Nonlinear Investigation of the Use of Controllable Primary Suspensions to Improve Hunting in Railway Vehicles

by

Anant Mohan

Thesis submitted to the faculty of

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of Master of Science

in

Mechanical Engineering

Approved:

____
Mehdi Ahmadian, Chairman

________________	_________________
Scott L. Hendricks	Mehrdaad Ghorashi

June 20, 2003

Blacksburg, Virginia

Keywords: Hunting, Rail Vehicles, Hopf Bifurcation, Lateral Stability, Semi-Active Suspension Control

Nonlinear Investigation of the Use of Controllable Primary Suspensions to Improve Hunting in Railway Vehicles

by

Anant Mohan

Mehdi Ahmadian, Chairman

Mechanical Engineering

Abstract

Hunting is a very common instability exhibited by rail vehicles operating at high speeds. The hunting phenomenon is a self excited lateral oscillation that is produced by the forward speed of the vehicle and the wheel-rail interactive forces that result from the conicity of the wheel-rail contours and the friction-creep characteristics of the wheel-rail contact geometry. Hunting can lead to severe ride discomfort and eventual physical damage to wheels and rails.

A comprehensive study of the lateral stability of a single wheelset, a single truck, and the complete rail vehicle has been performed. This study investigates bifurcation phenomenon and limit cycles in rail vehicle dynamics. Sensitivity of the critical hunting velocity to various primary and secondary stiffness and damping parameters has been examined.

This research assumes the rail vehicle to be moving on a smooth, level, and tangential track, and all parts of the rail vehicle to be rigid. Sources of nonlinearities in the rail vehicle model are the nonlinear wheel-rail profile, the friction-creep characteristics of the wheel-rail contact geometry, and the nonlinear vehicle suspension characteristics. This work takes both single-point and two-point wheel-rail contact conditions into account.

The results of the lateral stability study indicate that the critical velocity of the rail vehicle is most sensitive to the primary longitudinal stiffness. A method has been developed to eliminate hunting behavior in rail vehicles by increasing the critical velocity of hunting beyond the operational speed range. This method involves the semi-active control of the primary longitudinal stiffness using the wheelset yaw displacement. This approach is seen to considerably increase the critical hunting velocity.

Acknowledgments

I am very grateful to my advisor, Dr. Mehdi Ahmadian, for his guidance and support throughout this thesis. To work with an advisor as supportive, understanding, and persevering as Dr. Ahmadian is a rare luxury.

Thanks also to Dr. Scott Hendricks and Dr. Mehrdaad Ghorashi for serving on my graduate committee. Had I not taken Dr. Hendricks's well taught class on nonlinear dynamics, I would surely have remained oblivious to this wonderful area of science.

I am very thankful to my parents for their steadfast support and faith in me. I feel very fortunate to have such caring parents.

Thanks also to my very good friends Christine and Rakesh for their encouragement every time that I felt despondent about the progress of my thesis.

Finally, thank you to Virginia Tech for giving me this opportunity to further my learning. Surely, there cannot be very many places on this planet where a student can obtain quality education while enjoying remarkable natural beauty!

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Nomenclature

a	half of track gage
b	half of wheelbase
C_0	secondary yaw viscous damping
C_{PX}	primary longitudinal damping
C_{PY}	primary lateral damping
C_{RAIL}	effective lateral rail viscous damping
C_{SY}	secondary lateral damping
$C_{S\Psi}$	secondary yaw damping
d_p	half of lateral spacing between primary longitudinal springs
f_{ij}^*	nominal creep coefficients (ij = 11, 12, 22, 33)
f_{11}	lateral creep coefficient
f_{12}	lateral / spin creep coefficient
f_{22}	spin creep coefficient
f_{33}	longitudinal creep coefficient
F_{CPX} , F_{CPY}	creep force in longitudinal, lateral contact patch direction
F'_{CPX} , F'_{CPY}	unlimited creep force in longitudinal, lateral contact patch
F_{CXi}	longitudinal track component of creep force at i^{th} contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact
F_{CYi}	lateral track component of creep force at i^{th} contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact
F_N	normal force
F_N^*	nominal normal force

F_{Ni} normal force at ith contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact

F_{NYi} lateral track component of normal force at ith contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact

F_{NZi} vertical track component of normal force at ith contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact

$F_{RAIL,L}$, $F_{RAIL,R}$ lateral rail reaction force at left, right rail

$F_R^{'}$ unlimited resultant creep force

F_{SUSPY} lateral suspension force

F_{SUSPZ} vertical suspension force

g acceleration due to gravity

h_{cs} vertical distance from secondary lateral suspension to carbody center of mass

I_{BZ} yaw principal mass moment of inertia of bolster

I_{CX} roll principal mass moment of inertia of carbody

I_{CZ} yaw principal mass moment of inertia of carbody

I_{FZ} yaw principal mass moment of inertia of truck frame

I_{WY} pitch principal mass moment of inertia of wheelset

I_{WZ} yaw principal mass moment of inertia of wheelset

K_{PX} primary longitudinal suspension stiffness

K_{PY} primary lateral suspension stiffness

K_{RAIL} effective lateral rail stiffness

K_{SY} secondary lateral suspension stiffness

K_{SZ} secondary vertical suspension stiffness

$K_{S\Psi}$ secondary yaw suspension stiffness

l_s half of truck center pin spacing

m_{RAIL} effective lateral rail mass

M_{CP} creep moment normal to contact patch

M_{CXi} longitudinal track component of creep force at i^{th} contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact

M_{CYi} lateral track component of creep force at i^{th} contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact

M_{CZi} vertical track component of creep force at i^{th} contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact

M_{SUSPX} roll suspension moment

M_{SUSPZ} yaw suspension moment

R_i rolling radius measured from wheelset spin axis to i^{th} contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact

R_0 rolling radius for centered wheelset (nominal rolling radius)

T_0 secondary yaw suspension breakaway torque

T_{COUL} coulomb friction yaw moment

V vehicle forward speed

V_C critical forward speed of the vehicle

m_B bolster mass

m_C carbody mass

m_F truck frame mass

m_V total vehicle mass

m_W wheelset mass

x longitudinal coordinate

y lateral coordinate

y_{FC}	flange clearance
$y_{RAIL,L}$, $y_{RAIL,R}$	lateral displacement of left, right rail
z	vertical coordinate
β	normalized unlimited resultant creep force
δ_i	contact angle at ith contact patch; i = L (left), R (right) for single-point contact; i = LT (left tread), LF (left flange), RT (right tread), RF (right flange) for two-point contact
ε	creep force saturation constant
$\dot{\theta}_w$	wheelset spin speed
λ	wheel conicity
μ	coefficient of friction between wheel and track
ξ_R	resultant creepage
ξ_{SPi}	spin creepage in normal contact patch direction at ith contact patch
ξ_{Xi}	longitudinal creepage at ith contact patch
ξ_{Yi}	lateral creepage at ith contact patch
ϕ_w	wheelset roll angle with respect to track plane
ψ	yaw angle

Chapter 1

Introduction

This chapter provides a broad introduction into the area of non-linear railway vehicle dynamics. This introduction includes an overview of the various mathematical models of a rail vehicle, the problem of hunting commonly faced during the operation of rail vehicles, critical velocity and its implication in non-linear rail vehicle dynamics, and the Hopf bifurcation phenomenon seen in rail vehicle dynamics. This chapter then provides a review of existing literature concerning the subject of this thesis, followed by the objectives, and a brief outline of the thesis.

1.1 Background

The study of vehicle dynamics is a difficult task. On tangent track at slower speeds of operation, rock-and-roll problems occur. At higher speeds, a vehicle may hunt or bounce severely. While negotiating curved track, wheels may climb the rail, excessive lateral forces may occur, and rails may roll over. In classification yards, damage to freight may occur due to coupling impacts. In over-the-road operations, freight may be damaged by excessive vehicle vibrations. In addition, the train consist may buckle laterally or vertically. High drawbar forces that develop under different operating conditions may cause a train separation.

The technique of mathematically modeling the train and track has been extensively used to understand dynamic interactions between the track and the train. These dynamic interactions vary with operating conditions, type of terrain, wheel and rail profiles, and climatic conditions. It would be an impossible task to construct a single mathematical model that could be universally addressed to all aspects of train-track interactions. However, the complicated dynamic behavior that results from these interactions can be studied by using various mathematical models, each of which concentrates on a specific area of interest.

Generally, in constructing a mathematical model for studying the dynamic behavior of vehicles, or a train consist, the components of the system are assumed to be rigid bodies. A rigid body has six dynamic degrees of freedom, which correspond to three displacements (longitudinal, lateral, and vertical), and three rotations (roll, pitch, and yaw). Because each dynamic degree of freedom results in a second-order coupled differential equations, 6 x N differential equations will be required to represent the system mathematically, in which N denotes the number of components in the system. Solutions for all of these differential equations are not only expensive, but many times are unnecessary. Therefore, it is important to establish the objective of a mathematical model.

It has been observed that a relatively weak coupling exists between the vertical and lateral motions of a vehicle and, therefore, that it may not be necessary to include the vertical degrees of freedom in the study of the lateral response of the vehicle or the lateral degrees of freedom in the vertical response. For the vertical response, it would be adequate to include the bounce, pitch, and roll degrees of freedom of the components. Correspondingly, for the lateral response, one may use the lateral, yaw, and roll degrees of freedom of the components. In studies of the longitudinal dynamic behavior, the longitudinal, pitch, and roll degrees of freedom of the components can be included in the model. Thus depending on the model's objective, the total dynamic degrees of freedom in the system can be considerably reduced. This operation not only reduced the computational cost but also makes the interpretation of the results simpler.

The study of the dynamic behavior of rolling stock and train consists can be divided into two basic groups: the study of dynamic response and the study of dynamic stability. The response study is concerned with the prediction of the dynamic behavior of the system due to external inputs. On the other hand, the stability study is aimed at investigating the stability of the system under different operating conditions.

The model analysis of a railway vehicle system consists of the solution of either the forced vibration or the dynamic stability problem. The forced vibration analysis may involve a time-domain solution or a frequency-domain solution, in which the equations of motion are numerically integrated in time or in frequency respectively.

Just as a static system may be stable or unstable, so also may be the dynamic system. The criterion for stability of a static system is that it should return to its initial configuration after a small disturbance, and the same criterion holds for a dynamic system. An unstable vibrational system amplifies any infinitesimal displacement. In such oscillatory systems, the amplitude increases spontaneously when the system is subjected to the slightest disturbance. Ordinarily, there is some maximum limit to the amplitude of these spontaneous vibrations.

Many mathematical models have been developed by the Research and Test Department of the Association of American Railroads (AAR) under the Track-Train Dynamics (TTD) program to study the dynamic behavior of rolling stock and train consists. These models can be divided into eight groups. Figure 1-1 describes the various models for rail vehicle and train dynamics developed under the TTD program.

1.1.1 The hunting phenomenon

Hunting is a very common instability exhibited by railway vehicles. The hunting phenomenon is a self excited lateral oscillation that is produced by the forward speed of the vehicle and wheel-rail interactive forces, which result from the conicity of the wheel-rail contours and friction-creep characteristics of the wheel-rail contact geometry. The interactive forces act to change effectively the damping characteristics of railway vehicle systems. Generally, there are two critical speeds associated with the hunting phenomenon. One is observed at a relatively low forward speed, and is found in vehicles with small suspension-system damping. This is primarily caused by a large lateral (including yaw and roll) oscillation of the car body and is similar to a resonance condition that can be controlled by introducing the proper amount of damping between the car body and the truck frames. At higher speeds, the hunting phenomenon appears as a violent lateral oscillation of the wheel, axle, and truck assemblies. This is an inherent condition in all rail vehicles and cannot be totally eliminated. The effectiveness of flange forces in controlling hunting oscillations is strongly dependent on axle loading, wheel-rail interactive forces, and contact geometry. Hunting in rail vehicles is undesirable since it can wear down wheel and rail profiles and cause ride discomfort.

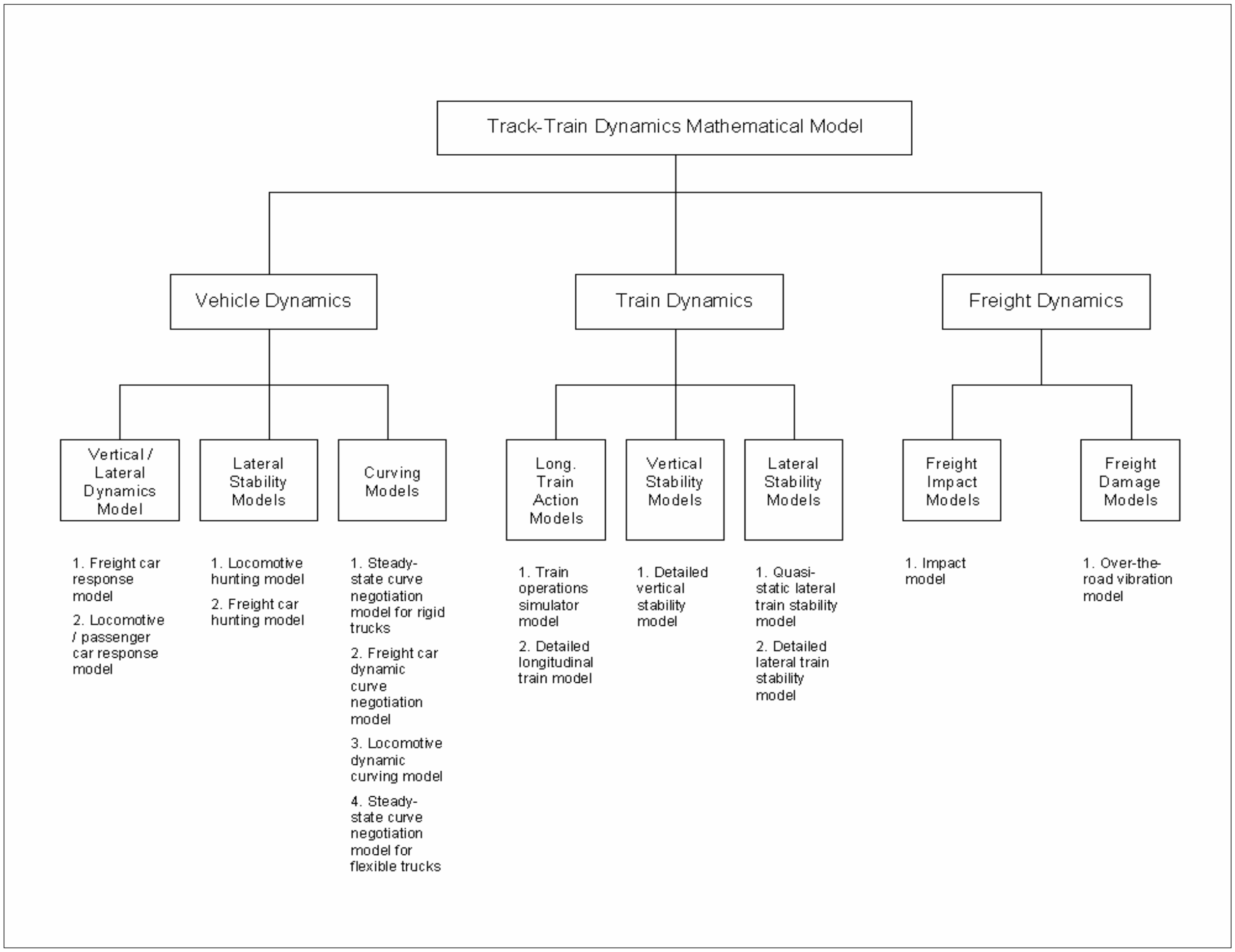

Track-Train Dynamics Mathematical Model
Vehicle Dynamics
Train Dynamics
Freight Dynamics
Vertical / Lateral Dynamics Model
Lateral Stability Models
Curving Models
Long. Train Action Models
Vertical Stability Models
Lateral Stability Models
Freight Impact Models
Freight Damage Models
1. Freight car response model
2. Locomotive / passenger car response model
1. Locomotive hunting model
2. Freight car hunting model
1. Steady-state curve negotiation model for rigid trucks
2. Freight car dynamic curve negotiation model
3. Locomotive dynamic curving model
4. Steady-state curve negotiation model for flexible trucks
1. Train operations simulator model
2. Detailed longitudinal train model
1. Detailed vertical stability model
1. Quasi-static lateral train stability model
2. Detailed lateral train stability model
1. Impact model
1. Over-the-road vibration model

Figure 1-1 Track-Train Dynamics Mathematical Models

1.1.2 Critical Velocity, Limit Cycle, and Hopf Bifurcation in Rail Vehicles

Critical velocity, in rail vehicle dynamics, can be defined as the forward speed of the vehicle beyond which the rail vehicle system exhibits an abrupt change in its dynamics. At speeds below the critical velocity, the system is stable about the origin (zero lateral displacement and zero yaw displacement). Thus, about the origin, the linearized system will have eigen values with negative real parts.

At speeds above the critical velocity, the origin loses its stability and the system trajectories move to an isolated closed trajectory called a limit cycle. Thus, in this case, the linearized system will have eigen values with positive real parts about the origin. These limit cycles, when stable or attracting, represent systems that exhibit self-sustained oscillations that can be physically detrimental to the system.

When the forward speed equals the critical velocity, the origin transitions from a stable to an unstable fixed point. At this condition, the eigen values of the linearized system will be purely imaginary. For simple dynamic systems that can be easily linearized, the critical velocity can be computed from the characteristic equation below:

$$|A - \lambda I| = 0 \tag{1.1}$$

In the above equation, A represents the linearized system matrix, λ represents the eigen values of the system, and I is the identity matrix. The number of roots λ_1, λ_2, . . ., λ_n of the characteristic equation depends on the order of the system A.

Note that $\lambda = f(V)$, where V is the forward speed of the vehicle.

Hence, $\mathrm{Re}(\lambda) = 0 \Leftrightarrow V = V_c$ (1.2)

For more complicated dynamic systems, however, system linearization can be very tedious. In such cases, the critical velocity has to be computed through simulations. The work reported in this thesis uses the MATLAB code [2] to study the dynamics of the railway vehicle system and to compute critical velocities.

A qualitative change in the dynamics of a nonlinear dynamic system is termed as a bifurcation, and the parameter values at which it occurs is called a bifurcation point. The critical velocity serves as a bifurcation point in rail vehicle dynamics, since it leads to a bifurcation in the system dynamics. Bifurcations are scientifically important since they provide models of transition and instabilities as some control parameter is varied. Several different kinds of bifurcation are seen in physical systems, but the most common type of bifurcation seen in railway vehicle lateral stability models is the Hopf bifurcation. Hopf bifurcation can be classified into supercritical and subcritical Hopf bifurcation. Supercritical Hopf bifurcation involves transition from a stable fixed point (damped trajectories) to an attracting limit cycle (self-sustained oscillations). In terms of the flow in phase space, a stable spiral changes into an unstable spiral surrounded by a small, nearly elliptical limit cycle. Typically, for speeds slightly over the critical velocity, the amplitude of the limit cycle gradually increases as the forward speed is increased beyond the critical velocity.

Before a Subcritical Hopf bifurcation, there are two attractors, a stable limit cycle and a stable fixed point at the origin. Between them lies an unstable limit cycle. At the bifurcation point, the unstable limit cycle shrinks and engulfs the origin, rendering it unstable. After the bifurcation point, the large amplitude limit cycle is the only attractor.

1.1.3 Eliminating hunting in rail vehicle operations

The hunting behavior in rail vehicles cannot be completely eliminated, since it is an intrinsic characteristic of the system dynamics. However, hunting can be avoided in normal rail vehicle operations. This can be achieved by pushing the critical velocity of hunting beyond the range of operational speeds of the vehicle. This thesis focuses on increasing the critical velocity of hunting by semi-active control of suspension elements such as springs and dampers. Semi-active control refers to a mode of control whereby the suspension properties are varied only when desired instead of in real time, as is the case with a feedback controller.

1.2 Literature Review

The first work on the stability problem faced by rail vehicles was done by DePater [8]. His work was followed by papers by Matsudaira [9] and Wickens [10]. These early works initiated a new wave of investigations around the world. Cooperrider [11] was the first to note the origins and implications of nonlinearities in railway vehicle systems. The first bifurcation analysis of the problem of the free running wheelset was performed by Huilgol [12] and revealed a Hopf bifurcation from the steady state. The first observation of chaotic oscillations in models of railway vehicles was by Kaas-Peterson [13].

In recent years, several authors have demonstrated Hopf bifurcation and chaotic oscillations in wheelsets and in railway vehicles. True et al. [14,15,16,17] have found for suspended wheelset models and for Cooperrider’s simple and complex bogie (truck) models, a stationary equilibrium point at low speeds and symmetric and asymmetric oscillations leading to chaotic motion at higher speeds. True has typically used the Vermeulen-Johnson creep force theory [7] while taking creep force saturation into account. He has, however, approximated wheel-rail normal forces by using a stiff linear spring with a deadband. Knudsen et al. [18,19] have further examined the chaotic oscillations in a suspended wheelset, and have investigated the effect of speed, suspension stiffness, and flange stiffness on the dynamics of the wheelset.

Theoretical analyses have been performed in order to predict the critical velocities in the nonlinear dynamics of wheelsets and rail vehicles, and the dependence of the critical velocity on various parameters. Law and Brand [20] have used the Krylov-Bogoliubov method to derive expressions for the amplitudes of stationary oscillations with curved wheel profiles and wheel-flange / rail contact. They have used perturbation analysis to derive conditions for the stability characteristics of the stationary oscillations. They have shown the influence of wheel profile curvature and flange clearance on the amplitudes of the stationary oscillations. Scheffel [21] has theoretically shown the dependence of hunting stability on the creep coefficient, and the influence of suspension damping on vehicle and wheelset stability.

In an effort to complement earlier studies by True et al., Ahmadian and Yang [3,22,23] have investigated the influence of nonlinear longitudinal yaw damping on the hunting behavior and the nature of Hopf bifurcation in an individual wheelset and a two-axle truck. Their study has shown that moderately increasing yaw damping can lower the critical hunting speed, and a large increase in yaw damping can improve hunting behavior.

Horak and Wormley [4] have examined the effect of track irregularities on the hunting behavior of a rail passenger truck. In addition, they have investigated the influence of vehicle suspension parameters and wheel / rail geometry on truck stability and tracking ability.

Nagurka [1], in examining curving behavior of rail vehicles, is perhaps the first author to incorporate a two-point wheel-rail contact condition in rail vehicle equations of motion.

A comprehensive review of existing literature demonstrates a lack of proper study on using controllable suspension components to enhance lateral stability of rail vehicles. While past literature has theoretically covered the influence of certain suspension parameters on the critical hunting velocity, these studies have typically used a linear model to mathematically derive the critical velocity as a function of various suspension parameters. Simulation based studies that take into account rail vehicle nonlinearities are needed to better understand the influence of suspension on the lateral stability of rail vehicles.

1.3 Objectives

The fundamental purpose of this thesis is to gain an understanding of the hunting phenomenon in rail vehicle dynamics. This is achieved by developing lateral stability models, and using analytical methods and simulations to study the behavior of these models. A principal objective of this thesis is to study the influence of rail vehicle suspension parameters on the hunting behavior.

The objectives of this thesis can be summarized as follows:

- A systematic development of a full rail vehicle model focusing on the lateral and yaw dynamics. The development includes lateral stability models of a single wheelset and of a simple truck.
- To incorporate non-linearities arising from the track-wheel interaction in each of the above models.
- To study the critical velocities, limit cycles, and Hopf bifurcation in each of the above models.
- To study the effect of variable suspension elements on the critical velocities.
- To propose a semi-active suspension control strategy in order to eliminate hunting in normal rail vehicle operation.

1.4 Outline of Thesis

Chapter 2 provides the mathematical formulation of a single wheelset model. The forces and moments that act on a single wheelset are obtained. The equations of motion that define the dynamics of the wheelset take into account both single-point and two-point wheel-rail contact conditions. Critical velocities are obtained through simulation for various primary suspension parameters. The sensitivity of the wheelset critical velocity to each suspension parameter is examined.

Chapter 3 provides the mathematical formulation for a single truck model. The truck considered is a conventional truck with a front and a rear wheelset each connected to the truck frame. The two wheelsets have independent lateral and yaw degrees of freedom. The forces and moments that act on the single truck are obtained. The equations of motion that define the dynamics of the single truck are then presented. Critical velocities are obtained through simulation for various primary and secondary suspension parameters. The sensitivity of the single truck critical velocity to each suspension parameter is then examined.

Chapter 4 provides the mathematical formulation for the complete rail vehicle model. The rail vehicle model consists of a front and a rear conventional truck, a front and a rear bolster, and the carbody. The forces and moments that act on the rail vehicle are obtained. The equations of motion that define the dynamics of the rail vehicle are then presented. Critical velocities are obtained through simulation for various primary and secondary suspension parameters. The sensitivity of the rail vehicle critical velocity to each suspension parameter is then examined.

Chapter 5 offers an overall assessment of the sensitivity of different suspension parameters on the critical velocity of the single wheelset, the single truck, and the complete rail vehicle as seen in the earlier chapters. This chapter then focuses on improving hunting behavior in rail vehicles by increasing the critical velocity of hunting beyond the operational speed range by semi-active control of suspension elements.

The appendix contains the MATLAB codes used for simulating the wheelset, truck, and rail vehicle lateral models.

Chapter 2

Single Wheelset Model

This chapter provides the mathematical formulation for a single wheelset model. The forces and moments which act on a single wheelset and which govern the lateral and yaw motions of the wheelset are obtained and the equations which define the dynamics of the wheelset are enumerated. The wheelset is suspended by springs and viscous dampers in a fixed frame that has no lateral or vertical displacements. The single wheelset is subjected to specific initial conditions. Both single-point and two-point wheel / rail contact conditions were considered. The mathematical equations governing the motion of a single wheelset [1] are simplified in order to reduce computation time without compromising on the accuracy of the solution. The simulation software MATLAB [2] was used to find time-domain solutions to the wheelset dynamic equations. The critical velocities obtained by varying primary suspension parameters are presented.

2.1 Mathematical Formulation

The wheelset represents the basic element of the rail vehicle steering and support system. Each wheelset consists of two steel wheels rigidly mounted to a solid axle. A typical wheelset cross-section is shown in Figure 2-1. Each wheel profile has a steep taper section at the inner edge known as the flange and a shallow taper (or sometimes cylindrical) section from the flange to the outer edge known as the tread. A variety of wheel profile shapes are used in the transit industry. This thesis makes use of the AAR 1 in 20 wheel, also known as the 'new wheel' that is described later in this chapter.

The simulations performed in this thesis neglect any track irregularities and assume the 'new wheel' to be rolling on a smooth, level, and tangent (straight) track. The radius of curvature of the track therefore is infinite and the track superelevation angle is zero. Hence, the centrifugal forces and the cant insufficiency are zero.

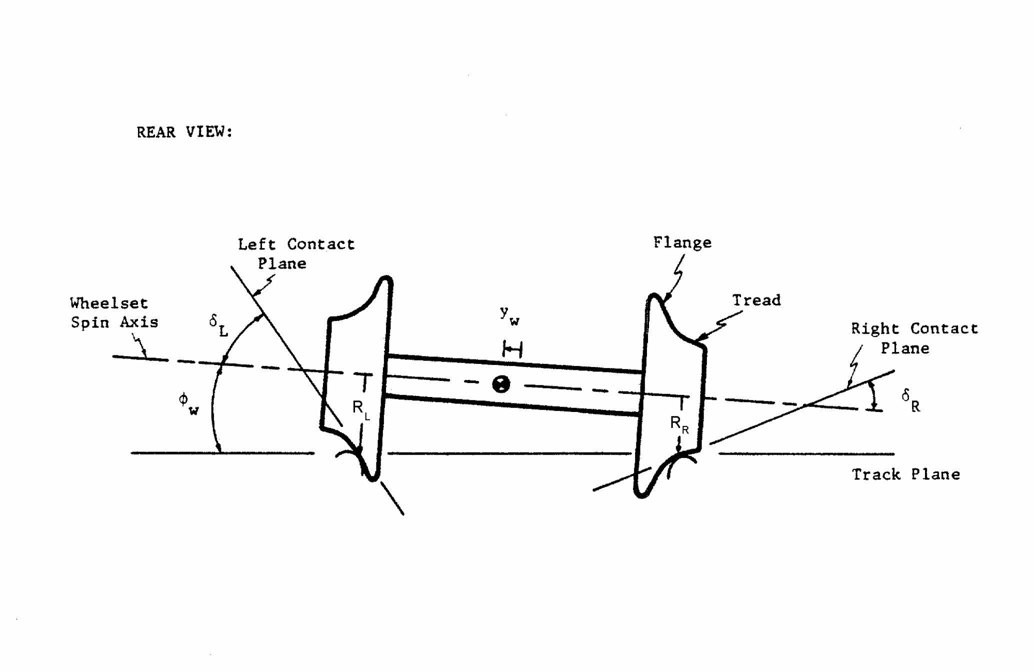


Figure 2-1 Typical Wheelset Cross-Section

A simple model of the track flexibility is adopted in which each rail is assumed to have lateral freedom only. In this model, rail rollover or overturning motion is neglected. The rail is assumed to have effective lateral mass, viscous damping, and linear stiffness, m_{RAIL}, C_{RAIL}, and K_{RAIL} respectively. The rail lateral displacement, y_{RAIL} is related to the net lateral wheel force by the rail equation of motion presented later in this chapter. The flexible rail model is shown in Figure 2-4.

The wheelset is held laterally, by two primary lateral springs with spring constant K_{PY} each in the y direction, which deform to accommodate the movements of the wheelset in the y direction. It is assumed that a primary lateral damper with damping constant C_{PY} is in parallel with each of the primary lateral springs. The wheelset has a mass m_W and mass moment of inertias I_{WX}, I_{WY}, and I_{WZ} about the x, y, and z axis respectively. Because of the symmetry of the wheelset about the x and the z axes, the moments of inertia I_{WX} and I_{WZ} will be equal and intechangeable. When the wheelset yaws (i.e. rotates about the z axis), restoring torques are produced by two primary longitudinal springs with spring constant K_{PX} each, which are placed a distance $2d_p$ apart from each other. As in the case of the primary lateral springs, primary longitudinal dampers with damping constant C_{PX} are in parallel to the springs, providing viscous damping torques proportional to the yaw rates, when the wheelset is yawing.

Imperfections in the laying of the rails will produce perturbations in the wheel's lateral position. In reality, there will be a statistical distribution of such imperfections leading to repeated but irregular forces on the wheelset. The imperfections in the laying of the rails is statistically described by a 'Roughness Parameter', which will decide the forcing function to be used in the simulation. However, in this thesis, only the effect of a single such initial disturbance on the subsequent lateral motion of the wheelset is considered. These disturbances have been modeled in the simulation as constant value inputs at time zero. The initial positional and velocity offsets will be in the plane of the track, and the forces of interaction between the track and the wheels will be both in and normal to the plane of contact between them. Since railway wheels have a taper and the rails also have a profile, the plane of contact will not in general be horizontal. Due to the heavy weight of the car, there will be crushing of the rails, leading to area contact between them and not point contact. The resulting contact patch is assumed elliptical (based on Kalker's theory) and the creep forces resulting from the deformation are also based on Kalker's theory [6]. These forces will be in the contact patch plane. The weight forces will be in the vertical plane. The forces in the rail coordinate system will need to be converted into the wheelset coordinate system by appropriate coordinate conversion.

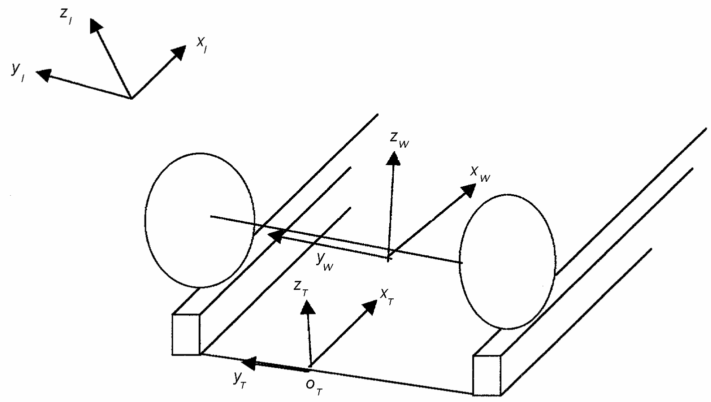
Figure 2-2 shows the assumed coordinate system. There are three systems, viz., the 'I' system, which is inertial, the 'T' system, which is track related and finally the 'W' system, which is wheelset related. Each of these systems is defined by a right-handed triad of vectors, with unit vectors $\hat{i}$, $\hat{j}$, and $\hat{k}$ along the x, y, and z directions. The three directions are defined as: x along the longitudinal direction positive in the direction of travel of the wheelset (or vehicle), y along the lateral direction, positive towards left when viewed from the rear of the car and z, which is vertical and positive upwards. The 'I' system is fixed in inertial space. The 'T' system is presumed to have its origin at the center point between the rails and it moves along the track centerline with the tangential speed V of the vehicle. Since the vehicle is not negotiating a curve and is moving in the horizontal plane x_T and y_T lie in the horizontal plane, while z_T is vertically upwards. The 'W' system has the three axes aligned with the principal directions of the wheelset, with its origin at the wheelset center of mass. The y_W axis is along the axle pointing left, the z_W axis is the yaw axis positive direction upwards and the x_W axis is perpendicular to both y_W and z_W (x_W is the cross product of y_W and z_W), i.e. $x_W = y_W \times z_W$. The wheelset orientation is defined with respect to the track system by yaw, roll and pitch angles. The wheelset orientation is obtained from the track orientation, by initially aligning the wheelset axes along the track axes, i.e. the W coordinate system coincides with the T coordinate system. When there is no roll or yaw, x_W coincides with the direction of the forward velocity and z_W is vertical upwards The present wheelset position is obtained by successively rotating the W axes by (1) yaw (ψ_W) about the z_T axis, (2) roll (ϕ_W) about the x_T axis and (3) pitch (θ_W) about the y_T axis. Since x, y, and z form a right handed triad of axes, ϕ_W is positive, when y_T is rotated into z_T, θ_W is positive, when z_T is rotated into x_T, and ψ_W is positive when x_T is rotated into y_T. The rotational transformations to obtain W coordinates from the T coordinates are also shown in Figure 2-2.

Railway wheels have a taper with a flange at the inboard end. The profile of the wheel used in this thesis is that of the AAR 1 in 20 wheel also referred to as the ‘new wheel’. The profile of the ‘new wheel’ and its contact angle are shown in Figure 2-3. The wheelset roll angle and its rate of change are given in equations (2.1) and (2.2).

$$\phi_W = \frac{(R_L - R_R)}{2a} \tag{2.1}$$

$$\dot{\phi}_W = (\lambda_L + \lambda_R)\frac{\dot{y}_W}{2a} \tag{2.2}$$

The wheel is approximated here to have a constant conicity λ equal to 0.125 up to a tread thickness of 8mm (flange clearance), followed by a sharp flange. For any lateral travel of the wheel up to the flange clearance the rise or fall of the wheel center from the horizontal will be linear, and the wheel is said to be operating in the ‘Tread Region’. The actual wheel has a sharp rise in radius at this point. In this thesis, however, the rise of the flange is assumed to extend from $y = y_{FC} = 0.008$ m (i.e. 8 mm) till $y = y_{FC} = 0.009$ m (i.e. 9 mm). The diameter change from tread to flange is assumed to take place over a lateral distance of 1.0 mm to avoid problems in digital simulation. After a lateral wheelset excursion of y = 0.009 m, the wheel is said to be operating in the ‘Flange Region’. The tread and the flange profiles [1] have been used with certain simplifying approximations to reduce computation load. The actual shape of the flange after 9 mm is really not important, since, if the wheel has started riding on the flange, then further motion of the wheel is irrelevant. Because of the taper of the wheels any lateral shift of the wheels produces two effects:



To Transform from T Coordinates to W coordinates

$$\begin{bmatrix} \hat{i}_W \\ \hat{j}_W \\ \hat{k}_W \end{bmatrix} = \begin{bmatrix} c\theta_W & 0 & -s\theta_W \\ 0 & 1 & 0 \\ s\theta_W & 0 & c\theta_W \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi_W & s\phi_W \\ 0 & -s\phi_W & c\phi_W \end{bmatrix} \begin{bmatrix} c\psi_W & s\psi_W & 0 \\ -s\psi_W & c\psi_W & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{i}_T \\ \hat{j}_T \\ \hat{k}_T \end{bmatrix}$$

When θ_W can be neglected and ψ_W, ϕ_W are small angles, this becomes

$$\begin{bmatrix} \hat{i}_W \\ \hat{j}_W \\ \hat{k}_W \end{bmatrix} = \begin{bmatrix} 1 & \psi_W & 0 \\ -\psi_W & 1 & \phi_W \\ 0 & -\phi_W & 1 \end{bmatrix} \begin{bmatrix} \hat{i}_T \\ \hat{j}_T \\ \hat{k}_T \end{bmatrix}$$

Figure 2-2 Wheelset-Track Coordinate System

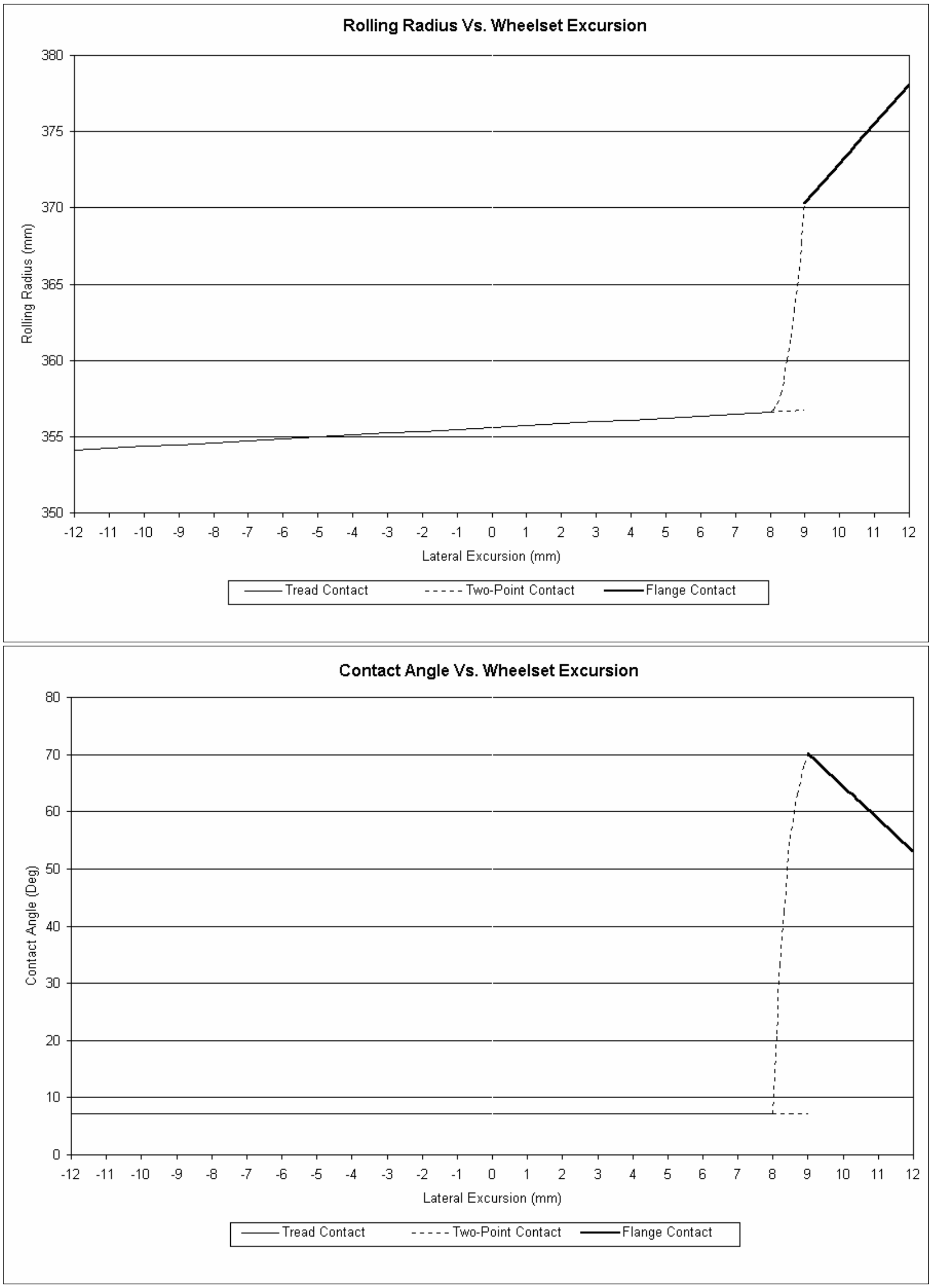


Figure 2-3 ‘New Wheel’ Assumed Rolling Radius and Contact Angle Profiles

(a) Due to the taper, one wheel of the wheelset rises, while the opposite wheel lowers. This leads to a restoring force due to gravity called Gravitational Stiffness. In the tread region, where the conicity λ is constant, this will lead to a restoring force proportional to the lateral excursion y. When one of the wheels is climbing the flange, it will produce a comparatively large restoring gravitational force. The restoring force will depend on the load on each wheel as also the angle of contact between the rail and the wheel. There will be a normal force at the contact patch, which will be perpendicular to the contact patch, which in turn will depend on the wheel / rail geometry. The resolved value of this force will depend on the angles of contact between the wheel and rail.

(b) Due to different radii of the two wheels in contact with the rails, when there is lateral excursion of the wheelset, the relative velocity of the wheel at the left and the right contact patches is different. This leads to creep forces and moments. Creep produces force components in the x and the y directions acting on the wheels in the plane of the contact patch as well as a moment normal to the contact patch trying to rotate the wheelset about the z axis. The actual forces and moments are dependent upon different creep coefficients, as well as the relative velocity of the two surfaces in contact. Here, they will depend on the value of the wheel contact patch velocity normalized with respect to the forward velocity of the vehicle. These relative velocities are known as creepages. The forces due to creep along the lateral axis y and along the longitudinal axis x are decided by the lateral creepage ξ_Y and the longitudinal creepage ξ_X respectively, while the moment about the vertical z axis is decided by the spin creepage ξ_{SP}.

2.1.1 Flexible Rail Model

In the derivation of the wheelset equations of motion, a simple spring-mass-damper model of the rail is assumed. This model is depicted in Figure 2-4. The equation of motion of the rail is:

$$-F_{RAIL} = m_{RAIL}\ddot{y}_{RAIL} + C_{RAIL}\dot{y}_{RAIL} + K_{RAIL}y_{RAIL}$$

where, $-F_{RAIL}$ is the force acting on the rail, i.e. it is the negative of the net lateral force acting on the wheel, inclusive of normal and creep forces. The values assumed in this simulation are 14.6e4 N-s/m and 14.6e7 N/m respectively for C_{RAIL} and K_{RAIL} respectively. Since these values are so high, the term $m_{RAIL}\ddot{y}_{RAIL}$ is neglected [1]. Therefore, the force on the rail is calculated as:

$$-F_{RAIL} = C_{RAIL}\dot{y}_{RAIL} + K_{RAIL}y_{RAIL} \tag{2.3}$$

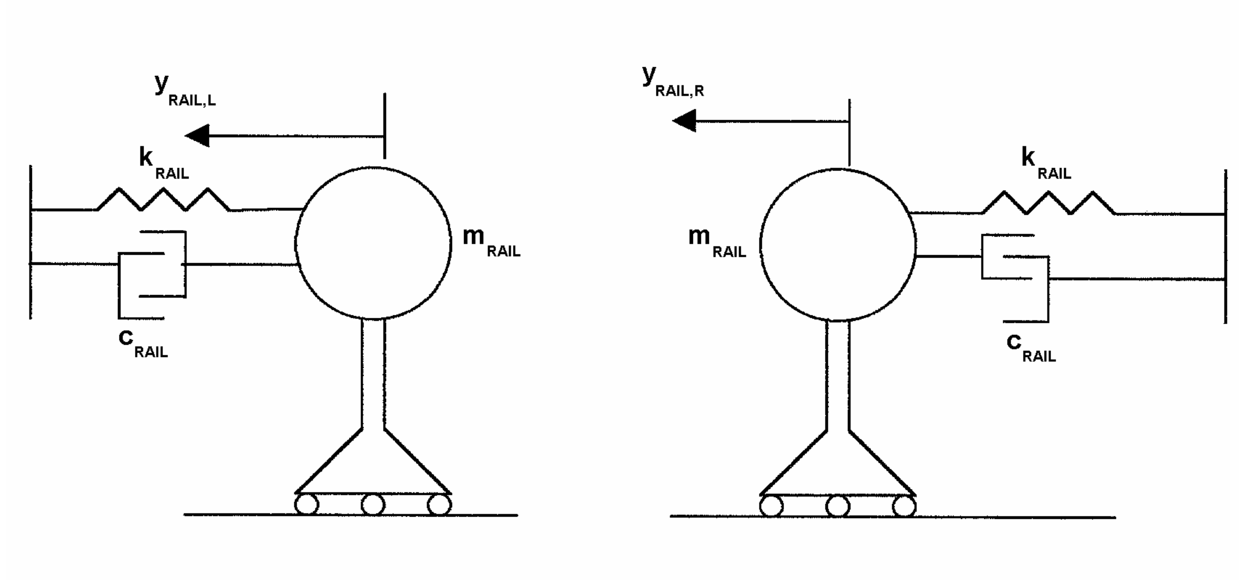


Figure 2-4 Flexible Rail Model

2.1.2 Suspension Forces and Moments

This simulation considers forces due to suspension in the y and the z directions, which are represented by the terms F_{SUSPYW} and M_{SUSPZW}. These are the forces on the wheelset due to the springs that connect the wheelset to the carbody through the truck. These springs, when in compression or elongation, will produce forces on the wheelset. In the z direction, the total weight of the carbody and trucks is distributed on the wheelsets. Since each carbody consists of two trucks and four axles, each wheelset will be subjected to the weight of a quarter of the entire carbody weight, by the compression of the vertical springs. This value is taken in the simulation for the vertical suspension load. In the lateral direction, when the wheelset is displaced with respect to the truck along the y axis, a restoring force will act on the wheelset. Since there are two springs (each having a stiffness of K_{PY}) and two viscous dampers (damping constant of C_{PY}) per each wheelset in the lateral direction, the lateral suspension force will be

$$F_{SUSPYW} = -2K_{PY}y - 2C_{PY}\dot{y} \qquad (2.4)$$

There will also be moments about the x and z directions. The moment, which opposes yaw and yaw rate due to the two longitudinal springs and dampers per each wheelset about the z axis will be

$$M_{SUSPZW} = -2d_p^2K_{PX}\psi - 2d_p^2C_{PX}\dot{\psi} \qquad (2.5)$$

This follows from the fact that if ψ is the yaw angle of the wheelset, the spring is elongated or compressed by a linear distance $d_p\psi$ and the associated spring force is $-K_{PX}d_p\psi$. Similarly, the viscous damping also produces a force $-C_{PX}d_p\dot{\psi}$.

2.1.3 Single - Point Wheel / Rail Contact

The lateral excursion required for the left or the right wheel from the centered position of the wheelset in order to make flange contact with the left or the right rail is known as the flange clearance between the wheel and the rail. When the lateral excursion of the wheelset is less than the flange clearance, both the left and the right wheels are in single-point tread contact with the rails. Alternately, when the lateral excursion of the wheelset is greater than the flange clearance, the left or the right wheel (depending on the direction of motion of the wheelset) starts climbing on the rail and thus makes a single-point flange contact with the rail. For example, when the wheelset moves to the left by a distance greater than the flange clearance, the left wheel makes a single-point flange contact with the left rail, whereas the right wheel maintains a single-point tread contact with the right rail. Figure 2-7 illustrates single-point contact for the left wheel.

The following sub-sections outline the wheel - rail interaction forces (creep and normal forces) and the wheelset dynamic equations that are applicable for a single-point contact situation on either wheel.

2.1.3.1 Single - Point Creep Forces and Moments

Assuming the forward velocity to be a constant and the vehicle to be travelling in a straight line, and taking into account the roll ($\dot{\phi}$), pitch ($\dot{\theta}$), and yaw ($\dot{\psi}$) rates, the creepages at the left and the right contact patches are given by:

$$\xi_{XL} = \frac{1}{V}[\ V - R_L\dot{\theta}_W - a\dot{\psi}_W]$$

$$\xi_{YL} = \frac{1}{V}[\dot{y}_W + R_L(\dot{\phi}_W - \dot{\theta}_W\psi_W) - \dot{y}_{RAIL,L}]/\cos(\delta_L + \phi_W)$$

$$\xi_{SPL} = \frac{1}{V}[-\dot{\theta}_W \sin(\delta_L + \phi_W) + (\dot{\psi}_W + \phi_W\dot{\theta}_W)\cos(\delta_L + \phi_W)] \tag{2.6}$$

$$\xi_{XR} = \frac{1}{V}[V - R_R\dot{\theta}_W + a\dot{\psi}_W]$$

$$\xi_{YR} = \frac{1}{V}[\dot{y}_W + R_R(\dot{\phi}_W - \dot{\theta}_W\psi_W) - \dot{y}_{RAIL,R}]/\cos(\delta_R - \phi_W)$$

$$\xi_{SPR} = \frac{1}{V}[\dot{\theta}_W \sin(\delta_R - \phi_W) + (\dot{\psi}_W + \phi_W\dot{\theta}_W)\cos(\delta_R - \phi_W)] \tag{2.7}$$

The wheel / rail geometry illustrating the parameters δ_L, δ_R, R_L, R_R, and ϕ_W are shown in Figure 2-1. The creep forces that are generated in the contact patch coordinate system are resolved in the wheel coordinate system. Since these forces act against the displacement, they will produce an oscillatory motion in the lateral plane. The actual motion also depends on the springs K_{PX} and K_{PY} and the dampers C_{PX} and C_{PY}.

There are four creep coefficients, viz. f_{11}, f_{12}, f_{22}, and f_{33}, which decide the creep forces and moment. These are calculated for a nominal value of the normal load according to Kalker's linear theory [6] and then reduced by 50% [1]. The actual values depend on the normal load F_N and are to be reduced to the actuals using the following relations:

$$f_{11} = f_{11}^*(F_N / F_N^*)^{2/3}$$

$$f_{12} = f_{12}^*(F_N / F_N^*)$$

$$f_{22} = f_{22}^*(F_N / F_N^*)^{4/3}$$

$$f_{33} = f_{33}^*(F_N / F_N^*) \tag{2.8}$$

In the above equations, coefficients f_{ij} are the values of the creep coefficients at the normal load F_N, while the coefficients f^*_{ij} are the creep coefficients for the nominal load F^*_N. The creep coefficients are functions of the wheel / rail geometry, material properties and the normal load.

Since they are functions of wheel / rail geometry, the actual values will be different for the tread and the flange contact regions.

Different papers have used different values for the coefficients. The values quoted for the 'new wheel' coefficients [1] (the values below are half Kalker values, after conversion to MKS units, for a nominal contact patch normal load of 15,000 lbs., i.e. 66,800 N) are as follows:

$$\begin{array}{ll} f_{11T} = 4.85e6\ \text{N} & f_{11F} = 3.27e6\ \text{N} \\ f_{12T} = 1.18e3\ \text{N.m} & f_{12F} = 9.35e3\ \text{N.m} \\ f_{22T} = 34.58\ \text{N.m}^2 & f_{22F} = 0.86\ \text{N.m}^2 \\ f_{33T} = 5.25e6\ \text{N} & f_{33F} = 3.0e6\ \text{N} \end{array}$$

where the subscripts T and F refer to the tread and the flange values respectively.
Horak and Wormley [4] assume for simulation values of $f_{11T} = 9.43e6\,\text{N}$ and $f_{33T} = 10.23e6\,\text{N}$ while neglecting the other coefficients. The values assumed by Ahmadian and Yang [3] are:
$f_{11T} = 6.728e6\,\text{N}$, $f_{12T} = 1.2e3\,\text{N.m}$, $f_{22T} = 1.0e3\,\text{N.m}^2$ $f_{33T} = 6.728e6\,\text{N}$.

It was found that using the different values quoted in these papers gave only slightly different values for the critical velocities. In other words the critical velocity is only a weak function of the creep coefficients. In this thesis, some values are taken from Reference 4, while some others are from Reference 3. Where the values are available only in Reference 1, they have been used. It was found that using all the values from Reference 1 gave only slightly different values for the critical speeds from what was obtained by using the present values. The same values of the creep coefficients were assumed for the tread and the flange. Since the flange comes into contact with the rail only for a lateral position of the wheelset from 8 mm to 9mm (i.e. a lateral excursion of 1mm only), assumption of different creep coefficient values for the flange do not make any significant difference to the behavior of the wheelset. The forces along the positive directions of the x and the y axes (in the contact patch plane) and the moment about the z axis are given by:

$$F'_{CPX} = -f_{33}\xi_X$$

$$F'_{CPY} = -f_{11}\xi_y - f_{12}\xi_{SP}$$

$$M'_{CPZ} = f_{12}\xi_y - f_{22}\xi_{SP} \tag{2.9}$$

Of these forces, the force in the y direction opposes the velocity of the motion and being similar to a frictional force, can be beneficial in damping out oscillations. But the force in the x direction produces a torque, which will set up yaw motion and thus produces oscillatory motion, causing the wheelset to hunt between the rails. The direction of this torque is such that when the wheelset is moving towards the left, the yaw tends to turn the wheelset so as to cause the wheelset move towards the right rail and vice versa, hence causing a hunting motion. This occurs as follows: When the wheelset has moved to the left, because of the taper λ, the radius of the left wheel above the point of contact R_L is larger than the centered value, while the radius at the right rail R_R is less than the centered value. Since for pure roll the point of contact of the left wheel will be moving back at a speed $R_L\dot{\theta}_W$, it follows that the point of contact of the wheel is moving back with a relative velocity of $(R_L - R_0)\dot{\theta}_W$, resulting in a normalized velocity (in the forward x direction) of $-(R_L - R_0)\dot{\theta}_W / V$. Similarly the normalized velocity at the right wheel point of contact is $-(R_R - R_0)\dot{\theta}_W / V$. Thus, qualitatively it is seen that the force on the right wheel will be forward, while that on the left wheel will be backward, producing a couple which will be in a direction to make the wheelset turn towards the right rail. The above description neglects the effects of angular rates and longitudinal accelerations.

It is found that for forward speeds below the critical speed, the disturbance caused by any initial perturbation dies out, while for forward speeds over the critical speed, the oscillation grows into a limit cycle, where the flanges start hitting against the rails. The higher the forward speed over the critical speed, the more the wheel climbs the flange and hence, the closer the wheel gets to derailment. Beyond the critical speed, oscillatory motion is observed even without the flange. The amplitude of the limit cycle varies with the value of the speed. The critical velocity is seen to vary inversely with the wheel conicity λ.

