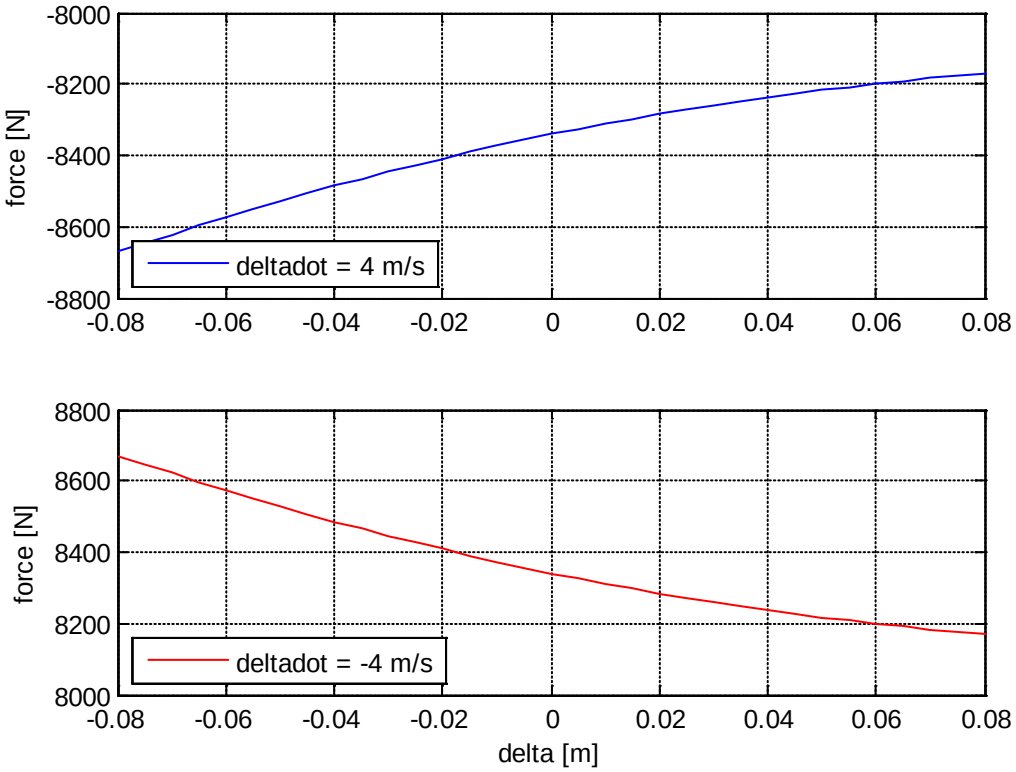

Figure 5.17: Vertical Damper Force at deldot Values of ±4 m/s

5.3.3 Table Lookup Implementation

To put the nonlinear damper forces in a table lookup, the nonlinear and linear portions of the damper needed to be separated. The linear portion of the damping was found to be 2085.3 N/(m/s) by taking the damping value at zero displacement and zero velocity. This was accomplished by finding the slope of the force with respect to velocity when delta was zero using the *gradient* function in MATLAB. A flat plane using the linear damping constant was then created by multiplying the linear damping constant by a range of velocities to obtain a vector of forces, then repeating the vector to obtain a matrix. This plane has no variation in the displacement direction. To find the nonlinear damping forces, the difference between the numerical simulation data and the flat plane was found and plotted in Figure 5.18.

