ADVANCED MULTIBODY DYNAMICS MODELING OF THE FREIGHT TRAIN TRUCK SYSTEM

by

Brent S. Ballew

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

Approved:

Corina Sandu, Chairperson

Mehdi Ahmadian, Member

Dennis W. Hong, Member

April 28, 2008

Blacksburg, Virginia

Keywords: Bogie, friction wedge, multibody dynamics, unilateral contact, train cab suspension, bolster, side frame, freight train

ADVANCED MULTIBODY DYNAMICS MODELING OF THE FREIGHT TRAIN TRUCK SYSTEM

By

Brent S. Ballew

Corina Sandu, Chairperson
Mechanical Engineering

ABSTRACT

Previous work in the Railway Technology Laboratory at Virginia Tech focused on better capturing the dynamics of the friction wedge, modeled as a 3D rigid body. The current study extends that work to a half-truck model treated as an application of multibody dynamics with unilateral contact to model the friction wedge interactions with the bolster and the sideframe. The half-truck model created in MATLAB is a 3D, dynamic, multibody dynamics model comprised of four rigid bodies: a bolster, two friction wedges, and a sideframe assembly. The model allows each wedge four degrees of freedom: vertical displacement, longitudinal

displacement (between the bolster and sideframe), pitch (rotation around the lateral axis), and yaw (rotation around the vertical axis). The bolster and the sideframe have only the vertical degree of freedom. The geometry of these bodies can be adjusted for various simulation scenarios. The bolster can be initialized with a pre-defined yaw (rotation around the vertical axis) and the sideframe may be initialized with a pre-defined pitch/toe (rotation around the lateral axis). The multibody dynamics half-truck model simulation results have been compared with results from NUCARS®, an industry standard train modeling software, for similar inputs.

The multibody dynamics models have also been extended to a variably damped full-truck model and a variably damped half-truck warping model. These models were reformulated to react dynamically to simulated truck warp inputs. The ability to better characterize truck warping properties can prevent train roll over and derailments from truck hunting.

In a quarter-truck variably damped configuration the effects of a curved wedge surface has also been explored. Actual friction wedges have surfaces which are slightly curved, this iteration in the multibody dynamics friction wedge modeling attempts to draw one step closer to actual friction wedge geometry. This model lays the ground work for a contact dependant wedge wearing model based on material properties and tribology.

Acknowledgments

I would like to thank my advisor, Dr. Sandu for her endless support and mentoring throughout this project and graduate school at Virginia Tech. I would also like to thank Nick Wilson and Curt Urban, of TTCi, for their vast knowledge and willingness to lend a helping hand in the development of this project. I would like to thank my committee members, Dr. Mehdi Ahmadian and Dr. Dennis Hong, for their support with this project as well.

I would also like to thank my family and friends. Without their support throughout my life none of this would have been possible.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The first section of this chapter discusses the motivation behind the project. The second section details previous work on the project completed at Virginia Tech. The next section outlines the research approach taken for this phase of the project. The background section provides a brief overview of freight train suspension technology and terminology. The literature review section provides an overview of the technical papers reviewed for this project. The last section is a summary of this thesis.

1.1. Motivation

Freight train suspensions have seen little changes in design since the early 1900s. With the lack of change in freight train suspension design and computing power, the creep equations derived by Carter in 1916 were considered sufficient for analyzing train suspension dynamics until the early 1950s. At this time, engineers and train dynamicists realized the importance of suspension stiffness on train dynamics. This transformed the solution to the equations of motion from Carters simplistic coupled second order linear differential equations into a system with many degrees of freedom and great complexity [1].

As computing power grew over the years so did the complexity of the equations of motion. One key element of the suspension, the friction wedge, is commonly treated as a simplistic force element. This assumption is made because the mass of the friction wedge is negligible as compared to other suspension components. A massless friction wedge model simplifies a three piece bogie model because friction wedge dynamics are difficult to accurately capture in a simulation environment due to their load sensitive behavior and non-linear friction characteristics.

Recent pushes in research tend to suggest modeling the friction wedge as a dynamic body will increase modeling accuracy. A multibody approach to friction wedge modeling offers an opportunity to capture friction wedge dynamics while including inertial properties. Using a multibody dynamics approach to wedge modeling allows for the study of forces and moments associated with critical train suspension events such as wedge lock up and warping. This study intends to use a multibody dynamic modeling approach to increase the accuracy of the current friction wedge model.

1.2. Background

The principal difference between railway vehicles and other types of wheeled vehicles is the guidance provided by the rails. The train suspension supports the car body and absorbs vibrational energy as well as guiding the train in both lateral and longitudinal directions. Three piece bogies consist of two sideframes connected by a bolster. A diagram of a complete train suspension can be seen in Figure 1-1. The primary suspension connects the sideframes to the axle and wheel assembly. Each sideframe is connected to the bolster by a central coil spring (load coil) and a friction wedge suspension system supported by smaller coil springs (control coils). The secondary suspension (friction wedges and load coil) provide damping in both the vertical and lateral directions.

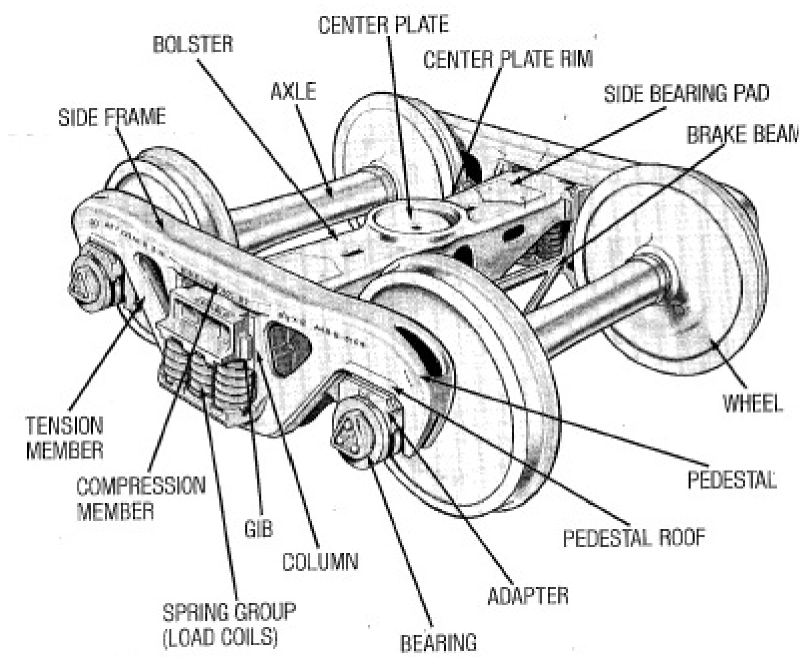


Figure 1-1. Commonly used three piece truck train suspension [2].

The damping from the secondary suspension is provided from friction between the bolster - wedge and sideframe - wedge interactions. The configuration of the column face of the sideframe dramatically affects the amount of frictional damping provided by the wedges. The sideframe can be configured as toed out or toed in. Figure 1-2 shows the toe-out and toe-in configurations.

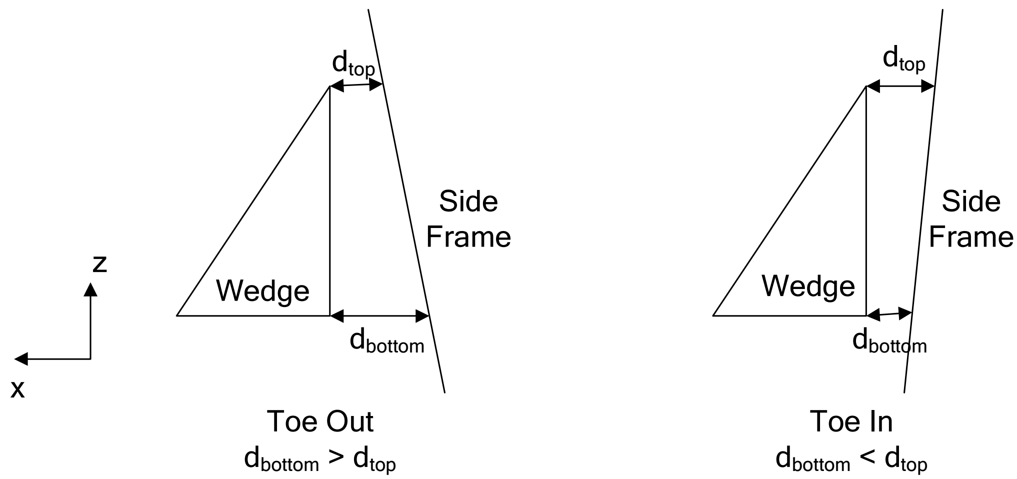


Figure 1-2. Toe-out and toe-in sidefame configurations.

The secondary suspension components can be arranged as a variably damped or constantly damped truck. The main difference between these arrangements is the way the friction wedges are attached to the secondary suspension. In a variably damped truck the control coils are attached to the sideframe. In this arrangement the damping applied by the control coil varies as the coil is compressed by the bolster downward displacement. A representation of a variably damped truck is shown in Figure 1-3.

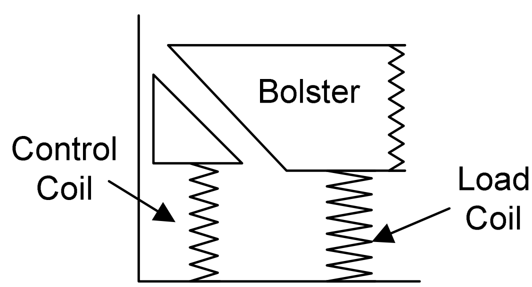


Figure 1-3. Representation of a variably damped truck.

A constantly damped truck contains a bolster pocket where the friction wedge and control coil are connected to the bolster. The bolster pocket allows the wedge relatively little vertical movement as compared to a variably damped truck. The small vertical control coil displacements result in damping that is relatively constant. A representation of a constantly damped truck can be seen in Figure 1-4.

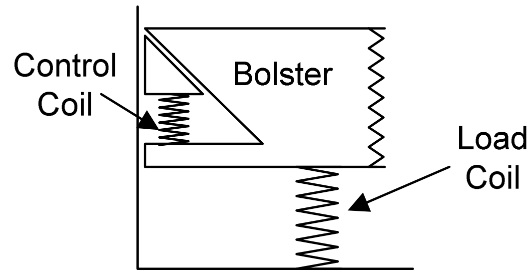


Figure 1-4. Representation of a constantly damped truck.

1.3. Previous Work

The first phase of this project required evaluating the current model used in the railroad industry[2]. Due to Virginia Tech being one of the universities affiliated with the Association of American Railroads (AAR), the train simulation software NUCARS® developed by the Transportation Technology Center, Inc. (TTCI) was made available for use in this project [3]. The friction wedge models used in this software, which are called types 6.8 and 6.9, were evaluated. The capabilities of the current friction wedge model were determined by running several simulations with different inputs and train suspension models (eg. Variably damped suspension).

The second phase of the project involved the creation of a multibody dynamics numerical friction wedge model. The model was designed using MATLAB with numerical integration and graphical visualization performed in Mambo [4]. Mambo is a freeware educational multibody dynamics program used at Virginia Tech. The Mambo program consists of a toolbox used in MATLAB and an independent visualization and integration program. A screen shot from the Mambo visualization can be seen in Figure 1-5.

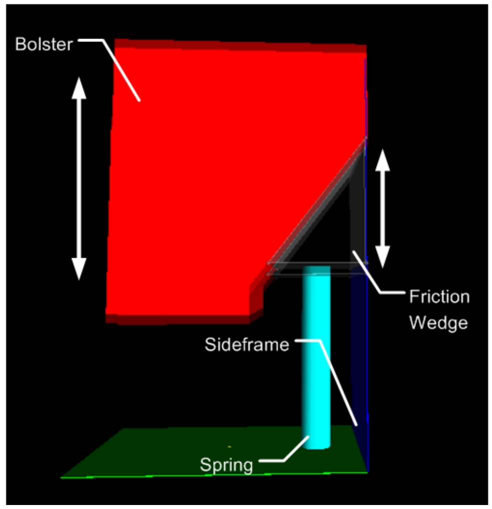


Figure 1-5. Screen shot of quarter-truck variably damped multibody dynamics model.

The multibody dynamics model was designed to be comparable with a simplified quarter-truck model created in NUCARS. The quarter-truck model consists of a sideframe, bolster, and friction wedge with supporting control coils. The study was expanded to include a variably damped truck and a constantly damped truck models comparable with similar NUCARS models.

The quarter-truck models were an initial attempt at friction wedge modeling within the multibody dynamics framework. The models provided insightful information about the friction wedge dynamics, but lacked the entire components of an actual three piece truck. The lack of components limits the multibody dynamics model to be compared only with the simplified NUCARS models.

1.4. Research Approach

Detailed in this thesis are the latest iterations in the multibody dynamics friction wedge modeling. The objectives of this thesis are as follows

1. Expand the quarter-truck variably damped and constantly damped models to half-truck models.
2. Benchmark multibody dynamics models with equivalent NUCARS models
3. Use the variably damped half-truck model to investigate truck warping
4. Develop a friction wedge model that includes the effects of wedge face curvature
5. Expand half-truck variably damped model to a half-truck model

The quarter-truck variably damped and constantly damped models have been expanded to half-truck models. The half-truck models include a sideframe, bolster, and two friction wedges. These models were evaluated with comparable models created in NUCARS. In addition to the half-truck variably damped multibody dynamics model, two modified versions were created. One version investigates the forces and stresses on reacting wedge surfaces, while the second version simulates truck warping behavior.

The quarter-truck variably damped multibody dynamics model was also further developed to investigate the effects of wedge surface curvature on wedge dynamics. The existing multibody dynamics model was modified to include a curved wedge surface in an attempt to draw closer to actual friction wedge geometry.

The half-truck variably damped model has been extened to a full-truck version. The full-truck variably damped model consists of a full bolster, two sideframes, and four friction wedges. New formulations for bolster movement were created allowing additional translational and rotational degrees of freedom.

1.5. Review of Literature

How Bogies Work by Isao Okamoto

In this paper, Okamoto defined the role of a railroad bogie in detail and discussed possible different configurations. Bogies are classified into types first by the number of axles in their configuration and the design of the suspension. The two axle bogie is the most common type found in rail vehicles and in the three-piece bogie. The suspension of the bogie is classified as either articulated or non-articulated. An articulated suspension is one that is located between two car bodies, holding the backside of one and the front side of the following car. A non-articulated suspension requires two separate trucks to support each end of one rail car. A Swing Hanger Bogie and a Small Lateral Stiffness Bolster Spring Bogie are two types of suspension designs which absorb rolling motion of the rail vehicle.

Bolster and bolster-less bogies are another way to differentiate the suspension. The bolster bogie has a solid bolster which is the third piece in a three-piece bogie and connects the sideframes. The bolster-less bogie has a center plate and 2 separate suspensions on the sideframes to support the rail vehicle. This paper also discusses the key elements of a bogie, which include the suspension gear, the bogie frame, the axle box suspension, wheels, axles, bearings, transmission and brakes. Some recent improvements include a tilting bogie, which tilts the rail vehicle toward the center of the circle when turning. Another improvement is the steering bogie which allows each of the axles on a bogie to steer along a rail separately from the other [5].

Modeling Friction Wedges, Part I: The State-of-the-Art by Peter E. Klauser

This paper discussed how the friction wedge is currently being modeled in train simulation software. Klauser discusses the two dimensional friction wedge model used in both NUCARS® and VAMPIRE® [6]. The current friction wedge model allows for translation in the vertical and lateral directions, which allows for the friction damping forces to exist in both directions. The "state-of-the-art" model also allows the user the option of having two different friction coefficients on each face of the wedge, the slope and column faces. Klauser also discusses how the model used in VAMPIRE compares to the model used in NUCARS. The results of this paper was a list of the shortcomings of both VAMPIRE and NUCARS and a list of the proposed features to improve the wedge model [7].

Modeling Friction Wedges, Part II: The State-of-the-Art by Peter E. Klauser

Klauser's second document on the subject of modeling the friction wedge is an extension of the first which introduces "An Improved Model." This paper discusses the implementation of a new, more complex wedge model in the VAMPIRE vehicle dynamics package as both a multibody dynamics model and incorporated into the entire car model. The improved model included a mass for the wedge, rows and columns of elements across both faces to represent pressure distribution along the length and width of the faces, and a wedge width which affects the warp resistance of the bogie. The paper documented the benefits of using an improved friction wedge model as well as listed ways of further improving the model [8].

Dynamic Models of Friction Wedge Dampers by J.P. Cusumano and J.F. Gardner

In this paper, John Gardner and Joseph Cusumano rederived the equations of motion of the friction wedges. The main difference in this paper from others is that it included the mass of the friction wedge; whereas the other papers and the model used in the train modeling software programs ignore the mass of the wedge [9].

Dynamic Modeling and Simulation of Three-Piece North American Freight Vehicle Suspensions with Non-linear Frictional Behavior Using ADAMS/Rail by Robert F. Harder

In this paper, Harder discussed the method in which they derived equations for and modeled a constant damping friction wedge. The author introduced a "toggle" to the program to combat the problem of loading and unloading on the wedge. Loading is when the bolster is moving downward, while unloading is when the bolster is moving upward. The author found that the complex non-linear modeling of the friction wedge lead to modeling challenges due to its complexity [10].

Track Settlement Prediction using Computer Simulation Tools by S.D. Iwnicki, S. Grassie and W. Kik

In this paper, Iwnicki, Grassie and Kik discussed the effects of different vehicles on track deterioration. The authors used MEDYNA, a simulation software, to determine the equations of motion of the track settlement, and then used ADAMS/Rail for the visualizations. The authors explained the three most common types of suspensions used on the tracks and determined how each affect the settlement. The track models they defined using the software represented the ballast in the vertical and lateral directions as parallel spring-damper systems which connect the track to the ground. The results of the paper were the higher the speed of the vehicles on the track, the greater the deterioration of the track [11].

Multibody Simulation of a Freight Bogie with Friction Dampers by N. Bosso, A. Gugliotta and A. Somà

In this paper, Bosso, Gugliotta and Somà discussed the method in which they modeled the friction elements in a Y25 freight bogie, most commonly used in Europe. The main difference in this type of bogie with those commonly used in the United States is that the friction wedge is replaced by friction surfaces between the bogie and axle-box directly. The weight of the car is transferred to the friction surfaces by a mechanical link called a "Lenoir Link". This link transfers the vertical load of the car to a normal force acting on the friction surfaces. The authors model this link and the subsequent friction damping forces in MATLAB as a transfer function between the axle-box and the bogie.

The authors replaced the discontinuities associated with friction force behavior with a non-linear vector of equations. The friction force vector is dependent on the vertical, lateral and longitudinal displacements and the spring forces. The model was compared to the ADAMS/Rail model through simple tests with vertical and lateral inputs. The numerical stability of the model was then tested using various tests commonly used in the rail industry. These tests included slant tests, ride stability tests and curving [12].

Consequences of Nonlinear Characteristics of a Secondary Suspension in a Three-Piece Freight Car Bogie by A. Berghuvud and A. Stensson

In this paper, Berghuvud and Stensson discuss the effects of weather and wear and different conditions on the behavior of the secondary suspension in three-piece bogies. The authors developed two different constantly-damped suspension models, which only included one half of the bogie. The first model was a single degree of freedom model which included the mass of the car and bolster, coil springs with some damping, friction damping and a massless sideframe which actuated the system. This model ignored the pitch moment of the wedge and the friction between the bolster and the wedge. Also the characteristics of each wedge were assumed identical. The second model was a planar model which allowed all bodies except the bolster to have vertical, longitudinal and pitch motion. The bolster could only move vertically and about the lateral axis. This model was also actuated through the sideframe at the axle locations to represent the wheel/rail interaction.

Two different simulations were studied in this paper. The first was excitation in response to a sinusoidal track input which represents track irregularities. The second was different friction configurations. Four configurations were studied. These configurations were two with friction between the bolster and wedge, once with the same friction coefficients between the sideframe and wedge and one with different coefficients, and two without friction between the bolster and wedge with the same friction coefficients as mentioned above. The results of the excitation were that the suspension locked up at frequencies below the natural frequencies for each model. Also for the different friction configurations, the resultant forces on each wedge were different which means that the wedges and sideframes would wear at different rates [13].

Possibility of Jamming and Wedging in the Three-Piece Trucks of a Moving Freight Car by A.D McKisic, V. Ushkalov and M. Zhechev

In this paper, McKisic, Ushkalov and Zhechev present a mathematical and numerical analysis of the secondary suspension in freight train motion. The main goals of this paper were to determine how wedging and jamming occur and how to decrease the frequency at which they occur.

Wedging occurs when there is no relative motion between the wedge and sideframe and wedge and bolster. Jamming depends on the direction of the sliding velocities of each force. The friction between the surfaces is assumed to be Coulomb Friction. The authors used multi-body dynamics to define the bolster-wedge-sideframe system and to obtain the equations of motion. The equations of motion were then used to determine equations for when wedging and jamming would occur. The authors discovered that wedging is not possible for a friction damping system. They were also able to determine equations for when jamming would occur. Using numerical analysis of the equations of motion, jamming was found to occur due to track irregularities. The equations also allow for the determination of how and why jamming occurs and which suspension parameters effect it and how they effect it [14].

Modeling and Dynamics of Friction Wedge Dampers in Railroad Freight Trucks by A.B. Kaiser, J.P. Cusumano, and J.F. Gardner

In this paper Kaiser, Cusumano, and Gardner discuss a friction wedge modeling approach with a highly sophisticated dry friction formulation. Their single degree of freedom model assumes a kinematic constraint based on wedge geometry with both wedge surfaces continuously in frictional contact. The frictional oscillations exhibited by the model were analyzed for Coulomb, static-dynamic, and slip rate dependant friction relations.

Also presented in this paper is an efficient method for analyzing the dynamic behaviors of the simple one degree of freedom system. A piece-wise smooth system equation is implemented to analyze bifurcations in the spatial locations of sticking events occurring for parameterizations in both the amplitude and frequency of the excitation. The numerical analyses reveal that for small track amplitudes and track speeds the response is dominated by the sticking events. Conversely, at high track amplitudes and track speeds the sticking events are absent [15].

Experimental study of stick-slip dynamics in a friction wedge damper by N.K. Chandiramani, K. Srinivasan, J. Nagendra.

In this paper Chandiramani, Srinivasan, and Nagendra perform investigations on a friction wedge damper model with the objective of studying stick-slip phenomena at various excitation frequencies. After performing tests on their friction wedge and bolster model they concluded that most stick-slip phenomena occur at 30hz excitation frequency. At lower frequencies inertial forces are insufficient to overcome static friction resulting in the stick-slip response. The stick-slip events decrease as frequency is increased.

The authors then characterized two types of stick-slip phenomena, one occurring at low frequencies and the other occurring at high frequencies. During low frequency stick-slip the relative velocities retain their direction after sticking where this is not see in the high frequency stick-slip phenomena [16].

1.6. Summary of Thesis

Chapter 2 describes the modeling approach taken for the construction of the multibody dynamics models. The bodies that comprise each model and their respective degrees of freedom are described along with the friction and contact algorithms that govern their motion. A kinematic and dynamic analysis is presented for each model created within the multibody dynamics model framework. Chapter 3 discusses a brief synopsis of the NUCARS software and presents the models created with NUCARS to be comparable with the multibody dynamics models. Results of the simulations are shown in Chapter 4. Each multibody dynamics model compares toe-in, toe-out, and no-toe configurations. In addition to toe case studies, the half-truck variably and constantly damped models were compared with the equivalent NUCARS models. Finally, in Chapter 5 conclusions for the project and future work are discussed.

2. MULTIBODY DYNAMICS MODEL

This chapter discusses the creation of the multibody dynamics models. Discussed in the beginning of the chapter are modeling assumptions and contact condition formulation. Later in the chapter each multibody dynamics model is discussed in general then explained with a kinematic and dynamic analysis.

The multibody dynamics model of the friction wedge is based on analytical and computational methods and implemented using MATLAB. Visualization of the simulations was attained using MAMBO [4], a freeware educational dynamics program used at Virginia Tech. The multibody dynamics model was designed to be comparable to a modified single truck of a 100 ton hopper car modeled in NUCARS [3]. Each model was assembled so that the external forces acting on the wedge due to other factors (e.g., wheel – rail interaction) are assumed to exist only as simplistic inputs [2].

Within the multibody dynamics model the friction wedges and bolster are modeled as three dimensional rigid bodies with mass and inertial properties. The friction wedges are allowed four degrees of freedom: vertical translation, longitudinal translation, yaw, and pitch. These degrees of freedom are detailed in Figure 2-1. The bolster is allowed two degrees of freedom: vertical translation and yaw. The degrees of freedom given to the bolster are detailed in Figure 2-2.

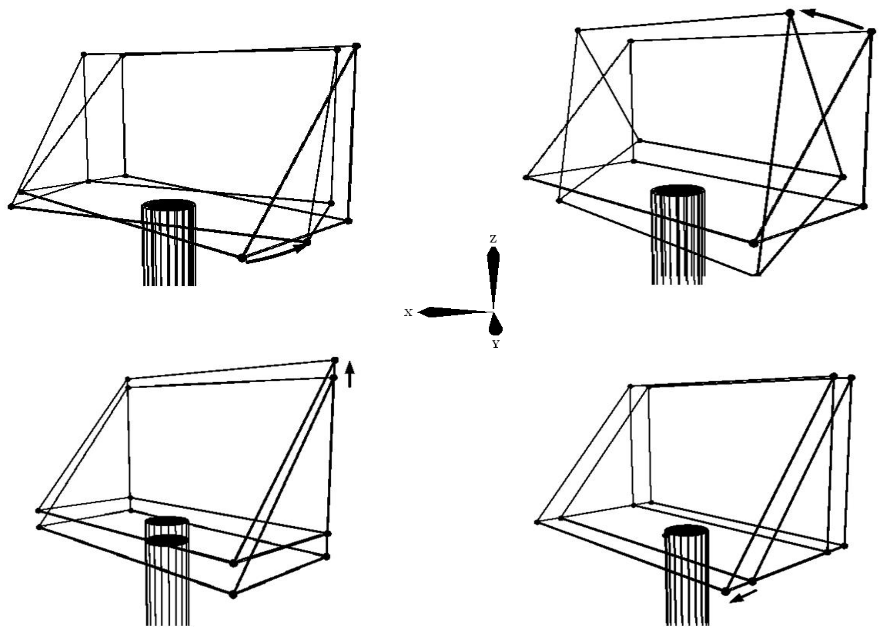
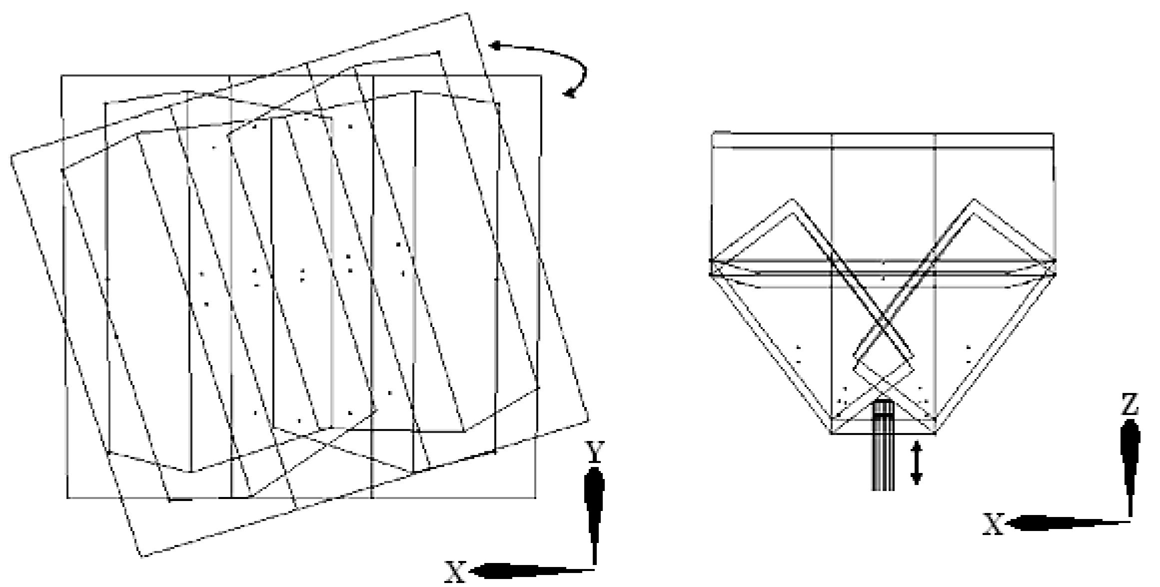


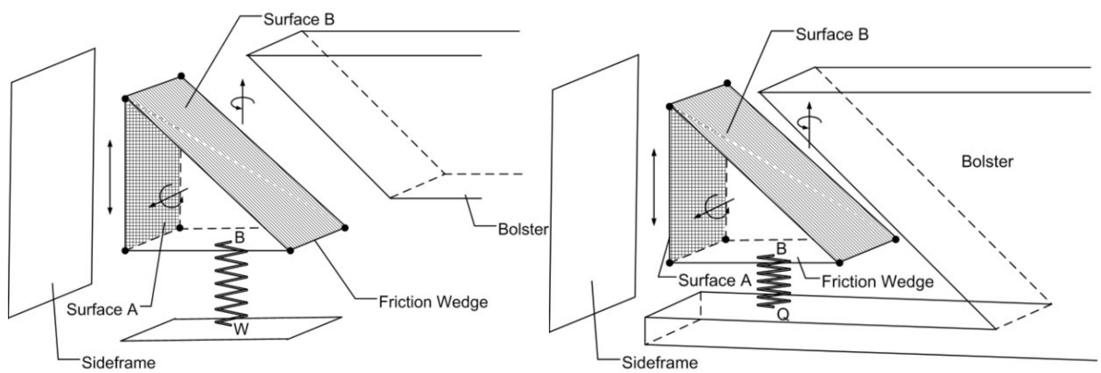
Figure 2-1. Four degrees of freedom given to the friction wedge for the multibody dynamics model simulations.



(a) (b)

Figure 2-2. The two degrees of freedom given to the bolster for the multibody dynamics model simulations. The yaw dof (a) is shown as a top view of the bolster rotating around the z-axis. The vertical translation dof (b) is shown as a side view of the bolster.

The wedges have been simplified as two 3-D connected surfaces, A and B in Figure 2-3, which contact the sideframe and bolster respectively. Instead of assuming one generalized contact force acting on each surface of the wedge, forces were assumed to appear at contact points mapped on surfaces A and B. The multibody dynamics models include four points on the edges of each surface to determine contact between surface A and the sideframe and surface B and the bolster.



(a) (b)

Figure 2-3. Exploded view of the sideframe-friction wedge-bolster system for the variably damped (a) and constantly damped (b) multibody dynamics models.

The friction damping behavior of a three piece bogie is modeled as tangential forces that depend explicitly on the coefficient of friction μ of the surface and the normal contact force. The frictional damping forces arise as a function of the tangential velocity and contact force acting at the contact points on the surfaces as seen in Figure 2-3. Friction forces at the surface interactions are modeled using a sign function to approximate the stick-slip friction seen in actual friction wedge behavior. The stick-slip friction is represented graphically in Figure 2-4.

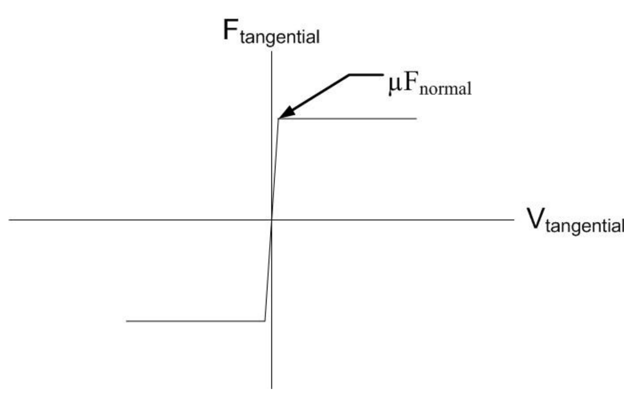


Figure 2-4. Tangential force vs the tangential velocity used to approximate stick-slip friction.

From the contact points, reaction normal and friction forces and their associated moments are determined by the kinematics, dynamics and stick-slip friction algorithm produced by the simulations. The forces and moments generated as a result of the friction couple will, in turn, excite the translational and rotational degrees of freedom of the wedges.

Many assumptions were made in the design of the multibody dynamics friction wedge models. The first is that the geometry of the wedges is defined by contact points mapped along the edges of the wedge surfaces. The geometry modeled assumes that the surfaces are flat whereas in reality they are slightly curved. The next assumption is that the wedge, bolster and sideframe faces interacting are not rigid. This assumption allows for stick-slip motion, this means there is a small shear displacement occurring at the interfaces between these surfaces before slipping. It is also assumed the wedge may not be in contact with the bolster or with the sideframe at all times. Contact between the surfaces will be governed by their geometry and the motions of the bodies during the simulations. The sideframe and the bolster forces resulting from contact are assumed as compliant reaction forces resulting from a unilateral spring, which only appear when the wedge comes into contact with either the sideframe, bolster or with both simultaneously. Lastly, the control coils and load springs are assumed to be linear springs. This is a reasonable hypothesis for within the useful range of these springs, which is seen in the simulations.

The parameters used for the creation of the multibody dynamics models were taken from commonly accepted sizes and weights of a 100 ton freight train hopper. These values were used in calculating the kinematic and dynamic effects experienced by the friction wedges and bolster. The values for the parameters can be seen in Table 1 and 2. A figure showing the geometric parameters can be seen in Figure 2-5.

Table 1. Geometric parameters used in the half-truck variably damped simulations

Wedge angle, θ°		37.500
Wedge dimensions, (in)	h_x	3.827
	h_y	10.250
	h_z	6.125
CG location relative to point B, (in)	p_1	0.783
	p_2	0.000
	p_3	2.0417

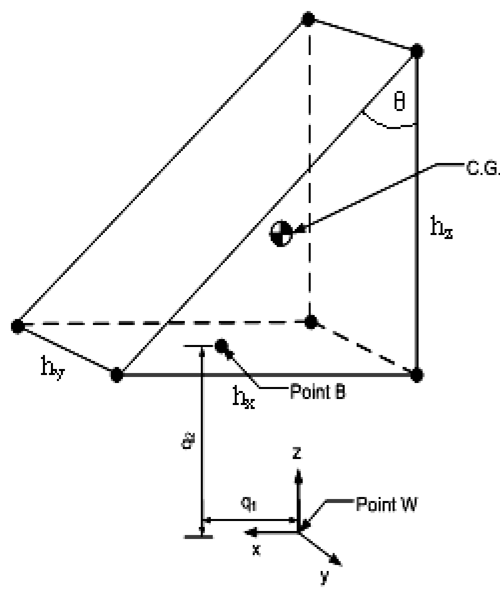


Figure 2-5. Diagram showing the geometric parameters with respect to the half-truck wedge

Table 2. Dynamic modeling parameters for the half-truck variably damped model simulations

Sideframe coefficient of friction, μ_s		0.4
Bolster coefficient of friction, μ_b		0.25
Mass of the wedge, $m_{w,}$ (lb-m)		0.875
Mass of the bolster, $m_{b,}$ (lb-m)		468
Moment of inertia - wedges (lb-in^2)	I_{11}	10.7746
	I_{22}	3.8034
	I_{33}	9.1070
Moment of inertia - bolster (lb-in^2)	I_{11}	10443
	I_{22}	6705.8
	I_{33}	7494.9
l_0 (in)		10.25
$C_{damp,}$ (lbf-s/in)		1
Control Coil Spring Stiffness, (lbs/in)		1491
Load Coil Spring Stiffness, (lbs/in)		22417.8
Sideframe-wedge stiffness, (lbs/in)		1000000
Bolster-wedge stiffness, (lbs/in)		1000000

2.1.1. Kinematics Analysis

The wedge in Figure 2-6 shows how the state variables q_1 and q_2 are used to describe the translational motion of the left friction wedge (defined by Figure 2-8) in the longitudinal and the vertical direction, and how the state variables q_3 and q_4 are used to describe the two rotations about the pitch and the yaw axes, respectively.

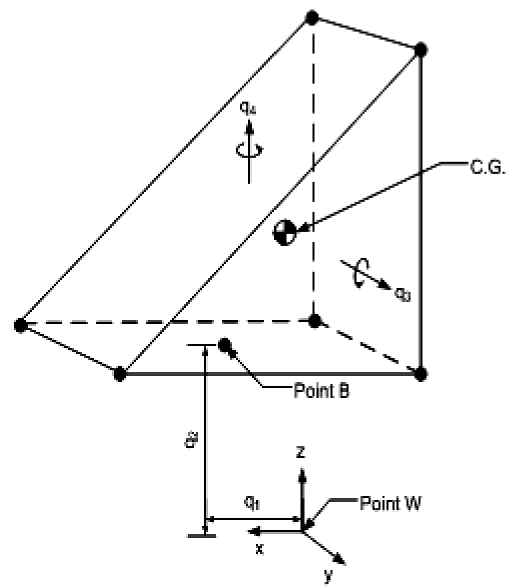


Figure 2-6. Translational and rotational degrees of freedom of the half-truck variably damped model wedge.

There is no translation in the y-direction or rotation about the x-axis. Hence, in 3-D vector form, we can say that the position vector from point W to point B to be:

$$\vec{r}^{WB} = \{q_1, 0, q_2\}^T \tag{1}$$

The position vector from point B to the C.G. is then:

$$\vec{r}^{BCG} = \{p_1, p_2, p_3\}^T \tag{2}$$

Where p_1, p_2, p_3 define the location of the C.G. of the wedge with respect to point B. The rotation of the wedge is represented by a composite rotation matrix;

$$R^{wb} = \begin{pmatrix} \cos(q_4) & \sin(q_4) & 0 \\ -\sin(q_4) & \cos(q_4) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(q_3) & 0 & -\sin(q_3) \\ 0 & 1 & 0 \\ \sin(q_3) & 0 & -\cos(q_3) \end{pmatrix} \tag{3}$$

The wedge positioned to the right has the same translational and rotational degrees of freedom as mentioned for the left wedge. In 3-D vector form the right wedge is defined in the same way as the left wedge, but with differently numbered variables assigned to define the translations and rotations.

2.1.2. Dynamic Analysis

In the multibody dynamics model, the mass and the rotational inertia of the wedges are included to account for inertial effects. The contact forces act on all six points on the edges of the wedge. The six points are in place to detect the "penetration" of the wedge into the contacting surfaces. The mapping of the points relative to point B is shown in Figure 2-7.

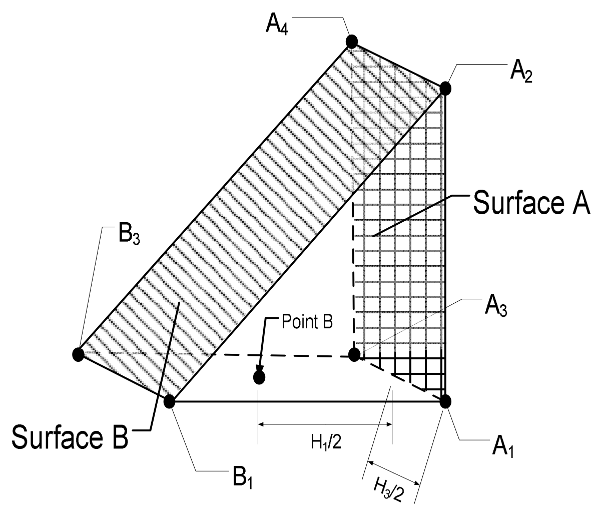


Figure 2-7. Diagram detailing the surfaces used to define the wedges in the half-truck variably damped model.

The unilateral contact forces work using the surfaces A and B on the wedges, the bolster surface and sideframe surface contacting the wedges. The contact conditions allow for the "penetration" of a wedge point into the opposing surface. When the contact is detected spring forces corresponding to material properties of the surfaces appear and resist the movement of the point into the surface.

Using Kane's Method and D'Alembert's Principle [4,15], the force and moment equations can be derived for the wedges. The resulting force and moment equations are derived symbolically as:

$$
\begin{aligned}
\vec{F}_{w1} = {} & m_{w1} g \cdot \vec{z} \\
& - \left(F^{w1B1}_{normalB} + F^{w1A2}_{normalB} + F^{w1B3}_{normalB} + F^{w1A4}_{normalB} \right) \cdot \vec{n}_B \\
& + \left(F^{w1B1}_{\tan gentB} + F^{w1A2}_{\tan gentB} + F^{w1B3}_{\tan gentB} + F^{w1A4}_{\tan gentB} \right) \cdot \vec{t}_B \\
& - \left(F^{w1A1}_{normalSF} + F^{w1A2}_{normalSF} + F^{w1A3}_{normalSF} + F^{w1A4}_{normalSF} \right) \cdot \vec{n}_{SF} \\
& - \left(F^{w1A1}_{\tan gentSF} + F^{w1A2}_{\tan gentSF} + F^{w1A3}_{\tan gentSF} + F^{w1A4}_{\tan gentSF} \right) \cdot \vec{t}_{SF} \\
& + K_s \left(l_o - \left\| \vec{r}^{\,w1B} \right\| \right) \frac{\vec{r}^{\,w1B}}{\left\| \vec{r}^{\,w1B} \right\|} \\
& - c_{damp} u_1 \cdot \vec{x} - c_{damp} u_2 \cdot \vec{z}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
\vec{T}_{w1} = {} & - \left(\vec{r}^{\,w1cgA1} \times \left(\vec{F}^{w1A1}_{normalSF} \cdot \vec{n}_{SF} \right) + \vec{r}^{\,w1cgA2} \times \left(\vec{F}^{w1A2}_{normalSF} \cdot \vec{n}_{SF} \right) \right) \\
& - \left(\vec{r}^{\,w1cgA3} \times \left(\vec{F}^{w1A3}_{normalSF} \cdot \vec{n}_{SF} \right) + \vec{r}^{\,w1cgA4} \times \left(\vec{F}^{w1A4}_{normalSF} \cdot \vec{n}_{SF} \right) \right) \\
& - \left(\vec{r}^{\,w1cgA1} \times \left(\vec{F}^{w1A1}_{\tan gentSF} \cdot \vec{t}_{SF} \right) + \vec{r}^{\,w1cgA2} \times \left(\vec{F}^{w1A2}_{\tan gentSF} \cdot \vec{t}_{SF} \right) \right) \\
& - \left(\vec{r}^{\,w1cgA3} \times \left(\vec{F}^{w1A3}_{\tan gentSF} \cdot \vec{t}_{SF} \right) + \vec{r}^{\,w1cgA4} \times \left(\vec{F}^{w1A4}_{\tan gentSF} \cdot \vec{t}_{SF} \right) \right) \\
& - \left(\vec{r}^{\,w1cgB1} \times \left(\vec{F}^{w1B1}_{normalB} \cdot \vec{n}_{SF} \right) + \vec{r}^{\,w1cgB2} \times \left(\vec{F}^{w1B2}_{normalB} \cdot \vec{n}_{SF} \right) \right) \\
& - \left(\vec{r}^{\,w1cgB3} \times \left(\vec{F}^{w1B3}_{normalB} \cdot \vec{n}_{B} \right) + \vec{r}^{\,w1cgB4} \times \left(\vec{F}^{w1B4}_{normalB} \cdot \vec{n}_{B} \right) \right) \\
& + \left(\vec{r}^{\,w1cgB1} \times \left(\vec{F}^{w1B1}_{\tan gentB} \cdot \vec{t}_{B} \right) + \vec{r}^{\,w1cgB2} \times \left(\vec{F}^{w1B2}_{\tan gentB} \cdot \vec{t}_{B} \right) \right) \\
& + \left(\vec{r}^{\,w1cgB3} \times \left(\vec{F}^{w1B3}_{\tan gentB} \cdot \vec{t}_{B} \right) + \vec{r}^{\,w1cgB4} \times \left(\vec{F}^{w1B4}_{\tan gentB} \cdot \vec{t}_{B} \right) \right) \\
& - \left(c_{damp} u_6 \cdot \vec{y} - c_{damp} u_7 \cdot \vec{z} \right)
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
\vec{F}_{w2} = {} & m_{w2} g \cdot \vec{z} \\
& -\left(F_{normalB}^{w2B1} + F_{normalB}^{w2A2} + F_{normalB}^{w2B3} + F_{normalB}^{w2A4}\right)\cdot \vec{n}_B \\
& +\left(F_{\tan gentB}^{w2B1} + F_{\tan gentB}^{w2A2} + F_{\tan gentB}^{w2B3} + F_{\tan gentB}^{w2A4}\right)\cdot \vec{t}_B \\
& -\left(F_{normalSF}^{w2A1} + F_{normalSF}^{w2A2} + F_{normalSF}^{w2A3} + F_{normalSF}^{w2A4}\right)\cdot \vec{n}_{SF} \\
& +\left(F_{\tan gentSF}^{w2A1} + F_{\tan gentSF}^{w2A2} + F_{\tan gentSF}^{w2A3} + F_{\tan gentSF}^{w2A4}\right)\cdot \vec{t}_{SF} \\
& + K_s\left(l_o - \left\|\vec{r}^{\,w2B}\right\|\right)\frac{\vec{r}^{\,w2B}}{\left\|\vec{r}^{\,w2B}\right\|} \\
& - c_{damp} u_4 \cdot \vec{x} - c_{damp} u_5 \cdot \vec{z}
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
\vec{T}_{w2} = {} & -\left(\vec{r}^{\,w2cgA1} \times \left(\vec{F}_{normalSF}^{w2A1} \cdot \vec{n}_{SF}\right) + \vec{r}^{\,w2cgA2} \times \left(\vec{F}_{normalSF}^{w2A2} \cdot \vec{n}_{SF}\right)\right) \\
& -\left(\vec{r}^{\,w2cgA3} \times \left(\vec{F}_{normalSF}^{w2A3} \cdot \vec{n}_{SF}\right) + \vec{r}^{\,w1cgA4} \times \left(\vec{F}_{normalSF}^{w2A4} \cdot \vec{n}_{SF}\right)\right) \\
& -\left(\vec{r}^{\,w2cgA1} \times \left(\vec{F}_{\tan gentSF}^{w2A1} \cdot \vec{t}_{SF}\right) + \vec{r}^{\,w2cgA2} \times \left(\vec{F}_{\tan gentSF}^{w2A2} \cdot \vec{t}_{SF}\right)\right) \\
& -\left(\vec{r}^{\,w2cgA3} \times \left(\vec{F}_{\tan gentSF}^{w2A3} \cdot \vec{t}_{SF}\right) + \vec{r}^{\,w2cgA4} \times \left(\vec{F}_{\tan gentSF}^{w2A4} \cdot \vec{t}_{SF}\right)\right) \\
& -\left(\vec{r}^{\,w2cgB1} \times \left(\vec{F}_{normalB}^{w2B1} \cdot \vec{n}_{SF}\right) + \vec{r}^{\,w2cgB2} \times \left(\vec{F}_{normalB}^{w2B2} \cdot \vec{n}_{SF}\right)\right) \\
& -\left(\vec{r}^{\,w2cgB3} \times \left(\vec{F}_{normalB}^{w2B3} \cdot \vec{n}_{B}\right) + \vec{r}^{\,w2cgB4} \times \left(\vec{F}_{normalB}^{w2B4} \cdot \vec{n}_{B}\right)\right) \\
& +\left(\vec{r}^{\,w2cgB1} \times \left(\vec{F}_{\tan gentB}^{w2B1} \cdot \vec{t}_{B}\right) + \vec{r}^{\,w2cgB2} \times \left(\vec{F}_{\tan gentB}^{w2B2} \cdot \vec{t}_{B}\right)\right) \\
& +\left(\vec{r}^{\,w2cgB3} \times \left(\vec{F}_{\tan gentB}^{w2B3} \cdot \vec{t}_{B}\right) + \vec{r}^{\,w2cgB4} \times \left(\vec{F}_{\tan gentB}^{w2B4} \cdot \vec{t}_{B}\right)\right) \\
& -\left(c_{damp} u_9 \cdot \vec{y} - c_{damp} u_{10} \cdot \vec{z}\right)
\end{aligned}
\tag{7}
$$

Where

$\frac{\vec{r}^{\,wiB}}{\left\|\vec{r}^{\,wiB}\right\|}$ = unit vector along length of ith spring

$(\cdot)$ = dot product operator

$(\times)$ = cross product operator

$F_{normalB}^{wiBi}$ = force acting normal to surface B on ith wedge on ith point B

$F_{\tan gentB}^{wiBi}$ = force acting tangent to surface B on ith wedge on ith point B

$F_{normalSF}^{wiAi}$ = force acting normal to surface A on ith wedge on ith point A

$F_{\tan gentSF}^{wiAi}$ = force acting tangentto surface A on ith wedge on ith point A

$F_{normalB}^{wiAi}$ = force acting normal to surface B on ith wedge on ith point A

$F_{\tan gentB}^{wiAi}$ = force acting tangentto surface B on ith wedge on ith point A

$\vec{n}_{SF}$ = unit vector normal to surface A

$\vec{n}_{B}$ = unit vector normal to surface B

$\vec{t}_{SF}$ = unit vector tangent to surface A

$\vec{t}_{B}$ = unit vector tangent to surface B

2.2. Half-truck Variably Damped Model

A variably damped truck has a friction damping system where the force of the springs biasing the friction wedges against the sideframe increases as the bolster is depressed vertically. The control coils supporting the friction wedges are connected to the sideframe with no other supports. This configuration allows the stroke of the bolster to compress the control coils as needed to aid in damping inputs. The variably damped half-truck model was created to mimic one half of a simplistic variably damped truck. The bodies modeled include a sideframe, a half-bolster, and two friction wedges, along with these the springs supporting the bolster (the load coil) and the friction wedges (control coils) were also included. A visualization of the multibody dynamics model created in Mambo can be seen in Figure 2-8. The Matlab code for this model can be seen in Appendix 7.1.1.

The variably damped half-truck model is based on the previous modeling assumptions and will use the sideframe as the input body, as opposed to the bolster as in the quarter-truck models. Using the sideframe as the input body to the system will create more naturalistic inputs and more closely represent a three-piece bogie.

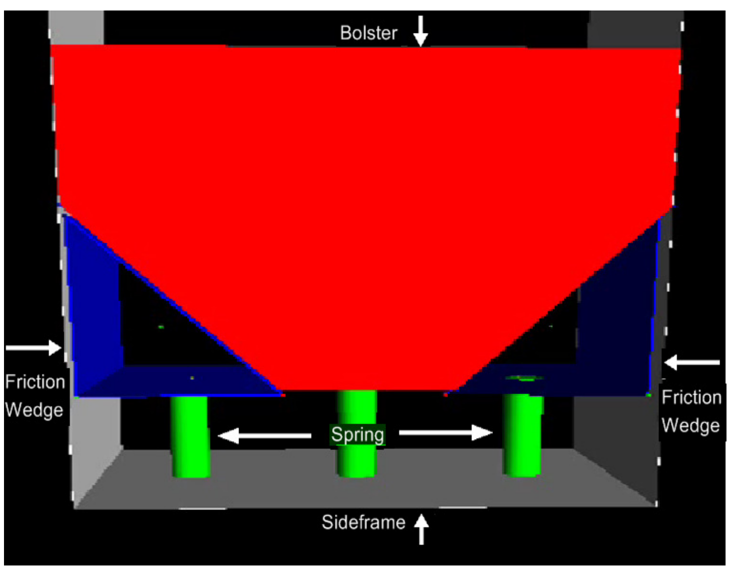


Figure 2-8. Variably damped half-truck model visualization taken from Mambo.

As a starting point for a wearing wedge model the half-truck variably damped model was used as a test bed for experimental simulations. A wedge wear model is of particular interest because of the safety implications offered by a model that can predict safe service life . By placing 29 contact points, arranged symmetrically, over the two wedge faces a crude estimation of the stresses and contact forces across the wedge surfaces can be calculated. With the stresses across the wedge surfaces and material properties, surface wear over repeated contact can be estimated for the wedge surfaces. A visual comparison of the two wedge elements can be seen in Figure 2-9.

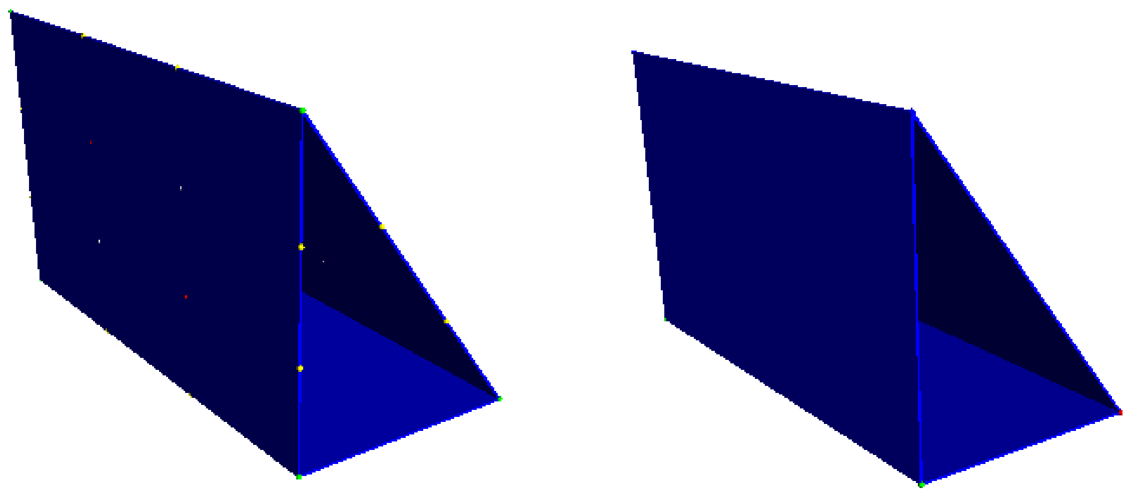


Figure 2-9. Wedge curvature friction wedge(left) and the standard friction wedge(right) used in the variably damped simulations.

2.3. Half-truck Constantly Damped Model

A constantly damped truck has a friction damping system where the force of the springs biasing the friction wedges against the sideframe remains relatively constant during the stroke of the bolster. This constant damping is due to the configuration of the geometry of a constantly damped truck. Instead of the control coils being connected to the sideframe, as in a variably damped truck, they are connected directly to a pocket within the bolster. As the bolster strokes vertically the length of the control coils remain relatively constant. The constantly damped half-truck model was created to mimic one half of a simplistic constantly damped truck, as shown in Figure 2-10. The bodies modeled include a sideframe, a half-bolster, and two friction wedges, along with these the springs supporting the bolster (the load coil) and the friction wedges (control coils) were also included. The Matlab code for this model can be seen in Appendix 7.1.2.

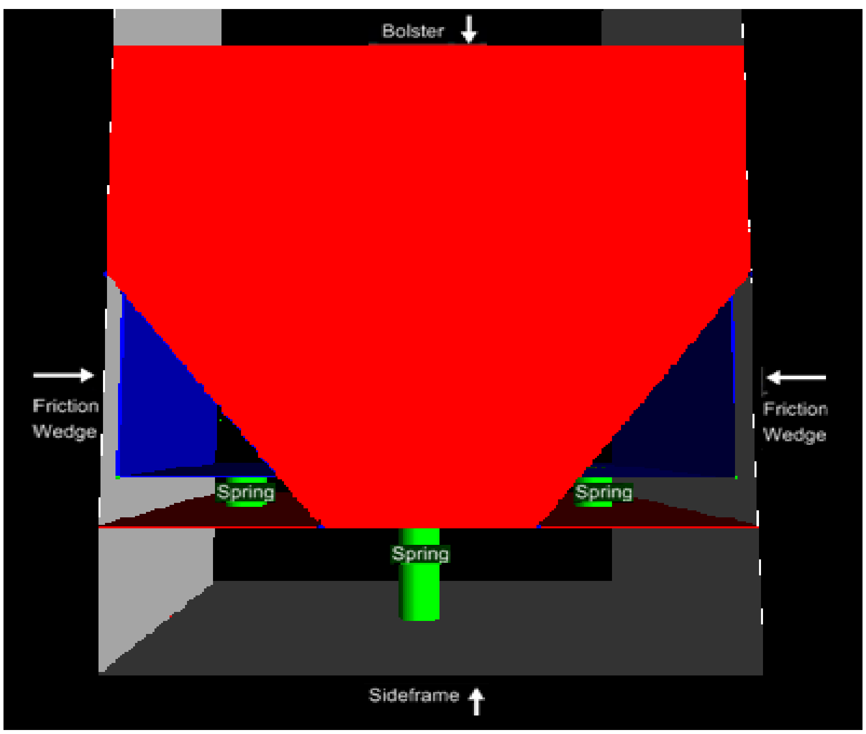


Figure 2-10. Constantly damped half-truck model visualization taken from Mambo.

The constantly damped half-truck model is based on the same modeling assumptions as the variably damped half-truck model. The constantly damped model was created with the same contact point geometry mapped on the wedges and bolster. Using the same contact point geometry will result in identical force and moment formulations for the constantly damped and variably damped models.

Although the force and moment formulations are identical, the values for the equations of motion differ greatly because of the altered reference frame configurations. For the variably damped model the reference frames are defined to neighbor each other according to Figure 2-11.

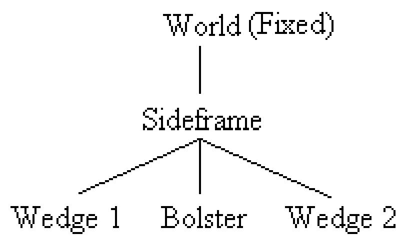


Figure 2-11. Tree diagram showing neighbors for var damp

The wedge reference frames are defined to neighbor the sideframe reference frame. Accordingly the state variables for the wedges are defined with respect to the sideframe reference frame. In the constantly damped model the reference frames are re-configured to account for geometry differences between a variably damped truck and a constantly damped truck. The constantly damped model neighboring reference frames are defined according to Figure 2-12. The wedge reference frames are defined to neighbor the bolster reference frame and the state variables are taken with respect to the bolster reference frame.

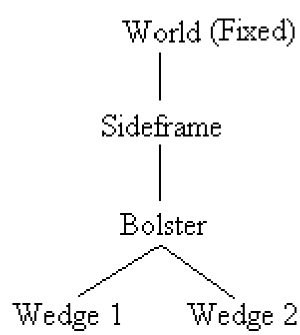


Figure 2-12. Tree diagram showing neighbors for const damp

2.4. Variably Damped Warp Model

Freight train suspensions provide the steering for a freight train moving on track. It is desirable to have a truck that steers well and also provides an adequate amount of warp stiffness. Warp stiffness is defined as the rotational stiffness that resists the moments generated by the suspension moving on track.

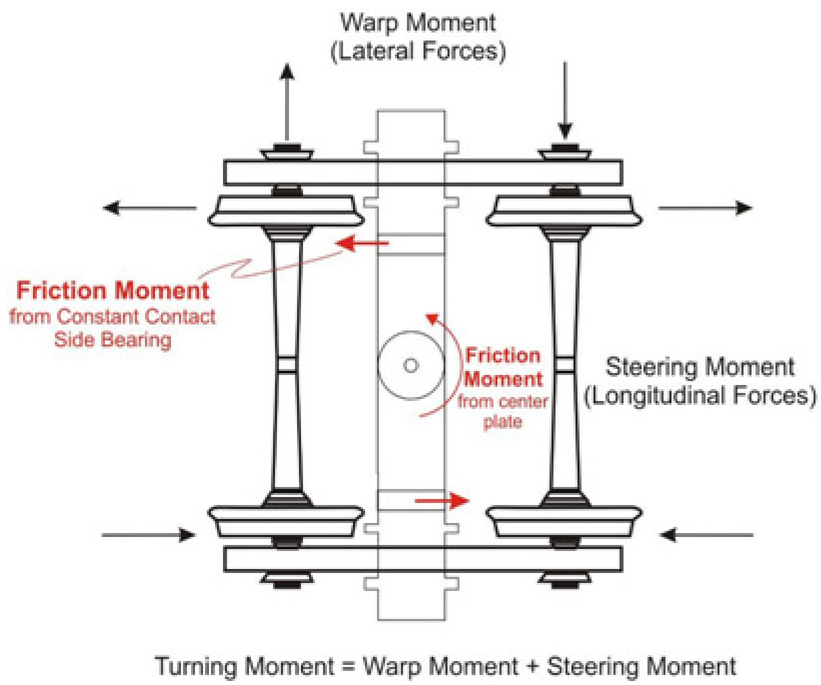


Figure 2-13. Moments generated by the suspension traveling on track.

A truck with low warp stiffness will steer through turns very well, but be prone to an unstable oscillation while moving down track, which is known as hunting. Conversely, a truck with high warp stiffness will understeer through turns, but resist hunting. The resolution of this dilemma becomes much like the tradeoff between ride quality and performance for an automobile suspension. Manufacturers sacrifice one favorable suspension quality for another to improve overall ride performance.

Hunting can lead to rapid truck component wear, car body damage, and high speed derailments. These issues make maintaining adequate warp stiffness to prevent hunting a substantial safety and monetary concern for truck manufacturers and railroad companies. Many new truck designs have been created to improve truck warp stiffness. Some designs have seen success while others have not been so fortunate. Components ranging from additional truck frame supports to lateral hydraulic dampers have been added to mitigate hunting conditions.

One method aimed at improving warp stiffness involves improvement of friction wedges. Friction wedges that provide more damping will effectively increase the warp stiffness of a truck by simply replacing the wedge components within a truck. Using the previously created half-truck variably damped model a simplistic analysis of truck warp stiffness can be performed. The half-truck variably damped model was modified to include lateral translational degrees of freedom for the friction wedges as well as a new friction coefficient formulation.

In the previous models friction forces are only needed for contact occurring during vertical wedge movement (lateral movement was constrained). Due to the addition of the wedge lateral translational degree of freedom a new approach to modeling the friction generated due to bolster – wedge – sideframe contact is needed.