But there is a limiting condition to the value of these forces. The resultant creep force cannot exceed that available due to adhesion. If F_N is the normal force at the rail contact patch, then the resultant of the creep forces $F^{'}_{CPX}$ and $F^{'}_{CPY}$ cannot exceed that due to available adhesion at the wheel / rail contact patch, i.e.

$$\sqrt{F'^{\,2}_{CPX} + F'^{\,2}_{CPY}} \leq \mu F_N$$

In simulation, this condition is achieved by using a modified Vermeulen-Johnson model [1,4,7]. In this method, which includes the effect of spin creepage, a saturation coefficient, ε is calculated using the following relation:

$$\varepsilon = (1/\beta)\times[\beta - \beta^2/3 + \beta^3/27] \qquad \text{For } \beta < 3$$
$$\varepsilon = 1/\beta \qquad \text{For } \beta \geq 3 \qquad (2.10)$$

where β is the normalized unlimited creep force, given by:

$$\beta = (\frac{1}{\mu F_N}) \times \sqrt{F'^{\,2}_{CPX} + F'^{\,2}_{CPY}}$$

The saturated contact patch creep forces and moments are then given by:

$$F_{CPX} = \varepsilon \times F'_{CPX}$$
$$F_{CPY} = \varepsilon \times F'_{CPY}$$
$$M_{CPZ} = \varepsilon \times M'_{CPZ} \qquad (2.11)$$

These forces are calculated for the left and the right contact patches separately as F_{CPXL}, F_{CPYL}, M_{CPZL} and F_{CPXR}, F_{CPYR}, and M_{CPZR} respectively and are resolved in the track plane as follows:

Left Contact Patch:

$$F_{CXL} = F_{CPXL}$$
$$F_{CYL} = F_{CPYL}\cos(\delta_L + \phi_W)$$
$$F_{CZL} = F_{CPYL}\sin(\delta_L + \phi_W)$$
$$M_{CXL} = 0$$

$$M_{CYL} = -M_{CPZL}\sin(\delta_L + \phi_W)$$
$$M_{CZL} = M_{CPZL}\cos(\delta_L + \phi_W) \tag{2.12}$$

<u>Right Contact Patch</u>:

$$F_{CXR} = F_{CPXR}$$
$$F_{CYR} = F_{CPYR}\cos(\delta_R - \phi_W)$$
$$F_{CZR} = -F_{CPYR}\sin(\delta_R - \phi_W)$$
$$M_{CXR} = 0$$
$$M_{CYR} = M_{CPZR}\sin(\delta_R - \phi_W)$$
$$M_{CZR} = M_{CPZR}\cos(\delta_R - \phi_W) \tag{2.13}$$

Since the creep force along the x direction is what causes yaw oscillations and consequent flanging of the wheelset, limiting of the contact patch creep forces is beneficial in increasing the critical velocity. The contact patch creep force corresponds to a situation in between pure slip and pure roll. This is illustrated in Figure 2-5.

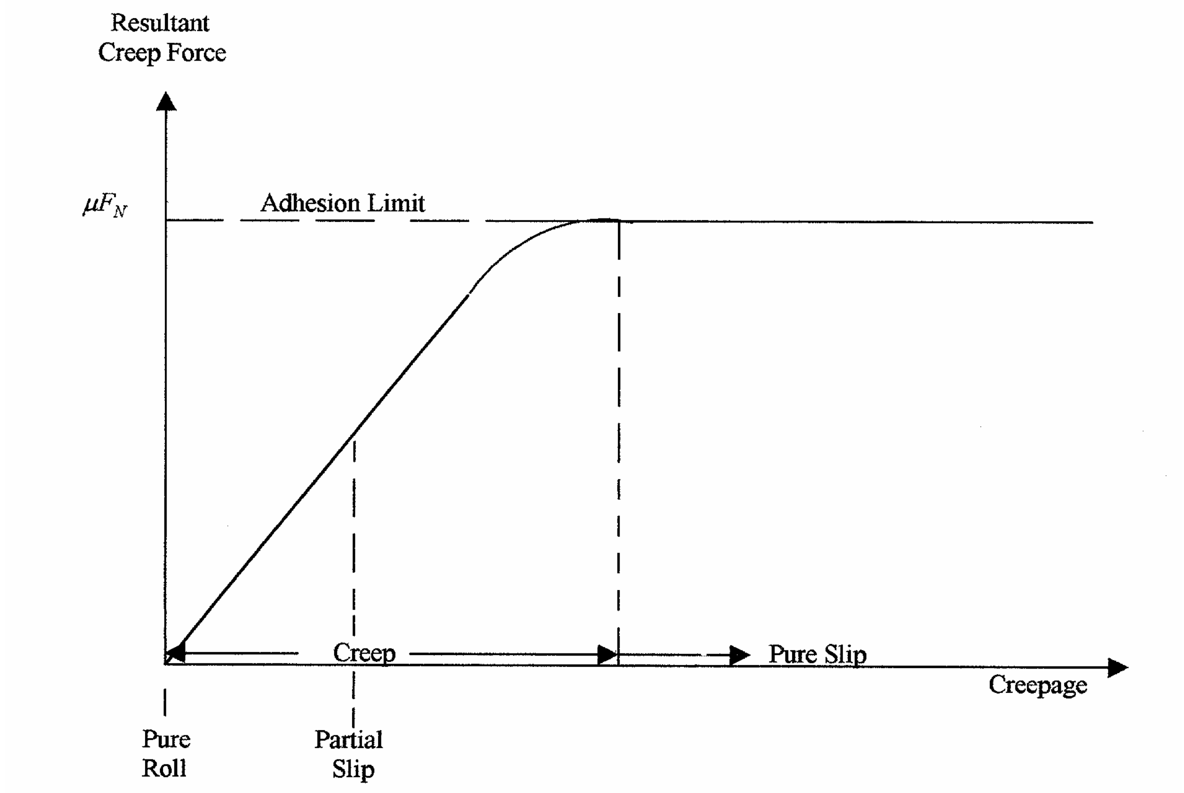


Figure 2-5 Contact Patch Creep Force

The forces acting on the wheel depend on the point or points where the wheel comes in contact with the rail. The flange clearance is assumed to be 8 mm (0.32 in). When the wheelset excursion is less than the flange clearance, tread contact occurs. When the wheelset excursion equals or exceeds the flange clearance, flanging occurs. Single-point contact situation is illustrated in Figure 2-7.

2.1.3.2 Single - Point Normal Forces and Moments

The normal forces at the two rails are required to be calculated at each time step, since the exact value of the normal force will depend on the angle of contact as well as the roll angle. It is convenient to calculate the normal forces at the contact patches from the wheelset vertical and roll equations by simultaneously solving the equations. The normal forces at the left and the right contact patches work out as:

$$F_{NL} = \nu_L / \Delta \quad \text{and} \quad F_{NR} = \nu_R / \Delta, \tag{2.14}$$

where

$$\nu_L = F_Z^* \{ a \cos \ (\delta_R - \phi_W) - R_R \sin (\delta_R - \phi_W) \} + M_\phi^* \cos (\delta_R - \phi_W)$$
$$\nu_R = F_Z^* \{ a \cos(\delta_L + \phi_W) - R_L \sin(\delta_L + \phi_W) \} - M_\phi^* \cos(\delta_L + \phi_W)$$
$$\Delta = 2a \cos(\delta_L + \phi_W) \cos (\delta_R - \phi_W) - R_R \cos(\delta_L + \phi_W) \sin (\delta_R - \phi_W) - R_L \sin(\delta_L + \phi_W) \cos (\delta_R - \phi_W) \tag{2.15}$$

In the above expressions, F_Z^* and M_ϕ^* are equivalent vertical force and equivalent roll moment given by:

$$F_Z^* = -F_{CZL} - F_{CZR} - F_{SUSPZW} + m_W g$$
$$M_\phi^* = a(F_{CZR} - F_{CZL}) - R_L (F_{CYL} - \psi_W F_{CXL}) - R_R (F_{CYR} - \psi_W F_{CXR}) - \psi_W (M_{CYL} + M_{CYR}) - I_{WY} \dot{\theta}_W \dot{\psi}_W \tag{2.16}$$

The normal forces on the left and the right wheels, F_{NL} and F_{NR}, act perpendicular to the contact patch plane and can be resolved into lateral and vertical components in the track plane. For single point wheel / rail contact, the resolved normal force components are:

$$F_{NYL} = -F_{NL} \sin(\delta_L + \phi_W)$$
$$F_{NZL} = F_{NL} \cos(\delta_L + \phi_W)$$
$$F_{NYR} = F_{NR} \sin(\delta_R - \phi_W)$$
$$F_{NZR} = F_{NR} \cos(\delta_R - \phi_W) \tag{2.17}$$

2.1.3.3 Single - Point Wheelset Dynamic Equations

The dynamic equations of a single wheelset are obtained from Newton's laws applied to the wheelset, both for force and moments. Thus the sum of all forces acting on the wheelset in the lateral and vertical directions will equal the product of the mass and the lateral and vertical accelerations respectively. Similarly, the sum of moments acting about any axis will equal the product of the mass moment of inertia and the angular acceleration. In this case, gyroscopic torques (when angular rates about two mutually perpendicular axes are present) are to be taken into account also.

The free body diagram of the wheelset in single point contact is given in Figure 2-6. The equations below include the simplifications that since ψ_W and ϕ_W are small angles, $\cos(\psi_W)$, $\cos(\phi_W)$, $\sin(\psi_W)$, and $\sin(\phi_W)$ are replaced by 1,1, ψ_W, and ϕ_W respectively.

Lateral Equation

$$F_{CYL} + F_{CYR} + F_{NYL} + F_{NYR} + F_{SUSPYW} - m_W g\phi_W = m_W \ddot{y}_W \tag{2.18}$$

Yaw Equation

$$\begin{aligned} I_{WZ}\ddot{\psi}_W = &- I_{WY}\dot{\theta}_W\dot{\phi}_W - a(F_{CXL} - F_{CXR}) - \psi_W\{(a - R_L \tan(\delta_L + \phi_W))(F_{CYL} + F_{NYL}) \\ &- (a - R_R \tan(\delta_R - \phi_W))(F_{CYR} + F_{NYR})\} + M_{CZL} + M_{CZR} + M_{SUSPZW} - \phi_W(M_{CYL} + M_{CYR}) \end{aligned} \tag{2.19}$$

In equation (2.19) above, the term $-I_{WY}\dot{\theta}_W\dot{\phi}_W$ is a gyroscopic torque term. As mentioned earlier, the term related to the acceleration of the rail in the rail equations below is neglected, leading to the following equations

Left Rail Equation

$$C_{RAIL}\dot{y}_{RAIL,L} + K_{RAIL}y_{RAIL,L} = -F_{NYL} - F_{CYL} \tag{2.20}$$

Right Rail Equation

$$C_{RAIL}\dot{y}_{RAIL,R} + K_{RAIL}y_{RAIL,R} = -F_{NYR} - F_{CYR} \tag{2.21}$$

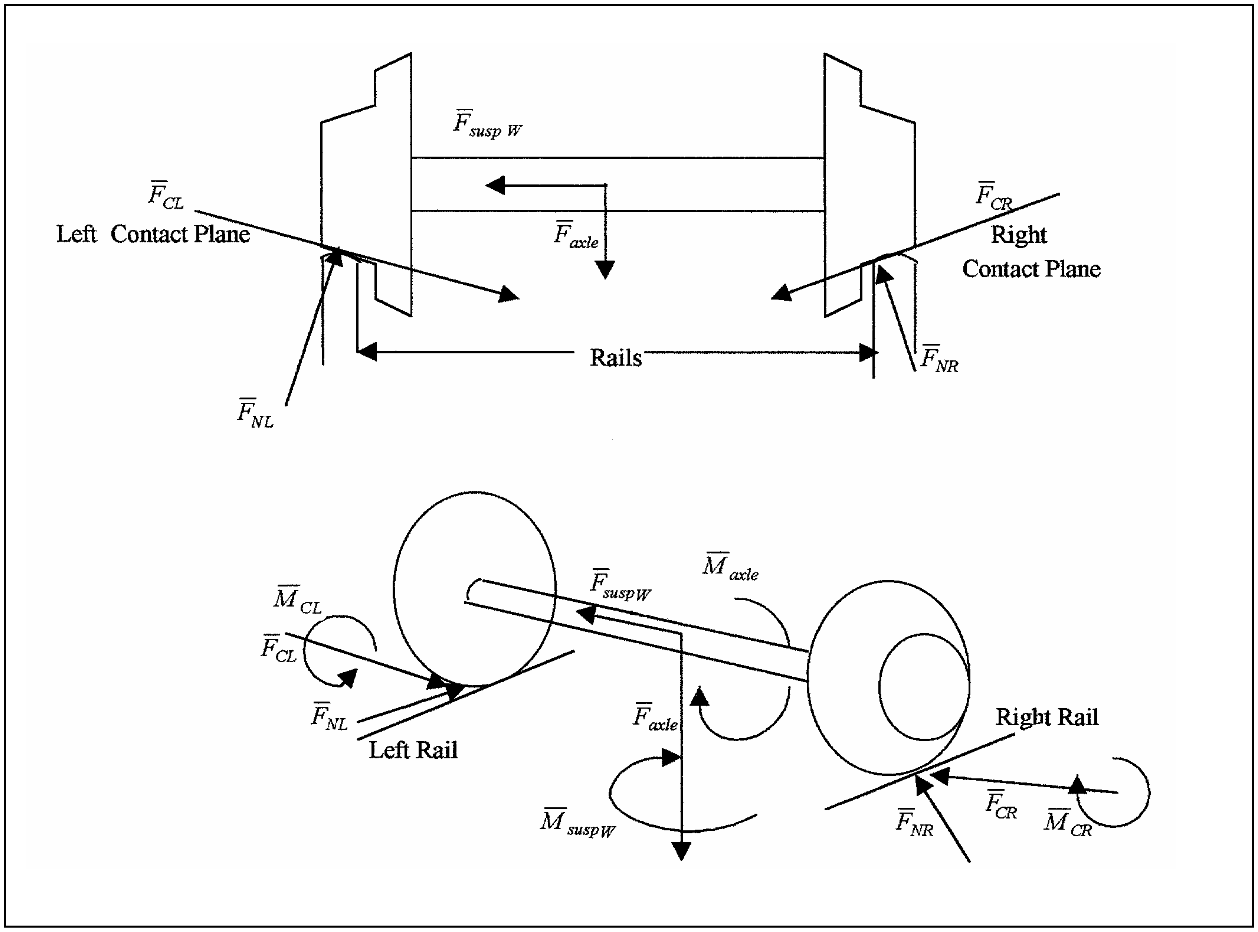


Figure 2-6 Free-Body Diagram of Wheelset in Single-Point Contact

2.1.4 Two - Point Wheel / Rail Contact

For the AAR 1 in 20 wheel (new wheel), which has an abrupt flange, unlike the Heumann wheel used widely in Europe, when the lateral wheelset excursion becomes equal to the flange clearance, both the tread and the flange of the wheel make contact with the rail. Hence, a two-point contact condition involves three different contact patches (two at the flanging wheel and one at the other wheel). The equations pertaining to the left and the right wheels, when the two-point contact condition occurs at the left wheel / rail interface are listed below. These equations can be similarly written for a two-point contact condition at the right wheel. The conditions under which single-point and two-point contact occur are shown in Figure 2-7. The condition for the left rail to have two-point contact is then:

$$y_W - y_{RAIL,L} = y_{FC} \tag{2.22}$$

When the flange is touching the left rail, the velocity and acceleration of the wheel and rail will be the same, i.e.

$$\dot{y}_W = \dot{y}_{RAIL,L} \qquad \ddot{y}_W = \ddot{y}_{RAIL,L} \tag{2.23}$$

However, as mentioned earlier, the product of the mass and acceleration term for the rail is neglected in simulation, since the spring force and viscous friction terms are dominant compared to the mass term for the rails.

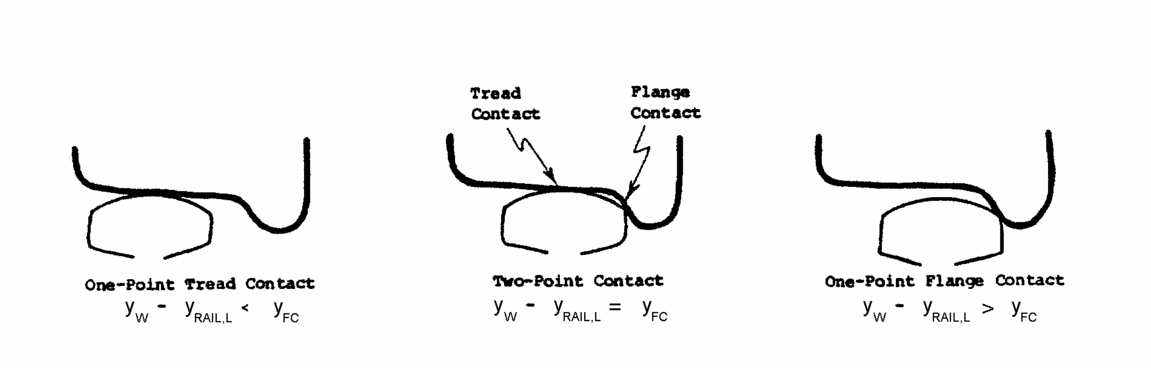


Figure 2-7 Single-Point and Two-Point Left Wheel / Rail Contact Situations

The right wheel, however, will still be making tread contact and the single-point equations outlined previously will be valid for the right wheel. In the present thesis, it is assumed that for the wheel making two-point contact, both the tread and the flange make contact from a lateral excursion of 8 mm onwards to a lateral excursion of 9 mm. Hence, for ease of simulation, a two-point contact window of 1 mm is assumed. The angles of contact δ_T and δ_F as well as the rolling radii R_T and R_F for the tread and the flange points of contact will be different. As already mentioned, the flange angle of contact is assumed to rise up to 70 degrees, when the lateral excursion is 9 mm. The radii for the tread and the flange contact points are taken as per the 'new wheel' dimensions [1]. When the wheel has completely climbed the flange, a single-point contact condition will occur again on both the wheels: tread contact for the right wheel and flange contact for the left wheel. When the wheelset excursion is equal to the flange clearance, the forces on the flanging wheel will include forces due to both tread contact as well as flange contact.

2.1.4.1 Two - Point Creep Forces and Moments

The two-point creep forces and moments presented below assumes two-point contact to be occurring at the left wheel / rail interface. These equations can be easily written for a two-point contact condition at the right wheel / rail interface. Assuming the forward velocity to be a constant and the vehicle to be travelling in a straight line, and taking into account the roll ($\dot{\phi}$), pitch ($\dot{\theta}$), and yaw ($\dot{\psi}$) rates, the creepages at the left contact (tread and flange) and the right contact (tread) patches are given by:

$$\xi_{XLT} = \frac{1}{V}[\ V - R_{LT}\dot{\theta}_W - a\dot{\psi}_W]$$

$$\xi_{YLT} = \frac{1}{V}[\dot{y}_W + R_{LT}(\dot{\phi}_W - \dot{\theta}_W\psi_W) - \dot{y}_{RAIL,L}]/\cos(\delta_{LT} + \phi_W)$$

$$\xi_{SPLT} = \frac{1}{V}[-\dot{\theta}_W \sin(\delta_{LT} + \phi_W) + (\dot{\psi}_W + \phi_W\dot{\theta}_W)\cos(\delta_{LT} + \phi_W)] \tag{2.24}$$

$$\xi_{XLF} = \frac{1}{V}[\ V - R_{LF}\dot{\theta}_W - a\dot{\psi}_W]$$

$$\xi_{YLF} = \frac{1}{V}[\dot{y}_W + R_{LF}(\dot{\phi}_W - \dot{\theta}_W\psi_W) - \dot{y}_{RAIL,L}]/\cos(\delta_{LF} + \phi_W)$$

$$\xi_{SPLF} = \frac{1}{V}[-\dot{\theta}_W \sin(\delta_{LF} + \phi_W) + (\dot{\psi}_W + \phi_W\dot{\theta}_W)\cos(\delta_{LF} + \phi_W)] \tag{2.25}$$

$$\xi_{XR} = \frac{1}{V}[V - R_R\dot{\theta}_W + a\dot{\psi}_W]$$

$$\xi_{YR} = \frac{1}{V}[\dot{y}_W + R_R(\dot{\phi}_W - \dot{\theta}_W\psi_W) - \dot{y}_{RAIL,R}]/\cos(\delta_R - \phi_W)$$

$$\xi_{SPR} = \frac{1}{V}[\dot{\theta}_W \sin(\delta_R - \phi_W) + (\dot{\psi}_W + \phi_W\dot{\theta}_W)\cos(\delta_R - \phi_W)] \tag{2.26}$$

The wheel / rail geometry illustrating the parameters δ_L, δ_R, R_L, R_R, and ϕ_W are shown in Figure 2-1. The creep forces that are generated in the contact patch coordinate system are resolved in the wheel coordinate system. Since these forces act against the displacement, they will produce an oscillatory motion in the lateral plane. The actual motion also depends on the springs K_{PX} and K_{PY} and the dampers C_{PX} and C_{PY}.

There are four Creep Coefficients, viz. f_{11}, f_{12}, f_{22}, and f_{33}, which decide the creep forces and moment. These are calculated for a nominal value of the normal load according to Kalker's linear theory [6] and then reduced by 50% [1]. The actual values depend on the normal load F_N and are to be reduced to the actuals using the following relations:

$$f_{11} = f_{11}^*(F_N / F_N^*)^{2/3}$$

$$f_{12} = f_{12}^*(F_N / F_N^*)$$

$$f_{22} = f_{22}^*(F_N / F_N^*)^{4/3}$$

$$f_{33} = f_{33}^*(F_N / F_N^*) \tag{2.27}$$

In the above equations, coefficients f_{ij} are the values of the creep coefficients at the normal load F_N, while the coefficients f^*_{ij} are the creep coefficients for the nominal load F^*_N. The creep coefficients are functions of the wheel / rail geometry, material properties and the normal load. Since they are functions of wheel / rail geometry, the actual values will be different for the tread and the flange contact regions.

Different papers have used different values for the coefficients. The values quoted for the 'new wheel' coefficients [1] (the values below are half Kalker values, after conversion to MKS units, for a nominal contact patch normal load of 15,000 lbs., i.e. 66,800 N) are as follows:

$f_{11T} = 4.85e6$ N	$f_{11F} = 3.27e6$ N
$f_{12T} = 1.18e3$ N.m	$f_{12F} = 9.35e3$ N.m
$f_{22T} = 34.58$ N.m^2	$f_{22F} = 0.86$ N.m^2
$f_{33T} = 5.25e6$ N	$f_{33F} = 3.0e6$ N

where the subscripts T and F refer to the tread and the flange values respectively.

Horak and Wormley [4] assume for simulation values of $f_{11T} = 9.43e6\,N$ and $f_{33T} = 10.23e6\,N$ while neglecting the other coefficients. The values assumed by Ahmadian and Yang [3] are:
$f_{11T} = 6.728e6\,N$, $f_{12T} = 1.2e3\,N.m$, $f_{22T} = 1.0e3\,N.m^2$ $f_{33T} = 6.728e6\,N$.

It was found that using the different values quoted in these papers gave only slightly different values for the critical velocities. In other words the critical velocity is only a weak function of the creep coefficients. In this thesis, some values are taken from Reference 4, while some others are from Reference 3. Where the values are available only in Reference 1, they have been used. It was found that using all the values from Reference 1 gave only slightly different values for the critical speeds from what was obtained by using the present values. The same values of the creep coefficients were assumed for the tread and the flange. Since the flange comes into contact with the rail only for a lateral position of the wheelset from 8 mm to 9 mm (i.e. a lateral excursion of 1 mm only), assumption of different creep coefficient values for the flange do not make any significant difference to the behavior of the wheelset. The forces along the positive directions of the x and the y axes (in the contact patch plane) and the moment about the z axis are given by:

$$F'_{CPX} = -f_{33}\xi_X$$

$$F'_{CPY} = -f_{11}\xi_y - f_{12}\xi_{SP}$$

$$M'_{CPZ} = f_{12}\xi_y - f_{22}\xi_{SP} \tag{2.28}$$

Of these forces, the force in the y direction opposes the velocity of the motion and being similar to a frictional force, can be beneficial in damping out oscillations. But the force in the x direction produces a torque, which will set up yaw motion and thus produces oscillatory motion, causing the wheelset to hunt between the rails. The direction of this torque is such that when the wheelset is moving towards the left, the yaw tends to turn the wheelset so as to cause the wheelset to move towards the right rail and vice versa, hence causing a hunting motion.

It is found that for forward speeds below the critical speed, the disturbance caused by any initial perturbation dies out, while for forward speeds over the critical speed, the oscillation grows into a limit cycle, where the flanges start hitting against the rails. The higher the forward speed over the critical speed, the more the wheel climbs the flange and hence, the closer the wheel gets to derailment. Beyond the critical speed, oscillatory motion is observed even without the flange. The amplitude of the limit cycle varies with the value of the speed. The critical velocity is seen to vary inversely with the wheel conicity λ.

But there is a limiting condition to the value of these forces. The resultant creep force cannot exceed that available due to adhesion. If F_N is the normal force at the rail contact patch, then the resultant of the creep forces F'_{CPX} and F'_{CPY} cannot exceed that due to available adhesion at the wheel / rail contact patch, i.e.

$$\sqrt{{F'_{CPX}}^2 + {F'_{CPY}}^2} \leq \mu F_N$$

In simulation, this condition is achieved by using a modified Vermeulen-Johnson model [1,4,7]. In this method, which includes the effect of spin creepage, a saturation coefficient, ε is calculated at each of the three contact patches using the following relation:

$$\begin{aligned} &\varepsilon = (1/\beta)\times[\beta - \beta^2/3 + \beta^3/27] && \text{For } \beta < 3 \\ &\varepsilon = 1/\beta && \text{For } \beta \geq 3 \end{aligned} \qquad (2.29)$$

where β is the normalized unlimited creep force, given by:

$$\beta = (\frac{1}{\mu F_N})\times\sqrt{{F'_{CPX}}^2 + {F'_{CPY}}^2}$$

The saturated contact patch creep forces and moments are then given by:

$$\begin{aligned} F_{CPX} &= \varepsilon\times F'_{CPX} \\ F_{CPY} &= \varepsilon\times F'_{CPY} \\ M_{CPZ} &= \varepsilon\times M'_{CPZ} \end{aligned} \qquad (2.30)$$

These forces are calculated for the left (tread and flange) and the right contact patches separately as F_{CPXLT}, F_{CPYLT}, M_{CPZLT}, F_{CPXLF}, F_{CPYLF}, M_{CPZLF}, and F_{CPXR}, F_{CPYR}, and M_{CPZR} respectively and are resolved in the track plane as follows:

Left Tread Contact Patch:

$$\begin{aligned}
F_{CXLT} &= F_{CPXLT} \\
F_{CYLT} &= F_{CPYLT}\cos(\delta_{LT}+\phi_W) \\
F_{CZLT} &= F_{CPYLT}\sin(\delta_{LT}+\phi_W) \\
M_{CXLT} &= 0 \\
M_{CYLT} &= -M_{CPZLT}\sin(\delta_{LT}+\phi_W) \\
M_{CZLT} &= M_{CPZLT}\cos(\delta_{LT}+\phi_W)
\end{aligned} \tag{2.31}$$

Left Flange Contact Patch:

$$\begin{aligned}
F_{CXLF} &= F_{CPXLF} \\
F_{CYLF} &= F_{CPYLF}\cos(\delta_{LF}+\phi_W) \\
F_{CZLF} &= F_{CPYLF}\sin(\delta_{LF}+\phi_W) \\
M_{CXLF} &= 0 \\
M_{CYLF} &= -M_{CPZLF}\sin(\delta_{LF}+\phi_W) \\
M_{CZLF} &= M_{CPZLF}\cos(\delta_{LF}+\phi_W)
\end{aligned} \tag{2.32}$$

Right Contact Patch:

$$\begin{aligned}
F_{CXR} &= F_{CPXR} \\
F_{CYR} &= F_{CPYR}\cos(\delta_R-\phi_W) \\
F_{CZR} &= -F_{CPYR}\sin(\delta_R-\phi_W) \\
M_{CXR} &= 0 \\
M_{CYR} &= M_{CPZR}\sin(\delta_R-\phi_W) \\
M_{CZR} &= M_{CPZR}\cos(\delta_R-\phi_W)
\end{aligned} \tag{2.33}$$

Since the creep force along the x direction is what causes yaw oscillations and consequent flanging of the wheelset, limiting of the contact patch creep forces is beneficial in increasing the critical velocity. The contact patch creep force corresponds to a situation in between pure slip and pure roll. This is illustrated in Figure 2-5.

The forces acting on the wheel depend on the point or points where the wheel comes in contact with the rail. The flange clearance is assumed to be 8 mm (0.32 in). When the wheelset excursion is less than the flange clearance, tread contact occurs. When the wheelset excursion equals or exceeds the flange clearance, flanging occurs. As shown in Figure 2-7, there is a small region, where two-point contact occurs, viz. both the tread and the flange are in contact with the rail. While the flange rises very sharply in the actual wheel, the profile used in the simulation assumes a lateral wheel travel of 1mm (between 8mm and 9mm total wheelset excursion) during which two-point contact occurs. The contact angle becomes equal to 70 degrees for a total wheelset excursion of 9mm. Beyond this point the contact angle is assumed to be constant, and again a single point of contact at the flange is assumed, though since the wheel has already climbed the flange, derailing can be deemed to have occurred at this point, notwithstanding the subsequent actual behavior of the wheel. It is not possible for two- point contact to occur at both the left and right rails at the same time. Single-point contact and two-point contact situations are illustrated in Figure 2-7.

2.1.4.2 Two – Point Contact Normal Forces and Moments

Assuming two-point contact condition at the left wheel / rail interface, the normal forces at the left tread, left flange and right tread, F_{NLT}, F_{NLF}, and F_{NR} are obtained by solving simultaneously, the wheelset vertical and roll equations as well as the left rail lateral equations under the above conditions.

$$F_{NLT} = \nu_{LT} / \Delta 2 \qquad F_{NLF} = \nu_{LF} / \Delta 2 \qquad F_{NR} = \nu_{2R} / \Delta 2 \tag{2.34}$$

where,

$$\begin{aligned}\nu_{LT} &= F''_Y\{2a\cos(\delta_{LF}+\phi_W)\cos(\delta_R-\phi_W)-R_{LF}\sin(\delta_{LF}+\phi_W)\cos(\delta_R-\phi_W)\\ &-R_R\cos(\delta_{LF}+\phi_W)\sin(\delta_R-\phi_W)\}+F''_Z\{\sin(\delta_{LF}+\phi_W)[a\cos(\delta_R-\phi_W)-R_R\sin(\delta_R-\phi_W)]\}\\ &+M''_\phi\sin(\delta_{LF}+\phi_W)\cos(\delta_R-\phi_W)\end{aligned}$$

$$\begin{aligned}\nu_{LF} &= F''_Y\{-2a\cos(\delta_{LT}+\phi_W)\cos(\delta_R-\phi_W)+R_{LT}\sin(\delta_{LT}+\phi_W)\cos(\delta_R-\phi_W)+R_R\cos(\delta_{LT}+\phi_W)\sin(\delta_R-\phi_W)\}\\ &-F''_Z\{\sin(\delta_{LT}+\phi_W)[a\cos(\delta_R-\phi_W)-R_R\sin(\delta_R-\phi_W)]\}-M''_\phi\sin(\delta_{LT}+\phi_W)\cos(\delta_R-\phi_W)\end{aligned}$$

$$
\begin{aligned}
\nu_{2R} = {} & F''_{Y}\{R_{LF}\cos(\delta_{LT}+\phi_{W})\sin(\delta_{LF}+\phi_{W}) - R_{LT}\sin(\delta_{LT}+\phi_{W})\cos(\delta_{LF}+\phi_{W})\} \\
& - M''_{\phi}\{\cos(\delta_{LT}+\phi_{W})\sin(\delta_{LF}+\phi_{W}) - \sin(\delta_{LT}+\phi_{W})\cos(\delta_{LF}+\phi_{W})\} \\
& + F''_{Z}\{a[\cos(\delta_{LT}+\phi_{W})\sin(\delta_{LF}+\phi_{W}) - \sin(\delta_{LT}+\phi_{W})\cos(\delta_{LF}+\phi_{W})] \\
& + (R_{LF} - R_{LT})\sin(\delta_{LT}+\phi_{W})\sin(\delta_{LF}+\phi_{W})
\end{aligned}
$$

$$
\begin{aligned}
& \Delta 2 = [2a\cos(\delta_{R}-\phi_{W}) - R_{R}\sin(\delta_{R}-\phi_{W})]\{\cos(\delta_{LT}+\phi_{W})\sin(\delta_{LF}+\phi_{W}) \\
& -\sin(\delta_{LT}+\phi_{W})\cos(\delta_{LF}+\phi_{W})\} + (R_{LF}-R_{LT})\sin(\delta_{LT}+\phi_{W})\sin(\delta_{LF}+\phi_{W})\cos(\delta_{R}-\phi_{W})
\end{aligned}
$$

(2.35)

Also, F_Y" and F_Z" are the equivalent lateral forces and M_ϕ" is an equivalent roll moment given by the following expressions:

$$
\begin{aligned}
& F_y^{**} = -F_{CYLT} - F_{CYLF} - C_{RAIL}\dot{y}_{W} - K_{RAIL}(y_{W} - y_{FC}) \\
& F_z^{**} = -F_{CZLT} - F_{CZLF} - F_{CZR} - F_{SUSPZW} + m_{W}g \\
& M_\phi^{**} = -a(F_{CZLT} + F_{CZLF} - F_{CZR}) - R_{LT}(F_{CYLT} - \psi_{W}F_{CXLT}) - R_{LF}(F_{CYLF} - \psi_{W}F_{CXLF}) \\
& - R_{R}(F_{CYR} - \psi_{W}F_{CXR}) - \psi_{W}(M_{CYLT} + M_{CYLF} + M_{CYR}) - I_{WY}\dot{\theta}_{W}\psi_{W}
\end{aligned}
$$

(2.36)

The normal forces on the left and the right wheels, F_{NLT}, F_{NLF}, and F_{NR}, act perpendicular to the contact patch plane and can be resolved into lateral and vertical components in the track plane. The resolved normal force components are:

$$
\begin{aligned}
& F_{NYLT} = -F_{NLT}\sin(\delta_{LT} + \phi_{W}) \\
& F_{NZLT} = F_{NLT}\cos(\delta_{LT} + \phi_{W}) \\
& F_{NYLF} = -F_{NLF}\sin(\delta_{LF} + \phi_{W}) \\
& F_{NZLF} = F_{NLF}\cos(\delta_{LF} + \phi_{W}) \\
& F_{NYR} = F_{NR}\sin(\delta_{R} - \phi_{W}) \\
& F_{NZR} = F_{NR}\cos(\delta_{R} - \phi_{W})
\end{aligned}
$$

(2.37)

2.1.4.3 Two - Point Wheelset Dynamic Equations

The dynamic equations of a single wheelset are obtained from Newton's laws applied to the wheelset, both for force and moments. Thus the sum of all forces acting on the wheelset in the lateral and vertical directions will equal the product of the mass and the lateral and vertical accelerations respectively. Similarly, the sum of moments acting about any axis will equal the product of the mass moment of inertia and the angular acceleration. In this case, gyroscopic torques (when angular rates about two mutually perpendicular axes are present) are to be taken into account also.

The free body diagram of the wheelset in two-point contact is given in Figure 2-8. The equations below are presented for a two-point contact condition at the left wheel / rail interface. These equations include the simplifications that since ψ_W and ϕ_W are small angles, $\cos(\psi_W)$, $\cos(\phi_W)$, $\sin(\psi_W)$, and $\sin(\phi_W)$ are replaced by 1,1, ψ_W, and ϕ_W respectively.

Lateral Equation

$$F_{CYLT} + F_{CYLF} + F_{CYR} + F_{NYLT} + F_{NYLF} + F_{NYR} + F_{SUSPYW} - m_W g \phi_W = m_W \ddot{y}_W \tag{2.38}$$

Yaw Equation

$$\begin{aligned} I_{WZ}\ddot{\psi}_W = & -I_{WY}\dot{\theta}_W\dot{\psi}_W - a(F_{CXLT} + F_{CXLF} - F_{CXR}) - \psi_W\{(a - R_{LT}\tan(\delta_{LT} + \phi_W))(F_{CYLT} + F_{NYLT}) \\ & + (a - R_{LF}\tan(\delta_{LF} + \phi_W))(F_{CYLF} + F_{NYLF}) - (a - R_R\tan(\delta_R - \phi_W))(F_{CYR} + F_{NYR})\} \\ & + M_{CZLT} + M_{CZLF} + M_{CZR} + M_{SUSPZW} + \phi_W(M_{CYLT} + M_{CYLF} + M_{CYR}) \end{aligned} \tag{2.39}$$

Left Rail Equation

$$C_{RAIL}\dot{y}_{RAIL,L} + K_{RAIL}y_{RAIL,L} = -F_{NYLT} - F_{NYLF} - F_{CYLT} - F_{CYLF} \tag{2.40}$$

Right Rail Equation

$$C_{RAIL}\dot{y}_{RAIL,R} + K_{RAIL}y_{RAIL,R} = -F_{NYR} - F_{CYR} \tag{2.41}$$

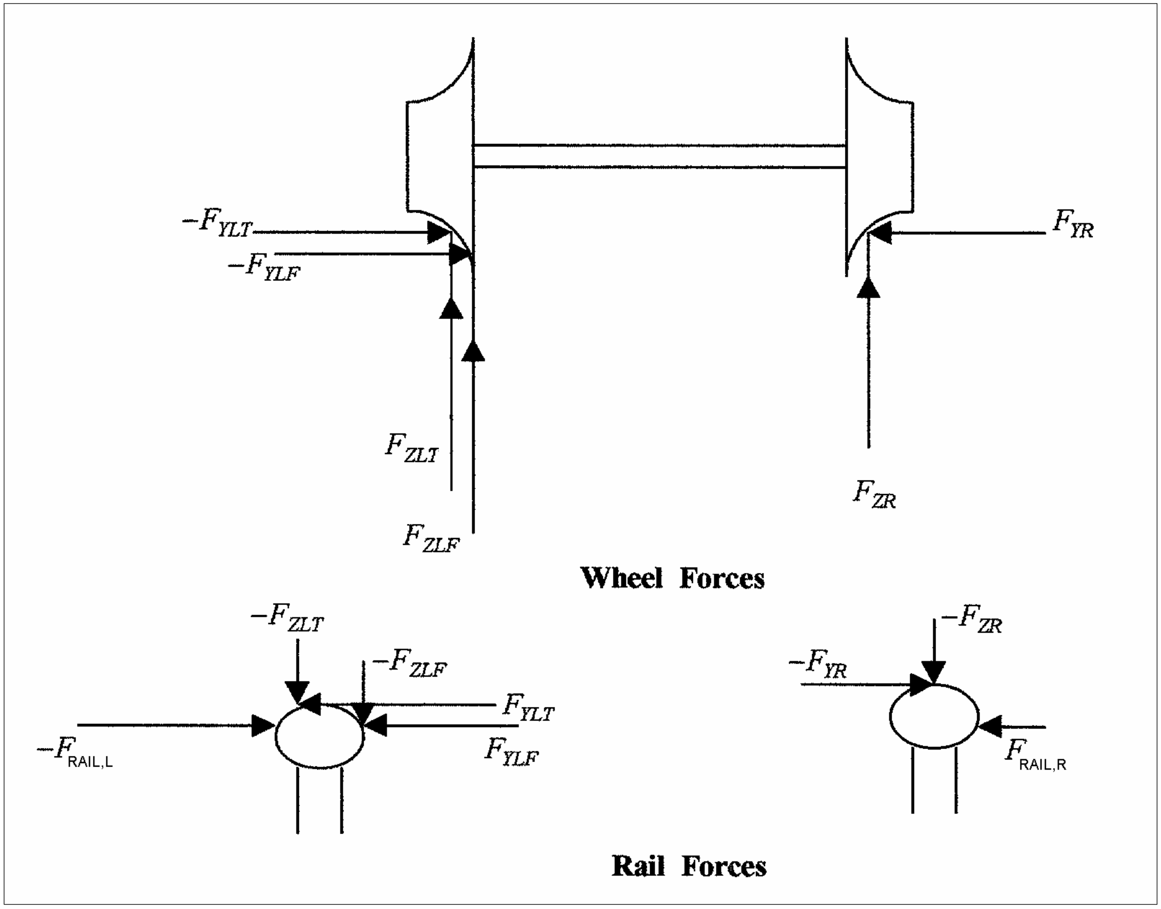


Figure 2-8 Wheel and Rail Forces for Two-Point Contact at Left Wheel / Rail

While the equations when the wheel is making two-point contact at the right rail will be similar, it will not be identical, since there is nothing symmetrical about a right handed triad. The situation, when the right wheel flange is making contact with the right rail will be identical if the set of coordinates used at the right rail is a mirror image of the left wheel coordinates reflected on the x-z plane. This will be the plane of reflection, because, the forward velocity direction and the upward direction do not change. Hence, if a left handed triad of coordinates is set up at the right rail, as shown in Figure 2-9 (forward direction of velocity V is positive x direction, upward direction is positive z, while right is positive direction for y, instead of the left direction) the equations will be identical. Such a coordinate system will have $\hat{i}_1, \hat{j}_1, \hat{k}_1$ as the unit vectors, which will be related to the right handed triad unit vectors $\hat{i}, \hat{j}, \hat{k}$ as follows:

$$\hat{i}_1 = \hat{i}; \hat{j}_1 = -\hat{j}; \hat{k}_1 = \hat{k} \tag{2.42}$$

Similarly, if ϕ, θ, and ψ are the positive angles in the right handed coordinate system and ϕ_1, θ_1, and ψ_1 are the positive angles in the left handed coordinate system, then a given angular position in the two systems will be

$$\phi_1 = -\phi\,;\ \theta_1 = \theta\,;\ \psi_1 = -\psi \tag{2.43}$$

These changes will have to be made to the all the equations (in velocities, accelerations, forces and moments), since all the equations should give results in the same frame of forward positive x direction, left positive y direction and up positive z direction.

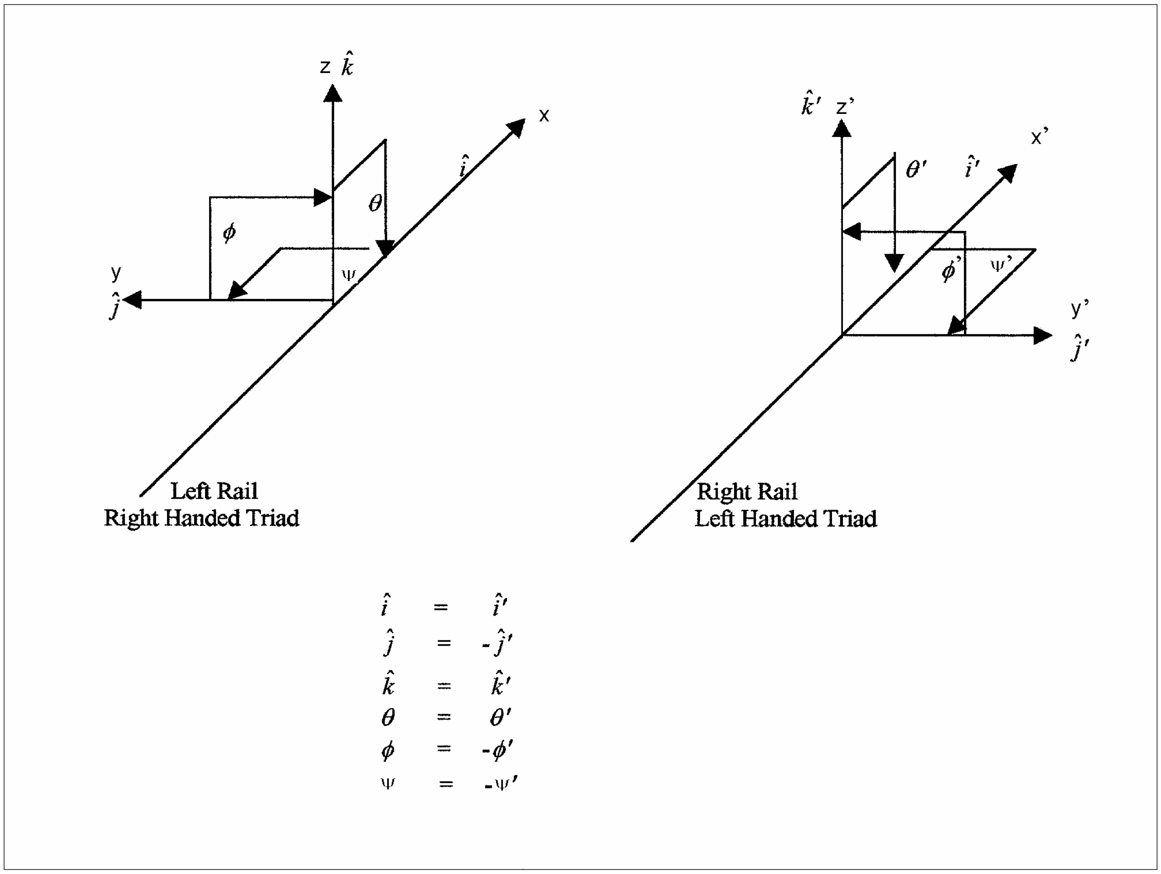


Figure 2-9 Left Handed Coordinate System at Right Rail

2.2 Numerical Simulation

The single wheelset model presented in Section 2.1 was simulated using MATLAB [2] in order to obtain the time-domain solution of the dynamics of a single wheelset moving on a flexible tangent track. This section presents a general method for wheelset analysis which accounts for both single-point and two-point wheel / rail contact conditions. This section also presents the layout of the simulation program that was used to obtain the wheelset dynamic response.

The simulations were carried out by choosing the forward speed of the wheelset as the bifurcation parameter. The critical forward speed was obtained by increasing the forward speed gradually until the response of the wheelset became marginally stable. Sensitivity of the critical velocity to suspension parameters was studied.

Simplifications were made in order to make the memory requirements less and speed up the computation. In a wheelset dynamic analysis, the lateral dynamics are very important since they determine whether or not flanging occurs. The lateral dynamics are essentially decoupled from the vertical and the longitudinal dynamics. Hence, this simulation neglects the vertical and the longitudinal dynamics of the wheelset. This assumption eliminates two degrees of freedom for each wheelset and greatly reduces computation time.

It is also assumed that the effective lateral mass of the rail, m_{RAIL}, is zero. This is justified since the rail lateral stiffness and viscous damping forces dominate. Further, it is assumed that the influence of lateral rail velocity on lateral creepage is negligible. According to British Rail [5], this assumption is reasonable since the lateral creep force is generally saturated during flange contact.

The maximum adhesion force was assumed as constant rather than calculating the creep force saturation value at each time step iteratively. This iterative process involves solving the creep and the normal force equations simultaneously, which tends to be computationally very intense. From past experience, values of creep forces were found to be considerably less than the adhesion limit.

The creep coefficients were taken to be the same for both the tread and the flange contact patches. In reality, the flange contact patch will have smaller values, but it was found that this assumption makes insignificant difference to the resulting value of the critical forward speed and the wheelset response.

Initial conditions were assumed for the lateral and yaw position and velocity of the wheelset. The initial lateral displacement and velocity of the left and the right rails were assumed to be zero. The initial conditions assumed for simulation can be found in the MATLAB program files that are included in the Appendix.

The time step for solving the dynamic wheelset equations was automatically chosen by MATLAB. The following time-domain solutions were obtained through simulation:

1. The lateral displacement and velocity of the wheelset

2. The yaw displacement and velocity of the wheelset

3. The lateral displacement of the left and the right rails.

The parametric values used for simulation are shown in Table 2-1 below.

Table 2-1 Single Wheelset Simulation Constants

Parameter	Description	Value
Wheel Type	Wheel Type	AAR 1 in 20
Wheel / Rail Constants		
f_{11T}	Creep Coefficient (Tread)	9.43e6 N
f_{12T}	Creep Coefficient (Tread)	1.20e3 N.m
f_{22T}	Creep Coefficient (Tread)	1.00e3 N.m^2
f_{33T}	Creep Coefficient (Tread)	10.23e6 N
f_{11F}	Creep Coefficient (Flange)	9.43e6 N
f_{12F}	Creep Coefficient (Flange)	1.20e3 N.m
f_{22F}	Creep Coefficient (Flange)	1.00e3 N.m^2
f_{33F}	Creep Coefficient (Flange)	10.23e6 N
λ	Conicity	0.125
μ	Coefficient of friction	0.15
Geometric Dimensions		
R_0	Centered rolling radius of wheel	0.3556 m
a	Half of track gage	0.716 m
d_p	Half distance between primary longitudinal springs	0.61 m
Weights and Moments of Inertia		
m_W	Mass of wheelset	1751 kg
I_{WY}	Pitch mass moment of inertia of wheelset	130 kg.m^2
I_{WZ}	Yaw mass moment of inertia of wheelset	761 kg.m^2
Suspension Stiffness and Damping		
K_{PX}	Primary longitudinal stiffness	Various
C_{PX}	Primary longitudinal damping	Various
K_{PY}	Primary lateral stiffness	Various
C_{PY}	Primary lateral damping	Various

A flowchart of the general algorithm used for wheelset analysis is shown in Figure 2-10. At each time step, the net lateral excursion ($y_W - y_{RAIL}$) at the left and the right wheels is checked for single-point or two-point contact condition. The flange portion of the wheel has an abrupt climb that poses numerical convergence problems during simulation. In order to counter this problem, a small lateral tolerance (y_{FCTOL}) was chosen to be used for ease of numerical integration. This tolerance provides a small lateral margin within which a two-point contact condition is assumed to occur. The single-point and two-point contact conditions are numerically described by the equations below:

If $y_W - y_{RAIL} < y_{FC}$: Single-point tread contact

If $y_{FC} \leq y_W - y_{RAIL} \leq (y_{FC} + y_{FCTOL})$: Two-point contact

If $y_W - y_{RAIL} > (y_{FC} + y_{FCTOL})$: Single-point flange contact

(2.44)

The single-point or two-point contact equations are solved at each time step depending on which out of the above conditions is satisfied. The single-point and two-point contact equations are solved by MATLAB using a fourth order Runge-Kutta integration algorithm. This method requires the equations of motion to be transformed to a system of first-order differential equations (also known as state-space equations). In order to achieve this transformation, the actual variables have been converted to first-order state-space variables as shown below:

$$\begin{aligned} x_1 &= y_W \\ x_2 &= \psi_W \\ x_3 &= \dot{y}_W \\ x_4 &= \dot{\psi}_W \\ x_5 &= y_{RAIL,L} \\ x_6 &= y_{RAIL,R} \end{aligned} \tag{2.45}$$

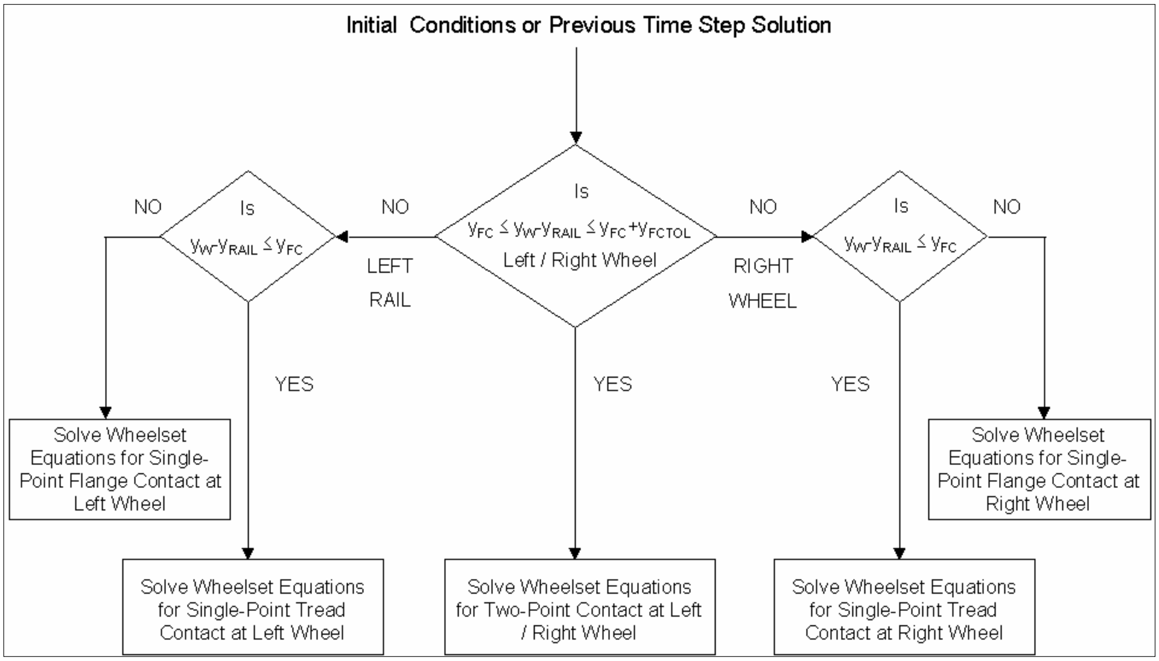

Initial Conditions or Previous Time Step Solution
Is
yFC ≤ yW-yRAIL ≤ yFC+yFCTOL
Left / Right Wheel
NO
LEFT
RAIL
NO
RIGHT
WHEEL
Is
yW-yRAIL ≤ yFC
NO
Is
yW-yRAIL ≤ yFC
NO
YES
YES
YES
Solve Wheelset Equations for Single-Point Flange Contact at Left Wheel
Solve Wheelset Equations for Single-Point Flange Contact at Right Wheel
Solve Wheelset Equations for Single-Point Tread Contact at Left Wheel
Solve Wheelset Equations for Two-Point Contact at Left / Right Wheel
Solve Wheelset Equations for Single-Point Tread Contact at Right Wheel

Figure 2-10 Single Wheelset Dynamic Analysis Algorithm

The equations of motion can now be written in state-space form as follows:

$$
\begin{aligned}
&\dot{x}_1 = x_3 && \text{(from above)}\\
&\dot{x}_2 = x_4 && \text{(from above)}\\
&\dot{x}_3 = f_1(x_1,x_2,x_3,x_4,x_5,x_6) && \text{(wheelset lateral equation)}\\
&\dot{x}_4 = f_2(x_1,x_2,x_3,x_4,x_5,x_6) && \text{(wheelset yaw equation)}\\
&\dot{x}_5 = f_3(x_1,x_2,x_3,x_4,x_5,x_6) && \text{(left rail lateral equation)}\\
&\dot{x}_6 = f_4(x_1,x_2,x_3,x_4,x_5,x_6) && \text{(right rail lateral equation)}
\end{aligned}
\tag{2.46}
$$

The MATLAB program used to simulate the dynamic behavior of a single wheelset and the functions used by the program are described below in Table 2-2. Figure 2-11 depicts the layout and interaction between the different functions at any time-step.