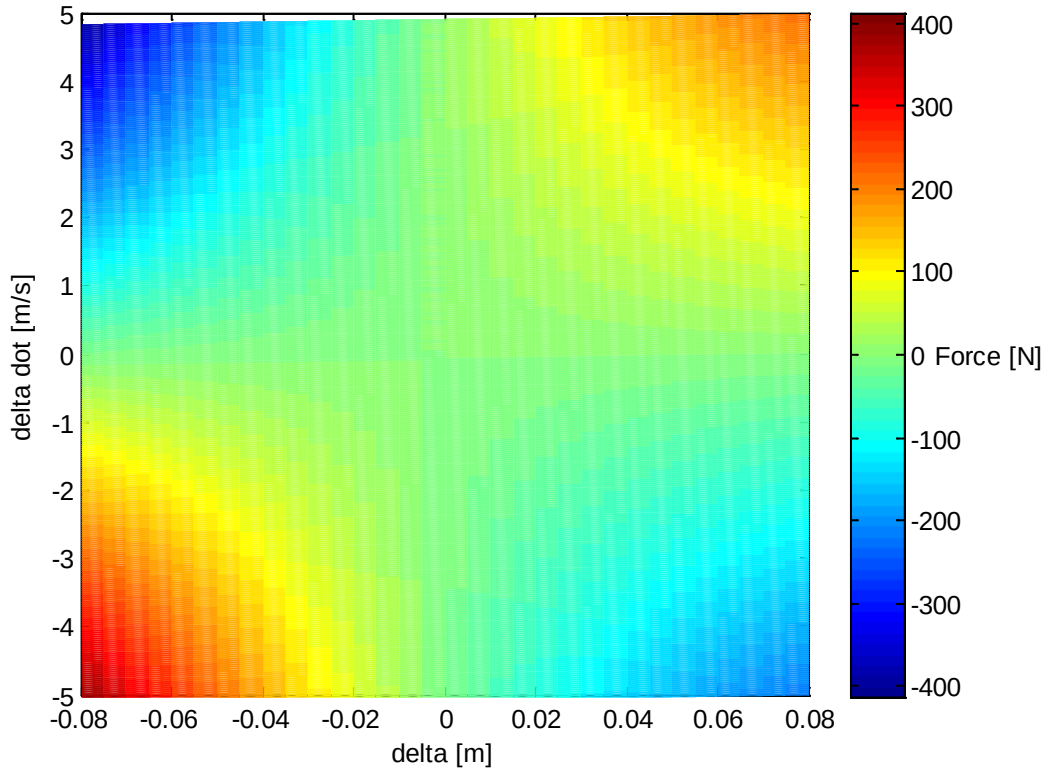

Figure 5.18: Nonlinear Damper Force

Figure 5.18 is not symmetrical with respect to the delta = 0 m line, as expected. As damper angle changes, the vertical portion of the damper force will also change for a specific velocity. Since the damper angle changes due to the vertical displacement of the unsprung mass, vertical damping force has a dependence on the unsprung mass displacement. The magnitude of the force is symmetric across the deltadot = 0 m/s line since negative deltadot values do not change the magnitude of the force in comparison to positive deltadot values, only the sign, from the relation in Equation 5.13.

Implementing this data in a lookup table is similar to the implementation of the nonlinear spring force, with the exception of using a 2D lookup table instead of a 1D lookup table. Two inputs are required to determine the nonlinear force from the lookup table, displacement and velocity. For

displacements and velocities that lie in between the discrete points in the data set, a linear interpolation algorithm is utilized. To reduce the amount of error associated with interpolation, reducing the spacing between displacement and velocity points to increase the size of the matrix can be done.

6 Numerical Validation Study

To validate the suspension models, a numerical study was performed to compare the model responses. The experimental data was first compared to the SimMechanics DAE model after finding the optimized parameters. After the linear model was characterized, the outputs of the model and the SimMechanics DAE model were then compared. The effects of the nonlinear elements were then studied by varying the nonlinear elements. Since the nonlinear components are relatively small compared to the linear part, based on the analysis, you would not expect to see much of a difference between the DAE solution and the proposed model.

6.1 Baseline system study

6.1.1 Comparison of Experimental and SimMechanics Outputs

This study focuses only on Run 1, the high energy wheel input test case. A comparison of the experimental data and the SimMechanics DAE response is shown in Figure 6.1. The experimental response is plotted in blue, the simulation response is plotted in green, and the error between the two signals is plotted in red. The simulation uses the optimized parameters for Run 1 found in Table 4.2.

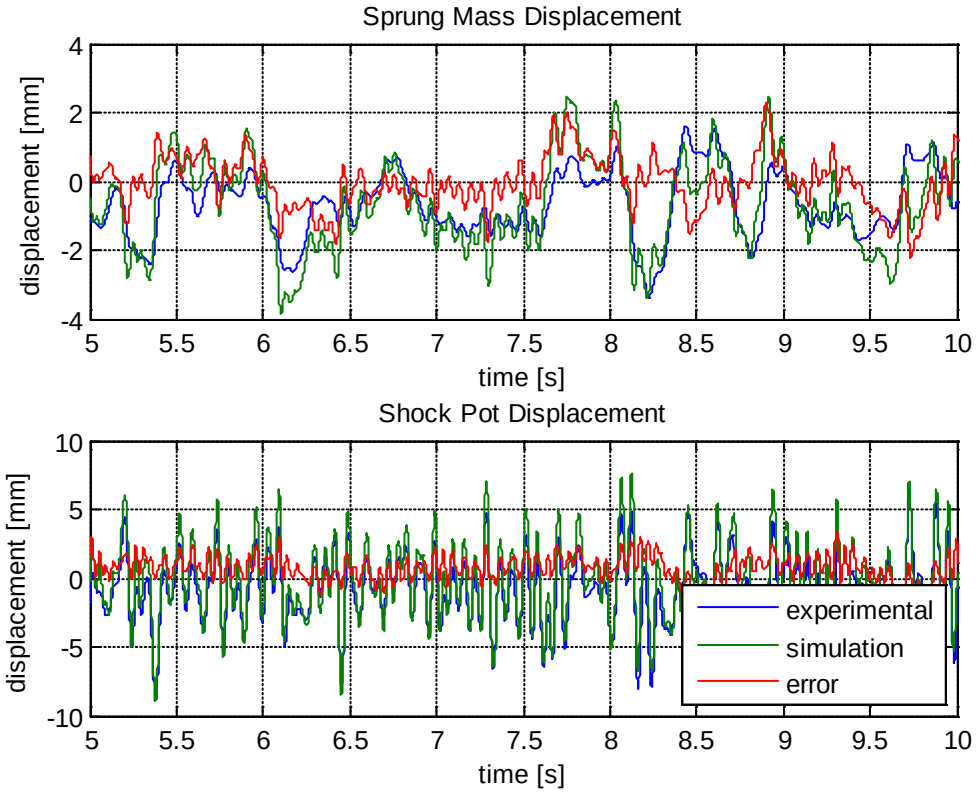

Figure 6.1: Comparison of Optimized Parameter Simulation and Experimental Data for Run 1

The experimental and simulation output correlate well. The simulation response matches the trends in the experimental response quite well. Notice that for the shock displacement, the error lies mainly in the positive region, suggesting that the negative region of the shock displacement matches very well. This may be due to the assumption that the damper is linear when the SimMechanics model was developed. In reality, dampers are nonlinear and behave differently in jounce and rebound. The experimental and simulation responses were not expected to match perfectly as several simplifications were made in the SimMechanics DAE model. For example, the parallel spring and damper used to model the tire accounts for some of the error between the experimental data and the simulation response. In reality, tires are nonlinear and these nonlinearities were not accounted for in this model. Even with these simplifications, the SimMechanics model matches very well with the experimental response.