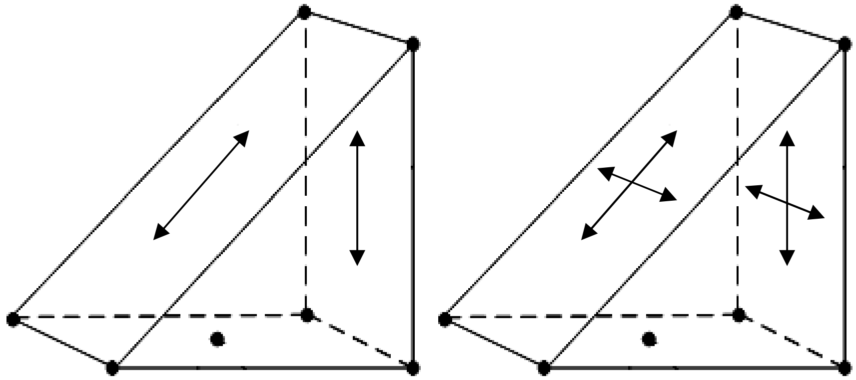
The warp model creates friction forces in the vertical and/or lateral directions depending on wedge movement. By using friction circle theory a more realistic representation of the friction forces generated can be developed. Applying this theory to the warp model means that the frictional load will be limited to a maximum frictional force and vertical and lateral translations will receive proportional friction coefficients based on their respective velocities. The following equations and figure detail the frictional formulation.

$$\mu_z wedge = \mu_{sideframe}\left(\frac{v_z wedge_{cg}}{\sqrt{\left(v_z wedge_{cg}\right)^2 + \left(v_y wedge_{cg}\right)^2}}\right) \tag{8}$$

$$\mu_y wedge = \mu_{sideframe}\left(\frac{v_y wedge_{cg}}{\sqrt{\left(v_z wedge_{cg}\right)^2 + \left(v_y wedge_{cg}\right)^2}}\right) \tag{9}$$

$$\mu_z bolster = \mu_{bolster}\left(\frac{v_z wedge_{cg}}{\sqrt{\left(v_z wedge_{cg}\right)^2 + \left(v_y wedge_{cg}\right)^2}}\right) \tag{10}$$

$$\mu_y bolster = \mu_{bolster}\left(\frac{v_y wedge_{cg}}{\sqrt{\left(v_z wedge_{cg}\right)^2 + \left(v_y wedge_{cg}\right)^2}}\right) \tag{11}$$



(a) (b)

Figure 2-14. Figure showing the previous friction formulation (a) and the warp model friction formulation (b).

The wedge in Figure 2-15 shows a diagram of the wedge used in the warp model. This diagram highlights the additional degrees of freedom used for the warp model. The Matlab code for this model can be seen in Appendix 7.1.3.

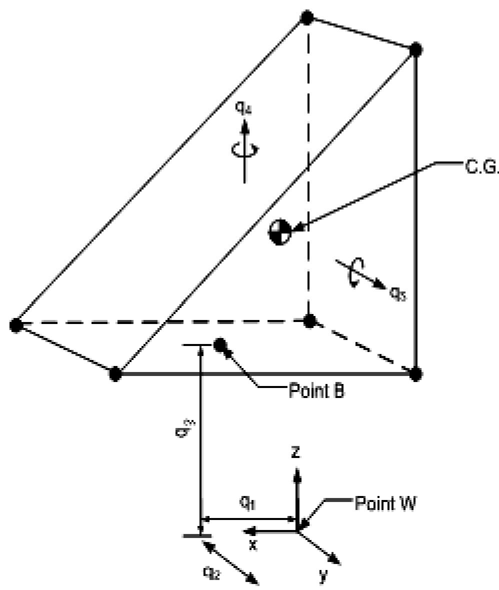


Figure 2-15. Translational and rotational degrees of freedom of the warp model wedge.

2.5. Quarter-truck Variably Damped Model with Wedge Face Curvature

The variably damped quarter-truck model with wedge face curvature was created to mimic one quarter of a simplistic variably damped truck. The bodies modeled include a sideframe, a quarter-bolster, and a friction wedge with a curved face on the bolster side, along with these bodies the springs supporting the bolster (the load coil) and the friction wedges (control coils) were also included.

This model includes the previously mentioned modeling assumptions for the multibody dynamics models with one important exception. This exception is the modeling of the wedge face contacting the bolster. In previous multibody dynamics models this surface was modeled as a flat surface whereas this model attempts to draw closer to actual wedge geometry by implementing a curved wedge surface. The curved surface was implemented in Mambo by using the edge of a cylinder. This modeling approach was used because Mambo only has the ablility to model simple geometric shapes.

With the implementation of the curved wedge face several additional modeling assumptions are needed. The first assumption is that there is unilateral contact between points mapped on the surface of the bolster and the radius of the cylinder representing the curved wedge face. The moment arm calculations, for moments acting on the wedge, are derived from points mapped on the corners of the flat wedge face. The bolster contact points can penetrate the curved wedge face at infinitely many locations, for this reason the moment arm for moments acting on the wedge must be approximated by the point mapped on the flat wedge face closest to the corresponding bolster contact point. Images of the wedge curvature model can be seen in Figure 2-16Figure 2-17. The Matlab code for this model can be seen in Appendix 7.1.4.

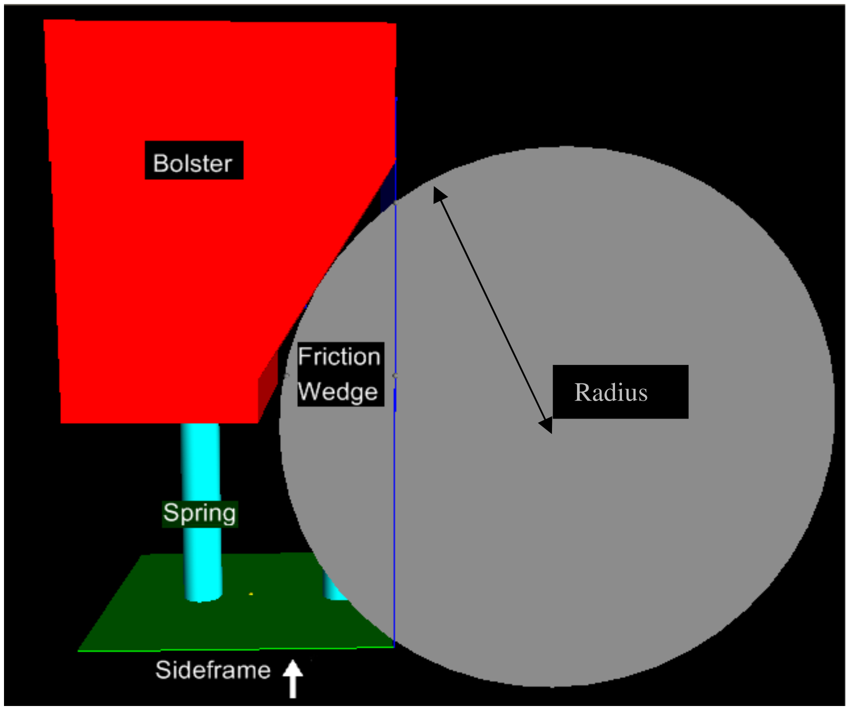


Figure 2-16. Variably damped quarter-truck model with wedge face curvature taken from Mambo.

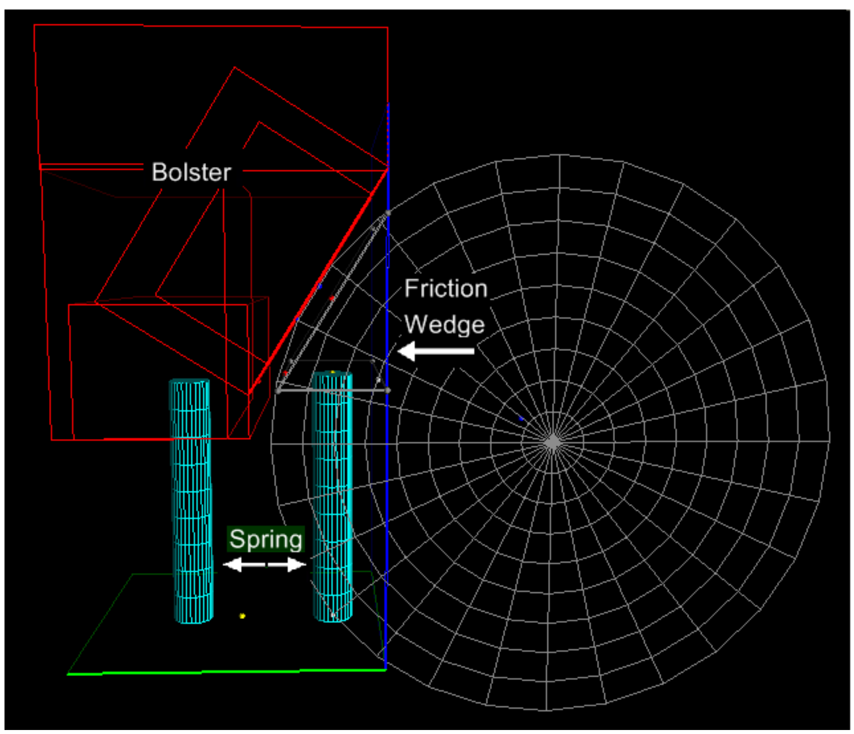


Figure 2-17. Wire image of the variably damped quarter-truck model with wedge face curvature showing the positioning of the cylinder to represent a curved wedge face.

The surfaces defined in the wedge face curvature model are shown in Figure 2-18. Surface A is defined by points A1-A4 and Surface B is now defined by the curvature of the cylinder created in Mambo.

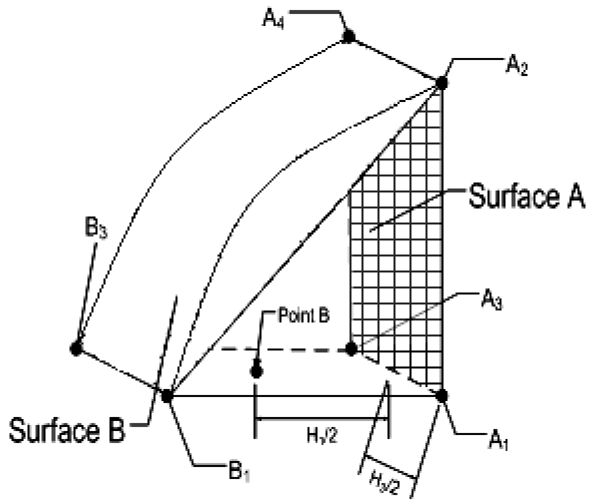


Figure 2-18. Surfaces used to define the wedges in the quarter-truck wedge curvature model..

2.6. Full-truck Variably Damped Model

Creating a full-truck model is the next iteration in the development of the multibody dynamics model of a freight train suspension. The full-truck variably damped model is an attempt at modeling a simplistic full-truck based on the half-truck variably damped modeling assumptions. The full-truck variably damped model was created using the original template of the half-truck design and connecting two sideframes (each containing two wedges) with a single bolster. An image of the model geometry can be seen in Figure 2-19. The Matlab code for this model can be seen in Appendix 7.1.5.

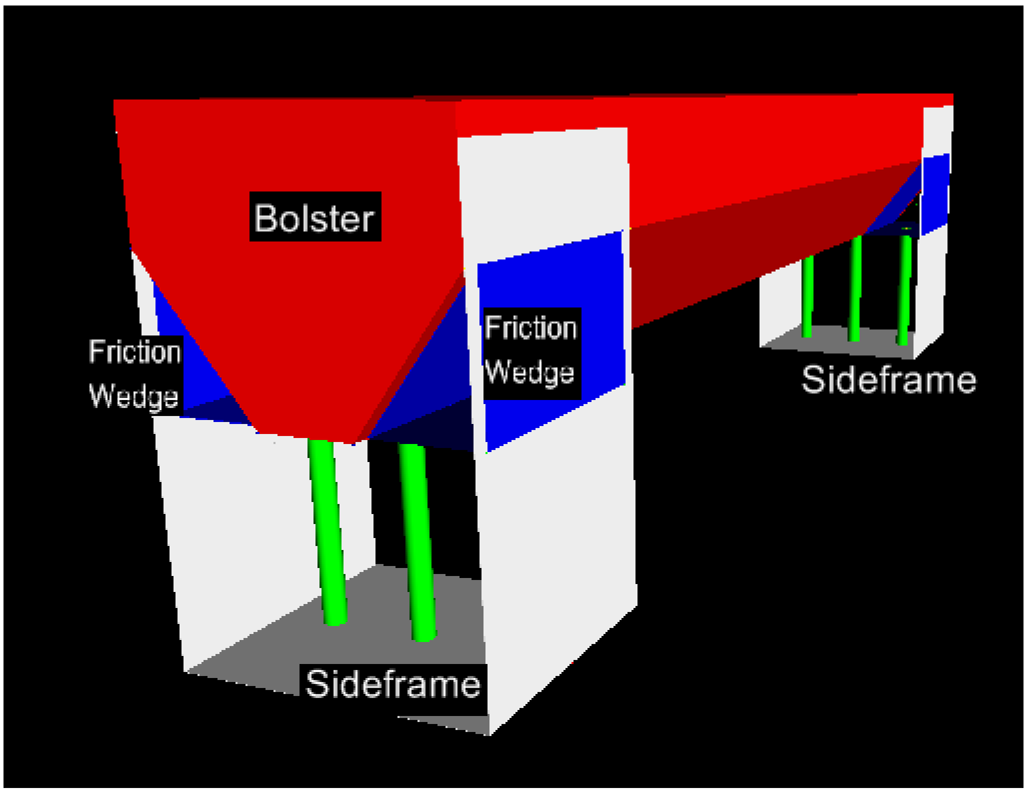


Figure 2-19. Variably damped full-truck model taken from Mambo.

The two sideframes are the input bodies to the model and are allowed vertical and longitudinal degrees of freedom. The inputs created by the sideframes have the robustness to be varied from one another in both amplitude and phase. This robustness allows for a more realistic simulation of truck warping by translating the sideframes out of phase, as in actual warping.

Due to the increased complexity of the inputs the degrees of freedom given to the bolster must be updated to retain proper dynamics for the bolster. In previous half-truck and quarter-truck models the bolster required only vertical and yaw degrees of freedom to respond properly to system inputs. The ability to alter the sideframe inputs in phase and out of phase in the full-truck model requires the bolster to be allowed a roll degree of freedom to respond to out of phase vertical sideframe inputs.

Whereas the sideframes and bolster needed to be restructured for the full-truck variably damped model the friction wedges retain the same degrees of freedom as in the half-truck models. The only change regarding the friction wedge in the full-truck model, is the addition of two friction wedges (and control coils) for the newly created sideframe to support the bolster.

3. NUCARS® MODEL

This chapter starts with a brief synopsis of the NUCARS software. The next sections discuss the modifications made to a standard 100 ton hopper NUCARS system file to be comparable to the multibody dynamics models. Also discussed are the types of wedge elements and their characteristics used within the NUCARS models.

The NUCARS (New and Untried Car Analytic Regime Simulation) software is a general purpose program developed by the Transportation Technology Center, Inc and used by various railroad companies and researchers over the world. NUCARS models the interaction of rigid or flexible bodies joined through suspension element connections. Degrees of freedom for each body can be defined by the user to include one degree of freedom or include a full six degrees of freedom (3 translational, 3 rotational). While nonlinear geometric effects are ignored, suspension nonlinearities are modeled exactly. Inter-body connection models are chosen from a library of element types. These include parallel or series springs and dampers, elements with hysteresis, connections with dry friction, and constraints. Connections may be defined as acting along or about the longitudinal, lateral, or vertical axes or any specified direction. Element characteristics are defined through piece-wise linear functions and nonlinearities are easily represented. Special characteristics are provided for the axle-to-track connection, using look-up tables for the wheel/rail profile contact geometry and the rolling contact forces.

Validation of NUCARS has been supported by both the AAR and the Federal Railroad Administration (FRA). In September 1989, the AAR performed the first of two Chapter XI validation tests. In each case, a full Chapter XI test was performed with the objective of critiquing the test procedures, analyses, and performance criteria based on the test results.

A second source of validation data has been the Lightweight Car (LWC) Test Program sponsored by the AAR and FRA. The objective of this test series was to determine whether relatively inexpensive tests and analyses could be devised to evaluate the safety aspects of new car and truck designs. Testing began in 1986 and included both stationary modal testing and running on perturbed and unperturbed track [3].

The NUCARS program has now been firmly established as a viable analytical tool and is a suitable program to use as a benchmark for the multibody dynamics models. The NUCARS models were modified to include the same degrees of freedom and body connections given to the multibody dynamics models. Within the NUCARS software a half-truck variably damped and half-truck constantly damped model have been created for comparison.

3.1. NUCARS Half-Truck Variably Damped Model

The NUCARS variably damped model was created by modifying an existing system file included within the program. The Lhoper-06-69-var-hollow (seen in Appendix 7.2.3) system file models a car supported by two full suspension trucks. To make an accurate comparison with the multibody dynamics model, the NUCARS model must be simplified to represent a half-truck geometry. Simplifying the connections used will reduce the likelihood of external forces, not modeled in the multibody dynamics model, which affect the wedge motion in the NUCARS model.

To simplify the NUCARS model many modifications needed to be made. First, the trailing truck was eliminated leaving only the leading truck. The car body was removed and simplified as additional mass lumped with the bolster mass. Next the right sideframe and connections between the bolster and car body were removed from the leading truck. Lastly the wheel – rail connections were eliminated placing the sideframe on the ground. Figure 3-1 shows a representation of the modified system file to be used in benchmarking the multibody dynamics model.

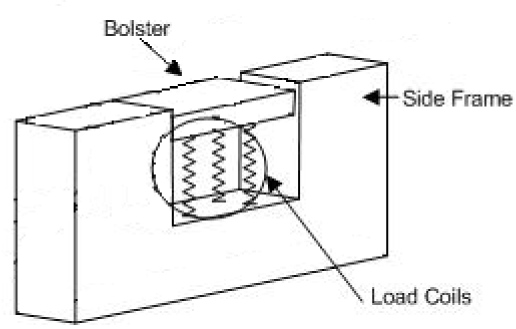


Figure 3-1. Representation of simplified NUCARS model used in benchmarking the multibody dynamics model.

After simplification of the NUCARS variably damped model comparisons with the multibody dynamics model can be made. For the comparison to the multibody dynamics model the type 6.9 wedge element within NUCARS were used in the model. The type 6.9 wedge element is a three dimensional model that includes lateral, longitudinal, and vertical wedge displacements and stick-slip friction.

NUCARS wedge elements lack mass and inertial properties of the wedges and are purely force elements within the model. The friction wedge elements are considered to be in quasi-static equilibrium meaning the net forces acting on the wedge are zero. This allows the friction forces on the column face (sideframe) to be calculated in terms of those on the slope face (bolster). The friction forces that arise are assumed to be the damping that acts on the system.

The model was modified to have the same characteristics as an actual 100 ton hopper suspension. These characteristics include one quarter of the car mass supported by a 22417.8 lb/in spring nest (supporting bolster) and a 1491 lb/in control coil (supporting each wedge). For these characteristics the static height and static load were calculated and used within the piece-wise functions that define each spring. Using a static height and load initializes the system from rest and allows the NUCARS simulations to begin at the same state as the multibody dynamics model simulations.

3.2. NUCARS Half-Truck Constantly Damped Model

The NUCARS constantly damped model was created by modifying an existing system file included within the program. The Lhoper-06-69-const-hollow (seen in Appendix 7.2.4) system file models a car supported by two full suspension trucks. To make an accurate comparison with the multibody dynamics model, the NUCARS model must be simplified to represent a half-truck geometry as previously mentioned for the variably damped NUCARS model.

After simplification of the NUCARS constantly damped model comparisons with the multibody dynamics model can be made. For the comparisons three types of wedge elements within NUCARS were used in the model. The first wedge element is a type 6.7 wedge element model. The type 6.7 wedge element is a two dimensional friction surface with slider dry friction damping that includes later and vertical wedge displacements. The next wedge element is a type 6.8 wedge element model. The type 6.8 wedge element is a two dimensional model that includes lateral and vertical wedge displacements with stick-slip friction. The last wedge element used in the simulations is a type 6.9 wedge element model. The type 6.9 wedge element is a three dimensional model that includes lateral, longitudinal, and vertical wedge displacements and stick-slip friction. All wedge elements used for the NUCARS simulations lack mass and inertial properties of the wedge.

The model was modified to have the same characteristics as an actual 100 ton hopper suspension. These characteristics include one quarter of the car mass supported by a 22417.8 lb/in spring nest (supporting bolster) and a 1979.3 lb/in control coil (supporting each wedge). To initialize the system for these characteristics the static height and static load were calculated and used within the piece-wise functions that define each spring.

4. RESULTS

This section presents the results from the multibody dynamics model and NUCARS model simulations. For the models a toe case comparison was first simulated within the multibody dynamics model. The toe cases include toe-out, no-toe, and toe-in. After a discussion of these results the multibody dynamics model results are compared with NUCARS results.

4.1. Half-truck Variably Damped Model Results

4.1.1. Multibody dynamics Model

The half-truck variably damped model was simulated using a 0.48 Hz input with amplitude of 1 inch to the sideframe. This input was kept constant for the simulations while three static toe angles were used to alter the model geometry. The toe angles used were; no-toe (0 rad), toe-out (-0.003 rad), and toe-in (0.003 rad). Varying the toe angle simulates sideframe wearing as seen in actual train suspensions. The sideframe starts in a toe-out configuration and wears through no-toe geometry and is considered worn when the sideframe geometry is toed in.

Figure 4-1 compares the vertical wedge forces generated by each toe configuration. All of the toe cases have similar trends in magnitude and shape throughout the oscillations with an exception at 1.5 sec. This dissimilar trend in the vertical forces can be linked to the system initializing to the sideframe input. The sharp change in wedge displacement that causes this slip during the beginning of the simulation can be seen just before 2 seconds in Figure 4-2. After two seconds in the simulation the sideframe movement and bolster movement approach a steady state oscillation and this large slip of the wedge on the sideframe surface disappears. One cycle of the stroke of the bolster is defined in Figure 4-3.

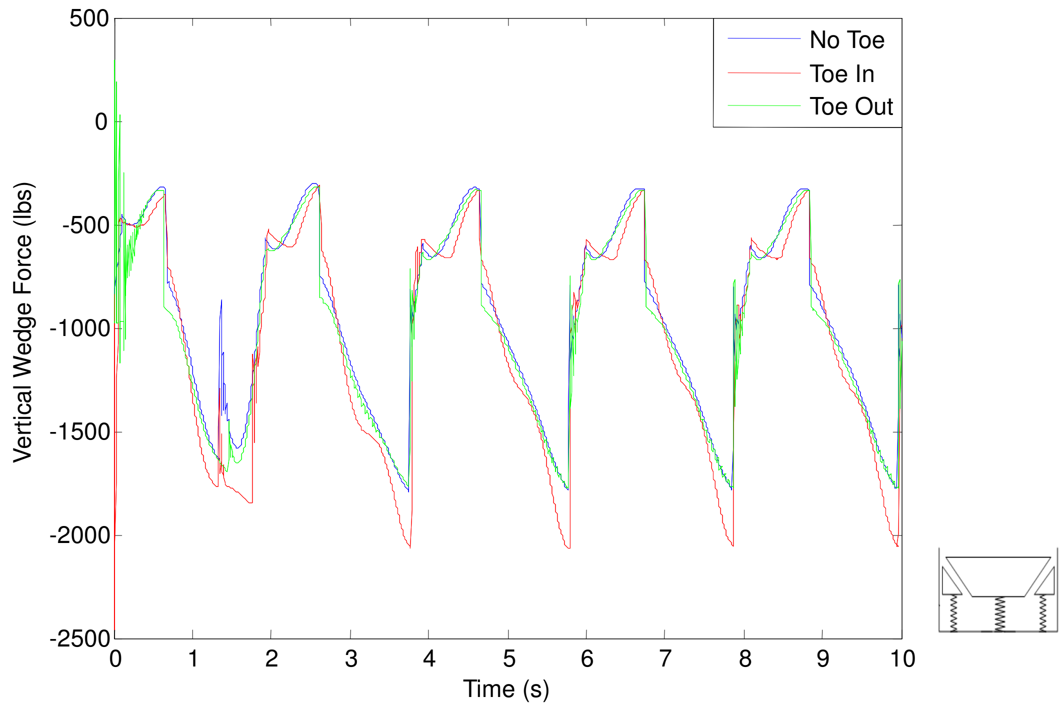


Figure 4-1. Comparison of vertical wedge forces for the multibody dynamics half-truck variably damped model

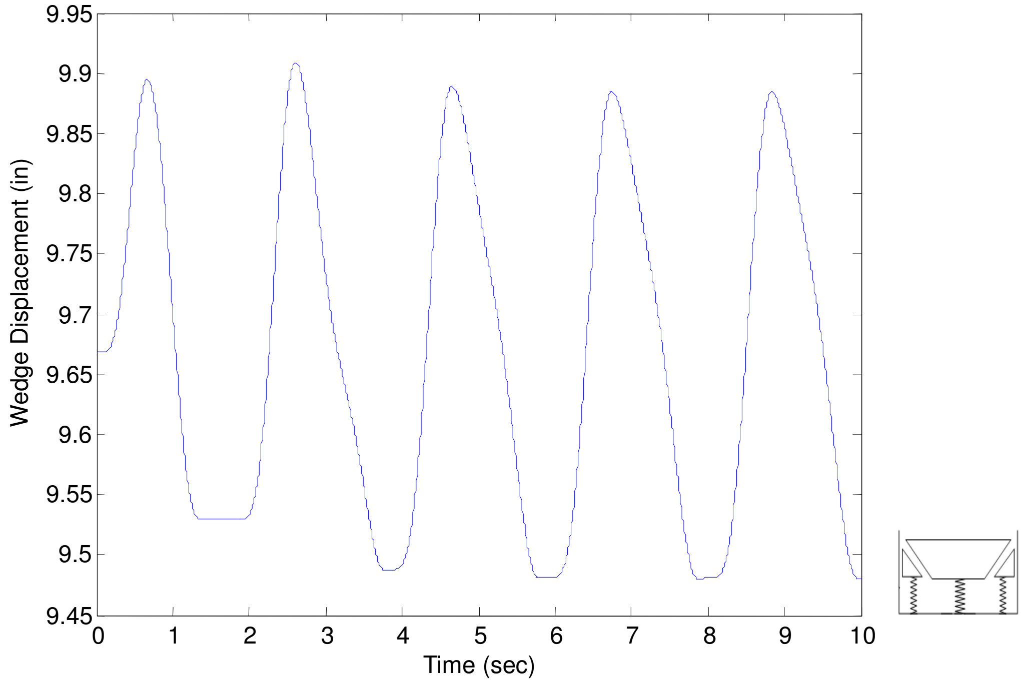


Figure 4-2. Vertical wedge displacement during simulation.

In Figure 4-3, section (1) represents the force due to the bolster moving down vertically, pressing against the wedge and forcing it to slip down the face of the sideframe. Section (2) is the stick phase of the motion when the wedge face friction force and the direction of the motion relative to the sideframe changes direction from sliding down to sliding up. Section (3) is the result of the wedge and bolster moving upwards and beginning to separate. In section (4) the bolster is at its maximum height and vertical force nears -300 lbs. If the magnitude of the inputs to the simulations had been large enough to induce liftoff, wedge forces at section 4 would oscillate around 0 lbs.

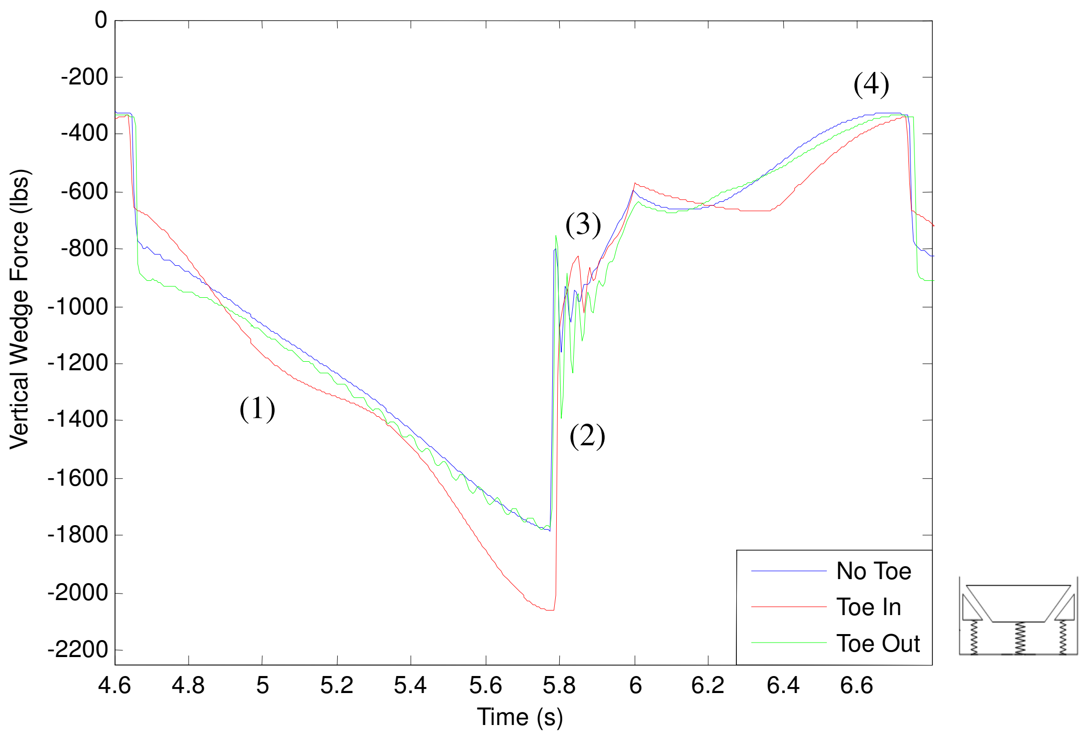


Figure 4-3. Comparison of the vertical wedge forces from 4.6 sec to 6.8 sec.

The vertical force hysteresis for one cycle of the bolster is shown in Figure 4-4. The sections labeled in Figure 4-4 correspond to those labeled in Figure 4-3. The amount of vertical force hysteresis indicated the amount of damping each toe configuration provides. Toe-in provides the most damping while toe-out provides the least. As expected a no-toe geometry lies between the toe-in and toe-out configurations.

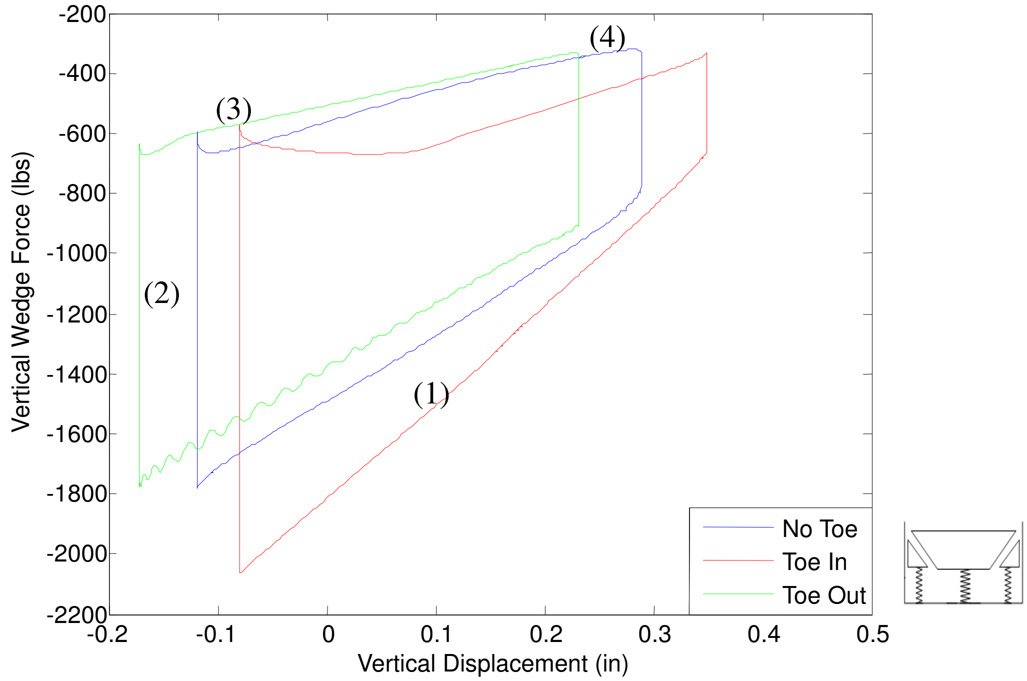


Figure 4-4. Vertical force hysteresis plot for the half-truck variably damped model.

The longitudinal forces acting on the wedge due to this contact can be seen in Figure 4-5. The toe-in geometry effectively increases the wedge angle. This increase in wedge angle, known as wedging, allows for more normal force to be transmitted from the wedge to the sideframe. The wedging effect can be seen in the greater longitudinal forces for toe-in geometry.

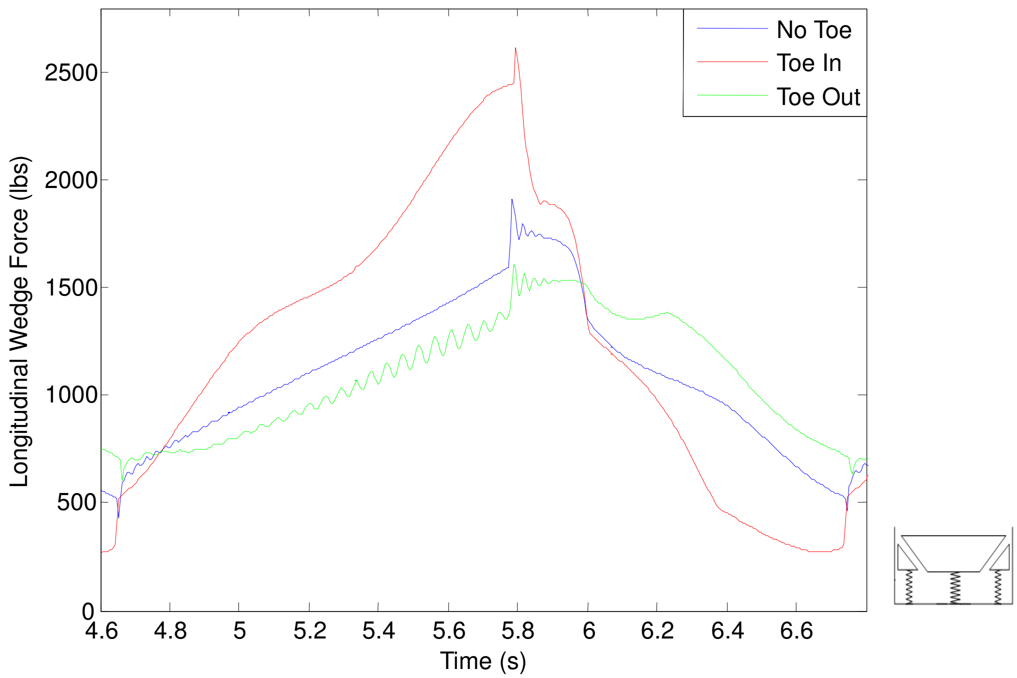


Figure 4-5. Longitudinal forces for toe configurations of the half-truck variably damped model.

From the longitudinal forces the rate of energy dissipated can be calculated to better analyze the damping characteristics of each toe configuration. A plot of the rate of energy dissipated for each toe case can be seen in Figure 4-6. In comparison the energy dissipated from the wedge – bolster contact is relatively small. For this reason only the sideframe rate of energy dissipated is shown.

The toe angles used in the toe cases are very small and as expected the energy dissipated is similar for each toe case. The toe-in case exhibits on average the highest energy dissipated due to the increased friction between the sideframe and wedge from the wedging phenomenon associated with toe-in geometry. The peaks on the energy dissipation figure correspond to the bolster position during the simulation. At 5.6 seconds the bolster is at the bottom of its stroke creating larger frictional and longitudinal forces for the toe-in geometry due to wedging. At 6.7

seconds the bolster is at the top of its stroke when the wedge and bolster are nearly separated. The toe-out geometry produces larger longitudinal forces in this instance because of prolonged contact with the bolster and sideframe due to the toe-out geometry.

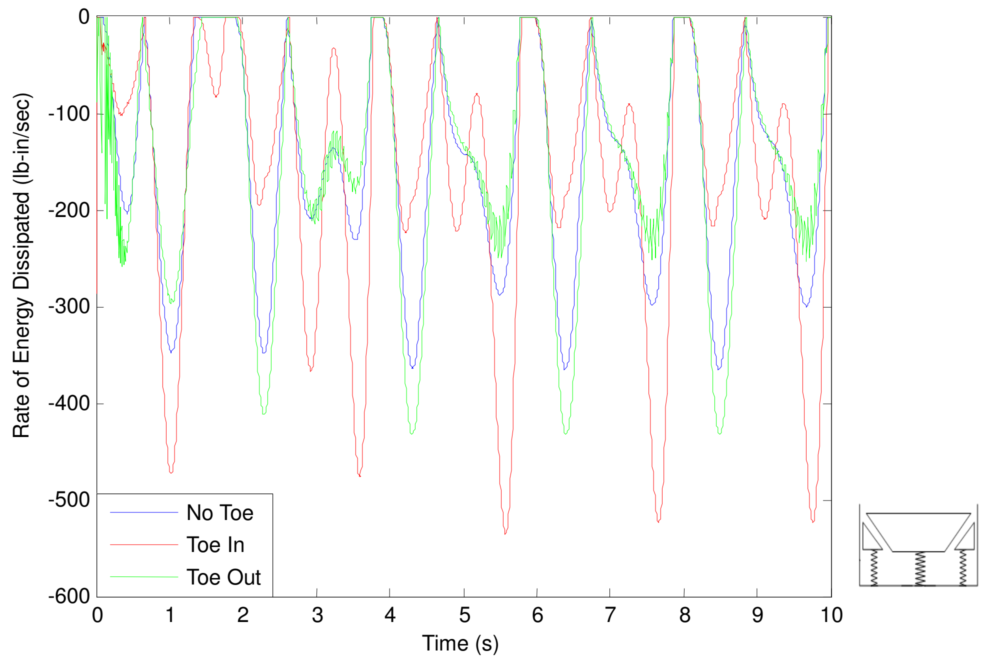


Figure 4-6. Rate of energy dissipated for each toe case in the half-truck variably damped model simulations.

One of the major advantages of the multibody dynamics model is its ability to calculate the moments acting on the wedge. Due to the simplistic inputs used for the simulations the salient moment generated is the pitch moment.

The moments generated for the no-toe and the toe-out configurations are negative because of the sign convention used and their geometry. The negative moment means that as the bolster moves down vertically the top of the wedge pitches into the bolster. Figure 4-7 demonstrates that as the toe angle is changed from no-toe to the toe-out configuration used in the simulations the magnitude of the pitch moment increases.

Conversely the toe-in configuration produces a positive pitch moment on the wedge. This positive moment is due to sign convention and the toe-in geometry. As the bolster moves down vertically the top of the wedge pitches away from the bolster and is pressed squarely against the sideframe.

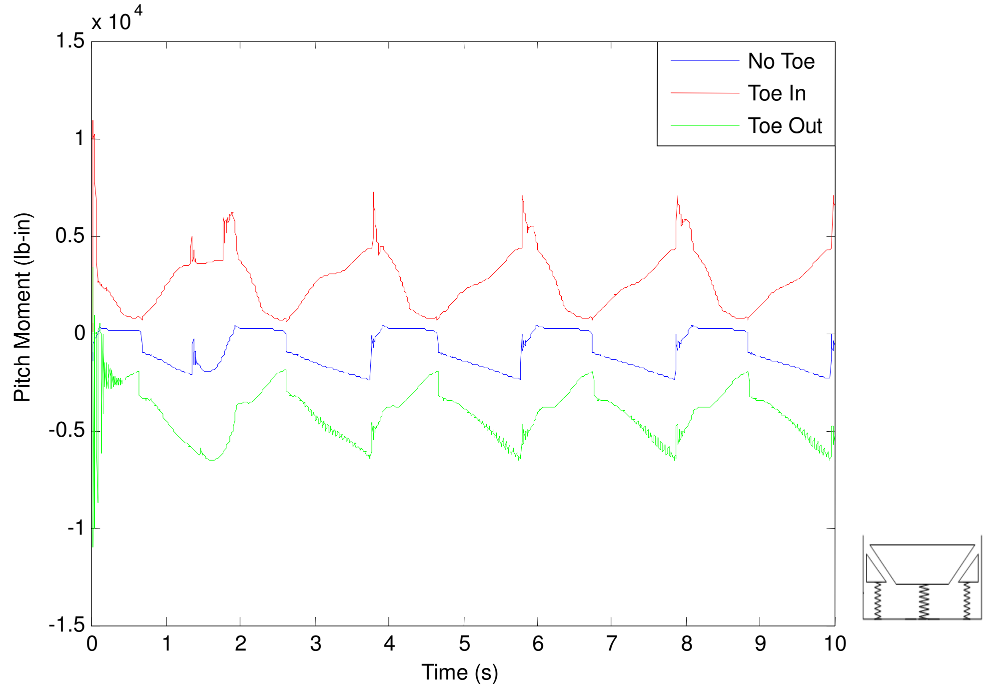


Figure 4-7. Pitch moments generated for each toe case in the half-truck variably damped model.

Increasing the number of contact points used in the multibody dynamics model allows for calculation of the forces and stresses across the wedge surfaces. Using force and stress distributions and material properties of the bodies a wedge model can be created that "wears out" over time based on wedge surface contact. Examples of the force outputs from the wedge - sideframe and wedge - bolster interaction can be seen in Figure 4-8 and Figure 4-9. These contour plots show the stress distribution across the wedge surfaces as the wedge reacts to contact with the sideframe and bolster. The "hotter" the area of the plot is, the higher the force in that area would be. Conversely the "colder" the area of the plot is, the lower the force in that area is expected to be.

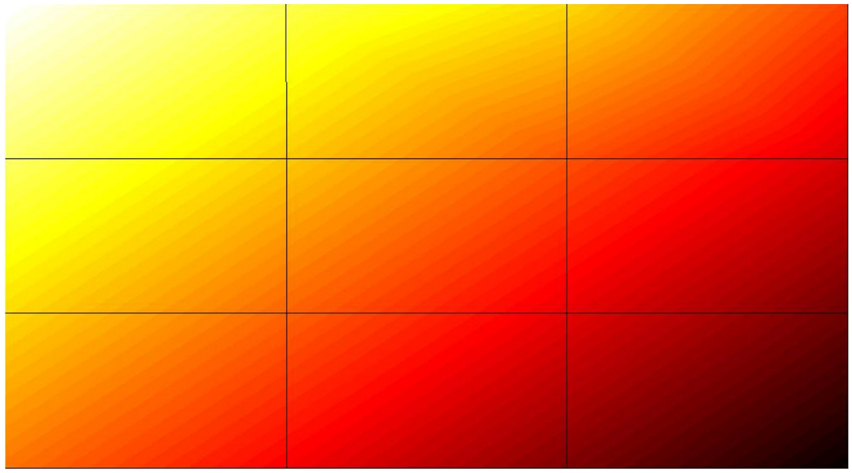


Figure 4-8. Wedge-bolster force distribution plot.

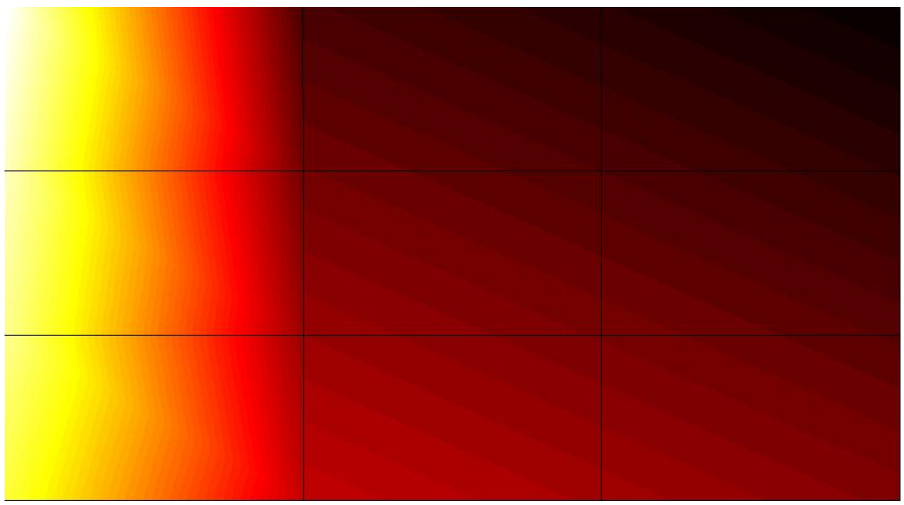


Figure 4-9. Wedge-sideframe force distribution plot.

These figures were captured during a simulation when the bolster was yawing. The bolster yawing "pinches" the front edge of the wedge into the sideframe as seen in the biased force distribution. The orientation of Figure 4-9 with respect to the wedge is shown seen in Figure 4-10.

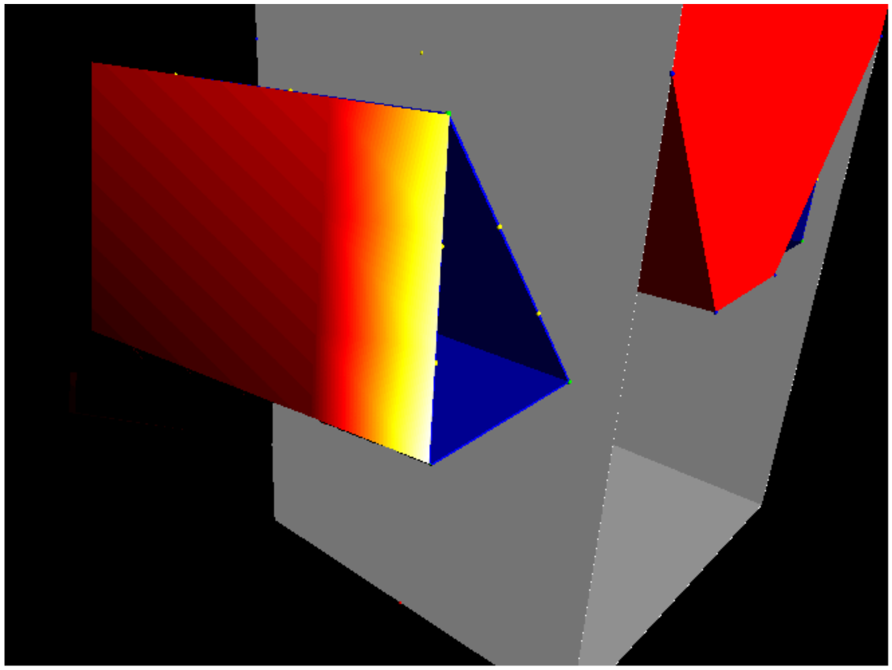


Figure 4-10. Orientation of the sideframe-wedge surface force plot.

4.1.2. NUCARS Comparison

In Figure 4-11 and Figure 4-12 the vertical wedge forces are compared for the NUCARS and the multibody dynamics model simulation. The toe case being compared is the no-toe configuration.

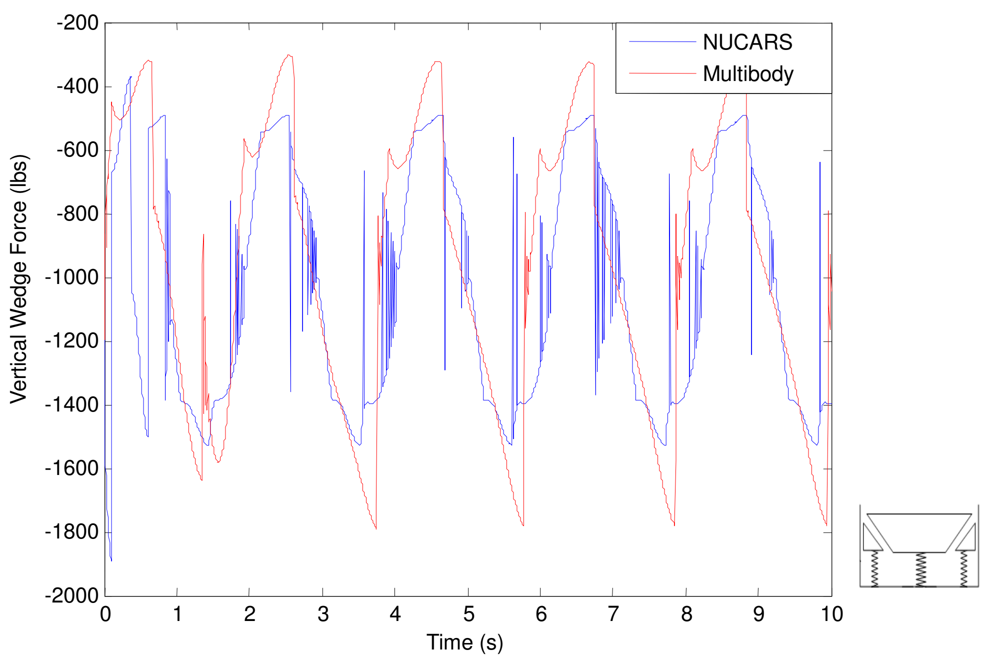


Figure 4-11. Comparison between NUCARS and the multibody dynamics model vertical wedge forces.

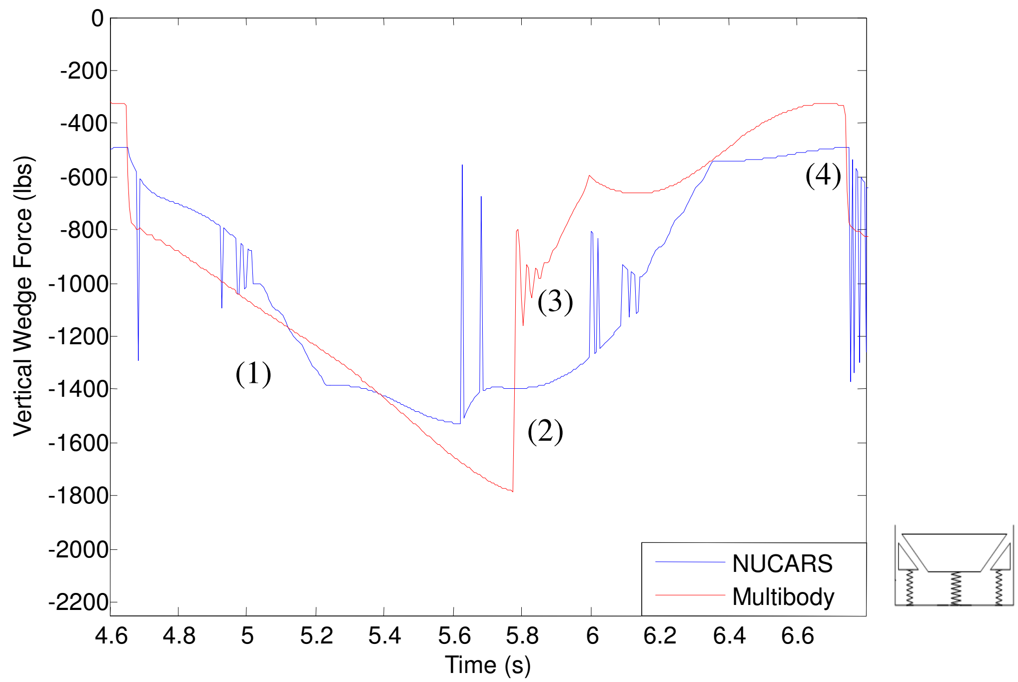


Figure 4-12. Comparison between NUCARS and the multibody dynamics model vertical wedge forces from 4.6 sec to 6.8 sec.

The two models exhibit similar behavior and vertical wedge force throughout the simulation. The labeled stages of the oscillation occur at approximately the same times during the simulations. Although the motion is relatively consistent between the models, the maximum and minimum vertical forces the models differ by approximately 250 lbs. The sharp changes in magnitude in the NUCARS model suggests stick-slip behavior is occurring more so than in the multibody dynamics model.

The increased sticking in the NUCARS model can be attributed to the difference in how the break out is defined in each model. NUCARS uses a force accumulator system to determine when the friction has been broken out where the multibody dynamics model relies on the multibody dynamic relations between the bodies. To better illustrate the increased sticking of the NUCARS model the vertical force hysteresis is shown in Figure 4-13.

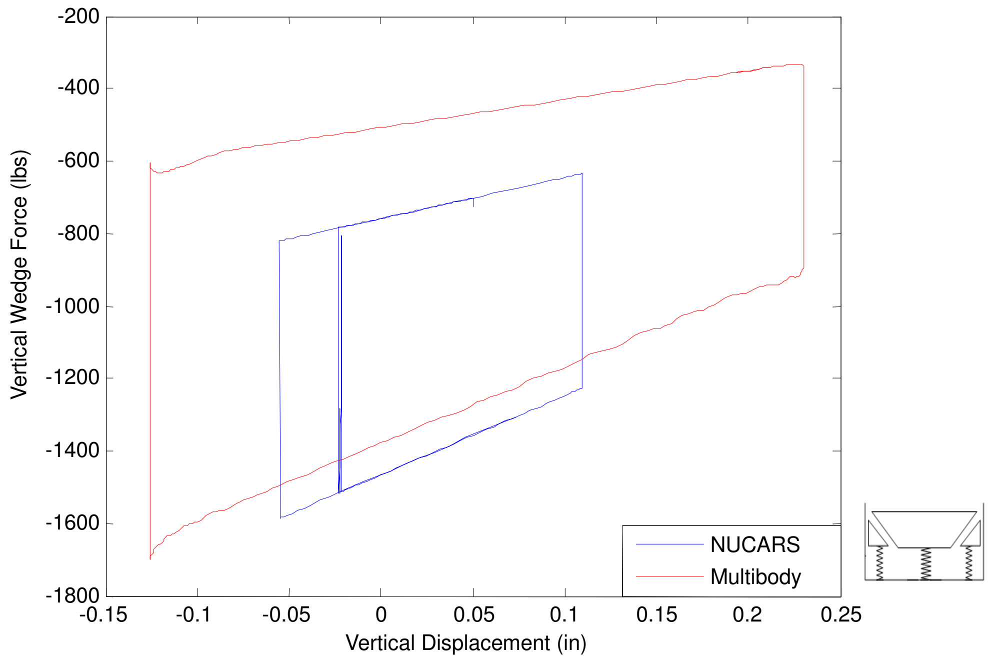


Figure 4-13. Vertical force hysteresis comparison of NUCARS to the multibody dynamics model.

The wedge in the multibody dynamics model displaces approximately 0.36 inches and the NUCARS model displaces approximately 0.16 inches. The difference in wedge displacement results in the greater wedge forces being generated in the multibody dynamics model.

A comparison of the NUCARS and multibody dynamics model longitudinal forces can be seen in Figure 4-14 and Figure 4-15. In the figures the NUCARS and the multibody dynamics model maintain the same oscillatory motion with similar magnitudes. The increased sticking of the friction wedge as mentioned earlier can be seen in Figure 4-14 at approximately 4 sec and 8 sec. At these points in the simulation the longitudinal forces drastically increase indicating that the wedge – bolster system has locked due to the friction. The increased sticking and slipping on the sideframe surface can be seen in the drastic changes in the longitudinal forces.

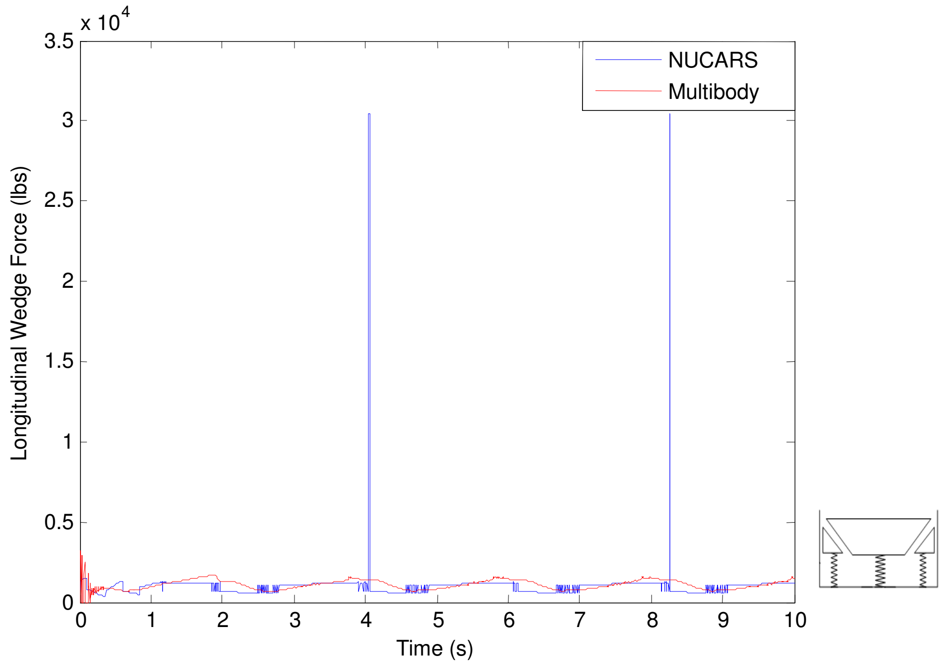


Figure 4-14. Comparison of NUCARS and multibody dynamics model longitudinal wedge forces.

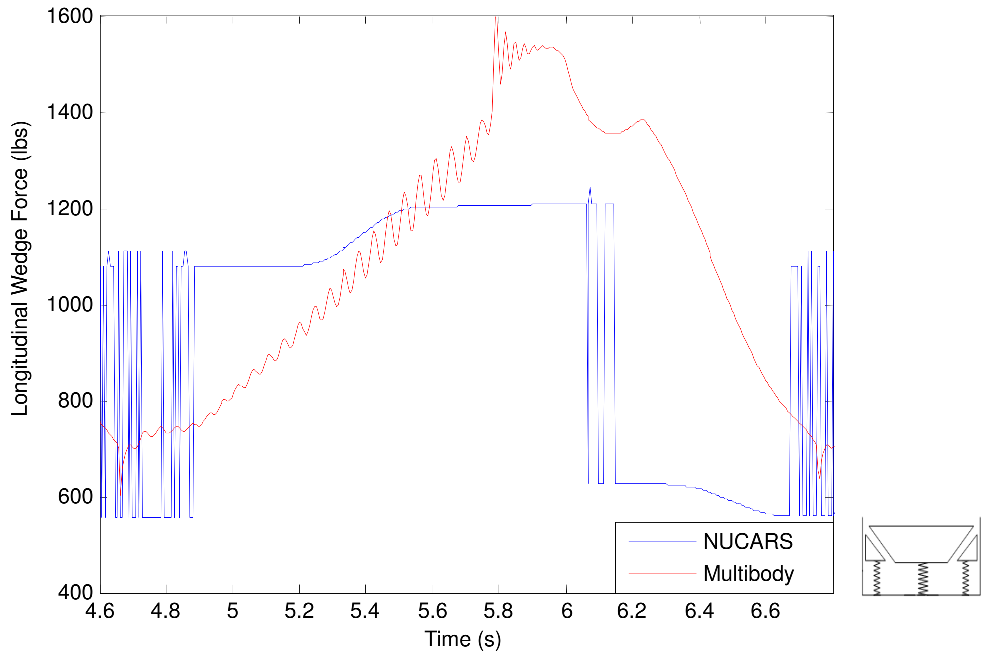


Figure 4-15. Comparison of NUCARS and multibody dynamics model longitudinal wedge forces from 4.6 sec to 6.8 sec.

4.2. Constantly Damped Model Results

4.2.1. Multibody dynamics Model

The half-truck constantly damped model was simulated using a 0.48 Hz input with amplitude of 3 inches to the sideframe. This input was kept constant for the simulations while three static toe angles were used to alter the model geometry. The toe angles used were; no-toe (0 rad), toe-out (-0.003 rad), and toe-in (0.003 rad).

Figure 4-16 compares the vertical wedge forces generated by each toe configuration. One cycle of the stroke of the bolster is defined in Figure 4-17.