Table 2-2   Single Wheelset Simulation Program and Functions

Name of Program / Function	Description
SINGLE_WHEELSET	Main program. This contains the initial conditions, the global variables, and instructions for plotting the time-responses. This program also contains the instruction and conditions to solve the ordinary differential equations contained in the function file WHEELSET
EQUATIONS	This function obtains variables from function WHEELSET_SUSPENSION and uses them to solve the single wheelset equations by invoking function WHEELSET
WHEELSET	Contains the single wheelset differential equations. This function reads the rolling radius and contact angle from functions ROLLING_RADIUS and CONTACT_ANGLE respectively. This function also reads the creep forces from ONEPT_CREEP and TWOPT_CREEP (depending on single or two-point contact) and the normal forces from ONEPT_NORMAL or TWOPT_NORMAL (depending on single or two-point contact)
ONEPT_CREEP	Reads the wheelset state variables at any time-step and calculates the single-point creep forces and moments using the Vermeulen-Johnson approach with creep force saturation. The output is then passed on to the function WHEELSET
TWOPT_CREEP	Reads the wheelset state variables at any time-step and calculates the two-point creep forces and moments using the Vermeulen-Johnson approach with creep force saturation. The output is then passed on to the function WHEELSET.
ONEPT_NORMAL	Reads the wheelset state variables at any time-step and calculates the single-point normal forces and moments. The output is then passed on to the function WHEELSET

TWOPT_NORMAL	Reads the wheelset state variables at any time-step and calculates the two-point normal forces and moments. The output is then passed on to the function WHEELSET
WHEELSET_SUSPENSION	Reads the wheelset state variables at any time-step and calculates the suspension forces and moments acting on the wheelset. The output is then passed on to the function EQUATIONS
ROLLING_RADIUS	Reads the wheelset and rail lateral excursion at any time-step and calculates the rolling radius at tread and/or flange contact patches on the left and the right wheels.
CONTACT_ANGLE	Reads the wheelset and rail lateral excursion at any time-step and calculates the contact angle at tread and/or flange contact patches on the left and the right wheels.

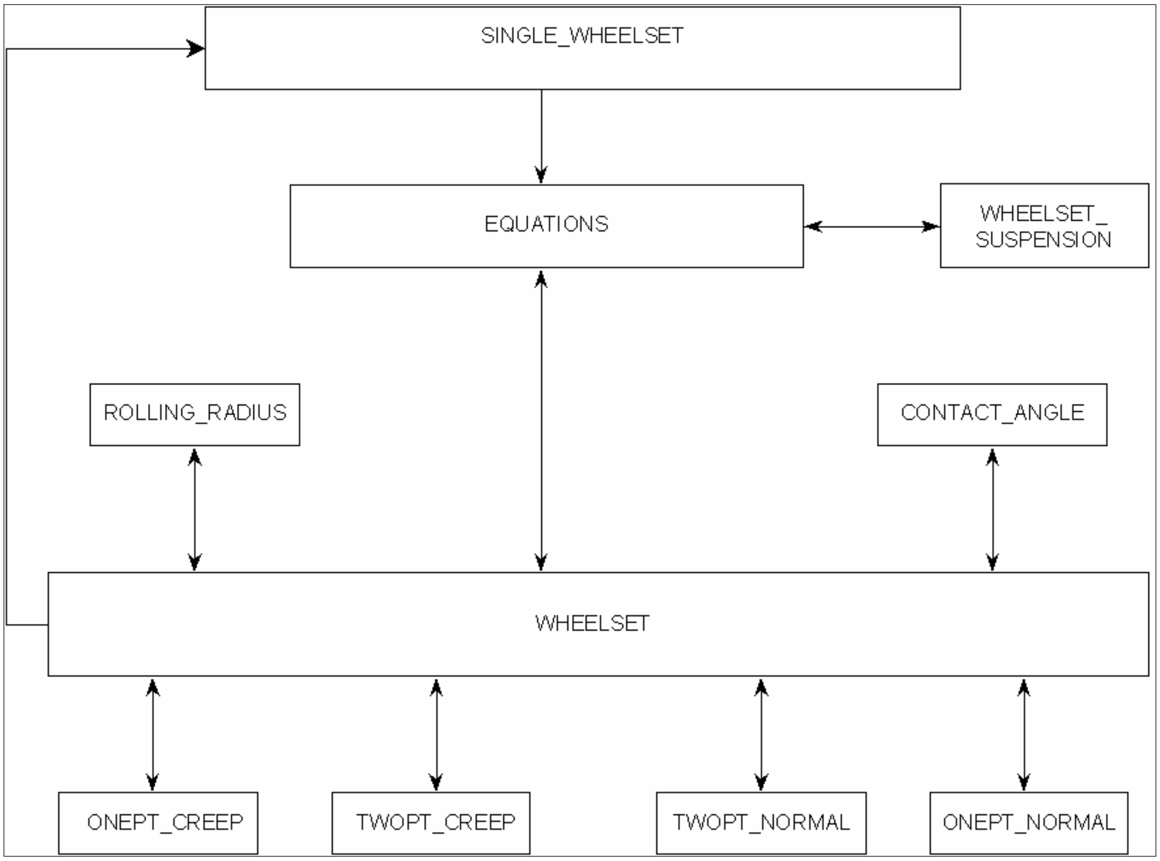


Figure 2-11 Single Wheelset Simulation Program Layout

2.3 Simulation Results

The simulation essentially involved the determination of the critical velocity under different constraints imposed on the equations enumerated in Section 2.1. The data enumerated in Table 2-1 were used except when the effect of variation of a given parameter was being investigated.

Initially, the critical velocity was determined for a single wheelset supported by the lateral suspension elements of spring and damping constants K_{PY}, C_{PY} and longitudinal constants K_{PX}, C_{PX}. The secondary suspensions were assumed to be infinite, i.e. the truck is considered to be stationary. This assumption is inherent for all the results quoted in this chapter. This assumption will ensure that the full spring and damping constants will be acting as restoring forces against the motion of the wheelset. In case the truck is not held rigidly, the movement of the truck will lead to less compression of the primary spring as compared to a fixed truck, thus leading to reduced restoring force, which is equivalent to a reduced value of the spring constant. Similarly, when the truck moves, the effective velocity is also less, being the difference between the wheelset velocity and the truck velocity, leading to reduced viscous friction damping. Thus the damping constant will also be less than when the truck is held immobile. Similarly, when the truck is not held immobile, its rotation reduces the relative angular displacement and relative angular velocity, leading to less compression of the longitudinal springs and less damping force from the viscous dampers. By holding the truck position and orientation constant, the effect of the wheelset suspension elements alone can be studied.

2.3.1 Wheels with Constant Conicity

Initially, the effect of a linear slope in the wheel (by removing the flange from the wheel and assuming a constant conicity λ) was studied. It was found from simulations that the critical velocity of the single wheelset is inversely proportional to the square root of λ. The wheel rolling radius and contact angle profiles were modified so that the wheel radius increased without any limit with a slope of λ. Thus the rolling radius, for a given lateral excursion Δy is:

$$R = R_0 \pm \lambda \Delta y \qquad (2.47)$$

In the above equation, R_0 is the centered rolling radius of the wheel. For a leftward (positive) lateral excursion of the wheelset, the '+' sign is used for the left wheel and the '-' sign is used for the right wheel. The initial values used for y_W, $\dot{y}_W$, and ψ_W were small. The forward speed of the wheelset was gradually increased in each subsequent run. In each case, the phase portrait of the lateral displacement and lateral velocity was plotted, as also the lateral displacement against time.

For low forward speeds the trajectory of the phase portrait spirals towards the origin which acts as a stable equilibrium point. When the forward speed reaches a critical value (critical velocity), the phase portrait shows a diverging trajectory, which spirals into a stable limit cycle. Figures 2-12 through 2-26 show the time response of the single wheelset before, at, and after the critical velocity has been reached for various values of λ ranging from 0.05 to 0.250. For this simulation, the values of K_{PY} and K_{PX} were taken as 1.84e5 N/m and 2.85e5 N/m respectively, though the nominal values are 5.84e6 N/m and 9.12e6 N/m respectively. The lower values are taken because as already mentioned, when the truck is present, the secondary suspension will play a large part in effectively reducing the primary suspension stiffness. Since the secondary spring stiffness is quite low, of the order of 3.0e5 N/m, the lower values of K_{PY} and K_{PX} were considered more representative. The values of C_{PY} and C_{PX} were taken as 45240 N-s/m and 83760 N-s/m respectively.

Table 2-3 and Figure 2-27 show the effect of wheel conicity on the critical velocity of a single wheelset. It is seen that the critical velocity V_C of a single wheelset is proportional to the inverse of the square root of the conicity λ, i.e.

$$V_C \propto \frac{1}{\sqrt{\lambda}} \tag{2.48}$$

This result is seen to be in agreement with the relationship given in Reference 4.

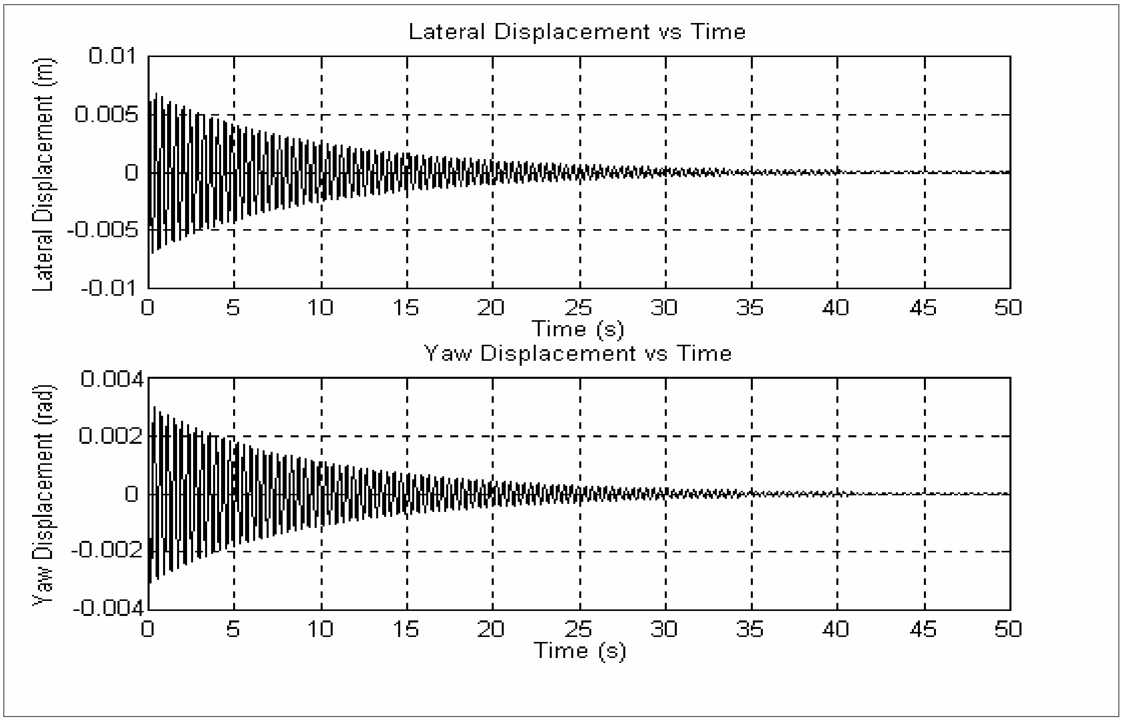


Figure 2-12 Wheelset Response: λ = 0.050, Velocity (< Vc) = 45 m/s

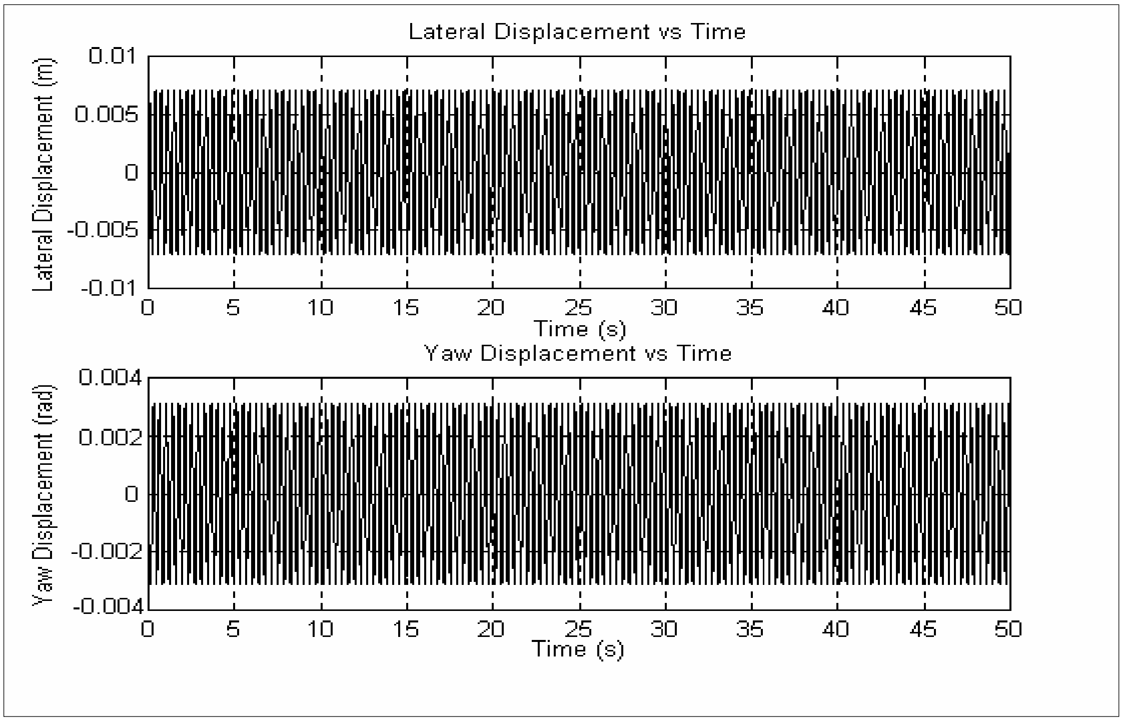


Figure 2-13 Wheelset Response: λ = 0.050, Velocity (Vc) = 50 m/s

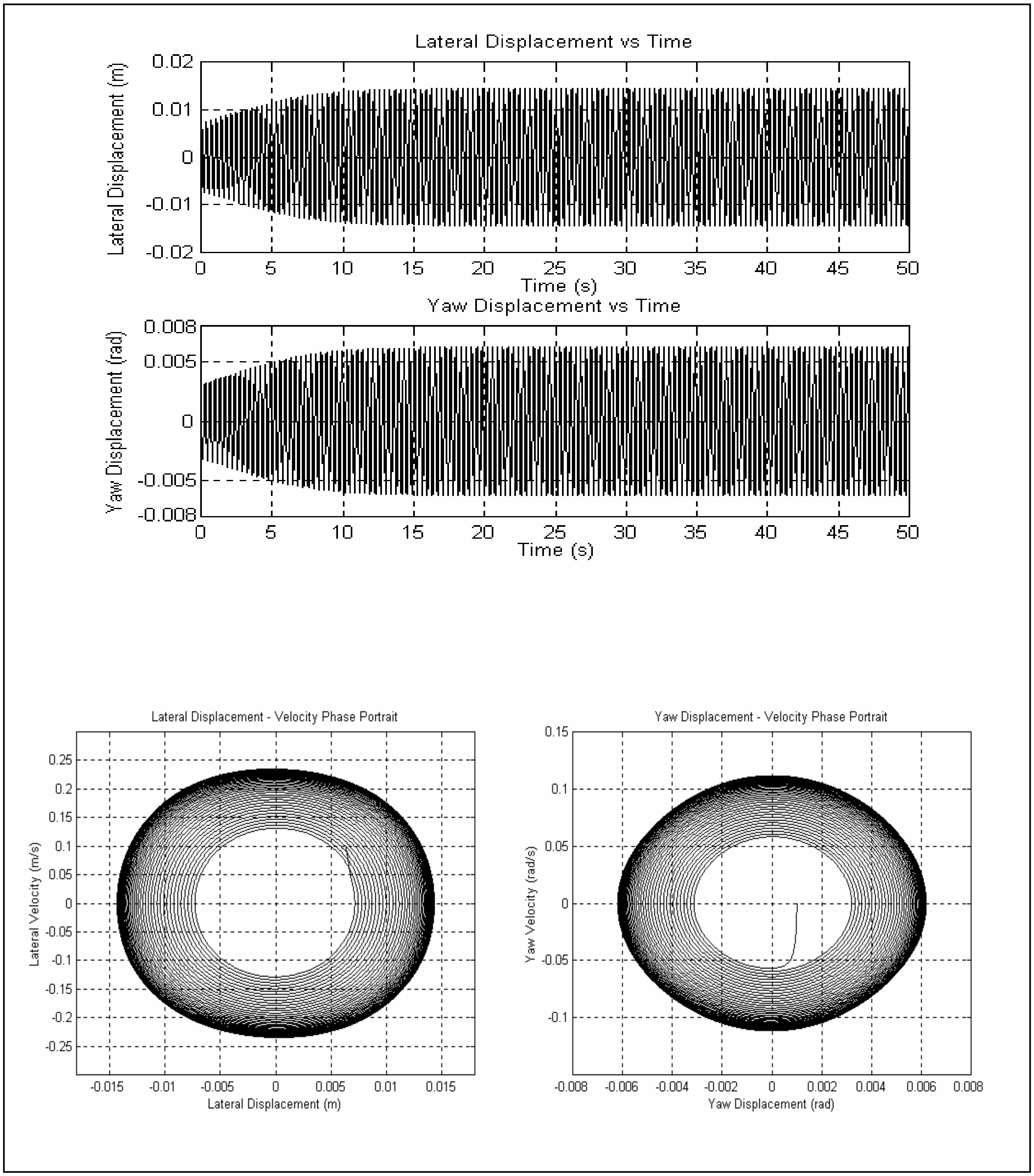


Figure 2-14 Wheelset Response: λ = 0.050, Velocity (>V_C) = 55 m/s

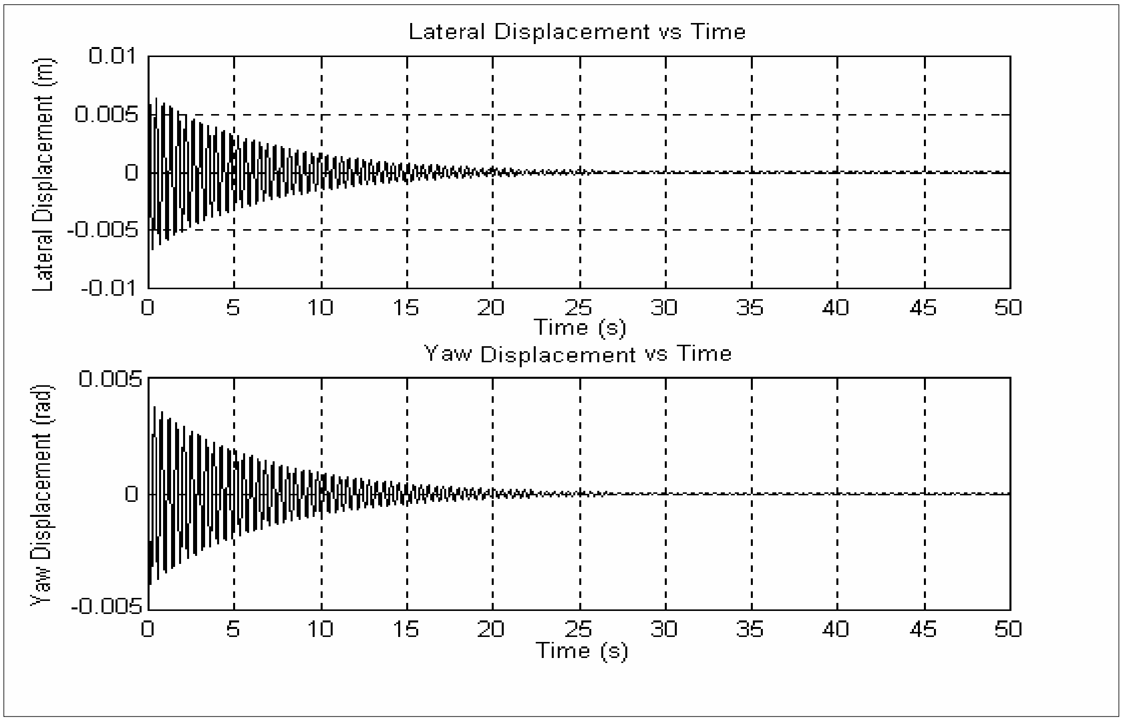


Figure 2-15 Wheelset Response: λ = 0.085, Velocity (< VC) = 30 m/s

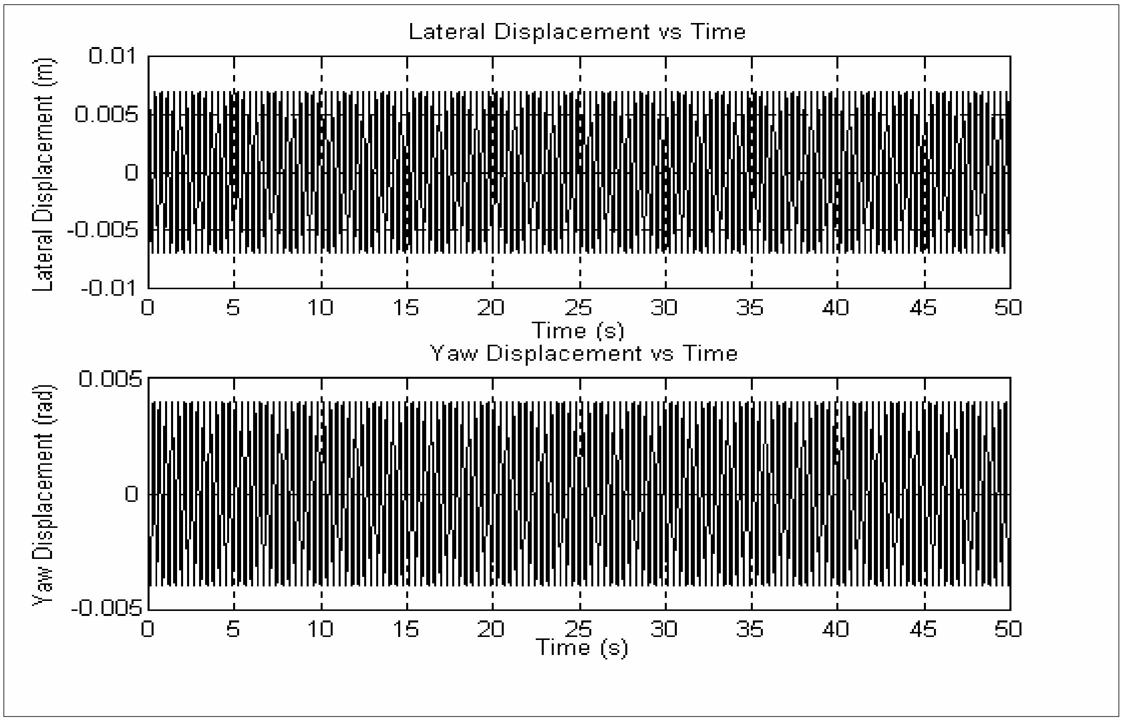


Figure 2-16 Wheelset Response: λ = 0.085, Velocity (VC) = 37 m/s

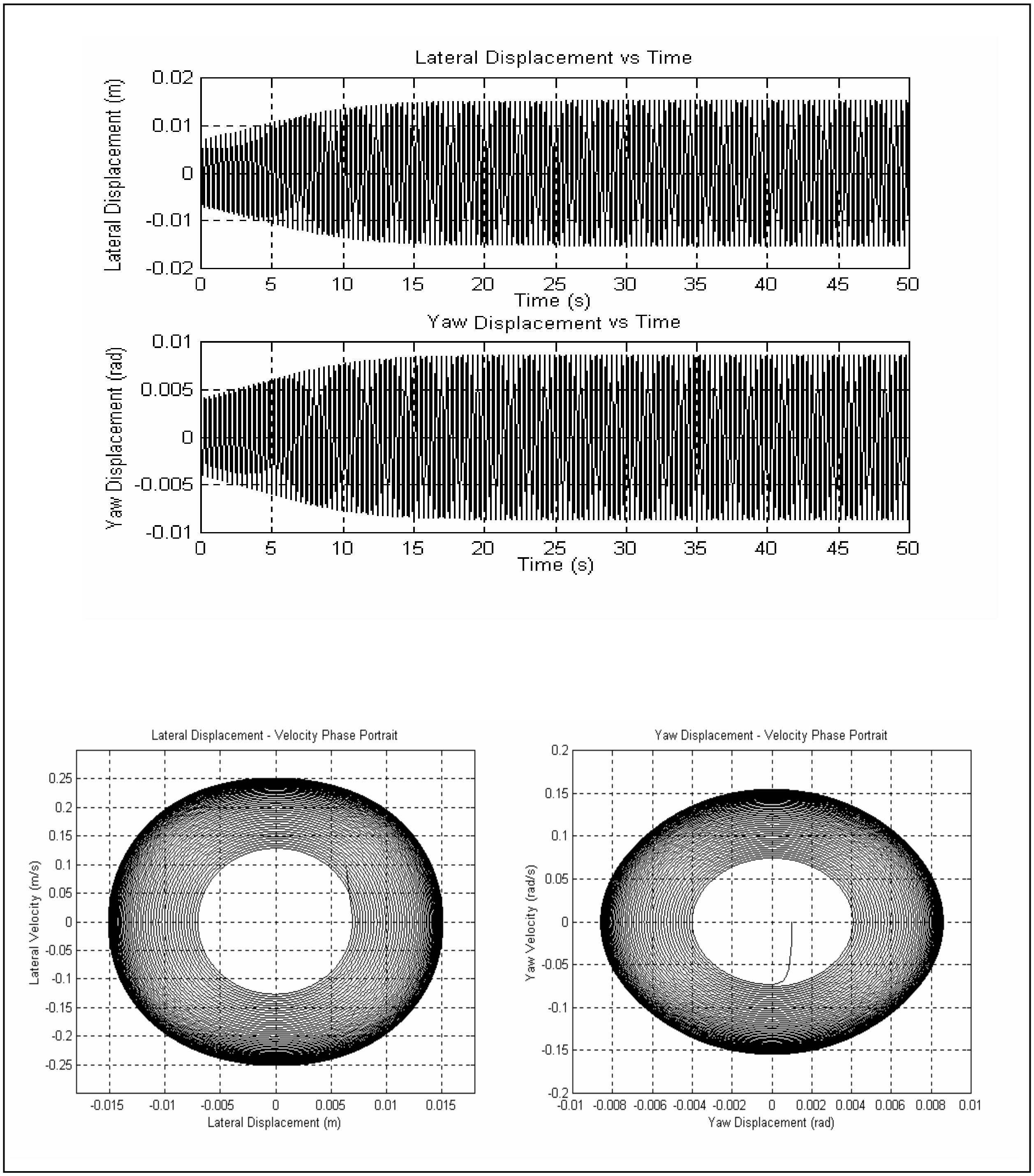


Figure 2-17   Wheelset Response: λ = 0.085, Velocity (>V$_C$) = 40 m/s

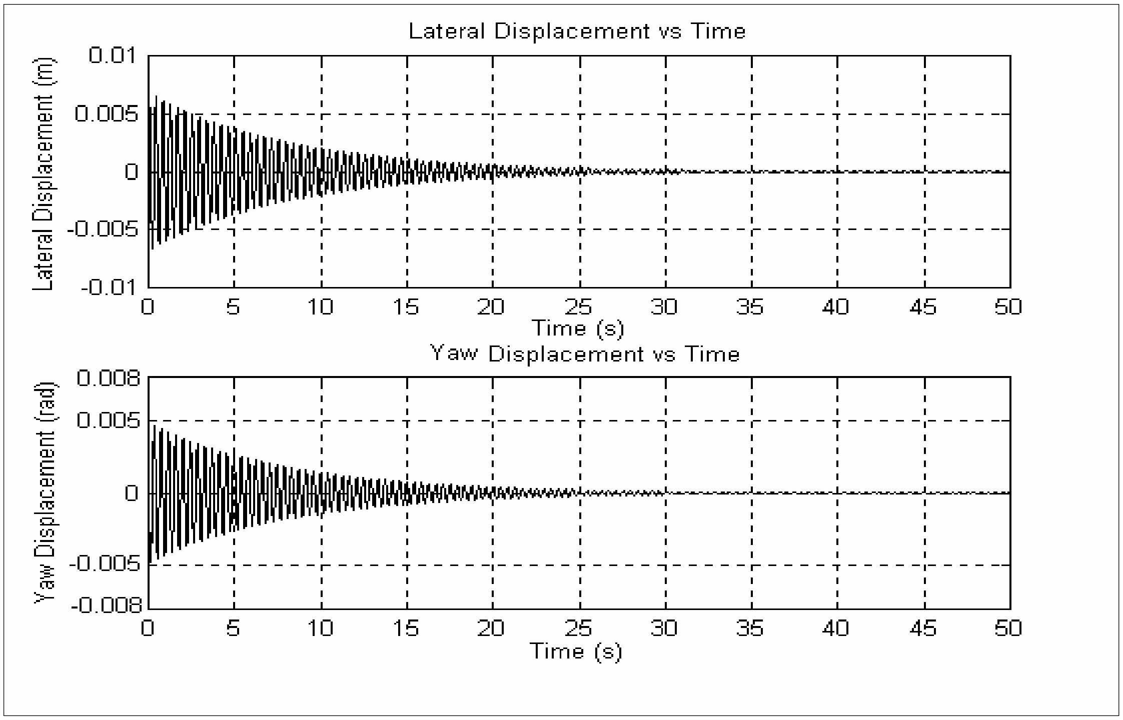


Figure 2-18 Wheelset Response: λ = 0.125, Velocity (< V_C) = 25 m/s

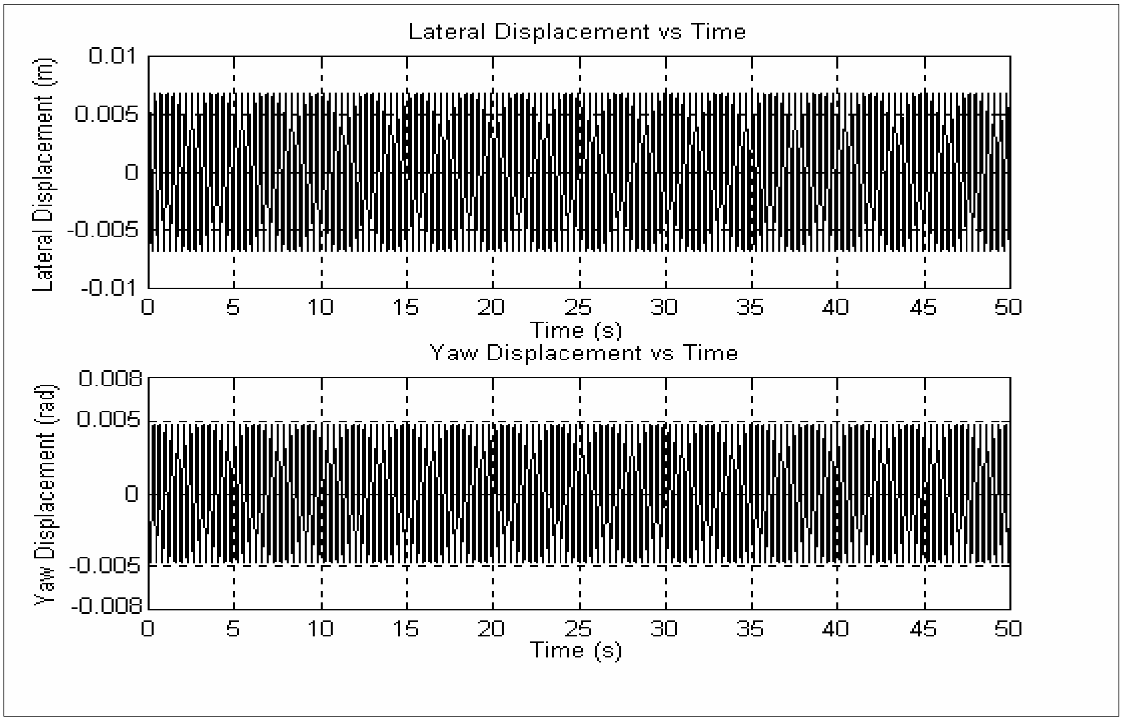


Figure 2-19 Wheelset Response: λ = 0.125, Velocity (V_C) = 30 m/s

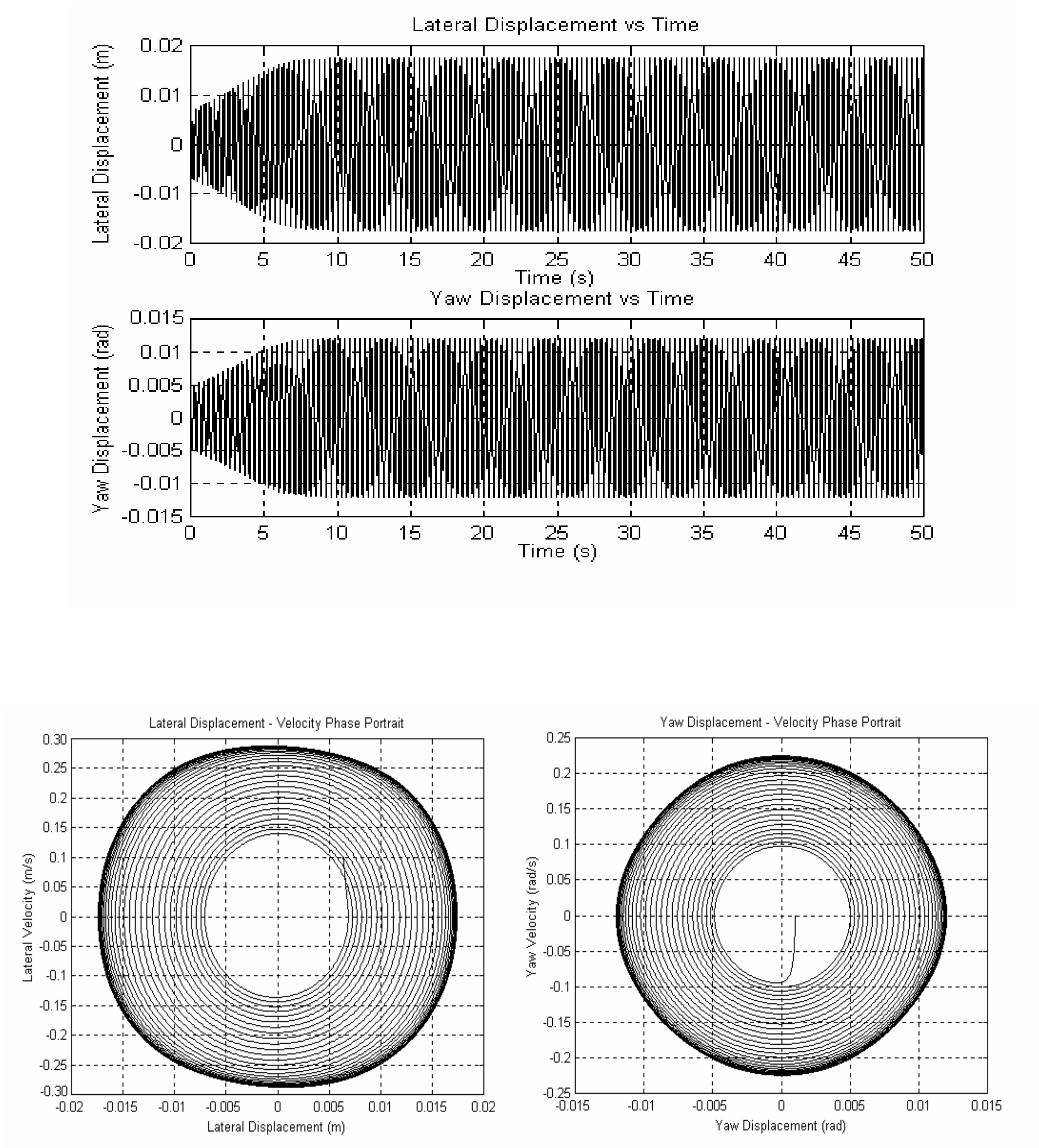


Figure 2-20 Wheelset Response: λ = 0.125, Velocity (>VC) = 35 m/s

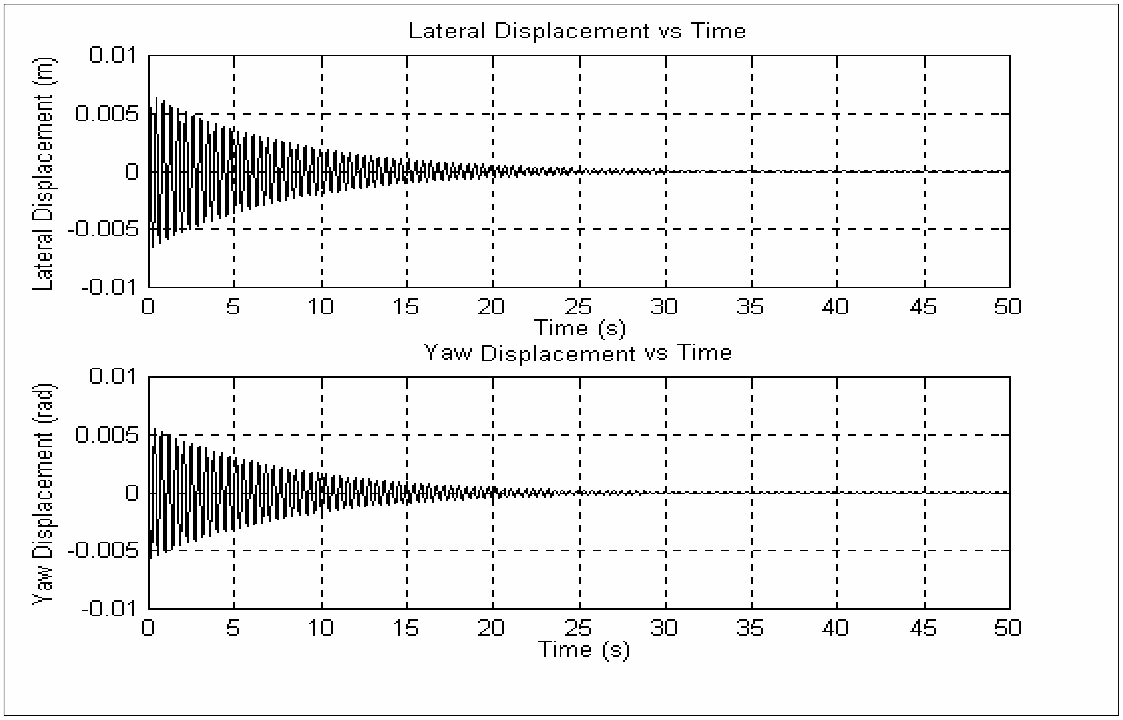


Figure 2-21 Wheelset Response: λ = 0.180, Velocity (< VC) = 20 m/s

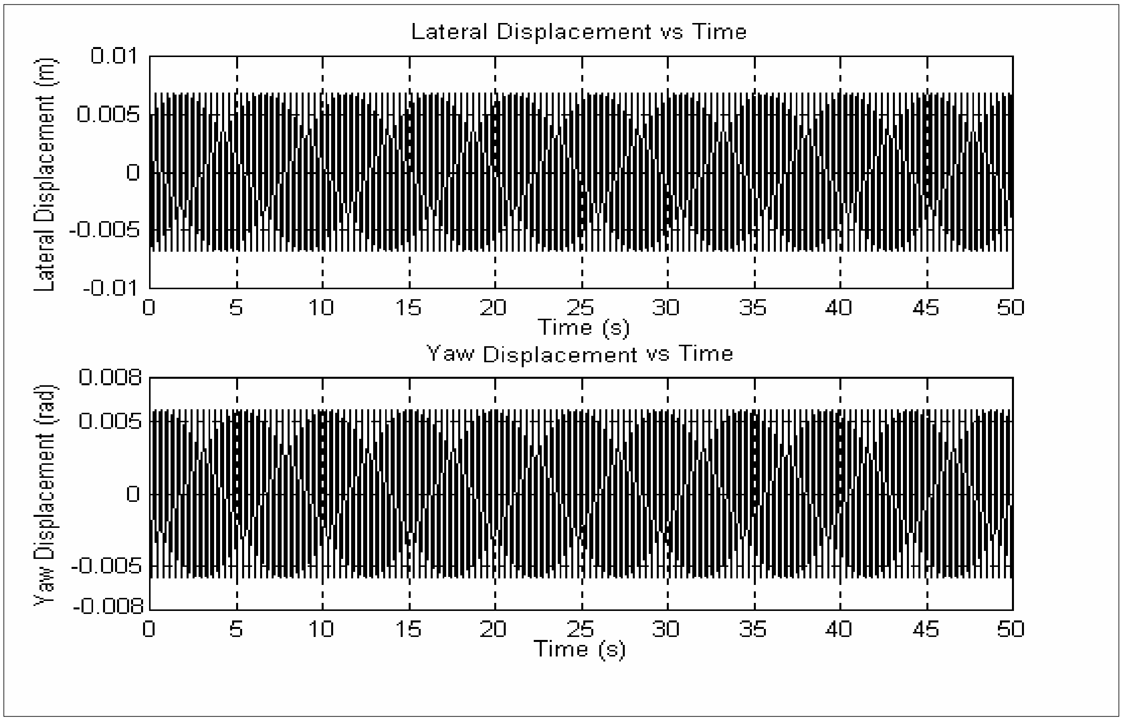


Figure 2-22 Wheelset Response: λ = 0.180, Velocity (VC) = 25 m/s

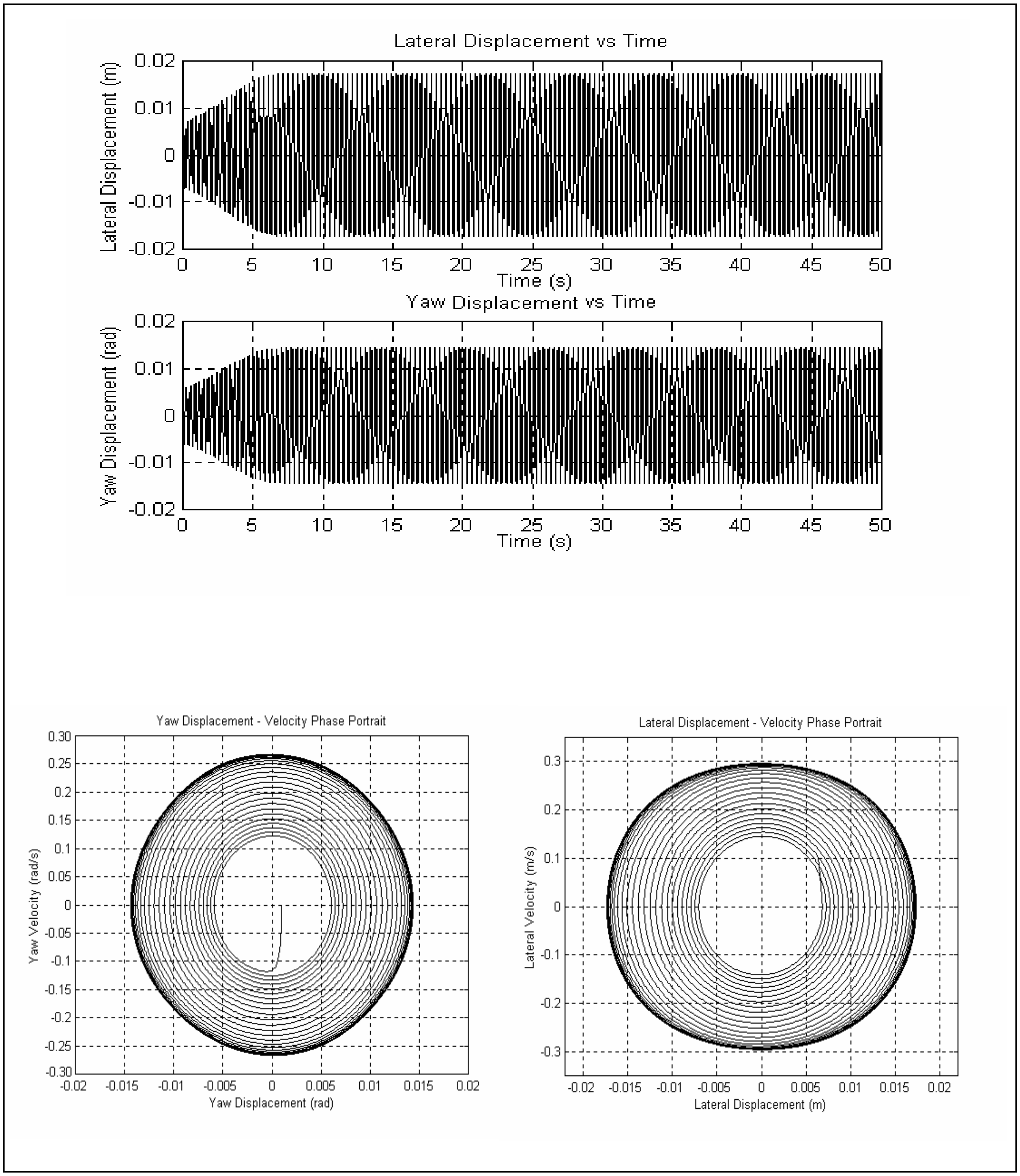


Figure 2-23 Wheelset Response: λ = 0.180, Velocity (>Vc) = 30 m/s

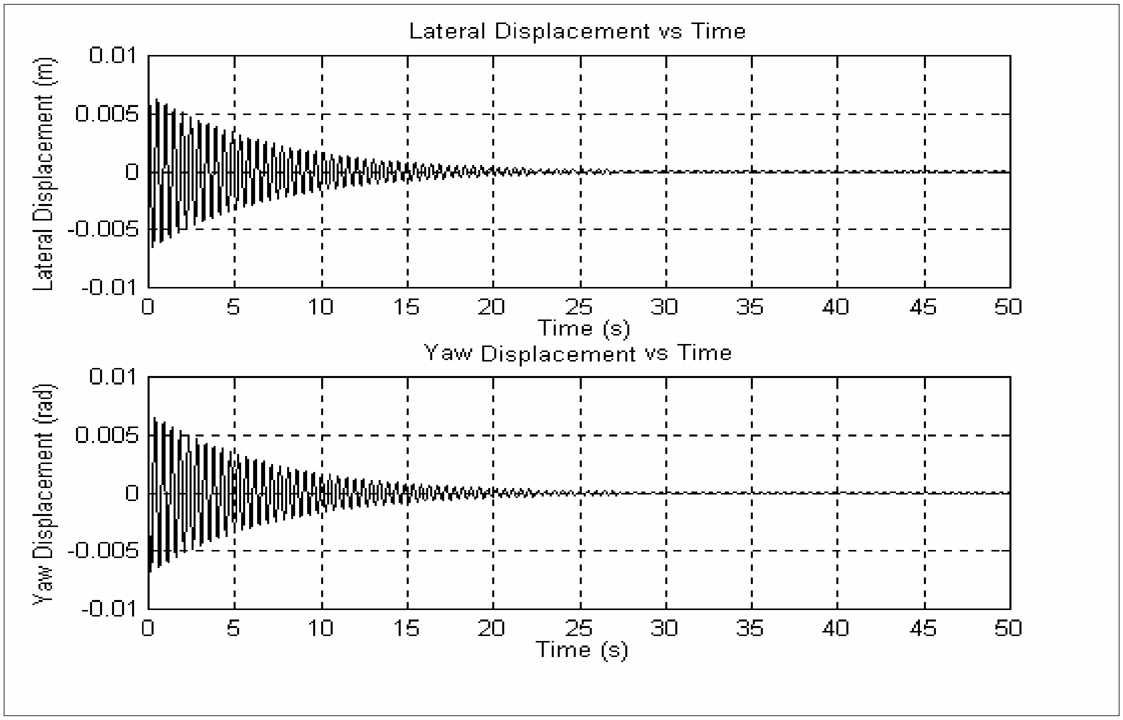


Figure 2-24 Wheelset Response: λ = 0.250, Velocity (< V_C) = 15 m/s

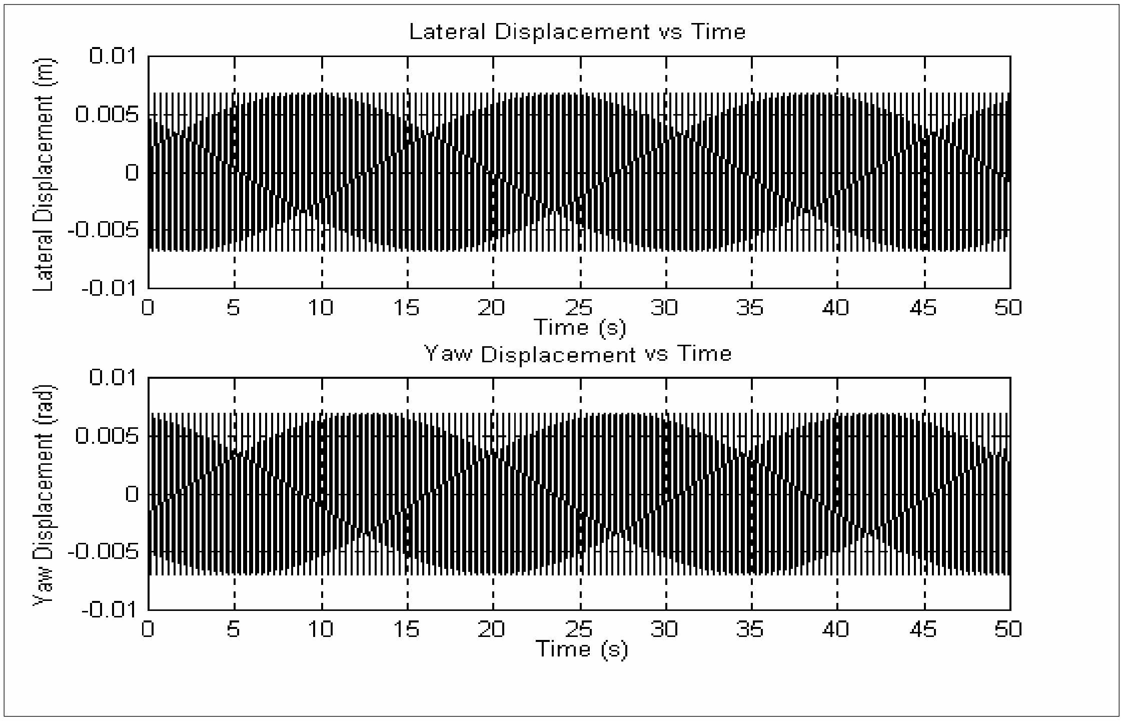


Figure 2-25 Wheelset Response: λ = 0.250, Velocity (V_C) = 22 m/s

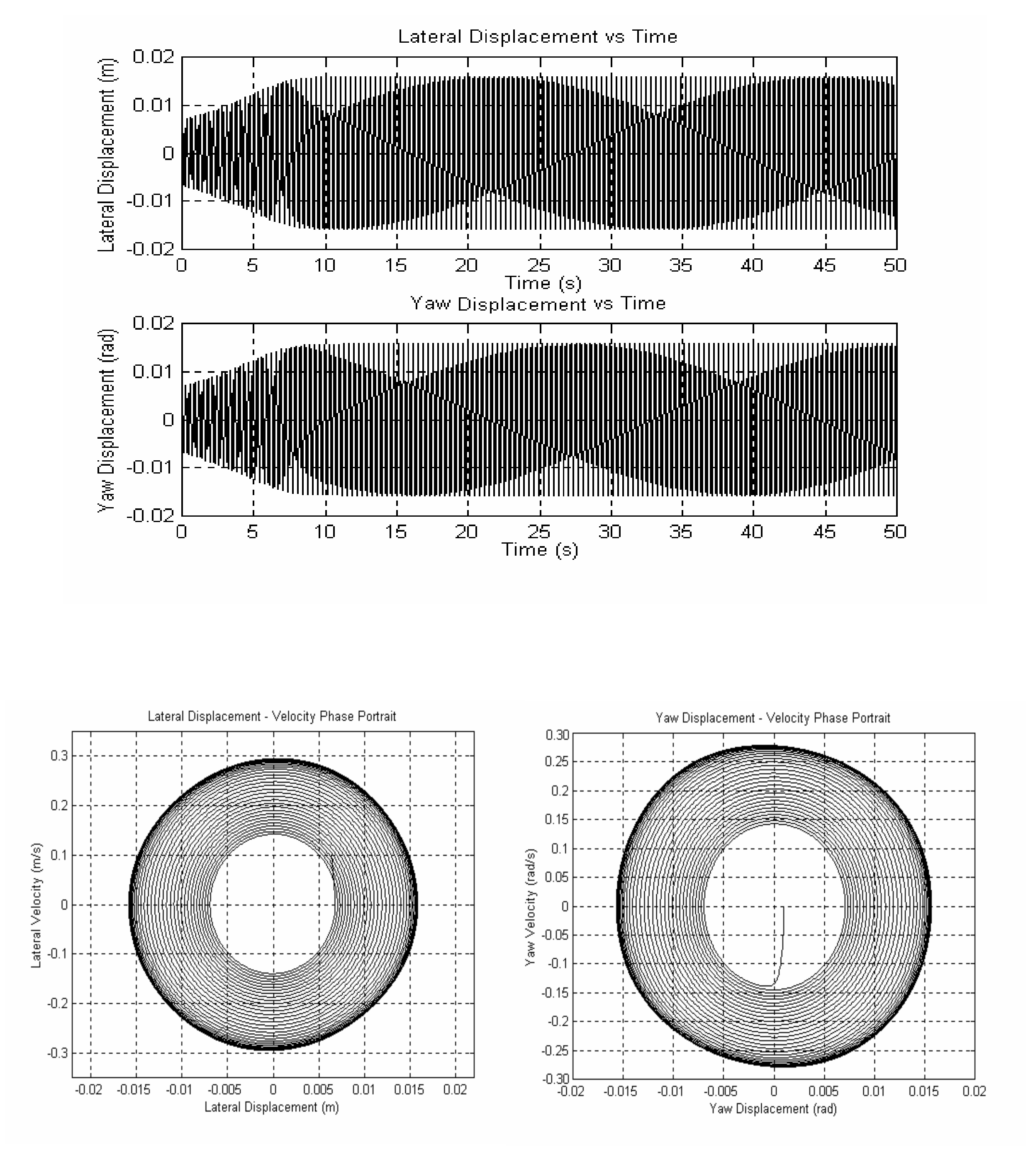


Figure 2-26 Wheelset Response: $\lambda = 0.250$, Velocity ($>V_C$) = 25 m/s

Table 2-3 Effect of Conicity on the Critical Velocity of a Single Wheelset

Conicity (λ)	Critical Velocity (V_C), m/s (km/hr)
0.05	50 (180)
0.085	37 (133)
0.125	30 (108)
0.180	25 (90)
0.250	22 (79)

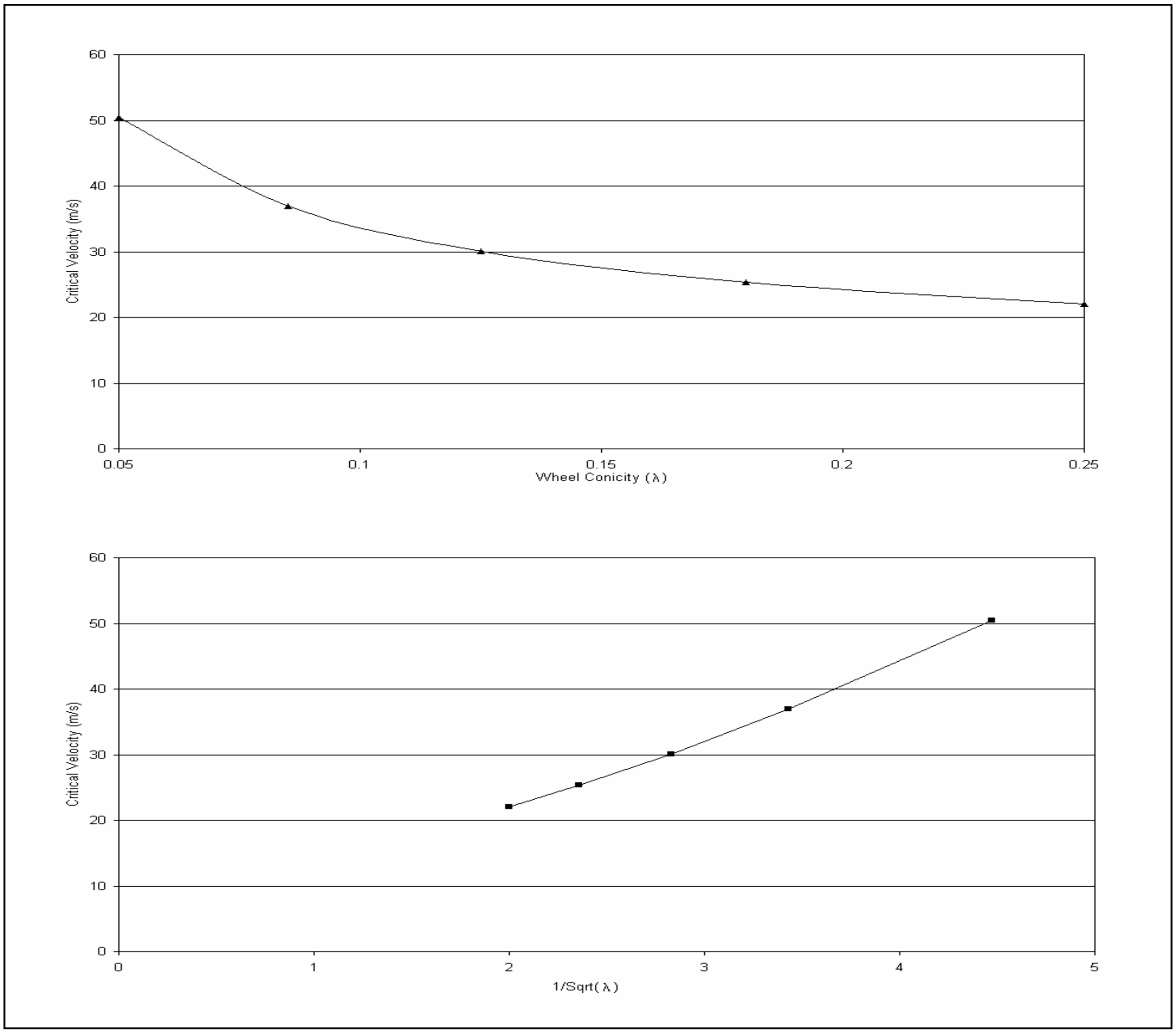


Figure 2-27 Effect of Conicity on the Critical Velocity of a Single Wheelset

2.3.2 Introduction of the Flange

It is seen from the constant conicity simulations that in the case of a wheel with linear taper, the maximum lateral excursion of the wheelset can be well in excess of 10 mm. But in reality, the wheel has a sharp flange. When the flange comes in contact with the rail, the corresponding rolling radius increases sharply, leading to a large yawing moment on the wheel. This causes the wheel to turn sharply towards the other rail. Though the wheel climbs part of the flange, unless it hits the rail with considerable velocity, it does not fully climb the flange. As already mentioned, though the actual AAR wheel has a very sharp flange, in this simulation it is modeled to rise through the flange height over a lateral distance of 1 mm. Because of the bouncing of the wheel, whenever the flange comes into contact with the rail, the wheel as well as the rail movement shows a spike. This spike occurs because of the very high values of spring constant of the rail.