6.1.2 Comparison of SimMechanics and Linear Model

Figure 6.2 shows a comparison of the outputs between the linear model with nonlinear elements in blue and the SimMechanics model in green. The error between the two signals is shown in red. The two models match fairly well for both the sprung mass displacement and the shock displacement.

The error between the two models might come from various sources. First, the linear model with nonlinear elements is a simplification of the DAE model. The linear model no longer has damping in the tire whereas the DAE model includes damping in the tire. There are also no rotational inertias associated with the masses in the linear model since the model can only travel along the prismatic axis whereas the DAE model includes rotational inertias. The linear model reduces the number of masses to two, a sprung mass and an unsprung mass. The DAE model has four mass bodies, a sprung mass, unsprung mass, upper control arm, and lower control arm. The introduction of these simplifications are most likely the cause for the slight error seen in Figure 6.2.

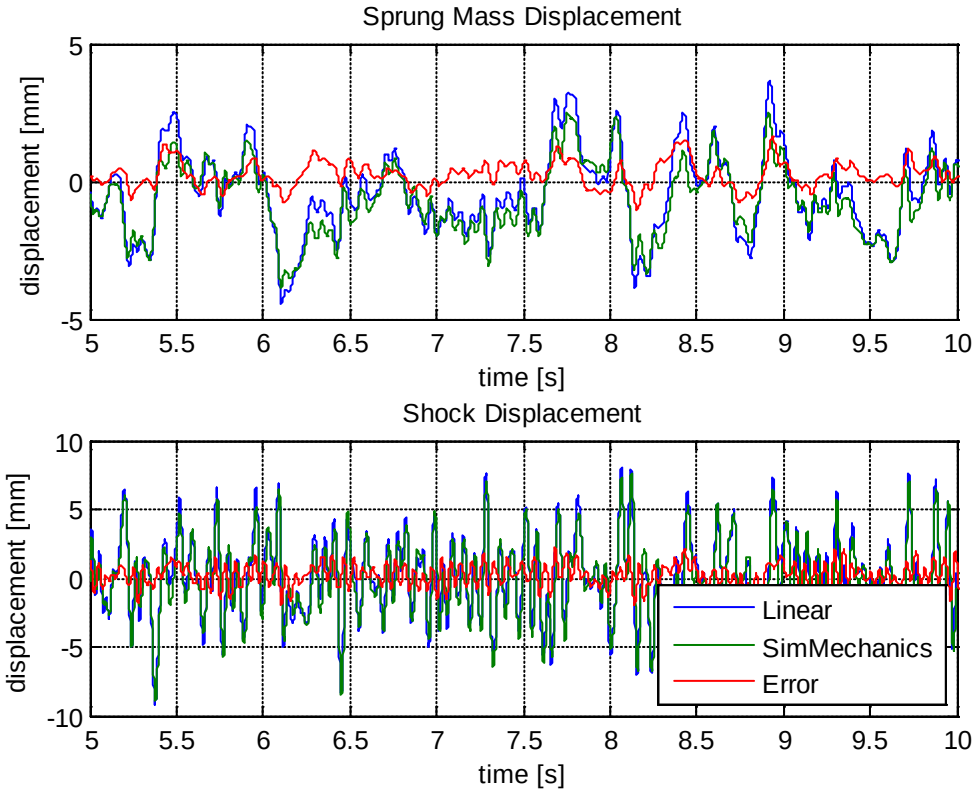

Figure 6.2: Comparison of the Linear Model with Nonlinear Elements and SimMechanics DAE model

The error between the responses of the linear model with nonlinear elements and the SimMechanics model are smaller than that of the experimental response and the SimMechanics model. This shows that many of the simplifications that were made to develop the linear model with nonlinear suspension elements have a relatively small effect on the system response. Neglecting rotational inertias for example does not affect the system response very much.

6.1.3 Comparison of linear models with and without nonlinear spring/damper

To further analyze the error associated with these models, the signal to noise ratio (SNR) is defined in these studies as the ratio between the standard deviation of the signal divided by the

standard deviation of the error between the linear model with nonlinear elements and the SimMechanics model, calculated in decibels. Larger SNR values indicate smaller error between the linear model with nonlinear elements and the SimMechanics DAE model and a better match of the response.

Table 6.1: SNR for Baseline Suspension System Study

Case	Sprung Mass Displacement SNR [dB]	Shock Displacement SNR [dB]	$\sigma^2_{sprung,err}$ $[mm^2]$	σ^2_{sprung} $[mm^2]$	$\sigma^2_{shock,err}$ $[mm^2]$	σ^2_{shock} $[mm^2]$
L Spring, L Damper	9.384	11.614	0.1683	1.4606	0.4471	6.4843
NL Spring, L Damper	9.386	11.615	0.1682	1.4606	0.4471	6.4843
L Spring, NL Damper	9.382	11.614	0.1684	1.4606	0.4471	6.4843
NL Spring, NL Damper	9.384	11.615	0.1683	1.4606	0.4471	6.4843

$$\sigma^2_{sprung\ ,err} = standard\ deviation\ squared\ of\ the\ sprung\ mass\ displacement\ error$$
$$\sigma^2_{sprung} = standard\ deviation\ squared\ of\ the\ sprung\ mass\ displacement\ signal$$
$$\sigma^2_{shock\ ,err} = standard\ deviation\ squared\ of\ the\ shock\ displacement\ error$$
$$\sigma^2_{shock} = standard\ deviation\ squared\ of\ the\ shock\ displacement\ signal$$

Four cases were investigated in Table 6.1; the case of a linear spring and linear damper, a nonlinear spring and linear damper, a linear spring and nonlinear damper, and a nonlinear spring and nonlinear damper. Linear in this case refers to only using the linear portion of the model, and replacing the nonlinear portion with a vector or matrix of zeros.