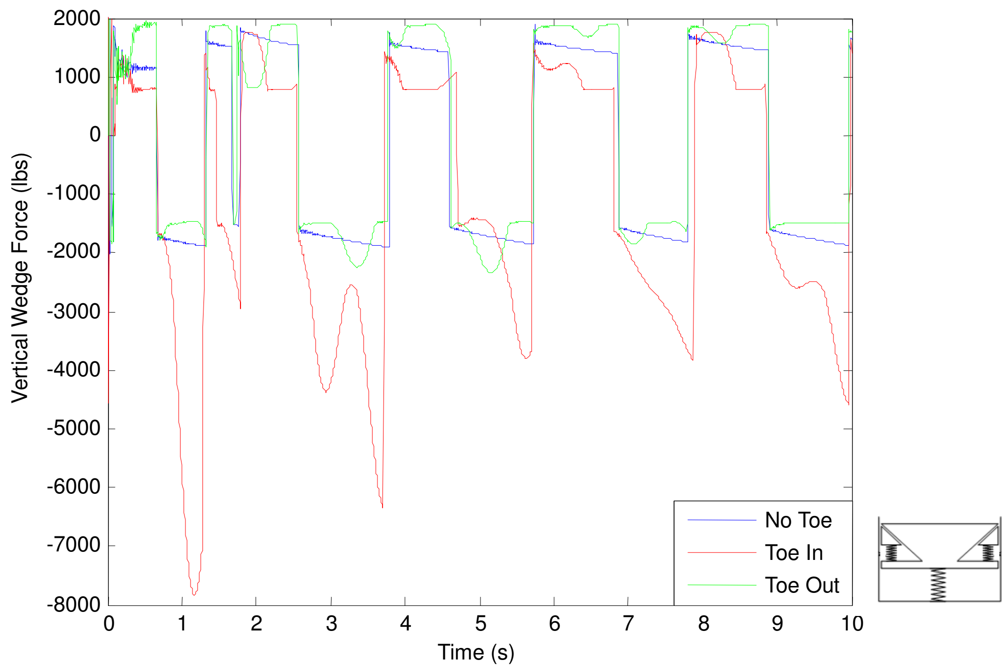


Figure 4-16. Comparison of vertical wedge forces for the multibody dynamics half-truck constantly damped model

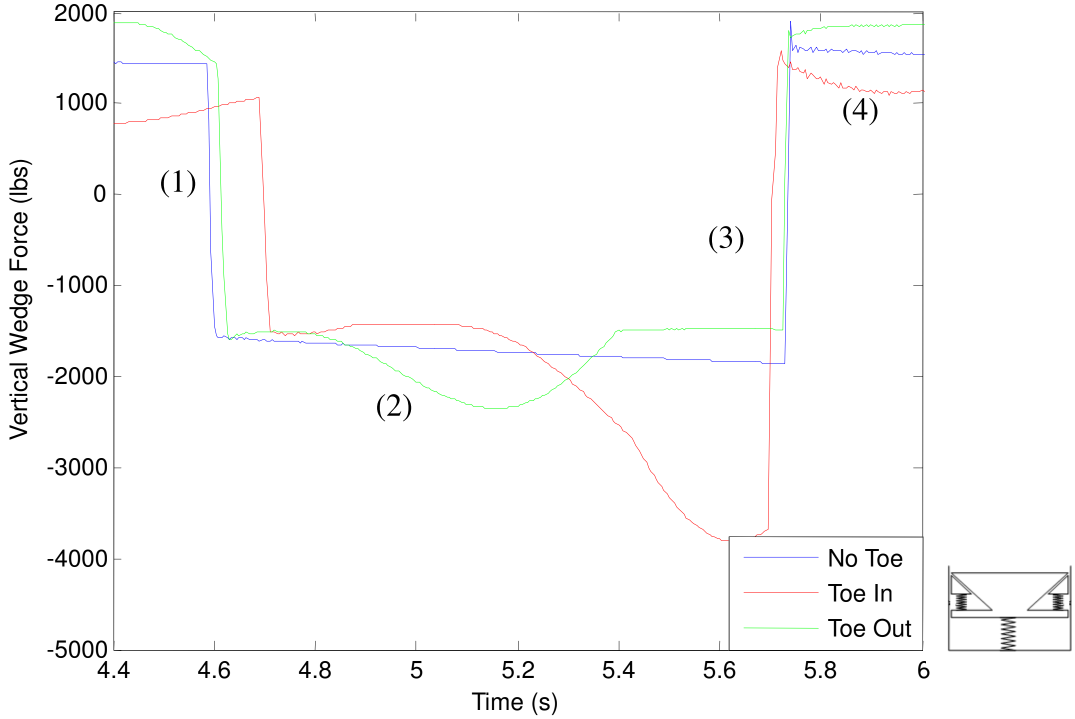


Figure 4-17. Comparison of the vertical wedge forces from 4.4 sec to 6.6 sec.

At the beginning of section (1) the wedge and bolster are at their maximum height. During section (1) the wedge sticks on the sideframe as the bolster displaces downwards, generating a large force. In section (2) the break out force needed to overcome the wedge – sideframe friction has been exceeded and the wedge slips upwards along the sideframe surface. Wedging in the toe-in geometry can be seen as the large vertical forces in section (2). In section (3) the wedge sticks on the sideframe as the bolster displaces upwards, generating a large force. In section (4) the break out force needed to overcome the wedge – sideframe friction has been exceeded and the wedge slips upwards along the sideframe surface.

Figure 4-18 shows the vertical force hysteresis plots for each of the toe cases. Due to the geometries defined by the toe configurations each wedge settles at a different static height and then displaces around that height. The wedge displacement for each case is small, so the hysteresis loops are narrow and little damping from friction is available. In Figure 4-19 the vertical force hysteresis plot for the toe-out configuration is labeled according to the sections labeled in Figure 4-17.

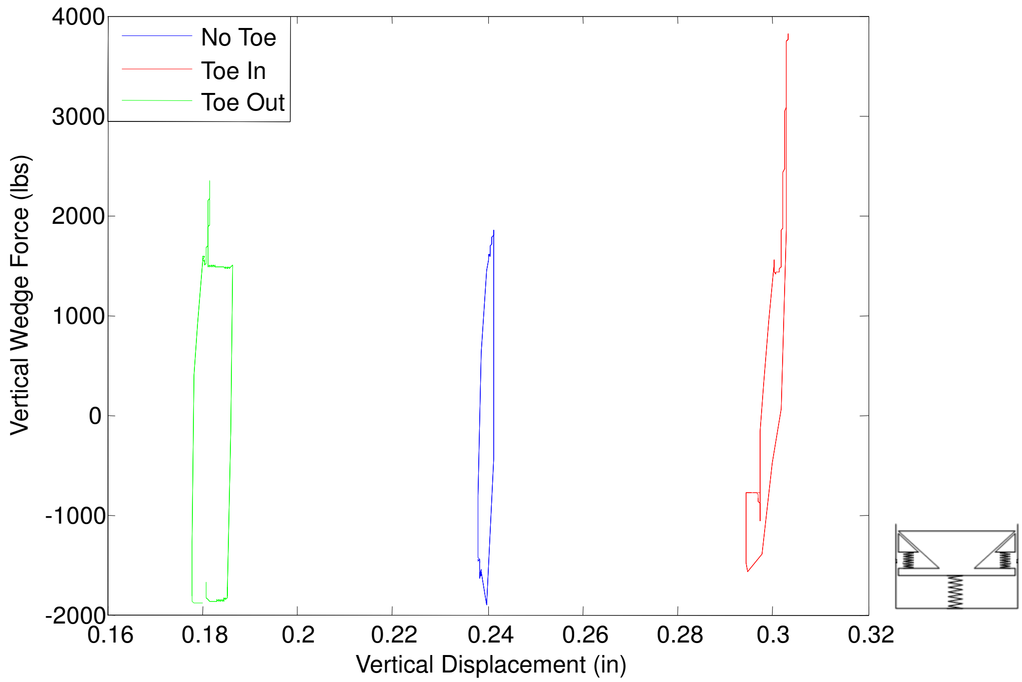
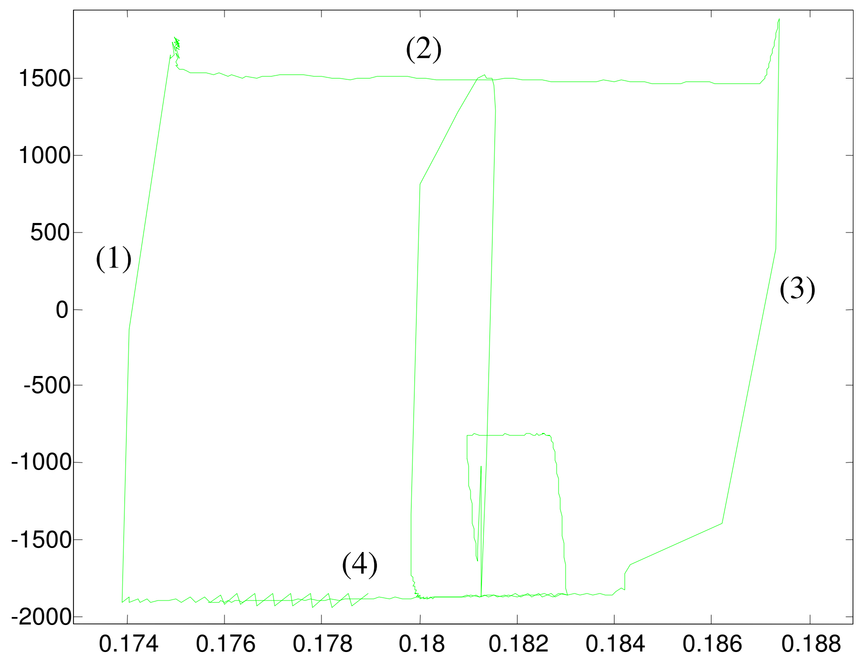


Figure 4-18. Vertical force hysteresis plot for the half-truck constantly damped model.



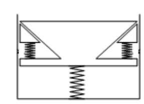


Figure 4-19. Vertical force hysteresis plot for the toe-out half-truck constantly damped model.

The longitudinal forces generated during the simulations can be seen in Figure 4-20. As expected the toe-in geometry creates larger forces due to wedging. Using the longitudinal forces the rate of energy dissipated due to wedge - sideframe contact can be calculated. The rate of energy dissipation graph, shown in Figure 4-21, has been labeled with arrows indicating the direction of bolster displacement. When the bolster is at the bottom of its stroke larger frictional and longitudinal forces are created for the toe-in geometry due to wedging. When the bolster is at the top of its stroke the wedge and bolster are nearly separated. The toe-out geometry produces larger longitudinal forces in this instance because of prolonged contact with the bolster and sideframe due to the toe-out geometry.

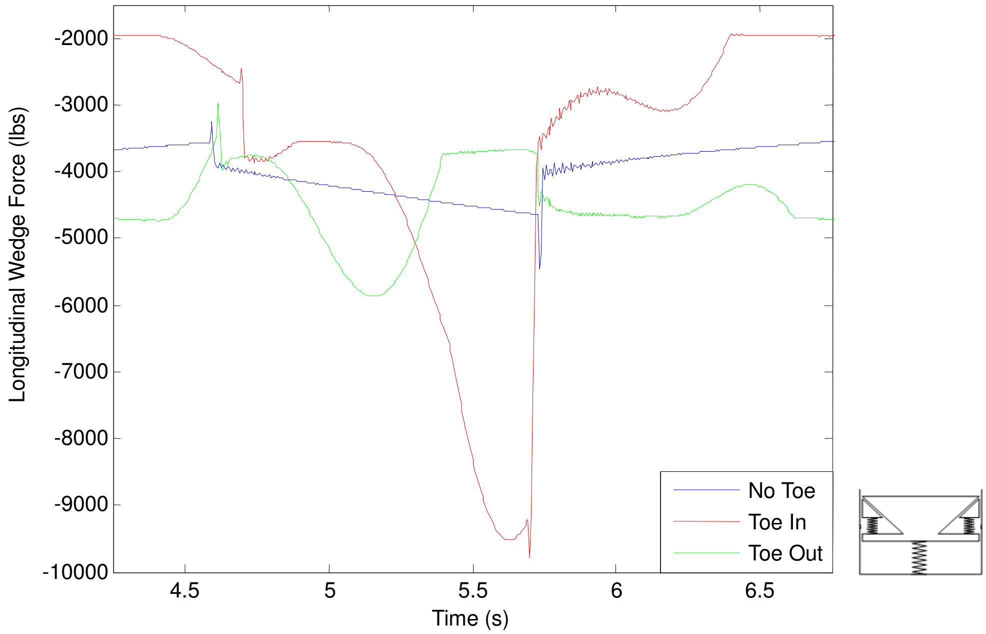


Figure 4-20. Plot showing longitudinal forces acting on the wedge due to wedge – sideframe interaction.

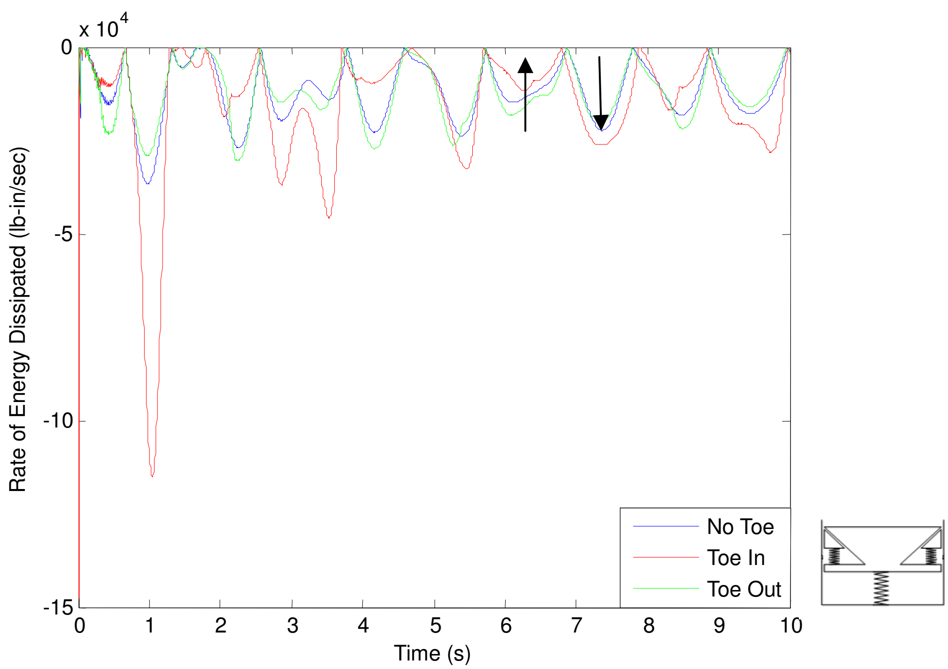


Figure 4-21. Comparison of rate of energy dissipated due to wedge - sideframe interaction for each toe case.

The pitch moment for the wedge can be computed for each toe case in the multibody dynamics model. For the simplistic inputs used in the simulations the roll and yaw moments are negligible as compared to the pitch moment. Ignoring the initial settling moments the constantly damped model generates small moments due to the geometric constraints of the system. In the constantly damped model the friction wedges sit within a pocket on the bolster and are allowed little clearance to displace vertically and longitudinally.

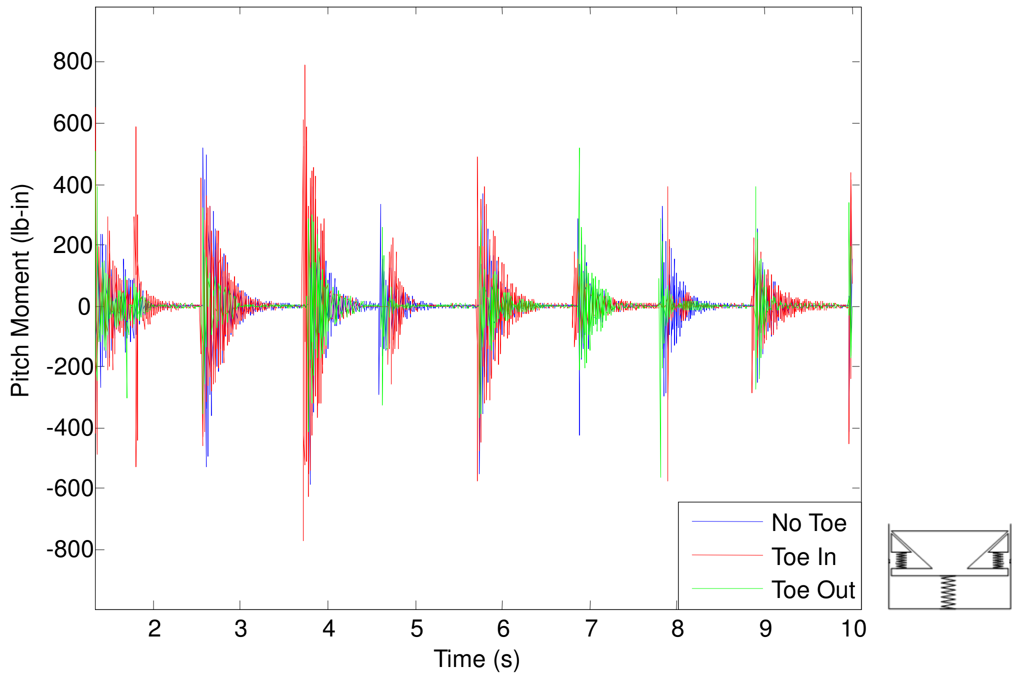


Figure 4-22. Pitch moment for toe cases in the half-truck constantly damped model simulations.

4.2.2. NUCARS Comparison

For comparison to the multibody dynamics model three different types of wedge element models were used in the NUCARS simulations. These wedge elements range from the least complex 6.7 wedge element to the latest iteration, the 6.9 wedge element with stick-slip friction. Type 6.7 wedge elements are two-dimensional friction surface with slider dry friction damping, with possibility of toe-in or toe out, and different coefficient on column wear plate and slope face. Type 6.8 wedge elements Friction wedge model with stick-slip friction. Type 6.9 wedge elements Stick-slip friction wedge model with longitudinal force transmission and concave column wear surface (variable toe-in/toe-out) [3]. Figure 4-23 and Figure 4-24 compare the vertical wedge forces for each model in the no-toe geometry.

The NUCARS models stick more than the multibody dynamics model in sections (1) and (3) of the motion. The increased sticking of the NUCARS models can be attributed to the force accumulator used to determine the breakout force needed to overcome frictional forces

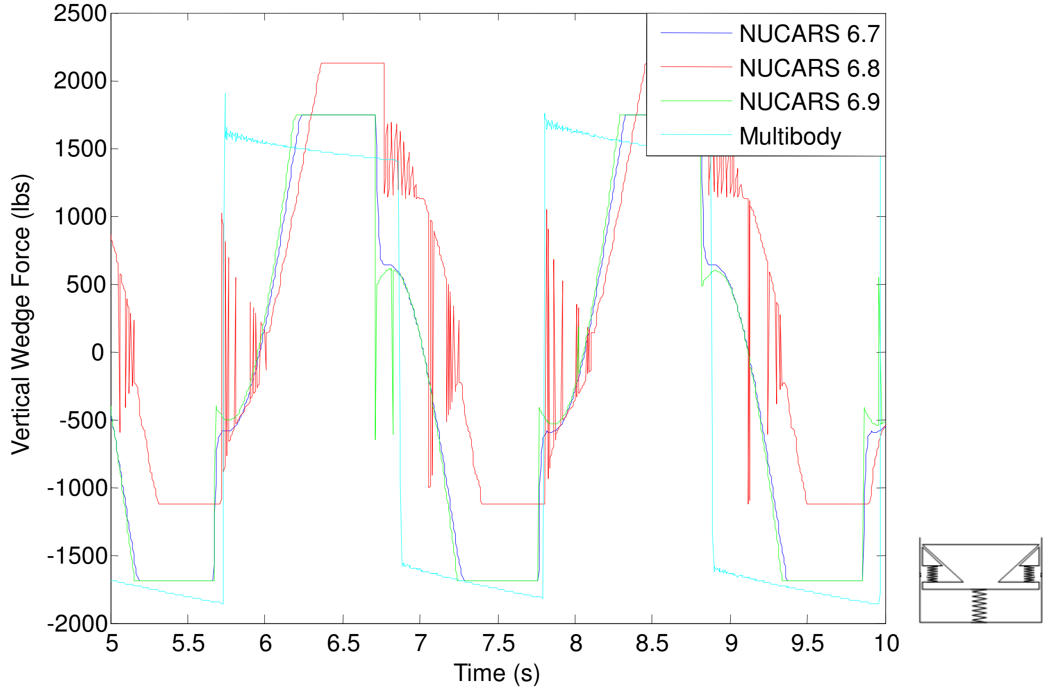


Figure 4-23. Comparison of vertical wedge forces between NUCARS models and multibody dynamics model.

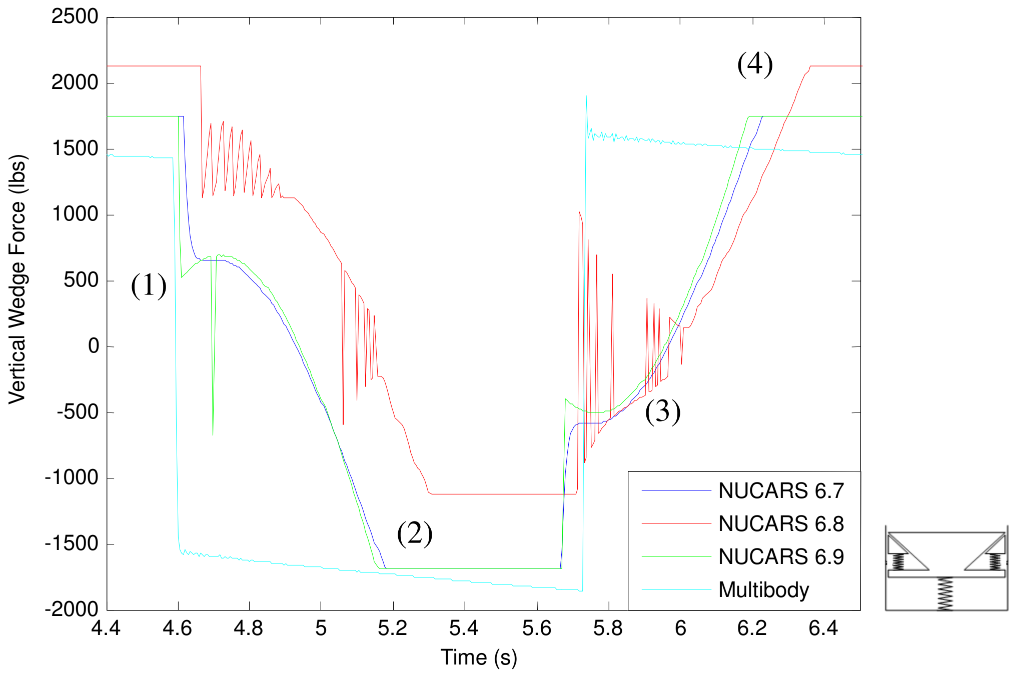


Figure 4-24. Comparison of NUCARS and multibody dynamics constantly damped model vertical forces for one cycle.

Figure 4-25 shows the vertical force hysteresis plots for the NUCARS and multibody dynamics model simulations. The multibody dynamics model has very little displacement as compared to the NUCARS models. For the multibody dynamics model the wedge is displacing vertically nearly the same amount as the bolster. Due to the force accumulator in NUCARS the wedge elements do not slip as easily causing the larger wedge displacements relative to the bolster.

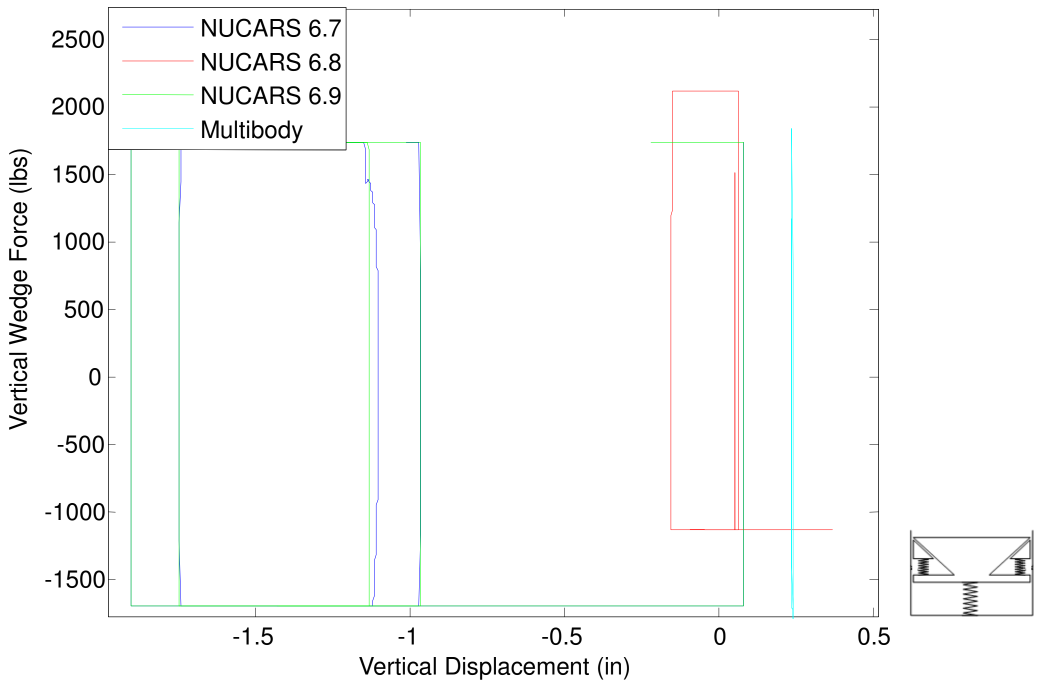


Figure 4-25. Vertical force hysteresis plot comparison of NUCARS results to the multibody dynamics model

In NUCARS type 6.7 and 6.8 wedge elements cannot calculate longitudinal forces, for this reason on the type 6.9 element is to be compared with the multibody dynamics model. Figure 4-26 shows a plot of the longitudinal wedge forces for the NUCARS type 6.9 wedge element and the multibody dynamics model.

The multibody dynamics model produces much larger longitudinal forces than the NUCARS model. The slipping along the sideframe is also much more gradual in the multibody dynamics model compared with the sharp instances of contact in the NUCARS model. The multibody dynamics model captures the gradual application of the load onto the wedge during the bolster motion. Section (1) defines the loading of the wedge and section (2) defines the unloading of the wedge.

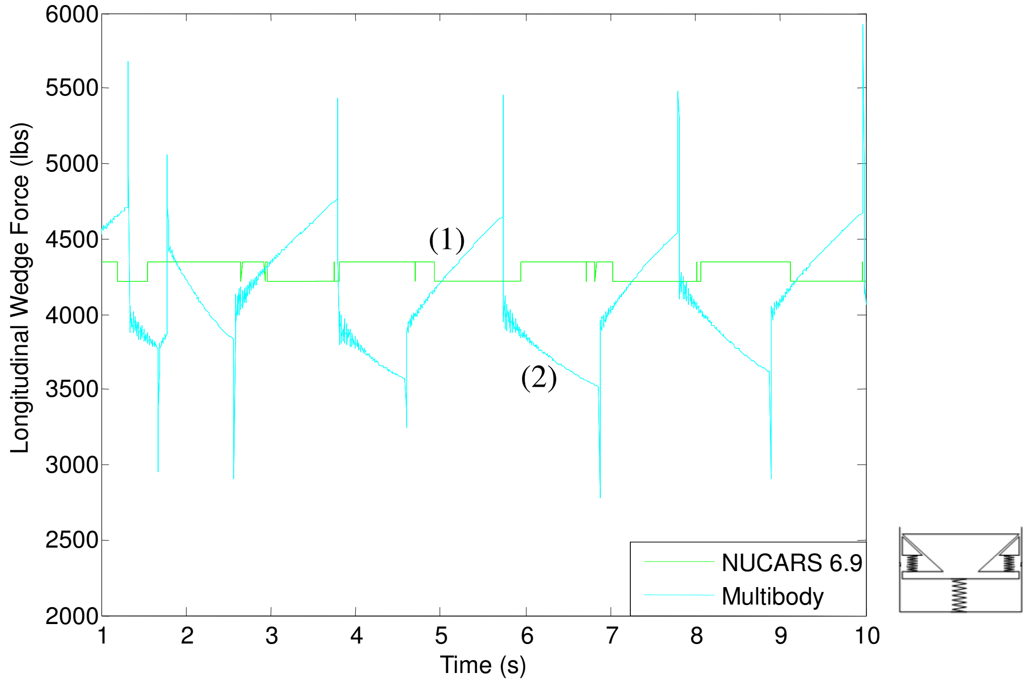


Figure 4-26. Plot of longitudinal wedge forces for NUCARS type 6.9 wedge element and the multibody dynamics model

4.3. Quarter-truck Variably Damped Model with Wedge Face Curvature Results

The wedge curvature model was simulated using a 0.8 Hz input with amplitude of 1 inch to the sideframe. While the input was kept constant for the simulations the toe angle for each simulation was changed to no-toe (0 rad), toe-out (-0.003 rad), and toe-in (0.003 rad).

Figure 4-27 compares the vertical wedge forces for each toe case for the wedge curvature model. The wedge forces for each case have similar magnitudes and shapes throughout the simulations. The simulations return to approximately 2000 lbs after each cycle.

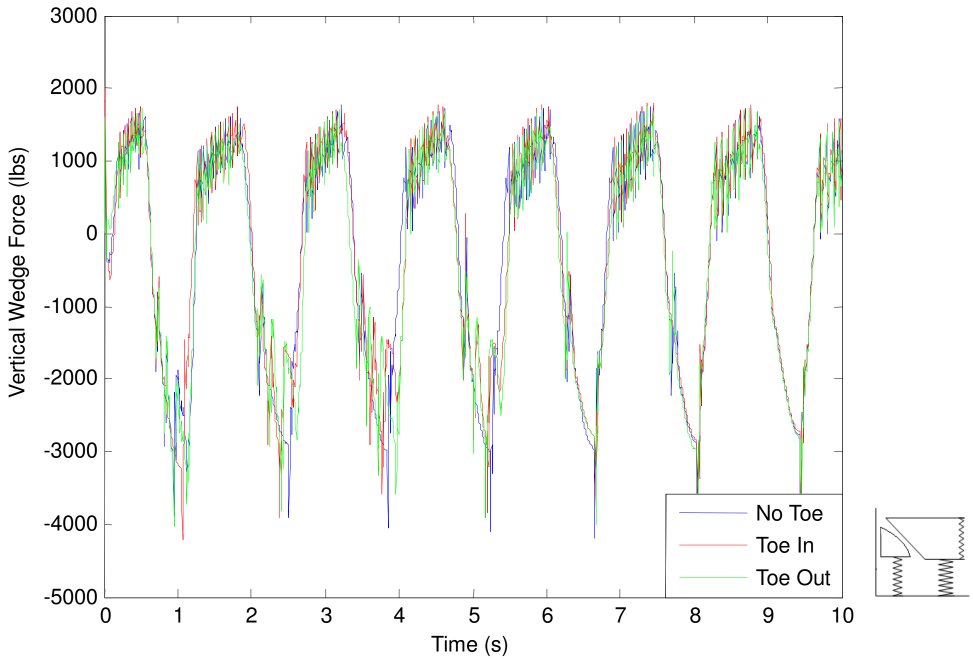


Figure 4-27. Comparison of vertical wedge forces for each toe case in the wedge curvature model simulations

In other simulations the vertical force would be expected to return to approximately 0 lbs because of lift-off or near lift-off conditions. During lift-off the only vertical force acting on the wedge is the mass of the wedge. The wedge mass is very small compared the bolster mass so the vertical forces generated are relatively small.

For the wedge curvature simulations the spring characteristics were maintained from the variably damped half-truck model. The variably damped model has an un-deformed control coil length of 10.25 in and a static height of 9.6 in. Due to the addition of the curved wedge surface the geometric interaction between the wedge and bolster changes, increasing the gap clearance between the two. This clearance decreases the static height of the control coil to 8.4 in. By using the same spring characteristics, the wedge curvature model now sees more load in the control coils due to the decrease in static spring height.

A large amount of high frequency noise is generated during the wedge curvature simulations. This high frequency noise originates from the contact conditions imposed on the model. The cylinder used to define the wedge surface can contact the bolster at any point, not just predefinded contact points as in the flat surfaced wedge models. This contact definition increases the number of contact instances for the wedge and because the wedge surface is now curved a pitch moment is created. The increased pitching results in the wedge ringing between the sideframe and bolster creating the high frequency noise.

Figure 4-28 illustrates the movement of the wedge by comparing the vertical wedge force to the wedge's vertical displacement for one period of the input. Section (1) represents the force due to the bolster moving down vertically, pressing against the wedge and forcing it to slip down the face of the sideframe. Section (2) shows the stick phase of the motion when the wedge face friction force and the direction of the motion relative to the sideframe changes direction from sliding down to sliding up. Sections (3) is the result of the wedge and bolster moving upwards and beginning to separate. In section (4) the bolster is at its maximum height and vertical wedge force nears 2000 lbs.

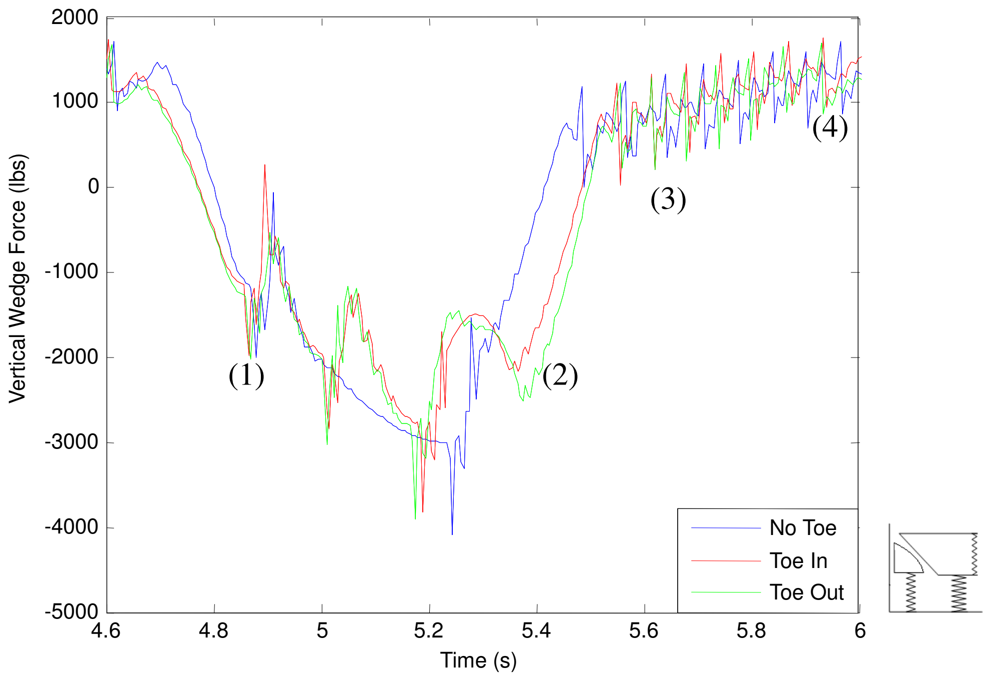


Figure 4-28. Comparison of vertical wedge forces for each toe case shown from 4.6 sec to 6 sec.

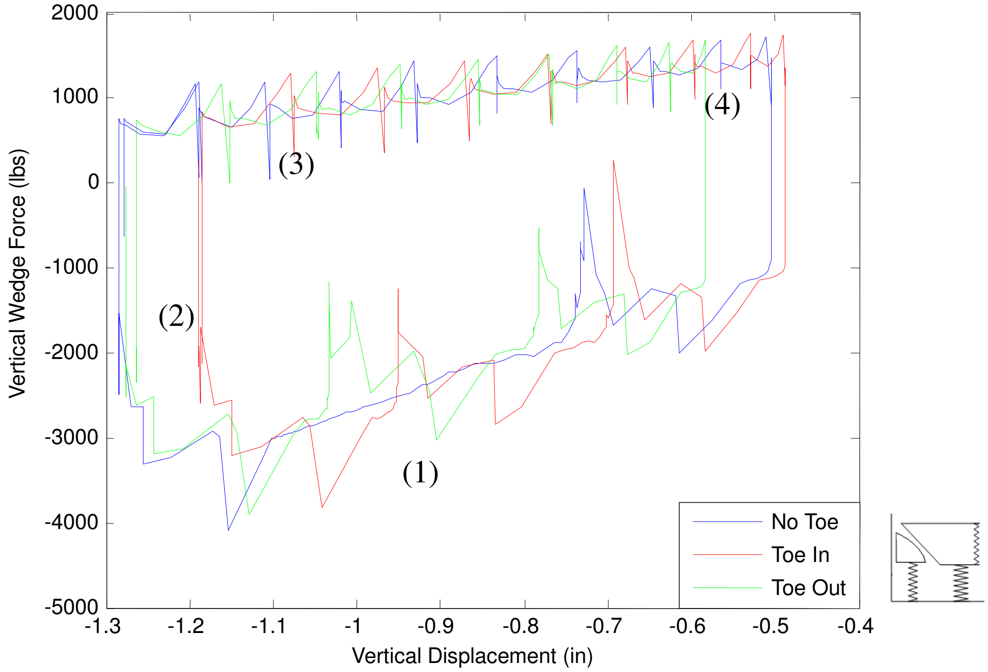


Figure 4-29. Vertical force hysteresis plot for the wedge curvature model toe cases.

The sections labeled in Figure 4-29 correspond to those labeled in Figure 4-28. This plot shows the vertical force hysteresis experienced during one period of the simulations. During section one of the oscillations the sharp peaks correspond to the stick-slip friction of the wedge. For the wedge curvature geometry the stick-slip friction is more prevalent than the half-truck variably damped model, which has a flat faced wedge. An example of the flat faced wedge can be seen in Figure 4-30. The labeled sections correspond to sections labeled in previous plots.

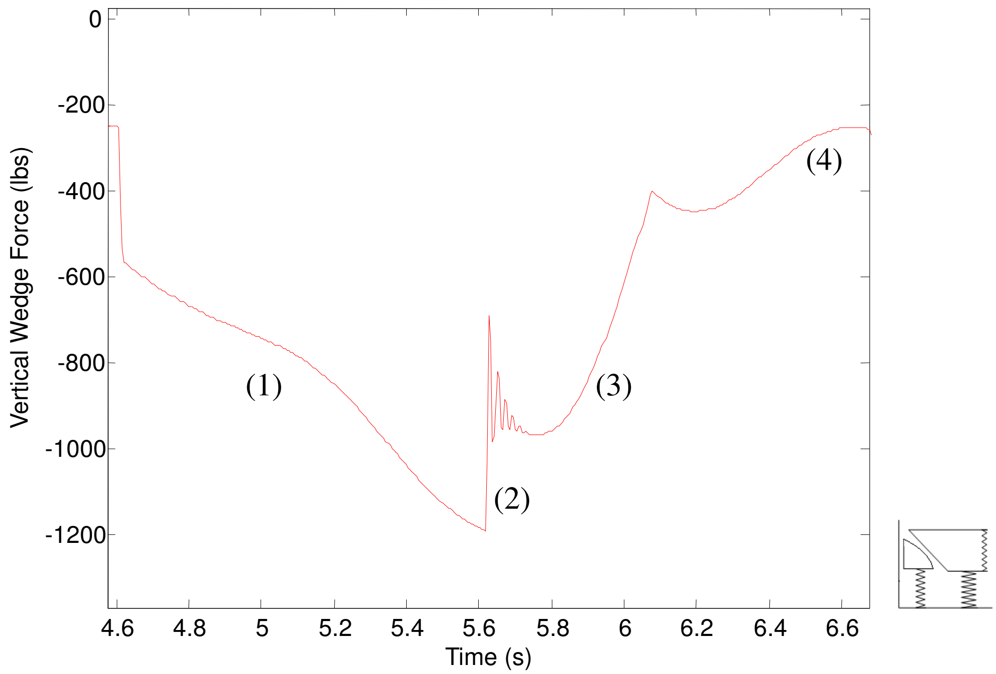


Figure 4-30. Figure demonstrating the smoother oscillations of the half-truck variably damped model with a flat wedge surface.

The longitudinal forces acting on the wedge due to this contact can be seen in Figure 4-31. From these forces the rate of energy dissipated can be calculated to better analyze the damping characteristics of each toe configuration. A plot of the rate of energy dissipated for each toe case can be seen in Figure 4-32.

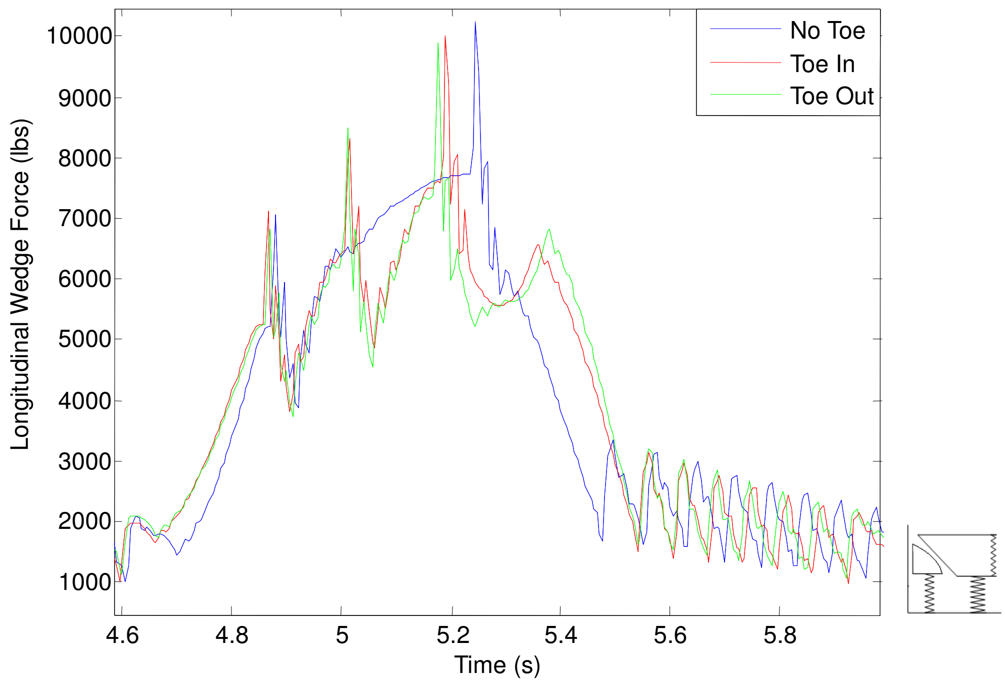


Figure 4-31. Plot showing longitudinal forces acting on the wedge due to wedge - sideframe interaction.

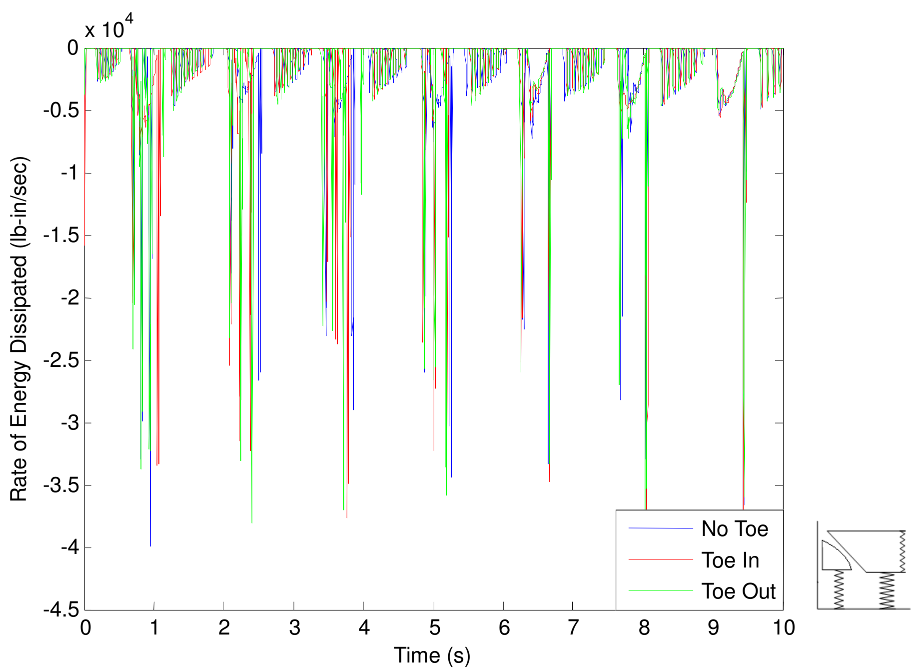


Figure 4-32. Comparison of rate of energy dissipated due to wedge - sideframe interaction for each toe case.

The pitch moment for the wedge can also be analyzed for each toe case in the wedge curvature model. For the simplistic inputs used in the simulations the roll and yaw moments are negligible as compared to the pitch moment. The pitching moment is usually generated by the stick-slip friction acting on the wedge faces. For the wedge curvature model the geometry of the wedge causes the wedge to pitch much more for the toe-out case as compared to the toe-in case. For these simulations the increased pitching of the wedge causes less wedge - sideframe contact and results in the lower rate of energy dissipated for the toe-out case. Figure 4-33 shows the pitch moments for each toe case.

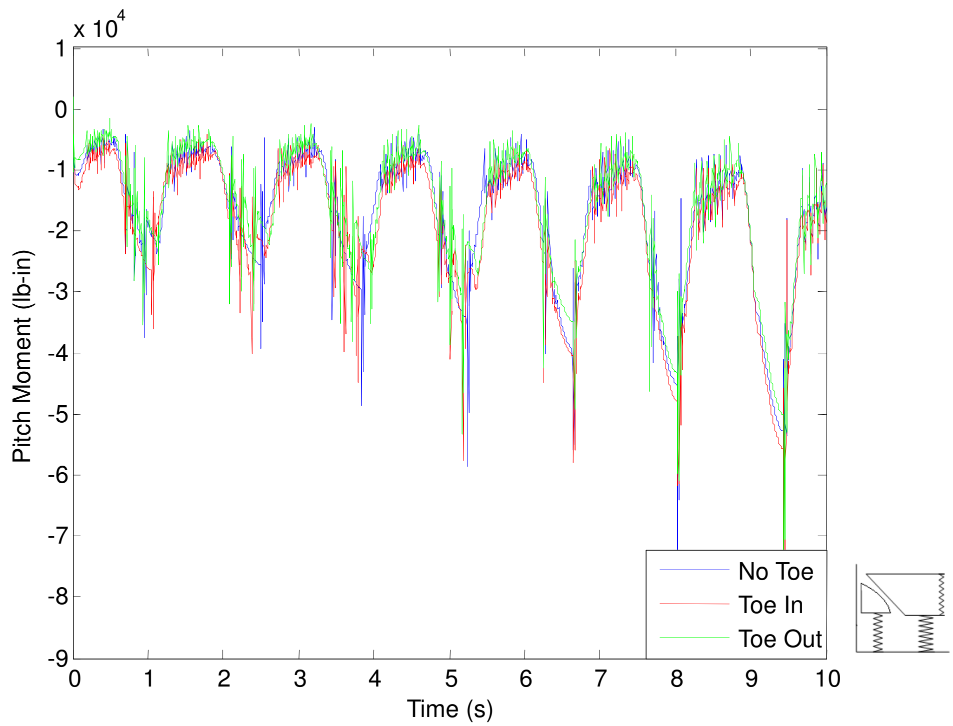


Figure 4-33. Pitch moment of the wedge for each toe case in the wedge curvature model.

5. CONCLUSIONS AND FUTURE WORK

This chapter discusses the conclusions drawn from the results presented in this thesis. In the second half of the chapter, future work for the next phases of this project is discussed.

5.1. Conclusions

This study focused on the continued development of the multibody dynamics friction wedge model within the multibody dynamics framework. The development of the multibody dynamics model gives an opportunity to improve wedge modeling by creating more realistic models. Improved wedge modeling can more accurately predict detrimental freight train phenomenon that can lead to derailment such as wedge lock up and truck hunting.

The variably and constantly damped quarter-truck models were expanded from quarter-truck to half-truck models. The friction wedges retain the same translational and rotational degrees of freedom as when modeled as a quarter-truck. These degrees of freedom include longitudinal and vertical translations and yaw and pitch rotations. For the half-truck models the input body was modified to receive inputs from the sideframe as with an actual three piece bogie, rather than the bolster in the quarter-truck models. These models were compared for all toe cases and benchmarked with equivalent models created in NUCARS software.

The variably damped model was also further modified to be able to better depict a warping scenario in addition to predicting forces and stresses across the wedge surfaces. For the warping model the wedges were given a lateral translation and the associated lateral friction. The formulation for friction used in previous models was modified to be limited to a maximum friction level based on the relative lateral and vertical wedge velocities.

Using a quarter-truck representation a curved surface wedge model was also created. Modifications to contact algorithms made unilateral contact over the curved surface possible. The geometry depicted in the curved surface model is the next iteration in the multibody dynamics friction wedge geometry and generates more robust friction wedge dynamics.

The variably damped half-truck and constantly damped half-truck models correlated well with the established train software NUCARS. For each model the NUCARS models produced more friction than the multibody dynamics models. This discrepancy in friction could be due to the friction formulations within each model. Questions about the frictional formulation can be solved by comparing the multibody dynamics model with test data. Further development of the full-truck variably damped model and warp model will give the opportunity to validate the multibody dynamics model.

5.2. Future Work

In the future the multibody dynamics friction wedge model will continue to be developed to more accurately capture friction wedge dynamics. The full-truck variably damped model will be benchmarked with an equivalent NUCARS model in addition to the creation of a full-truck constantly damped model. These models will be the first iterations in the multibody dynamics friction wedge modeling to be comparable to a simplified freight train full-truck suspension. The full-truck representations will be able to mimic actual sections of track rather than the simplistic inputs currently being used in the half and quarter-truck models.

With warp test data provided by TTCi the warp model can be validated in the future. Validation of the warp model offers the unique opportunity to improve the effectiveness of current warp testing procedures. Insightful information provided by the warp model can lead to additional instrumentation procedures to collect valuable data to aid in truck warp characterization.

Based on the wedge curvature model and half-truck variably damped model that can predict wedge surface forces and stresses a contact varying wedge model will be created. This model will be based on wedge, bolster, and sideframe material properties and apply principals of tribology based on observed contact forces and stresses.

6. REFERENCES

1 Iwnicki, Simon. Et al. "Handbook of Railway Vehicle Dynamics." Taylor & Francis group. New York, 2006.

2 Steets, Jennifer. "A Multibody dynamics approach to the modeling of friction wedge elements for freight train suspensions." Masters thesis, Virginia Tech 2007.

3 *NUCARS® Version 2006 Users Manual*, Transportation Technology Center, Inc., Pueblo, CO.

4 Dankowicz, H., *Multibody Mechanics and Visualization*, Springer-Verlag, 2004.

5 Okamoto, Isao, "How Bogies Work," *Railway Technology Today*, Japan Railway & Tranport Review 18, December 1998.

6 Vampire: Rail Vehicle Dynamics Software. http://www.vampire-dynamics.com/index.html

7 Klauser, P. E. "Modeling Friction Wedges Part I: The State-of-the-Art," of IMECE04, November 13-20, 2004, Anaheim, CA.

8 Klauser, P. E., "Modeling Friction Wedges Part II: An Improved Model," Proceedings of IMECE 04, November 13-20, 2004, Anaheim, CA.

9 Cusumano, J.P. and Gardner, J.F., "Dynamic Models of Friction Wedge Dampers," Proceedings of the 1997 IEEE/ASME Joint Rail Conference, March 18-20, Boston, MA.

10 Harder, R. F., "Dynamic Modeling and Simulation of Three-Piece North American Freight Vehicle Suspensions with Non-linear Frictional Behavior Using ADAMS/Rail," 5th ADAMS/Rail User's Conference, May 10-12, Haarlem, 2000.

11 Grassie, S., Iwnicki, S.D. and Kik, W., "Track Settlement Prediction using Computer Simulation Tools," Rail Technology Unit, Manchester Metropolitan University, Manchester, United Kingdom, http://www.railtechnologyunit.com, 2004.

12 Bosso, N., Gugliotta, A. and Somá A., "Multibody Simulation of a Freight Bogie with Friction Dampers," Proceedings of 2002 ASME/IEEE Joint Rail Conferecence, April 23-25 2002, Washington, DC.

13 Berghuvud, A. and Stensson, A., "Consequences of Nonlinear Characteristics of a Secondary Suspension in a Three-Piece Bogie," *Vehicle System Dynamics*, 2001, Vol. 36, No. 1, pp. 37-55.

14 McKisic, A.D., Ushkalov, V. F., and Zhechev, M., "Possibility of Jamming and Wedging in the Three-Piece Trucks of a Moving Freight Car," *Vehicle System Dynamics*, Vol. 45, No. 1, pp. 61-67, January 2007.

15 Kaiser, A. B., Cusumano, J.P., and Gardner, J. F., "Modeling and Dynamics of Friction Wedge Dampers in Railroad Freight Trucks." International Journal of Vehicle Mechanics and Mobility, 2002.

16 Chandiramani, N. K., Srinivasan, K., and Nagendra, J. "Experimental Study of stick-slip dynamics in a friction wedge damper." Journal of Sound and Vibration, 2006.