2.3.2.1 The Effect of Primary Spring Stiffness on the Critical Velocity

The critical velocity of a single wheelset with the AAR wheels was determined for various values of primary spring longitudinal and lateral stiffness by varying the vehicle forward velocity. When the values of K_{PY} and K_{PX} are low (i.e. of the order of e4 or e5), the transition from damped motion to increasing amplitude oscillations is found to be rather sudden. The hunting abruptly increases to such an extent that the flanges hit the rails, often, turning the wheelset back to touch the opposite flange. But when the values of K_{PY} and K_{PX} are high, there are several stable limit cycles within the linear or tread region itself. There is a wide interval of vehicle speeds, between spiraling to a stable center to a limit cycle hunting increasing in amplitude until the flange is reached.

The initial conditions used for the simulations in this section are:

Lateral displacement = 0.00635 m, Lateral velocity = 0.010 m/s
Initial yaw angle = 0.001 rad, Initial yaw velocity = 0 rad/s.

Figures 2-28 and 2-29 depict the wheelset and rail time-response at critical velocity (64.155 m/s) for K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, and C_{PY} = 9048.2 N-s/m. Figures 2-30 and 2-31 show the wheelset and rail response for the same set of parameters at a forward speed of 70 m/s. At this speed, it is seen that the wheels are flanging for a majority of the simulation time. In this case, the amplitude of the stable limit cycle is larger than the flange clearance. However, the lateral motion of the wheelset is limited by the flange width on both the left and the right wheels.

For high values of K_{PX} or K_{PY}, such as 2.85e6 N/m or 5.84e6 N/m respectively, the amplitude of the limit cycle is smaller than the flange clearance between the wheel flange and the rail at lower velocities. As the forward velocity is increased, the limit cycle amplitude increases until it equals the flange clearance. Hence, in this case, a wide range of critical velocities is associated with a corresponding range of limit cycle amplitudes, culminating in an amplitude equaling the flange clearance.

Figure 2-32 shows the wheelset lateral time response for K_{PX} = 2.85e4 N/m, K_{PY} = 5.84e6 N/m, C_{PX} = 8376.9 N-s/m, and C_{PY} = 9048.2 N-s/m. The lateral solution is stable at a velocity of 105 m/s. At 115 m/s, the solution exhibits a limit cycle oscillation with an amplitude of 0.0024 m. As the forward velocity is increased to 150 m/s, the limit cycle amplitude gradually increases to 0.00535 m. The amplitude of the limit cycle grows to become equal to the flange clearance at a velocity greater than 200 m/s.

The sensitivity of critical velocity to primary spring stiffness is tabulated below in Table 2-4. The values of K_{PY} and K_{PX} are varied in geometric progression, with a common ratio of 3.14. This leads to every alternate value being a multiple or a sub-multiple by a factor of 10. High critical velocity values seen in Table 2-4 are purely theoretical and occur because a single wheelset is assumed. In the actual vehicle, the truck is attached to the carbody by secondary springs, which act to effectively reduce the primary spring constants, and limit the critical velocities. Figures 2-33 and 2-34 plot the variation of critical velocity with longitudinal and lateral spring stiffness.

From Table 2-4 and Figures 2-33 and 2-34, it is seen that the variation of critical velocity is more sensitive to K_{PX} for lower values of K_{PY}. At higher values of K_{PY} the variation of critical velocity with K_{PX} is practically linear. This conclusion also holds true when the critical velocity is varied with K_{PY} for different values of K_{PX}. However, higher longitudinal and lateral spring stiffness always result in higher critical velocities.

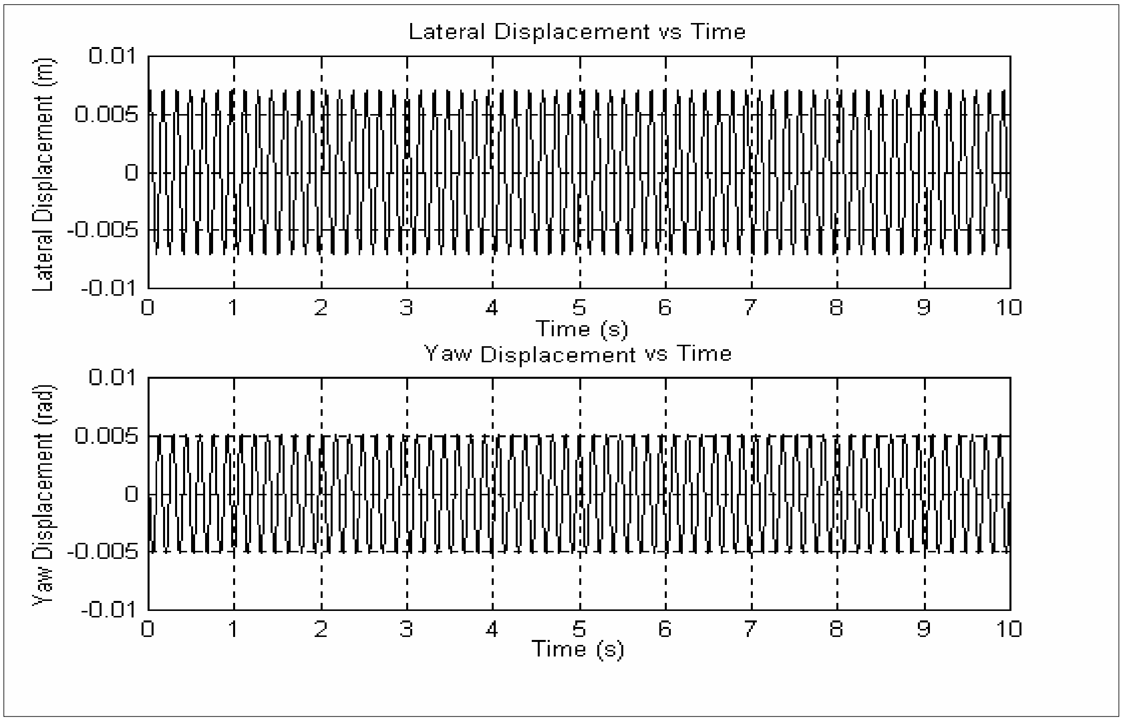


Figure 2-28 Wheelset Response: K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (V_C) = 64 m/s

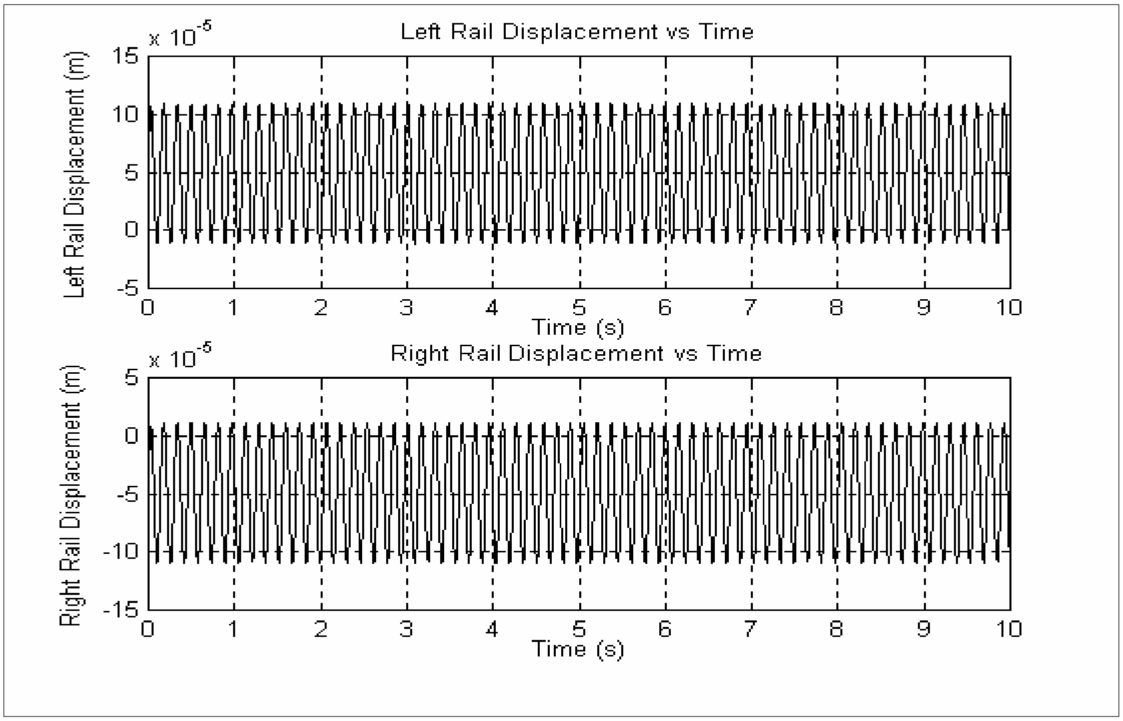


Figure 2-29 Rail Response: K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (V_C) = 64 m/s

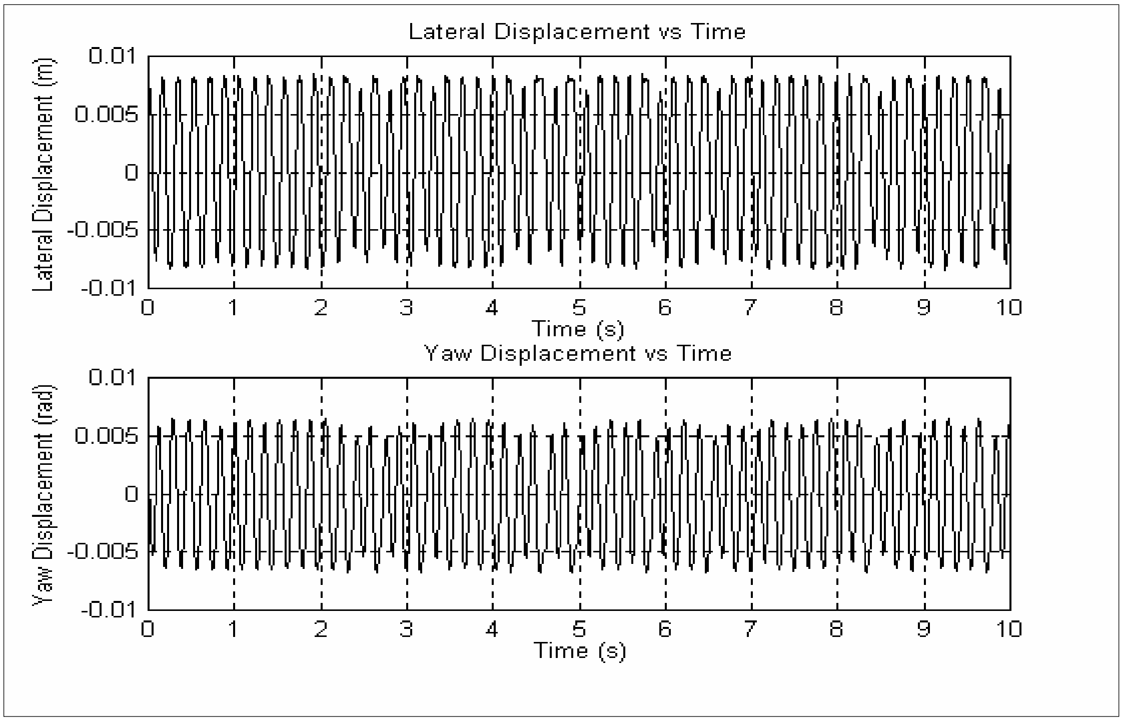


Figure 2-30 Wheelset Response: KPX = 2.85e6 N/m, KPY = 1.84e5 N/m, CPX = 8376.9 N-s/m, CPY = 9048.2 N-s/m, Velocity (>VC) = 70 m/s

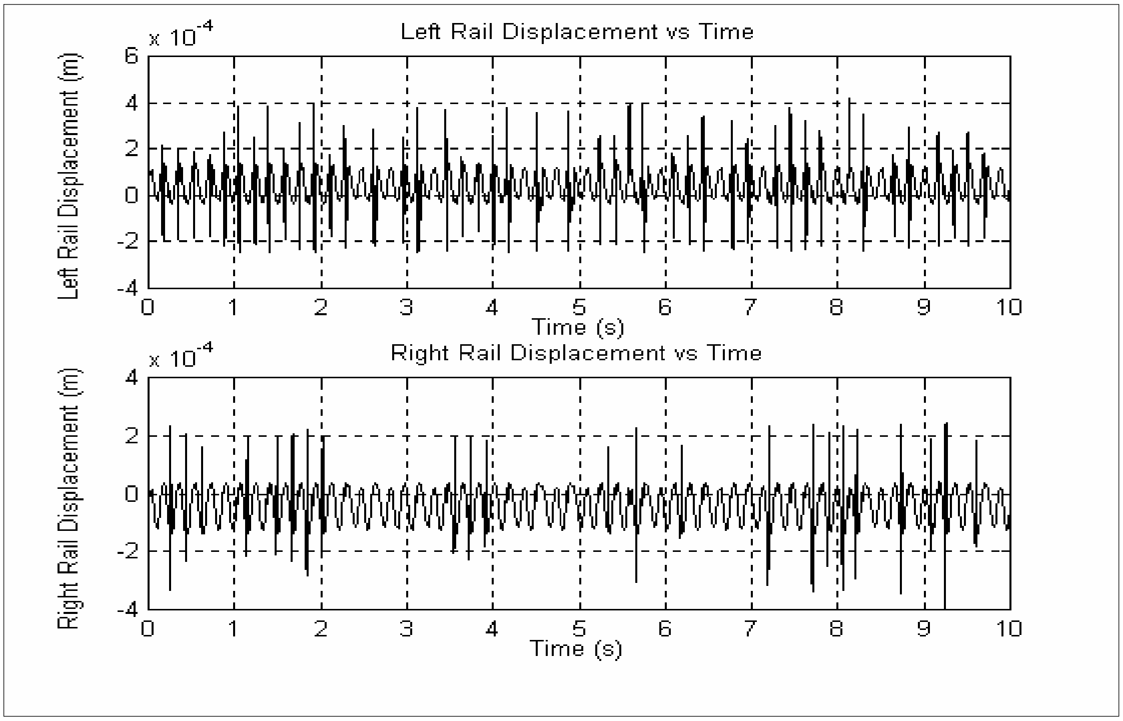
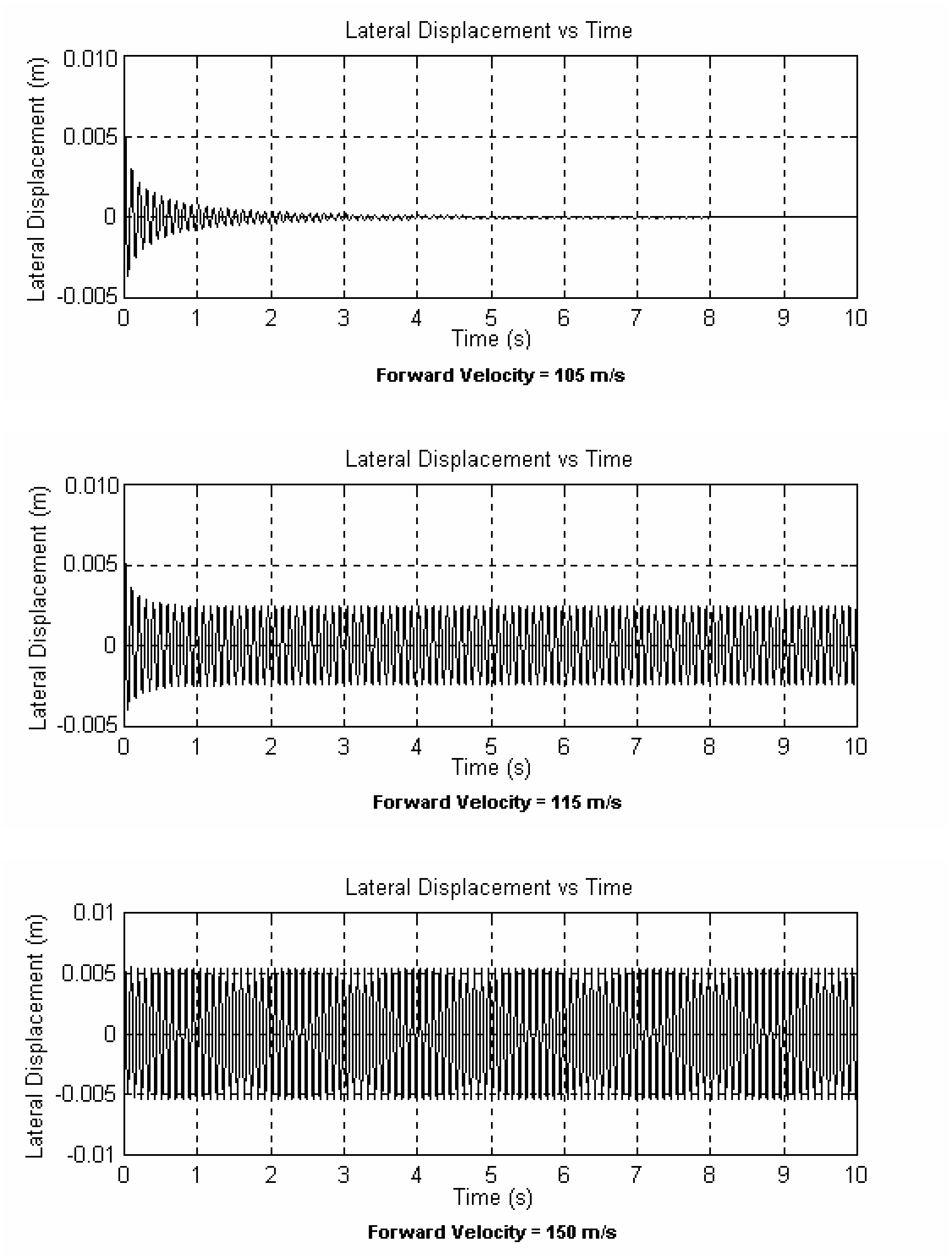


Figure 2-31 Rail Response: KPX = 2.85e6 N/m, KPY = 1.84e5 N/m, CPX = 8376.9 N-s/m, CPY = 9048.2 N-s/m, Velocity (>VC) = 70 m/s



Forward Velocity = 105 m/s

Forward Velocity = 115 m/s

Forward Velocity = 150 m/s

Figure 2-32 Wheelset Response: KPX = 2.85e4 N/m, KPY = 5.84e6 N/m, CPX = 8376.9 N-s/m, CPY = 9048.2 N-s/m

Table 2-4 Sensitivity of Single Wheelset Critical Velocity to Primary Longitudinal and Lateral Spring Stiffness

K_{PX} (N/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Wheelset (V_C), m/s (km/hr)			
2.85e4	11 (40)	18 (65)	31 (112)	56 (202)
9.12e4	14 (50)	20 (72)	32 (115)	56 (202)
2.85e5	21 (76)	25 (90)	35 (126)	58 (209)
9.12e5	35 (126)	37 (133)	45 (162)	64 (230)
2.85e6	63 (227)	64 (230)	69 (248)	82 (295)

Note: All values in Table 2-4 above have been obtained with C_{PY} = 9048.2 N-s/m and C_{PX} = 8376.9 N-s/m.

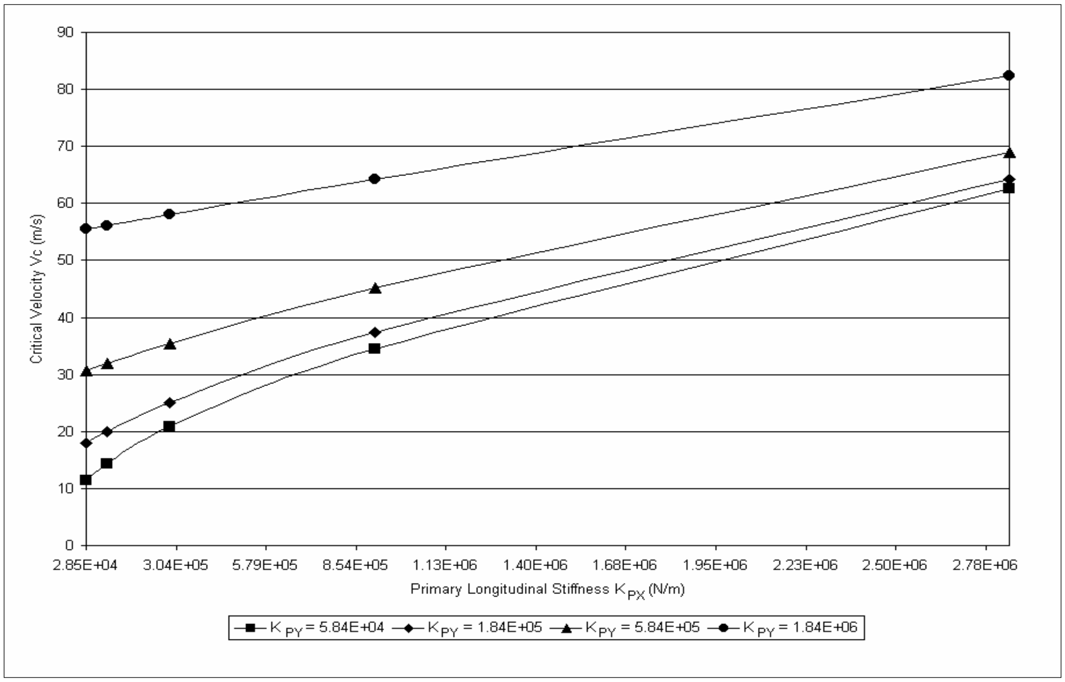


Figure 2-33 Variation of Single Wheelset Critical Velocity (VC) with Primary Longitudinal Spring Stiffness (KPX)

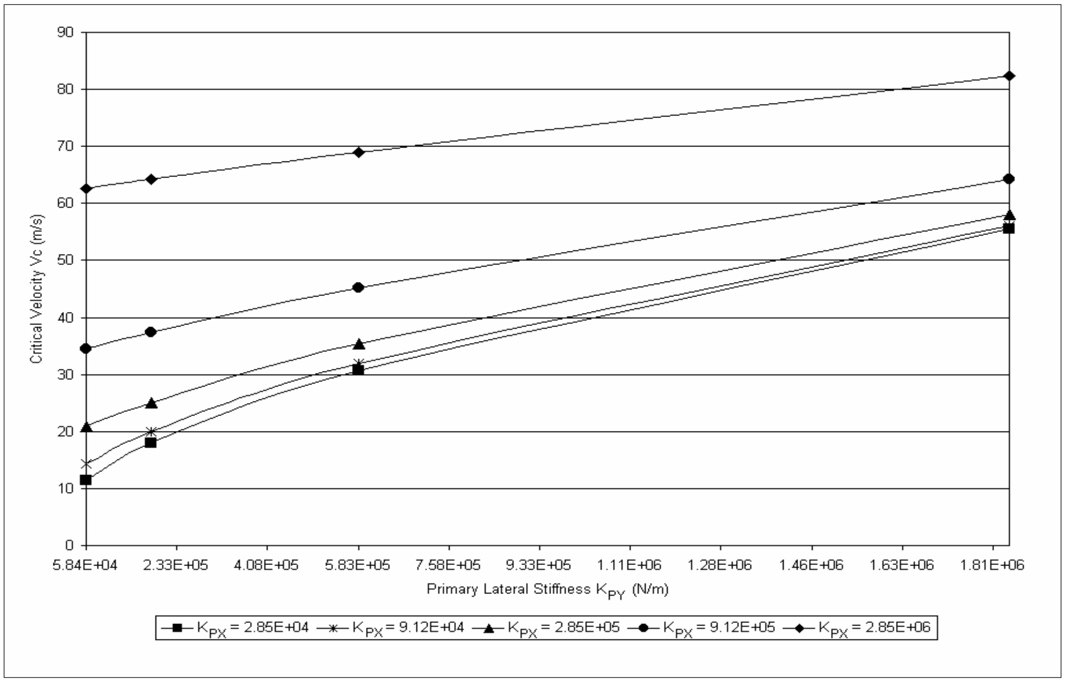


Figure 2-34 Variation of Single Wheelset Critical Velocity (VC) with Primary Lateral Spring Stiffness (KPY)

2.3.2.2 The Effect of Primary Damping on the Critical Velocity

The effect on the single wheelset critical velocity of varying C_{PX} and C_{PY} is illustrated in Tables 2-5 and 2-6 respectively. The tables show the effect of varying the damping factors while keeping the values of K_{PX} and K_{PY} fixed. The data contained in these tables are presented graphically in Figures 2-43 and 2-44.

Figures 2-35 and 2-36 depict the wheelset and rail time-response at critical velocity (18.473 m/s) for K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 41880 N-s/m, and C_{PY} = 9048.2 N-s/m. Figures 2-37 and 2-38 show the wheelset and rail response for the same set of parameters at a forward speed of 20 m/s. At this speed, it is seen that the wheels are flanging for a majority of the simulation time. In this case, the amplitude of the stable limit cycle is larger than the flange clearance. However, the lateral motion of the wheelset is limited by the flange width on both the left and the right wheels.

Figures 2-39 and 2-40 depict the wheelset and rail time-response at critical velocity (17.606 m/s) for K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, and C_{PY} = 1810 N-s/m. Figures 2-41 and 2-42 illustrate flanging for the same set of parameters at a forward speed of 20 m/s.

From Figures 2-43 and 2-44, it is seen that the variation of critical speed with both C_{PX} and C_{PY} is an almost linear relationship. The single wheelset critical velocity is not nearly as sensitive to primary damping as it is to primary stiffness. Higher values of longitudinal and lateral damping result in higher critical velocities. Also, larger spring constants result in higher critical velocities as well as a higher sensitivity to primary damping. Note that the high critical velocities listed in Tables 2-5 and 2-6 are due to the fact that the spring constants are unrealistically large in a single wheelset, and will be considerably reduced in a full vehicle due to the presence of secondary suspension elements.

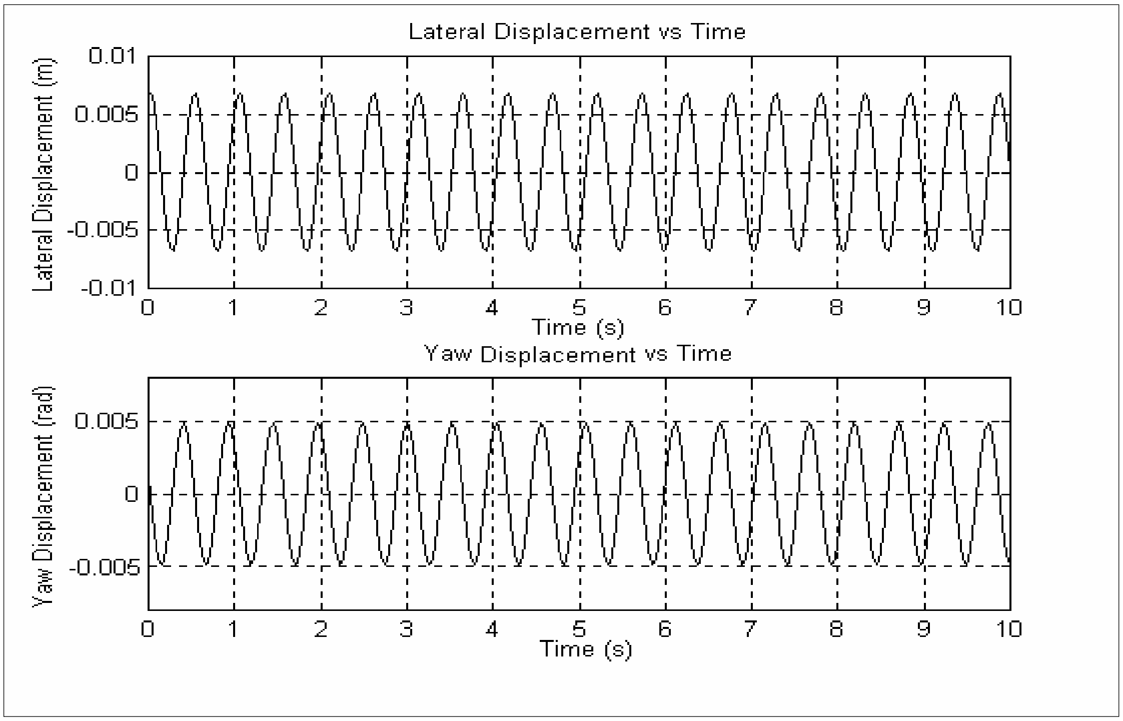


Figure 2-35 Wheelset Response: K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 41880 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (V_C) = 18 m/s

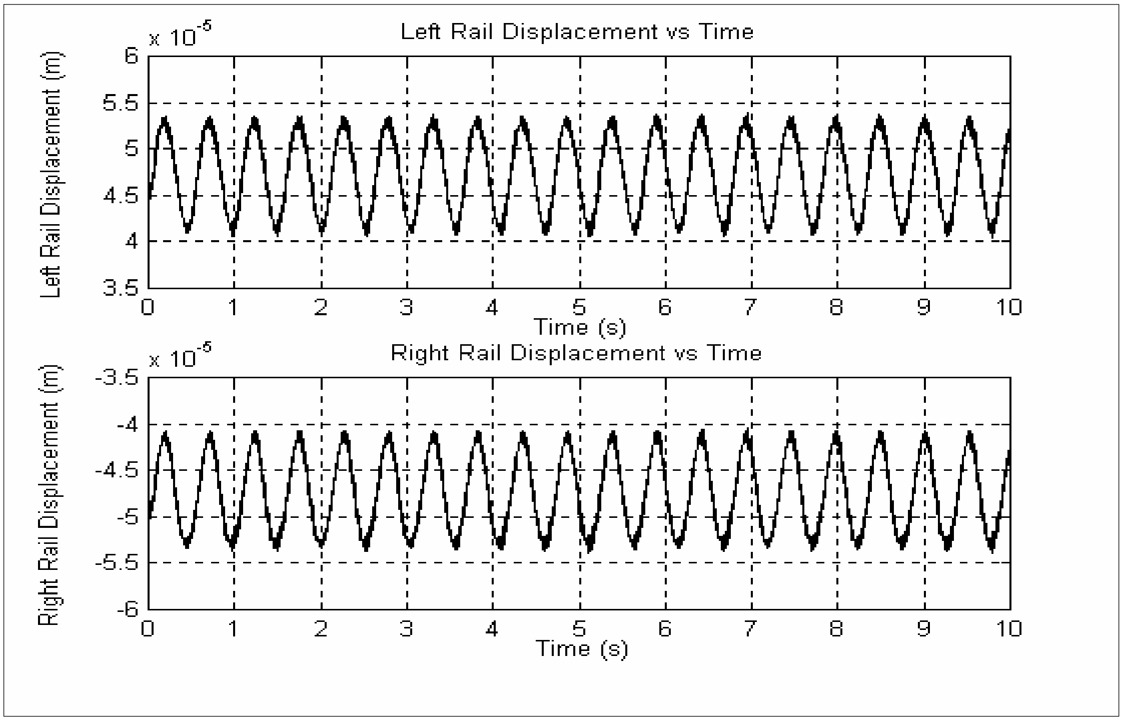


Figure 2-36 Rail Response: K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 41880 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (V_C) = 18 m/s

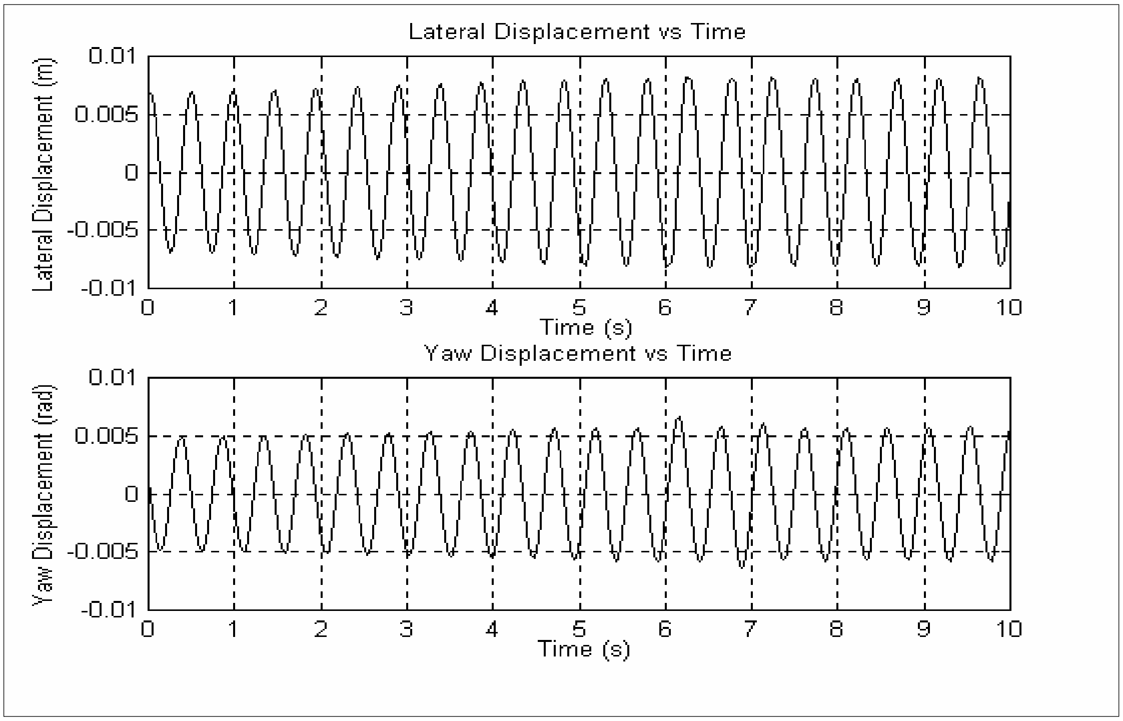


Figure 2-37 Wheelset Response: K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 41880 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (>V_C) = 20 m/s

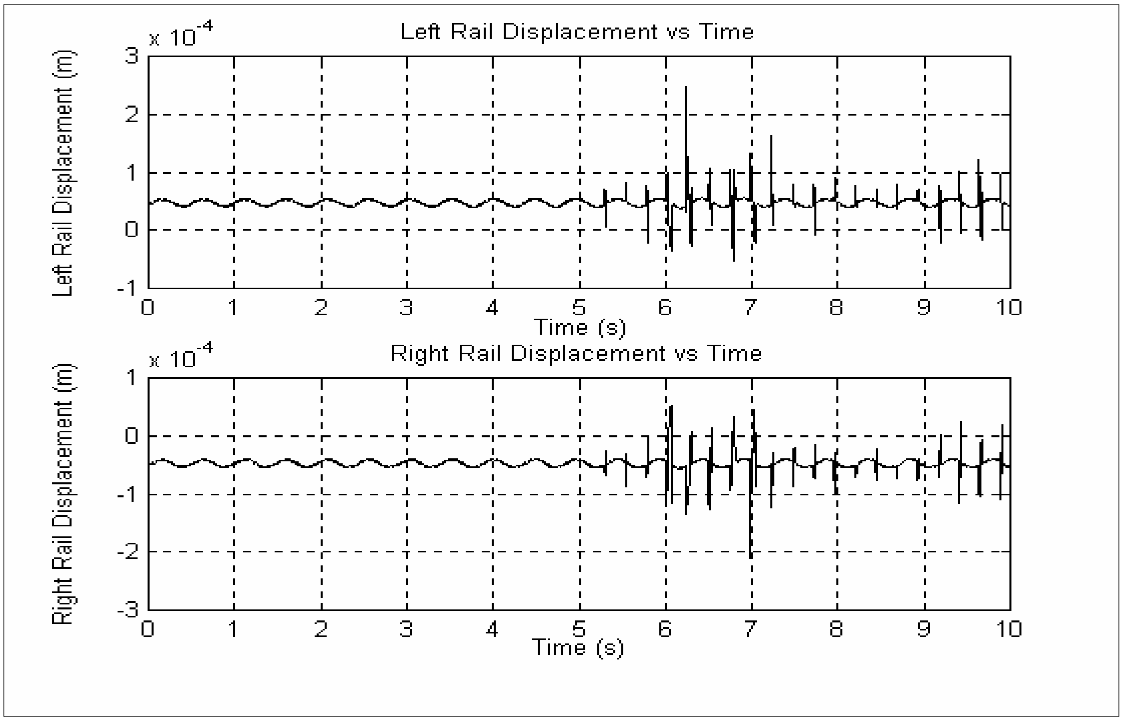


Figure 2-38 Rail Response: K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 41880 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (>V_C) = 20 m/s

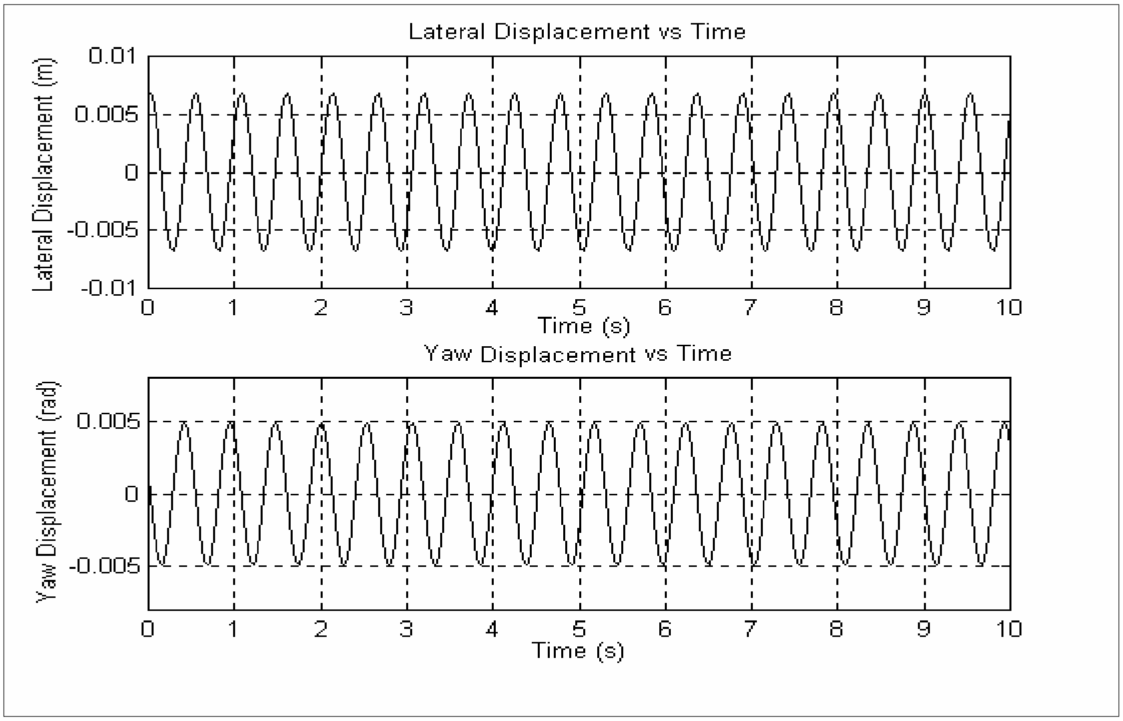


Figure 2-39 Wheelset Response: K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 1810 N-s/m, Velocity (V_C) = 18 m/s

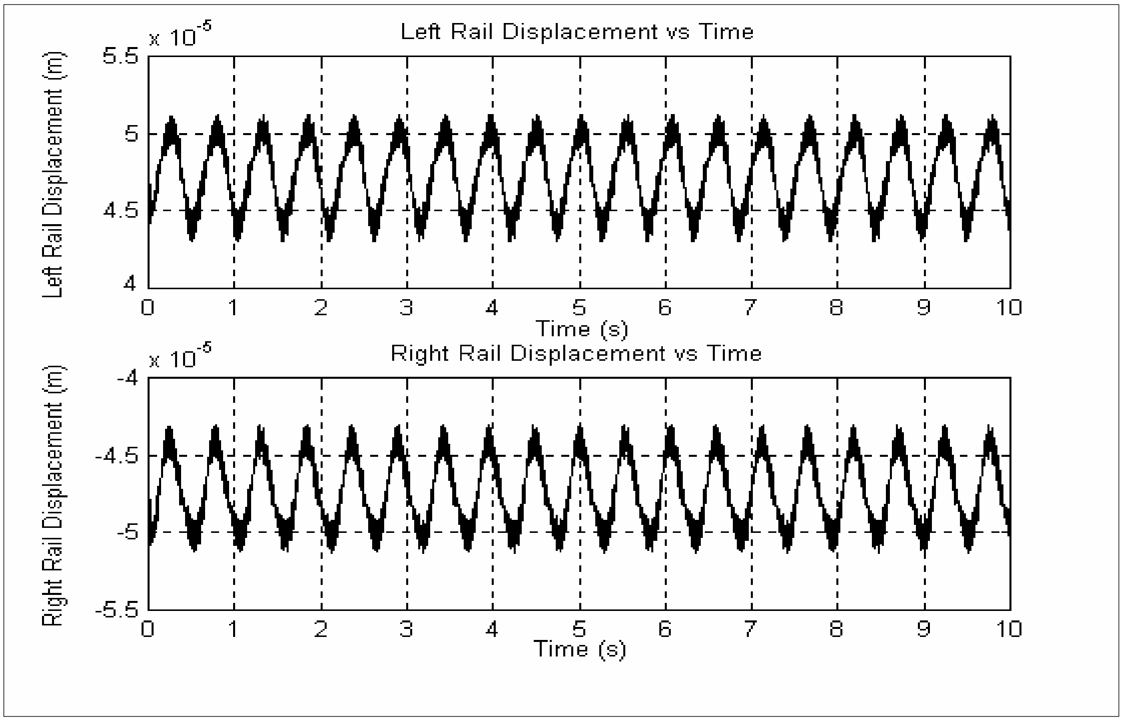


Figure 2-40 Rail Response: K_{PX} = 2.85e4 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 1810 N-s/m, Velocity (V_C) = 18 m/s

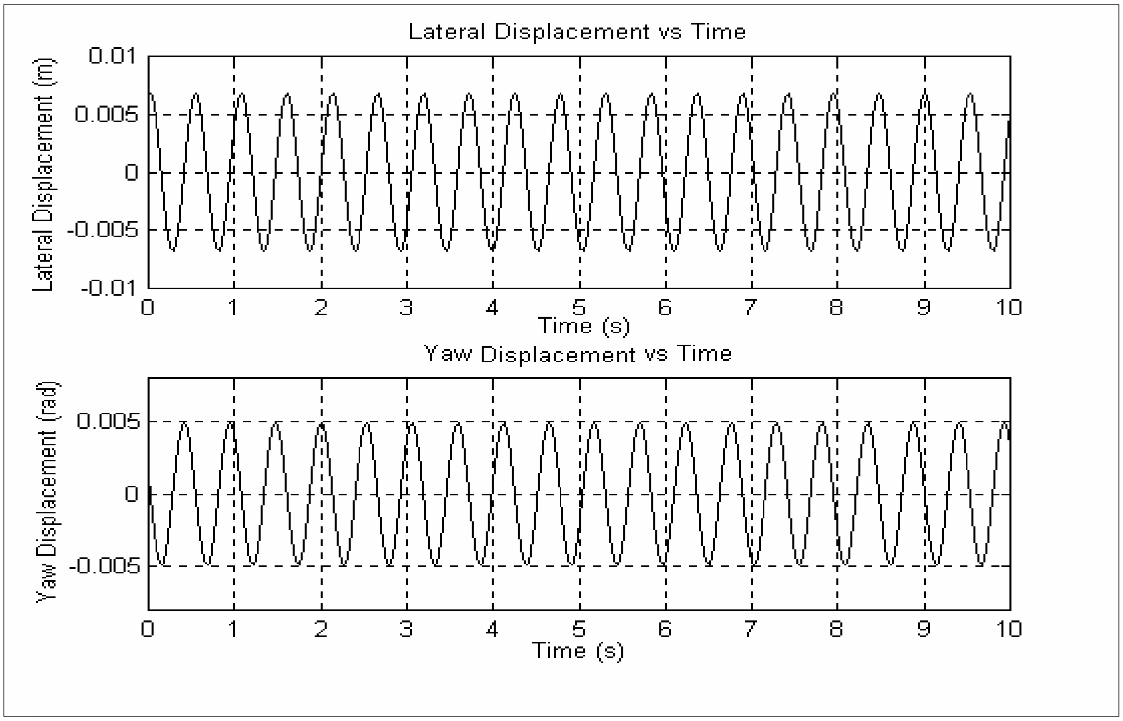


Figure 2-41 Wheelset Response: KPX = 2.85e4 N/m, KPY = 1.84e5 N/m, CPX = 8376.9 N-s/m, CPY = 1810 N-s/m, Velocity (>VC) = 20 m/s

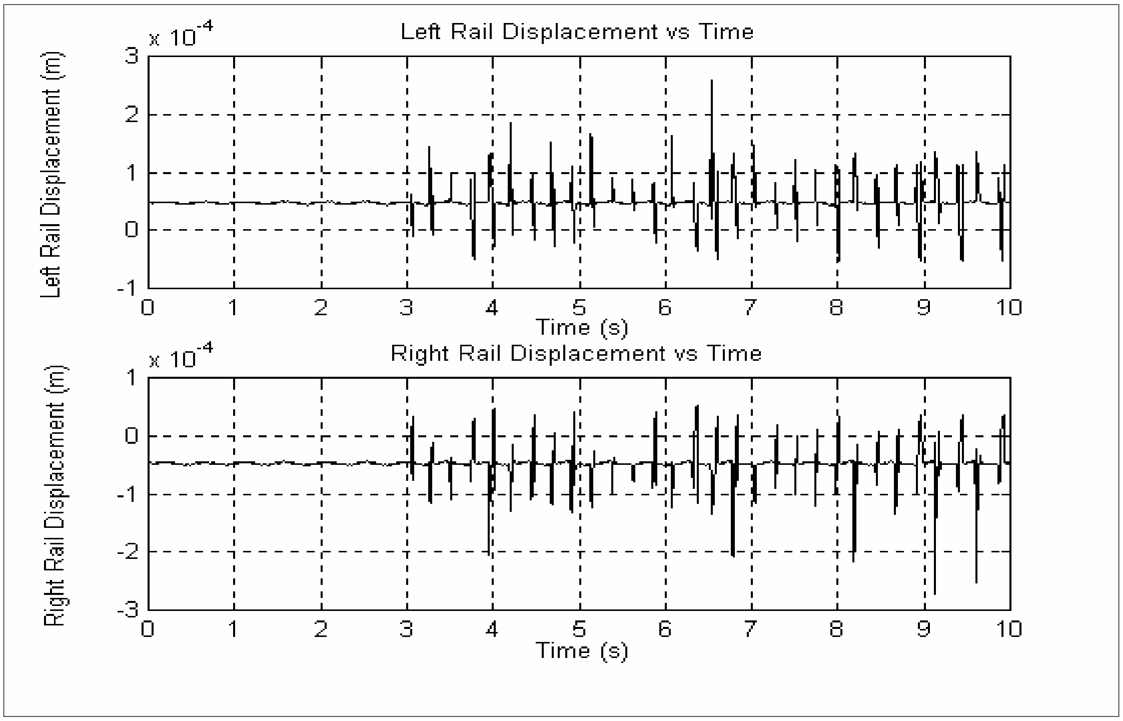


Figure 2-42 Rail Response: KPX = 2.85e4 N/m, KPY = 1.84e5 N/m, CPX = 8376.9 N-s/m, CPY = 1810 N-s/m, Velocity (>VC) = 20 m/s

Table 2-5 Sensitivity of Single Wheelset Critical Velocity to Primary Longitudinal Damping

K_{PX} (N/m)	K_{PY} (N/m)		
	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Wheelset (V_C), m/s (km/hr)		
C_{PX} = 1810 N-s/m			
2.85e4	18 (65)	30 (108)	54 (194)
9.12e4	20 (72)	32 (115)	55 (198)
2.85e5	25 (90)	35 (126)	57 (205)
9.12e5	37 (133)	45 (162)	63 (227)
2.85e6	63 (227)	68 (245)	81 (292)
C_{PX} = 4188 N-s/m			
2.85e4	18 (65)	30 (108)	55 (198)
9.12e4	20 (72)	32 (115)	55 (198)
2.85e5	25 (90)	35 (126)	57 (205)
9.12e5	37 (133)	45 (162)	63 (227)
2.85e6	64 (230)	68 (245)	81 (292)
C_{PX} = 8376.9 N-s/m			
2.85e4	18 (65)	31 (112)	56 (202)
9.12e4	20 (72)	32 (115)	56 (202)
2.85e5	25 (90)	35 (126)	58 (209)
9.12e5	37 (133)	45 (162)	64 (230)
2.85e6	64 (230)	69 (248)	82 (295)

C_{PX} = 25128 N-s/m			
2.85e4	18 (65)	32 (115)	59 (212)
9.12e4	20 (72)	33 (119)	59 (212)
2.85e5	25 (90)	36 (130)	61 (220)
9.12e5	38 (137)	46 (166)	68 (245)
2.85e6	67 (241)	72 (259)	87 (313)
C_{PX} = 41880 N-s/m			
2.85e4	18 (65)	33 (119)	63 (227)
9.12e4	20 (72)	34 (122)	63 (227)
2.85e5	26 (94)	37 (133)	65 (234)
9.12e5	39 (140)	48 (173)	72 (259)
2.85e6	69 (248)	75 (270)	93 (335)

Note: All values in Table 2-5 above have been obtained with C_{PY} = 9048.2 N-s/m.

Table 2-6 Sensitivity of Single Wheelset Critical Velocity to Primary Lateral Damping

K_{PX} (N/m)	K_{PY} (N/m)		
	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Wheelset (V_C), m/s (km/hr)		
C_{PY} = 1810 N-s/m			
2.85e4	18 (65)	30 (108)	55 (198)
9.12e4	20 (72)	31 (112)	55 (198)
2.85e5	25 (90)	35 (126)	57 (205)
9.12e5	36 (130)	44 (158)	63 (227)
2.85e6	61 (220)	66 (238)	79 (284)
C_{PY} = 4525 N-s/m			
2.85e4	18 (65)	30 (108)	55 (198)
9.12e4	20 (72)	32 (115)	56 (202)
2.85e5	25 (90)	35 (126)	57 (205)
9.12e5	37 (133)	44 (158)	63 (227)
2.85e6	62 (223)	67 (241)	80 (288)
C_{PY} = 9048.2 N-s/m			
2.85e4	18 (65)	31 (112)	56 (202)
9.12e4	20 (72)	32 (115)	56 (202)
2.85e5	25 (90)	35 (126)	58 (209)
9.12e5	37 (133)	45 (162)	64 (230)
2.85e6	64 (230)	69 (248)	82 (295)

C_{PY} = 27144 N-s/m			
2.85e4	19 (68)	32 (115)	58 (209)
9.12e4	21 (76)	33 (119)	59 (212)
2.85e5	26 (94)	37 (133)	61 (220)
9.12e5	40 (144)	48 (173)	69 (248)
C_{PY} = 45240 N-s/m			
2.85e4	19 (68)	33 (119)	62 (223)
9.12e4	21 (76)	34 (122)	63 (227)
2.85e5	27 (97)	39 (140)	66 (238)
9.12e5	43 (155)	52 (187)	77 (277)

Note: All values in Table 2-6 above have been obtained with C_{PX} = 8376.9 N-s/m. For C_{PY} = 27144 N-s/m and 45240 N-s/m, the critical velocities obtained are unrealistically high for K_{PX} = 2.85e6 N/m. Hence, these values have not been shown in the above table.