The numerical SNR analysis above indicates that the nonlinear elements have very little effect in our model. The largest difference in SNR for the sprung mass displacement is only 0.004 dB. For the shock displacement SNR, this difference is only 0.001 dB. The small effect of the nonlinear suspension elements could be due to the design of the suspension geometry or because

the input was not large enough for the nonlinear elements to have a significant effect. The suspension used in this study was modeled from a NASCAR suspension, which was likely designed to have a fairly linear suspension geometry. This study shows that for the input used in this study, the nonlinear suspension elements did not have a significant effect on the model responses.

6.2 Alternative System Study #1 – Higher Wheel Input Amplitudes

Since the results from the baseline study were not definitive in demonstrating our hypothesis, an increase in the wheel input amplitude was proposed. The wheel input signal amplitude was increased by 3 times in comparison to the original high energy wheel input in order to excite the system into a more nonlinear portion of the suspension geometry. This study was performed purely in simulation due to hardware limitations that would have prevented this type of excitation on our test rig.

6.2.1 Comparison of linear models with and without nonlinear spring/damper

Table 6.2 shows the comparison between the four test cases for a higher wheel input amplitude excitation. A comparison of the sprung mass displacement SNR shows that the nonlinear spring and nonlinear damper case has a higher SNR than the linear spring and linear damper case, however, the difference is only 0.004 dB. The case with the nonlinear spring and linear damper had the highest SNR for both the sprung mass displacement and the shock displacement. The case with the linear spring and nonlinear damper had the lowest SNR for both the sprung mass displacement and the shock displacement.

Table 6.2: SNR for Alternative Suspension System Study #1

Case	Sprung Mass Displacement SNR [dB]	Shock Displacement SNR [dB]	$\sigma^2_{sprung,err}$ $[mm^2]$	σ^2_{sprung} $[mm^2]$	$\sigma^2_{shock,err}$ $[mm^2]$	σ^2_{shock} $[mm^2]$
L Spring, L Damper	7.704	10.92	2.3701	13.97	4.7812	59.092
NL Spring, L Damper	7.718	10.925	2.3627	13.97	4.7752	59.092
L Spring, NL Damper	7.694	10.916	2.3756	13.97	4.7853	59.092
NL Spring, NL Damper	7.708	10.922	2.3681	13.97	4.7793	59.092

The results in this study did not definitively demonstrate that the nonlinear spring and nonlinear damper case produces a better approximation of the DAE model. Although the differences in the value of the SNRs are larger, they are not large enough to draw any concrete conclusions. The nonlinear suspension elements have a small effect for the input used in this study. As a result, a second alternative system study was completed.

6.3 Alternative System Study #2– Damper Relocation & New Input

An alternate system was proposed to investigate the effects of a more nonlinear damper geometry. A comparison of the baseline and alternative suspension geometry is shown in Table 6.3 and Figure 6.3. Relocating the actual damper pickup point on the chassis on the experimental rig would be very time consuming. Instead, this study was completed only in simulation utilizing the SimMechanics DAE model to validate the results. The damper pickup point was easily relocated by changing the parameters in the model. The alternative suspension geometry has a shorter x_d location and a shorter lower control arm pickup point than the baseline damper

geometry. This alternative test case was chosen by adjusting the parameters in Table 6.3 and studying the effect on the nonlinear portion of the damper.

Table 6.3: Parameters for Damper Position

Parameters	**Baseline [m]**	**Alternative [m]**
l	0.40	0.32
x_d	0.285	0.20
y_d	0.52	0.52