7. APPENDIX

7.1. Multibody Dynamics Models

7.1.1. Half-truck Variably Damped Model

7.1.1.1. Half-truck Variably Damped SRC file

```
with(Mambo):

#Declare observers, points, and triads
DeclareObservers(world,bolster,bolster3,bolster4,wedge1,wedge2,sideframe,sideframe1,sideframe2):
DeclarePoints(W,Bolster,Bolster1,Bolster2,Bolster_cg,Bolster3,Bolster4,Bolster5,Bolster6,Bolster7
,Bolster8,Bolster9,Bolster10,

Bolster11,Bolster12,Bolster13,Bolster14,Wedge1_1,Wedge1_2,Wedge1_3,Wedge1_4,Wedge1_5,Wedge1_6,Wed
ge1_7,Wedge1_cg,

Wedge1_8,Wedge1_9,Wedge1_10,Wedge1_11,Wedge1_12,Wedge1_13,Wedge1_14,Wedge1_15,Wedge1_16,Wedge1_17
,Wedge1_18,Wedge1_19,Wedge1_20,Wedge1_21,

Wedge1_22,Wedge1_23,Wedge1_24,Wedge1_25,Wedge1_26,Wedge1_27,Wedge1_28,Wedge1_29,Wedge2_1,Wedge2_2
,Wedge2_3,Wedge2_4,Wedge2_5,Wedge2_6,Wedge2_7,Wedge2_cg,

Spring1_1,Spring1_2,Spring1_3,Spring2_1,Spring2_2,Spring2_3,Spring3_1,Spring3_2,Spring3_3,S,Sidef
rame1,Sideframe2,Sideframe3,Sideframe4):
DeclareTriads(w,bolster,bolster1,wedge1,wedge2,s,s1,s2,bolster3,bolster4):


#Define observers, points, triads, and neighbors
DefineObservers([world,W,w],[bolster,Bolster,bolster],[wedge1,Wedge1_1,wedge1],[wedge2,Wedge2_1,w
edge2],

[sideframe,S,s],[bolster3,Bolster3,bolster3],[bolster4,Bolster4,bolster4],[sideframe1,Sideframe3,
s1],
                [sideframe2,Sideframe4,s2]):
DefinePoints([S,Wedge1_1,s,q1,0,q2],
            [Wedge1_1,Wedge1_2,wedge1,-(6.125*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge1_2,Wedge1_3,wedge1,(6.125*tan(37.5*3.1415927/180)),0,0],
            [Wedge1_3,Wedge1_4,wedge1,0,0,6.125],
            [Wedge1_2,Wedge1_5,wedge1,0,-10.5,0],
            [Wedge1_3,Wedge1_6,wedge1,0,-10.5,0],
            [Wedge1_4,Wedge1_7,wedge1,0,-10.5,0],
            [Wedge1_1,Wedge1_cg,wedge1,(1/2)*(6.125*tan(37.5*3.1415927/180))-
(1/3)*(6.125*tan(37.5*3.1415927/180)),0,6.125*(1/3)],
            [Wedge1_1,Spring1_1,wedge1,0,0,0],
            [Wedge1_2,Wedge1_8,wedge1,0,-3.5,0],
            [Wedge1_8,Wedge1_9,wedge1,0,-3.5,0],
            [Wedge1_4,Wedge1_10,wedge1,0,-3.5,0],
            [Wedge1_10,Wedge1_11,wedge1,0,-3.5,0],
            [Wedge1_4,Wedge1_12,wedge1,-
((6.125/cos(37.5*3.1415927/180))/3)*sin(37.5*3.1415927/180),0,-
((6.125/cos(37.5*3.1415927/180))/3)*cos(37.5*3.1415927/180)],
            [Wedge1_12,Wedge1_13,wedge1,-
((6.125/cos(37.5*3.1415927/180))/3)*sin(37.5*3.1415927/180),0,-
((6.125/cos(37.5*3.1415927/180))/3)*cos(37.5*3.1415927/180)],
            [Wedge1_7,Wedge1_14,wedge1,-
((6.125/cos(37.5*3.1415927/180))/3)*sin(37.5*3.1415927/180),0,-
((6.125/cos(37.5*3.1415927/180))/3)*cos(37.5*3.1415927/180)],
            [Wedge1_14,Wedge1_15,wedge1,-
((6.125/cos(37.5*3.1415927/180))/3)*sin(37.5*3.1415927/180),0,-
((6.125/cos(37.5*3.1415927/180))/3)*cos(37.5*3.1415927/180)],
            [Wedge1_3,Wedge1_16,wedge1,0,0,2.0416666666666667],
            [Wedge1_16,Wedge1_17,wedge1,0,0,2.0416666666666667],
            [Wedge1_6,Wedge1_18,wedge1,0,0,2.0416666666666667],
            [Wedge1_18,Wedge1_19,wedge1,0,0,2.0416666666666667],
            [Wedge1_3,Wedge1_20,wedge1,0,-3.5,0],
```

```
            [Wedge1_20,Wedge1_21,wedge1,0,-3.5,0],
            [Wedge1_12,Wedge1_22,wedge1,0,-3.5,0],
            [Wedge1_22,Wedge1_23,wedge1,0,-3.5,0],
            [Wedge1_13,Wedge1_24,wedge1,0,-3.5,0],
            [Wedge1_24,Wedge1_25,wedge1,0,-3.5,0],
            [Wedge1_17,Wedge1_26,wedge1,0,-3.5,0],
            [Wedge1_26,Wedge1_27,wedge1,0,-3.5,0],
            [Wedge1_16,Wedge1_28,wedge1,0,-3.5,0],
            [Wedge1_28,Wedge1_29,wedge1,0,-3.5,0],

            [W,Bolster,w,0,0,q3],
            [Bolster,Bolster_cg,bolster,0,0,6.29],
            [Bolster,Spring2_1,bolster,0,0,0],

            [Bolster,Bolster1,bolster,2+3.827/2,0,(3.827/6.125)*(2+3.827/2)],
            [Bolster,Bolster2,bolster,-2-3.827/2,0,(3.827/6.125)*(2+3.827/2)],
            [Bolster,Bolster3,bolster,2,0,0],
            [Bolster,Bolster4,bolster,-2,0,0],
            [Bolster,Bolster5,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,0,6.125],
            [Bolster,Bolster6,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,0,6.125],
            [Bolster,Bolster7,bolster,2,6,0],
            [Bolster,Bolster8,bolster,2,-6,0],
            [Bolster,Bolster9,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,6,6.125],
            [Bolster,Bolster10,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-6,6.125],
            [Bolster,Bolster11,bolster,-2,6,0],
            [Bolster,Bolster12,bolster,-2,-6,0],
            [Bolster,Bolster13,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,6,6.125],
            [Bolster,Bolster14,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-6,6.125],

            [S,Wedge2_1,s,q4,0,q5],
            [Wedge2_1,Wedge2_2,wedge2,(6.125*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge2_2,Wedge2_3,wedge2,-(6.125*tan(37.5*3.1415927/180)),0,0],
            [Wedge2_3,Wedge2_4,wedge2,0,0,6.125],
            [Wedge2_2,Wedge2_5,wedge2,0,-10.5,0],
            [Wedge2_3,Wedge2_6,wedge2,0,-10.5,0],
            [Wedge2_4,Wedge2_7,wedge2,0,-10.5,0],
            [Wedge2_1,Wedge2_cg,wedge2,-
(1/2)*(6.125*tan(37.5*3.1415927/180))+(1/3)*(6.125*tan(37.5*3.1415927/180)),0,6.125*(1/3)],
            [Wedge2_1,Spring3_1,wedge2,0,0,0],

            [S,Spring1_2,s,(6.125*tan(37.5*3.1415927/180))/2+2,0,0],
            [S,Spring1_3,s,(6.125*tan(37.5*3.1415927/180))/2+2,0,10.25],
            [S,Spring2_2,s,0,0,0],
            [S,Spring2_3,s,0,0,10.25],
            [S,Spring3_2,s,-(6.125*tan(37.5*3.1415927/180))/2-2,0,0],
            [S,Spring3_3,s,-(6.125*tan(37.5*3.1415927/180))/2-2,0,10.25],
            [S,Sideframe1,s,2+(-15/2*tan(toe))+(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe2,s,-2-(-15/2*tan(toe))-(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe3,s,(2+(6.125*tan(37.5*3.1415927/180))),0,0],
            [S,Sideframe4,s,(-2-(6.125*tan(37.5*3.1415927/180))),0,0],
            [W,S,w,0,0,q11]):

DefineTriads([w,s,[0,1]],
             [s,s1,[-toe,2]],
             [s,s2,[toe,2]],
             [s,wedge1,[q6,3],[q7,2]],
             [s,bolster,[q8,3]],
             [w,wedge2,[q9,3],[q10,2]],
             [bolster,bolster3,[(37.5*3.1415927/180),2]],
             [bolster,bolster4,[(-37.5*3.1415927/180),2]]):
DefineNeighbors([world,sideframe],[world,bolster],[sideframe,wedge1],[sideframe,wedge2],[sidefram
e,sideframe1],[sideframe,sideframe2],
                [bolster,bolster3],[bolster,bolster4]):

#Declares the independent velocities as states
DeclareStates(q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10):

#Define the kinematic differential equations
kde:={q1t=u1,q2t=u2,q3t=u3,q4t=u4,q5t=u5,q6t=u6,q7t=u7,q8t=u8,q9t=u9,q10t=u10,q11t=(Fext*omega*co
s(omega*t))}:
kde:=simplify(solve(kde,{q1t,q2t,q3t,q4t,q5t,q6t,q7t,q8t,q9t,q10t,q11t})):
```

```
#Calculate the matrix of vectors which span the subspace of allowable motions, beta
v:=subs(kde,VelocityDescription([LinearVelocity(world,Wedge1_cg),AngularVelocity(w,wedge1),
                                 LinearVelocity(world,Bolster_cg),AngularVelocity(w,bolster),
                                 LinearVelocity(world,Wedge2_cg),AngularVelocity(w,wedge2)])):
beta:=CoeffExtract(v,[u1,u2,u3,u4,u5,u6,u7,u8,u9,u10]):

#Inertial forces
p1:=LinearMomentum(world,Wedge1_cg,0.875):
h1:=AngularMomentum(world,wedge1,matrix([[((1/12)*0.875*(10.5^2+6.125^2)),0,0],[0,((1/12)*0.875*(
3.827^2+6.125^2)),0],[0,0,((1/12)*0.875*(3.827^2+10.5^2))]])):
p2:=LinearMomentum(world,Bolster_cg,468):
h2:=AngularMomentum(world,bolster,matrix([[((1/12)*468*((12^2)+((6.125+5)^2))),0,0],[0,((1/12)*46
8*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+((6.125+5)^2))),0],[0,0,((1/12)*468*((4+2*(6.125*tan(3
7.5*3.1415927/180))^2)+(12^2)))]])):
p3:=LinearMomentum(world,Wedge2_cg,0.875):
h3:=AngularMomentum(world,wedge2,matrix([[((1/12)*0.875*(10.5^2+6.125^2)),0,0],[0,((1/12)*0.875*(
3.827^2+6.125^2)),0],[0,0,((1/12)*0.875*(3.827^2+10.5^2))]])):

#I1XX=((1/12)*0.875*(10.5^2+6.125^2));
#I1YY=((1/12)*0.875*(3.827^2+6.125^2));
#I1ZZ=((1/12)*0.875*(3.827^2+10.5^2));
#I2XX=((1/12)*468*((12^2)+((6.125+5)^2)));
#I2YY=((1/12)*468*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+((6.125+5)^2)));
#I2ZZ=((1/12)*468*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+(12^2)));
#I3XX=((1/12)*0.875*(10.5^2+6.125^2));
#I3YY=((1/12)*0.875*(3.827^2+6.125^2));
#I3ZZ=((1/12)*0.875*(3.827^2+10.5^2));

rwedge1:=FindTranslation(Spring1_1,Spring1_2):
rbolster:=FindTranslation(Spring2_1,Spring2_2):
rwedge2:=FindTranslation(Spring3_1,Spring3_2):

rBolster1:=FindTranslation(Bolster_cg,Bolster1):
rBolster2:=FindTranslation(Bolster_cg,Bolster2):
rBolster3:=FindTranslation(Bolster_cg,Bolster3):
rBolster4:=FindTranslation(Bolster_cg,Bolster4):
rBolster5:=FindTranslation(Bolster_cg,Bolster5):
rBolster6:=FindTranslation(Bolster_cg,Bolster6):

rWedge1_2:=FindTranslation(Wedge1_cg,Wedge1_2):
rWedge1_3:=FindTranslation(Wedge1_cg,Wedge1_3):
rWedge1_4:=FindTranslation(Wedge1_cg,Wedge1_4):
rWedge1_5:=FindTranslation(Wedge1_cg,Wedge1_5):
rWedge1_6:=FindTranslation(Wedge1_cg,Wedge1_6):
rWedge1_7:=FindTranslation(Wedge1_cg,Wedge1_7):
rWedge1_8:=FindTranslation(Wedge1_cg,Wedge1_8):
rWedge1_9:=FindTranslation(Wedge1_cg,Wedge1_9):
rWedge1_10:=FindTranslation(Wedge1_cg,Wedge1_10):
rWedge1_11:=FindTranslation(Wedge1_cg,Wedge1_11):
rWedge1_12:=FindTranslation(Wedge1_cg,Wedge1_12):
rWedge1_13:=FindTranslation(Wedge1_cg,Wedge1_13):
rWedge1_14:=FindTranslation(Wedge1_cg,Wedge1_14):
rWedge1_15:=FindTranslation(Wedge1_cg,Wedge1_15):
rWedge1_16:=FindTranslation(Wedge1_cg,Wedge1_16):
rWedge1_17:=FindTranslation(Wedge1_cg,Wedge1_17):
rWedge1_18:=FindTranslation(Wedge1_cg,Wedge1_18):
rWedge1_19:=FindTranslation(Wedge1_cg,Wedge1_19):
rWedge1_20:=FindTranslation(Wedge1_cg,Wedge1_20):
rWedge1_21:=FindTranslation(Wedge1_cg,Wedge1_21):
rWedge1_22:=FindTranslation(Wedge1_cg,Wedge1_22):
rWedge1_23:=FindTranslation(Wedge1_cg,Wedge1_23):
rWedge1_24:=FindTranslation(Wedge1_cg,Wedge1_24):
rWedge1_25:=FindTranslation(Wedge1_cg,Wedge1_25):
rWedge1_26:=FindTranslation(Wedge1_cg,Wedge1_26):
rWedge1_27:=FindTranslation(Wedge1_cg,Wedge1_27):
rWedge1_28:=FindTranslation(Wedge1_cg,Wedge1_28):
rWedge1_29:=FindTranslation(Wedge1_cg,Wedge1_29):

rWedge2_2:=FindTranslation(Wedge2_cg,Wedge2_2):
rWedge2_3:=FindTranslation(Wedge2_cg,Wedge2_3):
```

```
rWedge2_4:=FindTranslation(Wedge2_cg,Wedge2_4):
rWedge2_5:=FindTranslation(Wedge2_cg,Wedge2_5):
rWedge2_6:=FindTranslation(Wedge2_cg,Wedge2_6):
rWedge2_7:=FindTranslation(Wedge2_cg,Wedge2_7):

rW1:=(1/VectorLength(FindTranslation(Bolster7,Bolster10)))                                    &**
FindTranslation(Bolster7,Bolster10):
rW2:=(1/VectorLength(FindTranslation(Bolster8,Bolster9))) &** FindTranslation(Bolster8,Bolster9):
rW3:=(rW2 &xx rW1):

rW4:=(1/VectorLength(FindTranslation(Bolster11,Bolster14)))                                   &**
FindTranslation(Bolster11,Bolster14):
rW5:=(1/VectorLength(FindTranslation(Bolster12,Bolster13)))                                   &**
FindTranslation(Bolster12,Bolster13):
rW6:=(rW4 &xx rW5):

rW7:=(1/VectorLength(FindTranslation(Bolster3,Bolster5))) &** FindTranslation(Bolster3,Bolster5):
rW8:=(1/VectorLength(FindTranslation(Bolster11,Bolster13)))                                   &**
FindTranslation(Bolster11,Bolster13):

vWedge1_3:=subs(kde,LinearVelocity(sideframe,Wedge1_3))&oo MakeTranslations(s,3):
vWedge1_4:=subs(kde,LinearVelocity(sideframe,Wedge1_4))&oo MakeTranslations(s,3):
vWedge1_6:=subs(kde,LinearVelocity(sideframe,Wedge1_6))&oo MakeTranslations(s,3):
vWedge1_7:=subs(kde,LinearVelocity(sideframe,Wedge1_7))&oo MakeTranslations(s,3):
vWedge1_10:=subs(kde,LinearVelocity(sideframe,Wedge1_10))&oo MakeTranslations(s,3):
vWedge1_11:=subs(kde,LinearVelocity(sideframe,Wedge1_11))&oo MakeTranslations(s,3):
vWedge1_16:=subs(kde,LinearVelocity(sideframe,Wedge1_16))&oo MakeTranslations(s,3):
vWedge1_17:=subs(kde,LinearVelocity(sideframe,Wedge1_17))&oo MakeTranslations(s,3):
vWedge1_18:=subs(kde,LinearVelocity(sideframe,Wedge1_18))&oo MakeTranslations(s,3):
vWedge1_19:=subs(kde,LinearVelocity(sideframe,Wedge1_19))&oo MakeTranslations(s,3):
vWedge1_20:=subs(kde,LinearVelocity(sideframe,Wedge1_20))&oo MakeTranslations(s,3):
vWedge1_21:=subs(kde,LinearVelocity(sideframe,Wedge1_21))&oo MakeTranslations(s,3):
vWedge1_26:=subs(kde,LinearVelocity(sideframe,Wedge1_26))&oo MakeTranslations(s,3):
vWedge1_27:=subs(kde,LinearVelocity(sideframe,Wedge1_27))&oo MakeTranslations(s,3):
vWedge1_28:=subs(kde,LinearVelocity(sideframe,Wedge1_28))&oo MakeTranslations(s,3):
vWedge1_29:=subs(kde,LinearVelocity(sideframe,Wedge1_29))&oo MakeTranslations(s,3):

vWedge2_3:=subs(kde,LinearVelocity(sideframe,Wedge2_3))&oo MakeTranslations(s,3):
vWedge2_6:=subs(kde,LinearVelocity(sideframe,Wedge2_6))&oo MakeTranslations(s,3):
vWedge2_4:=subs(kde,LinearVelocity(sideframe,Wedge2_4))&oo MakeTranslations(s,3):
vWedge2_7:=subs(kde,LinearVelocity(sideframe,Wedge2_7))&oo MakeTranslations(s,3):

vBolsterWedge1_2:=(subs(kde,LinearVelocity(bolster,Wedge1_2))&oo rW7):
vBolsterWedge1_4:=(subs(kde,LinearVelocity(bolster,Wedge1_4))&oo rW7):
vBolsterWedge1_5:=(subs(kde,LinearVelocity(bolster,Wedge1_5))&oo rW7):
vBolsterWedge1_7:=(subs(kde,LinearVelocity(bolster,Wedge1_7))&oo rW7):
vBolsterWedge1_8:=(subs(kde,LinearVelocity(bolster,Wedge1_8))&oo rW7):
vBolsterWedge1_9:=(subs(kde,LinearVelocity(bolster,Wedge1_9))&oo rW7):
vBolsterWedge1_10:=(subs(kde,LinearVelocity(bolster,Wedge1_10))&oo rW7):
vBolsterWedge1_11:=(subs(kde,LinearVelocity(bolster,Wedge1_11))&oo rW7):
vBolsterWedge1_12:=(subs(kde,LinearVelocity(bolster,Wedge1_12))&oo rW7):
vBolsterWedge1_13:=(subs(kde,LinearVelocity(bolster,Wedge1_13))&oo rW7):
vBolsterWedge1_14:=(subs(kde,LinearVelocity(bolster,Wedge1_14))&oo rW7):
vBolsterWedge1_15:=(subs(kde,LinearVelocity(bolster,Wedge1_15))&oo rW7):
vBolsterWedge1_22:=(subs(kde,LinearVelocity(bolster,Wedge1_22))&oo rW7):
vBolsterWedge1_23:=(subs(kde,LinearVelocity(bolster,Wedge1_23))&oo rW7):
vBolsterWedge1_24:=(subs(kde,LinearVelocity(bolster,Wedge1_24))&oo rW7):
vBolsterWedge1_25:=(subs(kde,LinearVelocity(bolster,Wedge1_25))&oo rW7):

vBolsterWedge2_2:=(subs(kde,LinearVelocity(bolster,Wedge2_2))&oo rW8):
vBolsterWedge2_5:=(subs(kde,LinearVelocity(bolster,Wedge2_5))&oo rW8):
vBolsterWedge2_4:=(subs(kde,LinearVelocity(bolster,Wedge2_4))&oo rW8):
vBolsterWedge2_7:=(subs(kde,LinearVelocity(bolster,Wedge2_7))&oo rW8):

Spring11:=(FindTranslation(Spring1_2,Spring1_3) &oo MakeTranslations(s,3)):
Spring12:=(FindTranslation(Spring1_2,Spring1_1) &oo MakeTranslations(s,3)):
Spring21:=(FindTranslation(Spring2_2,Spring2_3) &oo MakeTranslations(s,3)):
Spring22:=(FindTranslation(Spring2_2,Spring2_1) &oo MakeTranslations(s,3)):
Spring31:=(FindTranslation(Spring3_2,Spring3_3) &oo MakeTranslations(s,3)):
Spring32:=(FindTranslation(Spring3_2,Spring3_1) &oo MakeTranslations(s,3)):
```

```
#External forces
f1:=MakeTranslations(w,0,0,0.875*(-32.2))
    &++ ((1491*(Spring11-Spring12)) &** MakeTranslations(s,3))
    &-- (Cdamp &** LinearVelocity(sideframe1,Wedge1_cg))

    &-- ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_8*(FindTranslation(Bolster3,Wedge1_8) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_9*(FindTranslation(Bolster3,Wedge1_9) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_10*(FindTranslation(Bolster5,Wedge1_10) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_11*(FindTranslation(Bolster5,Wedge1_11) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_12*(FindTranslation(Bolster3,Wedge1_12) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_13*(FindTranslation(Bolster3,Wedge1_13) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_14*(FindTranslation(Bolster5,Wedge1_14) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_15*(FindTranslation(Bolster5,Wedge1_15) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_22*(FindTranslation(Bolster3,Wedge1_22) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_23*(FindTranslation(Bolster3,Wedge1_23) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_24*(FindTranslation(Bolster5,Wedge1_24) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_25*(FindTranslation(Bolster5,Wedge1_25) &oo rW3)) &** rW3)

    &++
((kb*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)  &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)  &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)  &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)  &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_8*0.25*tanh(vBolsterWedge1_8/0.001)*(FindTranslation(Bolster3,Wedge1_8)  &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_9*0.25*tanh(vBolsterWedge1_9/0.001)*(FindTranslation(Bolster3,Wedge1_9)  &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_10*0.25*tanh(vBolsterWedge1_10/0.001)*(FindTranslation(Bolster5,Wedge1_10)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_11*0.25*tanh(vBolsterWedge1_11/0.001)*(FindTranslation(Bolster5,Wedge1_11)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_12*0.25*tanh(vBolsterWedge1_12/0.001)*(FindTranslation(Bolster5,Wedge1_12)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_13*0.25*tanh(vBolsterWedge1_13/0.001)*(FindTranslation(Bolster3,Wedge1_13)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_14*0.25*tanh(vBolsterWedge1_14/0.001)*(FindTranslation(Bolster5,Wedge1_14)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_15*0.25*tanh(vBolsterWedge1_15/0.001)*(FindTranslation(Bolster3,Wedge1_15)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_22*0.25*tanh(vBolsterWedge1_22/0.001)*(FindTranslation(Bolster3,Wedge1_22)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_23*0.25*tanh(vBolsterWedge1_23/0.001)*(FindTranslation(Bolster3,Wedge1_23)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_24*0.25*tanh(vBolsterWedge1_24/0.001)*(FindTranslation(Bolster5,Wedge1_24)
&oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_25*0.25*tanh(vBolsterWedge1_25/0.001)*(FindTranslation(Bolster5,Wedge1_25)
&oo rW3)) &** rW7)
```

```
    &--             ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--             ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--             ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--             ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_10*(FindTranslation(Sideframe1,Wedge1_10)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_11*(FindTranslation(Sideframe1,Wedge1_11)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_16*(FindTranslation(Sideframe1,Wedge1_16)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_17*(FindTranslation(Sideframe1,Wedge1_17)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_18*(FindTranslation(Sideframe1,Wedge1_18)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_19*(FindTranslation(Sideframe1,Wedge1_19)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_20*(FindTranslation(Sideframe1,Wedge1_20)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_21*(FindTranslation(Sideframe1,Wedge1_21)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_26*(FindTranslation(Sideframe1,Wedge1_26)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_27*(FindTranslation(Sideframe1,Wedge1_27)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_28*(FindTranslation(Sideframe1,Wedge1_28)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_29*(FindTranslation(Sideframe1,Wedge1_29)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))

    &-- ((ks*rCSideframe1Wedge1_3*0.4*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &-- ((ks*rCSideframe1Wedge1_4*0.4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &-- ((ks*rCSideframe1Wedge1_6*0.4*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &-- ((ks*rCSideframe1Wedge1_7*0.4*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_10*0.4*tanh(vWedge1_10/0.001)*(FindTranslation(Sideframe1,Wedge1_10)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_11*0.4*tanh(vWedge1_11/0.001)*(FindTranslation(Sideframe1,Wedge1_11)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_16*0.4*tanh(vWedge1_16/0.001)*(FindTranslation(Sideframe1,Wedge1_16)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_17*0.4*tanh(vWedge1_17/0.001)*(FindTranslation(Sideframe1,Wedge1_17)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_18*0.4*tanh(vWedge1_18/0.001)*(FindTranslation(Sideframe1,Wedge1_18)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_19*0.4*tanh(vWedge1_19/0.001)*(FindTranslation(Sideframe1,Wedge1_19)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_20*0.4*tanh(vWedge1_20/0.001)*(FindTranslation(Sideframe1,Wedge1_20)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_21*0.4*tanh(vWedge1_21/0.001)*(FindTranslation(Sideframe1,Wedge1_21)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_26*0.4*tanh(vWedge1_26/0.001)*(FindTranslation(Sideframe1,Wedge1_26)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_27*0.4*tanh(vWedge1_27/0.001)*(FindTranslation(Sideframe1,Wedge1_27)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
```

```
    &--
((ks*rCSideframe1Wedge1_28*0.4*tanh(vWedge1_28/0.001)*(FindTranslation(Sideframe1,Wedge1_28)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_29*0.4*tanh(vWedge1_29/0.001)*(FindTranslation(Sideframe1,Wedge1_29)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))

    &++ NullVector():

t1:=NullVector()
    &-- (Cdamp &** AngularVelocity(s1,wedge1))

    &--  (rWedge1_2  &xx  ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2)  &oo  rW3))  &**
rW3))
    &--  (rWedge1_5  &xx  ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5)  &oo  rW3))  &**
rW3))
    &--  (rWedge1_4  &xx  ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4)  &oo  rW3))  &**
rW3))
    &--  (rWedge1_7  &xx  ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7)  &oo  rW3))  &**
rW3))
    &--  (rWedge1_8  &xx  ((kb*rCBolsterWedge1_8*(FindTranslation(Bolster3,Wedge1_8)  &oo  rW3))  &**
rW3))
    &--  (rWedge1_9  &xx  ((kb*rCBolsterWedge1_9*(FindTranslation(Bolster3,Wedge1_9)  &oo  rW3))  &**
rW3))
    &--  (rWedge1_10  &xx  ((kb*rCBolsterWedge1_10*(FindTranslation(Bolster5,Wedge1_10)  &oo  rW3))
&** rW3))
    &--  (rWedge1_11  &xx  ((kb*rCBolsterWedge1_11*(FindTranslation(Bolster5,Wedge1_11)  &oo  rW3))
&** rW3))
    &--  (rWedge1_12  &xx  ((kb*rCBolsterWedge1_12*(FindTranslation(Bolster5,Wedge1_12)  &oo  rW3))
&** rW3))
    &--  (rWedge1_13  &xx  ((kb*rCBolsterWedge1_13*(FindTranslation(Bolster3,Wedge1_13)  &oo  rW3))
&** rW3))
    &--  (rWedge1_14  &xx  ((kb*rCBolsterWedge1_14*(FindTranslation(Bolster5,Wedge1_14)  &oo  rW3))
&** rW3))
    &--  (rWedge1_15  &xx  ((kb*rCBolsterWedge1_15*(FindTranslation(Bolster3,Wedge1_15)  &oo  rW3))
&** rW3))
    &--  (rWedge1_22  &xx  ((kb*rCBolsterWedge1_22*(FindTranslation(Bolster5,Wedge1_22)  &oo  rW3))
&** rW3))
    &--  (rWedge1_23  &xx  ((kb*rCBolsterWedge1_23*(FindTranslation(Bolster5,Wedge1_23)  &oo  rW3))
&** rW3))
    &--  (rWedge1_24  &xx  ((kb*rCBolsterWedge1_24*(FindTranslation(Bolster3,Wedge1_24)  &oo  rW3))
&** rW3))
    &--  (rWedge1_25  &xx  ((kb*rCBolsterWedge1_25*(FindTranslation(Bolster3,Wedge1_25)  &oo  rW3))
&** rW3))

    &++                                      (rWedge1_2                                       &xx
((kb*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)   &oo
rW3)) &** rW7))
    &++                                      (rWedge1_5                                       &xx
((kb*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)   &oo
rW3)) &** rW7))
    &++                                      (rWedge1_4                                       &xx
((kb*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)   &oo
rW3)) &** rW7))
    &++                                      (rWedge1_7                                       &xx
((kb*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)   &oo
rW3)) &** rW7))
    &++                                      (rWedge1_8                                       &xx
((kb*rCBolsterWedge1_8*0.25*tanh(vBolsterWedge1_8/0.001)*(FindTranslation(Bolster3,Wedge1_8)   &oo
rW3)) &** rW7))
    &++                                      (rWedge1_9                                       &xx
((kb*rCBolsterWedge1_9*0.25*tanh(vBolsterWedge1_9/0.001)*(FindTranslation(Bolster3,Wedge1_9)   &oo
rW3)) &** rW7))
    &++                                     (rWedge1_10                                       &xx
((kb*rCBolsterWedge1_10*0.25*tanh(vBolsterWedge1_10/0.001)*(FindTranslation(Bolster5,Wedge1_10)
&oo rW3)) &** rW7))
    &++                                     (rWedge1_11                                       &xx
((kb*rCBolsterWedge1_11*0.25*tanh(vBolsterWedge1_11/0.001)*(FindTranslation(Bolster5,Wedge1_11)
&oo rW3)) &** rW7))
```

```
    &++                                    (rWedge1_12                              &xx
((kb*rCBolsterWedge1_12*0.25*tanh(vBolsterWedge1_12/0.001)*(FindTranslation(Bolster5,Wedge1_12)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_13                              &xx
((kb*rCBolsterWedge1_13*0.25*tanh(vBolsterWedge1_13/0.001)*(FindTranslation(Bolster3,Wedge1_13)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_14                              &xx
((kb*rCBolsterWedge1_14*0.25*tanh(vBolsterWedge1_14/0.001)*(FindTranslation(Bolster5,Wedge1_14)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_15                              &xx
((kb*rCBolsterWedge1_15*0.25*tanh(vBolsterWedge1_15/0.001)*(FindTranslation(Bolster3,Wedge1_15)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_22                              &xx
((kb*rCBolsterWedge1_22*0.25*tanh(vBolsterWedge1_22/0.001)*(FindTranslation(Bolster5,Wedge1_22)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_23                              &xx
((kb*rCBolsterWedge1_23*0.25*tanh(vBolsterWedge1_23/0.001)*(FindTranslation(Bolster5,Wedge1_23)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_24                              &xx
((kb*rCBolsterWedge1_24*0.25*tanh(vBolsterWedge1_24/0.001)*(FindTranslation(Bolster3,Wedge1_24)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_25                              &xx
((kb*rCBolsterWedge1_25*0.25*tanh(vBolsterWedge1_25/0.001)*(FindTranslation(Bolster3,Wedge1_25)
&oo rW3)) &** rW7))

    &--  (rWedge1_3   &xx   ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_6   &xx   ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_4   &xx   ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_7   &xx   ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_10  &xx  ((ks*rCSideframe1Wedge1_10*(FindTranslation(Sideframe1,Wedge1_10)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_11  &xx  ((ks*rCSideframe1Wedge1_11*(FindTranslation(Sideframe1,Wedge1_11)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_16  &xx  ((ks*rCSideframe1Wedge1_16*(FindTranslation(Sideframe1,Wedge1_16)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_17  &xx  ((ks*rCSideframe1Wedge1_17*(FindTranslation(Sideframe1,Wedge1_17)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_18  &xx  ((ks*rCSideframe1Wedge1_18*(FindTranslation(Sideframe1,Wedge1_18)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_19  &xx  ((ks*rCSideframe1Wedge1_19*(FindTranslation(Sideframe1,Wedge1_19)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_20  &xx  ((ks*rCSideframe1Wedge1_20*(FindTranslation(Sideframe1,Wedge1_20)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_21  &xx  ((ks*rCSideframe1Wedge1_21*(FindTranslation(Sideframe1,Wedge1_21)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_26  &xx  ((ks*rCSideframe1Wedge1_26*(FindTranslation(Sideframe1,Wedge1_26)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_27  &xx  ((ks*rCSideframe1Wedge1_27*(FindTranslation(Sideframe1,Wedge1_27)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_28  &xx  ((ks*rCSideframe1Wedge1_28*(FindTranslation(Sideframe1,Wedge1_28)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--  (rWedge1_29  &xx  ((ks*rCSideframe1Wedge1_29*(FindTranslation(Sideframe1,Wedge1_29)  &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))

    &--                                    (rWedge1_3                               &xx
((ks*rCSideframe1Wedge1_3*0.4*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)      &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                    (rWedge1_6                               &xx
((ks*rCSideframe1Wedge1_6*0.4*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)      &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                    (rWedge1_4                               &xx
((ks*rCSideframe1Wedge1_4*0.4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)      &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                    (rWedge1_7                               &xx
((ks*rCSideframe1Wedge1_7*0.4*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)      &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
```

```
    &--                                      (rWedge1_10                                       &xx
((ks*rCSideframe1Wedge1_10*0.4*tanh(vWedge1_10/0.001)*(FindTranslation(Sideframe1,Wedge1_10)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_11                                       &xx
((ks*rCSideframe1Wedge1_11*0.4*tanh(vWedge1_11/0.001)*(FindTranslation(Sideframe1,Wedge1_11)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_16                                       &xx
((ks*rCSideframe1Wedge1_16*0.4*tanh(vWedge1_16/0.001)*(FindTranslation(Sideframe1,Wedge1_16)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_17                                       &xx
((ks*rCSideframe1Wedge1_17*0.4*tanh(vWedge1_17/0.001)*(FindTranslation(Sideframe1,Wedge1_17)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_18                                       &xx
((ks*rCSideframe1Wedge1_18*0.4*tanh(vWedge1_18/0.001)*(FindTranslation(Sideframe1,Wedge1_18)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_19                                       &xx
((ks*rCSideframe1Wedge1_19*0.4*tanh(vWedge1_19/0.001)*(FindTranslation(Sideframe1,Wedge1_19)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_20                                       &xx
((ks*rCSideframe1Wedge1_20*0.4*tanh(vWedge1_20/0.001)*(FindTranslation(Sideframe1,Wedge1_20)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_21                                       &xx
((ks*rCSideframe1Wedge1_21*0.4*tanh(vWedge1_21/0.001)*(FindTranslation(Sideframe1,Wedge1_21)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_26                                       &xx
((ks*rCSideframe1Wedge1_26*0.4*tanh(vWedge1_26/0.001)*(FindTranslation(Sideframe1,Wedge1_26)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_27                                       &xx
((ks*rCSideframe1Wedge1_27*0.4*tanh(vWedge1_27/0.001)*(FindTranslation(Sideframe1,Wedge1_27)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_28                                       &xx
((ks*rCSideframe1Wedge1_28*0.4*tanh(vWedge1_28/0.001)*(FindTranslation(Sideframe1,Wedge1_28)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                      (rWedge1_29                                       &xx
((ks*rCSideframe1Wedge1_29*0.4*tanh(vWedge1_29/0.001)*(FindTranslation(Sideframe1,Wedge1_29)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))

    &++ NullVector():

f2:=MakeTranslations(w,0,0,468*(-32.2))
    &++ ((22417.8*(Spring21-Spring22)) &** MakeTranslations(s,3))

    &++ ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_8*(FindTranslation(Bolster3,Wedge1_8) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_9*(FindTranslation(Bolster3,Wedge1_9) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_10*(FindTranslation(Bolster5,Wedge1_10) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_11*(FindTranslation(Bolster5,Wedge1_11) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_12*(FindTranslation(Bolster3,Wedge1_12) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_13*(FindTranslation(Bolster3,Wedge1_13) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_14*(FindTranslation(Bolster5,Wedge1_14) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_15*(FindTranslation(Bolster5,Wedge1_15) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_22*(FindTranslation(Bolster3,Wedge1_22) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_23*(FindTranslation(Bolster3,Wedge1_23) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_24*(FindTranslation(Bolster5,Wedge1_24) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_25*(FindTranslation(Bolster5,Wedge1_25) &oo rW3)) &** rW3)

    &--
((kb*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)   &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)   &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)   &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)   &oo
rW3)) &** rW7)
```

```
    &--
((kb*rCBolsterWedge1_8*0.25*tanh(vBolsterWedge1_8/0.001)*(FindTranslation(Bolster3,Wedge1_8)   &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_9*0.25*tanh(vBolsterWedge1_9/0.001)*(FindTranslation(Bolster3,Wedge1_9)   &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_10*0.25*tanh(vBolsterWedge1_10/0.001)*(FindTranslation(Bolster5,Wedge1_10)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_11*0.25*tanh(vBolsterWedge1_11/0.001)*(FindTranslation(Bolster5,Wedge1_11)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_12*0.25*tanh(vBolsterWedge1_12/0.001)*(FindTranslation(Bolster5,Wedge1_12)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_13*0.25*tanh(vBolsterWedge1_13/0.001)*(FindTranslation(Bolster3,Wedge1_13)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_14*0.25*tanh(vBolsterWedge1_14/0.001)*(FindTranslation(Bolster5,Wedge1_14)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_15*0.25*tanh(vBolsterWedge1_15/0.001)*(FindTranslation(Bolster3,Wedge1_15)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_22*0.25*tanh(vBolsterWedge1_22/0.001)*(FindTranslation(Bolster3,Wedge1_22)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_23*0.25*tanh(vBolsterWedge1_23/0.001)*(FindTranslation(Bolster3,Wedge1_23)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_24*0.25*tanh(vBolsterWedge1_24/0.001)*(FindTranslation(Bolster5,Wedge1_24)
&oo rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_25*0.25*tanh(vBolsterWedge1_25/0.001)*(FindTranslation(Bolster5,Wedge1_25)
&oo rW3)) &** rW7)

    #&++ ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6)
    #&++ ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6)
    #&++ ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6)
    #&++ ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6)

    #&--
((kb*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)   &oo
rW6)) &** rW8)
    #&--
((kb*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)   &oo
rW6)) &** rW8)
    #&--
((kb*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)   &oo
rW6)) &** rW8)
    #&--
((kb*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)   &oo
rW6)) &** rW8)

    &++ NullVector():

t2:=NullVector()

    &++  (rBolster3  &xx  ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2)  &oo  rW3))  &**
rW3))
    &++  (rBolster3  &xx  ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5)  &oo  rW3))  &**
rW3))
    &++  (rBolster5  &xx  ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4)  &oo  rW3))  &**
rW3))
    &++  (rBolster5  &xx  ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7)  &oo  rW3))  &**
rW3))
    &++  (rBolster3  &xx  ((kb*rCBolsterWedge1_8*(FindTranslation(Bolster3,Wedge1_8)  &oo  rW3))  &**
rW3))
    &++  (rBolster3  &xx  ((kb*rCBolsterWedge1_9*(FindTranslation(Bolster3,Wedge1_9)  &oo  rW3))  &**
rW3))
```

```
    &++ (rBolster5 &xx ((kb*rCBolsterWedge1_10*(FindTranslation(Bolster5,Wedge1_10) &oo rW3)) &**
rW3))
    &++ (rBolster5 &xx ((kb*rCBolsterWedge1_11*(FindTranslation(Bolster5,Wedge1_11) &oo rW3)) &**
rW3))
    &++ (rBolster5 &xx ((kb*rCBolsterWedge1_12*(FindTranslation(Bolster5,Wedge1_12) &oo rW3)) &**
rW3))
    &++ (rBolster3 &xx ((kb*rCBolsterWedge1_13*(FindTranslation(Bolster3,Wedge1_13) &oo rW3)) &**
rW3))
    &++ (rBolster3 &xx ((kb*rCBolsterWedge1_14*(FindTranslation(Bolster3,Wedge1_14) &oo rW3)) &**
rW3))
    &++ (rBolster5 &xx ((kb*rCBolsterWedge1_15*(FindTranslation(Bolster5,Wedge1_15) &oo rW3)) &**
rW3))
    &++ (rBolster3 &xx ((kb*rCBolsterWedge1_22*(FindTranslation(Bolster3,Wedge1_22) &oo rW3)) &**
rW3))
    &++ (rBolster3 &xx ((kb*rCBolsterWedge1_23*(FindTranslation(Bolster3,Wedge1_23) &oo rW3)) &**
rW3))
    &++ (rBolster5 &xx ((kb*rCBolsterWedge1_24*(FindTranslation(Bolster5,Wedge1_24) &oo rW3)) &**
rW3))
    &++ (rBolster5 &xx ((kb*rCBolsterWedge1_25*(FindTranslation(Bolster5,Wedge1_25) &oo rW3)) &**
rW3))

    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2) &oo
rW3)) &** rW7))
    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5) &oo
rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4) &oo
rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7) &oo
rW3)) &** rW7))
    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_8*0.25*tanh(vBolsterWedge1_8/0.001)*(FindTranslation(Bolster3,Wedge1_8) &oo
rW3)) &** rW7))
    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_9*0.25*tanh(vBolsterWedge1_9/0.001)*(FindTranslation(Bolster3,Wedge1_9) &oo
rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_10*0.25*tanh(vBolsterWedge1_10/0.001)*(FindTranslation(Bolster5,Wedge1_10)
&oo rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_11*0.25*tanh(vBolsterWedge1_11/0.001)*(FindTranslation(Bolster5,Wedge1_11)
&oo rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_12*0.25*tanh(vBolsterWedge1_12/0.001)*(FindTranslation(Bolster5,Wedge1_12)
&oo rW3)) &** rW7))
    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_13*0.25*tanh(vBolsterWedge1_13/0.001)*(FindTranslation(Bolster3,Wedge1_13)
&oo rW3)) &** rW7))
    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_14*0.25*tanh(vBolsterWedge1_14/0.001)*(FindTranslation(Bolster3,Wedge1_14)
&oo rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_15*0.25*tanh(vBolsterWedge1_15/0.001)*(FindTranslation(Bolster5,Wedge1_15)
&oo rW3)) &** rW7))
    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_22*0.25*tanh(vBolsterWedge1_22/0.001)*(FindTranslation(Bolster3,Wedge1_22)
&oo rW3)) &** rW7))
    &-- (rBolster3 &xx
((kb*rCBolsterWedge1_23*0.25*tanh(vBolsterWedge1_23/0.001)*(FindTranslation(Bolster3,Wedge1_23)
&oo rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_24*0.25*tanh(vBolsterWedge1_24/0.001)*(FindTranslation(Bolster5,Wedge1_24)
&oo rW3)) &** rW7))
    &-- (rBolster5 &xx
((kb*rCBolsterWedge1_25*0.25*tanh(vBolsterWedge1_25/0.001)*(FindTranslation(Bolster5,Wedge1_25)
&oo rW3)) &** rW7))
```

```
    #&++ (rBolster4 &xx ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &**
rW6))
    #&++ (rBolster4 &xx ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &**
rW6))
    #&++ (rBolster6 &xx ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &**
rW6))
    #&++ (rBolster6 &xx ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &**
rW6))

    #&--                                    (rBolster4                                        &xx
((kb*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)  &oo
rW6)) &** rW8))
    #&--                                    (rBolster4                                        &xx
((kb*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)  &oo
rW6)) &** rW8))
    #&--                                    (rBolster6                                        &xx
((kb*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)  &oo
rW6)) &** rW8))
    #&--                                    (rBolster6                                        &xx
((kb*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)  &oo
rW6)) &** rW8))

    &++ NullVector():


f3:=NullVector()
    #MakeTranslations(w,0,0,0.875*(-32.2))
    #&++ ((1491*(Spring31-Spring32)) &** MakeTranslations(s,3))
    #&-- (Cdamp &** LinearVelocity(sideframe2,Wedge2_cg))

    #&-- ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6)
    #&-- ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6)
    #&-- ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6)
    #&-- ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6)

    #&++
((kb*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)  &oo
rW6)) &** rW8)
    #&++
((kb*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)  &oo
rW6)) &** rW8)
    #&++
((kb*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)  &oo
rW6)) &** rW8)
    #&++
((kb*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)  &oo
rW6)) &** rW8)

    #&--          ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    #&--          ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    #&--          ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    #&--          ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))

    #&++
((ks*rCSideframe2Wedge2_3*0.4*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    #&++
((ks*rCSideframe2Wedge2_6*0.4*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    #&++
((ks*rCSideframe2Wedge2_4*0.4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    #&++
((ks*rCSideframe2Wedge2_7*0.4*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)):

    &++ NullVector():
```

```
t3:=NullVector()
    #&-- (Cdamp &** AngularVelocity(s2,wedge2))

    #&-- (rWedge2_2 &xx ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &**
rW6))
    #&-- (rWedge2_5 &xx ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &**
rW6))
    #&-- (rWedge2_4 &xx ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &**
rW6))
    #&-- (rWedge2_7 &xx ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &**
rW6))

    #&++                                    (rWedge2_2                                   &xx
((kb*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)  &oo
rW6)) &** rW8))
    #&++                                    (rWedge2_5                                   &xx
((kb*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)  &oo
rW6)) &** rW8))
    #&++                                    (rWedge2_4                                   &xx
((kb*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)  &oo
rW6)) &** rW8))
    #&++                                    (rWedge2_7                                   &xx
((kb*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)  &oo
rW6)) &** rW8))

    #&--  (rWedge2_3  &xx  ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    #&--  (rWedge2_6  &xx  ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    #&--  (rWedge2_4  &xx  ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    #&--  (rWedge2_7  &xx  ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))

    #&++                                    (rWedge2_3                                   &xx
((ks*rCSideframe2Wedge2_3*0.4*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    #&++                                    (rWedge2_6                                   &xx
((ks*rCSideframe2Wedge2_6*0.4*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    #&++                                    (rWedge2_4                                   &xx
((ks*rCSideframe2Wedge2_4*0.4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    #&++                                    (rWedge2_7                                   &xx
((ks*rCSideframe2Wedge2_7*0.4*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)     &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3))):

    &++ NullVector():

#Puts all the forces in a vector
p:=subs(kde,MomentumDescription([p1,h1,p2,h2,p3,h3])):
appliedforce:=ForceDescription([f1,t1,f2,t2,f3,t3]):

eqs:=((subs(kde,DiffTime(p,w)) &-- appliedforce) &oo beta):
diffeqs:=kde union convert(eqs,set):


#Create bodies
DefineObjects([bolster,'Block',point=MakeTranslations(bolster,0,0,(6.125+5)/2),xlength=4,ylength=
12,zlength=6.125+5,color=red],

[bolster,'Block',point=MakeTranslations(bolster,0,0,6.125+2.5),xlength=4+2*(6.125*tan(37.5*3.1415
927/180)),ylength=12,zlength=5,color=red],
              [bolster3,'Block',point=MakeTranslations(bolster3,-
2,0,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2))/2),xlength=4,ylength=12,zlength=sqrt((6.125
*tan(37.5*3.1415927/180))^2+6.125^2),color=red],

[bolster4,'Block',point=MakeTranslations(bolster4,2,0,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.1
25^2))/2),xlength=4,ylength=12,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2),color=red]
,
```

```
[wedge1,'Block',point=Wedge1_1,xlength=(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,
color=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlengt
h=0.05,ylength=10.5,zlength=6.125,color=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,0,0,6.125/2),orient=MakeRotations([37.5*3.1415927/1
80,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2),color=bl
ue],

              [wedge2,'Block',point=Wedge2_1,xlength=-
(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,-
(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlength=0.05,ylength=10.5,zlength=6.125,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,0,0,6.125/2),orient=MakeRotations([-
37.5*3.1415927/180,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6
.125^2),color=blue],


[sideframe,'Cylinder',point=MakeTranslations(s,(6.125*tan(37.5*3.1415927/180))/2+2,0,(VectorLengt
h(FindTranslation(Spring1_1,Spring1_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring
1_1,Spring1_2)),color=green],

[sideframe,'Cylinder',point=MakeTranslations(s,0,0,(VectorLength(FindTranslation(Spring2_1,Spring
2_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring2_1,Spring2_2)),color=green],
              [sideframe,'Cylinder',point=MakeTranslations(s,-(6.125*tan(37.5*3.1415927/180))/2-
2,0,(VectorLength(FindTranslation(Spring3_1,Spring3_2)))/2),radius=0.5,length=VectorLength(FindTr
anslation(Spring3_1,Spring3_2)),color=green],


[sideframe,'Block',point=S,xlength=4+2*(6.125*tan(37.5*3.1415927/180)),ylength=12,zlength=0.01,co
lor=white],

[sideframe1,'Block',point=MakeTranslations(s1,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],

[sideframe2,'Block',point=MakeTranslations(s2,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],

#Visual placement of points
              [world,'Sphere',point=W,radius=0.05,color=green],

              [sideframe,'Sphere',point=Sideframe1,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe2,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe3,radius=0.05,color=red],
              [sideframe,'Sphere',point=Sideframe4,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster_cg,radius=0.05,color=green],
              [bolster,'Sphere',point=Bolster1,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster2,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster3,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster4,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster5,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster6,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster7,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster8,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster9,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster10,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster11,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster12,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster13,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster14,radius=0.05,color=blue],

              [wedge1,'Sphere',point=Wedge1_1,radius=0.05,color=white],
              [wedge1,'Sphere',point=Wedge1_2,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_3,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_4,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_5,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_6,radius=0.05,color=green],
```

```
                [wedge1,'Sphere',point=Wedge1_7,radius=0.05,color=green],
                [wedge1,'Sphere',point=Wedge1_8,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_9,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_10,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_11,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_12,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_13,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_14,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_15,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_16,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_17,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_18,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_19,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_20,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_21,radius=0.05,color=yellow],
                [wedge1,'Sphere',point=Wedge1_22,radius=0.05,color=red],
                [wedge1,'Sphere',point=Wedge1_23,radius=0.05,color=white],
                [wedge1,'Sphere',point=Wedge1_24,radius=0.05,color=white],
                [wedge1,'Sphere',point=Wedge1_25,radius=0.05,color=red],
                [wedge1,'Sphere',point=Wedge1_26,radius=0.05,color=white],
                [wedge1,'Sphere',point=Wedge1_27,radius=0.05,color=red],
                [wedge1,'Sphere',point=Wedge1_28,radius=0.05,color=red],
                [wedge1,'Sphere',point=Wedge1_29,radius=0.05,color=white],
                [wedge1,'Sphere',point=Wedge1_cg,radius=0.05,color=cyan],

                [wedge2,'Sphere',point=Wedge2_1,radius=0.05,color=yellow],
                [wedge2,'Sphere',point=Wedge2_2,radius=0.05,color=red],
                [wedge2,'Sphere',point=Wedge2_3,radius=0.05,color=green],
                [wedge2,'Sphere',point=Wedge2_4,radius=0.05,color=blue],
                [wedge2,'Sphere',point=Wedge2_5,radius=0.05,color=red],
                [wedge2,'Sphere',point=Wedge2_6,radius=0.05,color=green],
                [wedge2,'Sphere',point=Wedge2_7,radius=0.05,color=blue],
                [wedge2,'Sphere',point=Wedge2_cg,radius=0.05,color=green],

                [wedge1,'Sphere',point=Spring1_1,radius=0.05,color=white],
                [wedge1,'Sphere',point=Spring1_2,radius=0.05,color=yellow],
                [bolster,'Sphere',point=Spring2_1,radius=0.05,color=yellow],
                [bolster,'Sphere',point=Spring2_2,radius=0.05,color=yellow],
                [wedge2,'Sphere',point=Spring3_1,radius=0.05,color=yellow],
                [wedge2,'Sphere',point=Spring3_2,radius=0.05,color=yellow]):

#Outputs
GeometryOutput(main=world,
         parameters=[toe=0,omega=0,Fext=0,ks=62500,kb=62500,Cdamp=1],
        states=[q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10],
         filename="halftruckvardamp_ver3_4.geo"):
MotionOutput(ode=diffeqs,
         parameters=[toe=0,omega=0,Fext=0,ks=62500,kb=62500,Cdamp=1],
         states=[q1=4.350302,q2=9.668228,q3=9.666987,q4=-
4.350302,q5=9.668228,q6,q7,q8,q9,q10,q11,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10],
         insignals=[rCBolsterWedge1_2=((FindTranslation(Wedge1_2,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_4=((FindTranslation(Wedge1_4,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_5=((FindTranslation(Wedge1_5,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_7=((FindTranslation(Wedge1_7,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_8=((FindTranslation(Wedge1_8,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_9=((FindTranslation(Wedge1_9,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_10=((FindTranslation(Wedge1_10,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_11=((FindTranslation(Wedge1_11,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_12=((FindTranslation(Wedge1_12,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_13=((FindTranslation(Wedge1_13,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_14=((FindTranslation(Wedge1_14,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_15=((FindTranslation(Wedge1_15,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_22=((FindTranslation(Wedge1_22,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_23=((FindTranslation(Wedge1_23,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_24=((FindTranslation(Wedge1_24,Bolster3) &oo rW3)&>0),
                    rCBolsterWedge1_25=((FindTranslation(Wedge1_25,Bolster3) &oo rW3)&>0),
                    rCSideframe1Wedge1_3=((FindTranslation(Sideframe1,Wedge1_3)                &oo
MakeTranslations(s1,1))&>0),
                    rCSideframe1Wedge1_4=((FindTranslation(Sideframe1,Wedge1_4)                &oo
MakeTranslations(s1,1))&>0),
```

```
                   rCSideframe1Wedge1_6=((FindTranslation(Sideframe1,Wedge1_6)                 &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_7=((FindTranslation(Sideframe1,Wedge1_7)                 &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_10=((FindTranslation(Sideframe1,Wedge1_10)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_11=((FindTranslation(Sideframe1,Wedge1_11)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_16=((FindTranslation(Sideframe1,Wedge1_16)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_17=((FindTranslation(Sideframe1,Wedge1_17)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_18=((FindTranslation(Sideframe1,Wedge1_18)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_19=((FindTranslation(Sideframe1,Wedge1_19)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_20=((FindTranslation(Sideframe1,Wedge1_20)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_21=((FindTranslation(Sideframe1,Wedge1_21)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_26=((FindTranslation(Sideframe1,Wedge1_26)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_27=((FindTranslation(Sideframe1,Wedge1_27)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_28=((FindTranslation(Sideframe1,Wedge1_28)               &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_29=((FindTranslation(Sideframe1,Wedge1_29)               &oo
MakeTranslations(s1,1))&>0),
                   rCBolsterWedge2_2=((FindTranslation(Wedge2_2,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_4=((FindTranslation(Wedge2_4,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_5=((FindTranslation(Wedge2_5,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_7=((FindTranslation(Wedge2_7,Bolster4) &oo rW6)&>0),
                   rCSideframe2Wedge2_3=((FindTranslation(Sideframe2,Wedge2_3)                 &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_4=((FindTranslation(Sideframe2,Wedge2_4)                 &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_6=((FindTranslation(Sideframe2,Wedge2_6)                 &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_7=((FindTranslation(Sideframe2,Wedge2_7)                 &oo
MakeTranslations(s2,1))&<0)],
        filename="halftruckvardamp_ver3_4.dyn"):
```

7.1.1.2. Half-truck Variably Damped State Calculation M-file

```
global Fext muS muB ks kb omega l0 l02 k Pi r z1 z2 z3 theta m1 m2 m3 g Cdamp toe

%Parameters
Fext=1;
muS=0.4;
muB=0.25;
ks=1000000;
kb=1000000;
omega=3;
l0=10.25;
l02=10.25;
k=1491;
Pi=3.1415927;
r=0.05;
z1=3.827;
z2=6.125;
z3=10.5;
theta=37.5;
m1=0.875;
m2=468;
m3=0.875;
g=-32.2;
Cdamp=10;
toe=0;

q1=y(:,1);
q2=y(:,2);
q3=y(:,3);
q4=y(:,4);
q5=y(:,5);
q6=y(:,6);
q7=y(:,7);
q8=y(:,8);
q9=y(:,9);
q10=y(:,10);
q11=y(:,11);
u1=y(:,12);
u2=y(:,13);
u3=y(:,14);
u4=y(:,15);
u5=y(:,16);
u6=y(:,17);
u7=y(:,18);
u8=y(:,19);
u9=y(:,20);
u10=y(:,21);


for i = 1:length(t),

if        ((1.4439761050275938832888634726593-.72198805251379694164443173632964*q1(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q2(i)+1.696627844398436
5613546947974428*cos(q6(i))*cos(q7(i))*cos(q8(i))+1.6966278443984365613546947974428*sin(q6(i))*co
s(q7(i))*sin(q8(i))+1.3018683596265573072916282130853*sin(q7(i))+3.790437275697433943633266615730
6*sin(q6(i))*cos(q8(i))-3.7904372756974339436332666157306*cos(q6(i))*sin(q8(i)))>0),
        rCBolsterWedge1_2(i)=1;
    else
        rCBolsterWedge1_2(i)=0;
    end

if                                          ((-.72198805251379694164443173632964*q1(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q2(i)-
1.6966278443984365613546947974428*cos(q6(i))*cos(q7(i))*cos(q8(i))-
1.6966278443984365613546947974428*sin(q6(i))*cos(q7(i))*sin(q8(i))-
1.3018683596265573072916282130853*sin(q7(i))+3.7904372756974339436332666157306*sin(q6(i))*cos(q8(
```

```
i))-3.7904372756974339436332666157306*cos(q6(i))*sin(q8(i))-
4.4221768216470062675721443850190*cos(q6(i))*sin(q7(i))*cos(q8(i))-
4.4221768216470062675721443850190*sin(q6(i))*sin(q7(i))*sin(q8(i))+3.393255688796873122709389594 8
857*cos(q7(i))+1.4439761050275938832888634726593)>0);
        rCBolsterWedge1_4(i)=1;
    else
        rCBolsterWedge1_4(i)=0;
    end

if        ((1.4439761050275938832888634726593-.72198805251379694164443173632964*q1(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q2(i)+1.696627844398436
5613546947974428*cos(q6(i))*cos(q7(i))*cos(q8(i))+1.6966278443984365613546947974428*sin(q6(i))*co
s(q7(i))*sin(q8(i))+1.3018683596265573072916282130853*sin(q7(i))-
3.7904372756974339436332666157306*sin(q6(i))*cos(q8(i))+3.7904372756974339436332666157306*cos(q6(
i))*sin(q8(i)))>0),
        rCBolsterWedge1_5(i)=1;
    else
        rCBolsterWedge1_5(i)=0;
    end

if                                ((-.72198805251379694164443173632964*q1(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q2(i)-
1.6966278443984365613546947974428*cos(q6(i))*cos(q7(i))*cos(q8(i))-
1.6966278443984365613546947974428*sin(q6(i))*cos(q7(i))*sin(q8(i))-
1.3018683596265573072916282130853*sin(q7(i))-
3.7904372756974339436332666157306*sin(q6(i))*cos(q8(i))+3.7904372756974339436332666157306*cos(q6(
i))*sin(q8(i))-4.4221768216470062675721443850190*cos(q6(i))*sin(q7(i))*cos(q8(i))-
4.4221768216470062675721443850190*sin(q6(i))*sin(q7(i))*sin(q8(i))+3.393255688796873122709389594 8
857*cos(q7(i))+1.4439761050275938832888634726593)>0),
        rCBolsterWedge1_7(i)=1;
    else
        rCBolsterWedge1_7(i)=0;
    end

if                    (((-6.6998778954614753568235161050744+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q2(i))*sin(toe)+2.3499389477307376784117580525372*cos(q6(i))*cos(q7(i))*cos(toe)-
2.3499389477307376784117580525372*sin(q7(i))*sin(toe)-
5.2500000000000000000000000000000*sin(q6(i))*cos(toe))>0),
        rCSideframe1Wedge1_3(i)=1;
    else
        rCSideframe1Wedge1_3(i)=0;
    end

if                    (((-6.6998778954614753568235161050744+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q2(i))*sin(toe)+2.3499389477307376784117580525372*cos(q6(i))*cos(q7(i))*cos(toe)-
2.3499389477307376784117580525372*sin(q7(i))*sin(toe)-
5.2500000000000000000000000000000*sin(q6(i))*cos(toe)+6.125*cos(q6(i))*sin(q7(i))*cos(toe)+6.125*
cos(q7(i))*sin(toe))>0),
        rCSideframe1Wedge1_4(i)=1;
    else
        rCSideframe1Wedge1_4(i)=0;
    end

if                    (((-6.6998778954614753568235161050744+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q2(i))*sin(toe)+2.3499389477307376784117580525372*cos(q6(i))*cos(q7(i))*cos(toe)-
2.3499389477307376784117580525372*sin(q7(i))*sin(toe)+5.2500000000000000000000000000000*sin(q6(i)
)*cos(toe))>0),
        rCSideframe1Wedge1_6(i)=1;
    else
        rCSideframe1Wedge1_6(i)=0;
    end

if                    (((-6.6998778954614753568235161050744+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q2(i))*sin(toe)+2.3499389477307376784117580525372*cos(q6(i))*cos(q7(i))*cos(toe)-
2.3499389477307376784117580525372*sin(q7(i))*sin(toe)+5.2500000000000000000000000000000*sin(q6(i)
)*cos(toe)+6.125*cos(q6(i))*sin(q7(i))*cos(toe)+6.125*cos(q7(i))*sin(toe))>0),
        rCSideframe1Wedge1_7(i)=1;
    else
        rCSideframe1Wedge1_7(i)=0;
    end
```

```
if ((1.4439761050275938832888634726593+.72198805251379694164443173632964*q4(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q5(i)+1.696627844398436
5613546947974428*cos(q9(i))*cos(q10(i))*cos(q8(i))+1.6966278443984365613546947974428*sin(q9(i))*c
os(q10(i))*sin(q8(i))-1.3018683596265573072916282130853*sin(q10(i))-
3.7904372756974339436332666157306*sin(q9(i))*cos(q8(i))+3.7904372756974339436332666157306*cos(q9(
i))*sin(q8(i)))>0),
        rCBolsterWedge2_2(i)=1;
    else
        rCBolsterWedge2_2(i)=0;
    end

if ((.72198805251379694164443173632964*q4(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q5(i)-
1.6966278443984365613546947974428*cos(q9(i))*cos(q10(i))*cos(q8(i))-
1.6966278443984365613546947974428*sin(q9(i))*cos(q10(i))*sin(q8(i))+1.301868359626557307291628213
0853*sin(q10(i))-
3.7904372756974339436332666157306*sin(q9(i))*cos(q8(i))+3.7904372756974339436332666157306*cos(q9(
i))*sin(q8(i))+4.4221768216470062675721443850190*cos(q9(i))*sin(q10(i))*cos(q8(i))+4.422176821647
0062675721443850190*sin(q9(i))*sin(q10(i))*sin(q8(i))+3.3932556887968731227093895948857*cos(q10(i
))+1.4439761050275938832888634726593)>0),
        rCBolsterWedge2_4(i)=1;
    else
        rCBolsterWedge2_4(i)=0;
    end

if ((1.4439761050275938832888634726593+.72198805251379694164443173632964*q4(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q5(i)+1.696627844398436
5613546947974428*cos(q9(i))*cos(q10(i))*cos(q8(i))+1.6966278443984365613546947974428*sin(q9(i))*c
os(q10(i))*sin(q8(i))-
1.3018683596265573072916282130853*sin(q10(i))+3.7904372756974339436332666157306*sin(q9(i))*cos(q8
(i))-3.7904372756974339436332666157306*cos(q9(i))*sin(q8(i)))>0),
        rCBolsterWedge2_5(i)=1;
    else
        rCBolsterWedge2_5(i)=0;
    end

if ((.72198805251379694164443173632964*q4(i)*cos(q8(i))-
.55400092878316295880969626038950*q3(i)+.55400092878316295880969626038950*q5(i)-
1.6966278443984365613546947974428*cos(q9(i))*cos(q10(i))*cos(q8(i))-
1.6966278443984365613546947974428*sin(q9(i))*cos(q10(i))*sin(q8(i))+1.301868359626557307291628213
0853*sin(q10(i))+3.7904372756974339436332666157306*sin(q9(i))*cos(q8(i))-
3.7904372756974339436332666157306*cos(q9(i))*sin(q8(i))+4.4221768216470062675721443850190*cos(q9(
i))*sin(q10(i))*cos(q8(i))+4.4221768216470062675721443850190*sin(q9(i))*sin(q10(i))*sin(q8(i))+3.
3932556887968731227093895948857*cos(q10(i))+1.4439761050275938832888634726593)>0),
        rCBolsterWedge2_7(i)=1;
    else
        rCBolsterWedge2_7(i)=0;
    end

if (((6.6998778954614753568235161050744-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.3499389477307376784117580525372*cos(q9(i))*cos(q10(i))*cos(toe)-
2.3499389477307376784117580525372*sin(q10(i))*sin(toe)-
5.2500000000000000000000000000000*sin(q9(i))*cos(toe))<0),
        rCSideframe2Wedge2_3(i)=1;
    else
        rCSideframe2Wedge2_3(i)=0;
    end

if (((6.6998778954614753568235161050744-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.3499389477307376784117580525372*cos(q9(i))*cos(q10(i))*cos(toe)-
2.3499389477307376784117580525372*sin(q10(i))*sin(toe)-
5.2500000000000000000000000000000*sin(q9(i))*cos(toe)+6.125*cos(q9(i))*sin(q10(i))*cos(toe)-
6.125*cos(q10(i))*sin(toe))<0),
        rCSideframe2Wedge2_4(i)=1;
    else
        rCSideframe2Wedge2_4(i)=0;
    end

if (((6.6998778954614753568235161050744-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.3499389477307376784117580525372*cos(q9(i))*cos(q10(i))*cos(toe)-
```

```
2.3499389477307376784117580525372*sin(q10(i))*sin(toe)+5.2500000000000000000000000000000*sin(q9(i
))*cos(toe))<0),
        rCSideframe2Wedge2_6(i)=1;
    else
        rCSideframe2Wedge2_6(i)=0;
    end

if      (((6.6998778954614753568235161050744-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.3499389477307376784117580525372*cos(q9(i))*cos(q10(i))*cos(toe)-
2.3499389477307376784117580525372*sin(q10(i))*sin(toe)+5.2500000000000000000000000000000*sin(q9(i
))*cos(toe)+6.125*cos(q9(i))*sin(q10(i))*cos(toe)-6.125*cos(q10(i))*sin(toe))<0),
        rCSideframe2Wedge2_7(i)=1;
    else
        rCSideframe2Wedge2_7(i)=0;
end
```