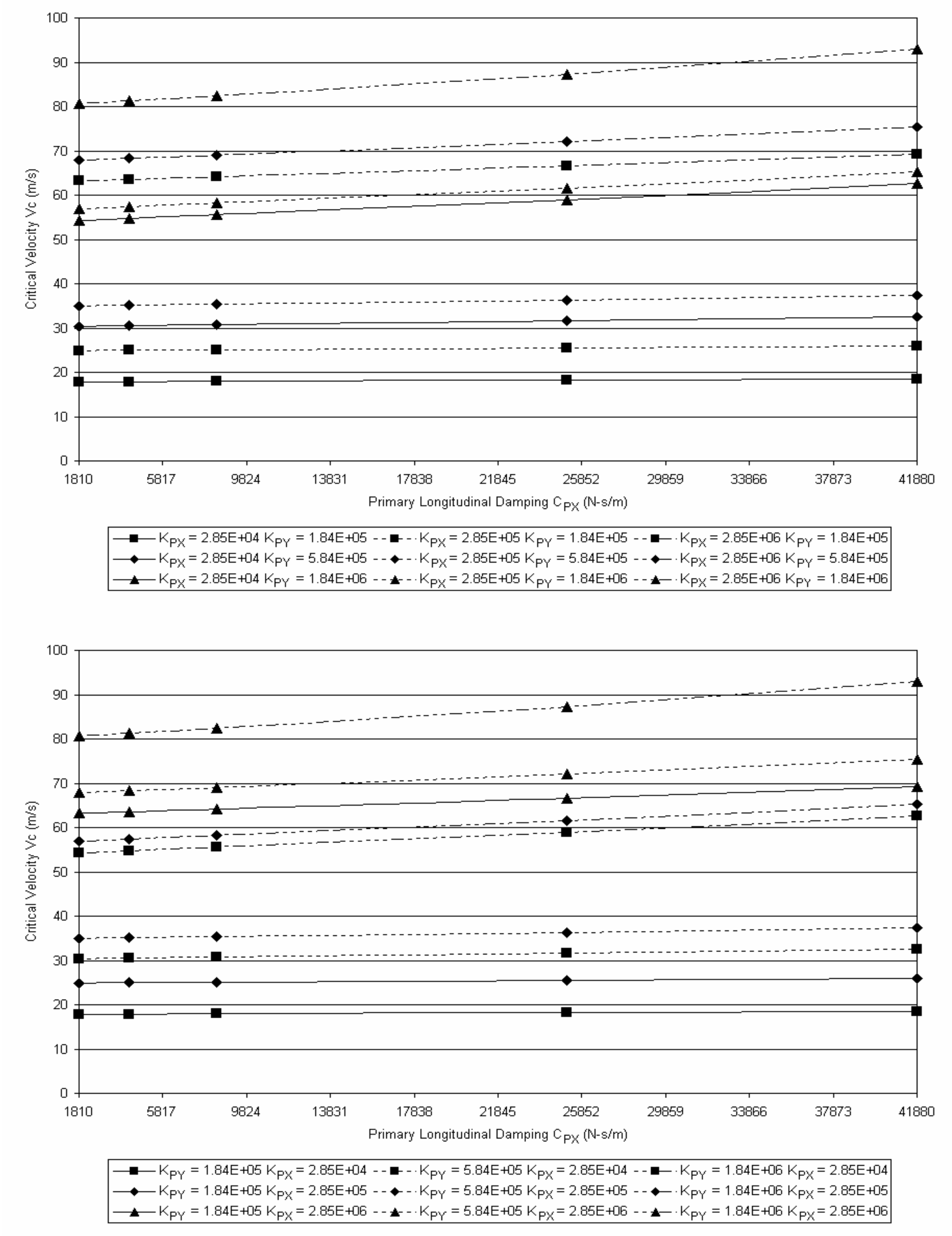


Figure 2-43 Variation of Single Wheelset Critical Velocity (V_C) with Primary Longitudinal Damping (C_{PX})

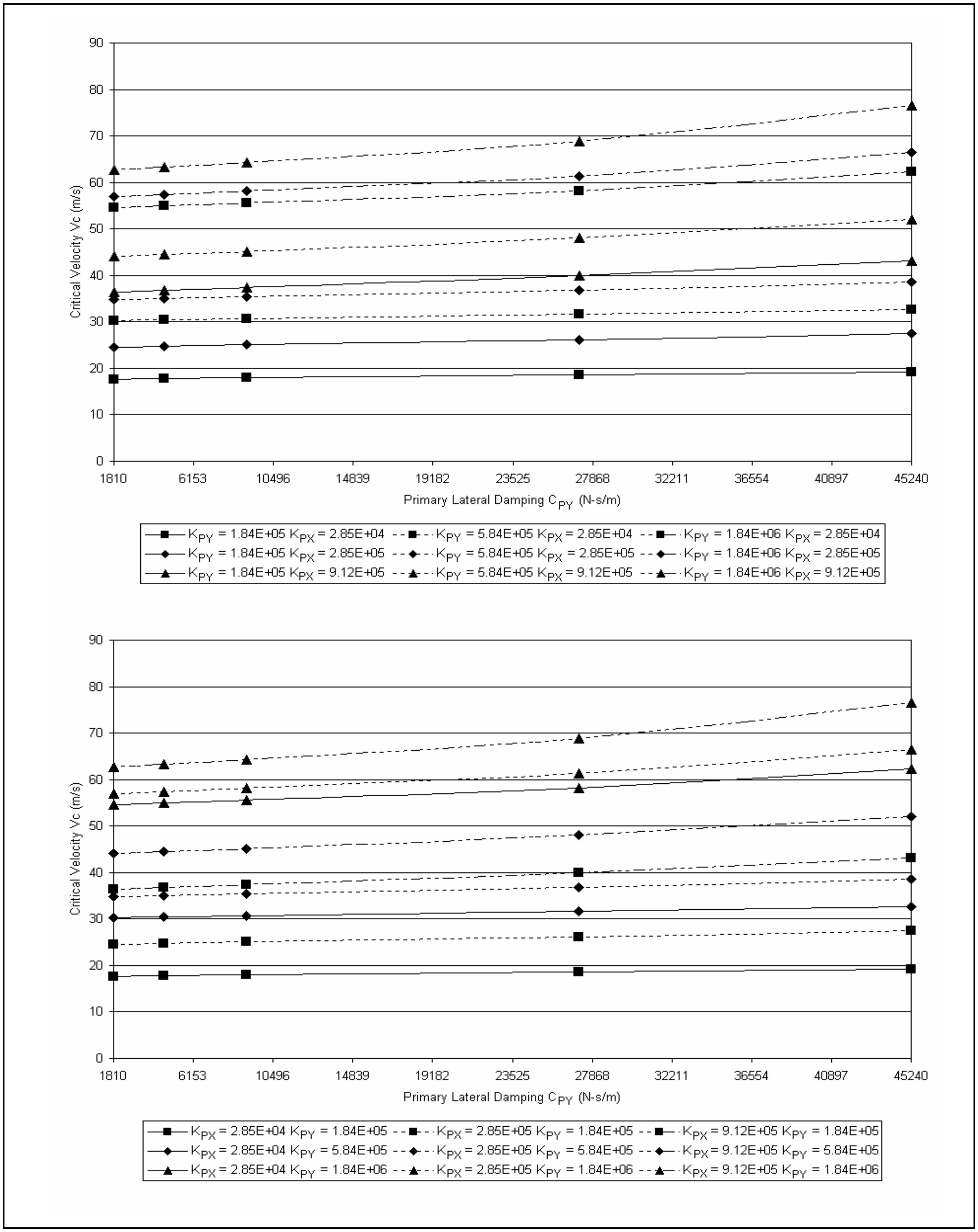


Figure 2-44 Variation of Single Wheelset Critical Velocity (V_C) with Primary Lateral Damping (C_{PY})

2.4 Conclusions

This chapter provided the mathematical formulation for a single wheelset model. The forces and moments which act on a single wheelset and which govern the lateral and yaw motions of the wheelset were obtained and the equations which define the dynamics of the wheelset were enumerated. Both single-point and two-point wheel / rail contact conditions were considered.

Mathematical simulations were first performed for a single wheelset with conical wheels. The effect of conicity on the critical velocity of the wheelset can be seen from Table 2-3 and Figure 2-27. It is seen that the critical velocity V_C of a single wheelset is proportional to the inverse of the square root of the conicity λ, i.e.

$$V_C \propto \frac{1}{\sqrt{\lambda}}$$

Figures 2-12 through 2-26 show the time response of the single wheelset before, at, and after the critical velocity has been reached for various values of λ ranging from 0.05 to 0.250.

A modified AAR wheel flange was then introduced in place of the conical profile in order to simulate two-point contact condition between the wheel and the rail. The effect of primary suspension parameters on the critical velocity of the single wheelset was presented.

Table 2-4 and Figures 2-33 and 2-34 depict the sensitivity of the single wheelset critical velocity to variations in the primary longitudinal and lateral spring stiffness. The figures indicate a supercritical Hopf bifurcation phenomenon, where the origin loses its stability after the critical velocity is reached, and gives rise to a nearly elliptical limit cycle about the origin. The results also show that the variation of critical velocity is more sensitive to the primary longitudinal spring stiffness K_{PX} for lower values of the primary lateral spring stiffness K_{PY}. At higher values of K_{PY} the variation of critical velocity with K_{PX} is practically linear. This conclusion also holds true when the critical velocity is varied with K_{PY} for different values of K_{PX}. However, higher longitudinal and lateral spring stiffness always result in higher critical velocities.

Tables 2-5 and 2-6 and Figures 2-43 and 2-44 depict the sensitivity of the single wheelset critical velocity to variations in the primary longitudinal and lateral damping. The results show that the variation of critical speed with both the primary longitudinal damping (C_{PX}) and the primary lateral damping (C_{PY}) is an almost linear relationship. Higher values of longitudinal and lateral damping result in higher critical velocities. Also, larger spring constants result in higher critical velocities as well as a higher sensitivity to primary damping.

This chapter concludes that the single wheelset critical velocity is more sensitive to primary spring stiffness than to primary damping. Hence, the critical velocity of the single wheelset can be effectively controlled by active or semi-active adjustment of the primary longitudinal and lateral spring stiffness.

Chapter 3

Single Truck Model

This chapter provides the mathematical formulation for a single truck model. The forces and moments which act on a single truck and which govern the lateral and yaw motions of the truck are obtained and the equations which define the dynamics of the truck are enumerated. The truck considered for modeling is a conventional truck that consists of two wheelsets connected to a truck frame. The two wheelsets are unconnected. Each of the wheelsets is similar to the single wheelset considered in Chapter 2. The single truck model is subjected to certain initial conditions. Both single-point and two-point wheel / rail contact conditions were considered. The mathematical equations governing the motion of a single truck [1] are simplified in order to reduce computation time without compromising on the accuracy of the solution. The simulation software MATLAB [2] was used to find time-domain solutions to the single truck dynamic equations. The critical velocities obtained by varying primary suspension parameters are presented.

3.1 Mathematical Formulation

A conventional truck consists of two unconnected wheelsets, each one similar to the single wheelset discussed in Chapter 2. The single truck arrangement with the primary and secondary suspension, and the bolster is shown in Figure 3-1. The two wheelsets are treated as identical, each one having two longitudinal and two lateral springs and dampers. The only coupling between the front and the rear wheesets is through a rigid truck frame.

Figure 3-2 depicts a conventional single truck connected to the front and the rear wheelsets through primary lateral and longitudinal springs.

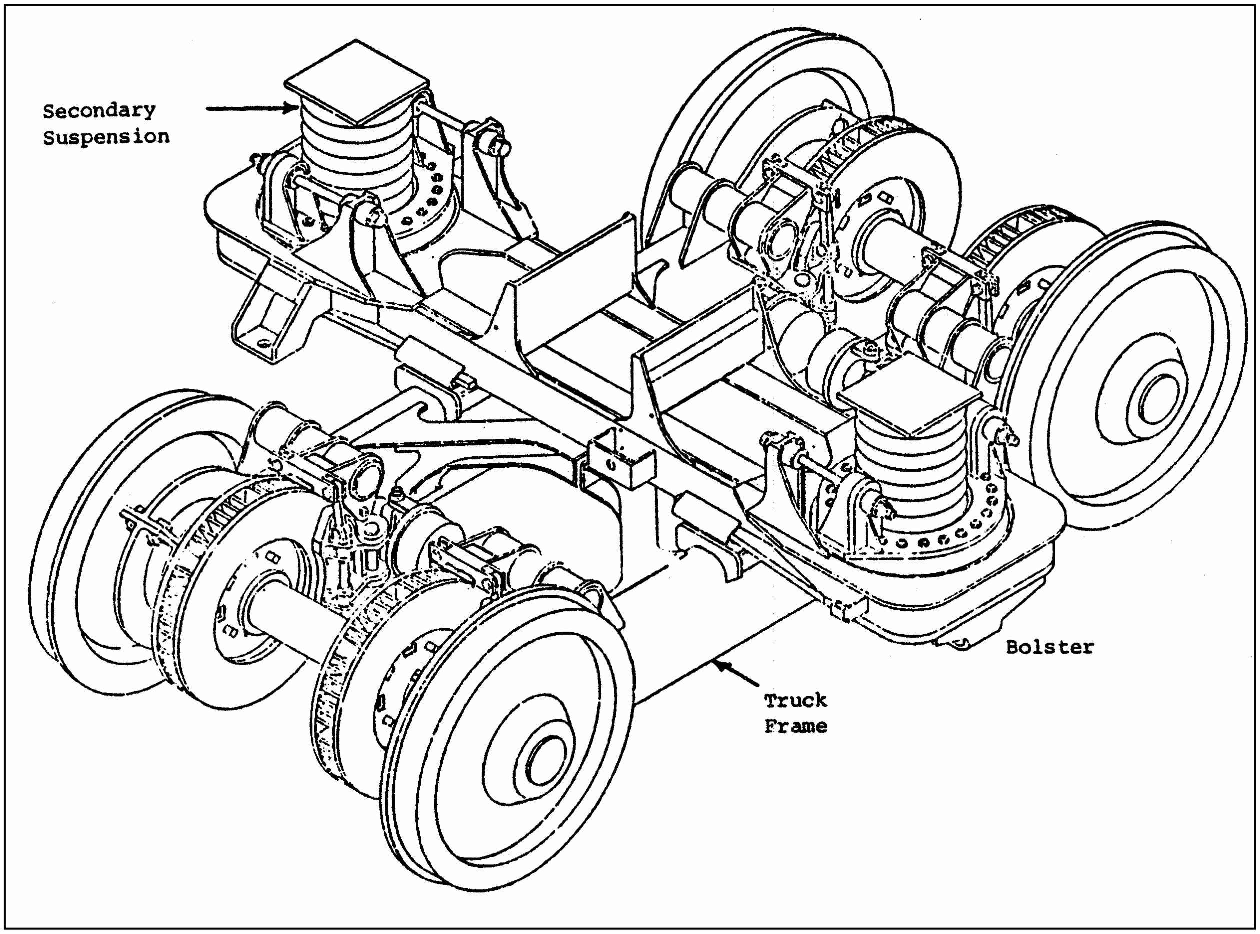


Figure 3-1 Single Truck and Bolster Arrangement

When the wheelsets move, they produce a resultant lateral force on the truck, which will tend to move the truck in the same direction as the movement of the wheelsets. Also, when a wheelset yaws, the longitudinal spring-damper combination will produce a truck yawing moment tending to rotate the truck in the same direction. The net force and moment on the truck will be the resultant of the two wheelset forces and moments.

The truck is connected to a bolster through a Coulomb friction arrangement. When the total yawing moment rotating the truck exceeds the moment produced by the Coulomb friction, the truck rotates relative to the bolster. Thus, the total restoring torque due to friction in the bolster cannot exceed the slipping value due to Coulomb friction. This Coulomb friction arrangement between the truck frame and each bolster is illustrated in Figure 3-3.

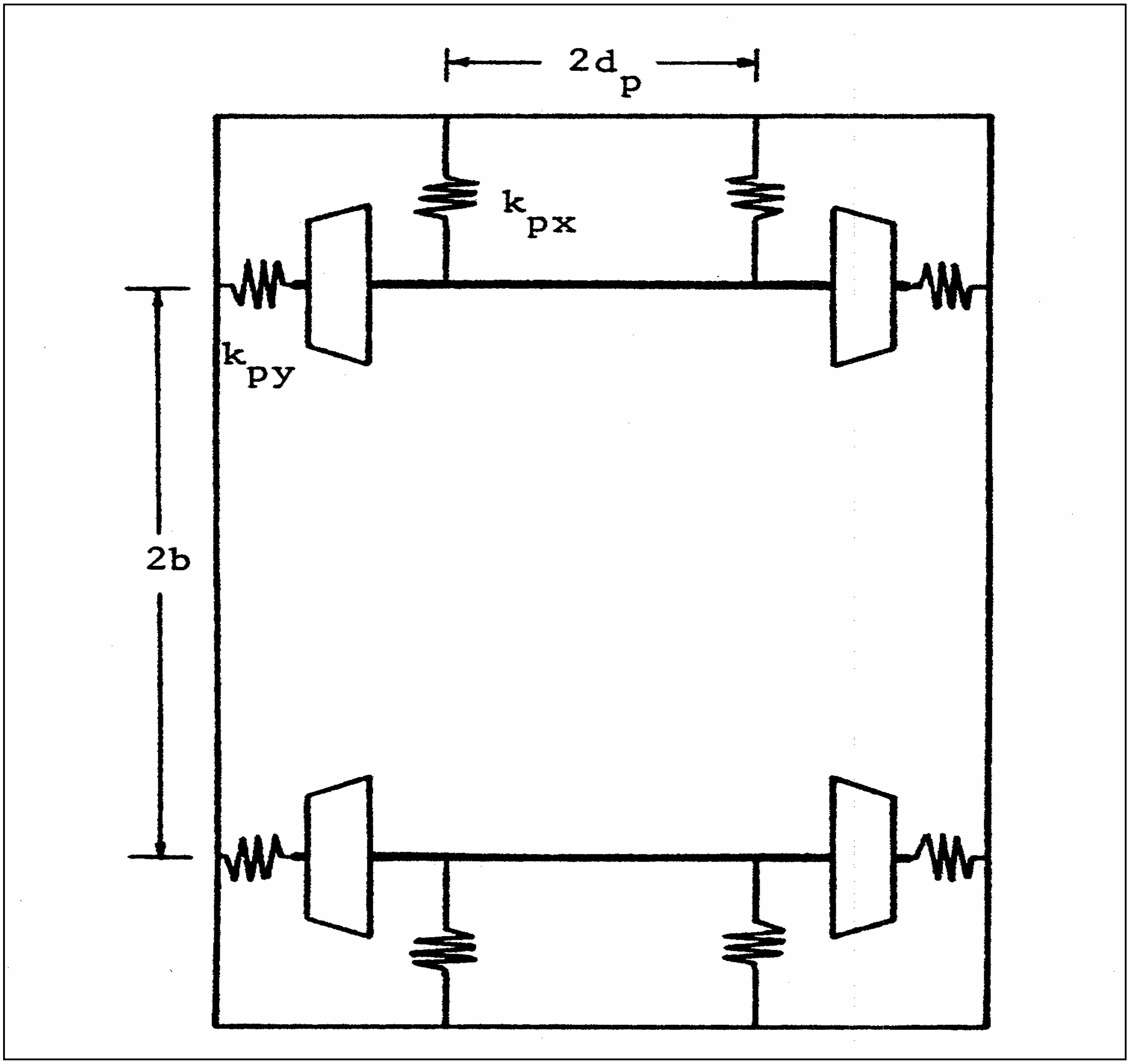


Figure 3-2 Schematic of a Conventional Single Truck

The bolster is connected to the carbody through a secondary suspension system that consists of a pair of lateral springs each having a stiffness K_{SY} and dampers each having a viscous damping C_{SY}. In addition, the carbody and the bolster are also coupled through a torsional spring / viscous damper combination in the yaw direction. The torsional spring has a stiffness $K_{S\psi}$ and a viscous damping $C_{S\psi}$. Since this chapter is concerned with the response of the truck, the simulations that follow in this chapter assume the carbody to be fixed.

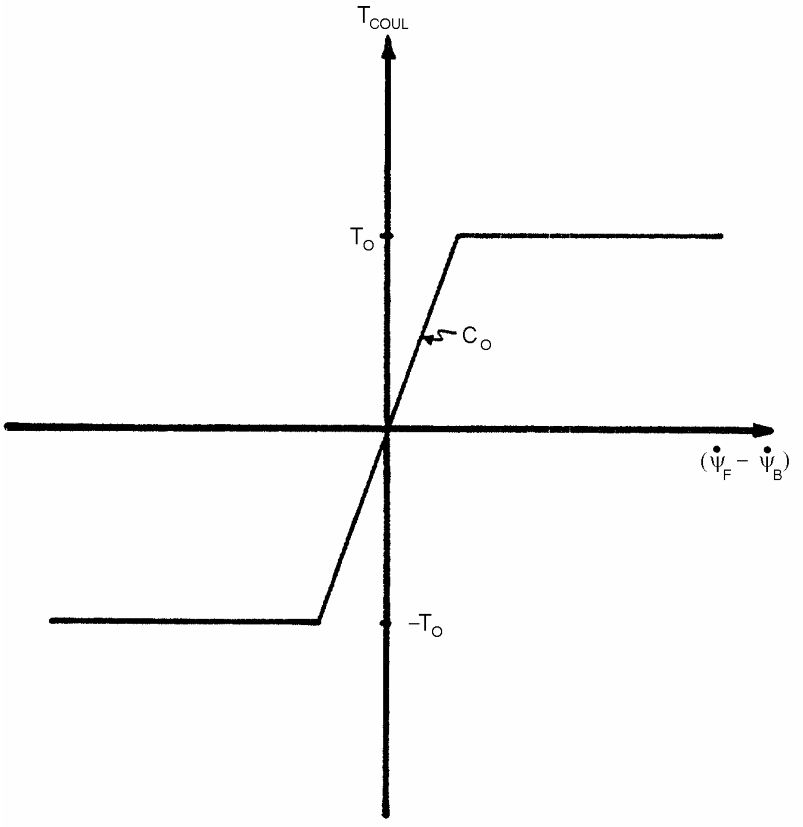


Figure 3-3 Truck Frame / Bolster Coulomb Friction Characteristic

The Coulomb-viscous characteristic shown above simulates two friction plates (truckframe and bolster) with hydraulic fluid in between. The secondary springs are typically considerably softer than the primary springs in order to prevent the truck and bolster motions from passing on to the carbody. These lower spring constants cause considerable movement of the truck frame, thereby reducing the primary spring forces. Hence, high values of primary spring constants will not show the same effectiveness as in the case of a single wheelset.

3.1.1 Wheelset Suspension Forces and Moments

Suspension forces and moments act on each wheelset due to primary longitudinal and lateral suspension elements. Since this analysis neglects the wheelset vertical and roll degrees of freedom, the vertical suspension force and the roll suspension moment are assumed to be zero.

The lateral suspension force and the yaw suspension moment acting on the leading wheelset of the truck are given by the expressions below.

$$F_{SUSPYW1} = -2K_{PY}y_{W1} + 2K_{PY}y_F + 2bK_{PY}\psi_F - 2C_{PY}\dot{y}_{W1} + 2C_{PY}\dot{y}_F + 2bC_{PY}\dot{\psi}_F \tag{3.1}$$

$$M_{SUSPZW1} = -2d_p^2K_{PX}\psi_{W1} + 2d_p^2K_{PX}\psi_F - 2d_p^2C_{PX}\dot{\psi}_{W1} + 2d_p^2C_{PX}\dot{\psi}_F \tag{3.2}$$

The lateral suspension force and the yaw suspension moment acting on the trailing wheelset (i = 2) of the truck are given by

$$F_{SUSPYW2} = -2K_{PY}y_{W2} + 2K_{PY}y_F - 2bK_{PY}\psi_F - 2C_{PY}\dot{y}_{W2} + 2C_{PY}\dot{y}_F - 2bC_{PY}\dot{\psi}_F \tag{3.3}$$

$$M_{SUSPZW2} = -2d_p^2K_{PX}\psi_{W2} + 2d_p^2K_{PX}\psi_F - 2d_p^2C_{PX}\dot{\psi}_{W2} + 2d_p^2C_{PX}\dot{\psi}_F \tag{3.4}$$

3.1.2 Truck Frame and Bolster Suspension Forces and Moments

Suspension forces and moments act on the truck frame and the bolster due to the primary longitudinal and lateral suspension elements, as also the torsional Coulomb damper between the truck frame and the bolster. The damper only allows yaw motion between the truck frame and the bolster. Figure 3-4 shows the secondary yaw suspension model.

The lateral suspension force and the yaw suspension moment acting on the truck are given by the expressions below.

$$F_{SUSPYF} = 2K_{PY}y_{W1} + 2K_{PY}y_{W2} - 4K_{PY}y_F - 2K_{SY}y_F + 2C_{PY}\dot{y}_{W1} + 2C_{PY}\dot{y}_{W2} - 4C_{PY}\dot{y}_F - 2C_{SY}\dot{y}_F \tag{3.5}$$

$$\begin{aligned} M_{SUSPZF} = {} & 2bK_{PY}y_{W1} + 2d_p^2K_{PX}\psi_{W1} - 2bK_{PY}y_{W2} + 2d_p^2K_{PX}\psi_{W2} + (-4d_p^2K_{PX} - 4b^2K_{PY})\psi_F \\ & + 2bC_{PY}\dot{y}_{W1} + 2d_p^2C_{PX}\dot{\psi}_{W1} - 2bC_{PY}\dot{y}_{W2} + 2d_p^2C_{PX}\dot{\psi}_{W2} + (-4d_p^2C_{PX} - 4b^2C_{PY})\dot{\psi}_F - T_{COUL} \end{aligned} \tag{3.6}$$

The yaw suspension moment acting on the bolster is given by:

$$M_{SUSPZB} = -K_{S\psi}\psi_B - C_{S\psi}\dot{\psi}_B + T_{COUL} \tag{3.7}$$

In equations (3.6) and (3.7), T_{COUL} represents the Coulomb friction yaw moment acting on the truck frame due to interaction with the bolster. For numerical purposes, the model of the Coulomb friction is modified to include a linear viscous band at the origin, as shown in equation

(3.8). At low relative yaw rates between the bolster and the truck frame, the model assumes viscous damping. At higher relative yaw rates, the model assumes Coulomb damping with the frictional torque saturating at the centerplate breakaway value. This method approximates the frictional torque levels below $|T_O|$.

$$T_{COUL} = \begin{cases} T_O \;\ldots\ldots\ldots\ldots\ \text{for } (\dot{\psi}_F - \dot{\psi}_B) \geq T_O / C_O \\ C_O(\dot{\psi}_F - \dot{\psi}_B) \;\ldots\ldots\ \text{for } -T_O / C_O < (\dot{\psi}_F - \dot{\psi}_B) < T_O / C_O \\ -T_O \;\ldots\ldots\ldots\ldots\ \text{for } (\dot{\psi}_F - \dot{\psi}_B) \leq -T_O / C_O \end{cases} \tag{3.8}$$

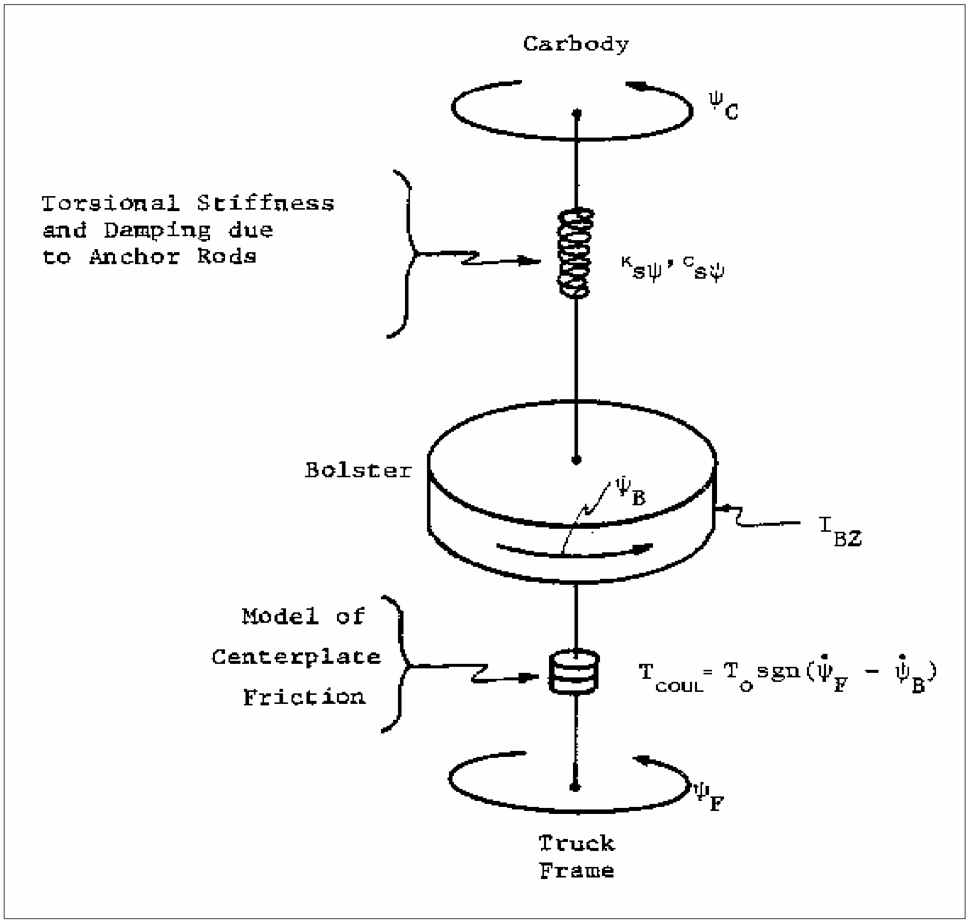

Carbody
ψ_C
Torsional Stiffness and Damping due to Anchor Rods
$K_{s\psi}, C_{s\psi}$
Bolster
ψ_B
I_{BZ}
Model of Centerplate Friction
$T_{COUL} = T_O \text{sgn}(\dot{\psi}_F - \dot{\psi}_B)$
ψ_F
Truck Frame

Figure 3-4 Secondary Yaw Suspension Arrangement

3.1.3 Truck Frame and Bolster Dynamic Equations

This section presents the dynamic equations motion for the truck frame and the bolster. The equations of motion for a wheelset were discussed in Chapter 2 under Sections 2.1.3.3 and 2.1.3.4. The truck frame and the bolster share the same lateral degree of freedom. The lateral equation is obtained by applying the principle of linear momentum in the lateral direction [1]. Taking into account the lateral components of truck frame and bolster weight and assuming small angles, the lateral equation of motion can be given as:

$$(m_F + m_B)\ddot{y}_F = -\frac{1}{2}(m_F + m_B)(\phi_{W1} + \phi_{W2})g + F_{SUSPYF} \tag{3.9}$$

where, ϕ_{W1} and ϕ_{W2} are the roll angles of the leading and the trailing wheelsets respectively.

The truck frame and the bolster have separate yaw degrees of freedom due to the torsional Coulomb element between the two components. The truck frame yaw equation is obtained by applying the principle of angular momentum in the yaw direction [1]. Assuming small angles, the truck frame yaw equation of motion can be given as:

$$I_{FZ}\ddot{\psi}_F = M_{SUSPZF} \tag{3.10}$$

Similarly, the bolster yaw equation of motion is:

$$I_{BZ}\ddot{\psi}_B = M_{SUSPZB} \tag{3.11}$$

3.2 Numerical Simulation

The single truck model presented in Section 3.1 was simulated using MATLAB [2] in order to obtain the time-domain solution of the dynamics of a single truck moving on a flexible tangent track. This section presents the layout of the simulation program that was used to obtain the truck dynamic response.

The simulations were carried out by choosing the forward speed of the truck as the bifurcation parameter. The critical forward speed was obtained by increasing the forward speed gradually until the response of the truck became marginally stable. Sensitivity of the critical velocity to suspension parameters was studied.

Simplifications were made in order to make the memory requirements less and speed up the computation. In a wheelset dynamic analysis, the lateral dynamics are very important since they determine whether or not flanging occurs. The lateral dynamics are essentially decoupled from the vertical and the longitudinal dynamics. Hence, this simulation neglects the vertical and the longitudinal dynamics of the wheelset. This assumption eliminates two degrees of freedom for each wheelset and greatly reduces computation time.

It is also assumed that the effective lateral mass of the rail, m_{RAIL}, is zero. This is justified since the rail lateral stiffness and viscous damping forces dominate. Further, it is assumed that the influence of lateral rail velocity on lateral creepage is negligible. According to British Rail [5], this assumption is reasonable since the lateral creep force is generally saturated during flange contact.

The maximum adhesion force was assumed as constant rather than calculating the creep force saturation value at each time step iteratively. This iterative process involves solving the creep and the normal force equations simultaneously, which tends to be computationally very intense. From past experience, values of creep forces were found to be considerably less than the adhesion limit.

The creep coefficients were taken to be the same for both the tread and the flange contact patches. In reality, the flange contact patch will have smaller values, but it was found that this assumption makes insignificant difference to the resulting value of the critical forward speed and the wheelset response.

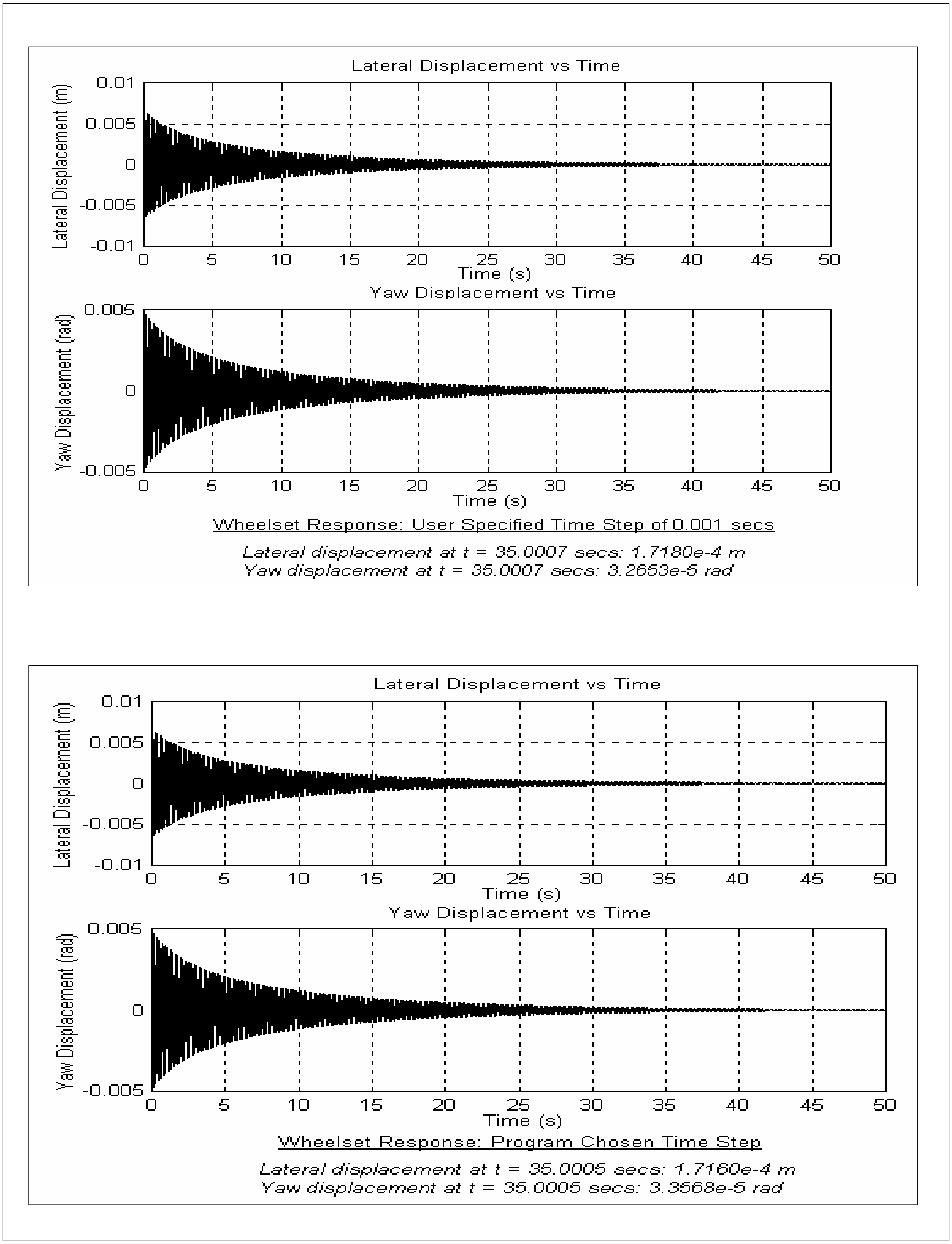
Initial conditions were assumed for the lateral and yaw position and velocity of the wheelsets and the truck. The initial lateral displacement and velocity of the left and the right rails were assumed to be zero. The initial conditions assumed for simulation can be found in the MATLAB program files that are included in the Appendix.

A time step of 0.001 secs was manually chosen for solving the dynamic truck equations in MATLAB. The time step chosen automatically by MATLAB resulted in lack of memory space for the desired simulation time. Figure 3-5 shows the wheelset response for the program chosen time step, as well as a manually chosen time step of 0.001 secs. It is seen that the two solutions are nearly identical.

The following time-domain solutions were obtained through simulation:

4. The lateral displacement and velocity of the wheelset
5. The yaw displacement and velocity of the wheelset
6. The lateral displacement of the left and the right rails
7. The lateral displacement and velocity of the truck
8. The yaw displacement and velocity of the truck and bolster

The parametric values used for simulation are shown in Table 3-1.



Wheelset Response: User Specified Time Step of 0.001 secs

Lateral displacement at t = 35.0007 secs: 1.7180e-4 m
Yaw displacement at t = 35.0007 secs: 3.2653e-5 rad

Wheelset Response: Program Chosen Time Step

Lateral displacement at t = 35.0005 secs: 1.7160e-4 m
Yaw displacement at t = 35.0005 secs: 3.3568e-5 rad

Figure 3-5 Wheelset Response Comparison - Automatic and Manual Time Step

Table 3-1 Single Truck Simulation Constants

Parameter	**Description**	**Value**
Wheel Type	Wheel Type	AAR 1 in 20
Truck Type	Truck Type	Conventional
	Wheel / Rail Constants	
f_{11T}	Creep Coefficient (Tread)	9.43e6 N
f_{12T}	Creep Coefficient (Tread)	1.20e3 N.m
f_{22T}	Creep Coefficient (Tread)	1.00e3 N.m^2
f_{33T}	Creep Coefficient (Tread)	10.23e6 N
f_{11F}	Creep Coefficient (Flange)	9.43e6 N
f_{12F}	Creep Coefficient (Flange)	1.20e3 N.m
f_{22F}	Creep Coefficient (Flange)	1.00e3 N.m^2
f_{33F}	Creep Coefficient (Flange)	10.23e6 N
λ	Conicity	0.125
μ	Coefficient of friction	0.15
	Geometric Dimensions	
R_0	Centered rolling radius of wheel	0.3556 m
a	Half of track gage	0.716 m
b	Half of wheel base	1.295 m
d_p	Half distance between primary longitudinal springs	0.61 m
	Weights and Moments of Inertia	
m_W	Mass of wheelset	1751 kg
I_{WY}	Pitch mass moment of inertia of wheelset	130 kg.m^2
I_{WZ}	Yaw mass moment of inertia of wheelset	761 kg.m^2
m_F	Mass of truck frame	4041 kg

Parameter	Description	Value
m_B	Mass of bolster	365 kg
I_{FZ}	Yaw mass moment of inertia of truck	3371 kg.m^2
I_{BZ}	Yaw mass moment of inertia of bolster	337 kg.m^2
Suspension Stiffness and Damping		
K_{PX}	Primary longitudinal stiffness	Various
C_{PX}	Primary longitudinal damping	Various
K_{PY}	Primary lateral stiffness	Various
C_{PY}	Primary lateral damping	Various
K_{SY}	Secondary lateral stiffness	3.5E+05 N/m
C_{SY}	Secondary lateral damping	1.75E+04 N.s/m
$K_{S\psi}$	Secondary torsional stiffness	3.8E+08 N.m/rad
$C_{S\psi}$	Secondary torsional damping	2.5E+07 N.m.s/rad
T_0	Centerplate breakaway torque	10168 N.m
C_0	Coulomb viscous yaw damping	3.5E+07 N.s/rad

The general algorithm used for wheelset analysis was described in Section 2.2 of Chapter 2. The single-point or two-point contact equations are solved at each time step depending on which out of the above conditions is satisfied. The single-point and two-point contact equations are solved by MATLAB using a fourth order Runge-Kutta integration algorithm. This method requires the equations of motion to be transformed to a system of 16 first-order differential equations (also known as state-space equations). In order to achieve this transformation, the actual variables have been converted to first-order state-space variables as shown below:

$x_1 = y_{W1}$

$x_2 = \psi_{W1}$

$x_3 = \dot{y}_{W1}$

$x_4 = \dot{\psi}_{W1}$

$x_5 = y_{RAIL,L1}$

$x_6 = y_{RAIL,R1}$

$x_7 = y_{W2}$

$x_8 = \psi_{W2}$

$x_9 = \dot{y}_{W2}$

$x_{10} = \dot{\psi}_{W2}$

$x_{11} = y_{RAIL,L2}$

$x_{12} = y_{RAIL,R2}$

$x_{13} = y_F$

$x_{14} = \psi_F$

$x_{15} = \dot{y}_F$

$x_{16} = \dot{\psi}_F$

$x_{17} = \psi_B$

$x_{18} = \dot{\psi}_B$

(3.12)

The equations of motion can now be written in state-space form as follows:

$\dot{x}_1 = x_3$ (from above)

$\dot{x}_2 = x_4$ (from above)

$\dot{x}_3 = f_1(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (front wheelset lateral equation)

$\dot{x}_4 = f_2(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (front wheelset yaw equation)

$\dot{x}_5 = f_3(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (front left rail lateral equation)

$\dot{x}_6 = f_4(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (front right rail lateral equation)

$\dot{x}_7 = x_9$ (from above)

$\dot{x}_8 = x_{10}$ (from above)

$\dot{x}_9 = f_5(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (rear wheelset lateral equation)

$\dot{x}_{10} = f_6(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (rear wheelset yaw equation)

$\dot{x}_{11} = f_7(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (rear left rail lateral equation)

$\dot{x}_{12} = f_8(x_1, x_2, x_3, x_4, x_5, x_6, x_{13}, x_{14}, x_{15}, x_{16})$ (rear right rail lateral equation)

$\dot{x}_{13} = x_{15}$ (from above)

$\dot{x}_{14} = x_{16}$ (from above)

$\dot{x}_{15} = f_9(x_1, x_2, x_3, x_4, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{16})$ (truck lateral equation)

$\dot{x}_{16} = f_{10}(x_1, x_2, x_3, x_4, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18})$ (truck yaw equation)

$\dot{x}_{17} = x_{18}$ (from above)

$\dot{x}_{18} = f_{11}(x_1, x_2, x_3, x_4, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18})$ (bolster yaw equation)

(3.13)

The MATLAB program used to simulate the dynamic behavior of a single truck and the functions used by the program are described below in Table 3-2. Figure 3-6 depicts the layout and interaction between the different functions at any time-step.

Table 3-2 Single Truck Simulation Program and Functions

Name of Program / Function	Description
SINGLE_TRUCK	Main program. This contains the initial conditions, the global variables, and instructions for plotting the time-responses. This program also contains the instruction and conditions to solve the ordinary differential equations contained in the function files WHEELSET and TRUCK
EQUATIONS	This function obtains variables from functions WHEELSET_SUSPENSION and TRUCK_SUSPENSION and uses them to solve the single wheelset and truck equations by invoking functions WHEELSET and TRUCK
WHEELSET	Contains the single wheelset differential equations for the front and the rear wheelsets. This function reads the rolling radius and contact angle from functions ROLLING_RADIUS and CONTACT_ANGLE respectively, and the wheelset suspension forces and moments from EQUATIONS. This function also reads the creep forces from ONEPT_CREEP and TWOPT_CREEP (depending on single or two-point contact) and the normal forces from ONEPT_NORMAL or TWOPT_NORMAL (depending on single or two-point contact)
TRUCK	Contains the single truck differential equations. This function reads the truck suspension forces and moments from EQUATIONS.
ONEPT_CREEP	Reads the wheelset state variables at any time-step and calculates the single-point creep forces and moments using the Vermeulen-Johnson approach with creep force saturation. The output is then passed on to the function WHEELSET

Name of Program / Function	Description
TWOPT_CREEP	Reads the wheelset state variables at any time-step and calculates the two-point creep forces and moments using the Vermeulen-Johnson approach with creep force saturation. The output is then passed on to the function WHEELSET.
ONEPT_NORMAL	Reads the wheelset state variables at any time-step and calculates the single-point normal forces and moments. The output is then passed on to the function WHEELSET
TWOPT_NORMAL	Reads the wheelset state variables at any time-step and calculates the two-point normal forces and moments. The output is then passed on to the function WHEELSET
WHEELSET_ SUSPENSION	Reads the wheelset and truck state variables at any time-step and calculates the suspension forces and moments acting on the wheelset. The output is then passed on to the function EQUATIONS
TRUCK_ SUSPENSION	Reads the wheelset and truck state variables at any time-step and calculates the suspension forces and moments acting on the truck frame. The output is then passed on to the function EQUATIONS
ROLLING_RADIUS	Reads the wheelset and rail lateral excursion at any time-step and calculates the rolling radius at tread and/or flange contact patches on the left and the right wheels. The output is then passed on to the function WHEELSET.
CONTACT_ANGLE	Reads the wheelset and rail lateral excursion at any time-step and calculates the contact angle at tread and/or flange contact patches on the left and the right wheels. The output is then passed on to the function WHEELSET.

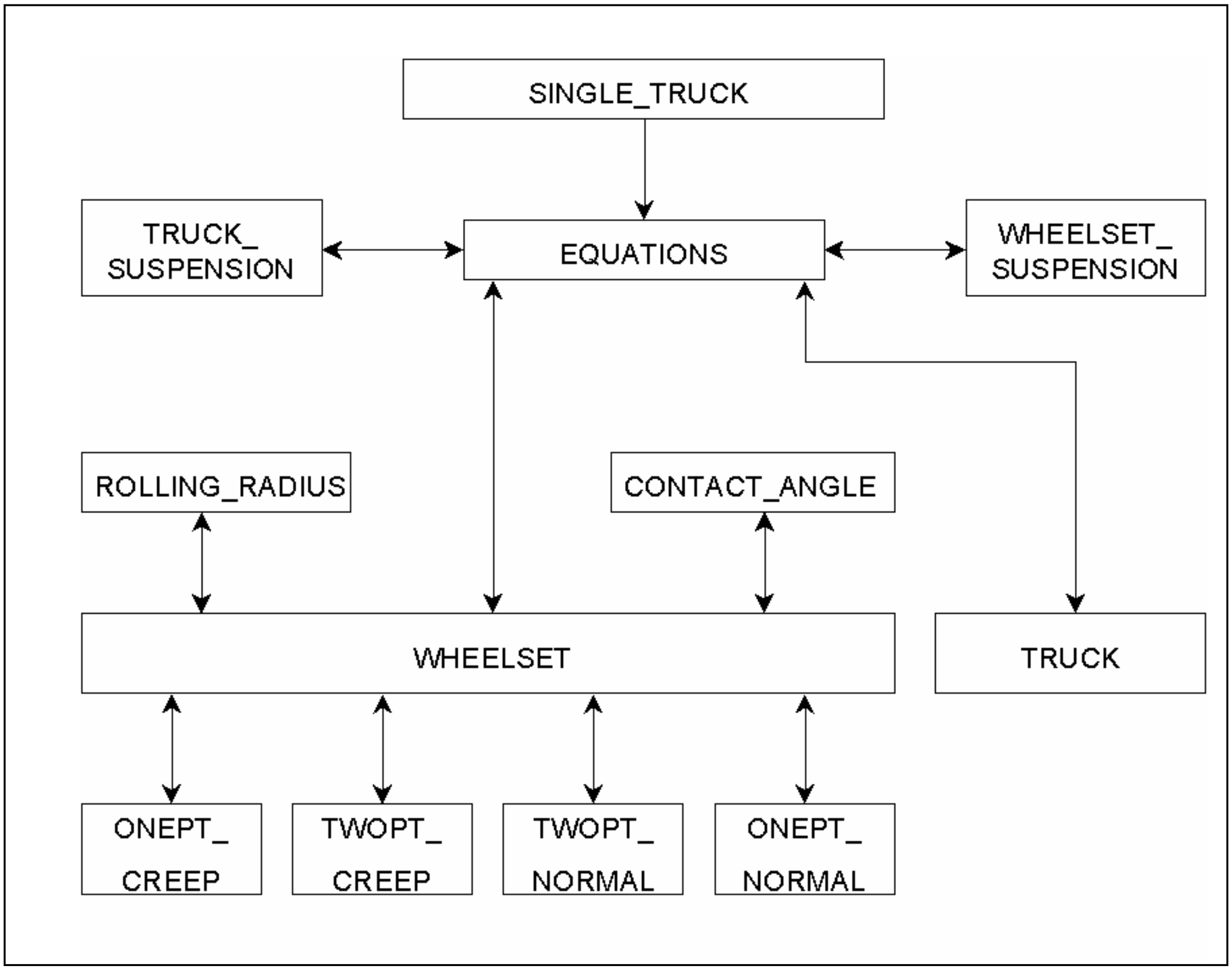


Figure 3-6 Single Truck Simulation Program Layout

3.3 Simulation Results

The simulation essentially involved the determination of the critical velocity under different constraints imposed on the equations enumerated in Section 3.1. The data enumerated in Table 3-1 were used except when the effect of variation of a given parameter was being investigated.

In order to study the response of the truck, the carbody was assumed to be fixed. Since the truck modeled for the simulation is a conventional truck, the front and the rear wheelsets have independent degrees of freedom. Hence, the initial conditions of the two wheelsets can have a wide range of combinations. The two wheelsets can be initially at the same position at one of the rails, or they can be at the opposite rails, or anywhere in between. The various combinations for the front and the rear wheelsets is shown in Figure 3-7.

3.3.1 The Effect of Primary Spring Stiffness on the Critical Velocity

The critical velocity of a single truck with the AAR wheels was determined for various values of primary spring longitudinal and lateral stiffness by varying the vehicle forward velocity.

The initial conditions used for the simulations in this section are:

Lateral displacement of front wheelset = 0.00635 m,
Lateral velocity of front wheelset = 0.01 m/s,
Yaw displacement of front wheelset = 0.001 rad,
Yaw velocity of front wheelset = 0.0 rad/s,
Lateral displacement of rear wheelset = 0.003 m,
Lateral velocity of rear wheelset = 0.01 m/s,
Yaw displacement of rear wheelset = 0.001 rad,
Yaw velocity of rear wheelset = 0.0 rad/s,
Lateral displacement of truck frame = 0.0 m,
Lateral velocity of truck frame = 0.0 m/s,
Yaw displacement of truck frame = 0.0 rad,

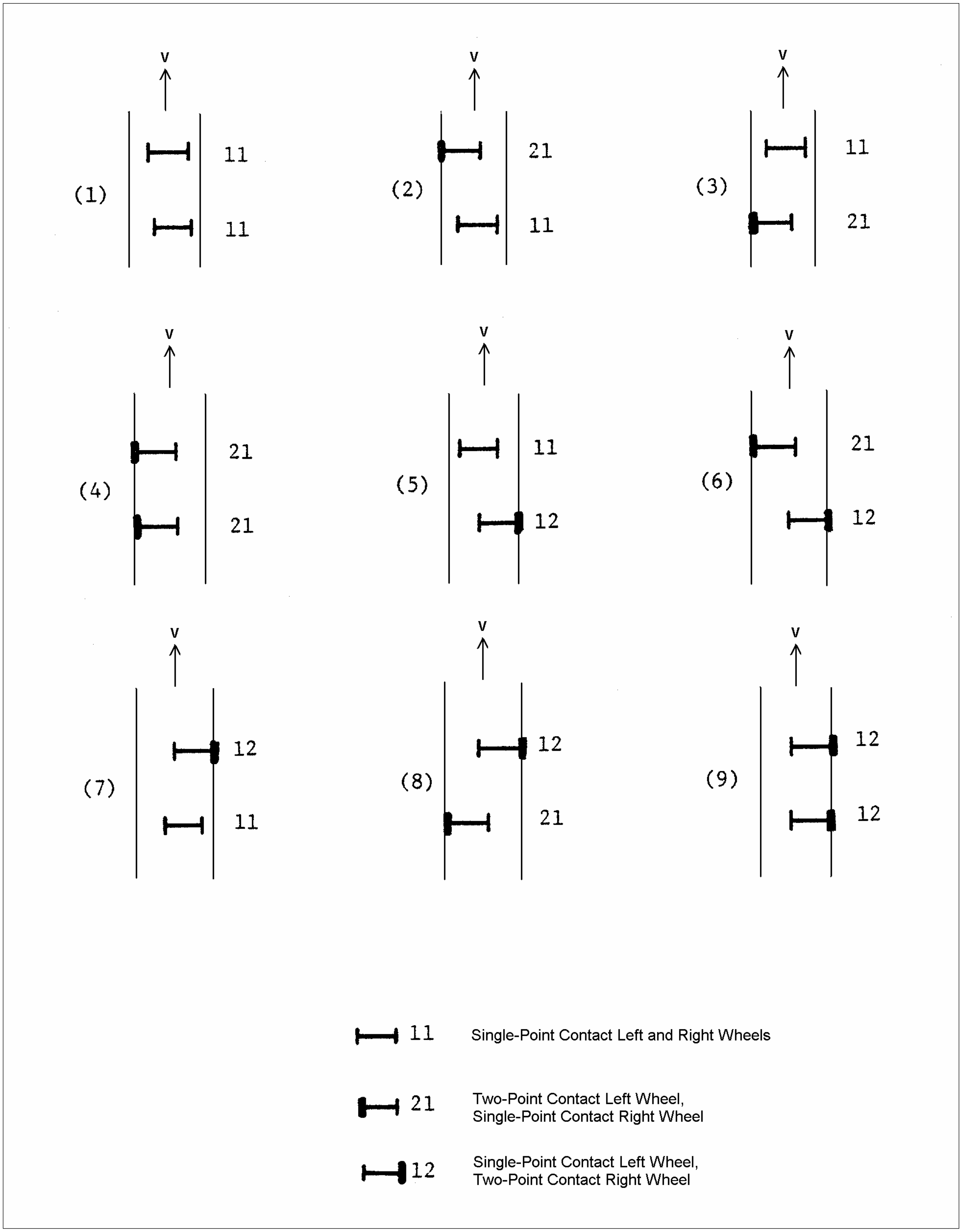

v
(1)
11
11
(2)
21
11
(3)
11
21
(4)
21
21
(5)
11
12
(6)
21
12
(7)
12
11
(8)
12
21
(9)
12
12
11 Single-Point Contact Left and Right Wheels
21 Two-Point Contact Left Wheel, Single-Point Contact Right Wheel
12 Single-Point Contact Left Wheel, Two-Point Contact Right Wheel

Figure 3-7 Wheel/Rail Contact Scenarios in a Single Truck

Yaw velocity of truck frame = 0.0 rad/s,
Left and right rail displacements = 0.0 m.