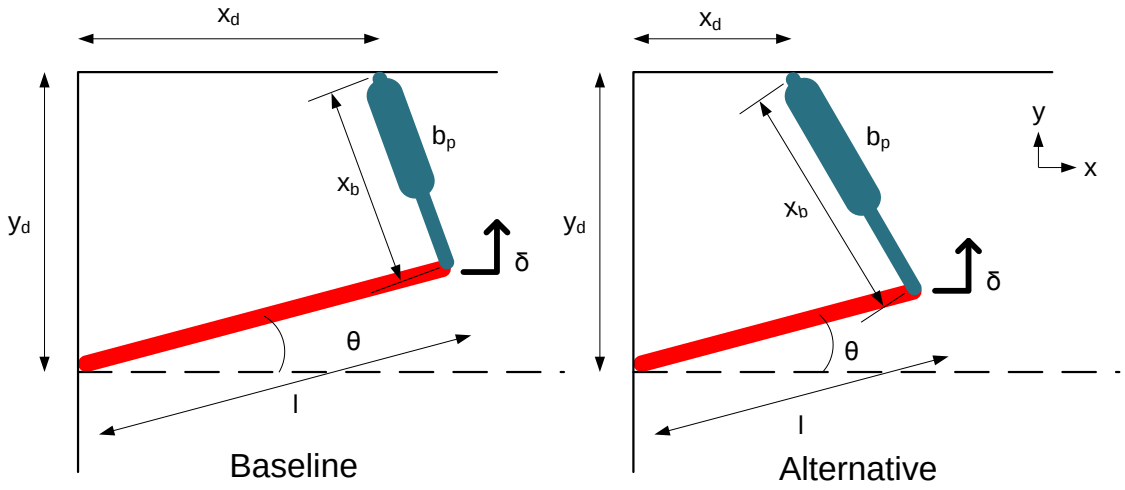

Figure 6.3: Baseline and Alternative Damper Geometry

A new input was proposed that has a lower bandwidth and a larger amplitude. The high energy wheel loader input was filtered with a 4th order low pass Butterworth filter with a cutoff frequency at 5 Hz. The magnitude was then increased by 4 times by multiplying the filtered signal by 4. The resultant signal has a maximum displacement of 18.4 mm and a standard deviation 5.001 mm. Figure 6.4 shows the power spectral density of the proposed wheel loader input signal. The slope of the power spectral density plot becomes less negative at approximately -85 dB possibly due to round off error. The impact from this change in slope is negligible since it is over 40 dB smaller than the magnitude during the passband of the signal.

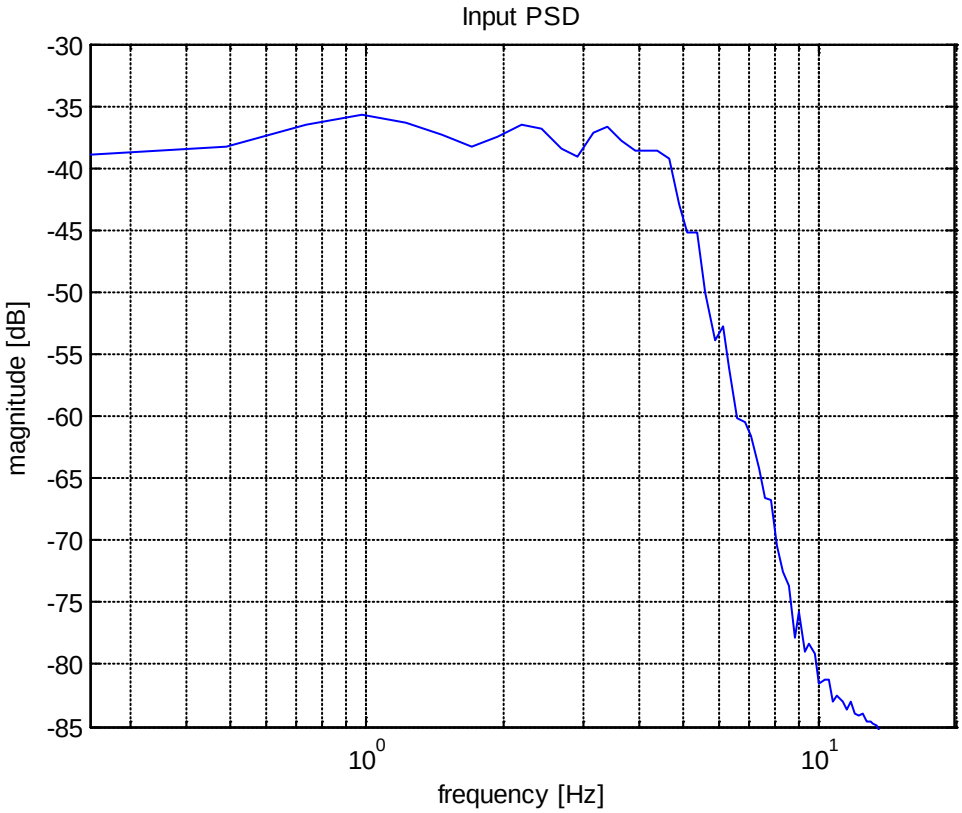

Figure 6.4: Power Spectral Density of 5Hz Wheel Loader Input

6.3.1 Comparison of linear models with and without nonlinear spring/damper

Table 6.4 shows the SNR comparison for the same four cases in Section 6.1.3 but with the alternative damper geometry. The nonlinear spring and nonlinear damper had higher SNRs for both the sprung mass displacement and shock displacement whereas the linear spring and linear damper had the smallest SNR for both displacements.

Table 6.4: SNR for Alternate Suspension System Study

Case	Sprung Mass Displacement SNR [dB]	Shock Displacement SNR [dB]	$\sigma^2_{sprung,err}$ $[mm^2]$	σ^2_{sprung} $[mm^2]$	$\sigma^2_{shock,err}$ $[mm^2]$	σ^2_{shock} $[mm^2]$
L Spring, L Damper	6.176	3.407	5.2317	21.687	6.3555	13.927
NL Spring, L Damper	6.179	3.409	5.2271	21.687	6.3522	13.927
L Spring, NL Damper	6.178	3.411	5.2285	21.687	6.349	13.927
NL Spring, NL Damper	6.182	3.414	5.2239	21.687	6.3457	13.927

The results from this study show that the difference between the SNR s are increasing between the nonlinear spring and nonlinear damper case and the linear spring and linear damper case, however, the results are not conclusive since the difference is still quite small. In order for the damper to have a large nonlinearity, wheel loader displacements and velocities must be large enough to excite the system into those regions. Although the new input has a larger amplitude compared to the excitation for Run 1, the wheel loader velocities may not be high enough to have significant nonlinear damper forces.