7.1.2. Half-truck Constantly Damped Model

7.1.2.1. Half-truck Constantly Damped SRC file

```
with(Mambo):

#Declare observers, points, and triads
DeclareObservers(world,bolster,bolster1,bolster3,bolster4,wedge1,wedge2,sideframe,sideframe1,side
frame2):
DeclarePoints(W,Bolster,Bolster1,Bolster2,Bolster_cg,Bolster3,Bolster4,Bolster5,Bolster6,Bolster7
,
              Bolster8,Bolster9,Bolster10,Bolster11,Bolster12,Bolster13,Bolster14,
              Wedge1_1,Wedge1_2,Wedge1_3,Wedge1_4,Wedge1_5,Wedge1_6,Wedge1_7,Wedge1_cg,
              Wedge2_1,Wedge2_2,Wedge2_3,Wedge2_4,Wedge2_5,Wedge2_6,Wedge2_7,Wedge2_cg,

Spring1_1,Spring1_2,Spring1_3,Spring2_1,Spring2_2,Spring2_3,Spring3_1,Spring3_2,Spring3_3,S,Sidef
rame1,Sideframe2,Sideframe3,Sideframe4):
DeclareTriads(w,bolster,wedge1,wedge2,s,s1,s2,bolster3,bolster4):


#Define observers, points, triads, and neighbors
DefineObservers([world,W,w],[bolster,Bolster,bolster],[wedge1,Wedge1_1,wedge1],[wedge2,Wedge2_1,w
edge2],

[sideframe,S,s],[bolster3,Bolster3,bolster3],[bolster4,Bolster4,bolster4],[sideframe1,Sideframe3,
s1],
                [sideframe2,Sideframe4,s2]):
DefinePoints([Bolster,Wedge1_1,s,q1,0,q2],
            [Wedge1_1,Wedge1_2,wedge1,-(4*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge1_2,Wedge1_3,wedge1,(4*tan(37.5*3.1415927/180)),0,0],
            [Wedge1_3,Wedge1_4,wedge1,0,0,4],
            [Wedge1_2,Wedge1_5,wedge1,0,-10.5,0],
            [Wedge1_3,Wedge1_6,wedge1,0,-10.5,0],
            [Wedge1_4,Wedge1_7,wedge1,0,-10.5,0],
            [Wedge1_1,Wedge1_cg,wedge1,(1/2)*(4*tan(37.5*3.1415927/180))-
(1/3)*(4*tan(37.5*3.1415927/180)),0,4*(1/3)],
            [Wedge1_1,Spring1_1,wedge1,0,0,0],

            [W,Bolster,s,0,0,q3],
            [Bolster,Bolster_cg,bolster,0,0,6.29],
            [Bolster,Spring2_1,bolster,0,0,0],

            [Bolster,Bolster1,bolster,(2.5+2*(4*tan(37.5*3.1415927/180)))/2,0,3],
            [Bolster,Bolster2,bolster,-(2.5+2*(4*tan(37.5*3.1415927/180)))/2,0,3],
            [Bolster,Bolster3,bolster,2,0,0],
            [Bolster,Bolster4,bolster,-2,0,0],
            [Bolster,Bolster5,bolster,(6+2*(4*tan(37.5*3.1415927/180)))/2,0,5.25],
            [Bolster,Bolster6,bolster,-(6+2*(4*tan(37.5*3.1415927/180)))/2,0,5.25],
            [Bolster,Bolster7,bolster,2,6,0],
            [Bolster,Bolster8,bolster,2,-6,0],
            [Bolster,Bolster9,bolster,(6+2*(4*tan(37.5*3.1415927/180)))/2,6,5.25],
            [Bolster,Bolster10,bolster,(6+2*(4*tan(37.5*3.1415927/180)))/2,-6,5.25],
            [Bolster,Bolster11,bolster,-2,6,0],
            [Bolster,Bolster12,bolster,-2,-6,0],
            [Bolster,Bolster13,bolster,-(6+2*(4*tan(37.5*3.1415927/180)))/2,6,5.25],
            [Bolster,Bolster14,bolster,-(6+2*(4*tan(37.5*3.1415927/180)))/2,-6,5.25],

            [Bolster,Wedge2_1,s,q4,0,q5],
            [Wedge2_1,Wedge2_2,wedge2,(4*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge2_2,Wedge2_3,wedge2,-(4*tan(37.5*3.1415927/180)),0,0],
            [Wedge2_3,Wedge2_4,wedge2,0,0,4],
            [Wedge2_2,Wedge2_5,wedge2,0,-10.5,0],
            [Wedge2_3,Wedge2_6,wedge2,0,-10.5,0],
            [Wedge2_4,Wedge2_7,wedge2,0,-10.5,0],
            [Wedge2_1,Wedge2_cg,wedge2,-
(1/2)*(4*tan(37.5*3.1415927/180))+(1/3)*(4*tan(37.5*3.1415927/180)),0,4*(1/3)],
```

```
            [Wedge2_1,Spring3_1,wedge2,0,0,0],

            [Bolster,Spring1_2,bolster,(4*tan(37.5*3.1415927/180))/2+2.7,0,0],
            [Bolster,Spring1_3,bolster,(4*tan(37.5*3.1415927/180))/2+2.7,0,3.07],
            [S,Spring2_2,s,0,0,0],
            [S,Spring2_3,s,0,0,10.25],
            [Bolster,Spring3_2,bolster,-(4*tan(37.5*3.1415927/180))/2-2.7,0,0],
            [Bolster,Spring3_3,bolster,-(4*tan(37.5*3.1415927/180))/2-2.7,0,3.07],
            [S,Sideframe1,s,1.35+(-15/2*tan(toe))+(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe2,s,-1.35-(-15/2*tan(toe))-(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe3,s,(2.9+3*(4*tan(37.5*3.1415927/180)))/2,0,0],
            [S,Sideframe4,s,(-2.9-3*(4*tan(37.5*3.1415927/180)))/2,0,0],
            [W,S,w,0,0,q6]):

DefineTriads([w,s,[0,1]],
             [s,s1,[-toe,2]],
             [s,s2,[toe,2]],
             [w,wedge1,[q7,3],[q8,2]],
             [w,bolster,[q9,3]],
             [w,wedge2,[q10,3],[q11,2]],
             [bolster,bolster3,[(37.5*3.1415927/180),2]],
             [bolster,bolster4,[(-37.5*3.1415927/180),2]]):
DefineNeighbors([world,sideframe],[sideframe,bolster],[sideframe,wedge1],[sideframe,sideframe1],[
sideframe,sideframe2],[sideframe,wedge2],[bolster,bolster3],
                [bolster,bolster4]):

#Declares the independent velocities as states
DeclareStates(q1,q2,q3,q4,q5,q6t,q7,q8,q9,q10,q11,u1,u2,u3,u4,u5,u7,u8,u9,u10,u11):

#Define the kinematic differential equations
kde:={q1t=u1,q2t=u2,q3t=u3,q4t=u4,q5t=u5,q6t=Fext*omega*cos(omega*t),q7t=u7,q8t=u8,q9t=u9,q10t=u1
0,q11t=u11}:
kde:=simplify(solve(kde,{q1t,q2t,q3t,q4t,q5t,q6t,q7t,q8t,q9t,q10t,q11t})):

#Calculate the matrix of vectors which span the subspace of allowable motions, beta
v:=subs(kde,VelocityDescription([LinearVelocity(world,Wedge1_cg),AngularVelocity(w,wedge1),

LinearVelocity(world,Bolster_cg),AngularVelocity(w,bolster),

LinearVelocity(world,Wedge2_cg),AngularVelocity(w,wedge2)])):
beta:=CoeffExtract(v,[u1,u2,u3,u4,u5,u7,u8,u9,u10,u11]):

#Inertial forces
p1:=LinearMomentum(world,Wedge1_cg,0.875):
h1:=AngularMomentum(world,wedge1,matrix([[((1/12)*1*(10.5^2+4^2)),0,0],[0,((1/12)*1*(3.827^2+4^2)
),0],[0,0,((1/12)*1*(3.827^2+10.5^2))]])):
p2:=LinearMomentum(world,Bolster_cg,468):
h2:=AngularMomentum(world,bolster,matrix([[((1/12)*468*((12^2)+((4+5)^2))),0,0],[0,((1/12)*468*((
4+2*(4*tan(37.5*3.1415927/180))^2)+((4+5)^2))),0],[0,0,((1/12)*468*((4+2*(4*tan(37.5*3.1415927/18
0))^2)+(12^2)))]])):
p3:=LinearMomentum(world,Wedge2_cg,0.875):
h3:=AngularMomentum(world,wedge2,matrix([[((1/12)*1*(10.5^2+4^2)),0,0],[0,((1/12)*1*(3.827^2+4^2)
),0],[0,0,((1/12)*1*(3.827^2+10.5^2))]])):

#I1XX=((1/12)*1*(10.5^2+4^2))
#I1YY=((1/12)*1*(3.827^2+4^2))
#I1ZZ=((1/12)*1*(3.827^2+10.5^2))
#I2XX=((1/12)*468*((12^2)+((4+5)^2))),
#I2YY=((1/12)*468*((4+2*(4*tan(37.5*3.1415927/180))^2)+((4+5)^2))),
#I2ZZ=((1/12)*468*((4+2*(4*tan(37.5*3.1415927/180))^2)+(12^2))),

rwedge1:=FindTranslation(Spring1_1,Spring1_2):
rbolster:=FindTranslation(Spring2_1,Spring2_2):
rwedge2:=FindTranslation(Spring3_1,Spring3_2):

rBolster1:=FindTranslation(Bolster_cg,Bolster1):
rBolster2:=FindTranslation(Bolster_cg,Bolster2):
rBolster3:=FindTranslation(Bolster_cg,Bolster3):
rBolster4:=FindTranslation(Bolster_cg,Bolster4):
rBolster5:=FindTranslation(Bolster_cg,Bolster5):
rBolster6:=FindTranslation(Bolster_cg,Bolster6):
```

```
rWedge1_2:=FindTranslation(Wedge1_cg,Wedge1_2):
rWedge1_3:=FindTranslation(Wedge1_cg,Wedge1_3):
rWedge1_4:=FindTranslation(Wedge1_cg,Wedge1_4):
rWedge1_5:=FindTranslation(Wedge1_cg,Wedge1_5):
rWedge1_6:=FindTranslation(Wedge1_cg,Wedge1_6):
rWedge1_7:=FindTranslation(Wedge1_cg,Wedge1_7):

rWedge2_2:=FindTranslation(Wedge2_cg,Wedge2_2):
rWedge2_3:=FindTranslation(Wedge2_cg,Wedge2_3):
rWedge2_4:=FindTranslation(Wedge2_cg,Wedge2_4):
rWedge2_5:=FindTranslation(Wedge2_cg,Wedge2_5):
rWedge2_6:=FindTranslation(Wedge2_cg,Wedge2_6):
rWedge2_7:=FindTranslation(Wedge2_cg,Wedge2_7):

rW1:=(1/VectorLength(FindTranslation(Bolster7,Bolster10)))                                    &**
FindTranslation(Bolster7,Bolster10):
rW2:=(1/VectorLength(FindTranslation(Bolster8,Bolster9))) &** FindTranslation(Bolster8,Bolster9):
rW3:=(rW2 &xx rW1):

rW4:=(1/VectorLength(FindTranslation(Bolster11,Bolster14)))                                   &**
FindTranslation(Bolster11,Bolster14):
rW5:=(1/VectorLength(FindTranslation(Bolster12,Bolster13)))                                   &**
FindTranslation(Bolster12,Bolster13):
rW6:=(rW4 &xx rW5):

rW7:=(1/VectorLength(FindTranslation(Bolster3,Bolster5))) &** FindTranslation(Bolster3,Bolster5):
rW8:=(1/VectorLength(FindTranslation(Bolster11,Bolster13)))                                   &**
FindTranslation(Bolster11,Bolster13):

vWedge1_3:=subs(kde,LinearVelocity(sideframe,Wedge1_3))&oo MakeTranslations(s,3):
vWedge1_6:=subs(kde,LinearVelocity(sideframe,Wedge1_6))&oo MakeTranslations(s,3):
vWedge1_4:=subs(kde,LinearVelocity(sideframe,Wedge1_4))&oo MakeTranslations(s,3):
vWedge1_7:=subs(kde,LinearVelocity(sideframe,Wedge1_7))&oo MakeTranslations(s,3):

vWedge2_3:=subs(kde,LinearVelocity(sideframe,Wedge2_3))&oo MakeTranslations(s,3):
vWedge2_6:=subs(kde,LinearVelocity(sideframe,Wedge2_6))&oo MakeTranslations(s,3):
vWedge2_4:=subs(kde,LinearVelocity(sideframe,Wedge2_4))&oo MakeTranslations(s,3):
vWedge2_7:=subs(kde,LinearVelocity(sideframe,Wedge2_7))&oo MakeTranslations(s,3):

vBolsterWedge1_2:=(subs(kde,LinearVelocity(bolster,Wedge1_2))&oo rW7):
vBolsterWedge1_5:=(subs(kde,LinearVelocity(bolster,Wedge1_5))&oo rW7):
vBolsterWedge1_4:=(subs(kde,LinearVelocity(bolster,Wedge1_4))&oo rW7):
vBolsterWedge1_7:=(subs(kde,LinearVelocity(bolster,Wedge1_7))&oo rW7):

vBolsterWedge2_2:=(subs(kde,LinearVelocity(bolster,Wedge2_2))&oo rW8):
vBolsterWedge2_5:=(subs(kde,LinearVelocity(bolster,Wedge2_5))&oo rW8):
vBolsterWedge2_4:=(subs(kde,LinearVelocity(bolster,Wedge2_4))&oo rW8):
vBolsterWedge2_7:=(subs(kde,LinearVelocity(bolster,Wedge2_7))&oo rW8):

Spring11:=(FindTranslation(Spring1_2,Spring1_3) &oo MakeTranslations(bolster,3)):
Spring12:=(FindTranslation(Spring1_2,Spring1_1) &oo MakeTranslations(bolster,3)):
Spring21:=(FindTranslation(Spring2_2,Spring2_3) &oo MakeTranslations(s,3)):
Spring22:=(FindTranslation(Spring2_2,Spring2_1) &oo MakeTranslations(s,3)):
Spring31:=(FindTranslation(Spring3_2,Spring3_3) &oo MakeTranslations(bolster,3)):
Spring32:=(FindTranslation(Spring3_2,Spring3_1) &oo MakeTranslations(bolster,3)):

#External forces
f1:=NullVector()
    &++ MakeTranslations(w,0,0,0.875*(-32.2))
    &++ ((k1*(Spring11-Spring12)) &** MakeTranslations(bolster,3))
    &-- (Cdamp &** LinearVelocity(bolster,Wedge1_cg))

    &-- ((1000000*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &** rW3)
    &-- ((1000000*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &** rW3)
    &-- ((1000000*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &** rW3)
    &-- ((1000000*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &** rW3)

    &++
((1000000*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)
&oo rW3)) &** rW7)
```

```
    &++
((1000000*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)
&oo rW3)) &** rW7)
    &++
((1000000*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)
&oo rW3)) &** rW7)
    &++
((1000000*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)
&oo rW3)) &** rW7)

    &--          ((1000000*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)          &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--          ((1000000*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)          &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--          ((1000000*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)          &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--          ((1000000*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)          &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))

    &--
((1000000*rCSideframe1Wedge1_3*0.4*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((1000000*rCSideframe1Wedge1_6*0.4*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((1000000*rCSideframe1Wedge1_4*0.4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((1000000*rCSideframe1Wedge1_7*0.4*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))

    &++ NullVector():

t1:=NullVector()
    &-- (Cdamp &** AngularVelocity(s,wedge1))

    &-- (rWedge1_2 &xx ((1000000*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3))
&** rW3))
    &-- (rWedge1_5 &xx ((1000000*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3))
&** rW3))
    &-- (rWedge1_4 &xx ((1000000*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3))
&** rW3))
    &-- (rWedge1_7 &xx ((1000000*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3))
&** rW3))

    &++                                    (rWedge1_2                                          &xx
((1000000*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_5                                          &xx
((1000000*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_4                                          &xx
((1000000*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)
&oo rW3)) &** rW7))
    &++                                    (rWedge1_7                                          &xx
((1000000*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)
&oo rW3)) &** rW7))

    &-- (rWedge1_3 &xx ((1000000*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3) &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &-- (rWedge1_6 &xx ((1000000*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6) &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &-- (rWedge1_4 &xx ((1000000*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4) &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &-- (rWedge1_7 &xx ((1000000*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7) &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))

    &--                                    (rWedge1_3                                          &xx
((1000000*rCSideframe1Wedge1_3*0.4*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
```

```
    &--                                 (rWedge1_6                              &xx
((1000000*rCSideframe1Wedge1_6*0.4*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                 (rWedge1_4                              &xx
((1000000*rCSideframe1Wedge1_4*0.4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                 (rWedge1_7                              &xx
((1000000*rCSideframe1Wedge1_7*0.4*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))

    &++ NullVector():


f2:=NullVector()
    &++ MakeTranslations(w,0,0,468*(-32.2))
    &++ ((k2*(Spring21-Spring22)) &** MakeTranslations(s,3))
    &-- (Cdamp &** LinearVelocity(sideframe,Bolster_cg))

    &++ ((1000000*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo (rW3))) &** (rW3))
    &++ ((1000000*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo (rW3))) &** (rW3))
    &++ ((1000000*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo (rW3))) &** (rW3))
    &++ ((1000000*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo (rW3))) &** (rW3))

    &--
((1000000*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)
&oo rW3)) &** rW7)
    &--
((1000000*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)
&oo rW3)) &** rW7)
    &--
((1000000*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)
&oo rW3)) &** rW7)
    &--
((1000000*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)
&oo rW3)) &** rW7)

    &++ ((1000000*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6)
    &++ ((1000000*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6)
    &++ ((1000000*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6)
    &++ ((1000000*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6)

    &--
((1000000*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)
&oo rW6)) &** rW8)
    &--
((1000000*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)
&oo rW6)) &** rW8)
    &--
((1000000*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)
&oo rW6)) &** rW8)
    &--
((1000000*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)
&oo rW6)) &** rW8)

    &++ NullVector():

t2:=NullVector()
    &-- (Cdamp &** AngularVelocity(s,bolster))

    &++ (rBolster3 &xx ((1000000*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3))
&** rW3))
    &++ (rBolster3 &xx ((1000000*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3))
&** rW3))
    &++ (rBolster5 &xx ((1000000*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3))
&** rW3))
    &++ (rBolster5 &xx ((1000000*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3))
&** rW3))

    &--                                 (rBolster3                              &xx
((1000000*rCBolsterWedge1_2*0.25*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)
&oo rW3)) &** rW7))
```

```
    &--                                    (rBolster3                                   &xx
((1000000*rCBolsterWedge1_5*0.25*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)
&oo rW3)) &** rW7))
    &--                                    (rBolster5                                   &xx
((1000000*rCBolsterWedge1_4*0.25*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)
&oo rW3)) &** rW7))
    &--                                    (rBolster5                                   &xx
((1000000*rCBolsterWedge1_7*0.25*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)
&oo rW3)) &** rW7))

    &++ (rBolster4 &xx ((1000000*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6))
&** rW6))
    &++ (rBolster4 &xx ((1000000*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6))
&** rW6))
    &++ (rBolster6 &xx ((1000000*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6))
&** rW6))
    &++ (rBolster6 &xx ((1000000*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6))
&** rW6))

    &--                                    (rBolster4                                   &xx
((1000000*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)
&oo rW6)) &** rW8))
    &--                                    (rBolster4                                   &xx
((1000000*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)
&oo rW6)) &** rW8))
    &--                                    (rBolster6                                   &xx
((1000000*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)
&oo rW6)) &** rW8))
    &--                                    (rBolster6                                   &xx
((1000000*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)
&oo rW6)) &** rW8))

    &++ NullVector():


f3:=NullVector()
    &++ MakeTranslations(w,0,0,0.875*(-32.2))
    &++ ((k1*(Spring31-Spring32)) &** MakeTranslations(bolster,3))
    &-- (Cdamp &** LinearVelocity(bolster,Wedge2_cg))

    &-- ((1000000*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6)
    &-- ((1000000*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6)
    &-- ((1000000*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6)
    &-- ((1000000*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6)

    &++
((1000000*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)
&oo rW6)) &** rW8)
    &++
((1000000*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)
&oo rW6)) &** rW8)
    &++
((1000000*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)
&oo rW6)) &** rW8)
    &++
((1000000*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)
&oo rW6)) &** rW8)

    &--         ((1000000*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)         &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--         ((1000000*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)         &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--         ((1000000*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)         &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--         ((1000000*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)         &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))

    &++
((1000000*rCSideframe2Wedge2_3*0.4*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
```

```
    &++
((1000000*rCSideframe2Wedge2_6*0.4*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++
((1000000*rCSideframe2Wedge2_4*0.4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++
((1000000*rCSideframe2Wedge2_7*0.4*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))

    &++ NullVector():

t3:=NullVector()
    &-- (Cdamp &** AngularVelocity(s,wedge2))

    &-- (rWedge2_2 &xx ((1000000*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6))
&** rW6))
    &-- (rWedge2_5 &xx ((1000000*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6))
&** rW6))
    &-- (rWedge2_4 &xx ((1000000*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6))
&** rW6))
    &-- (rWedge2_7 &xx ((1000000*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6))
&** rW6))

    &++                                      (rWedge2_2                                     &xx
((1000000*rCBolsterWedge2_2*0.25*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)
&oo rW6)) &** rW8))
    &++                                      (rWedge2_5                                     &xx
((1000000*rCBolsterWedge2_5*0.25*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)
&oo rW6)) &** rW8))
    &++                                      (rWedge2_4                                     &xx
((1000000*rCBolsterWedge2_4*0.25*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)
&oo rW6)) &** rW8))
    &++                                      (rWedge2_7                                     &xx
((1000000*rCBolsterWedge2_7*0.25*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)
&oo rW6)) &** rW8))

    &-- (rWedge2_3 &xx ((1000000*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3) &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    &-- (rWedge2_6 &xx ((1000000*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6) &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    &-- (rWedge2_4 &xx ((1000000*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4) &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    &-- (rWedge2_7 &xx ((1000000*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7) &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))

    &++                                      (rWedge2_3                                     &xx
((1000000*rCSideframe2Wedge2_3*0.4*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    &++                                      (rWedge2_6                                     &xx
((1000000*rCSideframe2Wedge2_6*0.4*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    &++                                      (rWedge2_4                                     &xx
((1000000*rCSideframe2Wedge2_4*0.4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    &++                                      (rWedge2_7                                     &xx
((1000000*rCSideframe2Wedge2_7*0.4*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))

    &++ NullVector():


#Puts all the forces in a vector
p:=subs(kde,MomentumDescription([p1,h1,p2,h2,p3,h3])):
appliedforce:=ForceDescription([f1,t1,f2,t2,f3,t3]):

eqs:=((subs(kde,DiffTime(p,w)) &-- appliedforce) &oo beta):
diffeqs:=(kde union convert(eqs,set)):


#Create bodies
```

```
DefineObjects([bolster,'Block',point=MakeTranslations(bolster,0,0,(4+5)/2),xlength=4,ylength=12,z
length=4+5,color=red],

[bolster,'Block',point=MakeTranslations(bolster,0,0,5.25+2.5),xlength=2.9+3*(4*tan(37.5*3.1415927
/180)),ylength=12,zlength=5,color=red],
              [bolster3,'Block',point=MakeTranslations(bolster3,-
2,0,0.85+(sqrt((4*tan(37.5*3.1415927/180))^2+4^2))/2),xlength=4,ylength=12,zlength=1.5+(sqrt((4*t
an(37.5*3.1415927/180))^2+4^2)),color=red],

[bolster4,'Block',point=MakeTranslations(bolster4,2,0,0.85+(sqrt((4*tan(37.5*3.1415927/180))^2+4^
2))/2),xlength=4,ylength=12,zlength=1.5+(sqrt((4*tan(37.5*3.1415927/180))^2+4^2)),color=red],

[bolster,'Block',point=MakeTranslations(bolster,0,0,0),xlength=2.9+3*(4*tan(37.5*3.1415927/180)),
ylength=12,zlength=0.05,color=red],


[wedge1,'Block',point=Wedge1_1,xlength=(4*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,colo
r=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,(4*tan(37.5*3.1415927/180))/2,0,4/2),xlength=0.05,y
length=10.5,zlength=4,color=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,0,0,4/2),orient=MakeRotations([37.5*3.1415927/180,2
]),xlength=0.05,ylength=10.5,zlength=sqrt((4*tan(37.5*3.1415927/180))^2+4^2),color=blue],

              [wedge2,'Block',point=Wedge2_1,xlength=-
(4*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,-
(4*tan(37.5*3.1415927/180))/2,0,4/2),xlength=0.05,ylength=10.5,zlength=4,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,0,0,4/2),orient=MakeRotations([-
37.5*3.1415927/180,2]),xlength=0.05,ylength=10.5,zlength=sqrt((4*tan(37.5*3.1415927/180))^2+4^2),
color=blue],


[bolster,'Cylinder',point=MakeTranslations(bolster,(4*tan(37.5*3.1415927/180))/2+2.7,0,(VectorLen
gth(FindTranslation(Spring1_1,Spring1_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spri
ng1_1,Spring1_2)),color=green],

[sideframe,'Cylinder',point=MakeTranslations(s,0,0,(VectorLength(FindTranslation(Spring2_1,Spring
2_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring2_1,Spring2_2)),color=green],
              [bolster,'Cylinder',point=MakeTranslations(bolster,-(4*tan(37.5*3.1415927/180))/2-
2.7,0,(VectorLength(FindTranslation(Spring3_1,Spring3_2)))/2),radius=0.5,length=VectorLength(Find
Translation(Spring3_1,Spring3_2)),color=green],


[sideframe,'Block',point=S,xlength=2.9+3*(4*tan(37.5*3.1415927/180)),ylength=12,zlength=0.01,colo
r=white],

[sideframe1,'Block',point=MakeTranslations(s1,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],

[sideframe2,'Block',point=MakeTranslations(s2,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],
#orient=MakeRotations([toe,2])
#Visual placement of points
              [world,'Sphere',point=W,radius=0.05,color=green],

              [sideframe,'Sphere',point=Sideframe1,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe2,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe3,radius=0.05,color=red],
              [sideframe,'Sphere',point=Sideframe4,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster_cg,radius=0.05,color=green],
              [bolster,'Sphere',point=Bolster1,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster2,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster3,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster4,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster5,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster6,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster7,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster8,radius=0.05,color=blue],
```

```
              [bolster,'Sphere',point=Bolster9,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster10,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster11,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster12,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster13,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster14,radius=0.05,color=blue],

              [wedge1,'Sphere',point=Wedge1_1,radius=0.05,color=yellow],
              [wedge1,'Sphere',point=Wedge1_2,radius=0.05,color=red],
              [wedge1,'Sphere',point=Wedge1_3,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_4,radius=0.05,color=blue],
              [wedge1,'Sphere',point=Wedge1_5,radius=0.05,color=red],
              [wedge1,'Sphere',point=Wedge1_6,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_7,radius=0.05,color=blue],
              [wedge1,'Sphere',point=Wedge1_cg,radius=0.05,color=green],

              [wedge2,'Sphere',point=Wedge2_1,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Wedge2_2,radius=0.05,color=red],
              [wedge2,'Sphere',point=Wedge2_3,radius=0.05,color=green],
              [wedge2,'Sphere',point=Wedge2_4,radius=0.05,color=blue],
              [wedge2,'Sphere',point=Wedge2_5,radius=0.05,color=red],
              [wedge2,'Sphere',point=Wedge2_6,radius=0.05,color=green],
              [wedge2,'Sphere',point=Wedge2_7,radius=0.05,color=blue],
              [wedge2,'Sphere',point=Wedge2_cg,radius=0.05,color=green],

              [wedge1,'Sphere',point=Spring1_1,radius=0.05,color=yellow],
              [wedge1,'Sphere',point=Spring1_2,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Spring2_1,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Spring2_2,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Spring3_1,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Spring3_2,radius=0.05,color=yellow]):

#Outputs
GeometryOutput(main=world,
        parameters=[omega=0,omega2=0,toe=0,k1=1979,k2=22418.8,Cdamp=10,Fext=1,Fext2=1],
        states=[q1,q2,q3,q4,q5,q6t,q7,q8,q9,q10,q10,u1,u2,u3,u4,u5,u7,u8,u9,u10,u11],
        filename="halftruckconstdamp_ver3_1.geo"):
MotionOutput(ode=diffeqs,
        parameters=[omega=0,omega2=0,toe=0,k1=1979,k2=22418.8,Cdamp=10,Fext=1,Fext2=1],
        states=[q1=4.4,q2=1.15,q3=10.25,q4=-
4.4,q5=1.15,q6,q7,q8,q9,q10,q11,u1,u2,u3,u4,u5,u7,u8,u9,u10,u11],
        insignals=[rCBolsterWedge1_2=((FindTranslation(Wedge1_2,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_4=((FindTranslation(Wedge1_4,Bolster5) &oo rW3)&>0),
                   rCBolsterWedge1_5=((FindTranslation(Wedge1_5,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_7=((FindTranslation(Wedge1_7,Bolster5) &oo rW3)&>0),
                   rCSideframe1Wedge1_3=((FindTranslation(Sideframe1,Wedge1_3)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_4=((FindTranslation(Sideframe1,Wedge1_4)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_6=((FindTranslation(Sideframe1,Wedge1_6)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_7=((FindTranslation(Sideframe1,Wedge1_7)                &oo
MakeTranslations(s1,1))&>0),
                   rCBolsterWedge2_2=((FindTranslation(Wedge2_2,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_4=((FindTranslation(Wedge2_4,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_5=((FindTranslation(Wedge2_5,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_7=((FindTranslation(Wedge2_7,Bolster4) &oo rW6)&>0),
                   rCSideframe2Wedge2_3=((FindTranslation(Sideframe2,Wedge2_3)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_4=((FindTranslation(Sideframe2,Wedge2_4)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_6=((FindTranslation(Sideframe2,Wedge2_6)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_7=((FindTranslation(Sideframe2,Wedge2_7)                &oo
MakeTranslations(s2,1))&<0)],
        filename="halftruckconstdamp_ver3_1.dyn"):
```

7.1.2.2. *Half-truck Constantly Damped State Calculation M-file*

```
%Parameters
k1=1979;
k2=22418.8;
Fext=3;
omega=3;
Fext2=0;
omega2=0;
Cdamp=10;
toe=0.003;

q1=y(:,1);
q2=y(:,2);
q3=y(:,3);
q4=y(:,4);
q5=y(:,5);
q6=y(:,6);
q7=y(:,7);
q8=y(:,8);
q9=y(:,9);
q10=y(:,10);
q11=y(:,11);
u1=y(:,12);
u2=y(:,13);
u3=y(:,14);
u4=y(:,15);
u5=y(:,16);
u7=y(:,17);
u8=y(:,18);
u9=y(:,19);
u10=y(:,20);
u11=y(:,21);

for i = 1:length(t),
if ((1.339557900-.6697789500*q1(i)*cos(q9(i))+.5191498758*q2(i)+1.027878949*cos(q7(i))*cos(q8(i))*cos(q9(i))+1.027878949*sin(q7(i))*cos(q8(i))*sin(q9(i))+.7967154366*sin(q8(i))+3.516339488*sin(q7(i))*cos(q9(i))-3.516339488*cos(q7(i))*sin(q9(i)))>0),
        rCBolsterWedge1_2(i) = 1;
    else
        rCBolsterWedge1_2(i) = 0;
    end

if ((1.339557900-.6697789500*q1(i)*cos(q9(i))+.5191498758*q2(i)-1.027878949*cos(q7(i))*cos(q8(i))*cos(q9(i))-1.027878949*sin(q7(i))*cos(q8(i))*sin(q9(i))-.7967154366*sin(q8(i))+3.516339488*sin(q7(i))*cos(q9(i))-3.516339488*cos(q7(i))*sin(q9(i))-2.679115800*cos(q7(i))*sin(q8(i))*cos(q9(i))-2.679115800*sin(q7(i))*sin(q8(i))*sin(q9(i))+2.076599503*cos(q8(i)))>0),
        rCBolsterWedge1_4(i) = 1;
    else
        rCBolsterWedge1_4(i) = 0;
    end

if ((1.339557900-.6697789500*q1(i)*cos(q9(i))+.5191498758*q2(i)+1.027878949*cos(q7(i))*cos(q8(i))*cos(q9(i))+1.027878949*sin(q7(i))*cos(q8(i))*sin(q9(i))+.7967154366*sin(q8(i))-3.516339488*sin(q7(i))*cos(q9(i))+3.516339488*cos(q7(i))*sin(q9(i)))>0),
        rCBolsterWedge1_5(i) = 1;
    else
        rCBolsterWedge1_5(i) = 0;
    end

if ((1.339557900-.6697789500*q1(i)*cos(q9(i))+.5191498758*q2(i)-1.027878949*cos(q7(i))*cos(q8(i))*cos(q9(i))-1.027878949*sin(q7(i))*cos(q8(i))*sin(q9(i))-.7967154366*sin(q8(i))-3.516339488*sin(q7(i))*cos(q9(i))+3.516339488*cos(q7(i))*sin(q9(i))-
```

```
2.679115800*cos(q7(i))*sin(q8(i))*cos(q9(i))-
2.679115800*sin(q7(i))*sin(q8(i))*sin(q9(i))+2.076599503*cos(q8(i)))>0),
        rCBolsterWedge1_7(i) = 1;
    else
        rCBolsterWedge1_7(i) = 0;
    end

if                      (((-6.049877893+15/2*tan(toe)+q1(i))*cos(toe)+(-15/2+q3(i)+q2(i))*sin(toe)-
q6(i)*sin(toe)+1.534654006*cos(q7(i))*cos(q8(i))*cos(toe)-1.534654006*sin(q8(i))*sin(toe)-
5.250000000*sin(q7(i))*cos(toe))>0),
        rCSideframe1Wedge1_3(i) = 1;
    else
        rCSideframe1Wedge1_3(i) = 0;
    end

if                      (((-6.049877893+15/2*tan(toe)+q1(i))*cos(toe)+(-15/2+q3(i)+q2(i))*sin(toe)-
q6(i)*sin(toe)+1.534654006*cos(q7(i))*cos(q8(i))*cos(toe)-1.534654006*sin(q8(i))*sin(toe)-
5.250000000*sin(q7(i))*cos(toe)+4*cos(q7(i))*sin(q8(i))*cos(toe)+4*cos(q8(i))*sin(toe))>0),
        rCSideframe1Wedge1_4(i) = 1;
    else
        rCSideframe1Wedge1_4(i) = 0;
    end

if                      (((-6.049877893+15/2*tan(toe)+q1(i))*cos(toe)+(-15/2+q3(i)+q2(i))*sin(toe)-
q6(i)*sin(toe)+1.534654006*cos(q7(i))*cos(q8(i))*cos(toe)-
1.534654006*sin(q8(i))*sin(toe)+5.250000000*sin(q7(i))*cos(toe))>0),
        rCSideframe1Wedge1_6(i) = 1;
    else
        rCSideframe1Wedge1_6(i) = 0;
    end

if                      (((-6.049877893+15/2*tan(toe)+q1(i))*cos(toe)+(-15/2+q3(i)+q2(i))*sin(toe)-
q6(i)*sin(toe)+1.534654006*cos(q7(i))*cos(q8(i))*cos(toe)-
1.534654006*sin(q8(i))*sin(toe)+5.250000000*sin(q7(i))*cos(toe)+4*cos(q7(i))*sin(q8(i))*cos(toe)+
4*cos(q8(i))*sin(toe))>0),
        rCSideframe1Wedge1_7(i) = 1;
    else
        rCSideframe1Wedge1_7(i) = 0;
end
```

7.1.3. Half-truck Variably Damped Warp Model

7.1.3.1. Half-truck Variably Damped Warp SRC file

```
with(Mambo):

#Declare observers, points, and triads
DeclareObservers(world,bolster3,bolster4,wedge1,wedge2,sideframe,sideframe1,sideframe2,bolsterpsi
):
DeclarePoints(W,Bolster1,Bolster2,Bolster_cg,Bolster3,Bolster4,Bolster5,Bolster6,Bolster7,Bolster
8,Bolster9,Bolster10,Bolsterpsi,

Bolster11,Bolster12,Bolster13,Bolster14,Wedge1_1,Wedge1_2,Wedge1_3,Wedge1_4,Wedge1_5,Wedge1_6,Wed
ge1_7,Wedge1_cg,

Wedge1_8,Wedge1_9,Wedge1_10,Wedge1_11,Wedge1_12,Wedge1_13,Wedge1_14,Wedge1_15,Wedge1_16,Wedge1_17
,Wedge1_18,Wedge1_19,Wedge1_20,Wedge1_21,

Wedge2_1,Wedge2_2,Wedge2_3,Wedge2_4,Wedge2_5,Wedge2_6,Wedge2_7,Wedge2_cg,Rad1,Rad2,Spring2_L,Spri
ng2_L2,

Spring1_1,Spring1_2,Spring1_3,Spring2_1,Spring2_2,Spring2_3,Spring3_1,Spring3_2,Spring3_3,S,Sidef
rame1,Sideframe2,Sideframe3,Sideframe4):
DeclareTriads(w,bolster1,wedge1,wedge2,s,s1,s2,bolster3,bolster4,bolsterpsi):


#Define observers, points, triads, and neighbors
DefineObservers([world,W,w],[bolsterpsi,Bolsterpsi,bolsterpsi],[wedge1,Wedge1_1,wedge1],[wedge2,W
edge2_1,wedge2],

[sideframe,S,s],[bolster3,Bolster3,bolster3],[bolster4,Bolster4,bolster4],[sideframe1,Sideframe3,
s1],
                [sideframe2,Sideframe4,s2]):
DefinePoints([S,Wedge1_1,s,q1,q2,q3],
            [Wedge1_1,Wedge1_2,wedge1,-(6.125*tan(37.5*Pi/180))/2,10.5/2,0],
            [Wedge1_2,Wedge1_3,wedge1,(6.125*tan(37.5*Pi/180)),0,0],
            [Wedge1_3,Wedge1_4,wedge1,0,0,6.125],
            [Wedge1_2,Wedge1_5,wedge1,0,-10.5,0],
            [Wedge1_3,Wedge1_6,wedge1,0,-10.5,0],
            [Wedge1_4,Wedge1_7,wedge1,0,-10.5,0],
            [Wedge1_1,Wedge1_cg,wedge1,(1/2)*(6.125*tan(37.5*Pi/180))-
(1/3)*(6.125*tan(37.5*Pi/180)),0,6.125*(1/3)],
            [Wedge1_1,Spring1_1,wedge1,0,0,0],

            [W,Bolsterpsi,w,0,-36,q12],
            [Bolsterpsi,Bolster_cg,bolsterpsi,0,0,6.29],
            [Bolsterpsi,Spring2_1,bolsterpsi,0,36,0],
            [Spring2_1,Spring2_L,bolsterpsi,0,2.96,0],
            [Spring2_1,Spring2_L2,bolsterpsi,0,-2.96,0],

            [Bolsterpsi,Bolster1,bolsterpsi,2+3.827/2,36,(3.827/6.125)*(2+3.827/2)],
            [Bolsterpsi,Bolster2,bolsterpsi,-2-3.827/2,36,(3.827/6.125)*(2+3.827/2)],
            [Bolsterpsi,Bolster3,bolsterpsi,2,36,0],
            [Bolsterpsi,Bolster4,bolsterpsi,-2,36,0],
            [Bolsterpsi,Bolster5,bolsterpsi,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,36,6.125],
            [Bolsterpsi,Bolster6,bolsterpsi,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,36,6.125],
            [Bolsterpsi,Bolster7,bolsterpsi,2,6+36,0],
            [Bolsterpsi,Bolster8,bolsterpsi,2,-6+36,0],
            [Bolsterpsi,Bolster9,bolsterpsi,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,6+36,6.125],
            [Bolsterpsi,Bolster10,bolsterpsi,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-
6+36,6.125],
            [Bolsterpsi,Bolster11,bolsterpsi,-2,6+36,0],
            [Bolsterpsi,Bolster12,bolsterpsi,-2,-6+36,0],
            [Bolsterpsi,Bolster13,bolsterpsi,-
(4+2*(6.125*tan(37.5*3.1415927/180)))/2,6+36,6.125],
```

```
            [Bolsterpsi,Bolster14,bolsterpsi,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-
6+36,6.125],

            [S,Wedge2_1,s,q4,q5,q6],
            [Wedge2_1,Wedge2_2,wedge2,(6.125*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge2_2,Wedge2_3,wedge2,-(6.125*tan(37.5*3.1415927/180)),0,0],
            [Wedge2_3,Wedge2_4,wedge2,0,0,6.125],
            [Wedge2_2,Wedge2_5,wedge2,0,-10.5,0],
            [Wedge2_3,Wedge2_6,wedge2,0,-10.5,0],
            [Wedge2_4,Wedge2_7,wedge2,0,-10.5,0],
            [Wedge2_1,Wedge2_cg,wedge2,-
(1/2)*(6.125*tan(37.5*3.1415927/180))+(1/3)*(6.125*tan(37.5*3.1415927/180)),0,6.125*(1/3)],
            [Wedge2_1,Spring3_1,wedge2,0,0,0],

            [S,Spring1_2,s,(6.125*tan(37.5*3.1415927/180))/2+2,0,0],
            [S,Spring1_3,s,(6.125*tan(37.5*3.1415927/180))/2+2,0,10.25],
            [S,Spring2_2,s,0,0,0],
            [S,Spring2_3,s,0,0,10.25],
            [S,Spring3_2,s,-(6.125*tan(37.5*3.1415927/180))/2-2,0,0],
            [S,Spring3_3,s,-(6.125*tan(37.5*3.1415927/180))/2-2,0,10.25],
            [S,Sideframe1,s,2+(-15/2*tan(toe))+(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe2,s,-2-(-15/2*tan(toe))-(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe3,s,(2+(6.125*tan(37.5*3.1415927/180))),0,0],
            [S,Sideframe4,s,(-2-(6.125*tan(37.5*3.1415927/180))),0,0],
            [W,S,w,q13,0,0]):

DefineTriads([w,s,[0,3]],
             [s,s1,[-toe,2]],
             [s,s2,[toe,2]],
             [s,wedge1,[q7,3],[q8,2]],
             [w,bolsterpsi,[q9,3]],
             [s,wedge2,[q10,3],[q11,2]],
             [bolsterpsi,bolster3,[(37.5*3.1415927/180),2]],
             [bolsterpsi,bolster4,[(-37.5*3.1415927/180),2]]):
DefineNeighbors([world,sideframe],[world,bolsterpsi],[sideframe,wedge1],[sideframe,wedge2],[sidef
rame,sideframe1],[sideframe,sideframe2],
                [bolsterpsi,bolster3],[bolsterpsi,bolster4]):

#Declares the independent velocities as states
DeclareStates(q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q12,q13,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11):

#Define the kinematic differential equations
kde:={q1t=u1,q2t=u2,q3t=u3,q4t=u4,q5t=u5,q6t=u6,q7t=u7,q8t=u8,q9t=u9,q10t=u10,q11t=u11,q12t=(Fext
*omega*cos(omega*t)),q13t=(Fext2*omega2*cos(omega2*t))}:
kde:=simplify(solve(kde,{q1t,q2t,q3t,q4t,q5t,q6t,q7t,q8t,q9t,q10t,q11t,q12t,q13t})):

#Calculate the matrix of vectors which span the subspace of allowable motions, beta
v:=subs(kde,VelocityDescription([LinearVelocity(world,Wedge1_cg),AngularVelocity(w,wedge1),
                                 LinearVelocity(world,Bolster_cg),AngularVelocity(w,bolsterpsi),
                                 LinearVelocity(world,Wedge2_cg),AngularVelocity(w,wedge2)])):
beta:=CoeffExtract(v,[u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11]):

#Inertial forces
p1:=LinearMomentum(world,Wedge1_cg,0.875):
h1:=AngularMomentum(world,wedge1,matrix([[((1/12)*0.875*(10.5^2+6.125^2)),0,0],[0,((1/12)*0.875*(
3.827^2+6.125^2)),0],[0,0,((1/12)*0.875*(3.827^2+10.5^2))]])):
p2:=LinearMomentum(world,Bolster_cg,319):
h2:=AngularMomentum(world,bolsterpsi,matrix([[((1/12)*319*((12^2)+((6.125+5)^2))),0,0],[0,((1/12)
*319*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+((6.125+5)^2))),0],[0,0,((1/12)*319*((4+2*(6.125*ta
n(37.5*3.1415927/180))^2)+(12^2)))]])):
p3:=LinearMomentum(world,Wedge2_cg,0.875):
h3:=AngularMomentum(world,wedge2,matrix([[((1/12)*0.875*(10.5^2+6.125^2)),0,0],[0,((1/12)*0.875*(
3.827^2+6.125^2)),0],[0,0,((1/12)*0.875*(3.827^2+10.5^2))]])):

#I1XX=((1/12)*0.875*(10.5^2+6.125^2));
#I1YY=((1/12)*0.875*(3.827^2+6.125^2));
#I1ZZ=((1/12)*0.875*(3.827^2+10.5^2));
#I2XX=((1/12)*468*((12^2)+((6.125+5)^2)));
#I2YY=((1/12)*468*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+((6.125+5)^2)));
#I2ZZ=((1/12)*468*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+(12^2)));
#I3XX=((1/12)*0.875*(10.5^2+6.125^2));
```

```
#I3YY=((1/12)*0.875*(3.827^2+6.125^2));
#I3ZZ=((1/12)*0.875*(3.827^2+10.5^2));

#rwedge1:=FindTranslation(Spring1_1,Spring1_2):
#rbolster:=FindTranslation(Spring2_1,Spring2_2):
#rwedge2:=FindTranslation(Spring3_1,Spring3_2):

rBolster1:=FindTranslation(Bolster_cg,Bolster1):
rBolster2:=FindTranslation(Bolster_cg,Bolster2):
rBolster3:=FindTranslation(Bolster_cg,Bolster3):
rBolster4:=FindTranslation(Bolster_cg,Bolster4):
rBolster5:=FindTranslation(Bolster_cg,Bolster5):
rBolster6:=FindTranslation(Bolster_cg,Bolster6):
rBolster9:=FindTranslation(Bolster_cg,Bolster9):
rBolster10:=FindTranslation(Bolster_cg,Bolster10):
rBolster13:=FindTranslation(Bolster_cg,Bolster13):
rBolster14:=FindTranslation(Bolster_cg,Bolster14):

rWedge1_2:=FindTranslation(Wedge1_cg,Wedge1_2):
rWedge1_3:=FindTranslation(Wedge1_cg,Wedge1_3):
rWedge1_4:=FindTranslation(Wedge1_cg,Wedge1_4):
rWedge1_5:=FindTranslation(Wedge1_cg,Wedge1_5):
rWedge1_6:=FindTranslation(Wedge1_cg,Wedge1_6):
rWedge1_7:=FindTranslation(Wedge1_cg,Wedge1_7):

rWedge2_2:=FindTranslation(Wedge2_cg,Wedge2_2):
rWedge2_3:=FindTranslation(Wedge2_cg,Wedge2_3):
rWedge2_4:=FindTranslation(Wedge2_cg,Wedge2_4):
rWedge2_5:=FindTranslation(Wedge2_cg,Wedge2_5):
rWedge2_6:=FindTranslation(Wedge2_cg,Wedge2_6):
rWedge2_7:=FindTranslation(Wedge2_cg,Wedge2_7):

rW1:=(1/VectorLength(FindTranslation(Bolster7,Bolster10)))                      &**
FindTranslation(Bolster7,Bolster10):
rW2:=(1/VectorLength(FindTranslation(Bolster8,Bolster9))) &** FindTranslation(Bolster8,Bolster9):
rW3:=(rW2 &xx rW1):

rW4:=(1/VectorLength(FindTranslation(Bolster11,Bolster14)))                     &**
FindTranslation(Bolster11,Bolster14):
rW5:=(1/VectorLength(FindTranslation(Bolster12,Bolster13)))                     &**
FindTranslation(Bolster12,Bolster13):
rW6:=(rW4 &xx rW5):

rW7:=(1/VectorLength(FindTranslation(Bolster3,Bolster5))) &** FindTranslation(Bolster3,Bolster5):
rW8:=(1/VectorLength(FindTranslation(Bolster11,Bolster13)))                     &**
FindTranslation(Bolster11,Bolster13):
rW9:=(1/VectorLength(FindTranslation(Bolster7,Bolster8))) &** FindTranslation(Bolster7,Bolster8):
rW10:=(1/VectorLength(FindTranslation(Bolster11,Bolster12)))                    &**
FindTranslation(Bolster11,Bolster12):

vWedge1_3:=subs(kde,LinearVelocity(sideframe,Wedge1_3))&oo MakeTranslations(s1,3):
vWedge1_6:=subs(kde,LinearVelocity(sideframe,Wedge1_6))&oo MakeTranslations(s1,3):
vWedge1_4:=subs(kde,LinearVelocity(sideframe,Wedge1_4))&oo MakeTranslations(s1,3):
vWedge1_7:=subs(kde,LinearVelocity(sideframe,Wedge1_7))&oo MakeTranslations(s1,3):
vyWedge1_3:=subs(kde,LinearVelocity(sideframe,Wedge1_3))&oo MakeTranslations(s1,2):
vyWedge1_6:=subs(kde,LinearVelocity(sideframe,Wedge1_6))&oo MakeTranslations(s1,2):
vyWedge1_4:=subs(kde,LinearVelocity(sideframe,Wedge1_4))&oo MakeTranslations(s1,2):
vyWedge1_7:=subs(kde,LinearVelocity(sideframe,Wedge1_7))&oo MakeTranslations(s1,2):

vWedge2_3:=subs(kde,LinearVelocity(sideframe,Wedge2_3))&oo MakeTranslations(s2,3):
vWedge2_6:=subs(kde,LinearVelocity(sideframe,Wedge2_6))&oo MakeTranslations(s2,3):
vWedge2_4:=subs(kde,LinearVelocity(sideframe,Wedge2_4))&oo MakeTranslations(s2,3):
vWedge2_7:=subs(kde,LinearVelocity(sideframe,Wedge2_7))&oo MakeTranslations(s2,3):
vyWedge2_3:=subs(kde,LinearVelocity(sideframe,Wedge2_3))&oo MakeTranslations(s2,2):
vyWedge2_6:=subs(kde,LinearVelocity(sideframe,Wedge2_6))&oo MakeTranslations(s2,2):
vyWedge2_4:=subs(kde,LinearVelocity(sideframe,Wedge2_4))&oo MakeTranslations(s2,2):
vyWedge2_7:=subs(kde,LinearVelocity(sideframe,Wedge2_7))&oo MakeTranslations(s2,2):

vBolsterWedge1_2:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_2))&oo rW7):
vBolsterWedge1_5:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_5))&oo rW7):
vBolsterWedge1_4:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_4))&oo rW7):
```

```
vBolsterWedge1_7:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_7))&oo rW7):
vyBolsterWedge1_2:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_2))&oo rW9):
vyBolsterWedge1_5:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_5))&oo rW9):
vyBolsterWedge1_4:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_4))&oo rW9):
vyBolsterWedge1_7:=(subs(kde,LinearVelocity(bolsterpsi,Wedge1_7))&oo rW9):

vBolsterWedge2_2:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_2))&oo rW8):
vBolsterWedge2_5:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_5))&oo rW8):
vBolsterWedge2_4:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_4))&oo rW8):
vBolsterWedge2_7:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_7))&oo rW8):
vyBolsterWedge2_2:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_2))&oo rW10):
vyBolsterWedge2_5:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_5))&oo rW10):
vyBolsterWedge2_4:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_4))&oo rW10):
vyBolsterWedge2_7:=(subs(kde,LinearVelocity(bolsterpsi,Wedge2_7))&oo rW10):

Spring11:=(FindTranslation(Spring1_2,Spring1_3) &oo MakeTranslations(s,3)):
Spring12:=(FindTranslation(Spring1_2,Spring1_1) &oo MakeTranslations(s,3)):
Spring112:=(FindTranslation(Spring1_1,Spring1_1) &oo MakeTranslations(s,2)):
Spring122:=(FindTranslation(Spring1_2,Spring1_1) &oo MakeTranslations(s,2)):

Spring2L:=(FindTranslation(W,Spring2_L) &oo MakeTranslations(s,1)):
Spring2LL:=(FindTranslation(W,Spring2_2) &oo MakeTranslations(s,1)):
Spring2L2:=(FindTranslation(W,Spring2_L2) &oo MakeTranslations(s,1)):
Spring2LL2:=(FindTranslation(W,Spring2_2) &oo MakeTranslations(s,1)):

Spring31:=(FindTranslation(Spring3_2,Spring3_3) &oo MakeTranslations(s,3)):
Spring32:=(FindTranslation(Spring3_2,Spring3_1) &oo MakeTranslations(s,3)):
Spring312:=(FindTranslation(Spring3_1,Spring3_1) &oo MakeTranslations(s,2)):
Spring322:=(FindTranslation(Spring3_2,Spring3_1) &oo MakeTranslations(s,2)):

muzwedge1_3:=(0.4*sqrt(((vWedge1_3)/(sqrt((vWedge1_3)^2+(vyWedge1_3)^2)))^2)):
muzwedge1_6:=(0.4*sqrt(((vWedge1_6)/(sqrt((vWedge1_6)^2+(vyWedge1_6)^2)))^2)):
muzwedge1_4:=(0.4*sqrt(((vWedge1_4)/(sqrt((vWedge1_4)^2+(vyWedge1_4)^2)))^2)):
muzwedge1_7:=(0.4*sqrt(((vWedge1_7)/(sqrt((vWedge1_7)^2+(vyWedge1_7)^2)))^2)):
muywedge1_3:=(0.4*sqrt(((vyWedge1_3)/(sqrt((vWedge1_3)^2+(vyWedge1_3)^2)))^2)):
muywedge1_6:=(0.4*sqrt(((vyWedge1_6)/(sqrt((vWedge1_6)^2+(vyWedge1_6)^2)))^2)):
muywedge1_4:=(0.4*sqrt(((vyWedge1_4)/(sqrt((vWedge1_4)^2+(vyWedge1_4)^2)))^2)):
muywedge1_7:=(0.4*sqrt(((vyWedge1_7)/(sqrt((vWedge1_7)^2+(vyWedge1_7)^2)))^2)):

muzbolsterwedge1_2:=(0.25*sqrt(((vBolsterWedge1_2)/(sqrt((vBolsterWedge1_2)^2+(vyBolsterWedge1_2)
^2)))^2)):
muzbolsterwedge1_5:=(0.25*sqrt(((vBolsterWedge1_5)/(sqrt((vBolsterWedge1_5)^2+(vyBolsterWedge1_5)
^2)))^2)):
muzbolsterwedge1_4:=(0.25*sqrt(((vBolsterWedge1_4)/(sqrt((vBolsterWedge1_4)^2+(vyBolsterWedge1_4)
^2)))^2)):
muzbolsterwedge1_7:=(0.25*sqrt(((vBolsterWedge1_7)/(sqrt((vBolsterWedge1_7)^2+(vyBolsterWedge1_7)
^2)))^2)):
muybolsterwedge1_2:=(0.25*sqrt(((vyBolsterWedge1_2)/(sqrt((vBolsterWedge1_2)^2+(vyBolsterWedge1_2
)^2)))^2)):
muybolsterwedge1_5:=(0.25*sqrt(((vyBolsterWedge1_5)/(sqrt((vBolsterWedge1_5)^2+(vyBolsterWedge1_5
)^2)))^2)):
muybolsterwedge1_4:=(0.25*sqrt(((vyBolsterWedge1_4)/(sqrt((vBolsterWedge1_4)^2+(vyBolsterWedge1_4
)^2)))^2)):
muybolsterwedge1_7:=(0.25*sqrt(((vyBolsterWedge1_7)/(sqrt((vBolsterWedge1_7)^2+(vyBolsterWedge1_7
)^2)))^2)):

muzwedge2_3:=(0.4*sqrt(((vWedge2_3)/(sqrt((vWedge2_3)^2+(vyWedge2_3)^2)))^2)):
muzwedge2_6:=(0.4*sqrt(((vWedge2_6)/(sqrt((vWedge2_6)^2+(vyWedge2_6)^2)))^2)):
muzwedge2_4:=(0.4*sqrt(((vWedge2_4)/(sqrt((vWedge2_4)^2+(vyWedge2_4)^2)))^2)):
muzwedge2_7:=(0.4*sqrt(((vWedge2_7)/(sqrt((vWedge2_7)^2+(vyWedge2_7)^2)))^2)):
muywedge2_3:=(0.4*sqrt(((vyWedge2_3)/(sqrt((vWedge2_3)^2+(vyWedge2_3)^2)))^2)):
muywedge2_6:=(0.4*sqrt(((vyWedge2_6)/(sqrt((vWedge2_6)^2+(vyWedge2_6)^2)))^2)):
muywedge2_4:=(0.4*sqrt(((vyWedge2_4)/(sqrt((vWedge2_4)^2+(vyWedge2_4)^2)))^2)):
muywedge2_7:=(0.4*sqrt(((vyWedge2_7)/(sqrt((vWedge2_7)^2+(vyWedge2_7)^2)))^2)):

muzbolsterwedge2_2:=(0.25*sqrt(((vBolsterWedge2_2)/(sqrt((vBolsterWedge2_2)^2+(vyBolsterWedge2_2)
^2)))^2)):
muzbolsterwedge2_5:=(0.25*sqrt(((vBolsterWedge2_5)/(sqrt((vBolsterWedge2_5)^2+(vyBolsterWedge2_5)
^2)))^2)):
muzbolsterwedge2_4:=(0.25*sqrt(((vBolsterWedge2_4)/(sqrt((vBolsterWedge2_4)^2+(vyBolsterWedge2_4)
^2)))^2)):
```

```
muzbolsterwedge2_7:=(0.25*sqrt(((vBolsterWedge2_7)/(sqrt((vBolsterWedge2_7)^2+(vyBolsterWedge2_7)
^2)))^2)):
muybolsterwedge2_2:=(0.25*sqrt(((vyBolsterWedge2_2)/(sqrt((vBolsterWedge2_2)^2+(vyBolsterWedge2_2
)^2)))^2)):
muybolsterwedge2_5:=(0.25*sqrt(((vyBolsterWedge2_5)/(sqrt((vBolsterWedge2_5)^2+(vyBolsterWedge2_5
)^2)))^2)):
muybolsterwedge2_4:=(0.25*sqrt(((vyBolsterWedge2_4)/(sqrt((vBolsterWedge2_4)^2+(vyBolsterWedge2_4
)^2)))^2)):
muybolsterwedge2_7:=(0.25*sqrt(((vyBolsterWedge2_7)/(sqrt((vBolsterWedge2_7)^2+(vyBolsterWedge2_7
)^2)))^2)):

#External forces
f1:=MakeTranslations(w,0,0,0.875*(-32.2))
    &++ ((k1*(Spring11-Spring12)) &** MakeTranslations(s,3))
    &++ ((k2*(Spring112-Spring122)) &** MakeTranslations(s,2))
    &-- (Cdamp &** LinearVelocity(sideframe1,Wedge1_cg))

    &-- ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &** rW3)

    &++
((kb*rCBolsterWedge1_2*muzbolsterwedge1_2*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,
Wedge1_2) &oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_5*muzbolsterwedge1_5*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,
Wedge1_5) &oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_4*muzbolsterwedge1_4*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,
Wedge1_4) &oo rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_7*muzbolsterwedge1_7*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,
Wedge1_7) &oo rW3)) &** rW7)

    &++
((kb*rCBolsterWedge1_2*muybolsterwedge1_2*tanh(vyBolsterWedge1_2/0.001)*(FindTranslation(Bolster3
,Wedge1_2) &oo rW3)) &** rW9)
    &++
((kb*rCBolsterWedge1_5*muybolsterwedge1_5*tanh(vyBolsterWedge1_5/0.001)*(FindTranslation(Bolster3
,Wedge1_5) &oo rW3)) &** rW9)
    &++
((kb*rCBolsterWedge1_4*muybolsterwedge1_4*tanh(vyBolsterWedge1_4/0.001)*(FindTranslation(Bolster5
,Wedge1_4) &oo rW3)) &** rW9)
    &++
((kb*rCBolsterWedge1_7*muybolsterwedge1_7*tanh(vyBolsterWedge1_7/0.001)*(FindTranslation(Bolster5
,Wedge1_7) &oo rW3)) &** rW9)

    &--            ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    &--            ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)            &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))

    &--
((ks*rCSideframe1Wedge1_3*muzwedge1_3*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_4*muzwedge1_4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_6*muzwedge1_6*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &--
((ks*rCSideframe1Wedge1_7*muzwedge1_7*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
```

```
    &--
((ks*rCSideframe1Wedge1_3*muywedge1_3*tanh(vyWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2))
    &--
((ks*rCSideframe1Wedge1_4*muywedge1_4*tanh(vyWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2))
    &--
((ks*rCSideframe1Wedge1_6*muywedge1_6*tanh(vyWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2))
    &--
((ks*rCSideframe1Wedge1_7*muywedge1_7*tanh(vyWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2))

    &++ NullVector():

t1:=NullVector()
    &-- (Cdamp &** AngularVelocity(s1,wedge1))

    &-- (rWedge1_2 &xx ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &**
rW3))
    &-- (rWedge1_5 &xx ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &**
rW3))
    &-- (rWedge1_4 &xx ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &**
rW3))
    &-- (rWedge1_7 &xx ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &**
rW3))

    &++                                  (rWedge1_2                                  &xx
((kb*rCBolsterWedge1_2*muzbolsterwedge1_2*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,
Wedge1_2) &oo rW3)) &** rW7))
    &++                                  (rWedge1_5                                  &xx
((kb*rCBolsterWedge1_5*muzbolsterwedge1_2*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,
Wedge1_5) &oo rW3)) &** rW7))
    &++                                  (rWedge1_4                                  &xx
((kb*rCBolsterWedge1_4*muzbolsterwedge1_2*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,
Wedge1_4) &oo rW3)) &** rW7))
    &++                                  (rWedge1_7                                  &xx
((kb*rCBolsterWedge1_7*muzbolsterwedge1_2*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,
Wedge1_7) &oo rW3)) &** rW7))

    &++                                  (rWedge1_2                                  &xx
((kb*rCBolsterWedge1_2*muybolsterwedge1_2*tanh(vyBolsterWedge1_2/0.001)*(FindTranslation(Bolster3
,Wedge1_2) &oo rW3)) &** rW9))
    &++                                  (rWedge1_5                                  &xx
((kb*rCBolsterWedge1_5*muybolsterwedge1_5*tanh(vyBolsterWedge1_5/0.001)*(FindTranslation(Bolster3
,Wedge1_5) &oo rW3)) &** rW9))
    &++                                  (rWedge1_4                                  &xx
((kb*rCBolsterWedge1_4*muybolsterwedge1_4*tanh(vyBolsterWedge1_4/0.001)*(FindTranslation(Bolster5
,Wedge1_4) &oo rW3)) &** rW9))
    &++                                  (rWedge1_7                                  &xx
((kb*rCBolsterWedge1_7*muybolsterwedge1_7*tanh(vyBolsterWedge1_7/0.001)*(FindTranslation(Bolster5
,Wedge1_7) &oo rW3)) &** rW9))

    &--   (rWedge1_3   &xx   ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--   (rWedge1_6   &xx   ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--   (rWedge1_4   &xx   ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    &--   (rWedge1_7   &xx   ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))

    &--                                  (rWedge1_3                                  &xx
((ks*rCSideframe1Wedge1_3*muzwedge1_3*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                  (rWedge1_6                                  &xx
((ks*rCSideframe1Wedge1_6*muzwedge1_6*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    &--                                  (rWedge1_4                                  &xx
((ks*rCSideframe1Wedge1_4*muzwedge1_4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
```

```
    &--                                  (rWedge1_7                                 &xx
((ks*rCSideframe1Wedge1_7*muzwedge1_7*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)
&oo MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))

    &--                                  (rWedge1_3                                 &xx
((ks*rCSideframe1Wedge1_3*muzwedge1_3*tanh(vyWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2)))
    &--                                  (rWedge1_6                                 &xx
((ks*rCSideframe1Wedge1_4*muzwedge1_4*tanh(vyWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2)))
    &--                                  (rWedge1_4                                 &xx
((ks*rCSideframe1Wedge1_6*muzwedge1_6*tanh(vyWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2)))
    &--                                  (rWedge1_7                                 &xx
((ks*rCSideframe1Wedge1_7*muzwedge1_7*tanh(vyWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7
) &oo MakeTranslations(s1,1))) &** MakeTranslations(s1,2)))

    &++ NullVector():


f2:=NullVector()


    &++ NullVector():


t2:=NullVector()
    &--     ((FindTranslation(Bolster_cg,Spring2_L))     &xx     ((k3*(Spring2L-Spring2LL))     &**
MakeTranslations(s,1)))
    &--    ((FindTranslation(Bolster_cg,Spring2_L2))    &xx    ((k3*(Spring2L2-Spring2LL2))    &**
MakeTranslations(s,1)))
    &-- (Cdamp &** AngularVelocity(w,bolsterpsi))

    &++                        ((FindTranslation(Bolster_cg,Bolster3))                       &xx
((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &** rW3))
    &++                        ((FindTranslation(Bolster_cg,Bolster3))                       &xx
((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &** rW3))
    &++                        ((FindTranslation(Bolster_cg,Bolster5))                       &xx
((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &** rW3))
    &++                        ((FindTranslation(Bolster_cg,Bolster5))                       &xx
((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &** rW3))

    &--                         (FindTranslation(Bolster_cg,Bolster3)                        &xx
((kb*rCBolsterWedge1_2*muzbolsterwedge1_2*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,
Wedge1_2) &oo rW3)) &** rW7))
    &--                         (FindTranslation(Bolster_cg,Bolster3)                        &xx
((kb*rCBolsterWedge1_5*muzbolsterwedge1_2*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,
Wedge1_5) &oo rW3)) &** rW7))
    &--                         (FindTranslation(Bolster_cg,Bolster5)                        &xx
((kb*rCBolsterWedge1_4*muzbolsterwedge1_2*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,
Wedge1_4) &oo rW3)) &** rW7))
    &--                         (FindTranslation(Bolster_cg,Bolster5)                        &xx
((kb*rCBolsterWedge1_7*muzbolsterwedge1_2*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,
Wedge1_7) &oo rW3)) &** rW7))

    &--                         (FindTranslation(Bolster_cg,Bolster3)                        &xx
((kb*rCBolsterWedge1_2*muybolsterwedge1_2*tanh(vyBolsterWedge1_2/0.001)*(FindTranslation(Bolster3
,Wedge1_2) &oo rW3)) &** rW9))
    &--                         (FindTranslation(Bolster_cg,Bolster3)                        &xx
((kb*rCBolsterWedge1_5*muybolsterwedge1_5*tanh(vyBolsterWedge1_5/0.001)*(FindTranslation(Bolster3
,Wedge1_5) &oo rW3)) &** rW9))
    &--                         (FindTranslation(Bolster_cg,Bolster5)                        &xx
((kb*rCBolsterWedge1_4*muybolsterwedge1_4*tanh(vyBolsterWedge1_4/0.001)*(FindTranslation(Bolster5
,Wedge1_4) &oo rW3)) &** rW9))
    &--                         (FindTranslation(Bolster_cg,Bolster5)                        &xx
((kb*rCBolsterWedge1_7*muybolsterwedge1_7*tanh(vyBolsterWedge1_7/0.001)*(FindTranslation(Bolster5
,Wedge1_7) &oo rW3)) &** rW9))

    &++                        ((FindTranslation(Bolster_cg,Bolster4))                       &xx
((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6))
```

```
    &++                         ((FindTranslation(Bolster_cg,Bolster4))                        &xx
((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6))
    &++                         ((FindTranslation(Bolster_cg,Bolster6))                        &xx
((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6))
    &++                         ((FindTranslation(Bolster_cg,Bolster6))                        &xx
((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6))

    &--                          (FindTranslation(Bolster_cg,Bolster4)                         &xx
((kb*rCBolsterWedge2_2*muzbolsterwedge2_2*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,
Wedge2_2) &oo rW6)) &** rW8))
    &--                          (FindTranslation(Bolster_cg,Bolster4)                         &xx
((kb*rCBolsterWedge2_5*muzbolsterwedge2_2*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,
Wedge2_5) &oo rW6)) &** rW8))
    &--                          (FindTranslation(Bolster_cg,Bolster6)                         &xx
((kb*rCBolsterWedge2_4*muzbolsterwedge2_2*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,
Wedge2_4) &oo rW6)) &** rW8))
    &--                          (FindTranslation(Bolster_cg,Bolster6)                         &xx
((kb*rCBolsterWedge2_7*muzbolsterwedge2_2*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,
Wedge2_7) &oo rW6)) &** rW8))

    &--                          (FindTranslation(Bolster_cg,Bolster4)                         &xx
((kb*rCBolsterWedge2_2*muybolsterwedge2_2*tanh(vyBolsterWedge2_2/0.001)*(FindTranslation(Bolster4
,Wedge2_2) &oo rW6)) &** rW10))
    &--                          (FindTranslation(Bolster_cg,Bolster4)                         &xx
((kb*rCBolsterWedge2_5*muybolsterwedge2_5*tanh(vyBolsterWedge2_5/0.001)*(FindTranslation(Bolster4
,Wedge2_5) &oo rW6)) &** rW10))
    &--                          (FindTranslation(Bolster_cg,Bolster6)                         &xx
((kb*rCBolsterWedge2_4*muybolsterwedge2_4*tanh(vyBolsterWedge2_4/0.001)*(FindTranslation(Bolster6
,Wedge2_4) &oo rW6)) &** rW10))
    &--                          (FindTranslation(Bolster_cg,Bolster6)                         &xx
((kb*rCBolsterWedge2_7*muybolsterwedge2_7*tanh(vyBolsterWedge2_7/0.001)*(FindTranslation(Bolster6
,Wedge2_7) &oo rW6)) &** rW10))

    &++ NullVector():


f3:=MakeTranslations(w,0,0,0.875*(-32.2))
    &++ ((k1*(Spring31-Spring32)) &** MakeTranslations(s,3))
    &++ ((k2*(Spring312-Spring322)) &** MakeTranslations(s,2))
    &-- (Cdamp &** LinearVelocity(sideframe2,Wedge2_cg))

    &-- ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6)
    &-- ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6)
    &-- ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6)
    &-- ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6)

    &++
((kb*rCBolsterWedge2_2*muzbolsterwedge2_2*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,
Wedge2_2) &oo rW6)) &** rW8)
    &++
((kb*rCBolsterWedge2_5*muzbolsterwedge2_5*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,
Wedge2_5) &oo rW6)) &** rW8)
    &++
((kb*rCBolsterWedge2_4*muzbolsterwedge2_4*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,
Wedge2_4) &oo rW6)) &** rW8)
    &++
((kb*rCBolsterWedge2_7*muzbolsterwedge2_7*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,
Wedge2_7) &oo rW6)) &** rW8)

    &++
((kb*rCBolsterWedge2_2*muybolsterwedge2_2*tanh(vyBolsterWedge2_2/0.001)*(FindTranslation(Bolster4
,Wedge2_2) &oo rW6)) &** rW10)
    &++
((kb*rCBolsterWedge2_5*muybolsterwedge2_5*tanh(vyBolsterWedge2_5/0.001)*(FindTranslation(Bolster4
,Wedge2_5) &oo rW6)) &** rW10)
    &++
((kb*rCBolsterWedge2_4*muybolsterwedge2_4*tanh(vyBolsterWedge2_4/0.001)*(FindTranslation(Bolster6
,Wedge2_4) &oo rW6)) &** rW10)
    &++
((kb*rCBolsterWedge2_7*muybolsterwedge2_7*tanh(vyBolsterWedge2_7/0.001)*(FindTranslation(Bolster6
,Wedge2_7) &oo rW6)) &** rW10)
```

```
    &--            ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)            &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--            ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)            &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--            ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)            &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--            ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)            &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))

    &++
((ks*rCSideframe2Wedge2_3*muzwedge2_3*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++
((ks*rCSideframe2Wedge2_4*muzwedge2_4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++
((ks*rCSideframe2Wedge2_6*muzwedge2_6*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++
((ks*rCSideframe2Wedge2_7*muzwedge2_7*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))

    &++
((ks*rCSideframe2Wedge2_3*muywedge2_3*tanh(vyWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2))
    &++
((ks*rCSideframe2Wedge2_4*muywedge2_4*tanh(vyWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2))
    &++
((ks*rCSideframe2Wedge2_6*muywedge2_6*tanh(vyWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2))
    &++
((ks*rCSideframe2Wedge2_7*muywedge2_7*tanh(vyWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2))

    &++ NullVector():

t3:=NullVector()
    &-- (Cdamp &** AngularVelocity(s2,wedge2))

    &--  (rWedge2_2  &xx  ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2)  &oo  rW6))  &**
rW6))
    &--  (rWedge2_5  &xx  ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5)  &oo  rW6))  &**
rW6))
    &--  (rWedge2_4  &xx  ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4)  &oo  rW6))  &**
rW6))
    &--  (rWedge2_7  &xx  ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7)  &oo  rW6))  &**
rW6))

    &++                                (rWedge2_2                                          &xx
((kb*rCBolsterWedge2_2*muzbolsterwedge2_2*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,
Wedge2_2) &oo rW6)) &** rW8))
    &++                                (rWedge2_5                                          &xx
((kb*rCBolsterWedge2_5*muzbolsterwedge2_2*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,
Wedge2_5) &oo rW6)) &** rW8))
    &++                                (rWedge2_4                                          &xx
((kb*rCBolsterWedge2_4*muzbolsterwedge2_2*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,
Wedge2_4) &oo rW6)) &** rW8))
    &++                                (rWedge2_7                                          &xx
((kb*rCBolsterWedge2_7*muzbolsterwedge2_2*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,
Wedge2_7) &oo rW6)) &** rW8))

    &++                                (rWedge2_2                                          &xx
((kb*rCBolsterWedge2_2*muybolsterwedge2_2*tanh(vyBolsterWedge2_2/0.001)*(FindTranslation(Bolster4
,Wedge2_2) &oo rW6)) &** rW10))
    &++                                (rWedge2_5                                          &xx
((kb*rCBolsterWedge2_5*muybolsterwedge2_5*tanh(vyBolsterWedge2_5/0.001)*(FindTranslation(Bolster4
,Wedge2_5) &oo rW6)) &** rW10))
```

```
    &++                                     (rWedge2_4                                      &xx
((kb*rCBolsterWedge2_4*muybolsterwedge2_4*tanh(vyBolsterWedge2_4/0.001)*(FindTranslation(Bolster6
,Wedge2_4) &oo rW6)) &** rW10))
    &++                                     (rWedge2_7                                      &xx
((kb*rCBolsterWedge2_7*muybolsterwedge2_7*tanh(vyBolsterWedge2_7/0.001)*(FindTranslation(Bolster6
,Wedge2_7) &oo rW6)) &** rW10))

    &--  (rWedge2_3  &xx  ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    &--  (rWedge2_6  &xx  ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    &--  (rWedge2_4  &xx  ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    &--  (rWedge2_7  &xx  ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)  &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))

    &++                                     (rWedge2_3                                      &xx
((ks*rCSideframe2Wedge2_3*muzwedge2_3*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    &++                                     (rWedge2_6                                      &xx
((ks*rCSideframe2Wedge2_6*muzwedge2_6*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    &++                                     (rWedge2_4                                      &xx
((ks*rCSideframe2Wedge2_4*muzwedge2_4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    &++                                     (rWedge2_7                                      &xx
((ks*rCSideframe2Wedge2_7*muzwedge2_7*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))

    &++                                     (rWedge2_3                                      &xx
((ks*rCSideframe2Wedge2_3*muzwedge2_3*tanh(vyWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2)))
    &++                                     (rWedge2_6                                      &xx
((ks*rCSideframe2Wedge2_4*muzwedge2_4*tanh(vyWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2)))
    &++                                     (rWedge2_4                                      &xx
((ks*rCSideframe2Wedge2_6*muzwedge2_6*tanh(vyWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2)))
    &++                                     (rWedge2_7                                      &xx
((ks*rCSideframe2Wedge2_7*muzwedge2_7*tanh(vyWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7
) &oo MakeTranslations(s2,1))) &** MakeTranslations(s2,2)))

    &++ NullVector():


#Puts all the forces in a vector
p:=subs(kde,MomentumDescription([p1,h1,p2,h2,p3,h3])):
appliedforce:=ForceDescription([f1,t1,f2,t2,f3,t3]):

eqs:=((subs(kde,DiffTime(p,w)) &-- appliedforce) &oo beta):
diffeqs:=kde union convert(eqs,set):

#Create bodies
DefineObjects([bolsterpsi,'Block',point=MakeTranslations(bolsterpsi,0,19,(6.125+5)/2),xlength=4,y
length=46,zlength=6.125+5,color=red],

[bolsterpsi,'Block',point=MakeTranslations(bolsterpsi,0,19,6.125+2.5),xlength=4+2*(6.125*tan(37.5
*3.1415927/180)),ylength=46,zlength=5,color=red],
              [bolster3,'Block',point=MakeTranslations(bolster3,-2,-
17,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2))/2),xlength=4,ylength=46,zlength=sqrt((6.125*
tan(37.5*3.1415927/180))^2+6.125^2),color=red],
              [bolster4,'Block',point=MakeTranslations(bolster4,2,-
17,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2))/2),xlength=4,ylength=46,zlength=sqrt((6.125*
tan(37.5*3.1415927/180))^2+6.125^2),color=red],


[wedge1,'Block',point=Wedge1_1,xlength=(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,
color=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlengt
h=0.05,ylength=10.5,zlength=6.125,color=blue],
```

```
[wedge1,'Block',point=MakeTranslations(wedge1,0,0,6.125/2),orient=MakeRotations([37.5*3.1415927/1
80,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2),color=bl
ue],

              [wedge2,'Block',point=Wedge2_1,xlength=-
(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,-
(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlength=0.05,ylength=10.5,zlength=6.125,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,0,0,6.125/2),orient=MakeRotations([-
37.5*3.1415927/180,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6
.125^2),color=blue],


[sideframe,'Cylinder',point=MakeTranslations(s,(6.125*tan(37.5*3.1415927/180))/2+2,0,(VectorLengt
h(FindTranslation(Spring1_1,Spring1_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring
1_1,Spring1_2)),color=green],

[sideframe,'Cylinder',point=MakeTranslations(s,0,0,(VectorLength(FindTranslation(Spring2_1,Spring
2_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring2_1,Spring2_2)),color=green],
              [sideframe,'Cylinder',point=MakeTranslations(s,-(6.125*tan(37.5*3.1415927/180))/2-
2,0,(VectorLength(FindTranslation(Spring3_1,Spring3_2)))/2),radius=0.5,length=VectorLength(FindTr
anslation(Spring3_1,Spring3_2)),color=green],


[sideframe,'Block',point=S,xlength=4+2*(6.125*tan(37.5*3.1415927/180)),ylength=12,zlength=0.01,co
lor=white],

[sideframe1,'Block',point=MakeTranslations(s1,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],

[sideframe2,'Block',point=MakeTranslations(s2,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],

#Visual placement of points
              [world,'Sphere',point=W,radius=0.05,color=green],

              [sideframe,'Sphere',point=Sideframe1,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe2,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe3,radius=0.05,color=red],
              [sideframe,'Sphere',point=Sideframe4,radius=0.05,color=red],
              [bolsterpsi,'Sphere',point=Bolster_cg,radius=0.05,color=green],
              [bolsterpsi,'Sphere',point=Bolster1,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Bolster2,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Bolster3,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Bolster4,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Bolster5,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Bolster6,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Bolster7,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Bolster8,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Bolster9,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Bolster10,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Bolster11,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Bolster12,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Bolster13,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Bolster14,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Spring2_L,radius=0.05,color=blue],
              [bolsterpsi,'Sphere',point=Spring2_L2,radius=0.05,color=blue],

              [wedge1,'Sphere',point=Wedge1_1,radius=0.05,color=yellow],
              [wedge1,'Sphere',point=Wedge1_2,radius=0.05,color=red],
              [wedge1,'Sphere',point=Wedge1_3,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_4,radius=0.05,color=blue],
              [wedge1,'Sphere',point=Wedge1_5,radius=0.05,color=red],
              [wedge1,'Sphere',point=Wedge1_6,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_7,radius=0.05,color=blue],
              [wedge1,'Sphere',point=Wedge1_cg,radius=0.05,color=green],

              [wedge2,'Sphere',point=Wedge2_1,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Wedge2_2,radius=0.05,color=red],
              [wedge2,'Sphere',point=Wedge2_3,radius=0.05,color=green],
```

```
              [wedge2,'Sphere',point=Wedge2_4,radius=0.05,color=blue],
              [wedge2,'Sphere',point=Wedge2_5,radius=0.05,color=red],
              [wedge2,'Sphere',point=Wedge2_6,radius=0.05,color=green],
              [wedge2,'Sphere',point=Wedge2_7,radius=0.05,color=blue],
              [wedge2,'Sphere',point=Wedge2_cg,radius=0.05,color=green],

              [wedge1,'Sphere',point=Spring1_1,radius=0.05,color=yellow],
              [wedge1,'Sphere',point=Spring1_2,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Spring2_1,radius=0.05,color=yellow],
              [bolsterpsi,'Sphere',point=Spring2_2,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Spring3_1,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Spring3_2,radius=0.05,color=yellow]):

#Outputs
GeometryOutput(main=world,

parameters=[toe=0,omega=0,Fext=0,omega2=0,Fext2=0,ks=1000000,kb=1000000,Cdamp=1,k1=1491,k2=1,k3=1
],
       states=[q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q12,q13,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11],
        filename="halftruckvardamp_ver3_8.geo"):
MotionOutput(ode=diffeqs,

parameters=[toe=0,omega=0,Fext=0,omega2=0,Fext2=0,ks=1000000,kb=1000000,Cdamp=1,k1=1491,k2=1,k3=1
],
        states=[q1=4.35,q2,q3=10.25,q4=-
4.35,q5,q6=10.25,q7,q8,q9,q10,q11,q12=10.3,q13,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11],
        insignals=[rCBolsterWedge1_2=((FindTranslation(Wedge1_2,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_4=((FindTranslation(Wedge1_4,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_5=((FindTranslation(Wedge1_5,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_7=((FindTranslation(Wedge1_7,Bolster3) &oo rW3)&>0),
                   rCSideframe1Wedge1_3=((FindTranslation(Sideframe1,Wedge1_3)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_4=((FindTranslation(Sideframe1,Wedge1_4)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_6=((FindTranslation(Sideframe1,Wedge1_6)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_7=((FindTranslation(Sideframe1,Wedge1_7)                &oo
MakeTranslations(s1,1))&>0),
                   rCBolsterWedge2_2=((FindTranslation(Wedge2_2,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_4=((FindTranslation(Wedge2_4,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_5=((FindTranslation(Wedge2_5,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_7=((FindTranslation(Wedge2_7,Bolster4) &oo rW6)&>0),
                   rCSideframe2Wedge2_3=((FindTranslation(Sideframe2,Wedge2_3)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_4=((FindTranslation(Sideframe2,Wedge2_4)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_6=((FindTranslation(Sideframe2,Wedge2_6)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_7=((FindTranslation(Sideframe2,Wedge2_7)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe1Bolster9=((FindTranslation(Sideframe1,Bolster9)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Bolster10=((FindTranslation(Sideframe1,Bolster10)              &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe2Bolster13=((FindTranslation(Sideframe2,Bolster13)              &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Bolster14=((FindTranslation(Sideframe2,Bolster14)              &oo
MakeTranslations(s2,1))&<0)],
        filename="halftruckvardamp_ver3_8.dyn"):
```