Figures 3-8 depicts the front and the rear wheelset as well as the truck time-response at a vehicle velocity of 40 m/s for K_{PX} = 9.12e5 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 8376.9 N-s/m, and C_{PY} = 9048.2 N-s/m. The critical velocity for this configuration is 33 m/s. At this speed, it is seen that the front and the rear wheels are flanging for a majority of the simulation time. The lateral motion of the wheelsets is limited by the flange width on both the left and the right wheels. The truck exhibits lateral hunting motion as well with an amplitude of nearly 0.003 m.

The sensitivity of critical velocity to primary spring stiffness is tabulated below in Table 3-3. The values of K_{PY} and K_{PX} are varied in geometric progression, with a common ratio of 3.14. This leads to every alternate value being a multiple or a sub-multiple by a factor of 10. Figures 3-9 and 3-10 plot the variation of critical velocity with longitudinal and lateral spring stiffness.

From Table 3-3 and Figures 3-9 and 3-10, it is seen that the critical velocity increases with K_{PX} for the range of K_{PY} that was considered. The variation of critical velocity is larger at lower values of K_{PX}. The variation of critical velocity with higher values of K_{PX} is practically linear. Furthermore, it is seen that the variation of the critical velocity with K_{PY} is almost negligible at low values of K_{PX}. As the value of K_{PX} is increased to higher values, the critical velocity increases marginally at low values of K_{PY} and then linearly drops with increasing K_{PY}.

In conclusion, the critical velocity variation of a truck behaves very differently as compared to that of a single free wheelset. In a free wheelset, higher longitudinal and lateral spring stiffness always result in higher critical velocities. In a truck, higher longitudinal spring stiffness results in higher critical velocities. However, increase in the lateral spring stiffness results in roughly constant, if not decreasing critical velocities.

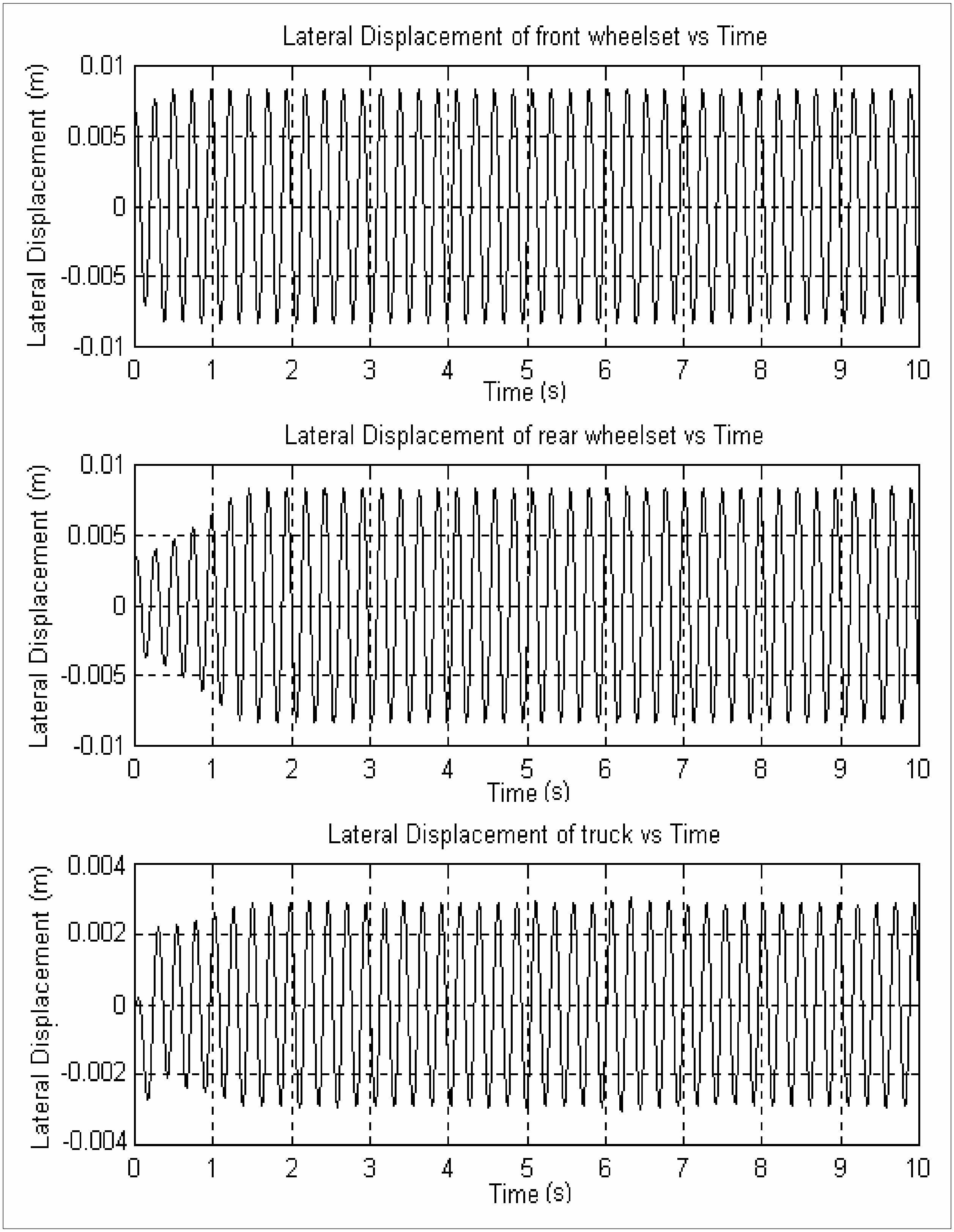


Figure 3-8 Single Truck Response: K_{PX} = 9.12e5 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (>V_C) = 40 m/s

Table 3-3 Sensitivity of Single Truck Critical Velocity to Primary Longitudinal and Lateral Spring Stiffness

K_{PX} (N/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Truck (V_C), m/s (km/hr)			
2.85e4	10 (36)	12 (43)	13 (47)	13 (47)
9.12e4	13 (47)	14 (50)	15 (54)	15 (54)
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	33 (119)	34 (122)	30 (108)	28 (101)
2.85e6	62 (223)	61 (220)	67 (241)	45 (162)

Note: All values in Table 3-3 above have been obtained with C_{PY} = 9048.2 N-s/m and C_{PX} = 8376.9 N-s/m.

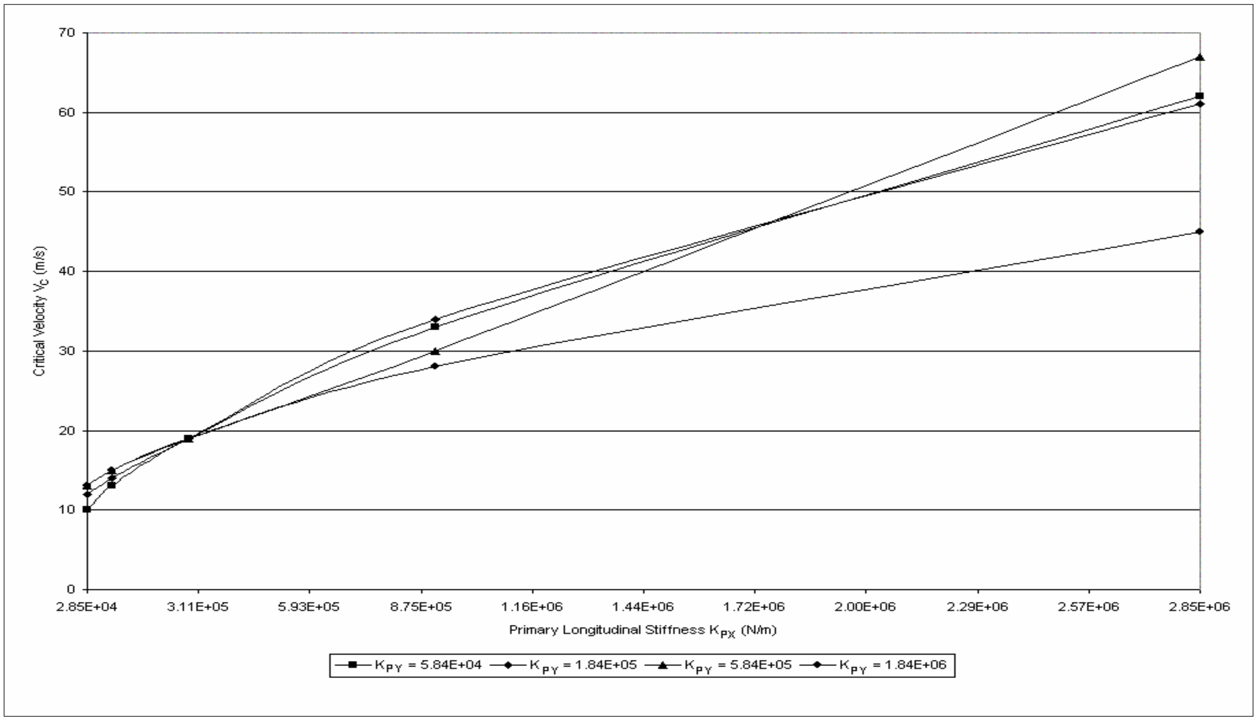


Figure 3-9 Variation of Single Truck Critical Velocity (V_C) with Primary Longitudinal Spring Stiffness (K_{PX})

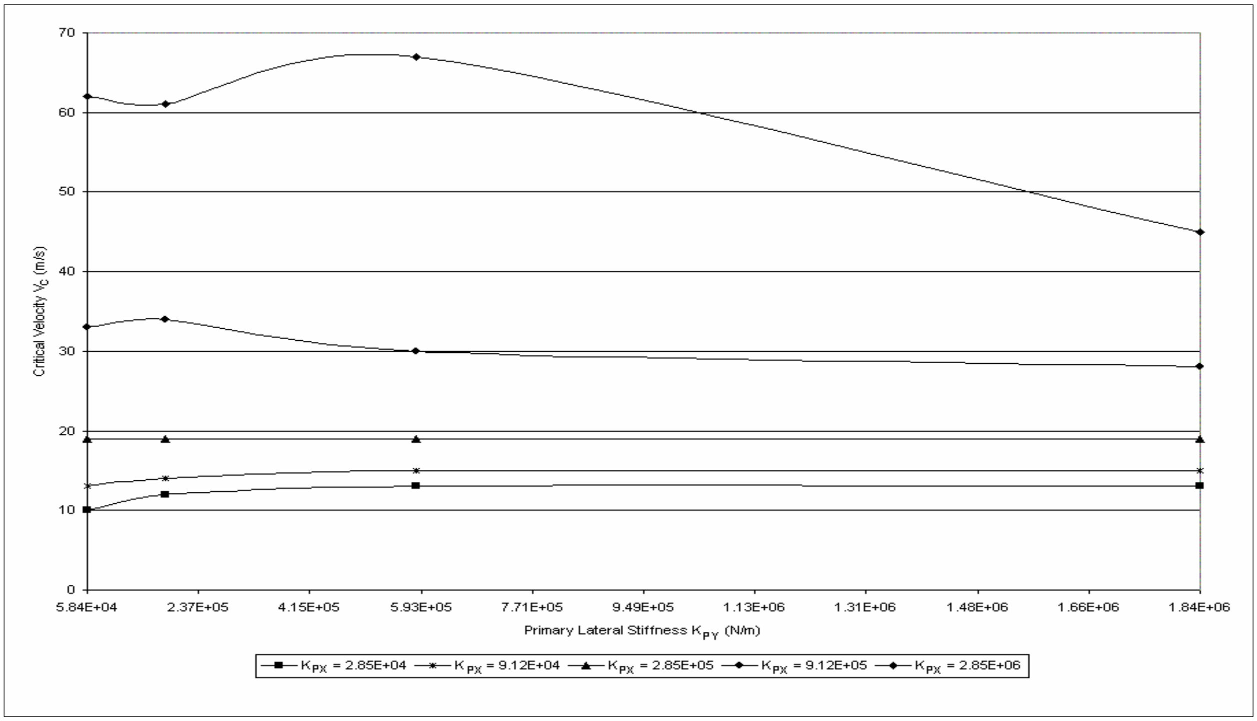


Figure 3-10 Variation of Single Truck Critical Velocity (V_C) with Primary Lateral Spring Stiffness (K_{PY})

3.3.2 The Effect of Primary Damping on the Critical Velocity

The effect on the single truck critical velocity of varying C_{PX} and C_{PY} is illustrated in Tables 3-4 and 3-5 respectively. The tables show the effect of varying the damping factors while keeping the values of K_{PX} and K_{PY} fixed. The data contained in these tables are presented graphically in Figures 3-13 and 3-14.

Figure 3-11 depicts the front and the rear wheelset as well as the truck time-response at a vehicle velocity of 25 m/s for K_{PX} = 2.85e5 N/m, K_{PY} = 5.84e5 N/m, C_{PX} = 3350 N-s/m, and C_{PY} = 9048.2 N-s/m. The critical velocity for this configuration is 19 m/s. At this speed, it is seen that the wheels are flanging for a majority of the simulation time. The lateral motion of the wheelsets is limited by the flange width on both the left and the right wheels. The truck exhibits lateral hunting motion as well with an amplitude of nearly 0.009 m.

Figure 3-12 depicts the front and the rear wheelset as well as the truck time-response at a vehicle velocity of 30 m/s for K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, and C_{PY} = 45240 N-s/m. The critical velocity for this configuration is 53 m/s. At this speed, it is seen that the response of the wheelsets and the truck damps out sharply and reaches the origin in about 2.5 seconds.

From Table 3-4 and Figure 3-13, it is seen that except at high values of K_{PX} there is no significant variation of critical speed with C_{PX}. At high values of K_{PX}, the critical velocity remains practically constant for lower values of C_{PX}, but decreases for higher values.

From Table 3-5 and Figure 3-14, it is seen that except at higher values of K_{PX} there is no significant variation of critical speed with C_{PY}. At high values of K_{PX}, the critical velocity increases for very low values of C_{PY}, but decreases for higher values.

In conclusion, the critical velocity of a single truck is barely sensitive to changes in primary damping. The critical velocity does not always increase with higher values of longitudinal and lateral damping, as was the case for a single wheelset.

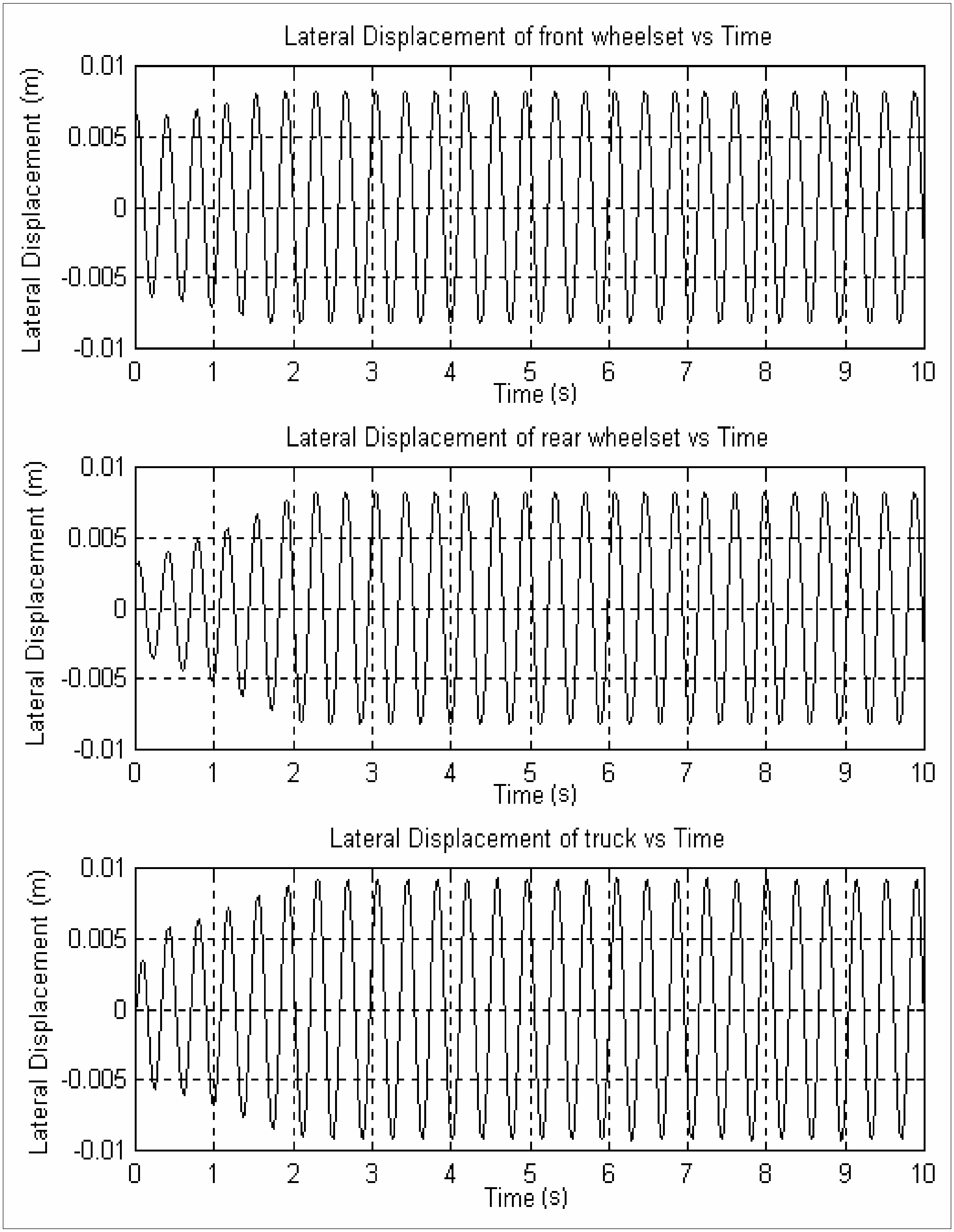


Figure 3-11 Single Truck Response: KPX = 2.85e5 N/m, KPY = 5.84e5 N/m, CPX = 3350 N-s/m, CPY = 9048.2 N-s/m, Velocity (>VC) = 25 m/s

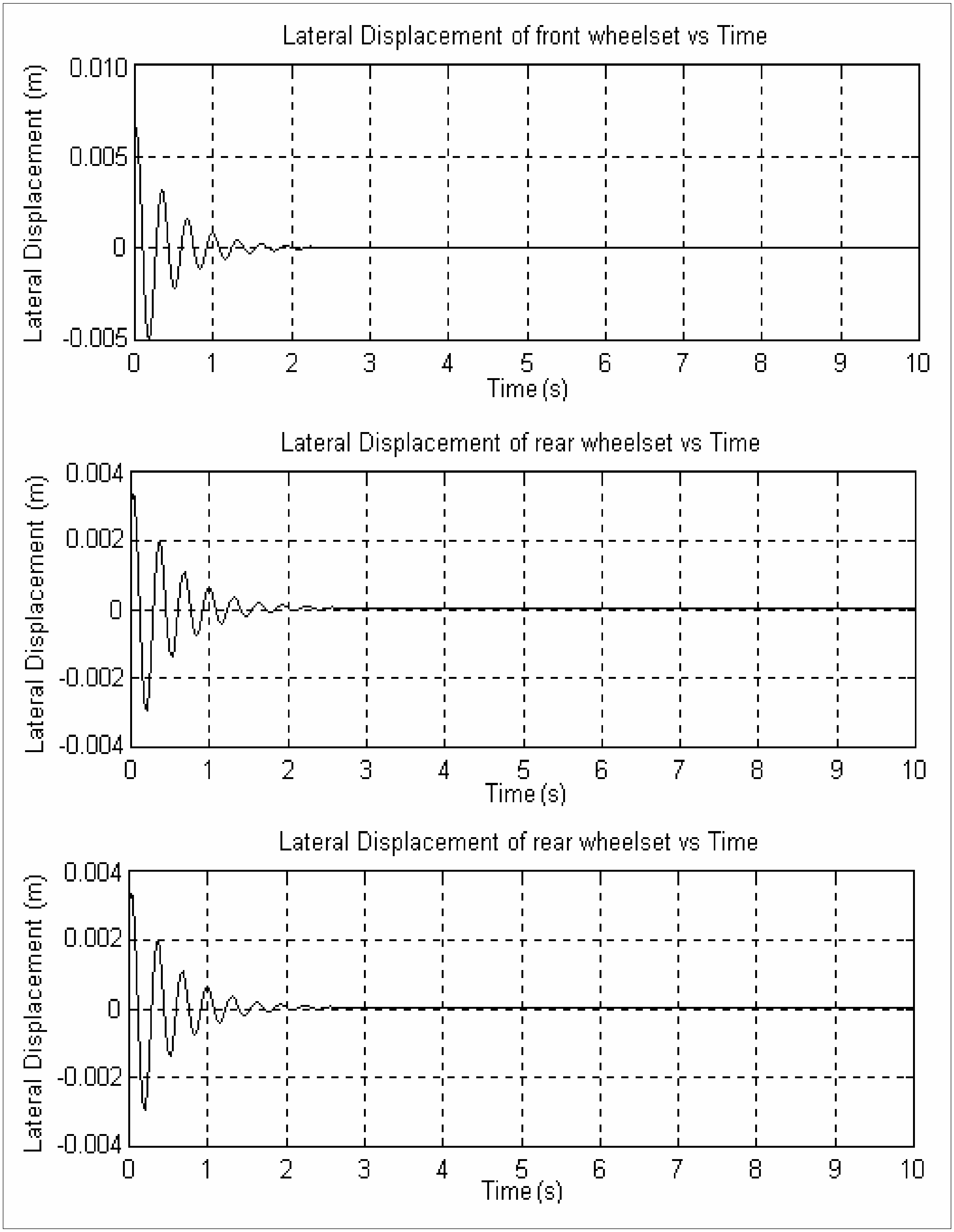


Figure 3-12 Single Truck Response: K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 45240 N-s/m, Velocity (<V_C) = 30 m/s

Table 3-4 Sensitivity of Single Truck Critical Velocity to Primary Longitudinal Damping

K_{PX} (N/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Truck (V_C), m/s (km/hr)			
C_{PX} = 1675 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	32 (115)	34 (122)	30 (108)	28 (101)
2.85e6	60 (216)	62 (223)	68 (245)	45 (162)
C_{PX} = 3350 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	32 (115)	34 (122)	30 (108)	28 (101)
2.85e6	60 (216)	62 (223)	68 (245)	45 (162)
C_{PX} = 8376.9 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	33 (119)	34 (122)	30 (108)	28 (101)
2.85e6	62 (223)	61 (220)	67 (241)	45 (162)
C_{PX} = 25120 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	33 (119)	33 (119)	30 (108)	28 (101)
2.85e6	61 (220)	60 (216)	67 (241)	44 (158)
C_{PX} = 41880 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	33 (119)	33 (119)	30 (108)	28 (101)
2.85e6	48 (173)	51 (184)	58 (209)	43 (155)

Note: All values in Table 3-4 above have been obtained with C_{PY} = 9048.2 N-s/m.

Table 3-5   Sensitivity of Single Truck Critical Velocity to Primary Lateral Damping

K_{PX} (N/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Truck (V_C), m/s (km/hr)			
C_{PY} = 1810 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	33 (119)	38 (137)	29 (104)	28 (101)
2.85e6	51 (184)	54 (194)	66 (238)	43 (155)
C_{PY} = 9048.2 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	33 (119)	34 (122)	30 (108)	28 (101)
2.85e6	62 (223)	61 (220)	67 (241)	45 (162)
C_{PY} = 45240 N-s/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	29 (104)	29 (104)	29 (104)	28 (101)
2.85e6	53 (191)	53 (191)	52 (187)	48 (173)

Note: All values in Table 3-5 above have been obtained with C_{PX} = 8376.9 N-s/m, K_{SY} = 3.5e5 N/m, and C_{SY} = 1.75e4 N-s/m.

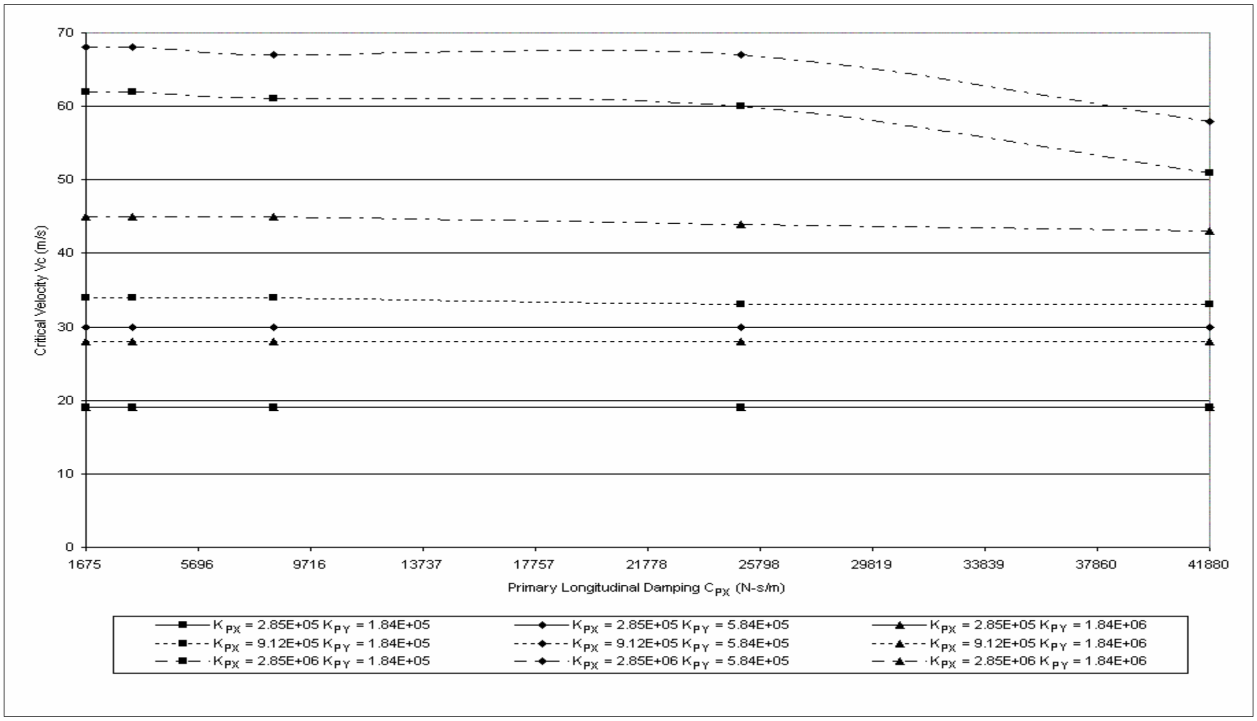


Figure 3-13 Variation of Single Truck Critical Velocity (Vc) with Primary Longitudinal Damping (CPX)

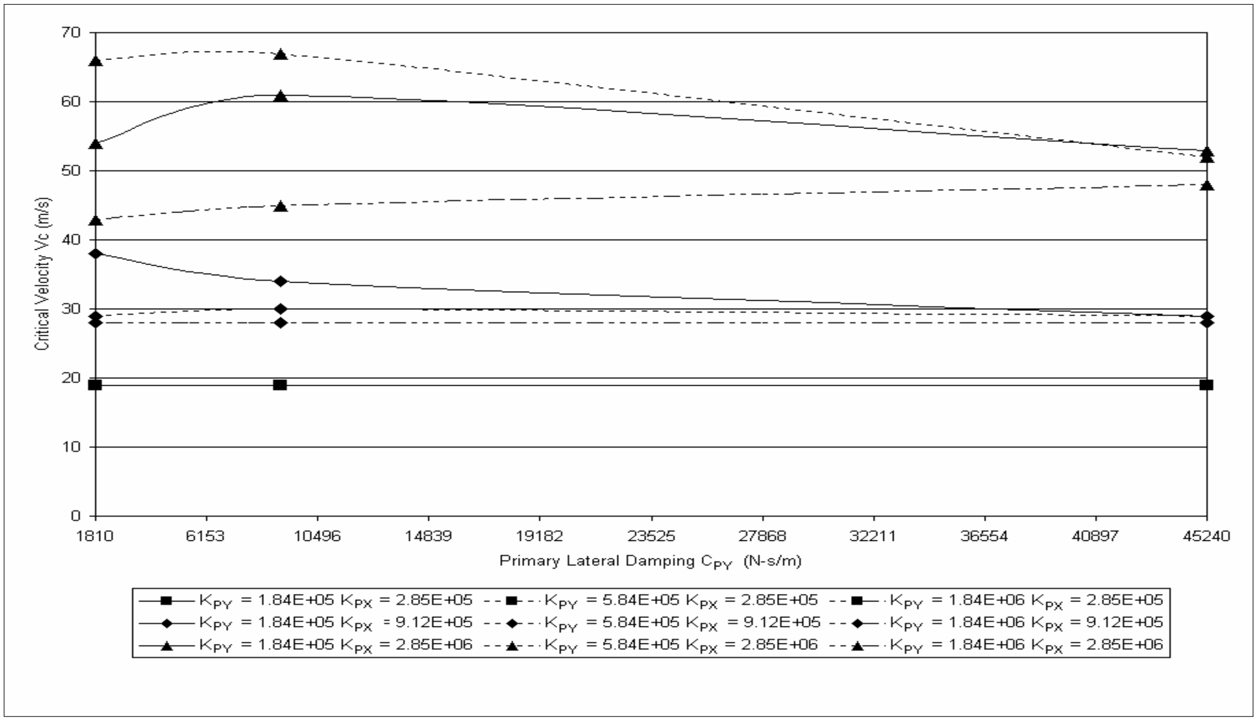


Figure 3-14 Variation of Single Truck Critical Velocity (Vc) with Primary Lateral Damping (CPY)

3.3.3 The Effect of Secondary Lateral Stiffness on the Critical Velocity

The sensitivity of single truck critical velocity to K_{SY} is illustrated in Table 3-6. The data contained in the table is presented graphically in Figure 3-17.

Figures 3-15 and 3-16 depict the front and the rear wheelset, the truck, and the left and right rail time-response at a vehicle velocity of 70 m/s for K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, and K_{SY} = 7e4 N/m. The critical velocity for this configuration is 61 m/s. At this speed, it is seen that the wheels are flanging for a majority of the simulation time. The lateral motion of the wheelsets is limited by the flange width on both the left and the right wheels. The truck exhibits lateral hunting motion as well with an amplitude of nearly 0.003 m. The left and the right rails show spikes in their response due to the flanging of the wheels. Peak rail lateral response amplitudes are seen to be around 6e-4 m.

From Table 3-6 and Figure 3-17, it is seen that the critical velocity typically remains unchanged with K_{SY} for low primary stiffness, but increases with K_{SY} for higher primary stiffness values.

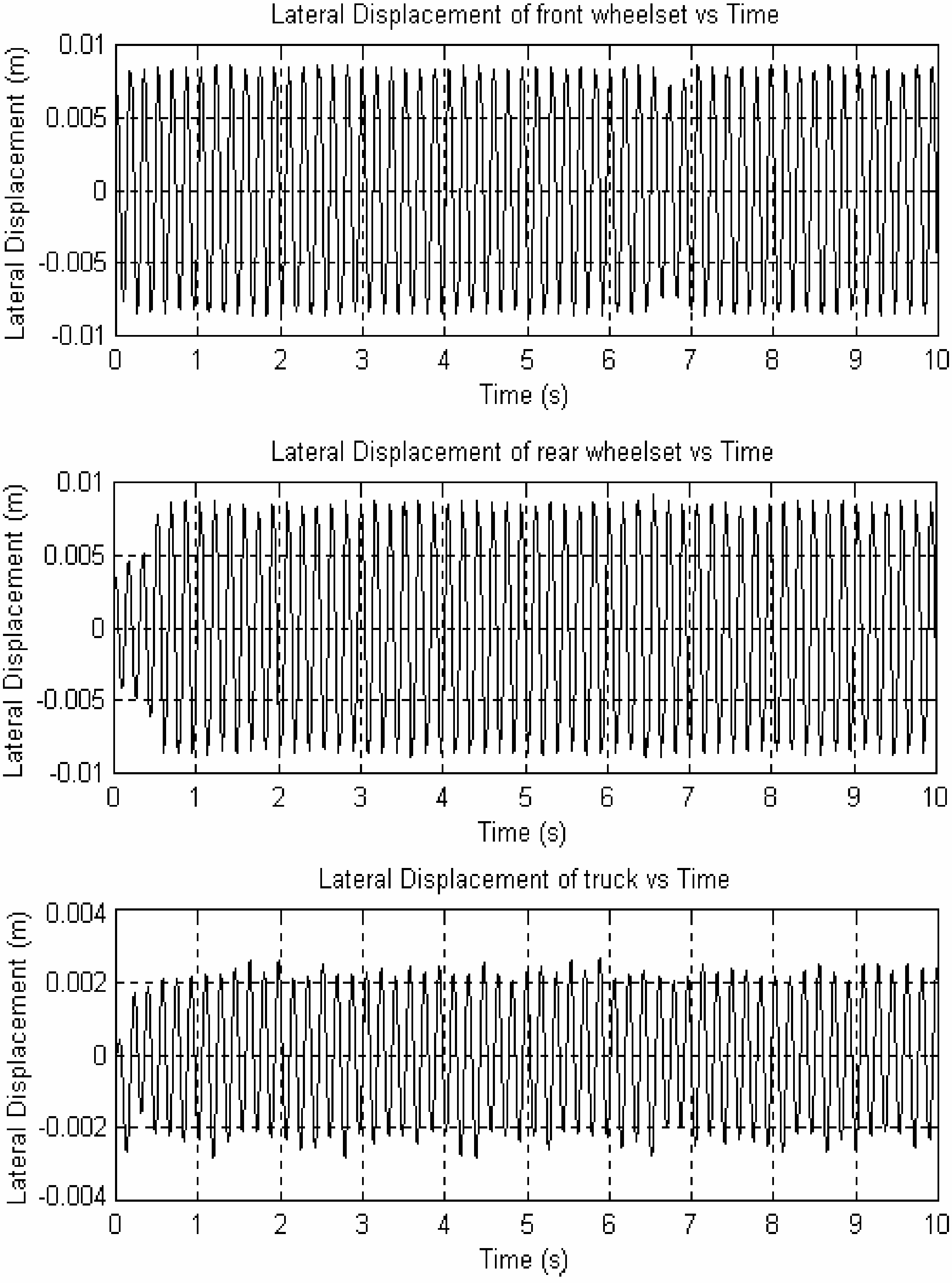


Figure 3-15 Single Truck Response: K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, K_{SY} = 7e4 N/m, Velocity (>V_C) = 70 m/s

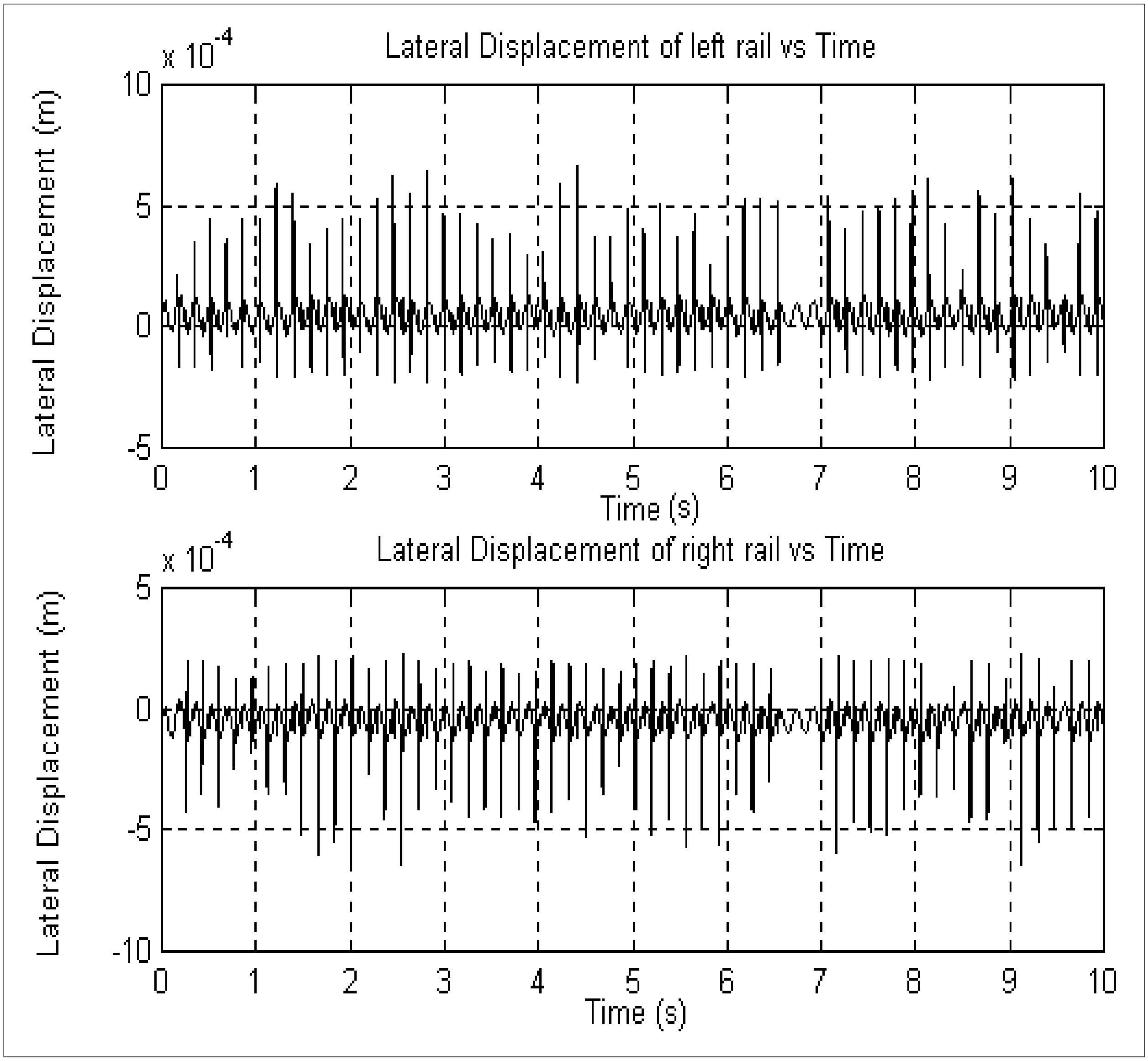


Figure 3-16 Leading Rail Response: K_{PX} = 2.85e6 N/m, K_{PY} = 1.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, K_{SY} = 7e4 N/m, Velocity (>V_C) = 70 m/s

Table 3-6 Sensitivity of Single Truck Critical Velocity to Secondary Lateral Stiffness

K_{PX} (N/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Truck (V_C), m/s (km/hr)			
K_{SY} = 7e4 N/m				
2.85e5	19 (68)	19 (68)	15 (54)	15 (54)
9.12e5	33 (119)	34 (122)	30 (108)	26 (94)
2.85e6	61 (220)	61 (220)	67 (241)	42 (151)
K_{SY} = 3.5e5 N/m				
2.85e5	19 (68)	19 (68)	19 (68)	19 (68)
9.12e5	33 (119)	34 (122)	30 (108)	28 (101)
2.85e6	62 (223)	61 (220)	67 (241)	45 (162)
K_{SY} = 1.75e6 N/m				
2.85e5	21 (76)	23 (83)	27 (97)	29 (104)
9.12e5	34 (122)	35 (126)	35 (126)	37 (133)
2.85e6	59 (212)	59 (212)	57 (205)	53 (191)

Note: All values in Table 3-6 above have been obtained with C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, and C_{SY} = 1.75e4 N-s/m.

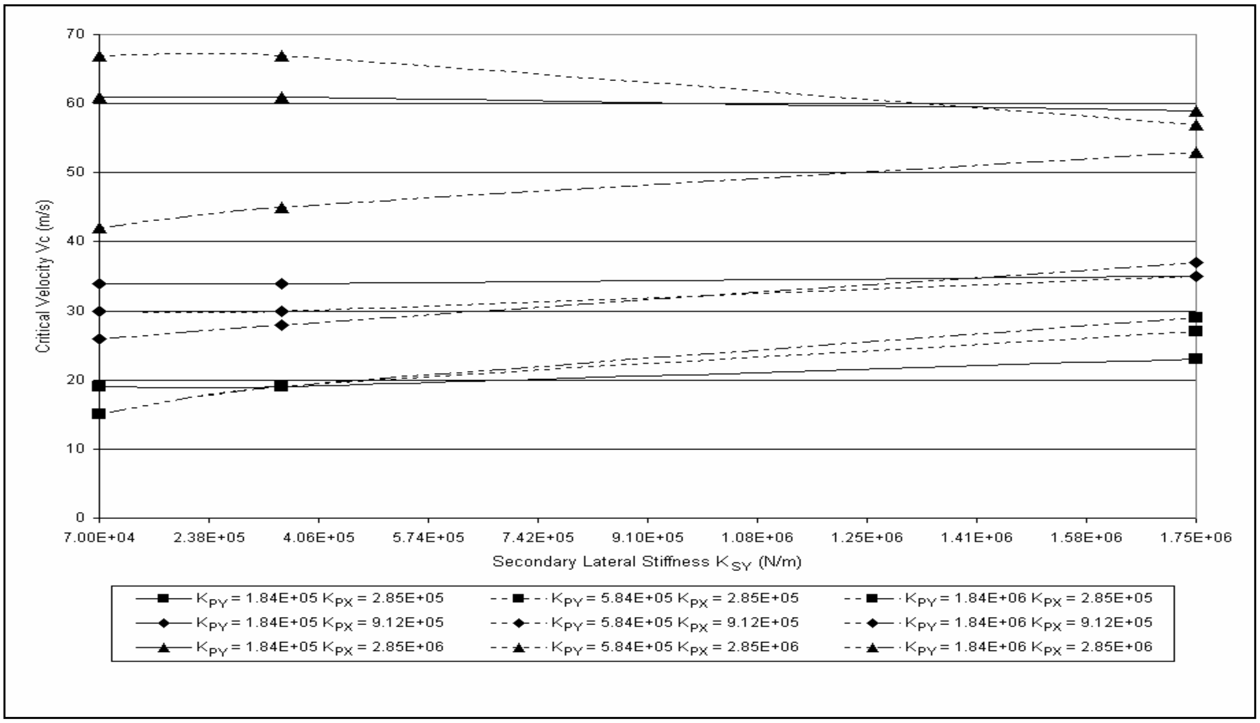


Figure 3-17 Variation of Single Truck Critical Velocity (V_C) with Secondary Lateral Stiffness (K_{SY})

3.3.4 The Effect of Breakaway Torque on the Critical Velocity

The sensitivity of single truck critical velocity to T_0 is illustrated in Table 3-7. The data contained in the table is presented graphically in Figure 3-20.

Figures 3-18 and 3-19 depict the front and the rear wheelset, as well as the truck and the bolster time-response at a vehicle velocity of 57 m/s for K_{PX} = 2.85e6 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 1675 N-s/m, C_{PY} = 9048.2 N-s/m, and T_0 = 15252 N-m. The critical velocity for this configuration is 61 m/s. At this speed, it is seen that the wheel, truck, and bolster response are well damped, reaching the origin after about 7 seconds.

From Table 3-7 and Figure 3-20, it is seen that for the range of K_{PY} considered, the critical velocity typically remains unchanged with T_0 at low values of primary longitudinal damping C_{PX}. However, at high values of C_{PX}, the critical velocity increases considerably with T_0.

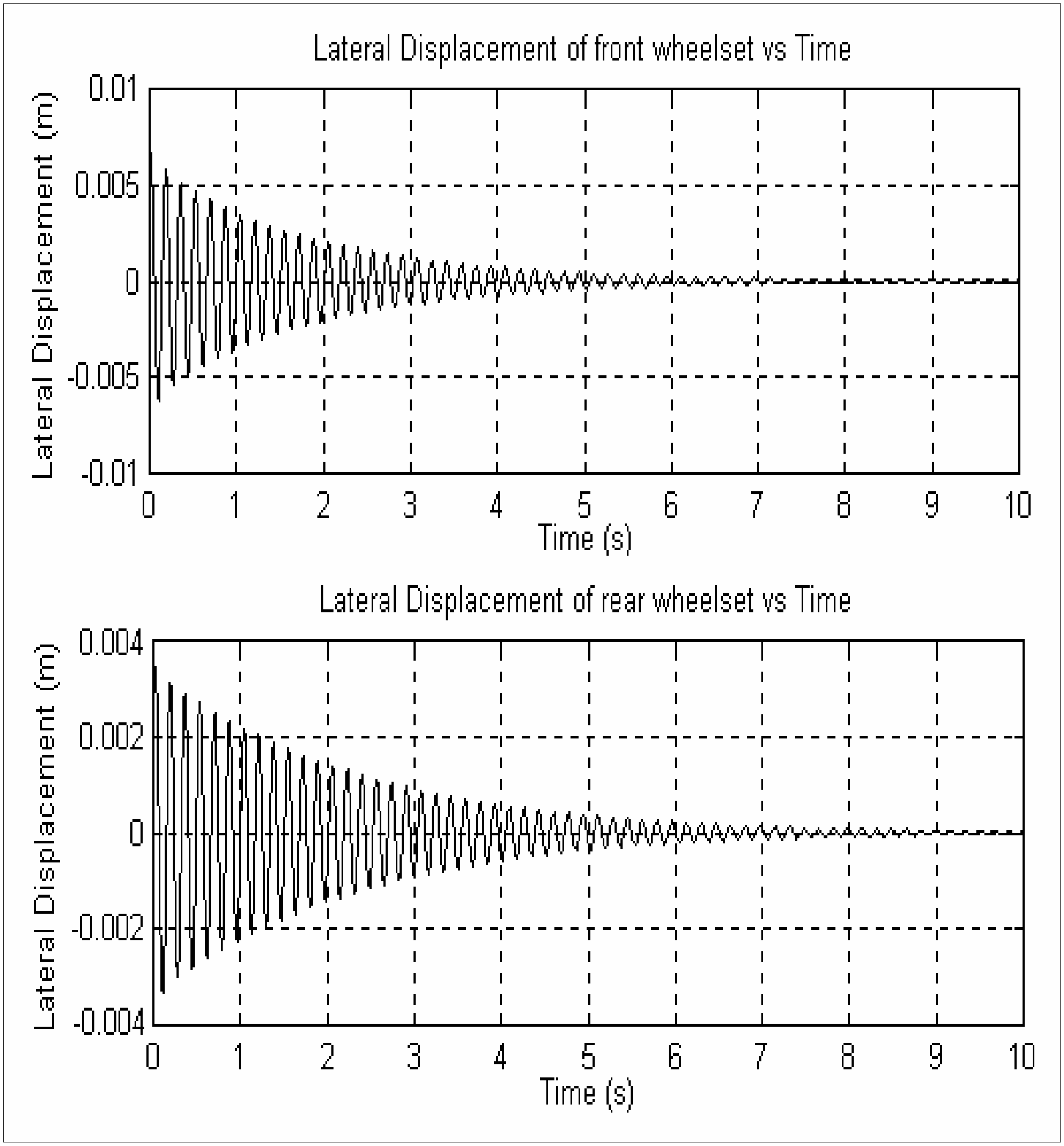


Figure 3-18 Wheelset Response: K_{PX} = 2.85e6 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 1675 N-s/m, C_{PY} = 9048.2 N-s/m, T_0 = 15252 N-m, Velocity (<V_C) = 57 m/s

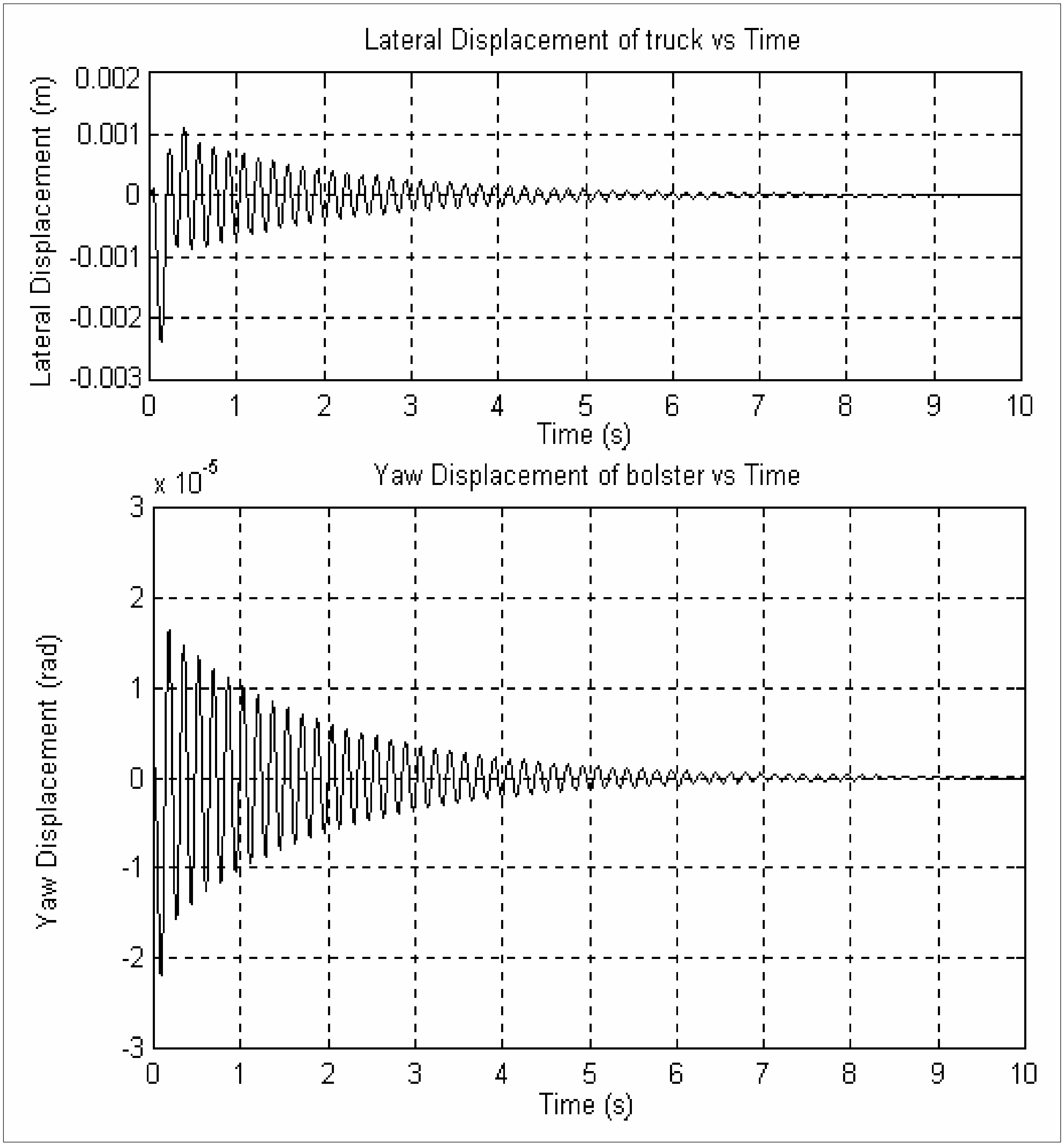


Figure 3-19 Truck and Bolster Response: K_{PX} = 2.85e6 N/m, K_{PY} = 5.84e4 N/m, C_{PX} =1675 N-s/m, C_{PY} = 9048.2 N-s/m, T_0 = 15252 N-m, Velocity (<V_C) = 57 m/s

Table 3-7 Sensitivity of Single Truck Critical Velocity to Breakaway Torque

C_{PX} (N-s/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Single Truck (V_C), m/s (km/hr)			
T_0 = 10168 N-m				
1675	60 (216)	62 (223)	68 (245)	45 (162)
3350	60 (216)	62 (223)	68 (145)	45 (162)
8376.9	62 (223)	61 (220)	67 (241)	45 (162)
25120	61 (220)	60 (216)	67 (241)	44 (158)
41880	48 (173)	51 (184)	58 (209)	43 (155)
T_0 = 1.5 x 10168 N-m				
1675	61 (220)	63 (227)	71 (256)	51 (184)
3350	61 (220)	63 (227)	72 (259)	51 (184)
8376.9	62 (223)	64 (230)	72 (259)	51 (184)
25120	62 (223)	64 (230)	73 (263)	51 (184)
41880	60 (216)	64 (230)	74 (266)	51 (184)
T_0 = 2.0 x 10168 N-m				
1675	62 (223)	63 (227)	70 (252)	51 (184)
3350	62 (223)	63 (227)	71 (256)	51 (184)
8376.9	62 (223)	64 (230)	72 (259)	51 (184)
25120	63 (227)	65 (234)	74 (266)	51 (184)
41880	65 (234)	66 (238)	79 (284)	51 (184)

Note: All values in Table 3-7 above have been obtained with K_{PX} = 2.85e6 N/m, C_{PY} = 9048.2 N-s/m, K_{SY} = 3.5e5 N/m, and C_{SY} = 1.75e4 N-s/m.

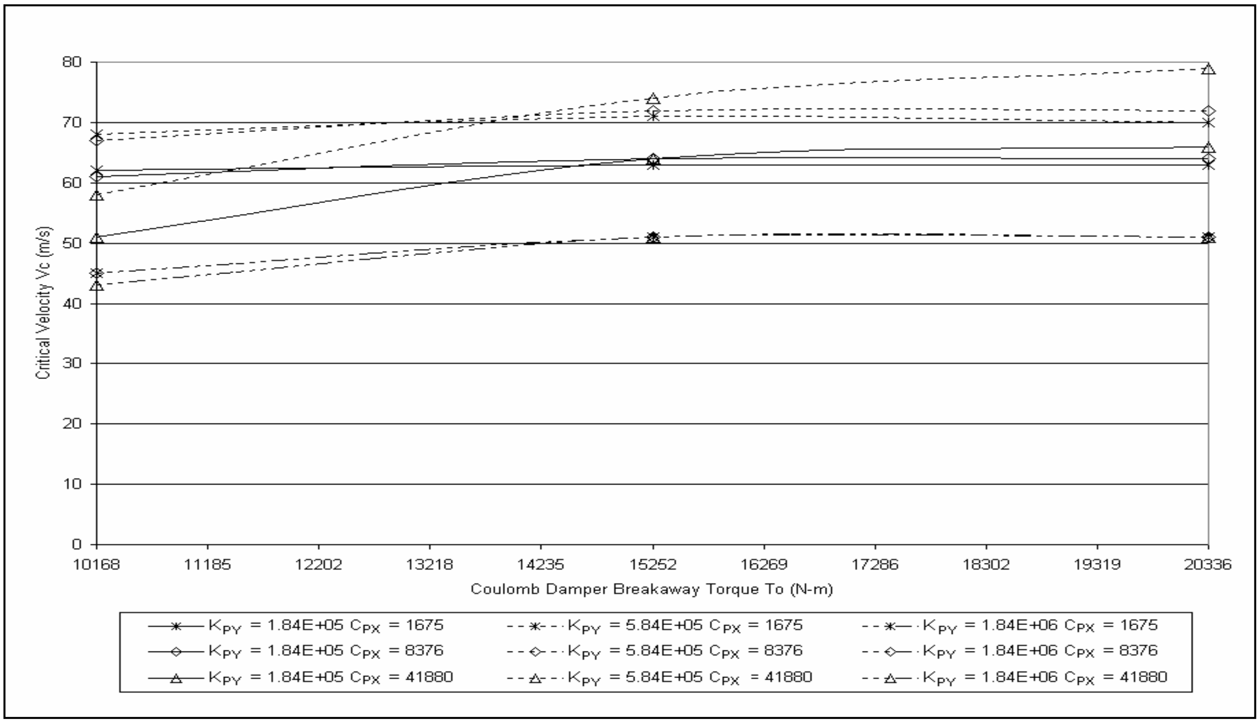


Figure 3-20 Variation of Single Truck Critical Velocity (V_C) with Coulomb Damper Breakaway Torque (T_0)

3.4 Conclusions

This chapter provided the mathematical formulation for a single conventional truck model. The forces and moments that act on a single truck and bolster, and which govern the lateral and yaw motions of the truck and bolster were obtained and the equations which define the dynamics of the conventional truck were enumerated. Both single-point and two-point wheel / rail contact conditions were considered.