6.4 Alternative System Study #3 – Spring Relocation & New Input

Another alternative system is proposed with a more nonlinear spring. Figure 6.5 and Table 6.5 shows a comparison of the baseline and alternative spring geometries. The difference between the two geometries is that the pickup point on the sprung mass in the x direction x_s is now shorter. As a result of the new spring position, the spring is significantly more nonlinear compared to the NASCAR geometry. Figure 6.6 shows the simulated K&C test in SimMechanics for the proposed spring geometry.

Table 6.5: Parameters for Spring Position

Parameters	**Baseline [m]**	**Alternative [m]**
l	0.285	0.285
x_s	0.26	0.04
y_s	0.115	0.115

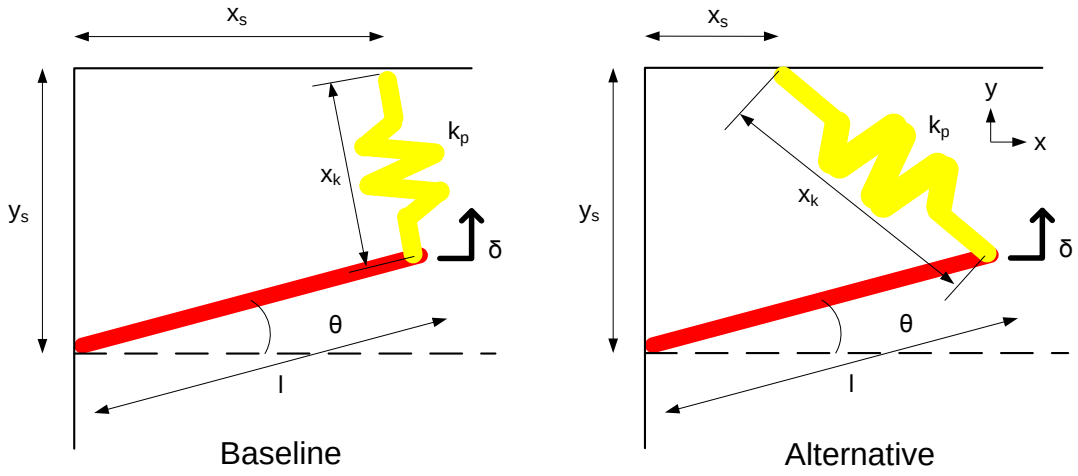

Figure 6.5: Baseline and Alternative Spring Geometry

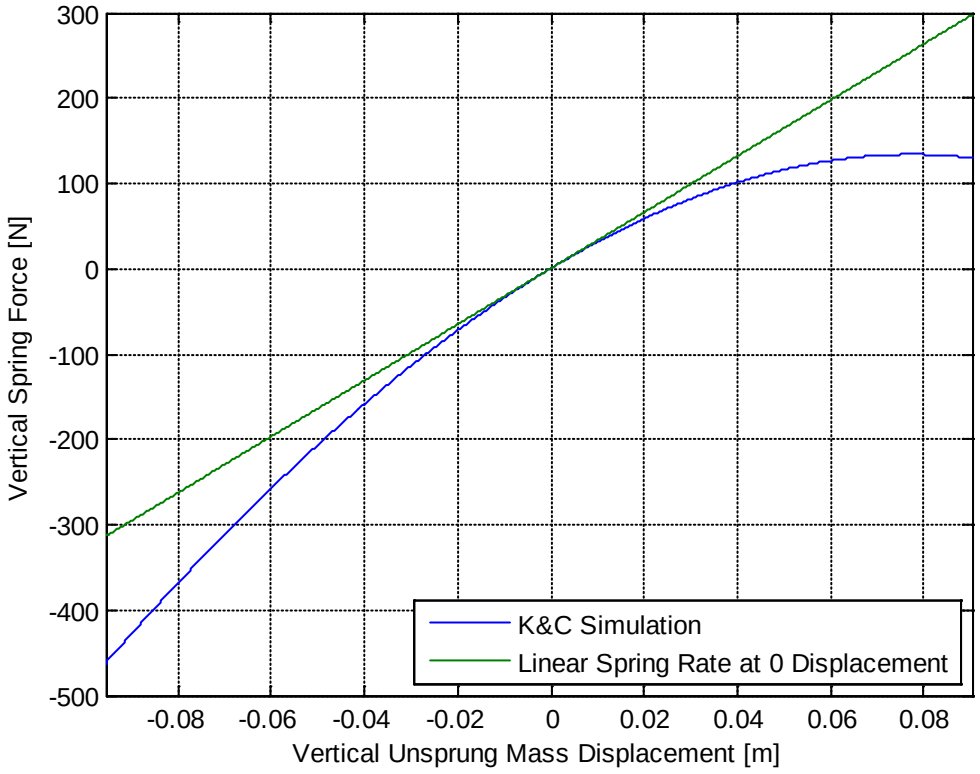

Figure 6.6: Spring Force from Simulated K&C Test in SimMechanics for Alternate Spring Geometry

In order to observe the effects of the nonlinearities on the fidelity of our model, wheel inputs must be large enough to excite the system into the nonlinear range of the suspension. The input used in this study was the 5 Hz filtered high energy wheel input from in Section 6.3, but the magnitude was increased by 25 times.

6.4.1 Comparison of linear models with and without nonlinear spring/damper

Table 6.6 shows the comparison of the SNR for the proposed spring geometry. The nonlinear spring and linear damper had the largest SNR for both displacements. The linear spring and nonlinear damper had the smallest SNR for both displacements. The nonlinear spring and

nonlinear damper has a larger SNR than the linear spring and linear damper by 2.255 dB for the sprung mass displacement and a 1.477 dB increase in shock displacement SNR.