7.1.3.2. Half-truck Variably Damped Warp State Calculation M-file

```
%Parameters
toe=0;
omega=12;
Fext=1;
omega2=1.5;
Fext2=0.25;
ks=1000000;
kb=1000000;
Cdamp=10;
k1=1491;

q1=y(:,1);
q2=y(:,2);
q3=y(:,3);
q4=y(:,4);
q5=y(:,5);
q6=y(:,6);
q7=y(:,7);
q8=y(:,8);
q10=y(:,9);
q11=y(:,10);
q12=y(:,11);
q13=y(:,12);
u1=y(:,13);
u2=y(:,14);
u3=y(:,15);
u4=y(:,16);
u5=y(:,17);
u6=y(:,18);
u7=y(:,19);
u8=y(:,20);
u10=y(:,21);
u11=y(:,22);


for i = 1:length(t),

if                           ((1.443976106-.7219880530*q1(i)-.5540009288*q12(i)+.5540009288*q3(i)-
.7219880530*q13(i)+3.062500000*tan(.2083333333*3.1415927)*(.7219880530*cos(q7(i))*cos(q8(i))+.554
0009288*sin(q8(i)))+3.790437278*sin(q7(i)))>0),
        rCBolsterWedge1_2(i) = 1;
    else
        rCBolsterWedge1_2(i) = 0;
    end

if                   ((-.7219880530*q1(i)-.5540009288*q12(i)+.5540009288*q3(i)-.7219880530*q13(i)-
3.062500000*tan(.2083333333*3.1415927)*(.7219880530*cos(q7(i))*cos(q8(i))+.5540009288*sin(q8(i)))
+3.790437278*sin(q7(i))-4.422176825*cos(q7(i))*sin(q8(i))+3.393255689*cos(q8(i))+1.443976106)>0),
       rCBolsterWedge1_4(i) = 1;
    else
       rCBolsterWedge1_4(i) = 0;
    end

if                           ((1.443976106-.7219880530*q1(i)-.5540009288*q12(i)+.5540009288*q3(i)-
.7219880530*q13(i)+3.062500000*tan(.2083333333*3.1415927)*(.7219880530*cos(q7(i))*cos(q8(i))+.554
0009288*sin(q8(i)))-3.790437278*sin(q7(i)))>0),
        rCBolsterWedge1_5(i) = 1;
    else
        rCBolsterWedge1_5(i) = 0;
    end

if                   ((-.7219880530*q1(i)-.5540009288*q12(i)+.5540009288*q3(i)-.7219880530*q13(i)-
3.062500000*tan(.2083333333*3.1415927)*(.7219880530*cos(q7(i))*cos(q8(i))+.5540009288*sin(q8(i)))
-3.790437278*sin(q7(i))-4.422176825*cos(q7(i))*sin(q8(i))+3.393255689*cos(q8(i))+1.443976106)>0),
        rCBolsterWedge1_7(i) = 1;
```

```
    else
        rCBolsterWedge1_7(i) = 0;
    end

if                                                  (((-6.699877893+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q3(i))*sin(toe)+3.062500000*tan(.2083333333*3.1415927)*(cos(q7(i))*cos(q8(i))*cos(toe)-
sin(q8(i))*sin(toe))-5.250000000*sin(q7(i))*cos(toe))>0),
        rCSideframe1Wedge1_3(i) = 1;
    else
        rCSideframe1Wedge1_3(i) = 0;
    end

if                                                  (((-6.699877893+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q3(i))*sin(toe)+3.062500000*tan(.2083333333*3.1415927)*(cos(q7(i))*cos(q8(i))*cos(toe)-
sin(q8(i))*sin(toe))-
5.250000000*sin(q7(i))*cos(toe)+6.125*cos(q7(i))*sin(q8(i))*cos(toe)+6.125*cos(q8(i))*sin(toe))>0
),
        rCSideframe1Wedge1_4(i) = 1;
    else
        rCSideframe1Wedge1_4(i) = 0;
    end

if                                                  (((-6.699877893+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q3(i))*sin(toe)+3.062500000*tan(.2083333333*3.1415927)*(cos(q7(i))*cos(q8(i))*cos(toe)-
sin(q8(i))*sin(toe))+5.250000000*sin(q7(i))*cos(toe))>0),
       rCSideframe1Wedge1_6(i) = 1;
    else
       rCSideframe1Wedge1_6(i) = 0;
    end

if                                                  (((-6.699877893+15/2*tan(toe)+q1(i))*cos(toe)+(-
15/2+q3(i))*sin(toe)+3.062500000*tan(.2083333333*3.1415927)*(cos(q7(i))*cos(q8(i))*cos(toe)-
sin(q8(i))*sin(toe))+5.250000000*sin(q7(i))*cos(toe)+6.125*cos(q7(i))*sin(q8(i))*cos(toe)+6.125*c
os(q8(i))*sin(toe))>0),
        rCSideframe1Wedge1_7(i) = 1;
    else
        rCSideframe1Wedge1_7(i) = 0;
    end

if                                                                  ((1.443976106+.7219880530*q4(i)-
.5540009288*q12(i)+.5540009288*q6(i)+.7219880530*q13(i)+1.696627844*cos(q10(i))*cos(q11(i))-
1.301868359*sin(q11(i))-3.790437278*sin(q10(i)))>0),
        rCBolsterWedge2_2(i) = 1;
    else
        rCBolsterWedge2_2(i) = 0;
    end

if                    ((.7219880530*q4(i)-.5540009288*q12(i)+.5540009288*q6(i)+.7219880530*q13(i)-
1.696627845*cos(q10(i))*cos(q11(i))+1.301868359*sin(q11(i))-
3.790437278*sin(q10(i))+4.422176825*cos(q10(i))*sin(q11(i))+3.393255689*cos(q11(i))+1.443976106)>
0),
        rCBolsterWedge2_4(i) = 1;
    else
        rCBolsterWedge2_4(i) = 0;
    end

if                                                                  ((1.443976106+.7219880530*q4(i)-
.5540009288*q12(i)+.5540009288*q6(i)+.7219880530*q13(i)+1.696627844*cos(q10(i))*cos(q11(i))-
1.301868359*sin(q11(i))+3.790437278*sin(q10(i)))>0),
        rCBolsterWedge2_5(i) = 1;
    else
        rCBolsterWedge2_5(i) = 0;
    end

if                    ((.7219880530*q4(i)-.5540009288*q12(i)+.5540009288*q6(i)+.7219880530*q13(i)-
1.696627845*cos(q10(i))*cos(q11(i))+1.301868359*sin(q11(i))+3.790437278*sin(q10(i))+4.422176825*c
os(q10(i))*sin(q11(i))+3.393255689*cos(q11(i))+1.443976106)>0),
        rCBolsterWedge2_7(i) = 1;
    else
        rCBolsterWedge2_7(i) = 0;
    end
```

```
if                            (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q6(i))*sin(toe)-
2.349938947*cos(q10(i))*cos(q11(i))*cos(toe)-2.349938947*sin(q11(i))*sin(toe)-
5.250000000*sin(q10(i))*cos(toe))<0),
        rCSideframe2Wedge2_3(i) = 1;
    else
        rCSideframe2Wedge2_3(i) = 0;
    end

if                            (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q6(i))*sin(toe)-
2.349938947*cos(q10(i))*cos(q11(i))*cos(toe)-2.349938947*sin(q11(i))*sin(toe)-
5.250000000*sin(q10(i))*cos(toe)+6.125*cos(q10(i))*sin(q11(i))*cos(toe)-
6.125*cos(q11(i))*sin(toe))<0),
        rCSideframe2Wedge2_4(i) = 1;
    else
        rCSideframe2Wedge2_4(i) = 0;
    end

if                            (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q6(i))*sin(toe)-
2.349938947*cos(q10(i))*cos(q11(i))*cos(toe)-
2.349938947*sin(q11(i))*sin(toe)+5.250000000*sin(q10(i))*cos(toe))<0),
        rCSideframe2Wedge2_6(i) = 1;
    else
        rCSideframe2Wedge2_6(i) = 0;
    end

if                            (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q6(i))*sin(toe)-
2.349938947*cos(q10(i))*cos(q11(i))*cos(toe)-
2.349938947*sin(q11(i))*sin(toe)+5.250000000*sin(q10(i))*cos(toe)+6.125*cos(q10(i))*sin(q11(i))*c
os(toe)-6.125*cos(q11(i))*sin(toe))<0),
        rCSideframe2Wedge2_7(i) = 1;
    else
        rCSideframe2Wedge2_7(i) = 0;
    end
```

7.1.4. Quarter-truck Variably Damped Model With Wedge Face Curvature

7.1.4.1. Quarter-truck Variably Damped Model With Wedge Face Curvature SRC file

```
with(Mambo):

#Declare observers, points, and triads
DeclareObservers(world,bolster,bolster3,bolster4,wedge1,wedge2,sideframe,sideframe1,sideframe2):
DeclarePoints(W,Bolster,Bolster1,Bolster2,Bolster_cg,Bolster3,Bolster4,Bolster5,Bolster6,Bolster7
,Bolster8,Bolster9,Bolster10,

Bolster11,Bolster12,Bolster13,Bolster14,Wedge1_1,Wedge1_2,Wedge1_3,Wedge1_4,Wedge1_5,Wedge1_6,Wed
ge1_7,Wedge1_cg,

Wedge2_1,Wedge2_2,Wedge2_3,Wedge2_4,Wedge2_5,Wedge2_6,Wedge2_7,Wedge2_cg,Rad,Wedge2_8,Wedge2_9,We
dge2_10,

Spring1_1,Spring1_2,Spring1_3,Spring2_1,Spring2_2,Spring2_3,Spring3_1,Spring3_2,Spring3_3,S,Sidef
rame1,Sideframe2,Sideframe3,Sideframe4):
DeclareTriads(w,bolster,bolster1,wedge1,wedge2,s,s1,s2,bolster3,bolster4):


#Define observers, points, triads, and neighbors
DefineObservers([world,W,w],[bolster,Bolster,bolster],[wedge1,Wedge1_1,wedge1],[wedge2,Wedge2_1,w
edge2],

[sideframe,S,s],[bolster3,Bolster3,bolster3],[bolster4,Bolster4,bolster4],[sideframe1,Sideframe3,
s1],
                [sideframe2,Sideframe4,s2]):
DefinePoints([S,Wedge1_1,s,q1,0,q2],
            [Wedge1_1,Wedge1_2,wedge1,-(6.125*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge1_2,Wedge1_3,wedge1,(6.125*tan(37.5*3.1415927/180)),0,0],
            [Wedge1_3,Wedge1_4,wedge1,0,0,6.125],
            [Wedge1_2,Wedge1_5,wedge1,0,-10.5,0],
            [Wedge1_3,Wedge1_6,wedge1,0,-10.5,0],
            [Wedge1_4,Wedge1_7,wedge1,0,-10.5,0],
            [Wedge1_1,Wedge1_cg,wedge1,(1/2)*(6.125*tan(37.5*3.1415927/180))-
(1/3)*(6.125*tan(37.5*3.1415927/180)),0,6.125*(1/3)],
            [Wedge1_1,Spring1_1,wedge2,0,0,0],

            [S,Bolster,s,0,0,q3],
            [Bolster,Bolster_cg,bolster,0,0,6.29],
            [Bolster,Spring2_1,bolster,0,0,0],

            [Bolster,Bolster1,bolster,2+3.827/2,0,(3.827/6.125)*(2+3.827/2)],
            [Bolster,Bolster2,bolster,-2-3.827/2,0,(3.827/6.125)*(2+3.827/2)],
            [Bolster,Bolster3,bolster,2,0,0],
            [Bolster,Bolster4,bolster,-2,0,0],
            [Bolster,Bolster5,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,0,6.125],
            [Bolster,Bolster6,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,0,6.125],
            [Bolster,Bolster7,bolster,2,6,0],
            [Bolster,Bolster8,bolster,2,-6,0],
            [Bolster,Bolster9,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,6,6.125],
            [Bolster,Bolster10,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-6,6.125],
            [Bolster,Bolster11,bolster,-2,5.25,0],
            [Bolster,Bolster12,bolster,-2,-5.25,0],
            [Bolster,Bolster13,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,5.25,6.125],
            [Bolster,Bolster14,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-5.25,6.125],

            [S,Wedge2_1,s,q4,0,q5],
            [Wedge2_1,Wedge2_2,wedge2,(6.125*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge2_2,Wedge2_3,wedge2,-(6.125*tan(37.5*3.1415927/180)),0,0],
            [Wedge2_2,Wedge2_8,wedge2,0,-10.5/2,0],
            [Wedge2_8,Wedge2_9,wedge2,-(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2],
```

```
            [Wedge2_3,Wedge2_4,wedge2,0,0,6.125],
            [Wedge2_2,Wedge2_5,wedge2,0,-10.5,0],
            [Wedge2_4,Wedge2_10,wedge2,0,-10.5/2,0],
            [Wedge2_3,Wedge2_6,wedge2,0,-10.5,0],
            [Wedge2_4,Wedge2_7,wedge2,0,-10.5,0],
            [Wedge2_1,Wedge2_cg,wedge2,-
(1/2)*(6.125*tan(37.5*3.1415927/180))+(1/3)*(6.125*tan(37.5*3.1415927/180)),0,6.125*(1/3)],
            [Wedge2_1,Spring3_1,wedge2,0,0,0],

            [S,Spring1_2,s,(6.125*tan(37.5*3.1415927/180))/2+2,0,0],
            [S,Spring1_3,s,(6.125*tan(37.5*3.1415927/180))/2+2,0,10.25],
            [S,Spring2_2,s,0,0,0],
            [S,Spring2_3,s,0,0,10.25],
            [S,Spring3_2,s,-(6.125*tan(37.5*3.1415927/180))/2-2,0,0],
            [S,Spring3_3,s,-(6.125*tan(37.5*3.1415927/180))/2-2,0,10.25],
            [S,Sideframe1,s,2+(-15/2*tan(toe))+(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe2,s,-2-(-15/2*tan(toe))-(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S,Sideframe3,s,(2+(6.125*tan(37.5*3.1415927/180))),0,0],
            [S,Sideframe4,s,(-2-(6.125*tan(37.5*3.1415927/180))),0,0],
            [W,S,w,0,0,q10],
            [Wedge2_1,Rad,wedge2,X,0,Z]):

DefineTriads([w,s,[0,1]],
             [s,s1,[-toe,2]],
             [s,s2,[toe,2]],
             [s,wedge1,[q6,3],[q7,2]],
             [s,bolster,[psi,3]],
             [s,wedge2,[q8,3],[q9,2]],
             [bolster,bolster3,[(37.5*3.1415927/180),2]],
             [bolster,bolster4,[(-37.5*3.1415927/180),2]]):
DefineNeighbors([world,sideframe],[sideframe,bolster],[sideframe,wedge1],[sideframe,wedge2],[side
frame,sideframe1],[sideframe,sideframe2],
                [bolster,bolster3],[bolster,bolster4]):

#Declares the independent velocities as states
DeclareStates(q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,u1,u2,u3,u4,u5,u6,u7,u8,u9):

#Define the kinematic differential equations
kde:={q1t=u1,q2t=u2,q3t=u3,q4t=u4,q5t=u5,q6t=u6,q7t=u7,q8t=u8,q9t=u9,q10t=(Fext*omega*cos(omega*t
))}:
kde:=simplify(solve(kde,{q1t,q2t,q3t,q4t,q5t,q6t,q7t,q8t,q9t,q10t})):

#Calculate the matrix of vectors which span the subspace of allowable motions, beta
v:=subs(kde,VelocityDescription([LinearVelocity(world,Wedge1_cg),AngularVelocity(w,wedge1),
                                 LinearVelocity(world,Bolster_cg),AngularVelocity(w,bolster),
                                 LinearVelocity(world,Wedge2_cg),AngularVelocity(w,wedge2)])):
beta:=CoeffExtract(v,[u1,u2,u3,u4,u5,u6,u7,u8,u9]):

#Inertial forces
p1:=LinearMomentum(world,Wedge1_cg,m1):
h1:=AngularMomentum(world,wedge1,matrix([[((1/12)*0.875*(10.5^2+6.125^2)),0,0],[0,((1/12)*0.875*(
3.827^2+6.125^2)),0],[0,0,((1/12)*0.875*(3.827^2+10.5^2))]])):
p2:=LinearMomentum(world,Bolster_cg,m2):
h2:=AngularMomentum(world,bolster,matrix([[((1/12)*m2*((12^2)+((6.125+5)^2))),0,0],[0,((1/12)*m2*
((4+2*(6.125*tan(37.5*3.1415927/180))^2)+((6.125+5)^2))),0],[0,0,((1/12)*m2*((4+2*(6.125*tan(37.5
*3.1415927/180))^2)+(12^2)))]])):
p3:=LinearMomentum(world,Wedge2_cg,m3):
h3:=AngularMomentum(world,wedge2,matrix([[((1/12)*0.875*(10.5^2+6.125^2)),0,0],[0,((1/12)*0.875*(
3.827^2+6.125^2)),0],[0,0,((1/12)*0.875*(3.827^2+10.5^2))]])):

#I1XX=((1/12)*0.875*(10.5^2+6.125^2));
#I1YY=((1/12)*0.875*(3.827^2+6.125^2));
#I1ZZ=((1/12)*0.875*(3.827^2+10.5^2));
#I2XX=((1/12)*468*((12^2)+((6.125+5)^2)));
#I2YY=((1/12)*468*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+((6.125+5)^2)));
#I2ZZ=((1/12)*468*((4+2*(6.125*tan(37.5*3.1415927/180))^2)+(12^2)));
#I3XX=((1/12)*0.875*(10.5^2+6.125^2));
#I3YY=((1/12)*0.875*(3.827^2+6.125^2));
#I3ZZ=((1/12)*0.875*(3.827^2+10.5^2));

rwedge1:=FindTranslation(Spring1_1,Spring1_2):
```

```
rbolster:=FindTranslation(Spring2_1,Spring2_2):
rwedge2:=FindTranslation(Spring3_1,Spring3_2):

rBolsterR2:=FindTranslation(Bolster_cg,Bolster2):
rBolsterR4:=FindTranslation(Bolster_cg,Bolster4):
rBolsterR6:=FindTranslation(Bolster_cg,Bolster6):
rBolsterR11:=FindTranslation(Bolster_cg,Bolster11):
rBolsterR12:=FindTranslation(Bolster_cg,Bolster12):
rBolsterR13:=FindTranslation(Bolster_cg,Bolster13):
rBolsterR14:=FindTranslation(Bolster_cg,Bolster14):

rWedge2R2:=FindTranslation(Wedge2_cg,Wedge2_9):
rWedge2R4:=FindTranslation(Wedge2_cg,Wedge2_8):
rWedge2R6:=FindTranslation(Wedge2_cg,Wedge2_10):
rWedge2R11:=FindTranslation(Wedge2_cg,Wedge2_2):
rWedge2R12:=FindTranslation(Wedge2_cg,Wedge2_5):
rWedge2R13:=FindTranslation(Wedge2_cg,Wedge2_4):
rWedge2R14:=FindTranslation(Wedge2_cg,Wedge2_7):

rWedge1_2:=FindTranslation(Wedge1_cg,Wedge1_2):
rWedge1_3:=FindTranslation(Wedge1_cg,Wedge1_3):
rWedge1_4:=FindTranslation(Wedge1_cg,Wedge1_4):
rWedge1_5:=FindTranslation(Wedge1_cg,Wedge1_5):
rWedge1_6:=FindTranslation(Wedge1_cg,Wedge1_6):
rWedge1_7:=FindTranslation(Wedge1_cg,Wedge1_7):

rWedge2_2:=FindTranslation(Wedge2_cg,Wedge2_2):
rWedge2_3:=FindTranslation(Wedge2_cg,Wedge2_3):
rWedge2_4:=FindTranslation(Wedge2_cg,Wedge2_4):
rWedge2_5:=FindTranslation(Wedge2_cg,Wedge2_5):
rWedge2_6:=FindTranslation(Wedge2_cg,Wedge2_6):
rWedge2_7:=FindTranslation(Wedge2_cg,Wedge2_7):

rW1:=(1/VectorLength(FindTranslation(Bolster7,Bolster10)))                                    &**
FindTranslation(Bolster7,Bolster10):
rW2:=(1/VectorLength(FindTranslation(Bolster8,Bolster9))) &** FindTranslation(Bolster8,Bolster9):
rW3:=(rW2 &xx rW1):

rW4:=(1/VectorLength(FindTranslation(Bolster11,Bolster14)))                                   &**
FindTranslation(Bolster11,Bolster14):
rW5:=(1/VectorLength(FindTranslation(Bolster12,Bolster13)))                                   &**
FindTranslation(Bolster12,Bolster13):
rW6:=(rW4 &xx rW5):

rW7:=(1/VectorLength(FindTranslation(Bolster3,Bolster5))) &** FindTranslation(Bolster3,Bolster5):
rW8:=(1/VectorLength(FindTranslation(Bolster11,Bolster13)))                                   &**
FindTranslation(Bolster11,Bolster13):

vWedge1_3:=subs(kde,LinearVelocity(sideframe,Wedge1_3))&oo MakeTranslations(s,3):
vWedge1_6:=subs(kde,LinearVelocity(sideframe,Wedge1_6))&oo MakeTranslations(s,3):
vWedge1_4:=subs(kde,LinearVelocity(sideframe,Wedge1_4))&oo MakeTranslations(s,3):
vWedge1_7:=subs(kde,LinearVelocity(sideframe,Wedge1_7))&oo MakeTranslations(s,3):

vWedge2_3:=subs(kde,LinearVelocity(sideframe,Wedge2_3))&oo MakeTranslations(s,3):
vWedge2_6:=subs(kde,LinearVelocity(sideframe,Wedge2_6))&oo MakeTranslations(s,3):
vWedge2_4:=subs(kde,LinearVelocity(sideframe,Wedge2_4))&oo MakeTranslations(s,3):
vWedge2_7:=subs(kde,LinearVelocity(sideframe,Wedge2_7))&oo MakeTranslations(s,3):

vBolsterR2:=(subs(kde,LinearVelocity(bolster,Wedge2_9))&oo rW8):
vBolsterR4:=(subs(kde,LinearVelocity(bolster,Wedge2_8))&oo rW8):
vBolsterR6:=(subs(kde,LinearVelocity(bolster,Wedge2_10))&oo rW8):
vBolsterR11:=(subs(kde,LinearVelocity(bolster,Wedge2_2))&oo rW8):
vBolsterR12:=(subs(kde,LinearVelocity(bolster,Wedge2_5))&oo rW8):
vBolsterR13:=(subs(kde,LinearVelocity(bolster,Wedge2_4))&oo rW8):
vBolsterR14:=(subs(kde,LinearVelocity(bolster,Wedge2_7))&oo rW8):

Spring11:=(FindTranslation(Spring1_2,Spring1_3) &oo MakeTranslations(s,3)):
Spring12:=(FindTranslation(Spring1_2,Spring1_1) &oo MakeTranslations(s,3)):
Spring21:=(FindTranslation(Spring2_2,Spring2_3) &oo MakeTranslations(s,3)):
Spring22:=(FindTranslation(Spring2_2,Spring2_1) &oo MakeTranslations(s,3)):
Spring31:=(FindTranslation(Spring3_2,Spring3_3) &oo MakeTranslations(s,3)):
```

```
Spring32:=(FindTranslation(Spring3_2,Spring3_1) &oo MakeTranslations(s,3)):

kb:=(10000-(10000*t/20)):

#External forces
f1:=NullVector()
    #MakeTranslations(w,0,0,m1*(-32.2))
    #&++ ((k1*(Spring11-Spring12)) &** MakeTranslations(s,3))
    #&-- (Cdamp &** LinearVelocity(sideframe1,Wedge1_cg))

    #&--            ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    #&--            ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    #&--            ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))
    #&--            ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)           &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1))

    #&--
((ks*rCSideframe1Wedge1_3*0.4*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    #&--
((ks*rCSideframe1Wedge1_6*0.4*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    #&--
((ks*rCSideframe1Wedge1_4*0.4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    #&--
((ks*rCSideframe1Wedge1_7*0.4*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3))
    &++ NullVector():

t1:=NullVector()
    #&-- (Cdamp &** AngularVelocity(s1,wedge1))

    #&--   (rWedge1_3   &xx   ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    #&--   (rWedge1_6   &xx   ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    #&--   (rWedge1_4   &xx   ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))
    #&--   (rWedge1_7   &xx   ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)   &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,1)))

    #&--                                      (rWedge1_3                                      &xx
((ks*rCSideframe1Wedge1_3*0.4*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    #&--                                      (rWedge1_6                                      &xx
((ks*rCSideframe1Wedge1_6*0.4*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    #&--                                      (rWedge1_4                                      &xx
((ks*rCSideframe1Wedge1_4*0.4*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))
    #&--                                      (rWedge1_7                                      &xx
((ks*rCSideframe1Wedge1_7*0.4*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)     &oo
MakeTranslations(s1,1))) &** MakeTranslations(s1,3)))

    &++ NullVector():

f2:=MakeTranslations(w,0,0,m2*(-32.2))
    &++ ((k2*(Spring21-Spring22)) &** MakeTranslations(s,3))
    &-- (Cdamp &** LinearVelocity(sideframe,Bolster_cg))

    &--           ((kb*rCBolsterR2*(R-VectorLength(FindTranslation(Bolster2,Rad))))          &**
FindTranslation(Bolster2,Rad))
    &--           ((kb*rCBolsterR4*(R-VectorLength(FindTranslation(Bolster4,Rad))))          &**
FindTranslation(Bolster4,Rad))
    &--           ((kb*rCBolsterR6*(R-VectorLength(FindTranslation(Bolster6,Rad))))          &**
FindTranslation(Bolster6,Rad))
```

```
    &--          ((kb*rCBolsterR11*(R-VectorLength(FindTranslation(Bolster11,Rad))))          &**
FindTranslation(Bolster11,Rad))
    &--          ((kb*rCBolsterR12*(R-VectorLength(FindTranslation(Bolster12,Rad))))          &**
FindTranslation(Bolster12,Rad))
    &--          ((kb*rCBolsterR13*(R-VectorLength(FindTranslation(Bolster13,Rad))))          &**
FindTranslation(Bolster13,Rad))
    &--          ((kb*rCBolsterR14*(R-VectorLength(FindTranslation(Bolster14,Rad))))          &**
FindTranslation(Bolster14,Rad))

    &++                                        ((kb*rCBolsterR2*0.25*tanh(vBolsterR2/0.001)*(R-
VectorLength(FindTranslation(Bolster2,Rad)))) &** FindTranslation(Bolster2,Rad))
    &++                                        ((kb*rCBolsterR4*0.25*tanh(vBolsterR4/0.001)*(R-
VectorLength(FindTranslation(Bolster4,Rad)))) &** FindTranslation(Bolster4,Rad))
    &++                                        ((kb*rCBolsterR6*0.25*tanh(vBolsterR6/0.001)*(R-
VectorLength(FindTranslation(Bolster6,Rad)))) &** FindTranslation(Bolster6,Rad))
    &++                                      ((kb*rCBolsterR11*0.25*tanh(vBolsterR11/0.001)*(R-
VectorLength(FindTranslation(Bolster11,Rad)))) &** FindTranslation(Bolster11,Rad))
    &++                                      ((kb*rCBolsterR12*0.25*tanh(vBolsterR12/0.001)*(R-
VectorLength(FindTranslation(Bolster12,Rad)))) &** FindTranslation(Bolster12,Rad))
    &++                                      ((kb*rCBolsterR13*0.25*tanh(vBolsterR13/0.001)*(R-
VectorLength(FindTranslation(Bolster13,Rad)))) &** FindTranslation(Bolster13,Rad))
    &++                                      ((kb*rCBolsterR14*0.25*tanh(vBolsterR14/0.001)*(R-
VectorLength(FindTranslation(Bolster14,Rad)))) &** FindTranslation(Bolster14,Rad))

    &++ NullVector():

t2:=NullVector()
    #&-- (Cdamp &** AngularVelocity(s,bolster))

    #&--  (rBolsterR2  &xx  ((kb*rCBolsterR2*(R-VectorLength(FindTranslation(Bolster2,Rad))))  &**
FindTranslation(Bolster2,Rad)))
    #&--  (rBolsterR4  &xx  ((kb*rCBolsterR4*(R-VectorLength(FindTranslation(Bolster4,Rad))))  &**
FindTranslation(Bolster4,Rad)))
    #&--  (rBolsterR6  &xx  ((kb*rCBolsterR6*(R-VectorLength(FindTranslation(Bolster6,Rad))))  &**
FindTranslation(Bolster6,Rad)))
    #&-- (rBolsterR11 &xx ((kb*rCBolsterR11*(R-VectorLength(FindTranslation(Bolster11,Rad)))) &**
FindTranslation(Bolster11,Rad)))
    #&-- (rBolsterR12 &xx ((kb*rCBolsterR12*(R-VectorLength(FindTranslation(Bolster12,Rad)))) &**
FindTranslation(Bolster12,Rad)))
    #&-- (rBolsterR13 &xx ((kb*rCBolsterR13*(R-VectorLength(FindTranslation(Bolster13,Rad)))) &**
FindTranslation(Bolster13,Rad)))
    #&-- (rBolsterR14 &xx ((kb*rCBolsterR14*(R-VectorLength(FindTranslation(Bolster14,Rad)))) &**
FindTranslation(Bolster14,Rad)))

    #&++         (rBolsterR2         &xx         ((kb*rCBolsterR2*0.25*tanh(vBolsterR2/0.001)*(R-
VectorLength(FindTranslation(Bolster2,Rad)))) &** FindTranslation(Bolster2,Rad)))
    #&++         (rBolsterR4         &xx         ((kb*rCBolsterR4*0.25*tanh(vBolsterR4/0.001)*(R-
VectorLength(FindTranslation(Bolster4,Rad)))) &** FindTranslation(Bolster4,Rad)))
    #&++         (rBolsterR6         &xx         ((kb*rCBolsterR6*0.25*tanh(vBolsterR6/0.001)*(R-
VectorLength(FindTranslation(Bolster6,Rad)))) &** FindTranslation(Bolster6,Rad)))
    #&++        (rBolsterR11        &xx        ((kb*rCBolsterR11*0.25*tanh(vBolsterR11/0.001)*(R-
VectorLength(FindTranslation(Bolster11,Rad)))) &** FindTranslation(Bolster11,Rad)))
    #&++        (rBolsterR12        &xx        ((kb*rCBolsterR12*0.25*tanh(vBolsterR12/0.001)*(R-
VectorLength(FindTranslation(Bolster12,Rad)))) &** FindTranslation(Bolster12,Rad)))
    #&++        (rBolsterR13        &xx        ((kb*rCBolsterR13*0.25*tanh(vBolsterR13/0.001)*(R-
VectorLength(FindTranslation(Bolster13,Rad)))) &** FindTranslation(Bolster13,Rad)))
    #&++        (rBolsterR14        &xx        ((kb*rCBolsterR14*0.25*tanh(vBolsterR14/0.001)*(R-
VectorLength(FindTranslation(Bolster14,Rad)))) &** FindTranslation(Bolster14,Rad)))

    &++ NullVector():

f3:=MakeTranslations(w,0,0,m3*(-32.2))
    &++ ((k1*(Spring31-Spring32)) &** MakeTranslations(s,3))
    &-- (Cdamp &** LinearVelocity(sideframe2,Wedge2_cg))

    &++           ((kb*rCBolsterR2*(R-VectorLength(FindTranslation(Bolster2,Rad))))           &**
FindTranslation(Bolster2,Rad))
    &++           ((kb*rCBolsterR4*(R-VectorLength(FindTranslation(Bolster4,Rad))))           &**
FindTranslation(Bolster4,Rad))
    &++           ((kb*rCBolsterR6*(R-VectorLength(FindTranslation(Bolster6,Rad))))           &**
FindTranslation(Bolster6,Rad))
```

```
    &++          ((kb*rCBolsterR11*(R-VectorLength(FindTranslation(Bolster11,Rad))))          &**
FindTranslation(Bolster11,Rad))
    &++          ((kb*rCBolsterR12*(R-VectorLength(FindTranslation(Bolster12,Rad))))          &**
FindTranslation(Bolster12,Rad))
    &++          ((kb*rCBolsterR13*(R-VectorLength(FindTranslation(Bolster13,Rad))))          &**
FindTranslation(Bolster13,Rad))
    &++          ((kb*rCBolsterR14*(R-VectorLength(FindTranslation(Bolster14,Rad))))          &**
FindTranslation(Bolster14,Rad))

    &--                                     ((kb*rCBolsterR2*0.25*tanh(vBolsterR2/0.001)*(R-
VectorLength(FindTranslation(Bolster2,Rad)))) &** FindTranslation(Bolster2,Rad))
    &--                                     ((kb*rCBolsterR4*0.25*tanh(vBolsterR4/0.001)*(R-
VectorLength(FindTranslation(Bolster4,Rad)))) &** FindTranslation(Bolster4,Rad))
    &--                                     ((kb*rCBolsterR6*0.25*tanh(vBolsterR6/0.001)*(R-
VectorLength(FindTranslation(Bolster6,Rad)))) &** FindTranslation(Bolster6,Rad))
    &--                                   ((kb*rCBolsterR11*0.25*tanh(vBolsterR11/0.001)*(R-
VectorLength(FindTranslation(Bolster11,Rad)))) &** FindTranslation(Bolster11,Rad))
    &--                                   ((kb*rCBolsterR12*0.25*tanh(vBolsterR12/0.001)*(R-
VectorLength(FindTranslation(Bolster12,Rad)))) &** FindTranslation(Bolster12,Rad))
    &--                                   ((kb*rCBolsterR13*0.25*tanh(vBolsterR13/0.001)*(R-
VectorLength(FindTranslation(Bolster13,Rad)))) &** FindTranslation(Bolster13,Rad))
    &--                                   ((kb*rCBolsterR14*0.25*tanh(vBolsterR14/0.001)*(R-
VectorLength(FindTranslation(Bolster14,Rad)))) &** FindTranslation(Bolster14,Rad))

    &--           ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--           ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--           ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))
    &--           ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)           &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1))

    &++ ((ks*rCSideframe2Wedge2_3*0.4*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++ ((ks*rCSideframe2Wedge2_6*0.4*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++ ((ks*rCSideframe2Wedge2_4*0.4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))
    &++ ((ks*rCSideframe2Wedge2_7*0.4*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)
&oo MakeTranslations(s2,1))) &** MakeTranslations(s2,3))

    &++ NullVector():

t3:=NullVector()
    #&-- (Cdamp &** AngularVelocity(s2,wedge2))

    #&++  (rWedge2R2  &xx  ((kb*rCBolsterR2*(R-VectorLength(FindTranslation(Bolster2,Rad))))  &**
FindTranslation(Bolster2,Rad)))
    #&++  (rWedge2R4  &xx  ((kb*rCBolsterR4*(R-VectorLength(FindTranslation(Bolster4,Rad))))  &**
FindTranslation(Bolster4,Rad)))
    #&++  (rWedge2R6  &xx  ((kb*rCBolsterR6*(R-VectorLength(FindTranslation(Bolster6,Rad))))  &**
FindTranslation(Bolster6,Rad)))
    #&++ (rWedge2R11 &xx ((kb*rCBolsterR11*(R-VectorLength(FindTranslation(Bolster11,Rad)))) &**
FindTranslation(Bolster11,Rad)))
    #&++ (rWedge2R12 &xx ((kb*rCBolsterR12*(R-VectorLength(FindTranslation(Bolster12,Rad)))) &**
FindTranslation(Bolster12,Rad)))
    #&++ (rWedge2R13 &xx ((kb*rCBolsterR13*(R-VectorLength(FindTranslation(Bolster13,Rad)))) &**
FindTranslation(Bolster13,Rad)))
    #&++ (rWedge2R14 &xx ((kb*rCBolsterR14*(R-VectorLength(FindTranslation(Bolster14,Rad)))) &**
FindTranslation(Bolster14,Rad)))

    #&--         (rWedge2R2         &xx         ((kb*rCBolsterR2*0.25*tanh(vBolsterR2/0.001)*(R-
VectorLength(FindTranslation(Bolster2,Rad)))) &** FindTranslation(Bolster2,Rad)))
    #&--         (rWedge2R4         &xx         ((kb*rCBolsterR4*0.25*tanh(vBolsterR4/0.001)*(R-
VectorLength(FindTranslation(Bolster4,Rad)))) &** FindTranslation(Bolster4,Rad)))
    #&--         (rWedge2R6         &xx         ((kb*rCBolsterR6*0.25*tanh(vBolsterR6/0.001)*(R-
VectorLength(FindTranslation(Bolster6,Rad)))) &** FindTranslation(Bolster6,Rad)))
    #&--        (rWedge2R11        &xx        ((kb*rCBolsterR11*0.25*tanh(vBolsterR11/0.001)*(R-
VectorLength(FindTranslation(Bolster11,Rad)))) &** FindTranslation(Bolster11,Rad)))
```

```
    #&--        (rWedge2R12        &xx        ((kb*rCBolsterR12*0.25*tanh(vBolsterR12/0.001)*(R-
VectorLength(FindTranslation(Bolster12,Rad)))) &** FindTranslation(Bolster12,Rad)))
    #&--        (rWedge2R13        &xx        ((kb*rCBolsterR13*0.25*tanh(vBolsterR13/0.001)*(R-
VectorLength(FindTranslation(Bolster13,Rad)))) &** FindTranslation(Bolster13,Rad)))
    #&--        (rWedge2R14        &xx        ((kb*rCBolsterR14*0.25*tanh(vBolsterR14/0.001)*(R-
VectorLength(FindTranslation(Bolster14,Rad)))) &** FindTranslation(Bolster14,Rad)))

    #&--   (rWedge2_3   &xx   ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)   &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    #&--   (rWedge2_6   &xx   ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)   &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    #&--   (rWedge2_4   &xx   ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)   &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))
    #&--   (rWedge2_7   &xx   ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)   &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,1)))

    #&++                               (rWedge2_3                                            &xx
((ks*rCSideframe2Wedge2_3*0.4*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)      &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    #&++                               (rWedge2_6                                            &xx
((ks*rCSideframe2Wedge2_6*0.4*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)      &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    #&++                               (rWedge2_4                                            &xx
((ks*rCSideframe2Wedge2_4*0.4*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)      &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))
    #&++                               (rWedge2_7                                            &xx
((ks*rCSideframe2Wedge2_7*0.4*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)      &oo
MakeTranslations(s2,1))) &** MakeTranslations(s2,3)))

    &++ NullVector():

#Puts all the forces in a vector
p:=subs(kde,MomentumDescription([p1,h1,p2,h2,p3,h3])):
appliedforce:=ForceDescription([f1,t1,f2,t2,f3,t3]):

eqs:=((subs(kde,DiffTime(p,w)) &-- appliedforce) &oo beta):
diffeqs:=kde union convert(eqs,set):


#Create bodies
DefineObjects([bolster,'Block',point=MakeTranslations(bolster,0,0,(6.125+5)/2),xlength=4,ylength=
12,zlength=6.125+5,color=red],

[bolster,'Block',point=MakeTranslations(bolster,0,0,6.125+2.5),xlength=4+2*(6.125*tan(37.5*3.1415
927/180)),ylength=12,zlength=5,color=red],
              [bolster3,'Block',point=MakeTranslations(bolster3,-
2,0,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2))/2),xlength=4,ylength=12,zlength=sqrt((6.125
*tan(37.5*3.1415927/180))^2+6.125^2),color=red],

[bolster4,'Block',point=MakeTranslations(bolster4,2,0,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.1
25^2))/2),xlength=4,ylength=12,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2),color=red]
,


[wedge1,'Block',point=Wedge1_1,xlength=(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,
color=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlengt
h=0.05,ylength=10.5,zlength=6.125,color=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,0,0,6.125/2),orient=MakeRotations([37.5*3.1415927/1
80,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2),color=bl
ue],

              [wedge2,'Block',point=Wedge2_1,xlength=-
(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,-
(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlength=0.05,ylength=10.5,zlength=6.125,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,0,0,6.125/2),orient=MakeRotations([-
37.5*3.1415927/180,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6
.125^2),color=blue],
```

```
[sideframe,'Cylinder',point=MakeTranslations(s,(6.125*tan(37.5*3.1415927/180))/2+2,0,(VectorLengt
h(FindTranslation(Spring1_1,Spring1_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring
1_1,Spring1_2)),color=green],

[sideframe,'Cylinder',point=MakeTranslations(s,0,0,(VectorLength(FindTranslation(Spring2_1,Spring
2_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring2_1,Spring2_2)),color=green],
              [sideframe,'Cylinder',point=MakeTranslations(s,-(6.125*tan(37.5*3.1415927/180))/2-
2,0,(VectorLength(FindTranslation(Spring3_1,Spring3_2)))/2),radius=0.5,length=VectorLength(FindTr
anslation(Spring3_1,Spring3_2)),color=green],


[sideframe,'Block',point=S,xlength=4+2*(6.125*tan(37.5*3.1415927/180)),ylength=12,zlength=0.01,co
lor=white],

[sideframe1,'Block',point=MakeTranslations(s1,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],

[sideframe2,'Block',point=MakeTranslations(s2,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],


[wedge2,'Cylinder',point=Rad,orient=MakeRotations(90*Pi/180,1),radius=R,length=10.5,color="{0.5,0
.5,0.5}"],

#Visual placement of points
              [world,'Sphere',point=W,radius=0.05,color=green],

              [sideframe,'Sphere',point=Sideframe1,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe2,radius=0.05,color=white],
              [sideframe,'Sphere',point=Sideframe3,radius=0.05,color=red],
              [sideframe,'Sphere',point=Sideframe4,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster_cg,radius=0.05,color=green],
              [bolster,'Sphere',point=Bolster1,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster2,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster3,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster4,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster5,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Bolster6,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster7,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster8,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster9,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster10,radius=0.05,color=blue],
              [bolster,'Sphere',point=Bolster11,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster12,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster13,radius=0.05,color=red],
              [bolster,'Sphere',point=Bolster14,radius=0.05,color=red],

              [wedge1,'Sphere',point=Wedge1_1,radius=0.05,color=yellow],
              [wedge1,'Sphere',point=Wedge1_2,radius=0.05,color=red],
              [wedge1,'Sphere',point=Wedge1_3,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_4,radius=0.05,color=blue],
              [wedge1,'Sphere',point=Wedge1_5,radius=0.05,color=red],
              [wedge1,'Sphere',point=Wedge1_6,radius=0.05,color=green],
              [wedge1,'Sphere',point=Wedge1_7,radius=0.05,color=blue],
              [wedge1,'Sphere',point=Wedge1_cg,radius=0.05,color=green],

              [wedge2,'Sphere',point=Wedge2_1,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Wedge2_2,radius=0.05,color=red],
              [wedge2,'Sphere',point=Wedge2_3,radius=0.05,color=green],
              [wedge2,'Sphere',point=Wedge2_4,radius=0.05,color=blue],
              [wedge2,'Sphere',point=Wedge2_5,radius=0.05,color=red],
              [wedge2,'Sphere',point=Wedge2_6,radius=0.05,color=green],
              [wedge2,'Sphere',point=Wedge2_7,radius=0.05,color=blue],
              [wedge2,'Sphere',point=Wedge2_cg,radius=0.05,color=green],
              [wedge2,'Sphere',point=Wedge2_8,radius=0.05,color=white],
              [wedge2,'Sphere',point=Wedge2_9,radius=0.05,color=white],
              [wedge2,'Sphere',point=Wedge2_10,radius=0.05,color=white],
```

```
              [wedge1,'Sphere',point=Spring1_1,radius=0.05,color=yellow],
              [wedge1,'Sphere',point=Spring1_2,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Spring2_1,radius=0.05,color=yellow],
              [bolster,'Sphere',point=Spring2_2,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Spring3_1,radius=0.05,color=yellow],
              [wedge2,'Sphere',point=Spring3_2,radius=0.05,color=yellow],

              [wedge2,'Sphere',point=Rad,radius=0.1,color=blue]):
#Outputs
GeometryOutput(main=world,
        parameters=[toe=0,omega=0,Fext=0,ks=100000,kb1=100000,Cdamp=10,X=-7.3,Z=-
2.53,R=10,m1=0.875,m2=234,m3=0.875,k1=1491,k2=11208.9,psi=0],
       states=[q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,u1,u2,u3,u4,u5,u6,u7,u8,u9],
        filename="newwedge_ver2_1.geo"):
MotionOutput(ode=diffeqs,
        parameters=[toe=0,omega=0,Fext=0,ks=100000,kb1=100000,Cdamp=10,X=-7.3,Z=-
2.53,R=10,m1=0.875,m2=234,m3=0.875,k1=1491,k2=11208.9,psi=0],
        states=[q1=4.350302,q2=9.668228,q3=9.789107,q4=-
4.359987,q5=8.445349,q6,q7,q8,q9,q10,u1,u2,u3,u4,u5,u6,u7,u8,u9],
        insignals=[rCSideframe1Wedge1_3=((FindTranslation(Sideframe1,Wedge1_3)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_4=((FindTranslation(Sideframe1,Wedge1_4)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_6=((FindTranslation(Sideframe1,Wedge1_6)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_7=((FindTranslation(Sideframe1,Wedge1_7)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe2Wedge2_3=((FindTranslation(Sideframe2,Wedge2_3)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_4=((FindTranslation(Sideframe2,Wedge2_4)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_6=((FindTranslation(Sideframe2,Wedge2_6)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_7=((FindTranslation(Sideframe2,Wedge2_7)                &oo
MakeTranslations(s2,1))&<0),
                   rCBolsterR2=(VectorLength(FindTranslation(Bolster2,Rad))&<R),
                   rCBolsterR4=(VectorLength(FindTranslation(Bolster4,Rad))&<R),
                   rCBolsterR6=(VectorLength(FindTranslation(Bolster6,Rad))&<R),
                   rCBolsterR11=(VectorLength(FindTranslation(Bolster11,Rad))&<R),
                   rCBolsterR12=(VectorLength(FindTranslation(Bolster12,Rad))&<R),
                   rCBolsterR13=(VectorLength(FindTranslation(Bolster13,Rad))&<R),
                   rCBolsterR14=(VectorLength(FindTranslation(Bolster14,Rad))&<R)],
        filename="newwedge_ver2_1.dyn"):
```