Sensitivity of the single truck critical velocity to primary longitudinal and lateral stiffness, primary longitudinal and lateral damping, secondary lateral stiffness, and Coulomb damper breakaway torque was studied.

Table 3-3 and Figures 3-9 and 3-10 depict the sensitivity of the single truck critical velocity to variations in the primary longitudinal and lateral spring stiffness. The simulations indicate a supercritical Hopf bifurcation phenomenon, where the origin loses its stability after the critical velocity is reached, and gives rise to a nearly elliptical limit cycle about the origin. The results also indicate that the critical velocity increases with K_{PX} for the range of K_{PY} that was considered. The variation of critical velocity is larger at lower values of K_{PX}. The variation of critical velocity with higher values of K_{PX} is practically linear. Furthermore, it is seen that the variation of the critical velocity with K_{PY} is almost negligible at low values of K_{PX}. As the value of K_{PX} is increased to higher values, the critical velocity increases marginally at low values of K_{PY} and then linearly drops with increasing K_{PY}. In conclusion, the critical velocity variation of a truck behaves very differently as compared to that of a single free wheelset. In a free wheelset, higher longitudinal and lateral spring stiffness always result in higher critical velocities. In a truck, higher longitudinal spring stiffness results in higher critical velocities. However, increase in the lateral spring stiffness results in roughly constant, if not decreasing critical velocities.

Tables 3-4 and 3-5 and Figures 3-13 and 3-14 depict the sensitivity of the single truck critical velocity to variations in the primary longitudinal and lateral damping. It is seen that except at high values of K_{PX} there is no significant variation of critical velocity with C_{PX} and C_{PY}. At high values of K_{PX}, the critical velocity remains practically constant for lower values of C_{PX} and C_{PY}, but decreases for higher values. In conclusion, the critical velocity of a single truck is barely sensitive to changes in primary damping. The critical velocity does not always increase with higher values of longitudinal and lateral damping, as was the case for a single wheelset.

Table 3-6 and Figure 3-17 depict the sensitivity of the single truck critical velocity to variations in the secondary lateral stiffness. It is seen that the critical velocity typically remains unchanged with K_{SY} for low primary stiffness, but increases with K_{SY} for higher primary stiffness values.

Table 3-7 and Figure 3-20 depict the sensitivity of the single truck critical velocity to variations in the Coulomb damper breakaway torque. The results show that for the range of K_{PY} considered, the critical velocity typically remains unchanged with T_0 at low values of primary longitudinal damping C_{PX}. However, at high values of C_{PX}, the critical velocity increases considerably with T_0.

This chapter concludes that for the range of parameters investigated, the single truck critical velocity is most sensitive to the primary longitudinal spring stiffness. An increase in the value of K_{PX} from 2.85e4 N/m to 2.85e6 N/m results in as much as a 55 m/s increase in the critical velocity. The Coulomb friction breakaway torque is the next most effective parameter in increasing the critical velocity. An increase in T_0 from 10168 N-m to 20336 N-m results in as much as a 21 m/s increase in critical velocity. The critical velocity of a single truck can thus be most effectively controlled by active or semi-active adjustment of the primary longitudinal spring stiffness.

Chapter 4

Rail Vehicle Model

This chapter provides the mathematical formulation for the full vehicle model. The forces and moments which act on the full vehicle and which govern the lateral, yaw, and roll motions of the carbody are obtained and the equations which define the dynamics of the rail vehicle are enumerated. The full vehicle configuration includes a front and a rear simple truck model. Each truck has a front (leading) and a rear (trailing) wheelset. The simple truck model considered in this chapter is identical to the single truck model considered in Chapter 3. Since only conventional trucks are considered, no steering connection exists between the front and the rear wheelsets of each truck, and each of the four wheelsets is free to move independently. The coupling between the two forward wheelsets is only through the front truck and the coupling between the two rear wheelsets is only through the rear truck. This coupling was described in the last chapter. In addition, the front and the rear truck are coupled through a front and a rear bolster and secondary suspension elements to the motion of the carbody. Each of the four wheelsets is identical to the single wheelset considered in Chapter 2. The full vehicle model is subjected to certain initial conditions. Both single-point and two-point wheel / rail contact conditions are considered. The mathematical equations governing the motion of the full vehicle [1] are simplified in order to reduce computation time without compromising on the accuracy of the solution. The simulation software MATLAB [2] was used to find time-domain solutions to the full vehicle dynamic equations. The critical velocities obtained for different primary and secondary suspension parameters are presented.

4.1 Mathematical Formulation

The full vehicle configuration consists of a front and a rear truck, each of which have a front (leading) and a rear (trailing) wheelset, a front and a rear bolster, and the carbody. The full vehicle model is depicted in Figure 4-1 and Figure 4-2. The truck model considered in this chapter is identical to the single truck model considered in Chapter 3. Since only conventional trucks are considered, no steering connection exists between the front and the rear wheelsets of each truck, and each of the four wheelsets is free to move independently. The coupling between the two forward wheelsets is only through the front truck and the coupling between the two rear wheelsets is only through the rear truck. This coupling was described in Chapter 3. In addition, the front and the rear truck are coupled through a front and a rear bolster and secondary suspension elements to the motion of the carbody. Each of the four wheelsets is identical to the single wheelset considered in Chapter 2.

The full vehicle model used in this chapter considers three independent degrees of freedom for the carbody, namely, the lateral (y_C), the yaw (ψ_C), and the roll (ϕ_C) motions. The carbody is connected to the each of the two trucks through a bolster. The front and the rear bolsters connect to the front and the rear trucks through a coulomb friction element. The bolsters move rigidly with the trucks in the lateral direction. The bolster is connected to the carbody by lateral secondary springs and by suspension elements, which produce very high restoring moments, ensuring that the bolster rotates with the car body.

The bolster and the carbody are coupled by two secondary lateral springs, each with a low spring constant K_{SY}, and two secondary lateral dampers, each with a viscous damping constant of c_{SY}. The secondary lateral suspension elements mentioned above are typically soft suspension elements in order to isolate the motion of the truck to the carbody. This results in a lower carbody movement and higher ride comfort. The carbody is coupled to the bolster via parallel spring / viscous damper elements in the lateral direction and a torsional spring / viscous damper combination in the yaw direction. The torsional spring and damper have very high values of stiffness K_ψ and viscous resistance C_ψ.

The force produced by the secondary springs not only acts on the truck, but as a reaction, also acts on the carbody and since the carbody is free to move, produces lateral movement of the carbody. The secondary dampers similarly produce lateral forces proportional to velocity terms coupling the carbody and the truck motions. The yaw motions of the carbody and the truck will also be coupled and the yawing moments due to secondary suspension including the coulomb friction term between the truck and the bolster will have carbody terms introduced in them. Similarly equations for the rear truck will also have the carbody lateral and angular movement terms, both in yaw and in roll degrees of freedom. The lateral springs will also produce roll of the carbody, since these forces are offset relative to the carbody center of gravity. Also, since the trucks are not at the center of the carbody, they will produce yawing motion of the carbody.

An important term in the coupling between the trucks and the carbody is the Coulomb friction between the bolster and the truck. The secondary suspension system comprising the spring and damper connect the bolster to the carbody. The secondary spring constant is low (nominal value taken to be 3.5e5 for simulation) to prevent violent lateral movement of the carbody. When the truck yaws, as long as the yawing moment produced by the truck is less than or equal to the coulomb friction moment, T_0, the bolster will try to move with the truck in the yaw degree of freedom. When the yawing moment exceeds T_0, the bolster will slip and move on its own. This ensures that yawing moments greater than T_0 are not applied to the carbody. The suspension between the carbody and the bolster typically has a high rotational stiffness and damping, so that the bolster turns almost rigidly with the carbody.

4.1.1 Truck Frame and Bolster Suspension Forces and Moments

Chapter 3 covered the suspension forces and moments for a single truck and bolster configuration with the carbody assumed to be fixed. This section presents the suspension forces and moments acting on the front and the rear truck and the corresponding bolsters.

Suspension forces act on the front and the rear truck frame in the lateral direction due to primary and secondary lateral springs and dampers. In addition, suspension moments also act on the front and the rear truck frame and bolster due to primary and secondary lateral springs and dampers, secondary yaw spring and damper, and the torsional Coulomb damper. Figure 4-3 shows the secondary yaw suspension model. Figures 4-4 and 4-5 show the suspension forces and moments acting on the front and the rear truck frame and bolster.

The suspension forces and moments acting on the leading and the trailing wheelsets of each truck were shown in equations (3.1) to (3.4) in Chapter 3.

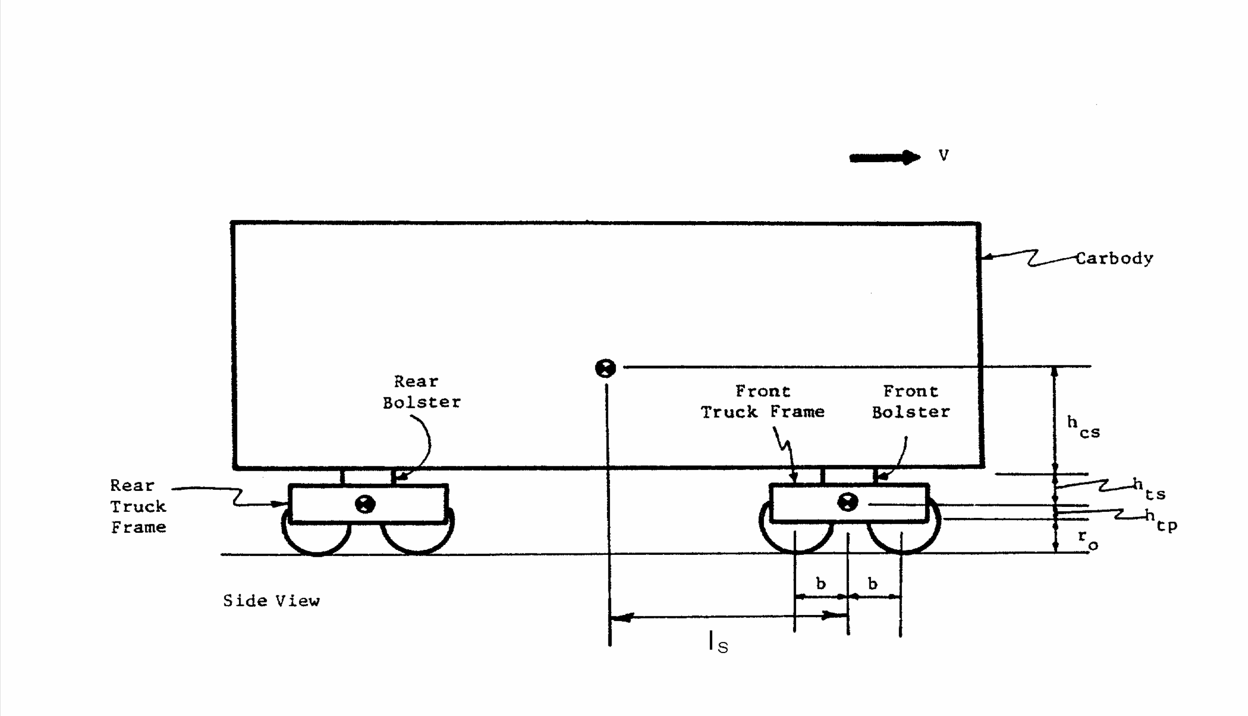


Figure 4-1 Rail Vehicle Model, Side View

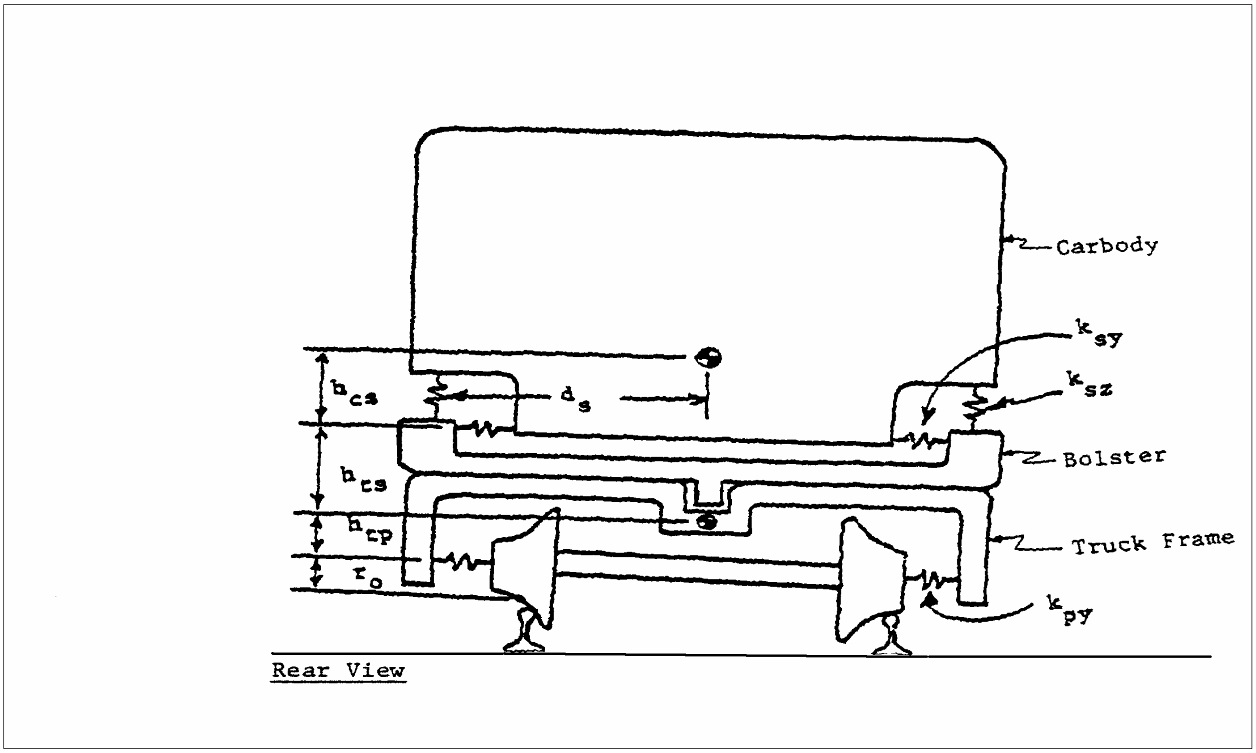


Figure 4-2 Rail Vehicle Model, Rear View

The lateral suspension force and the yaw suspension moment acting on the front truck are given by the expressions below.

$$\begin{aligned}F_{SUSPYF1} &= 2K_{PY}y_{W1}+2K_{PY}y_{W2}-4K_{PY}y_{F1}-2K_{SY}y_{F1}+2K_{SY}y_{C}+2K_{SY}l_{S}\psi_{C}+2C_{PY}\dot{y}_{W1}\\ &+2C_{PY}\dot{y}_{W2}-4C_{PY}\dot{y}_{F1}-2C_{SY}\dot{y}_{F1}+2C_{SY}\dot{y}_{C}+2C_{SY}l_{S}\dot{\psi}_{C}\end{aligned} \tag{4.1}$$

$$\begin{aligned}M_{SUSPZF1} &= 2bK_{PY}y_{W1}+2d_p^2K_{PX}\psi_{W1}-2bK_{PY}y_{W2}+2d_p^2K_{PX}\psi_{W2}+(-4d_p^2K_{PX}-4b^2K_{PY})\psi_{F1}\\ &+2bC_{PY}\dot{y}_{W1}+2d_p^2C_{PX}\dot{\psi}_{W1}-2bC_{PY}\dot{y}_{W2}+2d_p^2C_{PX}\dot{\psi}_{W2}+(-4d_p^2C_{PX}-4b^2C_{PY})\dot{\psi}_{F1}-T_{COUL}\end{aligned} \tag{4.2}$$

The lateral suspension force and the yaw suspension moment acting on the rear truck are given by the expressions below.

$$\begin{aligned}F_{SUSPYF2} &= 2K_{PY}y_{W3}+2K_{PY}y_{W4}-4K_{PY}y_{F2}-2K_{SY}y_{F2}+2K_{SY}y_{C}-2K_{SY}l_{S}\psi_{C}+2C_{PY}\dot{y}_{W3}\\ &+2C_{PY}\dot{y}_{W4}-4C_{PY}\dot{y}_{F2}-2C_{SY}\dot{y}_{F2}+2C_{SY}\dot{y}_{C}-2C_{SY}l_{S}\dot{\psi}_{C}\end{aligned} \tag{4.3}$$

$$\begin{aligned}M_{SUSPZF2} &= 2bK_{PY}y_{W3}+2d_p^2K_{PX}\psi_{W3}-2bK_{PY}y_{W4}+2d_p^2K_{PX}\psi_{W4}+(-4d_p^2K_{PX}-4b^2K_{PY})\psi_{F2}\\ &+2bC_{PY}\dot{y}_{W3}+2d_p^2C_{PX}\dot{\psi}_{W3}-2bC_{PY}\dot{y}_{W4}+2d_p^2C_{PX}\dot{\psi}_{W4}+(-4d_p^2C_{PX}-4b^2C_{PY})\dot{\psi}_{F2}-T_{COUL}\end{aligned} \tag{4.4}$$

The yaw suspension moment acting on the front bolster is given by:

$$M_{SUSPZB1} = -K_{S\psi}\psi_{B1}+K_{S\psi}\psi_{C}-C_{S\psi}\dot{\psi}_{B1}+C_{S\psi}\dot{\psi}_{C}+T_{COUL} \tag{4.5}$$

The yaw suspension moment acting on rear bolster is given by:

$$M_{SUSPZB2} = -K_{S\psi}\psi_{B2}+K_{S\psi}\psi_{C}-C_{S\psi}\dot{\psi}_{B2}+C_{S\psi}\dot{\psi}_{C}+T_{COUL} \tag{4.6}$$

In equations (4.4), (4.5), and (4.6), T_{COUL} represents the Coulomb friction yaw moment acting on the truck frame due to interaction with the bolster. For numerical purposes, the model of the Coulomb friction is modified to include a linear viscous band at the origin, as shown in equation (4.7). At low relative yaw rates between the bolster and the truck frame, the model assumes viscous damping. At higher relative yaw rates, the model assumes Coulomb damping with the frictional torque saturating at the centerplate breakaway value. This method approximates the frictional torque levels below $|T_O|$.

$$T_{COUL} = \begin{cases} T_O \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;. \text{ for } (\dot{\psi}_F - \dot{\psi}_B) \geq T_O / C_O \\ C_O(\dot{\psi}_F - \dot{\psi}_B) \;.\;.\;.\;.\;. \text{ for } -T_O / C_O < (\dot{\psi}_F - \dot{\psi}_B) < T_O / C_O \\ -T_O \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;. \text{ for } (\dot{\psi}_F - \dot{\psi}_B) \leq -T_O / C_O \end{cases} \tag{4.7}$$

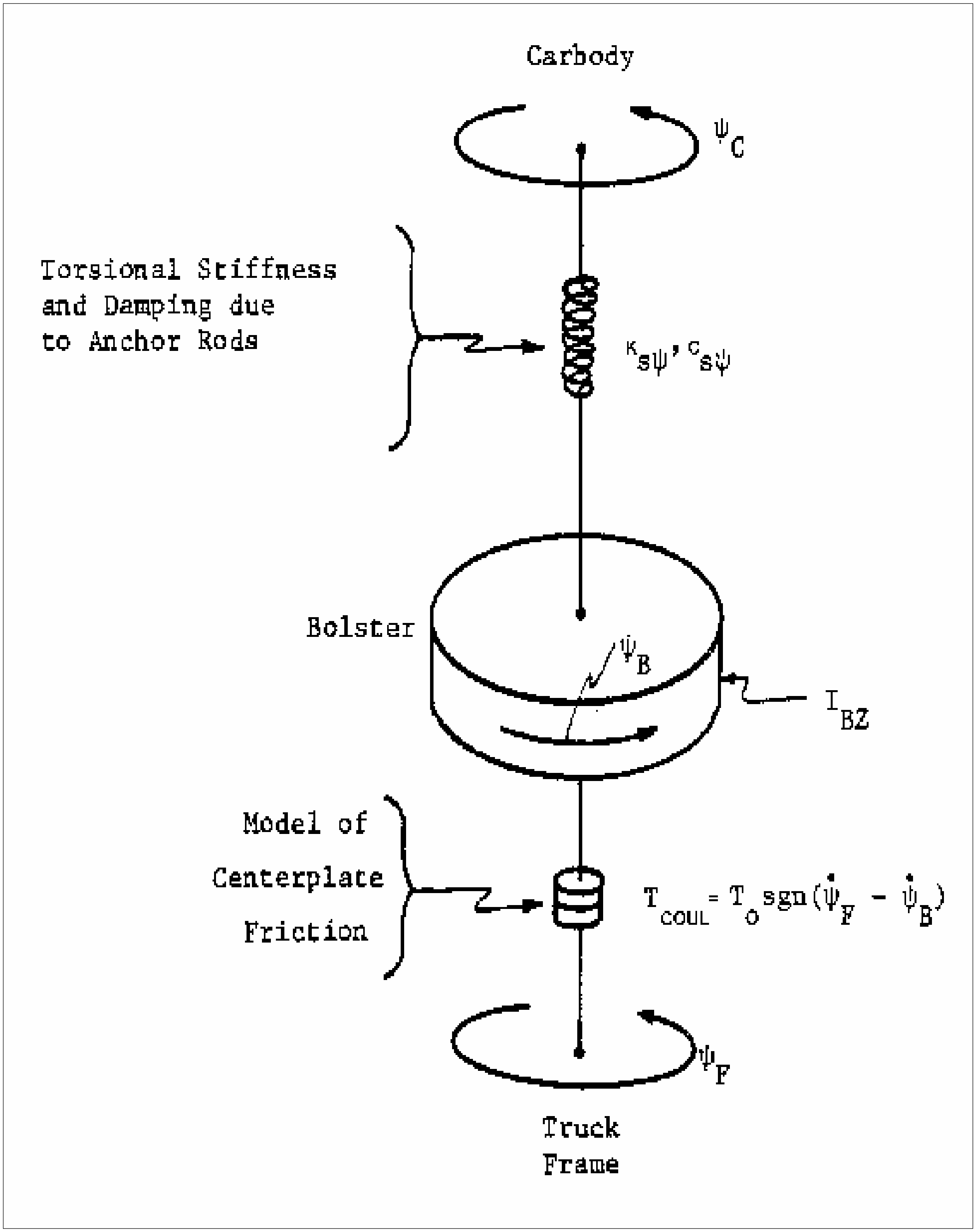


Figure 4-3 Secondary Yaw Suspension Arrangement

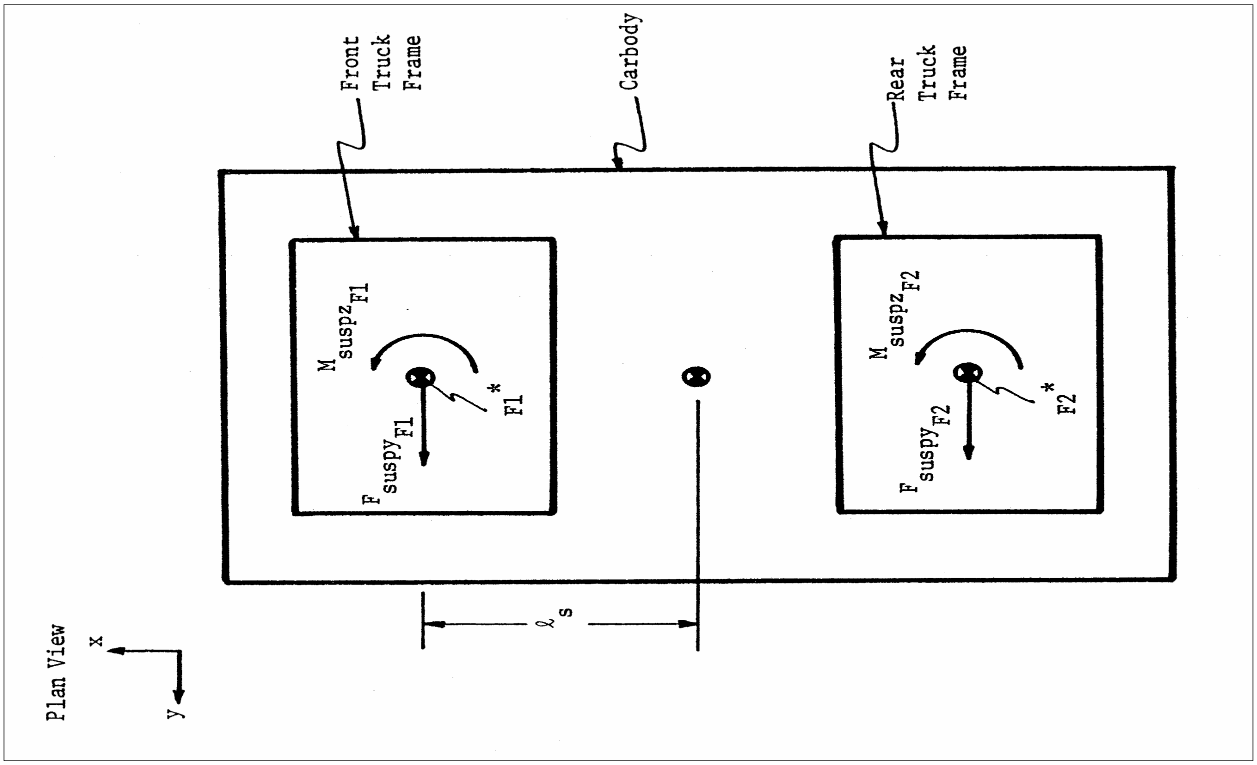


Figure 4-4 Suspension Forces and Moments on the Truck frames

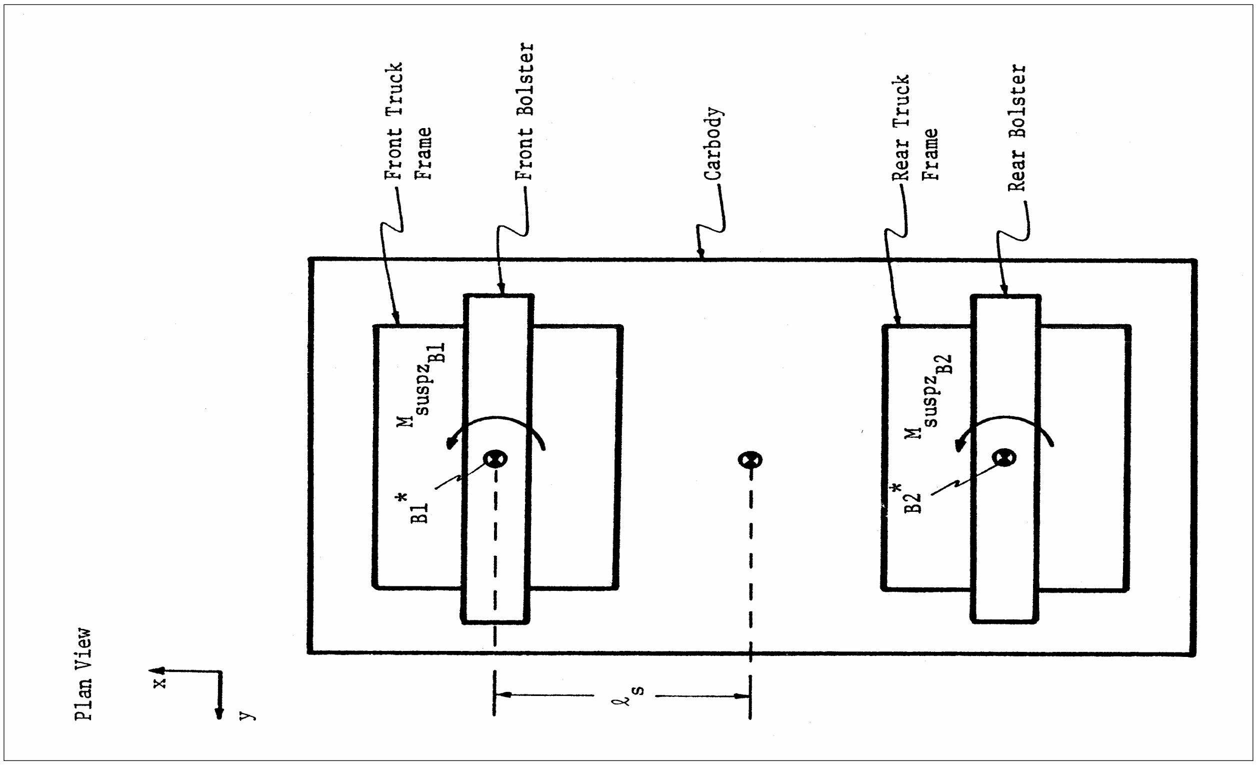


Figure 4-5 Suspension Forces and Moments on the Bolsters

4.1.2 Carbody Suspension Forces and Moments

Suspension forces act on the carbody due to its coupling with the front and the rear trucks. The trucks transmit motion to the carbody through secondary lateral, vertical, and yaw springs and dampers. Figure 4-6 shows the suspension forces and moments acting on the carbody.

The suspension forces act on the carbody in the lateral direction. The suspension moments act in the yaw as well as the roll directions. These forces and moments are given by the following equations.

Suspension force in the lateral direction due to the front truck and the rear truck:

$$F_{SUSPYC1} = 2K_{SY}y_{F1} - 2K_{SY}y_C + 2C_{SY}\dot{y}_{F1} - 2C_{SY}\dot{y}_C - 2h_{cs}K_{SY}\phi_C - 2h_{cs}C_{SY}\dot{\phi}_C \tag{4.8}$$

$$F_{SUSPYC2} = 2K_{SY}y_{F2} - 2K_{SY}y_C + 2C_{SY}\dot{y}_{F2} - 2C_{SY}\dot{y}_C - 2h_{cs}K_{SY}\phi_C - 2h_{cs}C_{SY}\dot{\phi}_C \tag{4.9}$$

Suspension moment in the yaw direction due to the front and the rear truck:

$$\begin{aligned} M_{SUSPZC1} &= 2l_sK_{SY}y_{F1} + K_{S\psi}\psi_{B1} + (-2l_s^2K_{SY} - K_{S\psi})\psi_C + 2l_sC_{SY}\dot{y}_{F1} + C_{S\psi}\dot{\psi}_{B1} \\ &+ (-2l_s^2C_{SY} - C_{S\psi})\dot{\psi}_C \end{aligned} \tag{4.10}$$

$$\begin{aligned} M_{SUSPZC2} &= -2l_sK_{SY}y_{F2} + K_{S\psi}\psi_{B2} + (-2l_s^2K_{SY} - K_{S\psi})\psi_C - 2l_sC_{SY}\dot{y}_{F2} + C_{S\psi}\dot{\psi}_{B2} \\ &+ (-2l_s^2C_{SY} - C_{S\psi})\dot{\psi}_C \end{aligned} \tag{4.11}$$

Suspension moment in the roll direction due to the front and the rear truck:

$$M_{SUSPXC1} = F_{SUSPYC1}h_{cs} - 2K_{SZ}d_s^2\phi_C - 2C_{SZ}d_s^2\dot{\phi}_C \tag{4.12}$$

$$M_{SUSPXC2} = F_{SUSPYC2}h_{cs} - 2K_{SZ}d_s^2\phi_C - 2C_{SZ}d_s^2\dot{\phi}_C \tag{4.13}$$

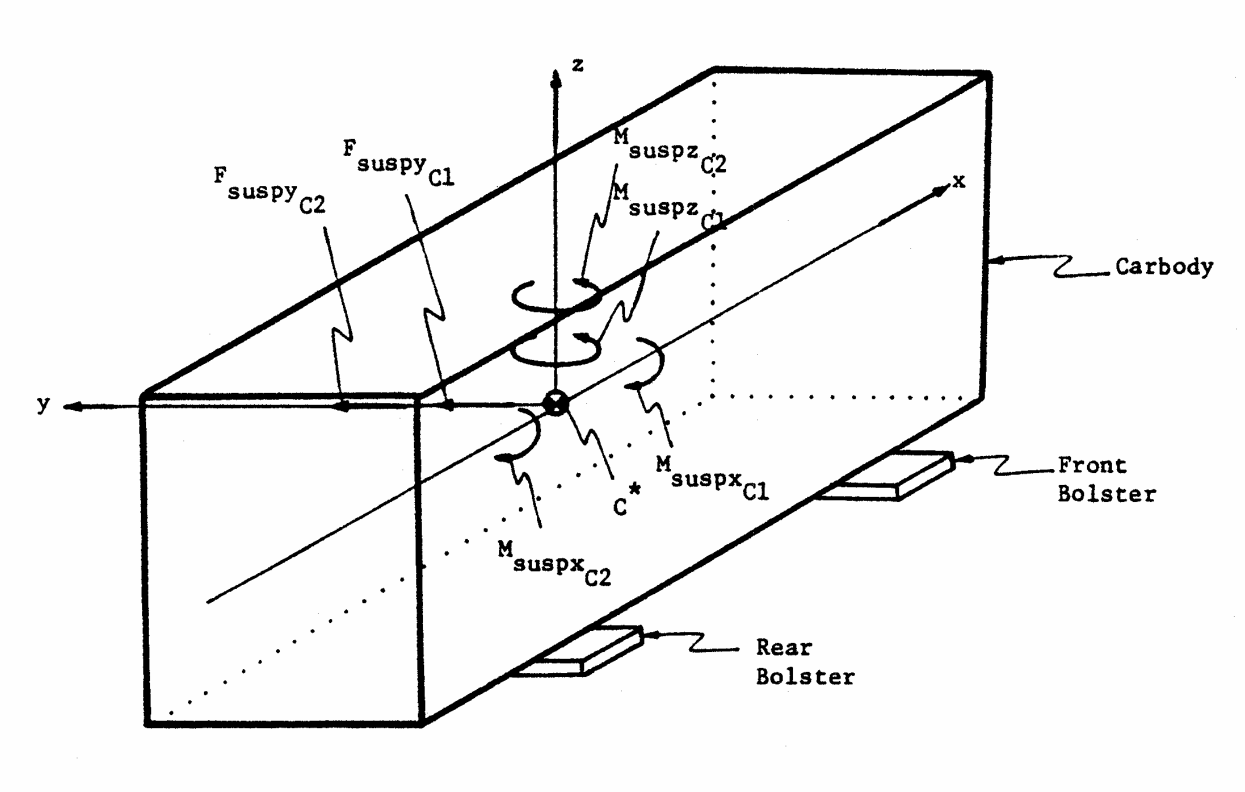


Figure 4-6 Suspension Forces and Moments on the Carbody

4.1.3 Carbody Dynamic Equations

This section presents the dynamic equations motion for the carbody. The equations of motion for a wheelset were discussed in Chapter 2 under Sections 2.1.3.3 and 2.1.3.4. The equations of motion for a single truck and bolster were covered in Chapter 3 under Section 3.1.3.

The lateral equation is obtained by applying the principle of linear momentum in the lateral direction [1]. Taking into account the lateral component of carbody weight and assuming small angles, the lateral equation of motion is given as:

$$\ddot{y}_C = \frac{1}{m_C}(-m_C g \phi_C + F_{SUSPYC1} + F_{SUSPYC2}) \tag{4.14}$$

The carbody yaw equation is obtained by applying the principle of angular momentum in the yaw direction [1]. Assuming small angles, the carbody yaw equation of motion is given as:

$$\ddot{\psi}_C = \frac{1}{I_{CZ}}(M_{SUSPZC1} + M_{SUSPZC2}) \tag{4.15}$$

The carbody roll equation is obtained by applying the principle of angular momentum in the roll direction [1]. Assuming small angles, the carbody roll equation of motion is given as:

$$\ddot{\phi}_C = \frac{1}{I_{CX}}(M_{SUSPXC1} + M_{SUSPXC2}) \tag{4.16}$$

4.2 Numerical Simulation

The full vehicle model consititutes of the following:

Carbody model: Section 4.1

2 single truck + bolster models: Section 3.1

4 single wheelset models: Section 2.1

This model was simulated using MATLAB [2] in order to obtain the time-domain solution of the dynamics of a rail vehicle moving on a flexible tangent track. This section presents the layout of the simulation program that was used to obtain the rail vehicle dynamic response.

The simulations were carried out by choosing the forward speed of the vehicle as the bifurcation parameter. The critical forward speed was obtained by increasing the forward speed gradually until the response of the vehicle became marginally stable. Sensitivity of the critical velocity to suspension parameters was studied.

Simplifications were made in order to make the memory requirements less and speed up the computation. In a wheelset dynamic analysis, the lateral dynamics are very important since they determine whether or not flanging occurs. The lateral dynamics are essentially decoupled from the vertical and the longitudinal dynamics. Hence, this simulation neglects the vertical and the longitudinal dynamics of the wheelset. This assumption eliminates two degrees of freedom for each wheelset and greatly reduces computation time.

It is also assumed that the effective lateral mass of the rail, m_{RAIL}, is zero. This is justified since the rail lateral stiffness and viscous damping forces dominate. Further, it is assumed that the influence of lateral rail velocity on lateral creepage is negligible. According to British Rail [5], this assumption is reasonable since the lateral creep force is generally saturated during flange contact.

The maximum adhesion force was assumed as constant rather than calculating the creep force saturation value at each time step iteratively. This iterative process involves solving the creep and the normal force equations simultaneously, which tends to be computationally very intense. From past experience, values of creep forces were found to be considerably less than the adhesion limit.

The creep coefficients were taken to be the same for both the tread and the flange contact patches. In reality, the flange contact patch will have smaller values, but it was found that this assumption makes insignificant difference to the resulting value of the critical forward speed and the wheelset response.

Initial conditions were assumed for the lateral and yaw position and velocity of the wheelsets, the trucks, and the carbody. The initial lateral displacement and velocity of the left and the right rails were assumed to be zero. The initial conditions assumed for simulation can be found in the MATLAB program files that are included in the Appendix.

A time step of 0.001 secs was manually chosen for solving the dynamic rail vehicle equations in MATLAB. The time step chosen automatically by MATLAB resulted in lack of memory space for the desired simulation time. Figure 4-7 shows the wheelset lateral response for the program chosen time step, as well as a manually chosen time step of 0.001 secs. It is seen that the two solutions are nearly identical.

The following time-domain solutions were obtained through simulation:

1. The lateral displacement and velocity of the wheelset

2. The yaw displacement and velocity of the wheelset

3. The lateral displacement of the left and the right rails

4. The lateral displacement and velocity of the truck

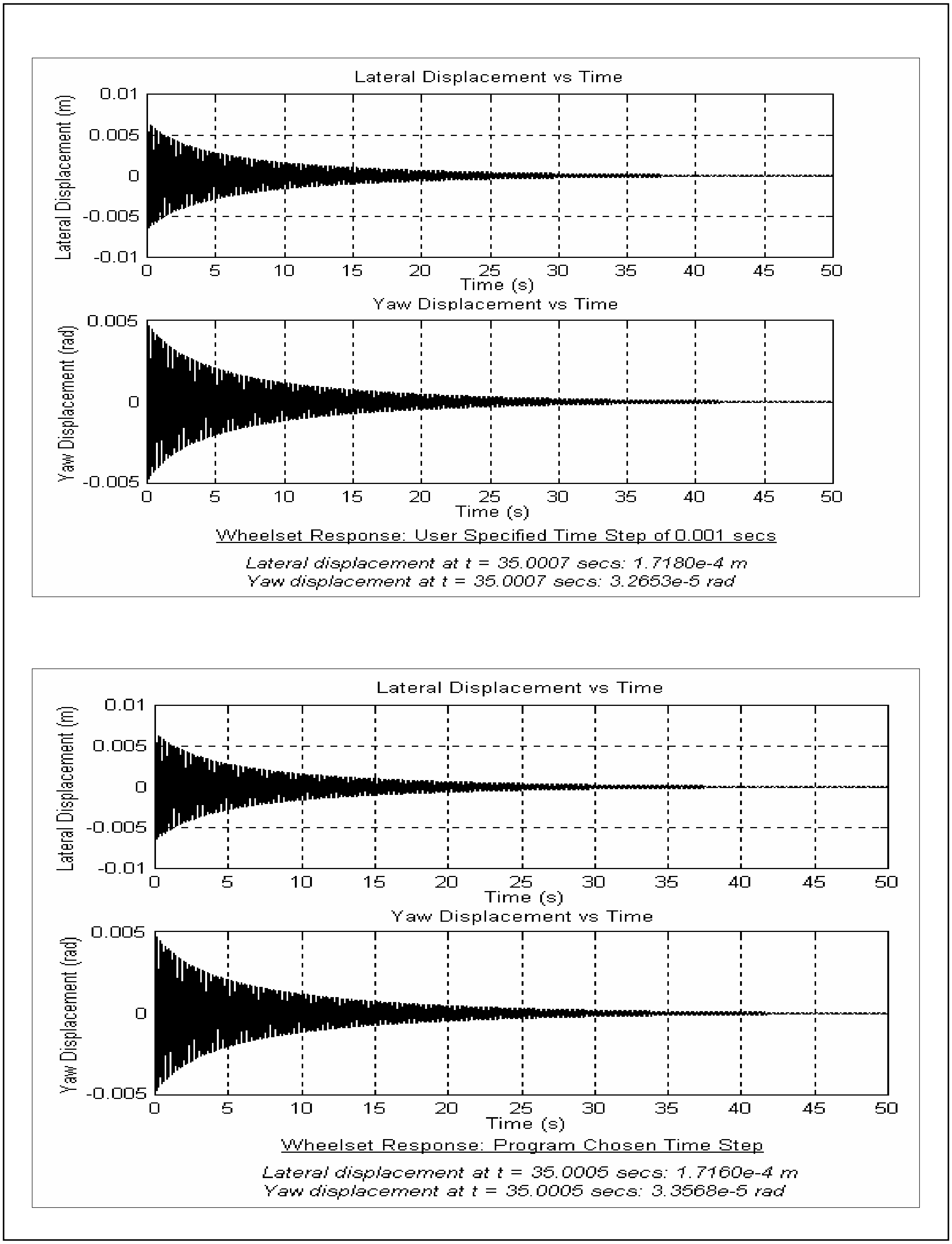
5. The yaw displacement and velocity of the truck and bolster

6. The lateral displacement and velocity of the carbody

7. The yaw displacement and velocity of the carbody

8. The roll displacement and velocity of the carbody

The parametric values used for simulation are shown in Table 4-1.



Wheelset Response: User Specified Time Step of 0.001 secs

Lateral displacement at t = 35.0007 secs: 1.7180e-4 m
Yaw displacement at t = 35.0007 secs: 3.2653e-5 rad

Wheelset Response: Program Chosen Time Step

Lateral displacement at t = 35.0005 secs: 1.7160e-4 m
Yaw displacement at t = 35.0005 secs: 3.3568e-5 rad

Figure 4-7 Wheelset Response Comparison - Automatic and Manual Time Step

Table 4-1 Rail Vehicle Simulation Constants

Parameter	Description	Value
Wheel Type	Wheel Type	AAR 1 in 20
Truck Type	Truck Type	Conventional
	Wheel / Rail Constants	
f_{11T}	Creep Coefficient (Tread)	9.43e6 N
f_{12T}	Creep Coefficient (Tread)	1.20e3 N.m
f_{22T}	Creep Coefficient (Tread)	1.00e3 N.m^2
f_{33T}	Creep Coefficient (Tread)	10.23e6 N
f_{11F}	Creep Coefficient (Flange)	9.43e6 N
f_{12F}	Creep Coefficient (Flange)	1.20e3 N.m
f_{22F}	Creep Coefficient (Flange)	1.00e3 N.m^2
f_{33F}	Creep Coefficient (Flange)	10.23e6 N
λ	Conicity	0.125
μ	Coefficient of friction	0.15
	Geometric Dimensions	
R_0	Centered rolling radius of wheel	0.3556 m
a	Half of track gage	0.716 m
b	Half of wheel base	1.295 m
d_p	Half of lateral spacing between primary longitudinal springs	0.61 m
d_s	Half of lateral spacing between secondary vertical springs	1.13 m
h_{cs}	Vertical distance from secondary lateral suspension to carbody center of mass	0.88 m
l_s	Longitudinal distance between carbody center of mass and truck center of mass	7.239 m

Parameter	Description	Value
Weights and Moments of Inertia		
m_W	Mass of wheelset	1751 kg
I_{WY}	Pitch mass moment of inertia of wheelset	130 kg.m^2
I_{WZ}	Yaw mass moment of inertia of wheelset	761 kg.m^2
m_F	Mass of truck frame	4041 kg
m_C	Mass of carbody	31900 kg
m_B	Mass of bolster	365 kg
I_{FZ}	Yaw mass moment of inertia of truck	3371 kg.m^2
I_{BZ}	Yaw mass moment of inertia of bolster	337 kg.m^2
I_{CZ}	Yaw mass moment of inertia of carbody	2414000 kg.m^2
I_{CX}	Roll mass moment of inertia of carbody	118500 kg.m^2
Suspension Stiffness and Damping		
K_{PX}	Primary longitudinal stiffness	Various
C_{PX}	Primary longitudinal damping	Various
K_{PY}	Primary lateral stiffness	Various
C_{PY}	Primary lateral damping	Various
K_{SY}	Secondary lateral stiffness	3.5E+05 N/m
C_{SY}	Secondary lateral damping	1.75E+04 N.s/m
K_{SZ}	Secondary vertical stiffness	3.8E+05 N/m
C_{SZ}	Secondary vertical damping	2.5E+04 N.s/m
$K_{S\psi}$	Secondary torsional stiffness	3.8E+08 N.m/rad
$C_{S\psi}$	Secondary torsional damping	2.5E+07 N.m.s/rad
T_0	Centerplate breakaway torque	10168 N.m
C_0	Coulomb viscous yaw damping	3.5E+07 N.s/rad

The general algorithm used for wheelset analysis was described in Section 2.2 of Chapter 2. The single-point or two-point contact equations are solved at each time step depending on which out of the above conditions is satisfied. The single-point and two-point contact equations are solved by MATLAB using a fourth order Runge-Kutta integration algorithm. This method requires the equations of motion to be transformed to a system of 42 first-order differential equations (also known as state-space equations). In order to achieve this transformation, the actual variables have been converted to first-order state-space variables as shown below:

$x_1 = y_{W1}$

$x_2 = \psi_{W1}$

$x_3 = \dot{y}_{W1}$

$x_4 = \dot{\psi}_{W1}$

$x_5 = y_{RAIL,L1}$

$x_6 = y_{RAIL,R1}$

$x_7 = y_{W2}$

$x_8 = \psi_{W2}$

$x_9 = \dot{y}_{W2}$

$x_{10} = \dot{\psi}_{W2}$

$x_{11} = y_{RAIL,L2}$

$x_{12} = y_{RAIL,R2}$

$x_{13} = y_{F1}$

$x_{14} = \psi_{F1}$

$x_{15} = \dot{y}_{F1}$

$x_{16} = \dot{\psi}_{F1}$

$x_{17} = \psi_{B1}$

$x_{18} = \dot{\psi}_{B1}$

$x_{19} = y_{W3}$

$$x_{20} = \psi_{W3}$$

$$x_{21} = \dot{y}_{W3}$$

$$x_{22} = \dot{\psi}_{W3}$$

$$x_{23} = y_{RAIL,L3}$$

$$x_{24} = y_{RAIL,R3}$$

$$x_{25} = y_{W4}$$

$$x_{26} = \psi_{W4}$$

$$x_{27} = \dot{y}_{W4}$$

$$x_{28} = \dot{\psi}_{W4}$$

$$x_{29} = y_{RAIL,L4}$$

$$x_{30} = y_{RAIL,R4}$$

$$x_{31} = y_{F2}$$

$$x_{32} = \psi_{F2}$$

$$x_{33} = \dot{y}_{F2}$$

$$x_{34} = \dot{\psi}_{F2}$$

$$x_{35} = \psi_{B2}$$

$$x_{36} = \dot{\psi}_{B2}$$

$$x_{37} = y_{C}$$

$$x_{38} = \psi_{C}$$

$$x_{39} = \phi_{C}$$

$$x_{40} = \dot{y}_{C}$$

$$x_{41} = \dot{\psi}_{C}$$

$$x_{42} = \dot{\phi}_{C}$$

(4.17)

The equations of motion can now be written in state-space form as follows:

$\dot{x}_1 = x_3$ (from above)

$\dot{x}_2 = x_4$ (from above)

$\dot{x}_3 = f_1(x_1, x_2, ---, x_{41}, x_{42})$ (front truck leading wheelset lateral equation)

$\dot{x}_4 = f_2(x_1, x_2, ---, x_{41}, x_{42})$ (front truck leading wheelset yaw equation)

$\dot{x}_5 = f_3(x_1, x_2, ---, x_{41}, x_{42})$ (front truck leading left rail lateral equation)

$\dot{x}_6 = f_4(x_1, x_2, ---, x_{41}, x_{42})$ (front truck leading right rail lateral equation)

$\dot{x}_7 = x_9$ (from above)

$\dot{x}_8 = x_{10}$ (from above)

$\dot{x}_9 = f_5(x_1, x_2, ---, x_{41}, x_{42})$ (front truck trailing wheelset lateral equation)

$\dot{x}_{10} = f_6(x_1, x_2, ---, x_{41}, x_{42})$ (front truck trailing wheelset yaw equation)

$\dot{x}_{11} = f_7(x_1, x_2, ---, x_{41}, x_{42})$ (front truck trailing left rail lateral equation)

$\dot{x}_{12} = f_8(x_1, x_2, ---, x_{41}, x_{42})$ (front truck trailing right rail lateral equation)

$\dot{x}_{13} = x_{15}$ (from above)

$\dot{x}_{14} = x_{16}$ (from above)

$\dot{x}_{15} = f_9(x_1, x_2, ---, x_{41}, x_{42})$ (front truck lateral equation)

$\dot{x}_{16} = f_{10}(x_1, x_2, ---, x_{41}, x_{42})$ (front truck yaw equation)

$\dot{x}_{17} = x_{18}$ (from above)

$\dot{x}_{18} = f_{11}(x_1, x_2, ---, x_{41}, x_{42})$ (front bolster yaw equation)

$\dot{x}_{19} = x_{21}$ (from above)

$\dot{x}_{20} = x_{22}$ (from above)

$\dot{x}_{21} = f_{12}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck leading wheelset lateral equation)

$\dot{x}_{22} = f_{13}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck leading wheelset yaw equation)

$\dot{x}_{23} = f_{14}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck leading left rail lateral equation)

$\dot{x}_{24} = f_{15}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck leading right rail lateral equation)

$\dot{x}_{25} = x_{27}$ (from above)

$\dot{x}_{26} = x_{28}$ (from above)

$\dot{x}_{27} = f_{16}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck trailing wheelset lateral equation)

$\dot{x}_{28} = f_{17}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck trailing wheelset yaw equation)

$\dot{x}_{29} = f_{18}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck trailing left rail lateral equation)

$\dot{x}_{30} = f_{19}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck trailing right rail lateral equation)

$\dot{x}_{31} = x_{33}$ (from above)

$\dot{x}_{32} = x_{34}$ (from above)

$\dot{x}_{33} = f_{20}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck lateral equation)

$\dot{x}_{34} = f_{21}(x_1, x_2, ---, x_{41}, x_{42})$ (rear truck yaw equation)

$\dot{x}_{35} = x_{36}$ (from above)

$\dot{x}_{36} = f_{22}(x_1, x_2, ---, x_{41}, x_{42})$ (rear bolster yaw equation)

$$\begin{aligned}
&\dot{x}_{37} = x_{40} &&\text{(from above)}\\
&\dot{x}_{38} = x_{41} &&\text{(from above)}\\
&\dot{x}_{39} = x_{42} &&\text{(from above)}\\
&\dot{x}_{40} = f_{23}(x_1, x_2, ---, x_{41}, x_{42}) &&\text{(carbody lateral equation)}\\
&\dot{x}_{41} = f_{24}(x_1, x_2, ---, x_{41}, x_{42}) &&\text{(carbody yaw equation)}\\
&\dot{x}_{42} = f_{25}(x_1, x_2, ---, x_{41}, x_{42}) &&\text{(carbody roll equation)}
\end{aligned} \tag{4.18}$$

The MATLAB program used to simulate the dynamic behavior of the full vehicle and the functions used by the program are described below in Table 4-2. Figure 4-8 depicts the layout and interaction between the different functions at any time-step.

Table 4-2   Rail Vehicle Simulation Program and Functions

Name of Program / Function	Description
FULL_VEHICLE	Main program. This contains the initial conditions, the global variables, and instructions for plotting the time-responses. This program also contains the instruction and conditions to solve the ordinary differential equations contained in the function files WHEELSET, TRUCK, and CARBODY
EQUATIONS	This function obtains variables from functions WHEELSET_SUSPENSION, TRUCK_SUSPENSION, and CARBODY_SUSPENSION and uses them to solve the single wheelset, truck, and carbody equations by invoking functions WHEELSET, TRUCK, and CARBODY
WHEELSET	Contains the single wheelset differential equations for the front and the rear wheelsets. This function reads the rolling radius and contact angle from functions ROLLING_RADIUS and CONTACT_ANGLE respectively, and the wheelset suspension forces and moments from EQUATIONS. This function also reads the creep forces from ONEPT_CREEP and TWOPT_CREEP (depending on single or two-point contact) and the normal forces from ONEPT_NORMAL or TWOPT_NORMAL (depending on single or two-point contact)
TRUCK	Contains the single truck differential equations. This function reads the truck suspension forces and moments from EQUATIONS.
CARBODY	Contains the carbody differential equations. This function reads the carbody suspension forces and moments from EQUATIONS.

Name of Program / Function	Description
ONEPT_CREEP	Reads the wheelset state variables at any time-step and calculates the single-point creep forces and moments using the Vermeulen-Johnson approach with creep force saturation. The output is then passed on to the function WHEELSET
TWOPT_CREEP	Reads the wheelset state variables at any time-step and calculates the two-point creep forces and moments using the Vermeulen-Johnson approach with creep force saturation. The output is then passed on to the function WHEELSET.
ONEPT_NORMAL	Reads the wheelset state variables at any time-step and calculates the single-point normal forces and moments. The output is then passed on to the function WHEELSET
TWOPT_NORMAL	Reads the wheelset state variables at any time-step and calculates the two-point normal forces and moments. The output is then passed on to the function WHEELSET
WHEELSET_ SUSPENSION	Reads the wheelset and truck state variables at any time-step and calculates the suspension forces and moments acting on the wheelset. The output is then passed on to the function EQUATIONS
TRUCK_ SUSPENSION	Reads the wheelset and truck state variables at any time-step and calculates the suspension forces and moments acting on the truck frame. The output is then passed on to the function EQUATIONS
CARBODY_ SUSPENSION	Reads the truck and carbody state variables at any time-step and calculates the suspension forces and moments acting on the carbody. The output is then passed on to the function EQUATIONS

Name of Program / Function	Description
ROLLING_RADIUS	Reads the wheelset and rail lateral excursion at any time-step and calculates the rolling radius at tread and/or flange contact patches on the left and the right wheels. The output is then passed on to the function WHEELSET.
CONTACT_ANGLE	Reads the wheelset and rail lateral excursion at any time-step and calculates the contact angle at tread and/or flange contact patches on the left and the right wheels. The output is then passed on to the function WHEELSET.