Table 6.6: SNR for Alternative Suspension System Study #3

Case	Sprung Mass Displacement SNR [dB]	Shock Displacement SNR [dB]	$\sigma^2_{sprung,err}$ $[mm^2]$	σ^2_{sprung} $[mm^2]$	$\sigma^2_{shock,err}$ $[mm^2]$	σ^2_{shock} $[mm^2]$
L Spring, L Damper	12.845	12.514	24.829	478.08	28.943	516.35
NL Spring, L Damper	15.076	14.014	14.855	478.08	20.491	516.35
L Spring, NL Damper	12.806	12.493	25.056	478.08	29.086	516.35
NL Spring, NL Damper	15.061	13.991	14.906	478.08	20.598	516.35

The difference in SNR values are sufficiently large enough to conclude that nonlinear suspension elements can have a noticeable impact in comparison to a translational model with linear suspension elements. This analysis shows that the nonlinear spring and nonlinear damper could result in a better match of the SimMechanics and linear model with nonlinear elements responses. How close the models match would be dependent on the suspension geometry and input.

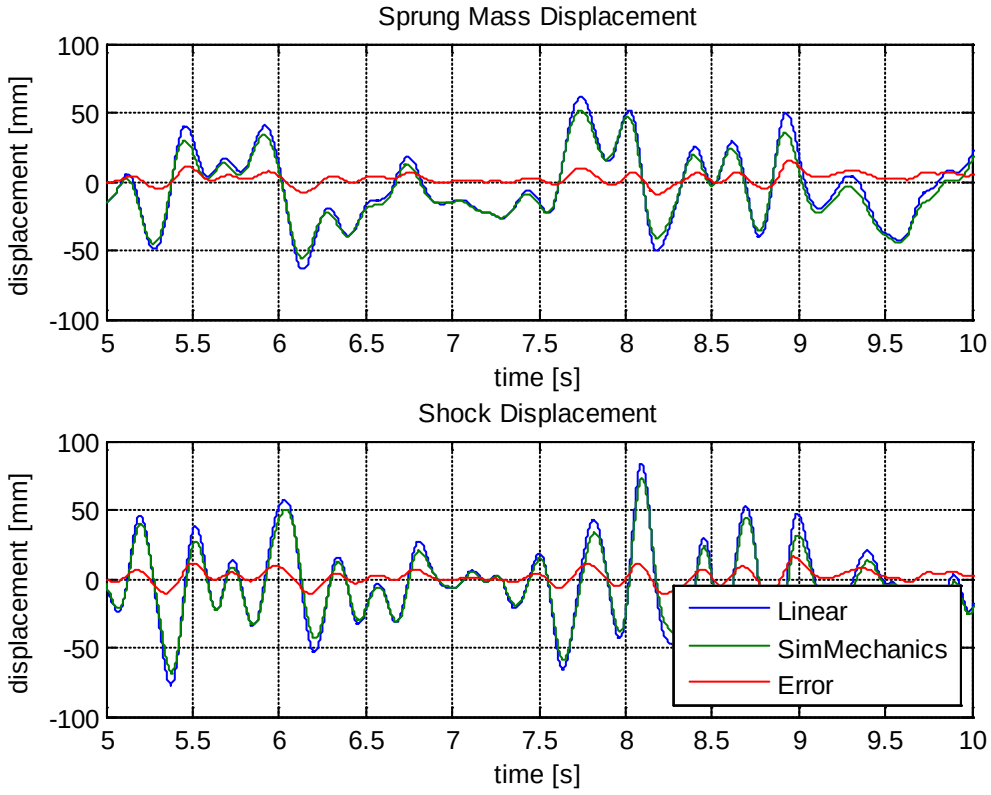

Figure 6.7: Comparison of Linear Model with Nonlinear Elements and SimMechanics Model for Alternate Study #3

Figure 6.7 shows a plot of the responses for the linear model with nonlinear suspension elements and the SimMechanics DAE model for the proposed spring geometry. The error plot in red is significantly smaller than both model responses, resulting in a higher SNR in comparison to the previous studies. Notice that the range of the shock displacement in Figure 6.7 is large enough to be in nonlinear region of the spring shown in Figure 6.6.

6.5 Simulation Time Study

The Matlab function *cputime* was used to calculate the simulation time for the SimMechanics model and the linear model with nonlinear elements. The function *cputime* measures the total

CPU time in seconds that the program in Matlab has used. This study was performed on a Dell Latitude Laptop D830 with 2 GB of RAM and a T7500 2.20 GHz Intel Core2 Duo processor using Matlab R2009b in Windows XP. Table 6.7 shows the comparison of the simulation time.

Time was measured only while the simulation was running and does not include defining and loading variables or post-processing data and visualizations. The linear model with nonlinear elements is 34.74 times faster than the SimMechanics DAE model. The linear model with nonlinear elements takes significantly less computation time.

Table 6.7: Comparison of Simulation Time for Computational Models

Model	**Time [s]**
SimMechanics DAE Model	23.28
Linear Model with Nonlinear Elements	0.67

Although this result is only for one specific simulation study, it is clear that computational savings will be considerably better for the proposed model.

7 Conclusions

The linear model with nonlinear suspension elements can adequately represent the DAE model. In Figure 6.2, a comparison of the linear model with nonlinear elements and the SimMechanics DAE model was shown. The response of the linear model with nonlinear elements was very close to the response of the SimMechanics DAE model. Although the linear model with nonlinear elements cannot perfectly replace the DAE model, it can achieve a reasonably accurate response.