7.1.4.2. Quarter-truck Variably Damped Model With Wedge Face Curvature State Calculation M-file

```
global Fext m1 m2 m3 ks kb omega Cdamp toe X Z R k1 k2 psi

%Parameters
toe=0;
omega=4.5;
Fext=1;
ks=100000;
kb=100000;
Cdamp=10;
X=-7.3;
Z=-2.53;
R=10;
m1=0.875;
m2=234;
m3=0.875;
k1=1491;
k2=11000;
psi=0;

q1=y(:,1);
q2=y(:,2);
q3=y(:,3);
q4=y(:,4);
q5=y(:,5);
q6=y(:,6);
q7=y(:,7);
q8=y(:,8);
q9=y(:,9);
q10=y(:,10);
u1=y(:,11);
u2=y(:,12);
u3=y(:,13);
u4=y(:,14);
u5=y(:,15);
u6=y(:,16);
u7=y(:,17);
u8=y(:,18);
u9=y(:,19);



for i = 1:length(t),

if                              (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.349938947*cos(q9(i))*cos(q8(i))*cos(toe)-2.349938947*sin(q9(i))*sin(toe)-
5.250000000*sin(q8(i))*cos(toe))<0),
        rCSideframe2Wedge2_3(i) = 1;
    else
        rCSideframe2Wedge2_3(i) = 0;
    end

if                              (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.349938947*cos(q9(i))*cos(q8(i))*cos(toe)-2.349938947*sin(q9(i))*sin(toe)-
5.250000000*sin(q8(i))*cos(toe)+6.125*sin(q9(i))*cos(q8(i))*cos(toe)-
6.125*cos(q9(i))*sin(toe))<0),
        rCSideframe2Wedge2_4(i) = 1;
    else
        rCSideframe2Wedge2_4(i) = 0;
    end

if                              (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.349938947*cos(q9(i))*cos(q8(i))*cos(toe)-
2.349938947*sin(q9(i))*sin(toe)+5.250000000*sin(q8(i))*cos(toe))<0),
        rCSideframe2Wedge2_6(i) = 1;
```

```
    else
        rCSideframe2Wedge2_6(i) = 0;
    end

if                    (((6.699877893-15/2*tan(toe)+q4(i))*cos(toe)-(-15/2+q5(i))*sin(toe)-
2.349938947*cos(q9(i))*cos(q8(i))*cos(toe)-
2.349938947*sin(q9(i))*sin(toe)+5.250000000*sin(q8(i))*cos(toe)+6.125*sin(q9(i))*cos(q8(i))*cos(t
oe)-6.125*cos(q9(i))*sin(toe))<0),
        rCSideframe2Wedge2_7(i) = 1;
    else
        rCSideframe2Wedge2_7(i) = 0;
    end

if                                              (((21.29457671+7.827000000*cos(psi)*q4(i)-
4.890437388*q5(i)+4.890437388*q3(i)+3.913500000*(cos(q9(i))*cos(q8(i))*cos(psi)+cos(q9(i))*sin(q8
(i))*sin(psi))*X+3.913500000*(sin(q9(i))*cos(q8(i))*cos(psi)+sin(q9(i))*sin(q8(i))*sin(psi))*Z+2.
445218694*sin(q9(i))*X-2.445218694*cos(q9(i))*Z+q4(i)^2+(q5(i)-
q3(i))^2+q4(i)*(cos(q9(i))*cos(q8(i))*X+sin(q9(i))*cos(q8(i))*Z)+(q5(i)-q3(i))*(-
sin(q9(i))*X+cos(q9(i))*Z)+X*(3.913500000*cos(q9(i))*cos(q8(i))*cos(psi)+3.913500000*cos(q9(i))*s
in(q8(i))*sin(psi)+2.445218694*sin(q9(i)))+Z*(3.913500000*sin(q9(i))*cos(q8(i))*cos(psi)+3.913500
000*sin(q9(i))*sin(q8(i))*sin(psi)-2.445218694*cos(q9(i)))+X*(cos(q9(i))*cos(q8(i))*q4(i)-
sin(q9(i))*(q5(i)-q3(i)))+Z*(sin(q9(i))*cos(q8(i))*q4(i)+cos(q9(i))*(q5(i)-
q3(i)))+X^2+Z^2)^(1/2))<R),
        rCBolsterR2(i) = 1;
    else
        rCBolsterR2(i) = 0;
    end

if
(((4+4*cos(psi)*q4(i)+2*(cos(q9(i))*cos(q8(i))*cos(psi)+cos(q9(i))*sin(q8(i))*sin(psi))*X+2*(sin(
q9(i))*cos(q8(i))*cos(psi)+sin(q9(i))*sin(q8(i))*sin(psi))*Z+q4(i)^2+(q5(i)-
q3(i))^2+q4(i)*(cos(q9(i))*cos(q8(i))*X+sin(q9(i))*cos(q8(i))*Z)+(q5(i)-q3(i))*(-
sin(q9(i))*X+cos(q9(i))*Z)+X*(2*cos(q9(i))*cos(q8(i))*cos(psi)+2*cos(q9(i))*sin(q8(i))*sin(psi))+
Z*(2*sin(q9(i))*cos(q8(i))*cos(psi)+2*sin(q9(i))*sin(q8(i))*sin(psi))+X*(cos(q9(i))*cos(q8(i))*q4
(i)-sin(q9(i))*(q5(i)-q3(i)))+Z*(sin(q9(i))*cos(q8(i))*q4(i)+cos(q9(i))*(q5(i)-
q3(i)))+X^2+Z^2)^(1/2))<R),
        rCBolsterR4(i) = 1;
    else
        rCBolsterR4(i) = 0;
    end

if                                              (((82.40398878+13.39975579*cos(psi)*q4(i)-
12.250*q5(i)+12.250*q3(i)+6.699877893*(cos(q9(i))*cos(q8(i))*cos(psi)+cos(q9(i))*sin(q8(i))*sin(p
si))*X+6.699877893*(sin(q9(i))*cos(q8(i))*cos(psi)+sin(q9(i))*sin(q8(i))*sin(psi))*Z+6.125*sin(q9
(i))*X-6.125*cos(q9(i))*Z+q4(i)^2+(q5(i)-
q3(i))^2+q4(i)*(cos(q9(i))*cos(q8(i))*X+sin(q9(i))*cos(q8(i))*Z)+(q5(i)-q3(i))*(-
sin(q9(i))*X+cos(q9(i))*Z)+X*(6.699877893*cos(q9(i))*cos(q8(i))*cos(psi)+6.699877893*cos(q9(i))*s
in(q8(i))*sin(psi)+6.125*sin(q9(i)))+Z*(6.699877893*sin(q9(i))*cos(q8(i))*cos(psi)+6.699877893*si
n(q9(i))*sin(q8(i))*sin(psi)-6.125*cos(q9(i)))+X*(cos(q9(i))*cos(q8(i))*q4(i)-sin(q9(i))*(q5(i)-
q3(i)))+Z*(sin(q9(i))*cos(q8(i))*q4(i)+cos(q9(i))*(q5(i)-q3(i)))+X^2+Z^2)^(1/2))<R),
        rCBolsterR6(i) = 1;
    else
        rCBolsterR6(i) = 0;
    end

if
(((31.5625+2*cos(psi)*q4(i)+5.25*sin(psi)*q4(i)+2*(cos(q9(i))*cos(q8(i))*cos(psi)+cos(q9(i))*sin(
q8(i))*sin(psi))*X+2*(sin(q9(i))*cos(q8(i))*cos(psi)+sin(q9(i))*sin(q8(i))*sin(psi))*Z-5.25*(-
cos(q9(i))*cos(q8(i))*sin(psi)+cos(q9(i))*sin(q8(i))*cos(psi))*X-5.25*(-
sin(q9(i))*cos(q8(i))*sin(psi)+sin(q9(i))*sin(q8(i))*cos(psi))*Z+q4(i)*(2*cos(psi)+5.25*sin(psi))
+q4(i)^2+(q5(i)-q3(i))^2+q4(i)*(cos(q9(i))*cos(q8(i))*X+sin(q9(i))*cos(q8(i))*Z)+(q5(i)-q3(i))*(-
sin(q9(i))*X+cos(q9(i))*Z)+X*(2*cos(q9(i))*cos(q8(i))*cos(psi)+2*cos(q9(i))*sin(q8(i))*sin(psi)+5
.25*cos(q9(i))*cos(q8(i))*sin(psi)-
5.25*cos(q9(i))*sin(q8(i))*cos(psi))+Z*(2*sin(q9(i))*cos(q8(i))*cos(psi)+2*sin(q9(i))*sin(q8(i))*
sin(psi)+5.25*sin(q9(i))*cos(q8(i))*sin(psi)-
5.25*sin(q9(i))*sin(q8(i))*cos(psi))+X*(cos(q9(i))*cos(q8(i))*q4(i)-sin(q9(i))*(q5(i)-
q3(i)))+Z*(sin(q9(i))*cos(q8(i))*q4(i)+cos(q9(i))*(q5(i)-q3(i)))+X^2+Z^2)^(1/2))<R),
        rCBolsterR11(i) = 1;
    else
        rCBolsterR11(i) = 0;
    end
```

```
if (((31.5625+2*cos(psi)*q4(i)-
5.25*sin(psi)*q4(i)+2*(cos(q9(i))*cos(q8(i))*cos(psi)+cos(q9(i))*sin(q8(i))*sin(psi))*X+2*(sin(q9
(i))*cos(q8(i))*cos(psi)+sin(q9(i))*sin(q8(i))*sin(psi))*Z+5.25*(-
cos(q9(i))*cos(q8(i))*sin(psi)+cos(q9(i))*sin(q8(i))*cos(psi))*X+5.25*(-
sin(q9(i))*cos(q8(i))*sin(psi)+sin(q9(i))*sin(q8(i))*cos(psi))*Z+q4(i)*(2*cos(psi)-
5.25*sin(psi))+q4(i)^2+(q5(i)-
q3(i))^2+q4(i)*(cos(q9(i))*cos(q8(i))*X+sin(q9(i))*cos(q8(i))*Z)+(q5(i)-q3(i))*(-
sin(q9(i))*X+cos(q9(i))*Z)+X*(2*cos(q9(i))*cos(q8(i))*cos(psi)+2*cos(q9(i))*sin(q8(i))*sin(psi)-
5.25*cos(q9(i))*cos(q8(i))*sin(psi)+5.25*cos(q9(i))*sin(q8(i))*cos(psi))+Z*(2*sin(q9(i))*cos(q8(i
))*cos(psi)+2*sin(q9(i))*sin(q8(i))*sin(psi)-
5.25*sin(q9(i))*cos(q8(i))*sin(psi)+5.25*sin(q9(i))*sin(q8(i))*cos(psi))+X*(cos(q9(i))*cos(q8(i))
*q4(i)-sin(q9(i))*(q5(i)-q3(i)))+Z*(sin(q9(i))*cos(q8(i))*q4(i)+cos(q9(i))*(q5(i)-
q3(i)))+X^2+Z^2)^(1/2))<R),
        rCBolsterR12(i) = 1;
    else
        rCBolsterR12(i) = 0;
    end

if (((109.9664888+6.699877893*cos(psi)*q4(i)+5.25*sin(psi)*q4(i)-
12.250*q5(i)+12.250*q3(i)+6.699877893*(cos(q9(i))*cos(q8(i))*cos(psi)+cos(q9(i))*sin(q8(i))*sin(p
si))*X+6.699877893*(sin(q9(i))*cos(q8(i))*cos(psi)+sin(q9(i))*sin(q8(i))*sin(psi))*Z-5.25*(-
cos(q9(i))*cos(q8(i))*sin(psi)+cos(q9(i))*sin(q8(i))*cos(psi))*X-5.25*(-
sin(q9(i))*cos(q8(i))*sin(psi)+sin(q9(i))*sin(q8(i))*cos(psi))*Z+6.125*sin(q9(i))*X-
6.125*cos(q9(i))*Z+q4(i)*(6.699877893*cos(psi)+5.25*sin(psi))+q4(i)^2+(q5(i)-
q3(i))^2+q4(i)*(cos(q9(i))*cos(q8(i))*X+sin(q9(i))*cos(q8(i))*Z)+(q5(i)-q3(i))*(-
sin(q9(i))*X+cos(q9(i))*Z)+X*(6.699877893*cos(q9(i))*cos(q8(i))*cos(psi)+6.699877893*cos(q9(i))*s
in(q8(i))*sin(psi)+5.25*cos(q9(i))*cos(q8(i))*sin(psi)-
5.25*cos(q9(i))*sin(q8(i))*cos(psi)+6.125*sin(q9(i)))+Z*(6.699877893*sin(q9(i))*cos(q8(i))*cos(ps
i)+6.699877893*sin(q9(i))*sin(q8(i))*sin(psi)+5.25*sin(q9(i))*cos(q8(i))*sin(psi)-
5.25*sin(q9(i))*sin(q8(i))*cos(psi)-6.125*cos(q9(i)))+X*(cos(q9(i))*cos(q8(i))*q4(i)-
sin(q9(i))*(q5(i)-q3(i)))+Z*(sin(q9(i))*cos(q8(i))*q4(i)+cos(q9(i))*(q5(i)-
q3(i)))+X^2+Z^2)^(1/2))<R),
        rCBolsterR13(i) = 1;
    else
        rCBolsterR13(i) = 0;
    end

if (((109.9664888+6.699877893*cos(psi)*q4(i)-5.25*sin(psi)*q4(i)-
12.250*q5(i)+12.250*q3(i)+6.699877893*(cos(q9(i))*cos(q8(i))*cos(psi)+cos(q9(i))*sin(q8(i))*sin(p
si))*X+6.699877893*(sin(q9(i))*cos(q8(i))*cos(psi)+sin(q9(i))*sin(q8(i))*sin(psi))*Z+5.25*(-
cos(q9(i))*cos(q8(i))*sin(psi)+cos(q9(i))*sin(q8(i))*cos(psi))*X+5.25*(-
sin(q9(i))*cos(q8(i))*sin(psi)+sin(q9(i))*sin(q8(i))*cos(psi))*Z+6.125*sin(q9(i))*X-
6.125*cos(q9(i))*Z+q4(i)*(6.699877893*cos(psi)-5.25*sin(psi))+q4(i)^2+(q5(i)-
q3(i))^2+q4(i)*(cos(q9(i))*cos(q8(i))*X+sin(q9(i))*cos(q8(i))*Z)+(q5(i)-q3(i))*(-
sin(q9(i))*X+cos(q9(i))*Z)+X*(6.699877893*cos(q9(i))*cos(q8(i))*cos(psi)+6.699877893*cos(q9(i))*s
in(q8(i))*sin(psi)-
5.25*cos(q9(i))*cos(q8(i))*sin(psi)+5.25*cos(q9(i))*sin(q8(i))*cos(psi)+6.125*sin(q9(i)))+Z*(6.69
9877893*sin(q9(i))*cos(q8(i))*cos(psi)+6.699877893*sin(q9(i))*sin(q8(i))*sin(psi)-
5.25*sin(q9(i))*cos(q8(i))*sin(psi)+5.25*sin(q9(i))*sin(q8(i))*cos(psi)-
6.125*cos(q9(i)))+X*(cos(q9(i))*cos(q8(i))*q4(i)-sin(q9(i))*(q5(i)-
q3(i)))+Z*(sin(q9(i))*cos(q8(i))*q4(i)+cos(q9(i))*(q5(i)-q3(i)))+X^2+Z^2)^(1/2))<R),
        rCBolsterR14(i) = 1;
    else
        rCBolsterR14(i) = 0;
    end
```

7.1.5. Full-truck Variably Damped Model SRC file

```
with(Mambo):

#Declare observers, points, and triads
DeclareObservers(world,bolster,bolster3,bolster4,wedge1,wedge2,sideframe,sideframe1,sideframe2,si
deframe21,sideframe22,sideframe23,wedge21,wedge22):
DeclarePoints(W,Bolster,Bolster1,Bolster2,Bolster_cg,Bolster3,Bolster4,Bolster5,Bolster6,Bolster7
,Bolster8,Bolster9,Bolster10,

Bolster11,Bolster12,Bolster13,Bolster14,Wedge1_1,Wedge1_2,Wedge1_3,Wedge1_4,Wedge1_5,Wedge1_6,Wed
ge1_7,Wedge1_cg,

Wedge2_1,Wedge2_2,Wedge2_3,Wedge2_4,Wedge2_5,Wedge2_6,Wedge2_7,Wedge2_cg,Bolsterl,Bolster21,Bolst
er22,Bolster23,Bolster24,Bolster25,Bolster26,Bolster27,Bolster28,Bolster29,Bolster210,
              Bolster211,Bolster212,Bolster213,Bolster214,Bolsterr,

Spring1_1,Spring1_2,Spring2_1,Spring2_2,Spring3_1,Spring3_2,S,Sideframe1,Sideframe2,Sideframe3,Si
deframe4,

Spring21_1,Spring21_2,Spring22_1,Spring22_2,Spring23_1,Spring23_2,S21,Sideframe22,Sideframe23,Sid
eframe24,Sideframe25,
              Wedge21_1,Wedge21_2,Wedge21_3,Wedge21_4,Wedge21_5,Wedge21_6,Wedge21_7,Wedge21_cg,
              Wedge22_1,Wedge22_2,Wedge22_3,Wedge22_4,Wedge22_5,Wedge22_6,Wedge22_7,Wedge22_cg):
DeclareTriads(w,bolster,bolster1,wedge1,wedge2,s,s1,s2,bolster3,bolster4,s21,s22,s23,wedge21,wedg
e22):


#Define observers, points, triads, and neighbors
DefineObservers([world,W,w],[bolster,Bolster,bolster],[wedge1,Wedge1_1,wedge1],[wedge2,Wedge2_1,w
edge2],

[sideframe,S,s],[bolster3,Bolster3,bolster3],[bolster4,Bolster4,bolster4],[sideframe1,Sideframe3,
s1],

[sideframe2,Sideframe4,s2],[sideframe21,S21,s21],[sideframe22,Sideframe24,s22],[sideframe23,Sidef
rame25,s23],
                [wedge21,Wedge21_1,wedge21],[wedge22,Wedge22_1,wedge22]):
DefinePoints([S,Wedge1_1,s,q1,0,q2],
            [Wedge1_1,Wedge1_2,wedge1,-(z2*tan(theta*Pi/180))/2,z3/2,0],
            [Wedge1_2,Wedge1_3,wedge1,(z2*tan(theta*Pi/180)),0,0],
            [Wedge1_3,Wedge1_4,wedge1,0,0,z2],
            [Wedge1_2,Wedge1_5,wedge1,0,-z3,0],
            [Wedge1_3,Wedge1_6,wedge1,0,-z3,0],
            [Wedge1_4,Wedge1_7,wedge1,0,-z3,0],
            [Wedge1_1,Wedge1_cg,wedge1,(1/2)*(z2*tan(theta*Pi/180))-
(1/3)*(z2*tan(theta*Pi/180)),0,z2*(1/3)],
            [Wedge1_1,Spring1_1,wedge1,0,0,0],

            [S21,Wedge21_1,s21,q3,0,q4],
            [Wedge21_1,Wedge21_2,wedge21,-(6.125*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge21_2,Wedge21_3,wedge21,(6.125*tan(37.5*3.1415927/180)),0,0],
            [Wedge21_3,Wedge21_4,wedge21,0,0,6.125],
            [Wedge21_2,Wedge21_5,wedge21,0,-10.5,0],
            [Wedge21_3,Wedge21_6,wedge21,0,-10.5,0],
            [Wedge21_4,Wedge21_7,wedge21,0,-10.5,0],
            [Wedge21_1,Wedge21_cg,wedge21,(1/2)*(6.125*tan(37.5*3.1415927/180))-
(1/3)*(6.125*tan(37.5*3.1415927/180)),0,6.125*(1/3)],
            [Wedge21_1,Spring21_1,wedge21,0,0,0],

            [W,Bolster,w,0,0,q5],
            [Bolster,Bolsterl,bolster,0,-(78/2)+6,0],
            [Bolster,Bolster_cg,bolster,0,0,6.29],
            [Bolsterl,Spring2_1,bolster,0,0,0],
            [Bolsterl,Bolster1,bolster,2+z1/2,0,(z1/z2)*(2+z1/2)],
            [Bolsterl,Bolster2,bolster,-2-z1/2,0,(z1/z2)*(2+z1/2)],
            [Bolsterl,Bolster3,bolster,2,0,0],
            [Bolsterl,Bolster4,bolster,-2,0,0],
```

```
            [Bolsterl,Bolster5,bolster,(4+2*(z2*tan(theta*Pi/180)))/2,0,z2],
            [Bolsterl,Bolster6,bolster,-(4+2*(z2*tan(theta*Pi/180)))/2,0,z2],
            [Bolsterl,Bolster7,bolster,2,6,0],
            [Bolsterl,Bolster8,bolster,2,-6,0],
            [Bolsterl,Bolster9,bolster,(4+2*(z2*tan(theta*Pi/180)))/2,6,z2],
            [Bolsterl,Bolster10,bolster,(4+2*(z2*tan(theta*Pi/180)))/2,-6,z2],
            [Bolsterl,Bolster11,bolster,-2,6,0],
            [Bolsterl,Bolster12,bolster,-2,-6,0],
            [Bolsterl,Bolster13,bolster,-(4+2*(z2*tan(theta*Pi/180)))/2,6,z2],
            [Bolsterl,Bolster14,bolster,-(4+2*(z2*tan(theta*Pi/180)))/2,-6,z2],

            [Bolster,Bolsterr,bolster,0,(78/2)-6,0],
            [Bolsterr,Spring22_1,bolster,0,0,0],
            [Bolsterr,Bolster21,bolster,2+3.827/2,0,(3.827/6.125)*(2+3.827/2)],
            [Bolsterr,Bolster22,bolster,-2-3.827/2,0,(3.827/6.125)*(2+3.827/2)],
            [Bolsterr,Bolster23,bolster,2,0,0],
            [Bolsterr,Bolster24,bolster,-2,0,0],
            [Bolsterr,Bolster25,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,0,6.125],
            [Bolsterr,Bolster26,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,0,6.125],
            [Bolsterr,Bolster27,bolster,2,6,0],
            [Bolsterr,Bolster28,bolster,2,-6,0],
            [Bolsterr,Bolster29,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,6,6.125],
            [Bolsterr,Bolster210,bolster,(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-6,6.125],
            [Bolsterr,Bolster211,bolster,-2,6,0],
            [Bolsterr,Bolster212,bolster,-2,-6,0],
            [Bolsterr,Bolster213,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,6,6.125],
            [Bolsterr,Bolster214,bolster,-(4+2*(6.125*tan(37.5*3.1415927/180)))/2,-6,6.125],

            [S,Wedge2_1,s,q6,0,q7],
            [Wedge2_1,Wedge2_2,wedge2,(z2*tan(theta*Pi/180))/2,z3/2,0],
            [Wedge2_2,Wedge2_3,wedge2,-(z2*tan(theta*Pi/180)),0,0],
            [Wedge2_3,Wedge2_4,wedge2,0,0,z2],
            [Wedge2_2,Wedge2_5,wedge2,0,-z3,0],
            [Wedge2_3,Wedge2_6,wedge2,0,-z3,0],
            [Wedge2_4,Wedge2_7,wedge2,0,-z3,0],
            [Wedge2_1,Wedge2_cg,wedge2,-
(1/2)*(z2*tan(theta*Pi/180))+(1/3)*(z2*tan(theta*Pi/180)),0,z2*(1/3)],
            [Wedge2_1,Spring3_1,wedge2,0,0,0],

            [S21,Wedge22_1,s21,q8,0,q9],
            [Wedge22_1,Wedge22_2,wedge22,(6.125*tan(37.5*3.1415927/180))/2,10.5/2,0],
            [Wedge22_2,Wedge22_3,wedge22,-(6.125*tan(37.5*3.1415927/180)),0,0],
            [Wedge22_3,Wedge22_4,wedge22,0,0,6.125],
            [Wedge22_2,Wedge22_5,wedge22,0,-10.5,0],
            [Wedge22_3,Wedge22_6,wedge22,0,-10.5,0],
            [Wedge22_4,Wedge22_7,wedge22,0,-10.5,0],
            [Wedge22_1,Wedge22_cg,wedge22,-
(1/2)*(6.125*tan(37.5*3.1415927/180))+(1/3)*(6.125*tan(37.5*3.1415927/180)),0,6.125*(1/3)],
            [Wedge22_1,Spring23_1,wedge22,0,0,0],

            [S,Spring1_2,s,(z2*tan(theta*Pi/180))/2+2,0,0],
            [S,Spring2_2,s,0,0,0],
            [S,Spring3_2,s,-(z2*tan(theta*Pi/180))/2-2,0,0],
            [S,Sideframe1,s,2+(z2*tan(theta*Pi/180)),0,15/2],
            [S,Sideframe2,s,-2-(z2*tan(theta*Pi/180)),0,15/2],
            [S,Sideframe3,s,(4+2*(z2*tan(theta*Pi/180)))/2,0,0],
            [S,Sideframe4,s,(-4-2*(z2*tan(theta*Pi/180)))/2,0,0],

            [S21,Spring21_2,s21,(6.125*tan(37.5*3.1415927/180))/2+2,0,0],
            [S21,Spring22_2,s21,0,0,0],
            [S21,Spring23_2,s21,-(6.125*tan(37.5*3.1415927/180))/2-2,0,0],
            [S21,Sideframe22,s21,2+(-15/2*tan(toe))+(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S21,Sideframe23,s21,-2-(-15/2*tan(toe))-(6.125*tan(37.5*3.1415927/180)),0,15/2],
            [S21,Sideframe24,s21,(2+(6.125*tan(37.5*3.1415927/180))),0,0],
            [S21,Sideframe25,s21,(-2-(6.125*tan(37.5*3.1415927/180))),0,0],

            [W,S,w,0,-(78/2)+6,q20],
            [W,S21,w,0,(78/2)-6,q21]):

DefineTriads([w,s,[0,1]],
             [s,s1,[-toe,2]],
```

```
            [s,s2,[toe,2]],
            [s,wedge1,[q10,3],[q11,2]],
            [w,bolster,[q13,3],[q20,1]],
            [s,wedge2,[q14,3],[q15,2]],
            [bolster,bolster3,[(theta*Pi/180),2]],
            [bolster,bolster4,[(-theta*Pi/180),2]],
            [w,s21,[0,1]],
            [s21,s22,[-toe,2]],
            [s21,s23,[toe,2]],
            [s21,wedge21,[q16,3],[q17,2]],
            [s21,wedge22,[q18,3],[q19,2]]]):
DefineNeighbors([world,sideframe],[world,bolster],[world,sideframe21],[sideframe,wedge1],[sidefra
me,wedge2],[sideframe,sideframe1],[sideframe,sideframe2],

[bolster,bolster3],[bolster,bolster4],[sideframe21,sideframe22],[sideframe21,sideframe23],[sidefr
ame21,wedge21],[sideframe,wedge22]):

#Declares the independent velocities as states
DeclareStates(q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q13,q14,q15,q16,q17,q18,q19,q20,q21,q22,u1,u2,u3
,u4,u5,u6,u7,u8,u9,u10,u11,u13,u14,u15,u16,u17,u18,u19,u20):

#Define the kinematic differential equations
kde:={q1t=u1,q2t=u2,q3t=u3,q4t=u4,q5t=u5,q6t=u6,q7t=u7,q8t=u8,q9t=u9,q10t=u10,q11t=u11,q13t=u13,q
14t=u14,q15t=u15,q16t=u16,q17t=u17,q18t=u18,q19t=u19,q20t=u20,q21t=Fext*omega*cos(omega*t),q22t=F
ext*omega*cos(omega*t)}:
kde:=simplify(solve(kde,{q1t,q2t,q3t,q4t,q5t,q6t,q7t,q8t,q9t,q10t,q11t,q13t,q14t,q15t,q16t,q17t,q
18t,q19t,q20t,q21t})):

#Calculate the matrix of vectors which span the subspace of allowable motions, beta
v:=subs(kde,VelocityDescription([LinearVelocity(world,Wedge1_cg),AngularVelocity(w,wedge1),
                                LinearVelocity(world,Bolster_cg),AngularVelocity(w,bolster),
                                LinearVelocity(world,Wedge2_cg),AngularVelocity(w,wedge2),
                                LinearVelocity(world,Wedge21_cg),AngularVelocity(w,wedge21),
                                LinearVelocity(world,Wedge22_cg),AngularVelocity(w,wedge22)])):
beta:=CoeffExtract(v,[u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u13,u14,u15,u16,u17,u18,u19,u20]):

#Inertial forces
p1:=LinearMomentum(world,Wedge1_cg,m1):
h1:=AngularMomentum(world,wedge1,matrix([[((1/12)*m1*(z3^2+z2^2)),0,0],[0,((1/12)*m1*(z1^2+z2^2))
,0],[0,0,((1/12)*m1*(z1^2+z3^2))]])):
p2:=LinearMomentum(world,Bolster_cg,m2):
h2:=AngularMomentum(world,bolster,matrix([[((1/12)*m2*((12^2)+((z2+5)^2))),0,0],[0,((1/12)*m2*((4
+2*(z2*tan(theta*Pi/180))^2)+((z2+5)^2))),0],[0,0,((1/12)*m2*((4+2*(z2*tan(theta*Pi/180))^2)+(12^
2)))]])):
p3:=LinearMomentum(world,Wedge2_cg,m3):
h3:=AngularMomentum(world,wedge2,matrix([[((1/12)*m3*(z3^2+z2^2)),0,0],[0,((1/12)*m3*(z1^2+z2^2))
,0],[0,0,((1/12)*m3*(z1^2+z3^2))]])):
p4:=LinearMomentum(world,Wedge21_cg,m1):
h4:=AngularMomentum(world,wedge21,matrix([[((1/12)*m1*(z3^2+z2^2)),0,0],[0,((1/12)*m1*(z1^2+z2^2)
),0],[0,0,((1/12)*m1*(z1^2+z3^2))]])):
p5:=LinearMomentum(world,Wedge22_cg,m3):
h5:=AngularMomentum(world,wedge22,matrix([[((1/12)*m3*(z3^2+z2^2)),0,0],[0,((1/12)*m3*(z1^2+z2^2)
),0],[0,0,((1/12)*m3*(z1^2+z3^2))]])):

#I1XX=((1/12)*m1*(z3^2+z2^2));
#I1YY=((1/12)*m1*(z1^2+z2^2));
#I1ZZ=((1/12)*m1*(z1^2+z3^2));
#I2XX=((1/12)*m2*((12^2)+((z2+5)^2)));
#I2YY=((1/12)*m2*((4+2*(z2*tan(theta*Pi/180))^2)+((z2+5)^2)));
#I2ZZ=((1/12)*m2*((4+2*(z2*tan(theta*Pi/180))^2)+(12^2)));
#I3XX=((1/12)*m3*(z3^2+z2^2));
#I3YY=((1/12)*m3*(z1^2+z2^2));
#I3ZZ=((1/12)*m3*(z1^2+z3^2));

rwedge1:=FindTranslation(Spring1_1,Spring1_2):
rbolster:=FindTranslation(Spring2_1,Spring2_2):
rwedge2:=FindTranslation(Spring3_1,Spring3_2):
rwedge21:=FindTranslation(Spring21_1,Spring21_2):
rbolster2:=FindTranslation(Spring22_1,Spring22_2):
rwedge22:=FindTranslation(Spring23_1,Spring23_2):
```

```
rBolster1:=FindTranslation(Bolster_cg,Bolster1):
rBolster2:=FindTranslation(Bolster_cg,Bolster2):
rBolster3:=FindTranslation(Bolster_cg,Bolster3):
rBolster4:=FindTranslation(Bolster_cg,Bolster4):
rBolster5:=FindTranslation(Bolster_cg,Bolster5):
rBolster6:=FindTranslation(Bolster_cg,Bolster6):
rBolster21:=FindTranslation(Bolster_cg,Bolster21):
rBolster22:=FindTranslation(Bolster_cg,Bolster22):
rBolster23:=FindTranslation(Bolster_cg,Bolster23):
rBolster24:=FindTranslation(Bolster_cg,Bolster24):
rBolster25:=FindTranslation(Bolster_cg,Bolster25):
rBolster26:=FindTranslation(Bolster_cg,Bolster26):
rBolsterl:=FindTranslation(Bolster_cg,Bolsterl):
rBolsterr:=FindTranslation(Bolster_cg,Bolsterr):

rWedge1_2:=FindTranslation(Wedge1_cg,Wedge1_2):
rWedge1_3:=FindTranslation(Wedge1_cg,Wedge1_3):
rWedge1_4:=FindTranslation(Wedge1_cg,Wedge1_4):
rWedge1_5:=FindTranslation(Wedge1_cg,Wedge1_5):
rWedge1_6:=FindTranslation(Wedge1_cg,Wedge1_6):
rWedge1_7:=FindTranslation(Wedge1_cg,Wedge1_7):
rWedge21_2:=FindTranslation(Wedge21_cg,Wedge21_2):
rWedge21_3:=FindTranslation(Wedge21_cg,Wedge21_3):
rWedge21_4:=FindTranslation(Wedge21_cg,Wedge21_4):
rWedge21_5:=FindTranslation(Wedge21_cg,Wedge21_5):
rWedge21_6:=FindTranslation(Wedge21_cg,Wedge21_6):
rWedge21_7:=FindTranslation(Wedge21_cg,Wedge21_7):

rWedge2_2:=FindTranslation(Wedge2_cg,Wedge2_2):
rWedge2_3:=FindTranslation(Wedge2_cg,Wedge2_3):
rWedge2_4:=FindTranslation(Wedge2_cg,Wedge2_4):
rWedge2_5:=FindTranslation(Wedge2_cg,Wedge2_5):
rWedge2_6:=FindTranslation(Wedge2_cg,Wedge2_6):
rWedge2_7:=FindTranslation(Wedge2_cg,Wedge2_7):
rWedge22_2:=FindTranslation(Wedge22_cg,Wedge22_2):
rWedge22_3:=FindTranslation(Wedge22_cg,Wedge22_3):
rWedge22_4:=FindTranslation(Wedge22_cg,Wedge22_4):
rWedge22_5:=FindTranslation(Wedge22_cg,Wedge22_5):
rWedge22_6:=FindTranslation(Wedge22_cg,Wedge22_6):
rWedge22_7:=FindTranslation(Wedge22_cg,Wedge22_7):

rW1:=(1/VectorLength(FindTranslation(Bolster7,Bolster10)))                                  &**
FindTranslation(Bolster7,Bolster10):
rW2:=(1/VectorLength(FindTranslation(Bolster8,Bolster9))) &** FindTranslation(Bolster8,Bolster9):
rW3:=(rW2 &xx rW1):

rW4:=(1/VectorLength(FindTranslation(Bolster11,Bolster14)))                                 &**
FindTranslation(Bolster11,Bolster14):
rW5:=(1/VectorLength(FindTranslation(Bolster12,Bolster13)))                                 &**
FindTranslation(Bolster12,Bolster13):
rW6:=(rW4 &xx rW5):

rW7:=(1/VectorLength(FindTranslation(Bolster3,Bolster5))) &** FindTranslation(Bolster3,Bolster5):
rW8:=(1/VectorLength(FindTranslation(Bolster11,Bolster13)))                                 &**
FindTranslation(Bolster11,Bolster13):

rW9:=(1/VectorLength(FindTranslation(Bolster27,Bolster210)))                                &**
FindTranslation(Bolster27,Bolster210):
rW10:=(1/VectorLength(FindTranslation(Bolster28,Bolster29)))                                &**
FindTranslation(Bolster28,Bolster29):
rW11:=(rW10 &xx rW9):

rW12:=(1/VectorLength(FindTranslation(Bolster211,Bolster214)))                              &**
FindTranslation(Bolster211,Bolster214):
rW13:=(1/VectorLength(FindTranslation(Bolster212,Bolster213)))                              &**
FindTranslation(Bolster212,Bolster213):
rW14:=(rW12 &xx rW13):

rW15:=(1/VectorLength(FindTranslation(Bolster23,Bolster25)))                                &**
FindTranslation(Bolster23,Bolster25):
```

```
rW16:=(1/VectorLength(FindTranslation(Bolster213,Bolster211)))                      &**
FindTranslation(Bolster213,Bolster211):

vWedge1_3:=subs(kde,LinearVelocity(sideframe,Wedge1_3))&oo MakeTranslations(s,3):
vWedge1_6:=subs(kde,LinearVelocity(sideframe,Wedge1_6))&oo MakeTranslations(s,3):
vWedge1_4:=subs(kde,LinearVelocity(sideframe,Wedge1_4))&oo MakeTranslations(s,3):
vWedge1_7:=subs(kde,LinearVelocity(sideframe,Wedge1_7))&oo MakeTranslations(s,3):
vWedge21_3:=subs(kde,LinearVelocity(sideframe21,Wedge21_3))&oo MakeTranslations(s21,3):
vWedge21_6:=subs(kde,LinearVelocity(sideframe21,Wedge21_6))&oo MakeTranslations(s21,3):
vWedge21_4:=subs(kde,LinearVelocity(sideframe21,Wedge21_4))&oo MakeTranslations(s21,3):
vWedge21_7:=subs(kde,LinearVelocity(sideframe21,Wedge21_7))&oo MakeTranslations(s21,3):

vWedge2_3:=subs(kde,LinearVelocity(sideframe,Wedge2_3))&oo MakeTranslations(s,3):
vWedge2_6:=subs(kde,LinearVelocity(sideframe,Wedge2_6))&oo MakeTranslations(s,3):
vWedge2_4:=subs(kde,LinearVelocity(sideframe,Wedge2_4))&oo MakeTranslations(s,3):
vWedge2_7:=subs(kde,LinearVelocity(sideframe,Wedge2_7))&oo MakeTranslations(s,3):
vWedge22_3:=subs(kde,LinearVelocity(sideframe21,Wedge22_3))&oo MakeTranslations(s21,3):
vWedge22_6:=subs(kde,LinearVelocity(sideframe21,Wedge22_6))&oo MakeTranslations(s21,3):
vWedge22_4:=subs(kde,LinearVelocity(sideframe21,Wedge22_4))&oo MakeTranslations(s21,3):
vWedge22_7:=subs(kde,LinearVelocity(sideframe21,Wedge22_7))&oo MakeTranslations(s21,3):

vBolsterWedge1_2:=(subs(kde,LinearVelocity(bolster,Wedge1_2))&oo rW7):
vBolsterWedge1_5:=(subs(kde,LinearVelocity(bolster,Wedge1_5))&oo rW7):
vBolsterWedge1_4:=(subs(kde,LinearVelocity(bolster,Wedge1_4))&oo rW7):
vBolsterWedge1_7:=(subs(kde,LinearVelocity(bolster,Wedge1_7))&oo rW7):
vBolsterWedge21_2:=(subs(kde,LinearVelocity(bolster,Wedge21_2))&oo rW15):
vBolsterWedge21_5:=(subs(kde,LinearVelocity(bolster,Wedge21_5))&oo rW15):
vBolsterWedge21_4:=(subs(kde,LinearVelocity(bolster,Wedge21_4))&oo rW15):
vBolsterWedge21_7:=(subs(kde,LinearVelocity(bolster,Wedge21_7))&oo rW15):

vBolsterWedge2_2:=(subs(kde,LinearVelocity(bolster,Wedge2_2))&oo rW8):
vBolsterWedge2_5:=(subs(kde,LinearVelocity(bolster,Wedge2_5))&oo rW8):
vBolsterWedge2_4:=(subs(kde,LinearVelocity(bolster,Wedge2_4))&oo rW8):
vBolsterWedge2_7:=(subs(kde,LinearVelocity(bolster,Wedge2_7))&oo rW8):
vBolsterWedge22_2:=(subs(kde,LinearVelocity(bolster,Wedge22_2))&oo rW16):
vBolsterWedge22_5:=(subs(kde,LinearVelocity(bolster,Wedge22_5))&oo rW16):
vBolsterWedge22_4:=(subs(kde,LinearVelocity(bolster,Wedge22_4))&oo rW16):
vBolsterWedge22_7:=(subs(kde,LinearVelocity(bolster,Wedge22_7))&oo rW16):

#External forces
f1:=MakeTranslations(w,0,0,m1*(-32.2))
    &++ ((-k*(l0/VectorLength(rwedge1)-1)) &** rwedge1)
    &-- (Cdamp &** LinearVelocity(sideframe,Wedge1_cg))

    &-- ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &** rW3)
    &-- ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &** rW3)

    &++
((kb*rCBolsterWedge1_2*muB*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)   &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_5*muB*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)   &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_4*muB*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)   &oo
rW3)) &** rW7)
    &++
((kb*rCBolsterWedge1_7*muB*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)   &oo
rW3)) &** rW7)

    &--            ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)           &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))
    &--            ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)           &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))
    &--            ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)           &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))
    &--            ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)           &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))
```

```
    &-- ((ks*rCSideframe1Wedge1_3*muS*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))
    &-- ((ks*rCSideframe1Wedge1_6*muS*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))
    &-- ((ks*rCSideframe1Wedge1_4*muS*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))
    &-- ((ks*rCSideframe1Wedge1_7*muS*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))

    &++ NullVector():

t1:=NullVector()
    &-- (Cdamp &** AngularVelocity(s,wedge1))

    &-- (rWedge1_2 &xx ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &**
rW3))
    &-- (rWedge1_5 &xx ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &**
rW3))
    &-- (rWedge1_4 &xx ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &**
rW3))
    &-- (rWedge1_7 &xx ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &**
rW3))

    &++                                         (rWedge1_2                                &xx
((kb*rCBolsterWedge1_2*muB*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)    &oo
rW3)) &** rW7))
    &++                                         (rWedge1_5                                &xx
((kb*rCBolsterWedge1_5*muB*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)    &oo
rW3)) &** rW7))
    &++                                         (rWedge1_4                                &xx
((kb*rCBolsterWedge1_4*muB*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)    &oo
rW3)) &** rW7))
    &++                                         (rWedge1_7                                &xx
((kb*rCBolsterWedge1_7*muB*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)    &oo
rW3)) &** rW7))

    &--   (rWedge1_3   &xx   ((ks*rCSideframe1Wedge1_3*(FindTranslation(Sideframe1,Wedge1_3)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))
    &--   (rWedge1_6   &xx   ((ks*rCSideframe1Wedge1_6*(FindTranslation(Sideframe1,Wedge1_6)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))
    &--   (rWedge1_4   &xx   ((ks*rCSideframe1Wedge1_4*(FindTranslation(Sideframe1,Wedge1_4)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))
    &--   (rWedge1_7   &xx   ((ks*rCSideframe1Wedge1_7*(FindTranslation(Sideframe1,Wedge1_7)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))

    &--                                         (rWedge1_3                                &xx
((ks*rCSideframe1Wedge1_3*muS*tanh(vWedge1_3/0.001)*(FindTranslation(Sideframe1,Wedge1_3)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))
    &--                                         (rWedge1_6                                &xx
((ks*rCSideframe1Wedge1_6*muS*tanh(vWedge1_6/0.001)*(FindTranslation(Sideframe1,Wedge1_6)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))
    &--                                         (rWedge1_4                                &xx
((ks*rCSideframe1Wedge1_4*muS*tanh(vWedge1_4/0.001)*(FindTranslation(Sideframe1,Wedge1_4)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))
    &--                                         (rWedge1_7                                &xx
((ks*rCSideframe1Wedge1_7*muS*tanh(vWedge1_7/0.001)*(FindTranslation(Sideframe1,Wedge1_7)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))

    &++ NullVector():


f2:=MakeTranslations(w,0,0,m2*(-32.2))
    &++ ((-k2*(l02/VectorLength(rbolster)-1)) &** rbolster)
    &++ ((-k2*(l02/VectorLength(rbolster2)-1)) &** rbolster2)
    &-- (Cdamp &** LinearVelocity(world,Bolster_cg))

    &++ ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4) &oo rW3)) &** rW3)
    &++ ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7) &oo rW3)) &** rW3)
```

```
    &--
((kb*rCBolsterWedge1_2*muB*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)    &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_5*muB*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)    &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_4*muB*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)    &oo
rW3)) &** rW7)
    &--
((kb*rCBolsterWedge1_7*muB*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)    &oo
rW3)) &** rW7)

    &++ ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6)
    &++ ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6)
    &++ ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6)
    &++ ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6)

    &--
((kb*rCBolsterWedge2_2*muB*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)    &oo
rW6)) &** rW8)
    &--
((kb*rCBolsterWedge2_5*muB*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)    &oo
rW6)) &** rW8)
    &--
((kb*rCBolsterWedge2_4*muB*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)    &oo
rW6)) &** rW8)
    &--
((kb*rCBolsterWedge2_7*muB*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)    &oo
rW6)) &** rW8)

    &++ ((kb*rCBolsterWedge21_2*(FindTranslation(Bolster23,Wedge21_2) &oo rW11)) &** rW11)
    &++ ((kb*rCBolsterWedge21_5*(FindTranslation(Bolster23,Wedge21_5) &oo rW11)) &** rW11)
    &++ ((kb*rCBolsterWedge21_4*(FindTranslation(Bolster25,Wedge21_4) &oo rW11)) &** rW11)
    &++ ((kb*rCBolsterWedge21_7*(FindTranslation(Bolster25,Wedge21_7) &oo rW11)) &** rW11)

    &--
((kb*rCBolsterWedge21_2*muB*tanh(vBolsterWedge21_2/0.001)*(FindTranslation(Bolster23,Wedge21_2)
&oo rW11)) &** rW15)
    &--
((kb*rCBolsterWedge21_5*muB*tanh(vBolsterWedge21_5/0.001)*(FindTranslation(Bolster23,Wedge21_5)
&oo rW11)) &** rW15)
    &--
((kb*rCBolsterWedge21_4*muB*tanh(vBolsterWedge21_4/0.001)*(FindTranslation(Bolster25,Wedge21_4)
&oo rW11)) &** rW15)
    &--
((kb*rCBolsterWedge21_7*muB*tanh(vBolsterWedge21_7/0.001)*(FindTranslation(Bolster25,Wedge21_7)
&oo rW11)) &** rW15)

    &++ ((kb*rCBolsterWedge22_2*(FindTranslation(Bolster24,Wedge22_2) &oo rW14)) &** rW14)
    &++ ((kb*rCBolsterWedge22_5*(FindTranslation(Bolster24,Wedge22_5) &oo rW14)) &** rW14)
    &++ ((kb*rCBolsterWedge22_4*(FindTranslation(Bolster26,Wedge22_4) &oo rW14)) &** rW14)
    &++ ((kb*rCBolsterWedge22_7*(FindTranslation(Bolster26,Wedge22_7) &oo rW14)) &** rW14)

    &--
((kb*rCBolsterWedge22_2*muB*tanh(vBolsterWedge22_2/0.001)*(FindTranslation(Bolster24,Wedge22_2)
&oo rW14)) &** rW16)
    &--
((kb*rCBolsterWedge22_5*muB*tanh(vBolsterWedge22_5/0.001)*(FindTranslation(Bolster24,Wedge22_5)
&oo rW14)) &** rW16)
    &--
((kb*rCBolsterWedge22_4*muB*tanh(vBolsterWedge22_4/0.001)*(FindTranslation(Bolster26,Wedge22_4)
&oo rW14)) &** rW16)
    &--
((kb*rCBolsterWedge22_7*muB*tanh(vBolsterWedge22_7/0.001)*(FindTranslation(Bolster26,Wedge22_7)
&oo rW14)) &** rW16)

    &++ NullVector():

t2:=NullVector()
    &++ (rBolsterl &xx ((-k2*(l02/VectorLength(rbolster)-1)) &** rbolster))
```

```
    &++ (rBolsterr &xx ((-k2*(l02/VectorLength(rbolster2)-1)) &** rbolster2))
    &-- (Cdamp &** AngularVelocity(w,bolster))

    &++  (rBolster3  &xx  ((kb*rCBolsterWedge1_2*(FindTranslation(Bolster3,Wedge1_2)  &oo  rW3))  &**
rW3))
    &++  (rBolster3  &xx  ((kb*rCBolsterWedge1_5*(FindTranslation(Bolster3,Wedge1_5)  &oo  rW3))  &**
rW3))
    &++  (rBolster5  &xx  ((kb*rCBolsterWedge1_4*(FindTranslation(Bolster5,Wedge1_4)  &oo  rW3))  &**
rW3))
    &++  (rBolster5  &xx  ((kb*rCBolsterWedge1_7*(FindTranslation(Bolster5,Wedge1_7)  &oo  rW3))  &**
rW3))

    &--                                     (rBolster3                                          &xx
((kb*rCBolsterWedge1_2*muB*tanh(vBolsterWedge1_2/0.001)*(FindTranslation(Bolster3,Wedge1_2)    &oo
rW3)) &** rW7))
    &--                                     (rBolster3                                          &xx
((kb*rCBolsterWedge1_5*muB*tanh(vBolsterWedge1_5/0.001)*(FindTranslation(Bolster3,Wedge1_5)    &oo
rW3)) &** rW7))
    &--                                     (rBolster5                                          &xx
((kb*rCBolsterWedge1_4*muB*tanh(vBolsterWedge1_4/0.001)*(FindTranslation(Bolster5,Wedge1_4)    &oo
rW3)) &** rW7))
    &--                                     (rBolster5                                          &xx
((kb*rCBolsterWedge1_7*muB*tanh(vBolsterWedge1_7/0.001)*(FindTranslation(Bolster5,Wedge1_7)    &oo
rW3)) &** rW7))

    &++  (rBolster4  &xx  ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2)  &oo  rW6))  &**
rW6))
    &++  (rBolster4  &xx  ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5)  &oo  rW6))  &**
rW6))
    &++  (rBolster6  &xx  ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4)  &oo  rW6))  &**
rW6))
    &++  (rBolster6  &xx  ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7)  &oo  rW6))  &**
rW6))

    &--                                     (rBolster4                                          &xx
((kb*rCBolsterWedge2_2*muB*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)    &oo
rW6)) &** rW8))
    &--                                     (rBolster4                                          &xx
((kb*rCBolsterWedge2_5*muB*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)    &oo
rW6)) &** rW8))
    &--                                     (rBolster6                                          &xx
((kb*rCBolsterWedge2_4*muB*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)    &oo
rW6)) &** rW8))
    &--                                     (rBolster6                                          &xx
((kb*rCBolsterWedge2_7*muB*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)    &oo
rW6)) &** rW8))

    &++ (rBolster23 &xx ((kb*rCBolsterWedge21_2*(FindTranslation(Bolster23,Wedge21_2) &oo rW11))
&** rW11))
    &++ (rBolster23 &xx ((kb*rCBolsterWedge21_5*(FindTranslation(Bolster23,Wedge21_5) &oo rW11))
&** rW11))
    &++ (rBolster25 &xx ((kb*rCBolsterWedge21_4*(FindTranslation(Bolster25,Wedge21_4) &oo rW11))
&** rW11))
    &++ (rBolster25 &xx ((kb*rCBolsterWedge21_7*(FindTranslation(Bolster25,Wedge21_7) &oo rW11))
&** rW11))

    &--                                    (rBolster23                                          &xx
((kb*rCBolsterWedge21_2*muB*tanh(vBolsterWedge21_2/0.001)*(FindTranslation(Bolster23,Wedge21_2)
&oo rW11)) &** rW15))
    &--                                    (rBolster23                                          &xx
((kb*rCBolsterWedge21_5*muB*tanh(vBolsterWedge21_5/0.001)*(FindTranslation(Bolster23,Wedge21_5)
&oo rW11)) &** rW15))
    &--                                    (rBolster25                                          &xx
((kb*rCBolsterWedge21_4*muB*tanh(vBolsterWedge21_4/0.001)*(FindTranslation(Bolster25,Wedge21_4)
&oo rW11)) &** rW15))
    &--                                    (rBolster25                                          &xx
((kb*rCBolsterWedge21_7*muB*tanh(vBolsterWedge21_7/0.001)*(FindTranslation(Bolster25,Wedge21_7)
&oo rW11)) &** rW15))

    &++ (rBolster24 &xx ((kb*rCBolsterWedge22_2*(FindTranslation(Bolster24,Wedge22_2) &oo rW14))
&** rW14))
```

```
    &++ (rBolster24 &xx ((kb*rCBolsterWedge22_5*(FindTranslation(Bolster24,Wedge22_5) &oo rW14))
&** rW14))
    &++ (rBolster26 &xx ((kb*rCBolsterWedge22_4*(FindTranslation(Bolster26,Wedge22_4) &oo rW14))
&** rW14))
    &++ (rBolster26 &xx ((kb*rCBolsterWedge22_7*(FindTranslation(Bolster26,Wedge22_7) &oo rW14))
&** rW14))

    &--                                    (rBolster24                                      &xx
((kb*rCBolsterWedge22_2*muB*tanh(vBolsterWedge22_2/0.001)*(FindTranslation(Bolster24,Wedge22_2)
&oo rW14)) &** rW16))
    &--                                    (rBolster24                                      &xx
((kb*rCBolsterWedge22_5*muB*tanh(vBolsterWedge22_5/0.001)*(FindTranslation(Bolster24,Wedge22_5)
&oo rW14)) &** rW16))
    &--                                    (rBolster26                                      &xx
((kb*rCBolsterWedge22_4*muB*tanh(vBolsterWedge22_4/0.001)*(FindTranslation(Bolster26,Wedge22_4)
&oo rW14)) &** rW16))
    &--                                    (rBolster26                                      &xx
((kb*rCBolsterWedge22_7*muB*tanh(vBolsterWedge22_7/0.001)*(FindTranslation(Bolster26,Wedge22_7)
&oo rW14)) &** rW16))

    &++ NullVector():


f3:=MakeTranslations(w,0,0,m3*(-32.2))
    &++ ((-k*(l0/VectorLength(rwedge2)-1)) &** rwedge2)
    &-- (Cdamp &** LinearVelocity(sideframe,Wedge2_cg))

    &-- ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &** rW6)
    &-- ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &** rW6)
    &-- ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &** rW6)
    &-- ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &** rW6)

    &++
((kb*rCBolsterWedge2_2*muB*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)   &oo
rW6)) &** rW8)
    &++
((kb*rCBolsterWedge2_5*muB*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)   &oo
rW6)) &** rW8)
    &++
((kb*rCBolsterWedge2_4*muB*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)   &oo
rW6)) &** rW8)
    &++
((kb*rCBolsterWedge2_7*muB*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)   &oo
rW6)) &** rW8)

    &--            ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)             &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))
    &--            ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)             &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))
    &--            ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)             &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))
    &--            ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)             &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1))

    &++ ((ks*rCSideframe2Wedge2_3*muS*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))
    &++ ((ks*rCSideframe2Wedge2_6*muS*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))
    &++ ((ks*rCSideframe2Wedge2_4*muS*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))
    &++ ((ks*rCSideframe2Wedge2_7*muS*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)
&oo MakeTranslations(s,1))) &** MakeTranslations(s,3))

    &++ NullVector():

t3:=NullVector()
    &-- (Cdamp &** AngularVelocity(s,wedge2))

    &-- (rWedge2_2 &xx ((kb*rCBolsterWedge2_2*(FindTranslation(Bolster4,Wedge2_2) &oo rW6)) &**
rW6))
```

```
    &-- (rWedge2_5 &xx ((kb*rCBolsterWedge2_5*(FindTranslation(Bolster4,Wedge2_5) &oo rW6)) &**
rW6))
    &-- (rWedge2_4 &xx ((kb*rCBolsterWedge2_4*(FindTranslation(Bolster6,Wedge2_4) &oo rW6)) &**
rW6))
    &-- (rWedge2_7 &xx ((kb*rCBolsterWedge2_7*(FindTranslation(Bolster6,Wedge2_7) &oo rW6)) &**
rW6))

    &++                                     (rWedge2_2                                      &xx
((kb*rCBolsterWedge2_2*muB*tanh(vBolsterWedge2_2/0.001)*(FindTranslation(Bolster4,Wedge2_2)    &oo
rW6)) &** rW8))
    &++                                     (rWedge2_5                                      &xx
((kb*rCBolsterWedge2_5*muB*tanh(vBolsterWedge2_5/0.001)*(FindTranslation(Bolster4,Wedge2_5)    &oo
rW6)) &** rW8))
    &++                                     (rWedge2_4                                      &xx
((kb*rCBolsterWedge2_4*muB*tanh(vBolsterWedge2_4/0.001)*(FindTranslation(Bolster6,Wedge2_4)    &oo
rW6)) &** rW8))
    &++                                     (rWedge2_7                                      &xx
((kb*rCBolsterWedge2_7*muB*tanh(vBolsterWedge2_7/0.001)*(FindTranslation(Bolster6,Wedge2_7)    &oo
rW6)) &** rW8))

    &--   (rWedge2_3   &xx   ((ks*rCSideframe2Wedge2_3*(FindTranslation(Sideframe2,Wedge2_3)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))
    &--   (rWedge2_6   &xx   ((ks*rCSideframe2Wedge2_6*(FindTranslation(Sideframe2,Wedge2_6)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))
    &--   (rWedge2_4   &xx   ((ks*rCSideframe2Wedge2_4*(FindTranslation(Sideframe2,Wedge2_4)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))
    &--   (rWedge2_7   &xx   ((ks*rCSideframe2Wedge2_7*(FindTranslation(Sideframe2,Wedge2_7)   &oo
MakeTranslations(s,1))) &** MakeTranslations(s,1)))

    &++                                     (rWedge2_3                                      &xx
((ks*rCSideframe2Wedge2_3*muS*tanh(vWedge2_3/0.001)*(FindTranslation(Sideframe2,Wedge2_3)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))
    &++                                     (rWedge2_6                                      &xx
((ks*rCSideframe2Wedge2_6*muS*tanh(vWedge2_6/0.001)*(FindTranslation(Sideframe2,Wedge2_6)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))
    &++                                     (rWedge2_4                                      &xx
((ks*rCSideframe2Wedge2_4*muS*tanh(vWedge2_4/0.001)*(FindTranslation(Sideframe2,Wedge2_4)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))
    &++                                     (rWedge2_7                                      &xx
((ks*rCSideframe2Wedge2_7*muS*tanh(vWedge2_7/0.001)*(FindTranslation(Sideframe2,Wedge2_7)      &oo
MakeTranslations(s,1))) &** MakeTranslations(s,3)))

    &++ NullVector():


f4:=MakeTranslations(w,0,0,0.875*(-32.2))
    &++ ((-k*(l0/VectorLength(rwedge21)-1)) &** rwedge21)
    &-- (Cdamp &** LinearVelocity(sideframe21,Wedge21_cg))

    &-- ((kb*rCBolsterWedge21_2*(FindTranslation(Bolster23,Wedge21_2) &oo rW11)) &** rW11)
    &-- ((kb*rCBolsterWedge21_5*(FindTranslation(Bolster23,Wedge21_5) &oo rW11)) &** rW11)
    &-- ((kb*rCBolsterWedge21_4*(FindTranslation(Bolster25,Wedge21_4) &oo rW11)) &** rW11)
    &-- ((kb*rCBolsterWedge21_7*(FindTranslation(Bolster25,Wedge21_7) &oo rW11)) &** rW11)

    &++
((kb*rCBolsterWedge21_2*muB*tanh(vBolsterWedge21_2/0.001)*(FindTranslation(Bolster23,Wedge21_2)
&oo rW11)) &** rW15)
    &++
((kb*rCBolsterWedge21_5*muB*tanh(vBolsterWedge21_5/0.001)*(FindTranslation(Bolster23,Wedge21_5)
&oo rW11)) &** rW15)
    &++
((kb*rCBolsterWedge21_4*muB*tanh(vBolsterWedge21_4/0.001)*(FindTranslation(Bolster25,Wedge21_4)
&oo rW11)) &** rW15)
    &++
((kb*rCBolsterWedge21_7*muB*tanh(vBolsterWedge21_7/0.001)*(FindTranslation(Bolster25,Wedge21_7)
&oo rW11)) &** rW15)

    &--          ((ks*rCSideframe21Wedge21_3*(FindTranslation(Sideframe22,Wedge21_3)          &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1))
    &--          ((ks*rCSideframe21Wedge21_6*(FindTranslation(Sideframe22,Wedge21_6)          &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1))
```

```
    &--          ((ks*rCSideframe21Wedge21_4*(FindTranslation(Sideframe22,Wedge21_4)          &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1))
    &--          ((ks*rCSideframe21Wedge21_7*(FindTranslation(Sideframe22,Wedge21_7)          &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1))

    &--
((ks*rCSideframe21Wedge21_3*muS*tanh(vWedge21_3/0.001)*(FindTranslation(Sideframe22,Wedge21_3)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3))
    &--
((ks*rCSideframe21Wedge21_6*muS*tanh(vWedge21_6/0.001)*(FindTranslation(Sideframe22,Wedge21_6)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3))
    &--
((ks*rCSideframe21Wedge21_4*muS*tanh(vWedge21_4/0.001)*(FindTranslation(Sideframe22,Wedge21_4)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3))
    &--
((ks*rCSideframe21Wedge21_7*muS*tanh(vWedge21_7/0.001)*(FindTranslation(Sideframe22,Wedge21_7)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3))

    &++ NullVector():

t4:=NullVector()
    &-- (Cdamp &** AngularVelocity(s21,wedge21))

    &-- (rWedge21_2 &xx ((kb*rCBolsterWedge21_2*(FindTranslation(Bolster23,Wedge21_2) &oo rW11))
&** rW11))
    &-- (rWedge21_5 &xx ((kb*rCBolsterWedge21_5*(FindTranslation(Bolster23,Wedge21_5) &oo rW11))
&** rW11))
    &-- (rWedge21_4 &xx ((kb*rCBolsterWedge21_4*(FindTranslation(Bolster25,Wedge21_4) &oo rW11))
&** rW11))
    &-- (rWedge21_7 &xx ((kb*rCBolsterWedge21_7*(FindTranslation(Bolster25,Wedge21_7) &oo rW11))
&** rW11))

    &++                                  (rWedge21_2                                  &xx
((kb*rCBolsterWedge21_2*muB*tanh(vBolsterWedge21_2/0.001)*(FindTranslation(Bolster23,Wedge21_2)
&oo rW11)) &** rW15))
    &++                                  (rWedge21_5                                  &xx
((kb*rCBolsterWedge21_5*muB*tanh(vBolsterWedge21_5/0.001)*(FindTranslation(Bolster23,Wedge21_5)
&oo rW11)) &** rW15))
    &++                                  (rWedge21_4                                  &xx
((kb*rCBolsterWedge21_4*muB*tanh(vBolsterWedge21_4/0.001)*(FindTranslation(Bolster25,Wedge21_4)
&oo rW11)) &** rW15))
    &++                                  (rWedge21_7                                  &xx
((kb*rCBolsterWedge21_7*muB*tanh(vBolsterWedge21_7/0.001)*(FindTranslation(Bolster25,Wedge21_7)
&oo rW11)) &** rW15))

    &-- (rWedge21_3 &xx ((ks*rCSideframe21Wedge21_3*(FindTranslation(Sideframe22,Wedge21_3) &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1)))
    &-- (rWedge21_6 &xx ((ks*rCSideframe21Wedge21_6*(FindTranslation(Sideframe22,Wedge21_6) &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1)))
    &-- (rWedge21_4 &xx ((ks*rCSideframe21Wedge21_4*(FindTranslation(Sideframe22,Wedge21_4) &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1)))
    &-- (rWedge21_7 &xx ((ks*rCSideframe21Wedge21_7*(FindTranslation(Sideframe22,Wedge21_7) &oo
MakeTranslations(s21,1))) &** MakeTranslations(s21,1)))

    &--                                  (rWedge21_3                                  &xx
((ks*rCSideframe21Wedge21_3*muS*tanh(vWedge21_3/0.001)*(FindTranslation(Sideframe22,Wedge21_3)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3)))
    &--                                  (rWedge21_6                                  &xx
((ks*rCSideframe21Wedge21_6*muS*tanh(vWedge21_6/0.001)*(FindTranslation(Sideframe22,Wedge21_6)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3)))
    &--                                  (rWedge21_4                                  &xx
((ks*rCSideframe21Wedge21_4*muS*tanh(vWedge21_4/0.001)*(FindTranslation(Sideframe22,Wedge21_4)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3)))
    &--                                  (rWedge21_7                                  &xx
((ks*rCSideframe21Wedge21_7*muS*tanh(vWedge21_7/0.001)*(FindTranslation(Sideframe22,Wedge21_7)
&oo MakeTranslations(s21,1))) &** MakeTranslations(s21,3)))

    &++ NullVector():

f5:=MakeTranslations(w,0,0,0.875*(-32.2))
    &++ ((-k*(l0/VectorLength(rwedge22)-1)) &** rwedge22)
```

```
    &-- (Cdamp &** LinearVelocity(sideframe21,Wedge22_cg))

    &-- ((kb*rCBolsterWedge22_2*(FindTranslation(Bolster24,Wedge22_2) &oo rW14)) &** rW14)
    &-- ((kb*rCBolsterWedge22_5*(FindTranslation(Bolster24,Wedge22_5) &oo rW14)) &** rW14)
    &-- ((kb*rCBolsterWedge22_4*(FindTranslation(Bolster26,Wedge22_4) &oo rW14)) &** rW14)
    &-- ((kb*rCBolsterWedge22_7*(FindTranslation(Bolster26,Wedge22_7) &oo rW14)) &** rW14)

    &++
((kb*rCBolsterWedge22_2*muB*tanh(vBolsterWedge22_2/0.001)*(FindTranslation(Bolster24,Wedge22_2)
&oo rW14)) &** rW16)
    &++
((kb*rCBolsterWedge22_5*muB*tanh(vBolsterWedge22_5/0.001)*(FindTranslation(Bolster24,Wedge22_5)
&oo rW14)) &** rW16)
    &++
((kb*rCBolsterWedge22_4*muB*tanh(vBolsterWedge22_4/0.001)*(FindTranslation(Bolster26,Wedge22_4)
&oo rW14)) &** rW16)
    &++
((kb*rCBolsterWedge22_7*muB*tanh(vBolsterWedge22_7/0.001)*(FindTranslation(Bolster26,Wedge22_7)
&oo rW14)) &** rW16)

    &--          ((ks*rCSideframe22Wedge22_3*(FindTranslation(Sideframe23,Wedge22_3)          &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1))
    &--          ((ks*rCSideframe22Wedge22_6*(FindTranslation(Sideframe23,Wedge22_6)          &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1))
    &--          ((ks*rCSideframe22Wedge22_4*(FindTranslation(Sideframe23,Wedge22_4)          &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1))
    &--          ((ks*rCSideframe22Wedge22_7*(FindTranslation(Sideframe23,Wedge22_7)          &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1))

    &++
((ks*rCSideframe22Wedge22_3*muS*tanh(vWedge22_3/0.001)*(FindTranslation(Sideframe23,Wedge22_3)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3))
    &++
((ks*rCSideframe22Wedge22_6*muS*tanh(vWedge22_6/0.001)*(FindTranslation(Sideframe23,Wedge22_6)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3))
    &++
((ks*rCSideframe22Wedge22_4*muS*tanh(vWedge22_4/0.001)*(FindTranslation(Sideframe23,Wedge22_4)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3))
    &++
((ks*rCSideframe22Wedge22_7*muS*tanh(vWedge22_7/0.001)*(FindTranslation(Sideframe23,Wedge22_7)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3))

    &++ NullVector():

t5:=NullVector()
    &-- (Cdamp &** AngularVelocity(s21,wedge22))

    &-- (rWedge22_2 &xx ((kb*rCBolsterWedge22_2*(FindTranslation(Bolster24,Wedge22_2) &oo rW14))
&** rW14))
    &-- (rWedge22_5 &xx ((kb*rCBolsterWedge22_5*(FindTranslation(Bolster24,Wedge22_5) &oo rW14))
&** rW14))
    &-- (rWedge22_4 &xx ((kb*rCBolsterWedge22_4*(FindTranslation(Bolster26,Wedge22_4) &oo rW14))
&** rW14))
    &-- (rWedge22_7 &xx ((kb*rCBolsterWedge22_7*(FindTranslation(Bolster26,Wedge22_7) &oo rW14))
&** rW14))

    &++                                   (rWedge22_2                                      &xx
((kb*rCBolsterWedge22_2*muB*tanh(vBolsterWedge22_2/0.001)*(FindTranslation(Bolster24,Wedge22_2)
&oo rW14)) &** rW16))
    &++                                   (rWedge22_5                                      &xx
((kb*rCBolsterWedge22_5*muB*tanh(vBolsterWedge22_5/0.001)*(FindTranslation(Bolster24,Wedge22_5)
&oo rW14)) &** rW16))
    &++                                   (rWedge22_4                                      &xx
((kb*rCBolsterWedge22_4*muB*tanh(vBolsterWedge22_4/0.001)*(FindTranslation(Bolster26,Wedge22_4)
&oo rW14)) &** rW16))
    &++                                   (rWedge22_7                                      &xx
((kb*rCBolsterWedge22_7*muB*tanh(vBolsterWedge22_7/0.001)*(FindTranslation(Bolster26,Wedge22_7)
&oo rW14)) &** rW16))

    &-- (rWedge22_3 &xx ((ks*rCSideframe22Wedge22_3*(FindTranslation(Sideframe23,Wedge22_3) &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1)))
```

```
    &-- (rWedge22_6  &xx  ((ks*rCSideframe22Wedge22_6*(FindTranslation(Sideframe23,Wedge22_6)  &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1)))
    &-- (rWedge22_4  &xx  ((ks*rCSideframe22Wedge22_4*(FindTranslation(Sideframe23,Wedge22_4)  &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1)))
    &-- (rWedge22_7  &xx  ((ks*rCSideframe22Wedge22_7*(FindTranslation(Sideframe23,Wedge22_7)  &oo
MakeTranslations(s22,1))) &** MakeTranslations(s22,1)))

    &++                                    (rWedge22_3                                    &xx
((ks*rCSideframe22Wedge22_3*muS*tanh(vWedge22_3/0.001)*(FindTranslation(Sideframe23,Wedge22_3)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3)))
    &++                                    (rWedge22_6                                    &xx
((ks*rCSideframe22Wedge22_6*muS*tanh(vWedge22_6/0.001)*(FindTranslation(Sideframe23,Wedge22_6)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3)))
    &++                                    (rWedge22_4                                    &xx
((ks*rCSideframe22Wedge22_4*muS*tanh(vWedge22_4/0.001)*(FindTranslation(Sideframe23,Wedge22_4)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3)))
    &++                                    (rWedge22_7                                    &xx
((ks*rCSideframe22Wedge22_7*muS*tanh(vWedge22_7/0.001)*(FindTranslation(Sideframe23,Wedge22_7)
&oo MakeTranslations(s22,1))) &** MakeTranslations(s22,3)))

    &++ NullVector():

#Puts all the forces in a vector
p:=subs(kde,MomentumDescription([p1,h1,p2,h2,p3,h3,p4,h4,p5,h5])):
appliedforce:=ForceDescription([f1,t1,f2,t2,f3,t3,f4,t4,f5,t5]):

eqs:=((subs(kde,DiffTime(p,w)) &-- appliedforce) &oo beta):
diffeqs:=kde union convert(eqs,set):


#Create bodies
DefineObjects([bolster,'Block',point=MakeTranslations(bolster,0,0,(6.125+5)/2),xlength=4,ylength=
78,zlength=6.125+5,color=red],

[bolster,'Block',point=MakeTranslations(bolster,0,0,6.125+2.5),xlength=4+2*(6.125*tan(37.5*3.1415
927/180)),ylength=78,zlength=5,color=red],
              [bolster3,'Block',point=MakeTranslations(bolster3,-2,(78/2)-
6,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2))/2),xlength=4,ylength=78,zlength=sqrt((6.125*t
an(37.5*3.1415927/180))^2+6.125^2),color=red],
              [bolster4,'Block',point=MakeTranslations(bolster4,2,(78/2)-
6,(sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2))/2),xlength=4,ylength=78,zlength=sqrt((6.125*t
an(37.5*3.1415927/180))^2+6.125^2),color=red],


[wedge1,'Block',point=Wedge1_1,xlength=(z2*tan(theta*Pi/180)),ylength=z3,zlength=0.05,color=blue]
,

[wedge1,'Block',point=MakeTranslations(wedge1,(z2*tan(theta*Pi/180))/2,0,z2/2),xlength=0.05,yleng
th=z3,zlength=z2,color=blue],

[wedge1,'Block',point=MakeTranslations(wedge1,0,0,z2/2),orient=MakeRotations([theta*Pi/180,2]),xl
ength=0.05,ylength=z3,zlength=sqrt((z2*tan(theta*Pi/180))^2+z2^2),color=blue],

              [wedge2,'Block',point=Wedge2_1,xlength=-
(z2*tan(theta*Pi/180)),ylength=z3,zlength=0.05,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,-
(z2*tan(theta*Pi/180))/2,0,z2/2),xlength=0.05,ylength=z3,zlength=z2,color=blue],
              [wedge2,'Block',point=MakeTranslations(wedge2,0,0,z2/2),orient=MakeRotations([-
theta*Pi/180,2]),xlength=0.05,ylength=z3,zlength=sqrt((z2*tan(theta*Pi/180))^2+z2^2),color=blue],


[sideframe,'Cylinder',point=MakeTranslations(s,(z2*tan(theta*Pi/180))/2+2,0,(VectorLength(FindTra
nslation(Spring1_1,Spring1_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring1_1,Sprin
g1_2)),color=green],

[sideframe,'Cylinder',point=MakeTranslations(s,0,0,(VectorLength(FindTranslation(Spring2_1,Spring
2_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring2_1,Spring2_2)),color=green],
              [sideframe,'Cylinder',point=MakeTranslations(s,-(z2*tan(theta*Pi/180))/2-
2,0,(VectorLength(FindTranslation(Spring3_1,Spring3_2)))/2),radius=0.5,length=VectorLength(FindTr
anslation(Spring3_1,Spring3_2)),color=green],
```

```
[sideframe,'Block',point=S,xlength=4+2*(6.125*tan(37.5*3.1415927/180)),ylength=12,zlength=0.01,co
lor=white],

[sideframe1,'Block',point=MakeTranslations(s1,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],

[sideframe2,'Block',point=MakeTranslations(s2,0,0,20/2),xlength=0.01,ylength=12,zlength=20,color=
white],


[sideframe21,'Block',point=S21,xlength=4+2*(6.125*tan(37.5*3.1415927/180)),ylength=12,zlength=0.0
1,color=white],

[sideframe22,'Block',point=MakeTranslations(s22,0,0,20/2),xlength=0.01,ylength=12,zlength=20,colo
r=white],

[sideframe23,'Block',point=MakeTranslations(s23,0,0,20/2),xlength=0.01,ylength=12,zlength=20,colo
r=white],


[wedge21,'Block',point=Wedge21_1,xlength=(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.0
5,color=blue],

[wedge21,'Block',point=MakeTranslations(wedge21,(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlen
gth=0.05,ylength=10.5,zlength=6.125,color=blue],

[wedge21,'Block',point=MakeTranslations(wedge21,0,0,6.125/2),orient=MakeRotations([37.5*3.1415927
/180,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6.125^2),color=
blue],

              [wedge22,'Block',point=Wedge22_1,xlength=-
(6.125*tan(37.5*3.1415927/180)),ylength=10.5,zlength=0.05,color=blue],
              [wedge22,'Block',point=MakeTranslations(wedge22,-
(6.125*tan(37.5*3.1415927/180))/2,0,6.125/2),xlength=0.05,ylength=10.5,zlength=6.125,color=blue],

[wedge22,'Block',point=MakeTranslations(wedge22,0,0,6.125/2),orient=MakeRotations([-
37.5*3.1415927/180,2]),xlength=0.05,ylength=10.5,zlength=sqrt((6.125*tan(37.5*3.1415927/180))^2+6
.125^2),color=blue],


[sideframe21,'Cylinder',point=MakeTranslations(s21,(6.125*tan(37.5*3.1415927/180))/2+2,0,(VectorL
ength(FindTranslation(Spring21_1,Spring21_2)))/2),radius=0.5,length=VectorLength(FindTranslation(
Spring21_1,Spring21_2)),color=green],

[sideframe21,'Cylinder',point=MakeTranslations(s21,0,0,(VectorLength(FindTranslation(Spring22_1,S
pring22_2)))/2),radius=0.5,length=VectorLength(FindTranslation(Spring22_1,Spring22_2)),color=gree
n],
              [sideframe21,'Cylinder',point=MakeTranslations(s21,-
(6.125*tan(37.5*3.1415927/180))/2-
2,0,(VectorLength(FindTranslation(Spring23_1,Spring23_2)))/2),radius=0.5,length=VectorLength(Find
Translation(Spring23_1,Spring23_2)),color=green],

#Visual placement of points
              [world,'Sphere',point=W,radius=r,color=green],

              [sideframe,'Sphere',point=Sideframe1,radius=r,color=white],
              [sideframe,'Sphere',point=Sideframe2,radius=r,color=white],
              [sideframe,'Sphere',point=Sideframe3,radius=r,color=white],
              [sideframe,'Sphere',point=Sideframe4,radius=r,color=white],

              [sideframe21,'Sphere',point=S21,radius=0.05,color=white],
              [sideframe21,'Sphere',point=Sideframe22,radius=0.05,color=white],
              [sideframe21,'Sphere',point=Sideframe23,radius=0.05,color=white],
              [sideframe21,'Sphere',point=Sideframe24,radius=0.05,color=red],
              [sideframe21,'Sphere',point=Sideframe25,radius=0.05,color=red],

              [bolster,'Sphere',point=Bolster,radius=r,color=yellow],
              [bolster,'Sphere',point=Bolster_cg,radius=r,color=green],
              [bolster,'Sphere',point=Bolster1,radius=r,color=yellow],
              [bolster,'Sphere',point=Bolster2,radius=r,color=yellow],
```

```
                    [bolster,'Sphere',point=Bolster3,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster4,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster5,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster6,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster7,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster8,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster9,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster10,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster11,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster12,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster13,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster14,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster21,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster22,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster23,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster24,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster25,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster26,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster27,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster28,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster29,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster210,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster211,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster212,radius=r,color=blue],
                    [bolster,'Sphere',point=Bolster213,radius=r,color=yellow],
                    [bolster,'Sphere',point=Bolster214,radius=r,color=yellow],

                    [wedge1,'Sphere',point=Wedge1_1,radius=r,color=yellow],
                    [wedge1,'Sphere',point=Wedge1_2,radius=r,color=red],
                    [wedge1,'Sphere',point=Wedge1_3,radius=r,color=green],
                    [wedge1,'Sphere',point=Wedge1_4,radius=r,color=blue],
                    [wedge1,'Sphere',point=Wedge1_5,radius=r,color=red],
                    [wedge1,'Sphere',point=Wedge1_6,radius=r,color=green],
                    [wedge1,'Sphere',point=Wedge1_7,radius=r,color=blue],
                    [wedge1,'Sphere',point=Wedge1_cg,radius=r,color=green],

                    [wedge2,'Sphere',point=Wedge2_1,radius=r,color=yellow],
                    [wedge2,'Sphere',point=Wedge2_2,radius=r,color=red],
                    [wedge2,'Sphere',point=Wedge2_3,radius=r,color=green],
                    [wedge2,'Sphere',point=Wedge2_4,radius=r,color=blue],
                    [wedge2,'Sphere',point=Wedge2_5,radius=r,color=red],
                    [wedge2,'Sphere',point=Wedge2_6,radius=r,color=green],
                    [wedge2,'Sphere',point=Wedge2_7,radius=r,color=blue],
                    [wedge2,'Sphere',point=Wedge2_cg,radius=r,color=green],

                    [wedge21,'Sphere',point=Wedge21_1,radius=r,color=yellow],
                    [wedge21,'Sphere',point=Wedge21_2,radius=r,color=red],
                    [wedge21,'Sphere',point=Wedge21_3,radius=r,color=green],
                    [wedge21,'Sphere',point=Wedge21_4,radius=r,color=blue],
                    [wedge21,'Sphere',point=Wedge21_5,radius=r,color=red],
                    [wedge21,'Sphere',point=Wedge21_6,radius=r,color=green],
                    [wedge21,'Sphere',point=Wedge21_7,radius=r,color=blue],
                    [wedge21,'Sphere',point=Wedge21_cg,radius=r,color=green],

                    [wedge22,'Sphere',point=Wedge22_1,radius=r,color=yellow],
                    [wedge22,'Sphere',point=Wedge22_2,radius=r,color=red],
                    [wedge22,'Sphere',point=Wedge22_3,radius=r,color=green],
                    [wedge22,'Sphere',point=Wedge22_4,radius=r,color=blue],
                    [wedge22,'Sphere',point=Wedge22_5,radius=r,color=red],
                    [wedge22,'Sphere',point=Wedge22_6,radius=r,color=green],
                    [wedge22,'Sphere',point=Wedge22_7,radius=r,color=blue],
                    [wedge22,'Sphere',point=Wedge22_cg,radius=r,color=green],

                    [wedge1,'Sphere',point=Spring1_1,radius=r,color=yellow],
                    [wedge1,'Sphere',point=Spring1_2,radius=r,color=yellow],
                    [bolster,'Sphere',point=Spring2_1,radius=r,color=yellow],
                    [bolster,'Sphere',point=Spring2_2,radius=r,color=yellow],
                    [wedge2,'Sphere',point=Spring3_1,radius=r,color=yellow],
                    [wedge2,'Sphere',point=Spring3_2,radius=r,color=yellow]):

#Outputs
```

```
GeometryOutput(main=world,

parameters=[Fext=1,Fext2=1,muS=0.4,muB=0.4,ks=1000000,kb=1000000,omega=0,omega2=0,l0=10.25,l02=10
.25,k=1491,k2=22000,

Pi=3.1415927,r=0.05,z1=3.827,z2=6.125,z3=10.5,theta=37.5,m1=0.875,m2=936,m3=0.875,
                            Cdamp=1,toe=0],

states=[q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q13,q14,q15,q16,q17,q18,q19,q20,q21,q22,u1,u2,u3,u4,u5
,u6,u7,u8,u9,u10,u11,u13,u14,u15,u16,u17,u18,u19,u20],
       filename="fulltruckvardamp_ver2_4.geo"):
MotionOutput(ode=diffeqs,

parameters=[Fext=1,Fext2=1,muS=0.4,muB=0.25,ks=1000000,kb=1000000,omega=0,omega2=0,l0=10.25,l02=1
0.25,k=1491,k2=22000,

Pi=3.1415927,r=0.05,z1=3.827,z2=6.125,z3=10.5,theta=37.5,m1=0.875,m2=468,m3=0.875,
                    Cdamp=1,toe=0],
       states=[q1=4.35,q2=9.94,q3=4.35,q4=9.94,q5=9.94,q6=-4.35,q7=9.94,q8=-
4.35,q9=9.94,q10,q11,q13,q14,q15,q16,q17,q18,q19,q20,q21,q22,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u
13,u14,u15,u16,u17,u18,u19,u20],
       insignals=[rCBolsterWedge1_2=((FindTranslation(Wedge1_2,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_4=((FindTranslation(Wedge1_4,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_5=((FindTranslation(Wedge1_5,Bolster3) &oo rW3)&>0),
                   rCBolsterWedge1_7=((FindTranslation(Wedge1_7,Bolster3) &oo rW3)&>0),
                   rCSideframe1Wedge1_3=((FindTranslation(Sideframe1,Wedge1_3)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_4=((FindTranslation(Sideframe1,Wedge1_4)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_6=((FindTranslation(Sideframe1,Wedge1_6)                &oo
MakeTranslations(s1,1))&>0),
                   rCSideframe1Wedge1_7=((FindTranslation(Sideframe1,Wedge1_7)                &oo
MakeTranslations(s1,1))&>0),
                   rCBolsterWedge2_2=((FindTranslation(Wedge2_2,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_4=((FindTranslation(Wedge2_4,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_5=((FindTranslation(Wedge2_5,Bolster4) &oo rW6)&>0),
                   rCBolsterWedge2_7=((FindTranslation(Wedge2_7,Bolster4) &oo rW6)&>0),
                   rCSideframe2Wedge2_3=((FindTranslation(Sideframe2,Wedge2_3)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_4=((FindTranslation(Sideframe2,Wedge2_4)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_6=((FindTranslation(Sideframe2,Wedge2_6)                &oo
MakeTranslations(s2,1))&<0),
                   rCSideframe2Wedge2_7=((FindTranslation(Sideframe2,Wedge2_7)                &oo
MakeTranslations(s2,1))&<0),
                   rCBolsterWedge21_2=((FindTranslation(Wedge21_2,Bolster23) &oo rW11)&>0),
                   rCBolsterWedge21_4=((FindTranslation(Wedge21_4,Bolster23) &oo rW11)&>0),
                   rCBolsterWedge21_5=((FindTranslation(Wedge21_5,Bolster23) &oo rW11)&>0),
                   rCBolsterWedge21_7=((FindTranslation(Wedge21_7,Bolster23) &oo rW11)&>0),
                   rCSideframe21Wedge21_3=((FindTranslation(Sideframe22,Wedge21_3)            &oo
MakeTranslations(s21,1))&>0),
                   rCSideframe21Wedge21_4=((FindTranslation(Sideframe22,Wedge21_4)            &oo
MakeTranslations(s21,1))&>0),
                   rCSideframe21Wedge21_6=((FindTranslation(Sideframe22,Wedge21_6)            &oo
MakeTranslations(s21,1))&>0),
                   rCSideframe21Wedge21_7=((FindTranslation(Sideframe22,Wedge21_7)            &oo
MakeTranslations(s21,1))&>0),
                   rCBolsterWedge22_2=((FindTranslation(Wedge22_2,Bolster24) &oo rW14)&>0),
                   rCBolsterWedge22_4=((FindTranslation(Wedge22_4,Bolster24) &oo rW14)&>0),
                   rCBolsterWedge22_5=((FindTranslation(Wedge22_5,Bolster24) &oo rW14)&>0),
                   rCBolsterWedge22_7=((FindTranslation(Wedge22_7,Bolster24) &oo rW14)&>0),
                   rCSideframe22Wedge22_3=((FindTranslation(Sideframe23,Wedge22_3)            &oo
MakeTranslations(s22,1))&<0),
                   rCSideframe22Wedge22_4=((FindTranslation(Sideframe23,Wedge22_4)            &oo
MakeTranslations(s22,1))&<0),
                   rCSideframe22Wedge22_6=((FindTranslation(Sideframe23,Wedge22_6)            &oo
MakeTranslations(s22,1))&<0),
                   rCSideframe22Wedge22_7=((FindTranslation(Sideframe23,Wedge22_7)            &oo
MakeTranslations(s22,1))&<0)],
        filename="fulltruckvardamp_ver2_4.dyn"):
```