For the range of motion of this test, the nonlinear portion of the linear model had a small effect, as predicted during the development of the model. A comparison of the relative magnitude of the spring force in Figure 5.13 and Figure 5.14 shows that the nonlinear portion was quite small in comparison to the linear portion. Figure 5.16 and Figure 5.18 shows that the nonlinear damper force is relatively small compared to the linear portion of the damper.

The alternate suspension geometry study in Section 6.4 showed that the nonlinear elements could decrease the error between the linear model and the DAE model. Whether the nonlinear elements increase the accuracy of the model depends on the suspension geometry.

The use of the Lur'e Formulation allows us to discretize the model and run a simulation much faster. Because the linear and nonlinear portions are separate, the linear portion can be discretized. The nonlinear portion is chosen to be a lookup table, which does not require much computational effort.

The linear model with nonlinear suspension elements developed in this research is suitable to running on a driving simulator. Table 6.7 showed that the linear model with nonlinear elements had a significantly lower simulation time in comparison to the DAE model. Due to the low computational requirements, this model is appropriate for operation in a real-time environment.

References

1. Huang ARW, Chihsiuh C, 2003, A low-cost driving simulator for full vehicle dynamics simulation, *Vehicular Technology, IEEE Transactions on*, pp. 162-172.
2. Yoshimoto K, Suetomi T, 2008, The History of Research and Development of Driving Simulators in Japan, *Journal of Mechanical Systems for Transportation and Logistics*, pp. 159-169.
3. Sauikat R. The New Dynamic Driving Simulator at DLR. Driving Simulation Conference. Orlando, FL; 2005.
4. Kemeny A. Virtual testing with driving simulators. Driving Simulation Conference. Europe, Monaco; 2008.
5. Chapron T, Collinnot JP. The new PSA Peugeot-Citroën Advanced Driving Simulator : Overall design and motion cue algorithm. Proc Driving Simul Conference. Iowa, US; 2007.
6. Suda Y, Shiiba T, Tanabe Y, Ounki M. Proposal of a tyre evaluation system with driving simulator under actual driving conditions. Vehicle System Dynamics: Taylor & Francis Ltd; 2005. p 475-482.
7. Bella F, 2008, Driving simulator for speed research on two-lane rural roads, *Accident Analysis & Prevention*, pp. 1078-1087.
8. Bella F, 2007, Parameters for Evaluation of Speed Differential: Contribution Using Driving Simulator, *Transportation Research Record: Journal of the Transportation Research Board*, pp. 37-43.
9. Heydinger GJ, Salaani MK, Garrott WR, Grygier PA. Vehicle dynamics modelling for the National Advanced Driving Simulator. Proceedings of the Institution of Mechanical Engineers -- Part D -- Journal of Automobile Engineering: Professional Engineering Publishing; 2002. p 307-318.
10. Salaani MK, Heydinger, G. J. , 2000, Model Validation of the 1997 Jeep Cherokee for the National Advanced Driving Simulator, *SAE Paper No 2000-01-0700*, pp.
11. Shiiba T, Suda Y, 2007, Evaluation of driver's behavior with multibody-based driving simulator, *Multibody System Dynamics*, pp. 195-208.
12. Shiiba T, Suda Y. Performance of Developed Driving Simulator with Full Vehicle Model of Multibody Dynamics. The International Mechanical Engineering Congress and Exposition. New York, NY; 2001. p pp. 27-32.
13. Shiiba T, Suda Y. Real-Time Multibody Analysis Environment for Driving Simulator. International Design Engineering Technical Conferences & Computers and Information In Engineering. Long Beach, CA; 2005.
14. Bae DS, Lee JK, Cho HJ, Yae H, 2000, An explicit integration method for realtime simulation of multibody vehicle models, *Computer Methods in Applied Mechanics and Engineering*, pp. 337-350.
15. Eichberger A, Rulka W. Process Save Reduction by Macro Joint Approach: The Key to Real Time and Efficient Vehicle Simulation. Vehicle System Dynamics: Taylor & Francis Ltd; 2004. p 401-413.
16. Kim S-S, Jeong W, 2007, Subsystem synthesis method with approximate function approach for a real-time multibody vehicle model, *Multibody System Dynamics*, pp. 141-156.

17. Kim S-S, 2002, A Subsystem Synthesis Method for Efficient Vehicle Multibody Dynamics, *Multibody System Dynamics*, pp. 189-207.
18. Kim S-S, Jeong W, 2009, Real-time multibody vehicle model with bush compliance effect using quasi-static analysis for HILS, *Multibody System Dynamics*, pp. 367-382.
19. Choi G, Yoo Y, Lee K, Yoon Y. Vehicle Modeling Methods for Real-time Dynamic Simulation Using Suspension Composite Joints*. Mechanics of Structures & Machines: Marcel Dekker Inc.; 2000. p 303.
20. Andersen E. Multibody Dynamics Modeling and System Identification for a Quarter-Car Test Rig with McPherson Strut Suspension Blacksburg, Virginia Virginia Polytechnic Institute and State University 2007
21. Langdon J. Design and Adaptive Control of a Lab-based, Tire-coupled, Quarter-car Suspension Test Rig for the Accurate Re-creation of Vehicle Response. Danville, VA: Virginia Polytechnic Institute and State University. 2007.
22. Ziegenmeyer J. Estimation of Disturbance Inputs to a Tire Coupled Quarter-car Suspension Test Rig Blacksburg, Virginia Virginia Polytechnic Institute and State University 2007.