7.2. NUCARS Models

7.2.1. DAT file

```
Data file (.DAT) for NUCARS Version 2.0
=======================================

$\EQUILIBRIUM

SELECTION OF OUTPUT
===================

!\CURVATURE
!   1
!   8

!\SUPERELEVATION
!   1
!   8

  Body #    No. & List of DoF's
--------------------------------------------------------------------------
\DOF DISPLACEMENT
  16
    1   4   1 2 3 4
    3   6   1 2 3 4 5 6
    4   6   1 2 3 4 5 6

$\DOF VELOCITY
  18
    1   6   1 2 3 4 5 6
    3   6   1 2 3 4 5 6
    4   6   1 2 3 4 5 6

$\DOF ACCELERATION

$\DOF FORCE
  18
    1   6   1 2 3 4 5 6
    3   6   1 2 3 4 5 6
    4   6   1 2 3 4 5 6


\CONNECTION FORCE
16
 127.1 127.2 128.1 128.2 129.1 129.2 130.1 130.2
 119.1 119.2 121.1 121.2 123.1 123.2 125.1 125.2
 33
     1     2     3
    101.1 102.1 103.1 104.1
    101.2 102.2 103.2 104.2
    105.1 106.1 107.1 108.1
    109   110   111   112
    105.2 106.2 107.2 108.2
    105.3 106.3 107.3 108.3
    113   114   213   214
    117.3 118.3



\CONNECTION FORCE SUM
    2
    'Ld Trk Lft Vert Sprg Nest'  1   4  119.2 121.2 123.2 125.2  1.0 1.0 1.0 1.0
    'Ld Trk Lft Latl Sprg Nest'  1   4  119.1 121.1 123.1 125.1  1.0 1.0 1.0 1.0
```

```
   'Ld Trk Lft Spg Grp Vt For'  1   6  119.2 121.2 123.2 125.2 127.1 129.1  1.0 1.0 1.0 1.0 1.0 1.0
   'Ld Trk Rgt Spg Grp Vt For'  1   6  120.2 122.2 124.2 126.2 128.1 130.1  1.0 1.0 1.0 1.0 1.0 1.0
   'Ld Trk Lft Spg Grp Lat Fr'  1   6  119.1 121.1 123.1 125.1 127.2 129.2  1.0 1.0 1.0 1.0 1.0 1.0
   'Ld Trk Rgt Spg Grp Lat Fr'  1   6  120.1 122.1 124.1 126.1 128.2 130.2  1.0 1.0 1.0 1.0 1.0 1.0
!    'LEAD TRK SPG GRP ROLL MOM'  2   2  13  14  39.5  -39.5

\CONNECTION STROKE
 12
 127.1 127.2 128.1 128.2 129.1 129.2 130.1 130.2
 119.2 121.2 123.2 125.2
   35
   119.2 120.2
     1     2     3
   101.1 102.1 103.1 104.1
   101.2 102.2 103.2 104.2
   105.1 106.1 107.1 108.1
   109   110   111   112
   105.2 106.2 107.2 108.2
   105.3 106.3 107.3 108.3
   113   114   213   214
   117.3 118.3


\CONNECTION VELOCITY
  8
 127.1 127.2 128.1 128.2 129.1 129.2 130.1 130.2
 35
   119.2 120.2
     1     2     3
   101.1 102.1 103.1 104.1
   101.2 102.2 103.2 104.2
   105.1 106.1 107.1 108.1
   109   110   111   112
   105.2 106.2 107.2 108.2
   105.3 106.3 107.3 108.3
   113   114   213   214
   117.3 118.3

$\PERCENT STATIC LOAD
   2
   13  14

$\AXLE SPIN SPEED
   2
   1  2
   4  4

$\AXLE YAW TORQUE
   4
   1  2  3  4

$\AXLE ROLL TORQUE
   4
   1  2  3  4

$\AXLE SPIN TORQUE
   8
   1  2  1  2  1  2  1  2
   1  1  2  2  3  3  4  4

$\LONGITUDINAL W/R FORCE
   8
   1  2  1  2  1  2  1  2
   1  1  2  2  3  3  4  4

$\LATERAL W/R FORCE
   8
   1  2  1  2  1  2  1  2
   1  1  2  2  3  3  4  4
```

```
$\VERTICAL W/R FORCE
   8
   1  2  1  2  1  2  1  2
   1  1  2  2  3  3  4  4

$\PERCENT STATIC WHEEL LOAD
   4
   1  2  1  2
   1  1  2  2

$\WHEEL L/V RATIO
   8
   1  2  1  2  1  2  1  2
   1  1  2  2  3  3  4  4

$\AXLE SUM L/V RATIO
   4
   1  2  3  4

$\SUMMED L/V RATIO
   4
   2  1  1  2
   2  2  1  2
   2  1  3  4
   2  2  3  4

$\LATERAL RAIL DEFLECTION
   2
   1  2
   1  1

$\LATERAL RAIL POSITION
   2
   1  2
   1  1

$\VERTICAL RAIL DEFLECTION
   4
   1  2  1  2
   1  1  2  2

$\VERTICAL RAIL POSITION
   2
   1  2
   1  1

$\CONTACT LONGITUDINAL FORCE
   4
   1  2  3  4
   1  1  1  1

$\CONTACT LATERAL FORCE
   4
   1  2  3  4
   1  1  1  1

$\CONTACT NORMAL FORCE
   4
   1  2  3  4
   1  1  1  1

$\CONTACT ANGLE
   4
   1  2  3  4
   1  1  1  1

$\CONTACT ROLLING RADIUS
   4
   1  2  3  4
   1  1  1  1
```

```
$\W/R WEAR INDEX
   16
   1  2  3  4  1  2  3  4  1  2  3  4  1  2  3  4
   1  1  1  1  2  2  2  2  3  3  3  3  4  4  4  4

$\CONNECTION POWER DISSIPATION
   2
   13  14

$\TOTAL CONNECTION POWER DISSIPATION

$\TOTAL W/R RESISTANCE

$\RAIL WEAR INDEX
   4  1.0
   1  2  3  4

  Body #  '   A 25 Character Name   '     Posn in X, Y, & Z     DoF  Type
-----------------------------------------------------------------------------
$\TRANSDUCER
  4
! vertical car body displacements
!   1     'Car Body Lead            '    25.0     0.0    68.1     3    1
!   1     'Car Body Middle          '  -278.0     0.0    68.1     3    1
!   1     'Car Body Trail           '  -581.0     0.0    68.1     3    1
! roll gyros
  1     'Car Body Lead Roll Gyro  '    25.0     0.0    68.1     4    1
!   1     'Car Body Mid Roll Gyro   '  -278.0     0.0    68.1     4    1
  1     'Car Body Trail Roll Gyro '  -581.0     0.0    68.1     4    1
! lateral accelerometers at leading and trailing truck center plates
  1     'Car Body Lead Latrl Accel'   -35.0     0.0    25.0     2    3
  1     'Car Body Trail Latl Accel'  -521.0     0.0    25.0     2    3
!   8     'Axl1 Left Whl Lat Dsplmnt'     0.0   -29.5     0.0     2    1
```

7.2.2. INP file

```
Input file (.INP) for NUCARS 2006
========================================
Yaw Rotation of Bolster as an input to the system

\INPUT TITLE
Sideframe Displacement as an Input

Give the number of inputs and for each, the input number, the number of input segments,
and an indicator determining whether it is time or distance based, ITD = 1 for time, 2 for distance.
For each segment, provide the segment number, the shape function number, 0 = NULL, 1 = CUSP,
2 = BEND, 3 = SIN, 4 = SSIN, the time or distance at the segment end, in seconds or feet and the shape
base level at the start and end of the segment, pounds or inches. Coef1 & Coef2 are two coefficients that
in general determine the wavelength and repeat distance of the shape.
In the case of the swept sine SSIN they are the start and end frequency in Hz.
Finally, the function amplitude, in pounds or inches for the English system and
Newton-meters or Newtons for the metric system.
-----------------------------------------------------------------------
Segment Shape  Segment  End Start & End Base  Coef.1  Coef.2  Amp.
-----------------------------------------------------------------------
\INPUT HISTORY DATA
!Number of individual input histories
6
! Unique History number one, number of segments/history, and time=1/distance=2
1  1  1
! History 1 (Longitudinal) is Null for this input
   1     0    1.0       0.0         0.0     0.0     0.0    0.0
! History 2 (Lateral) is Null for this input
2  1  1
   1     0    1.0       0.0         0.0     0.0     0.0    0.0
! History 3 (Vertical) is Null for this input
3  1  1
   1     4    2000.0       0.0         0.0    0.477    0.477   1
! History 4 (Roll) is Null for this input
4  1  1
   1     0    1.0       0.0         0.0     0   0   0
! History 5 (Pitch) is Null for this input
5  1  1
   1     0    1.0       0.0         0.0     0.0     0.0    0.0
! History 6 (Yaw) is defined as sine with nat'l freq = 2 rad/sec., with amp=0.0031645rad
6  1  1
   1     0    1.0       0.0        0.0    0     0   0
```

7.2.3. Half-truck Variably Damped Model SYS file

```
System file (.SYS) for NUCARS Version 2006
=========================================

\SYSTEM TITLE
Bolster+1/2 Car body combo - Wedge - Sideframe as Input body, Type 6.9 Wedge Connection

Details About Model
 - Half Car body = Loaded 100T Hopper Car
 - Toe angle of 0.2degrees
 - Bolster + Car body constrained to Lateral & Vertical Translation & Roll
 - No Longitudinal connections between sideframe & ground


09/15/06  Type 6.9 3dof stick slip wedge - hollow column face wear, constant damped
        Values for longitudinal Gib clearance not confirmed,may require adjustment
        Includes empirical values for truck warp restraint from LHOPR-04

Give the number of bodies, then for each, list the number, name, up to 15
characters in single quotes, and c.g. position, relative to a chosen datum,
followed by the number and list of degrees of freedom required (from 1=x,
2=y, 3=z, 4=phi, 5=theta, 6=psi, 7=epsx, 8=epsy, 9=epsz), and the mass and
inertias in roll, pitch, and yaw.  The degrees of freedom required for each
axle are 2, 3, 4, and 6.  A longitudinal degree of freedom, 1, is optional.


! Revised by Nick Wilson  10/25/07
! Halfcar/boslter mass. Constrained in yaw na dpitch.   Note may need to allow yaw motion
! and add in yaw stiffnes between boy and ground to allow wedges to work a bit.  Need to discuss what you
! have in MAMBO model
! Need to verify the mass and moments of inertia to match MATLAB model and also check against reality.
!
! Not completely checked out - this is just a start see various comments marked NGW in file
!
! More adjustments by NGW 3/22/08
!  inclcude static load in Vertical PWLs for main coils and wedge control coils
!  change to some hollowing with PWL for toe

 Body #  ' 15 Char Name  '         C.G. Posn in X, Y, & Z
                 No. & DoF List          Mass, Roll, Pitch, & Yaw Inertia
---------------------------------------------------------------------------
\BODY DATA
   1
    1   'Half Car+Bolst'         -35.0       0.0      83.0
              4   1  2  3  4          319.175    0.92276E06   0.8350204E07   0.835276E07
! Sideframes changed to input bodies to match MATLAB Simulation
!    3   'Lead Lft Sframe'         -35.0      39.5      18.0
!       6   1  2  3  4  5  6    2.98    9.0E2     1.37E03   1.37E03
!    4   'Lead Rgt Sframe'         -35.0     -39.5      18.0
!       6   1  2  3  4  5  6    2.98    9.0E2     1.37E03   1.37E03

\INPUT BODY DATA
  2
    3  'Lt SFrm Inp Bdy'  -35.0   39.5   18.0   6   1 2 3 4 5 6    1 2 3 4 5 6    .F.
    4  'Rt Sfrm Inp Bdy'  -35.0  -39.5   18.0   6   1 2 3 4 5 6    1 2 3 4 5 6    .F.

\CONNECTION DATA
  14
! No centerplate or contact bearings btwn car body & bolster reqd
! Long.,Pitch, and Yaw bolster to sideframe connections
  117   'Ld L Bol-SF LPY'   1.1   1  3 -35.0  39.5 18.0  3  1 5 6  8 9 10
  118   'Ld R Bol-SF LPY'   1.1   1  4 -35.0 -39.5 18.0  3  1 5 6  8 9 10
! Vertical bolster to sideframe connections split into 4 seperate springs
! at each nest, dropped down 5"
  119   'Ld L Bol-SF V 1'   1.3   1  3 -31.25  35.75 13.0  3  2 3 4  11 12 13
```

```
  120   'Ld R Bol-SF V 1'   1.3   1  4 -31.25 -35.75 13.0  3  2 3 4  11 12 13
  121   'Ld L Bol-SF V 2'   1.3   1  3 -31.25  43.25 13.0  3  2 3 4  11 12 13
  122   'Ld R Bol-SF V 2'   1.3   1  4 -31.25 -43.25 13.0  3  2 3 4  11 12 13
  123   'Ld L Bol-SF V 3'   1.3   1  3 -38.75  35.75 13.0  3  2 3 4  11 12 13
  124   'Ld R Bol-SF V 3'   1.3   1  4 -38.75 -35.75 13.0  3  2 3 4  11 12 13
  125   'Ld L Bol-SF V 4'   1.3   1  3 -38.75  43.25 13.0  3  2 3 4  11 12 13
  126   'Ld R Bol-SF V 4'   1.3   1  4 -38.75 -43.25 13.0  3  2 3 4  11 12 13
! 3D Stick-Slip Friction wedge connection between bolster and sideframe
! May need to experiment with 2 type 6.9 elements per wedge in order to capture the wedging
! effects of the Bolster roation relative to the sidframe NGW
  127   'Ld Bol-SF LL Wg'   6.9   1   3   -26.5    39.5    18.0   3 1 2  14
  128   'Ld Bol-SF LR Wg'   6.9   1   4   -26.5   -39.5    18.0   3 1 2  14
  129   'Ld Bol-SF TL Wg'   6.9   1   3   -43.5    39.5    18.0   3 1 2  114
  130   'Ld Bol-SF TR Wg'   6.9   1   4   -43.5   -39.5    18.0   3 1 2  114


For each connection characteristic, list its number, identification numbers
for the piecewise linear stiffness and damping characteristics, respectively,
zero if absent, and the force, moment, or stroke limits in extn and compn,
(if no limit exists, set the values outside the expected range).

 Pair #  Stiffness & Damping  F/S-extn.  F/S-comp.    K/D-parameters
--------------------------------------------------------------------------
\CHARACTERISTIC DATA
! ********************** Bolster to Sideframe Connections *******************
! Longitudinal
   8   11          12        1.0E09      -1.0E09
! Pitch stiffness and stops
   9   13          0        1.0E09      -1.0E09
! Warp/torsion
  10   14          0        1.0E09      -1.0E09
! Lateral stiffness and stops
  11   15          0        1.0E09      -1.0E09
! Vertical Springs
  12   16          0        1.0E09      -1.0E09
! Dummy roll characteristic for type 1.3 connection
  13    0          0        0.0         -1.0E09
! # 4 is a 6.3 wedge element with pwl numbers, wedge angle, force, LVB,
! and friction, Constant damped truck is 1979 lb/in in the control coils
! at zero wedge rise the control coils are compressed 1.8393 inches
! 0.0 inch wedge rise
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0        0       37.5       3.640E03  1.0E04  0.40
! 0.25 inch wedge rise (1.8393 - 0.25 = 1.5893 * 1979 = 3145)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0        0       37.5       3.1450E03 1.0E04  0.40
! 0.375 inch wedge rise (1.8393 -.375=1.4643*1979=2898)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!    14      0        0       37.5       2.898E03  1.0E04  0.40
! modified for new type 6.7, MU1 for slope Mu2 for Face, T=Toe-out, F=Toe-in
!   Ch #  Pwl Stf Pwl Damp  Wedge Ang  Force    LB      Mu1    Mu2    Toe
!   14      0        0       37.5   2.898E3  1.0E04  0.40   0.40   .T.
! 0.50 inch wedge rise (1.8393 - 0.50 = 1.3393 * 1979 = 2650)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0        0       37.5       2.650E03 1.0E04  0.40
! 0.75 inch wedge rise (1.8393 - 0.75 = 1.0893 * 1979 = 2155)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0        0       37.5       2.155E03 1.0E04  0.40
!
! Option for a VARIABLE DAMPED TRUCK design
!   Ch #  Pwl Stf Pwl Damp  Wedge Ang  Force    LB      Mu1    Mu2    Toe
!    14     17        0     32.0     0.0    1.0E04   0.40   0.40   .T.
!
!  NEW Type 6.9 3D Stick-Slip Wedge
!  Pwl-toe is for cloumn wear face
!  V/C = .F. for constant damped
!  Ka and Ca are force accumulator stiffness and damping
!  MU1 for slope Mu2 for Face
!  XDIR = 1.d0 for leading wedges, -1.d0 for trailing wedges
!  Column Face Normal Force limit FWNMAX = 1.0D5
```

```
!
! mu1 and mu2 to match MAMBO model
! The following is for Toed out wedge but no PWL for toe
!   Ch #  Pwl-Stf Pwl-Toe  Wedge Ang  V/C    Ka     Ca     Mu1  Mu2  XDIR  FWNMAX
!    14      17     0      37.5     .T.   1.0E6  1.0E3   0.25  0.4  1.D0   1.0D5
!   114      17     0      37.5     .T.   1.0E6  1.0E3   0.25  0.4 -1.D0   1.0D5
! the following uses PWL 24 for toe-in/out
    14      17     24      37.5     .T.   1.0E6  1.0E3   0.25  0.4  1.D0   1.0D5
   114      17     24      37.5     .T.   1.0E6  1.0E3   0.25  0.4 -1.D0   1.0D5


-----------------------------------------------------------------------------
\PWL DATA
! ********************** Bolster to Sideframe Connections *******************
! Longitudinal
! Includes shear stiffness of coils - same stiffness as lateral 17,800 lb/in
! Longitudinal clearances +/- 0.125 inches
! some damping for integration stability
!   11    3     0.0     1.225E3   1.02225E5
!               0.0     0.125     0.60
!
! This should be only the shear stiffness of the coils plus the gap clearance stiffness NGW
! variable Damped truck
    11    3     0.0     1.808E3   1.01808E5
                        0.0     0.125     0.60
    12    2     0.0     1.E3
                        0.0     1.0
! Pitch Stiffness and Stops Mid Tolerance is Approx +/- 2.4 Degrees=0.042mRad
    13    3     0.0     0.0   6.40E7
              0.0     0.042  1.042
! ******************* Warp resistance ****************************************
! these values are empirical to include the missing effects of wedging action
! worn truck
! Warp resistance for bolster to sideframe
!   14    3     0.0     5.25E04   6.924E05
!               0.0     0.030     0.040
! New truck
! Warp resistance for bolster to sideframe
! This should probably be only the shear stiffness of the coils plus the gap clearance stiffness
! Wedges should provide the rest? NGW
! variable Damped truck
   14    3     0.0     2.55E05   8.95E05
              0.0     0.030     0.040
! Stiff H-frame truck
! Warp resistance for bolster to sideframe
!   14    3     0.0     1.275E06  1.915E06
!               0.0     0.030     0.040
! Lateral Stiffness of bolster to sideframe connection divided by 4
! 17,800 lb/in
!   15    3     0.0     2.225E3   1.022E5
!                       0.0     0.50      0.60
! Lateral Stiffness of bolster to sideframe connection divided by 4, Variable Damped truck
! 14,666 lb/in
! This should be only the shear stiffness of the coils plus the gap clearance stiffness NGW
! variable Damped truck
    15    3     0.0     1.808E3   1.018E5
                        0.0     0.50      0.60
! Vertical Secondary Suspension
! 9-D5 outers and 5-D5 inners
!   16     5    -1.954E5 -9.5414E4  -3.50E2    0.0      0.0
!              -3.7875  -3.6875     0.0      0.0625   1.0
! stiffness divided by 4
!   16     5    -0.4885E5 -2.38535E4 -8.75E1    0.0      0.0
!              -3.7875  -3.6875     0.0       0.0625   1.0
! with initial offsets calculated for loaded static weight of -15403.78
!   16     6    -0.4885E5 -2.38535E4  -1.540378E4 -8.75E1   0.0      0.0
!              -1.39747  -1.29747     0.0         2.3900   2.4525   4.0
!
! Spring nest for the variable damped option (6 D5 Outers and 7 D5 Inners)
!  16     6    -4.48E+04 -1.98E+04   -1.22E+04   -1.23E+02 0.0      0.0
```

```
!             -1.5171   -1.4171      0.0         2.2704   2.3329   3.2079
! MAMBO model uses spring stiffness of 3000lb/in for Variably Damped Wedge WE NEED TO DISCUSS THIS VALUE !!NGW
!  OK for Mambo - NUCARS comparisons but if both MAMBO and NUCARS simulations are using full Half car mass
! then we need to put in real spring stiffness values other wise dynamics will be all wrong  NGW
! Stiffness from MAMBO 22417 lb/in.   Divided by 4 for the 4 springs, with static load => 5604.25 lb/in
! static deflection calculation, including main coils plus 2 control coils:
! Total stiffness = 2*1491 + 22417.8 = 25399 lb/in
! Static load per side = 386.09 * 319.175/2 = 61615 lbs =>
! static deflection = 61615/25399 = 2.426 in
!
! Static load for one main coil = -2.426 in * 5604.25 lb/in = -13595.9 lb
! PWL for 5604 lb/in
!   16      2     -1.920015e4  -1.35959E4
!               -1.0          0.0
! revised static load to match MAMBO for static load in wedge control coils of 1000 lb
! therefore static load in main coils is (61615 - 2000)/4 = 14904 lb
! include lift off
! PWL for 5604 lb/in
  16     3     -1.4904e4   0.0     0.0
              0.0        2.6595  5.0
!
!***********Optional control coils for a variably damped truck***************
! 2 B353 outers and 2 B354 inners, zero wedge rise - this is different than LHOPR-04
!  17      8    -5.903E+04 -9.0314E+03 -6.303E+03 -1.89E+02 -1.767E3 -72.25   0.0    0.0
!             -1.5085    -1.4085      0.0         2.279    2.3415   3.2165  3.3415 4.3415
! MAMBO model uses spring stiffness of 3000lb/in for Variably Damped Wedge WE NEED TO DISCUSS THIS VALUE !!NGW
!  OK for Mambo - NUCARS comparisons but if both MAMBO and NUCARS simualtions are using full Half car mass
! MAMBO model uses spring stiffness of 3000lb/in for Variably Damped Wedge
!
! assume control coil stiffness = 1491 lb/in
! Static load for one control coil with no wedge rise = -2.426 in * 1491 lb/in = -3617 lb
! PWL for 1491 lb/in
!   17      2      -5108.0   -3617.0
!                -1.0       0.0
! revised static load to match MAMBO: 1000 lbs per wedge, with lift off
! PWL for 1491 lb/in
  17     3      -1000.0   0.0     0.0
               0.0      0.67    1.0
!****************** Type 6.9 Wedge PWLs ********************
! Control Coil Stiffness - 100 ton constant damped truck theoretical data
! 3640 lbs at 0.0 wedge rise.   1908.3 lbs at 0.875 rise, 1979.3 lbs/in
!   23      2     -3640.2  -1908.3
!                0.0      0.875
! Column wear face - Hypothetical hollow, 0.1 inch deep centered on -0.75    -+toein +-toeout
 24      3     -0.015708   0.0    0.015708
            -4.5     0.0        4.5
!24      3     0.015708   0.0    -0.015708
```

7.2.4. Half-truck Constantly Damped Model SYS file

```
System file (.SYS) for NUCARS Version 2006
=========================================

\SYSTEM TITLE
Bolster+1/2 Car body combo - Wedge - Sideframe as Input body, Type 6.9 Wedge Connection - constant damped

Details About Model
 - Half Car body = Loaded 100T Hopper Car
 - Toe angle of 0.2degrees
 - Bolster + Car body constrained to Lateral & Vertical Translation & Roll
 - No Longitudinal connections between sideframe & ground


09/15/06  Type 6.9 3dof stick slip wedge - hollow column face wear, constant damped
        Values for longitudinal Gib clearance not confirmed,may require adjustment
        Includes empirical values for truck warp restraint from LHOPR-04

Give the number of bodies, then for each, list the number, name, up to 15
characters in single quotes, and c.g. position, relative to a chosen datum,
followed by the number and list of degrees of freedom required (from 1=x,
2=y, 3=z, 4=phi, 5=theta, 6=psi, 7=epsx, 8=epsy, 9=epsz), and the mass and
inertias in roll, pitch, and yaw.  The degrees of freedom required for each
axle are 2, 3, 4, and 6.  A longitudinal degree of freedom, 1, is optional.


! Revised by Nick Wilson  10/25/07
! Halfcar/boslter mass. Constrained in yaw na dpitch.   Note may need to allow yaw motion
! and add in yaw stiffnes between boy and ground to allow wedges to work a bit.  Need to discuss what you
! have in MAMBO model
! Need to verify the mass and moments of inertia to match MATLAB model and also check against reality.
!
! Not completely checked out - this is just a start see various comments marked NGW in file
!
! More adjustments by NGW 3/22/08
!  inclcude static load in Vertical PWLs for main coils and wedge control coils
!  change to NO-TOE with PWL for toe
!
! Constant damped version with No-toe PWL NGW 3/31/08

 Body #  ' 15 Char Name  '        C.G. Posn in X, Y, & Z
              No. & DoF List          Mass, Roll, Pitch, & Yaw Inertia
--------------------------------------------------------------------------
\BODY DATA
  1
    1   'Half Car+Bolst'         -35.0       0.0      83.0
            4   1  2  3  4          468.175    0.92276E06   0.8350204E07   0.835276E07
! Sideframes changed to input bodies to match MATLAB Simulation
!    3   'Lead Lft Sframe'         -35.0      39.5      18.0
!      6   1  2  3  4  5  6    2.98    9.0E2     1.37E03   1.37E03
!    4   'Lead Rgt Sframe'         -35.0     -39.5      18.0
!      6   1  2  3  4  5  6    2.98    9.0E2     1.37E03   1.37E03

\INPUT BODY DATA
  2
    3  'Lt SFrm Inp Bdy'  -35.0   39.5   18.0   6   1 2 3 4 5 6    1 2 3 4 5 6    .F.
    4  'Rt Sfrm Inp Bdy'  -35.0  -39.5   18.0   6   1 2 3 4 5 6    1 2 3 4 5 6    .F.

\CONNECTION DATA
  14
! No centerplate or contact bearings btwn car body & bolster reqd
! Long.,Pitch, and Yaw bolster to sideframe connections
  117   'Ld L Bol-SF LPY'   1.1    1  3 -35.0  39.5 18.0  3  1 5 6  8 9 10
  118   'Ld R Bol-SF LPY'   1.1    1  4 -35.0 -39.5 18.0  3  1 5 6  8 9 10
! Vertical bolster to sideframe connections split into 4 seperate springs
```

```
! at each nest, dropped down 5"
  119   'Ld L Bol-SF V 1'   1.3   1  3 -31.25  35.75 13.0  3  2 3 4  11 12 13
  120   'Ld R Bol-SF V 1'   1.3   1  4 -31.25 -35.75 13.0  3  2 3 4  11 12 13
  121   'Ld L Bol-SF V 2'   1.3   1  3 -31.25  43.25 13.0  3  2 3 4  11 12 13
  122   'Ld R Bol-SF V 2'   1.3   1  4 -31.25 -43.25 13.0  3  2 3 4  11 12 13
  123   'Ld L Bol-SF V 3'   1.3   1  3 -38.75  35.75 13.0  3  2 3 4  11 12 13
  124   'Ld R Bol-SF V 3'   1.3   1  4 -38.75 -35.75 13.0  3  2 3 4  11 12 13
  125   'Ld L Bol-SF V 4'   1.3   1  3 -38.75  43.25 13.0  3  2 3 4  11 12 13
  126   'Ld R Bol-SF V 4'   1.3   1  4 -38.75 -43.25 13.0  3  2 3 4  11 12 13
! 3D Stick-Slip Friction wedge connection between bolster and sideframe
! May need to experiment with 2 type 6.9 elements per wedge in order to capture the wedging
! effects of the Bolster roation relative to the sidframe NGW
  127   'Ld Bol-SF LL Wg'   6.9   1   3   -26.5    39.5    18.0   3 1 2  14
  128   'Ld Bol-SF LR Wg'   6.9   1   4   -26.5   -39.5    18.0   3 1 2  14
  129   'Ld Bol-SF TL Wg'   6.9   1   3   -43.5    39.5    18.0   3 1 2  114
  130   'Ld Bol-SF TR Wg'   6.9   1   4   -43.5   -39.5    18.0   3 1 2  114
```

For each connection characteristic, list its number, identification numbers
for the piecewise linear stiffness and damping characteristics, respectively,
zero if absent, and the force, moment, or stroke limits in extn and compn,
(if no limit exists, set the values outside the expected range).

```
 Pair #  Stiffness & Damping  F/S-extn.  F/S-comp.    K/D-parameters
----------------------------------------------------------------------------
\CHARACTERISTIC DATA
! ********************** Bolster to Sideframe Connections ******************
! Longitudinal
   8  11          12       1.0E09      -1.0E09
! Pitch stiffness and stops
   9  13          0        1.0E09     -1.0E09
! Warp/torsion
  10  14          0        1.0E09      -1.0E09
! Lateral stiffness and stops
  11  15          0        1.0E09      -1.0E09
! Vertical Springs
  12  16          0        1.0E09      -1.0E09
! Dummy roll characteristic for type 1.3 connection
  13   0          0        0.0         -1.0E09
! # 4 is a 6.3 wedge element with pwl numbers, wedge angle, force, LVB,
! and friction, Constant damped truck is 1979 lb/in in the control coils
! at zero wedge rise the control coils are compressed 1.8393 inches
! 0.0 inch wedge rise
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0         0       37.5      3.640E03  1.0E04  0.40
! 0.25 inch wedge rise (1.8393 - 0.25 = 1.5893 * 1979 = 3145)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0         0       37.5      3.1450E03 1.0E04  0.40
! 0.375 inch wedge rise (1.8393 -.375=1.4643*1979=2898)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0         0       37.5      2.898E03  1.0E04  0.40
! modified for new type 6.7, MU1 for slope Mu2 for Face, T=Toe-out, F=Toe-in
!   Ch #  Pwl Stf Pwl Damp  Wedge Ang  Force    LB      Mu1    Mu2    Toe
!   14      0         0       37.5  2.898E3  1.0E04  0.40   0.40   .T.
! 0.50 inch wedge rise (1.8393 - 0.50 = 1.3393 * 1979 = 2650)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0         0       37.5      2.650E03 1.0E04  0.40
! 0.75 inch wedge rise (1.8393 - 0.75 = 1.0893 * 1979 = 2155)
!   Ch #  Pwl Stiff Pwl Damp Wedge Angle    Force   LB      Mu
!   14      0         0       37.5      2.155E03 1.0E04  0.40
!
! Option for a VARIABLE DAMPED TRUCK design
!   Ch #  Pwl Stf Pwl Damp  Wedge Ang  Force    LB      Mu1    Mu2    Toe
!   14     17        0      32.0     0.0    1.0E04   0.40   0.40   .T.
!
! Type 6.9 CONSTANT DAMPED Truck !
!
!  NEW Type 6.9 3D Stick-Slip Wedge
!  Pwl-toe is for cloumn wear face
!  V/C = .F. for constant damped
```

```
!  Ka and Ca are force accumulator stiffness and damping
!  MU1 for slope Mu2 for Face
!  XDIR = 1.d0 for leading wedges, -1.d0 for trailing wedges
!  Column Face Normal Force limit FWNMAX = 1.0D5
!
! mu1 and mu2 to match MAMBO model
! The following is for Toed out wedge but no PWL for toe
!   Ch #  Pwl-Stf Pwl-Toe  Wedge Ang  V/C    Ka     Ca     Mu1  Mu2  XDIR  FWNMAX
    14      17     0      37.5     .F.  1.0E9  1.0E5   0.25  0.4  1.D0   1.0D8
   114      17     0      37.5     .F.  1.0E9  1.0E5   0.25  0.4 -1.D0   1.0D8
! the following uses PWL 24 for toe-in/out
!   14      17     24      37.5     .F.  1.0E6  1.0E3   0.25  0.4  1.D0   1.0D5
!   114      17     24      37.5     .F.  1.0E6  1.0E3   0.25  0.4 -1.D0   1.0D5


---------------------------------------------------------------------------
\PWL DATA
! ********************** Bolster to Sideframe Connections *******************
! Longitudinal
! Includes shear stiffness of coils - same stiffness as lateral 17,800 lb/in
! Longitudinal clearances +/- 0.125 inches
! some damping for integration stability
!    11     3     0.0     1.225E3   1.02225E5
!                 0.0      0.125     0.60
!
! This should be only the shear stiffness of the coils plus the gap clearance stiffness NGW
! variable Damped truck
    11     3     0.0     1.808E3   1.01808E5
                         0.0      0.125     0.60
    12     2     0.0     1.E3
                         0.0      1.0
! Pitch Stiffness and Stops Mid Tolerance is Approx +/- 2.4 Degrees=0.042mRad
    13     3     0.0     0.0   6.40E7
               0.0      0.042  1.042
! ******************** Warp resistance ******************************************
! these values are empirical to include the missing effects of wedging action
! worn truck
! Warp resistance for bolster to sideframe
!   14     3     0.0     5.25E04   6.924E05
!                0.0      0.030     0.040
! New truck
! Warp resistance for bolster to sideframe
! This should probably be only the shear stiffness of the coils plus the gap clearance stiffness
! Wedges should provide the rest? NGW
! variable Damped truck
   14     3     0.0     2.55E05   8.95E05
              0.0      0.030     0.040
! Stiff H-frame truck
! Warp resistance for bolster to sideframe
!   14     3     0.0     1.275E06  1.915E06
!                0.0      0.030     0.040
! Lateral Stiffness of bolster to sideframe connection divided by 4
! 17,800 lb/in
!   15     3     0.0     2.225E3   1.022E5
!                        0.0      0.50      0.60
! Lateral Stiffness of bolster to sideframe connection divided by 4, Variable Damped truck
! 14,666 lb/in
! This should be only the shear stiffness of the coils plus the gap clearance stiffness NGW
! variable Damped truck
    15     3     0.0     1.808E3   1.018E5
                         0.0      0.50      0.60
! Vertical Secondary Suspension
! 9-D5 outers and 5-D5 inners
!   16     5     -1.954E5 -9.5414E4  -3.50E2    0.0      0.0
!              -3.7875  -3.6875     0.0       0.0625   1.0
! stiffness divided by 4
!   16     5     -0.4885E5 -2.38535E4 -8.75E1    0.0      0.0
!              -3.7875  -3.6875     0.0       0.0625   1.0
! with initial offsets calculated for loaded static weight of -15403.78
!   16     6     -0.4885E5 -2.38535E4  -1.540378E4 -8.75E1   0.0      0.0
```

```
!               -1.39747  -1.29747     0.0         2.3900   2.4525   4.0
!
! Spring nest for the variable damped option (6 D5 Outers and 7 D5 Inners)
!  16      6     -4.48E+04 -1.98E+04   -1.22E+04   -1.23E+02 0.0      0.0
!              -1.5171   -1.4171      0.0         2.2704   2.3329   3.2079
! MAMBO model uses spring stiffness of 3000lb/in for Variably Damped Wedge WE NEED TO DISCUSS THIS VALUE !!NGW
!  OK for Mambo - NUCARS comparisons but if both MAMBO and NUCARS simulations are using full Half car mass
! then we need to put in real spring stiffness values other wise dynamics will be all wrong  NGW
! Stiffness from MAMBO 22417 lb/in.   Divided by 4 for the 4 springs, with static load => 5604.25 lb/in
! static deflection calculation, including main coils plus 2 control coils:
! Total stiffness = 2*1491 + 22417.8 = 25399 lb/in
! Static load per side = 386.09 * 319.175/2 = 61615 lbs =>
! static deflection = 61615/25399 = 2.426 in
!
! Static load for one main coil = -2.426 in * 5604.25 lb/in = -13595.9 lb
! PWL for 5604 lb/in
!   16      2     -1.920015e4  -1.35959E4
!               -1.0          0.0
! revised static load to match MAMBO for constant damped truck static load
! main control coils = 22418 lb/in per nest => 5604.5lb/in each for 4 coils
! therefore static load in main coils is (61615)/4 = 15404 lb
! include lift off
! PWL for 5604 lb/in
  16      3     -1.5404e4   0.0     0.0
               0.0        2.7485  5.0
!
!***********Optional control coils for a contant damped truck***************
!****************** Type 6.9 Wedge PWLs ********************
! Control Coil Stiffness - 100 ton constant damped truck theoretical data
! 900 lbs at static condition to match mambo.   1979.3 lbs/in
! for zero wedge rise coils are compressed 1.8393"
! therefore this sytem has 1.8393 - 0.4547 = 1.3846" wedge rise (worn out!)
! 4/3/08 set to 3625 lb static load (0 wedge rise) NGW
  17      2    -3625.0     0.0
               0.0     1.83
!****************** Type 6.9 Wedge PWLs ********************
! Column wear face - NO-TOE
!  18      3       0.0  0.0     0.0
!                -4.5  0.0     4.5
```