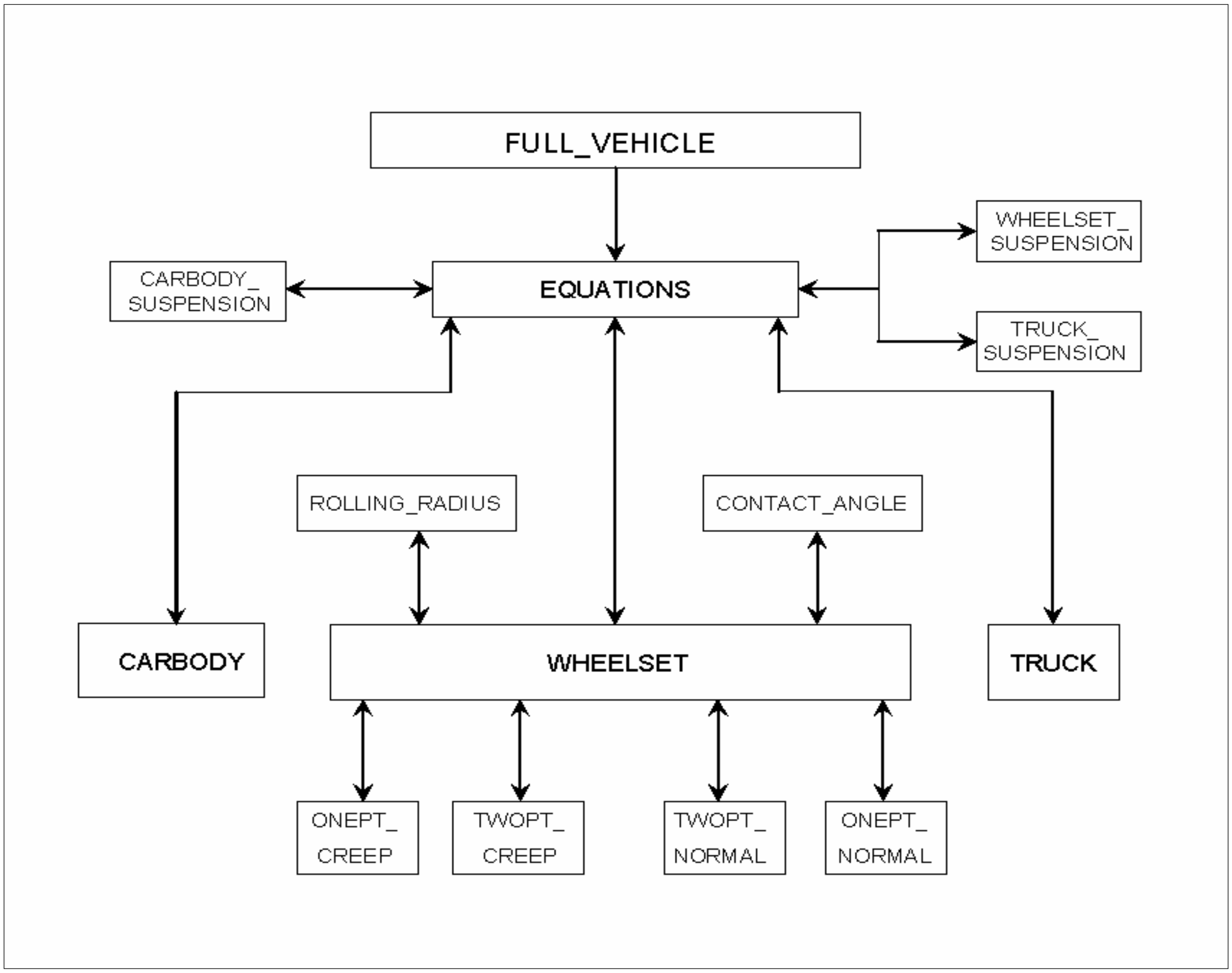


Figure 4-8 Rail Vehicle Simulation Program Layout

4.3 Simulation Results

The simulation essentially involved the determination of the critical velocity under different constraints imposed on the equations enumerated in Section 4.1. The data enumerated in Table 4-1 were used except when the effect of variation of a given parameter was being investigated.

4.3.1 The Effect of Primary Spring Stiffness on the Critical Velocity

The critical velocity of the rail vehicle with AAR wheels was determined for various values of primary spring longitudinal and lateral stiffness by varying the vehicle forward velocity.

The initial conditions used for the simulations in this section are:

Lateral displacement of front leading wheelset = 0.00635 m,
Lateral velocity of front leading wheelset = 0.01 m/s,
Yaw displacement of front leading wheelset = 0.001 rad,
Yaw velocity of front leading wheelset = 0.0 rad/s,
Lateral displacement of front trailing wheelset = 0.003 m,
Lateral velocity of front trailing wheelset = 0.01 m/s,
Yaw displacement of front trailing wheelset = 0.001 rad,
Yaw velocity of front trailing wheelset = 0.0 rad/s,
Lateral displacement of front truck frame = 0.0 m,
Lateral velocity of front truck frame = 0.0 m/s,
Yaw displacement of front truck frame = 0.0 rad,
Yaw velocity of front truck frame = 0.0 rad/s,
Yaw displacement of front bolster = 0.0 rad,
Yaw velocity of front bolster = 0.0 rad/s,
Lateral displacement of rear leading wheelset = 0.00635 m,
Lateral velocity of rear leading wheelset = 0.01 m/s,
Yaw displacement of rear leading wheelset = 0.001 rad,
Yaw velocity of rear leading wheelset = 0.0 rad/s,
Lateral displacement of rear trailing wheelset = 0.003 m,
Lateral velocity of rear trailing wheelset = 0.01 m/s,
Yaw displacement of rear trailing wheelset = 0.001 rad,

Yaw velocity of rear trailing wheelset = 0.0 rad/s,

Lateral displacement of rear truck frame = 0.0 m,

Lateral velocity of rear truck frame = 0.0 m/s,

Yaw displacement of rear truck frame = 0.0 rad,

Yaw velocity of rear truck frame = 0.0 rad/s,

Yaw displacement of rear bolster = 0.0 rad,

Yaw velocity of rear bolster = 0.0 rad/s,

Lateral displacement of carbody = 0.0 m,

Lateral velocity of carbody = 0.0 m/s,

Yaw displacement of carbody = 0.0 rad,

Yaw velocity of carbody = 0.0 rad/s,

Roll displacement of carbody = 0.0 rad,

Roll velocity of carbody = 0.0 rad/s,

Left and right rail displacements = 0.0 m.

Figures 4-9, 4-10, and 4-11 depict the front truck, the rear truck, and the carbody time-response respectively at a vehicle velocity of 20 m/s for K_{PX} = 2.85e4 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 8376.9 N-s/m, and C_{PY} = 9048.2 N-s/m. The critical velocity for this configuration is 12 m/s. At this speed, it is seen that all four of the wheelsets are flanging for a majority of the simulation time. The lateral motion of the wheelsets is limited by the flange width on both the left and the right wheels. The front and the rear trucks exhibit lateral hunting motion as well with an amplitude over 0.003 m. The carbody is seen to exhibit lateral hunting motion with an amplitude of around 0.0015 m. From the carbody response in Figure 4-11, it is evident that a second mode of oscillation with a considerably lower frequency coexists with the higher frequency primary mode.

Figures 4-12, 4-13, and 4-14 depict the front truck, the rear truck, and the carbody time-response respectively at a vehicle velocity of 18 m/s for K_{PX} = 2.85e5 N/m, K_{PY} = 5.84e5 N/m, C_{PX} = 8376.9 N-s/m, and C_{PY} = 9048.2 N-s/m. The critical velocity for this configuration is 21 m/s. At this speed, the response is slightly damped and approaches the origin, which serves as the stable equilibrium point for the oscillations.

The sensitivity of critical velocity to primary spring stiffness is tabulated below in Table 4-3. The values of K_{PY} and K_{PX} are varied in geometric progression, with a common ratio of 3.14. This leads to every alternate value being a multiple or a sub-multiple by a factor of 10. Figures 4-15 and 4-16 plot the variation of critical velocity with longitudinal and lateral spring stiffness.

From Table 4-3 and Figures 4-15 and 4-16, it is seen that the critical velocity increases with K_{PX} for the range of K_{PY} that was considered. Furthermore, it is seen that the variation of the critical velocity with K_{PY} is almost negligible at low values of K_{PX}. As the value of K_{PX} is increased to higher values, the critical velocity increases marginally at low values of K_{PY} and then linearly drops with increasing K_{PY}.

In conclusion, the critical velocity variation of a complete rail vehicle behaves very similar to that of a single truck, but different from that of a single wheelset. In a free wheelset, higher longitudinal and lateral spring stiffness always result in higher critical velocities. In a truck, higher longitudinal spring stiffness results in higher critical velocities. However, increase in the lateral spring stiffness results in roughly constant, if not decreasing critical velocities.

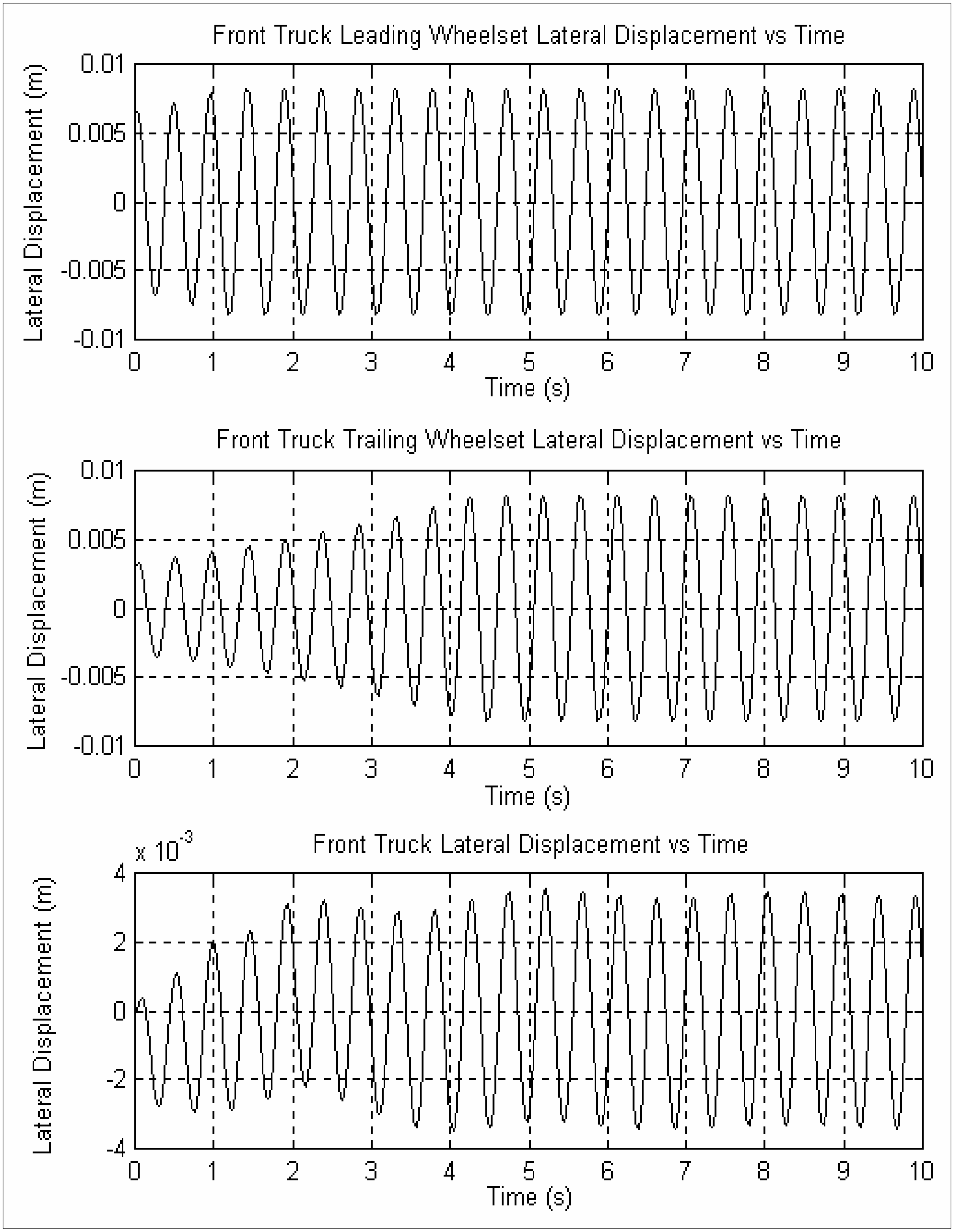


Figure 4-9 Front Truck Response: K_{PX} = 2.85e4 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (>V_C) = 20 m/s

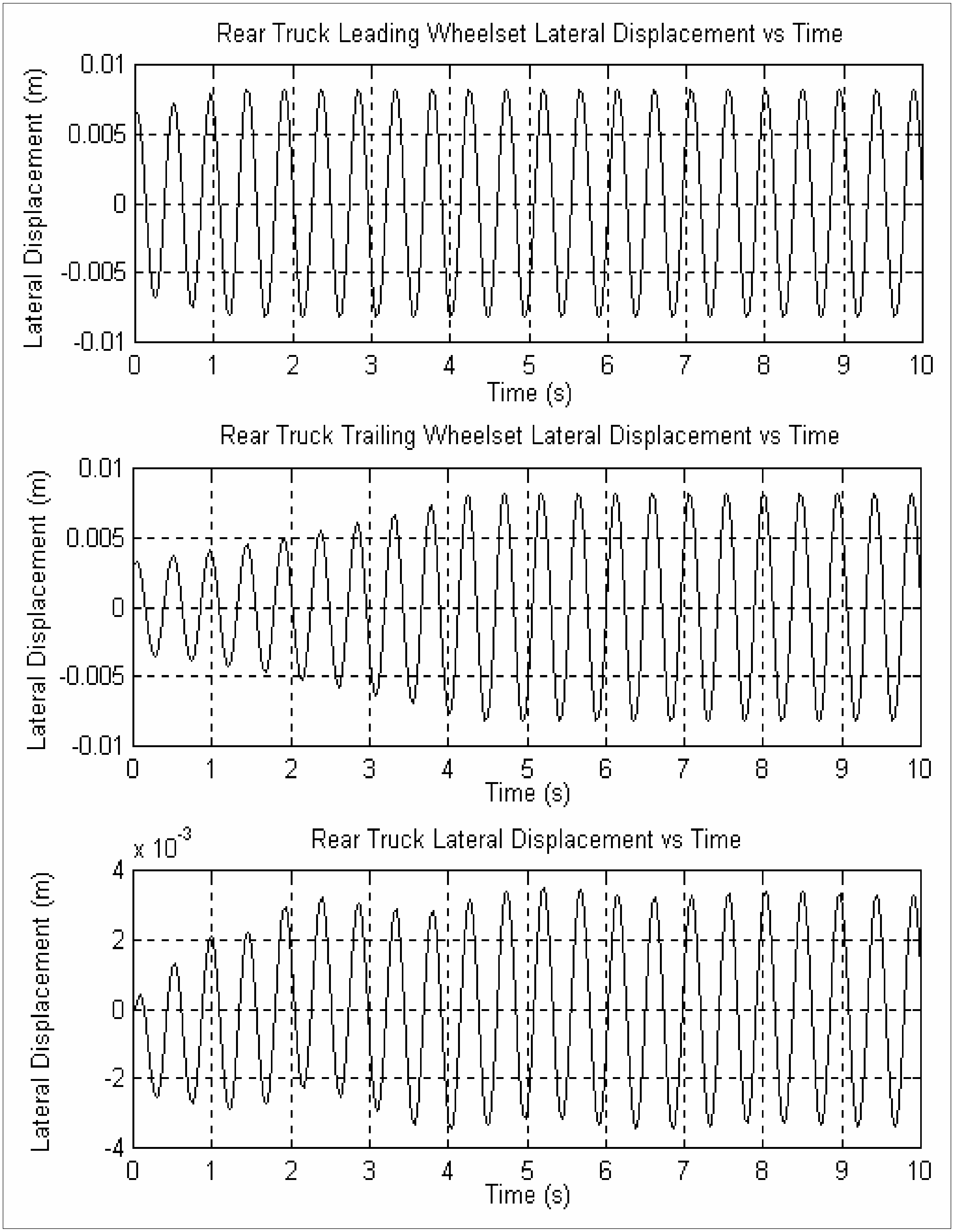


Figure 4-10 Rear Truck Response: K_{PX} = 2.85e4 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (>V_C) = 20 m/s

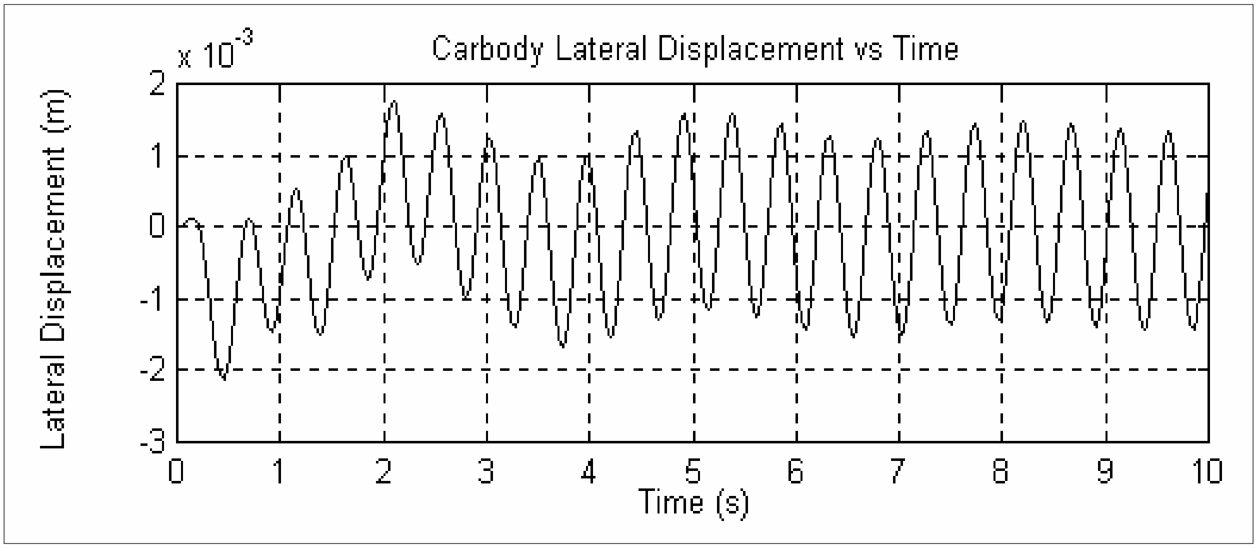


Figure 4-11 Carbody Response: K_{PX} = 2.85e4 N/m, K_{PY} = 5.84e4 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (>V_C) = 20 m/s

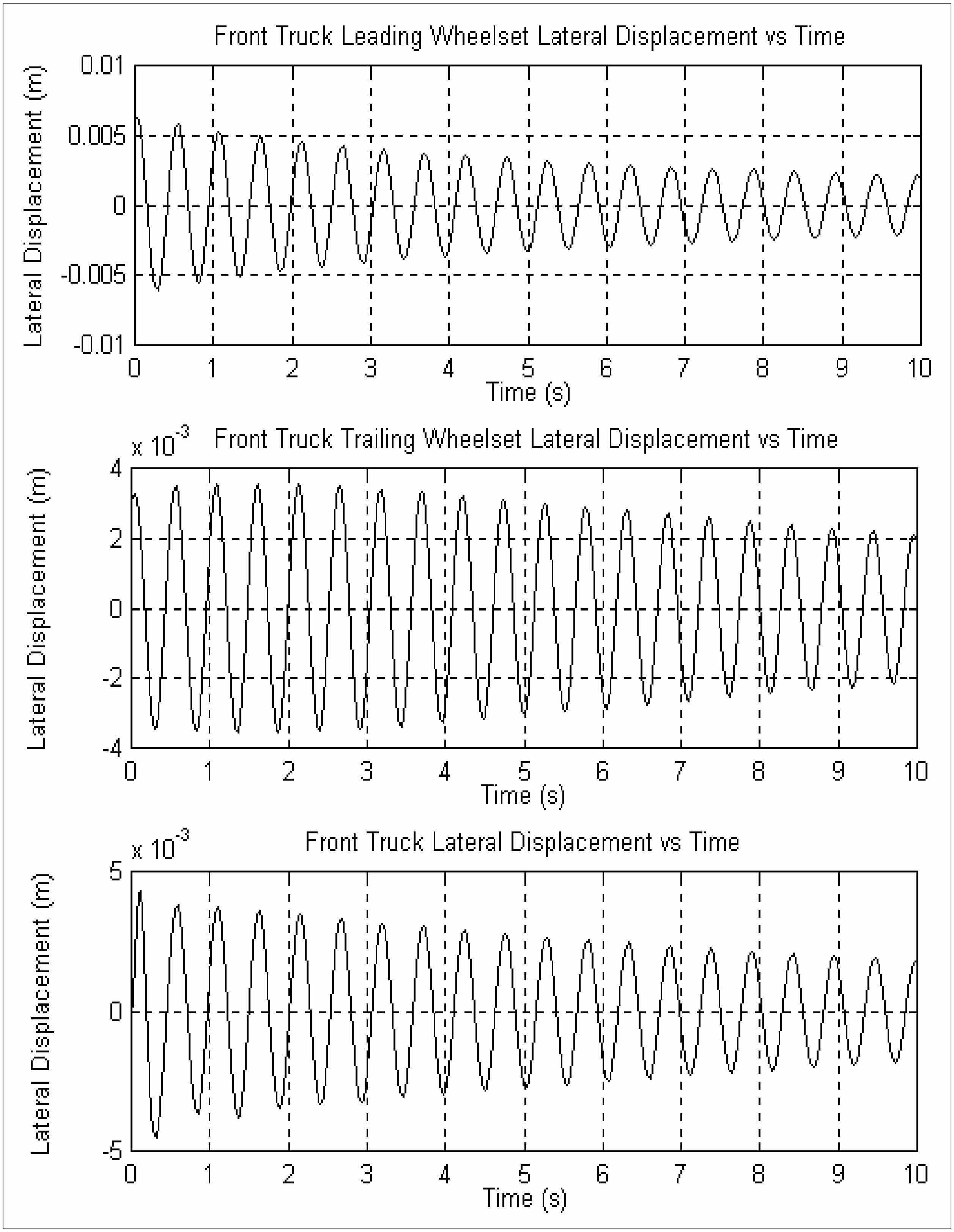


Figure 4-12 Front Truck Response: K_{PX} = 2.85e5 N/m, K_{PY} = 5.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (<V_C) = 18 m/s

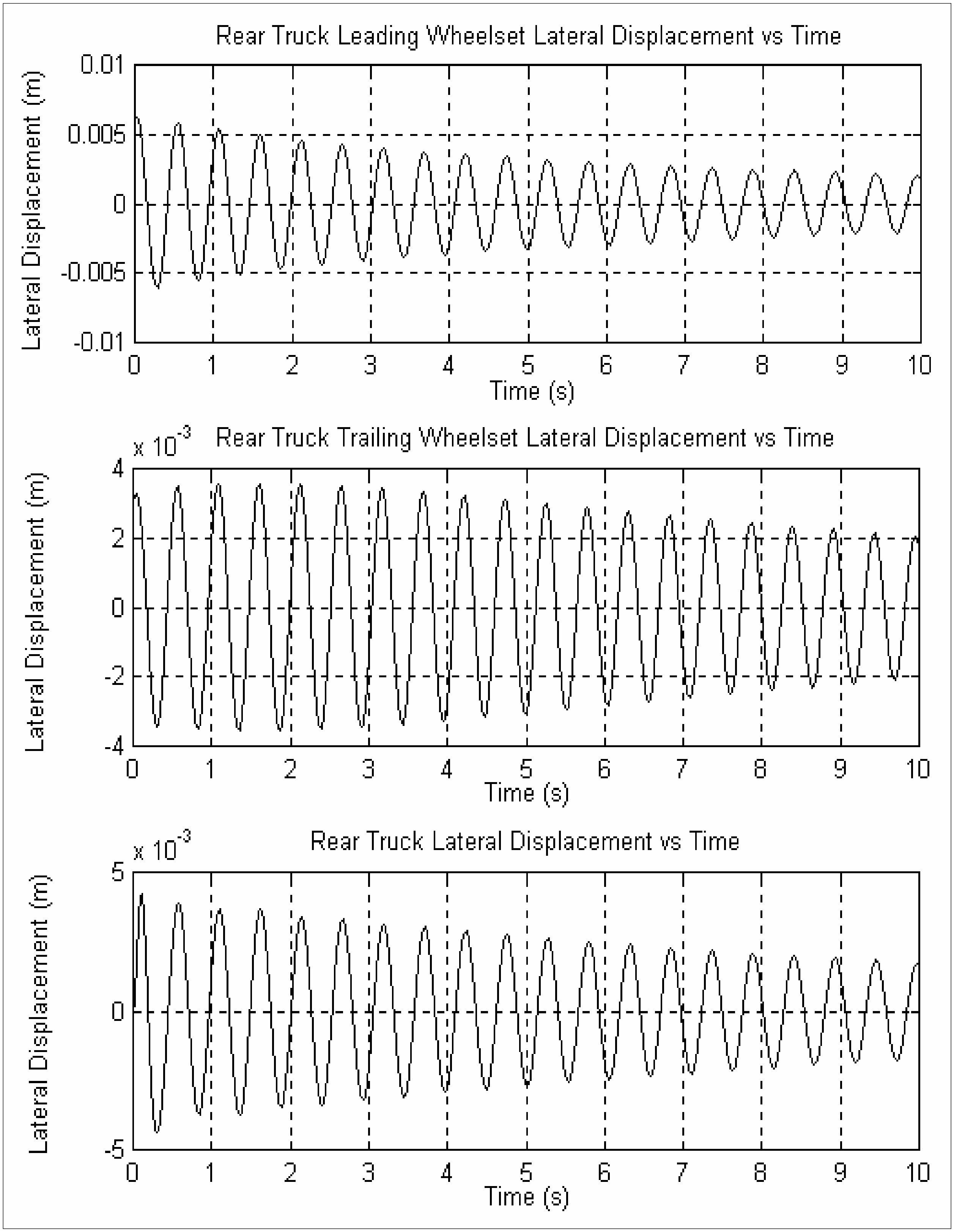


Figure 4-13 Rear Truck Response: K_{PX} = 2.85e5 N/m, K_{PY} = 5.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (<V_C) = 18 m/s

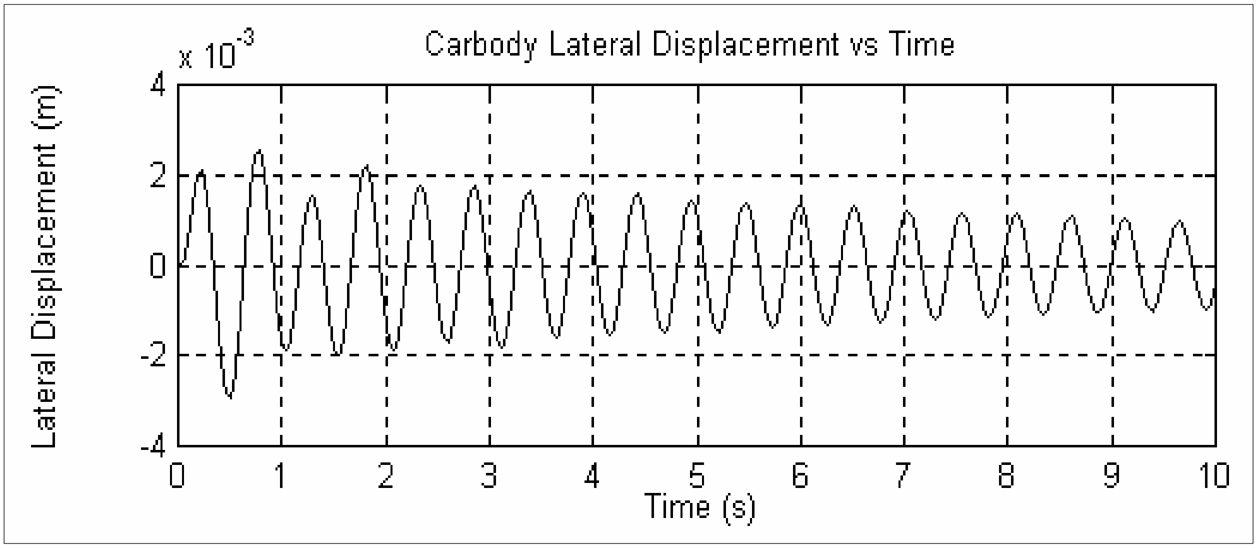


Figure 4-14 Carbody Response: K_{PX} = 2.85e5 N/m, K_{PY} = 5.84e5 N/m, C_{PX} = 8376.9 N-s/m, C_{PY} = 9048.2 N-s/m, Velocity (<V_C) = 18 m/s

Table 4-3 Sensitivity of Rail Vehicle Critical Velocity to Primary Longitudinal and Lateral Spring Stiffness

K_{PX} (N/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Rail Vehicle (V_C), m/s (km/hr)			
2.85e4	12 (43)	15 (54)	15 (54)	16 (58)
9.12e4	14 (50)	16 (58)	17 (61)	18 (65)
2.85e5	20 (72)	21 (76)	21 (76)	20 (72)
9.12e5	33 (119)	35 (126)	32 (115)	28 (101)
2.85e6	61 (220)	65 (234)	71 (256)	52 (187)

Note: All values in Table 4-3 above have been obtained with C_{PY} = 9048.2 N-s/m and C_{PX} = 8376.9 N-s/m.

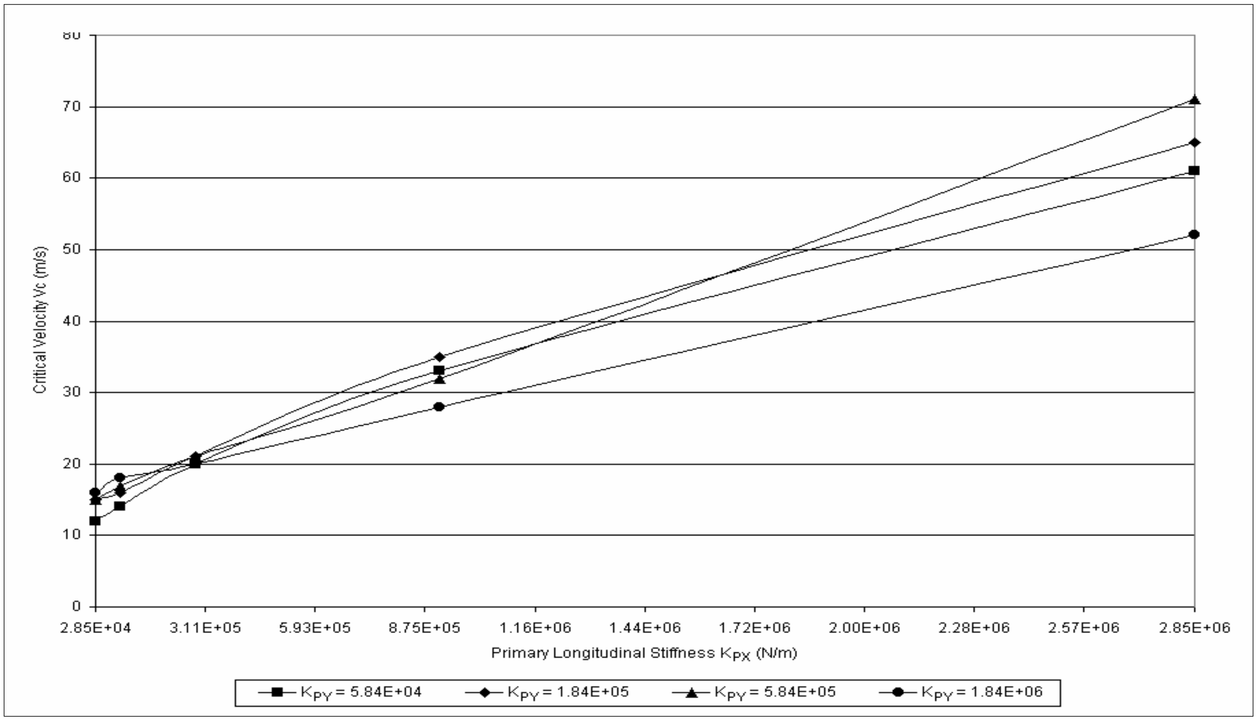


Figure 4-15 Variation of Rail Vehicle Critical Velocity (V_C) with Primary Longitudinal Spring Stiffness (K_{PX})

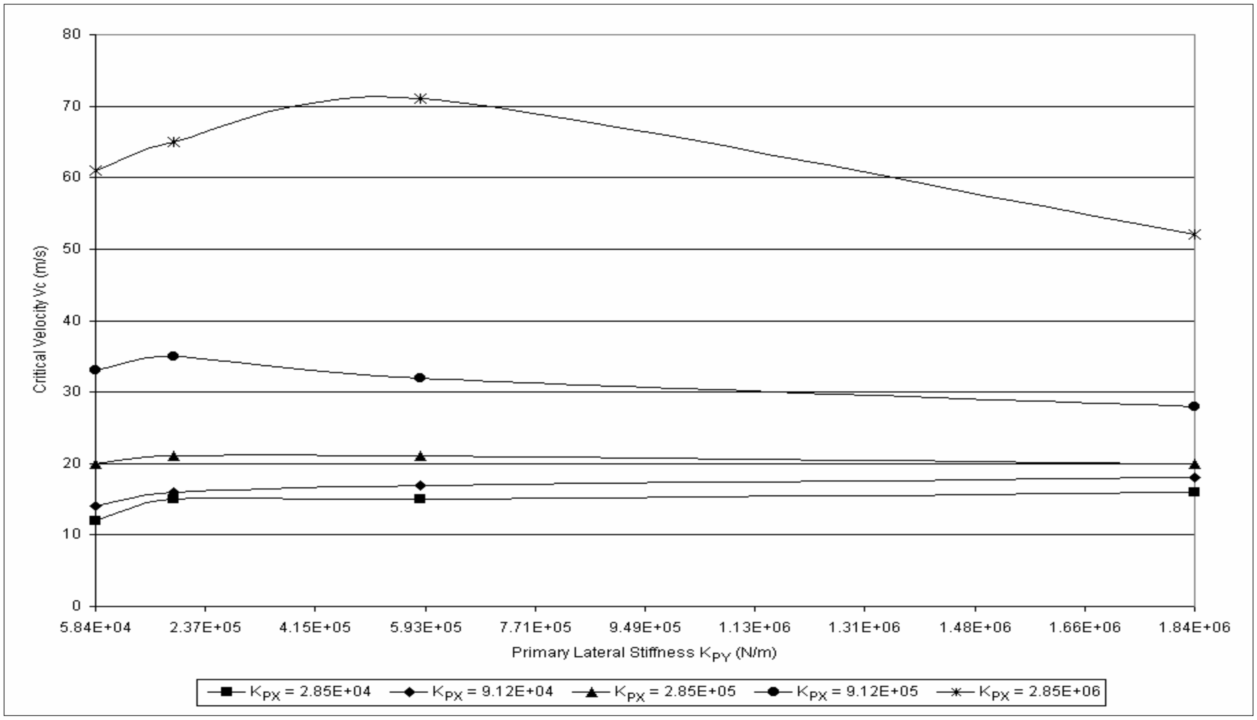


Figure 4-16 Variation of Full Vehicle Critical Velocity (V_C) with Primary Lateral Spring Stiffness (K_{PY})

4.3.2 The Effect of Other Suspension Parameters on the Critical Velocity

It was concluded from Chapter 3, Sections 3.3.2, 3.3.3, and 3.3.4 that the effect of primary damping and secondary lateral stiffness on the critical velocity of a single truck model was negligible. The Coulomb friction breakaway torque had some impact on the single truck critical velocity, but this was small as compared to the effect of primary lateral and longitudinal stiffness.

A similar conclusion can be drawn for the critical velocity of a complete rail vehicle. The large mass and yaw moment of inertia of the carbody, coupled with low values of the secondary suspension elements, will ensure that the carbody has negligible lateral and angular accelerations, or as if the carbody is practically at rest.

4.4 Conclusions

This chapter provided the mathematical formulation for a complete rail vehicle model. The forces and moments that act on the front and the rear truck and bolster, and the carbody, and which govern the lateral and yaw motions of the front and rear truck and bolster, and the carbody were obtained and the equations which define the dynamics of the rail vehicle were enumerated. Both single-point and two-point wheel / rail contact conditions were considered.

Table 4-3 and Figures 4-15 and 4-16 depict the sensitivity of the complete rail vehicle critical velocity to variations in the primary longitudinal and lateral spring stiffness. The simulations indicate a supercritical Hopf bifurcation phenomenon, where the origin loses its stability after the critical velocity is reached, and gives rise to a nearly elliptical limit cycle about the origin. The results also indicate that the critical velocity increases with K_{PX} for the range of K_{PY} that was considered. Furthermore, it is seen that the variation of the critical velocity with K_{PY} is almost negligible at low values of K_{PX}. As the value of K_{PX} is increased to higher values, the critical velocity increases marginally at low values of K_{PY} and then linearly drops with increasing K_{PY}. In conclusion, the critical velocity variation of a rail vehicle behaves similarly to that of a single truck, but differently as compared to that of a single free wheelset. In a free wheelset, higher longitudinal and lateral spring stiffness always result in higher critical velocities. In a truck, higher longitudinal spring stiffness results in higher critical velocities.

However, increase in the lateral spring stiffness results in roughly constant, if not decreasing critical velocities.

It was concluded from Chapter 3, Sections 3.3.2, 3.3.3, and 3.3.4 that the effect of primary damping and secondary lateral stiffness on the critical velocity of a single truck model was negligible. The Coulomb friction breakaway torque had some impact on the single truck critical velocity, but this was small as compared to the effect of primary lateral and longitudinal stiffness.

A similar conclusion can be drawn for the critical velocity of a complete rail vehicle. The large mass and yaw moment of inertia of the carbody, coupled with low values of the secondary suspension elements, will ensure that the carbody has negligible lateral and angular accelerations, or as if the carbody is practically at rest.

Chapter 5

Improving Hunting Behavior

Chapters 2, 3, and 4 demonstrated the hunting behavior in single wheelsets, simple trucks, and the complete rail vehicle model. Hunting can lead to severe ride discomfort and eventual physical damage to wheels and rails. As seen in the previous chapters, hunting behavior in rail vehicles starts once the vehicle reaches a critical forward velocity. In order for the hunting phenomenon to be eliminated during rail vehicle operation, this critical velocity has to be higher than the maximum speeds at which rail vehicles operate. This chapter focuses on improving hunting behavior in rail vehicles by increasing the critical velocity of hunting beyond the operational speed range. While there may be several ways to achieve an increase in critical hunting velocities, this thesis attempts to reach this objective by semi-active control of the suspension elements.

5.1 Summary of Parametric Variation

The effect of variation of suspension parameters on the critical velocity was simulated for a single wheelset, a single truck, and the complete rail vehicle, and the results are listed in the tables presented in Chapters 2, 3 and 4 respectively. In particular, the effects of varying K_{PX}, K_{PY}, C_{PX}, C_{PY}, K_{SY}, and T_0 were studied. The results of varying these parameters are summarized below.

5.1.1 Single Wheelset Model

- Increasing the values of K_{PX} and K_{PY} to larger values results in high critical velocities. Thus when K_{PY} is 1.84e6 N/m, the critical velocities are in excess of 55 m/s (123 miles/hour). Similarly if K_{PX} is 2.85e6 N/m, the critical velocity are in excess of 62 m/s (139 miles/hour).
- Increasing the values of C_{PX} and C_{PY} to larger values also results in higher critical velocities. But, the increase in critical velocities obtained by changing the primary damping is

considerably less than that obtained by increasing the primary stiffness values. The increase in critical velocity is higher for higher values of primary stiffness. The highest critical velocity obtained for the range of primary damping and stiffness studied is 93 m/s (208 miles/hour). This occurs at C_{PX} = 41880 N-s/m.

5.1.2 Single Truck Model

When a single truck is considered, with the carbody connected to the truck by the secondary suspension system, and the carbody is rigidly fixed in lateral, yaw, and roll directions, the critical velocities obtained are not as high as in the case of a single wheelset. Even if the primary spring constants are increased to very high values, the presence of the secondary suspension system ensures that the critical velocities do not increase inordinately.

- At low and moderate values of K_{PX} (9.12e4 N/m or lower), even if K_{PY} is varied from 5.84e4 N/m to 1.84e6 N/m, the critical velocity hardly varies, increasing only by 3 m/s. The critical velocity obtained with this moderate and desirable values of K_{PX} is only as high as 15 m/s (34 miles/hour).

- At high values of K_{PX} (higher than 9.12e4 N/m), the critical velocity initially increases with K_{PY}, reaches a maximum, and decreases thereafter. For K_{PX} = 2.85e6 N/m, the critical velocity reaches a maximum of 67 m/s (150 miles/hour) for K_{PY} = 5.84e5 N/m, after which it falls rather sharply.

- For all values of K_{PY} considered, increasing K_{PX} significantly increases the critical velocity. The highest increase in critical velocity (increase of 54 m/s, or 121 miles/hour) is obtained with K_{PY} = 5.84e5 N/m, with K_{PX} increased from 2.85e4 N/m to 2.85e6 N/m. At this value of K_{PX} the critical velocity obtained is as high as 67 m/s (150 miles/hour).

- Increasing the primary longitudinal and lateral damping only serves to maintain the critical velocity at the same value, if not even lower it.

- Increasing the secondary lateral stiffness, K_{SY}, from 7e4 N/m to 1.75e6 N/m results in an increase in critical velocity at lower values of K_{PX}. For K_{PX} = 2.85e5 N/m, increasing K_{SY} results in an increase in critical velocity of as much as 14 m/s (31 miles/hour). However, at higher values of K_{PX} (9.12e5 N/m or higher) increasing K_{SY} results in roughly constant, or even a decrease in critical velocities.

- At high values of K_{PX} (around 2.85e6 N/m) and high values of C_{PX} (around 41880 N-s/m), an increase in the Coulomb damper breakaway torque from 10168 N-m to 20336 N-m results in as much as a 21 m/s (47 miles/hour) increase in the critical velocity. A maximum critical velocity of 79 m/s (177 miles/hour) is achieved with K_{PX} = 2.85e6 N/m, C_{PX} = 41880 N-s/m, and K_{PY} = 5.84e5 N/m.

5.1.3 Full Vehicle Model

The critical speeds obtained for the full vehicle are similar to that obtained for a single truck. Table 5-1 below shows the sensitivity of the complete rail vehicle critical velocity to primary longitudinal and lateral spring stiffness. This information is the same as that shown in Table 4-3 of Chapter 4, and is re-shown here for convenience.

- At low and moderate values of K_{PX} (9.12e4 N/m or lower), varying K_{PY} from 5.84e4 to 1.84e6 results in an increase in critical velocity of only as much as 4 m/s. The critical velocity obtained with these moderate and desirable values of K_{PX} is only as high as 18 m/s (40 miles/hour). At higher values of K_{PX}, the same variation in K_{PY} serves merely to maintain the same critical velocity, if not even lower it.

- For all values of K_{PY} considered, increasing K_{PX} significantly increases the critical velocity. The highest increase in critical velocity (increase of 56 m/s, or 125 miles/hour) is obtained with K_{PY} = 5.84e5 N/m, with K_{PX} increased from 2.85e4 N/m to 2.85e6 N/m. At this value of K_{PX} the critical velocity obtained is as high as 71 m/s (159 miles/hour).

- The sensitivity of critical velocity to all other suspension parameters is expected to be very similar to that noted for the single truck case.

From the conclusions presented in Sections 5.1.1, 5.1.2, and 5.1.3 above, it can be noted that the most efficient method of increasing the critical velocity of the rail vehicle is by raising the primary longitudinal spring stiffness, K_{PX}.

Table 5-1 Sensitivity of Rail Vehicle Critical Velocity to Primary Longitudinal and Lateral Spring Stiffness

K_{PX} (N/m)	K_{PY} (N/m)			
	5.84e4	1.84e5	5.84e5	1.84e6
	Critical Velocity of Rail Vehicle (V_C), m/s (km/hr)			
2.85e4	12 (43)	15 (54)	15 (54)	16 (58)
9.12e4	14 (50)	16 (58)	17 (61)	18 (65)
2.85e5	20 (72)	21 (76)	21 (76)	20 (72)
9.12e5	33 (119)	35 (126)	32 (115)	28 (101)
2.85e6	61 (220)	65 (234)	71 (256)	52 (187)

5.2 Semi-Active Suspension Control

The results of the parametric variation of suspension elements presented in Section 5.1 demonstrate that the value of the primary lateral stiffness K_{PX} has to be high in order to achieve high critical velocities. But, having high values of K_{PX} is largely undesirable, since it makes the wheelset suspension very rigid. Any disturbance to the wheels due to imperfections in the rails will result in a forcing function in the equations of motion, thereby facilitating the transfer of the recurrent wheelset oscillations to the carbody, leading to a poorer ride quality.

As an alternative to using sustained high values of K_{PX} throughout the ride, a semi-active approach has been attempted, whereby, a nominal value of K_{PX} was assumed for the majority of the simulation time. On a need basis, this value was made to increase for limited portions of the oscillatory cycle.

5.2.1 Semi-Active Control using Wheelset Lateral Displacement

As a starting point, the value of K_{PX} is made to be a function of the wheelset lateral excursion. K_{PX} is taken at a moderate value of 2.85e5 N/m, and K_{PY} is fixed at a moderate value of 5.84e5 N/m. From Table 5-1, this configuration results in a critical velocity of 21 m/s (47 miles/hour). As the speed of the vehicle is increased, the excursion of the wheelset increases, and so does the yaw. When the displacement of the wheelset exceeds a certain amount, the value of K_{PX} can be made to increase suddenly to a high value. Since a high value of critical velocity is only available for values of K_{PX} around or greater than 2.85e6 N/m, the longitudinal spring constant can be made to increase to 2.85e6 N/m whenever the lateral excursion exceeds a preset threshold.

The preset threshold of lateral excursion is initially set at 5 mm, which is slightly greater than half the flange clearance. This semi-active longitudinal suspension control condition can be written as follows:

$$K_{PX} = \begin{cases} 2.85\text{e}5 \text{ N/m} \ldots\ldots\ldots\ldots \text{ for } \left|y_W\right| \leq 0.005 \\ \\ 2.85\text{e}6 \text{ N/m} \ldots\ldots\ldots\ldots \text{ for } \left|y_W\right| > 0.005 \end{cases} \tag{5.1}$$

where $\left|y_W\right|$ stands for the absolute value of wheelset lateral excursion.

But simulation showed that this strategy barely improved the critical velocity. The resulting critical velocity was only slightly better than the original critical velocity of 21 m/s. Lowering the lateral excursion threshold (in order to increase the duration that the higher value of K_{PX} is in effect) did not result in any significant improvement either.

The reason for this lack of improvement appears to be as follows: By changing K_{PX}, we are in fact controlling the yaw motion. The restoring force due to K_{PX} is maximum when the yaw angle is maximum, since the longitudinal spring is deflected by a linear distance d_p x ψ. Increasing the value of K_{PX} when the lateral excursion is maximum may not be the ideal way of improving the restoring force, if at that time, the yaw, and consequently the deflection of the longitudinal springs, is not high. The lateral and yaw wheelset response shown in Chapter 2 illustrate that the lateral and the yaw response are out of phase by about 90 degrees, meaning that when the lateral displacement is maximum positive or negative, the yaw is near zero, and vice versa.

5.2.2 Semi-Active Control using Wheelset Yaw Displacement

In view of the discussion in Section 5.2.1 above, the conclusion was reached that the value of K_{PX} should be increased, not when the lateral excursion is maximum, but when it is minimum, or when the yaw is maximum. Since the extension of the longitudinal springs is proportional to the yaw angle, making the spring constant high when the lateral excursion is maximum will produce no effect, since under this condition, the extension of the longitudinal springs is near zero and the force produced by them is also near zero. Therefore making the longitudinal gain dependent on the yaw angle, or what is equivalent, on the extension of the longitudinal springs, should yield better results.

We now consider K_{PX} to be a function of the yaw displacement of the wheelset such that we have two levels of the longitudinal spring constant, one below a certain yaw angle threshold and the other beyond that threshold. Also, the actual yaw angle of the wheelset, ψ_W, cannot be measured. Hence, the relative yaw angle of the wheelset and the truck, $(\psi_W - \psi_F)$, is measured.

K_{PX} is taken at a moderate value of 2.85e5 N/m, and K_{PY} is fixed at a moderate value of 5.84e5 N/m. As the speed of the vehicle is increased, the excursion of the wheelset increases, and so does the yaw. Once the yaw threshold has been reached, the value of K_{PX} is increased to 2.85e6 N/m.

This semi-active longitudinal suspension control condition can be written as follows:

$$K_{PX} = \begin{cases} 2.85e5 \text{ N/m} \ldots\ldots\ldots\ldots \text{ for } \left|\psi_W - \psi_F\right| \le \psi_{THRESHOLD} \\ \\ 2.85e6 \text{ N/m} \ldots\ldots\ldots\ldots \text{ for } \left|\psi_W - \psi_F\right| > \psi_{THRESHOLD} \end{cases} \tag{5.2}$$

where $|\psi|$ stands for the absolute value of yaw excursion.

The resulting critical velocity depends on the value of the yaw threshold chosen. If the threshold is very low, then the higher value of K_{PX} is in effect for a longer duration, and so a high critical velocity is obtained. If the threshold is very high, then the higher value of K_{PX} is in effect for a short duration, and so a low critical velocity is obtained. Table 5-2 shows the resulting critical velocities for different yaw threshold values. Figure 5-1 and 5-2 show the front truck leading wheelset, the front truck, and the carbody response for K_{PX} = 2.85e5 N/m and K_{PY} = 5.84e5 N/m, and for a velocity of 35 m/s. The critical velocity for this configuration is 21 m/s. A yaw threshold of 0.003 rad was chosen. When the yaw diplacement exceeded 0.003 rad, the value of K_{PX} was increased 10 times to 2.85e6 N/m. From the yaw phase portrait in Figure 5-1, the yaw response due to the abrupt increase in the longitudinal stiffness is very evidently seen as a sharp cut back in the rising yaw trajectory and a reversal of the yaw back to the center.

Table 5-2 Critical Velocity Versus Yaw Threshold

Yaw Threshold, rad	Critical Velocity (V_C), m/s (km/hr)
0.000	73 (263)
0.001	72 (259)
0.002	71 (256)
0.003	70 (252)
0.004	67 (241)
0.005	31 (112)
> 0.005	22 (79)

A similar methodology was tried out by increasing the primary lateral stiffness K_{PY} by a factor of 10 whenever the yaw threshold was exceeded. It was found that unless K_{PX} is kept at a very high level, there is no great improvement due to varying K_{PY}. This is evident from the fact that mainly K_{PX} decides the critical velocity. For, instance if K_{PX} is kept at a moderate value, say at 2.85e5 N/m, a change in K_{PY} from 1.84e5 N/m to 1.84e6 N/m hardly results in a change in the critical velocity of 21 m/s.

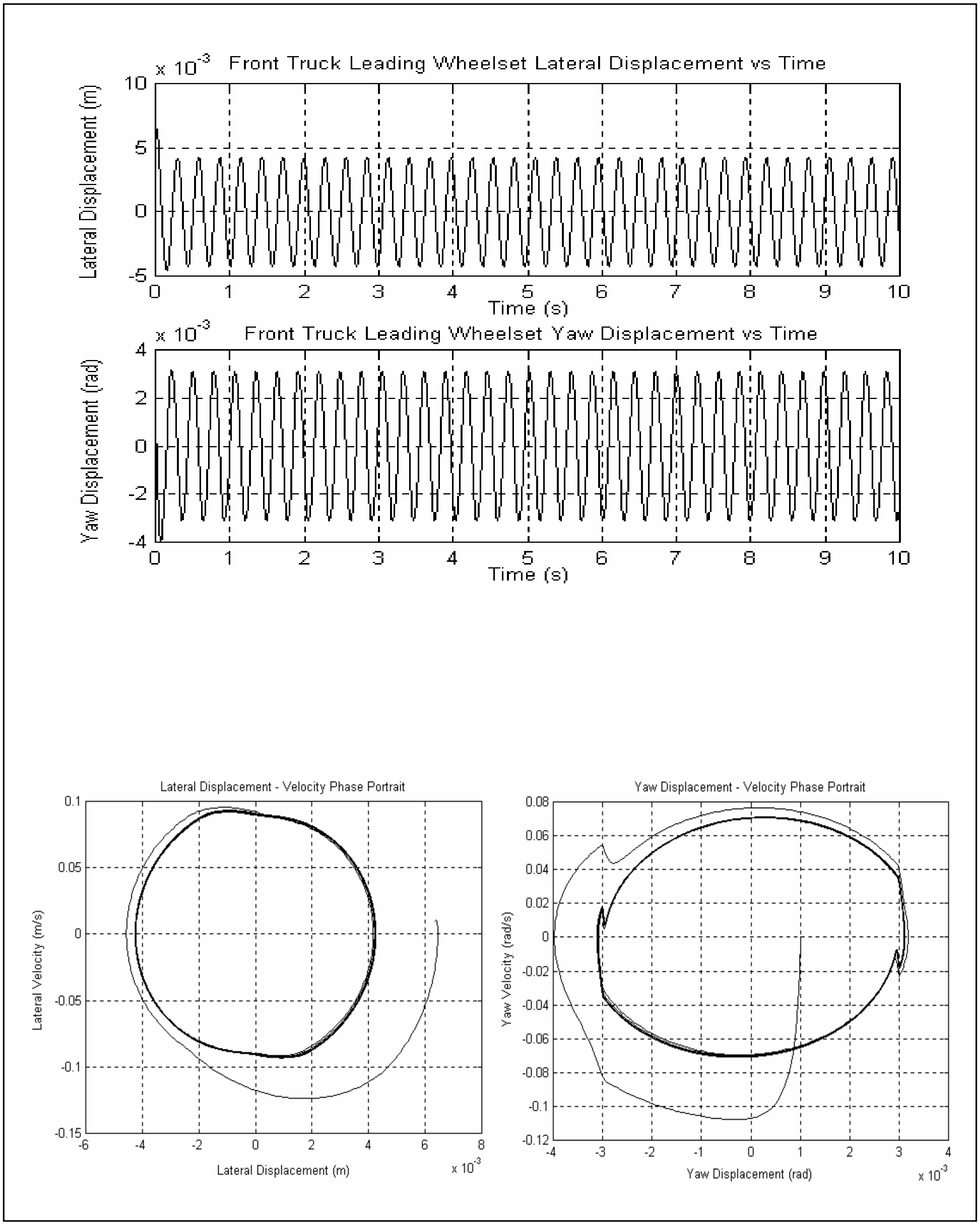


Figure 5-1 Front Truck Leading Wheelset Response for 0.003 rad Yaw Threshold

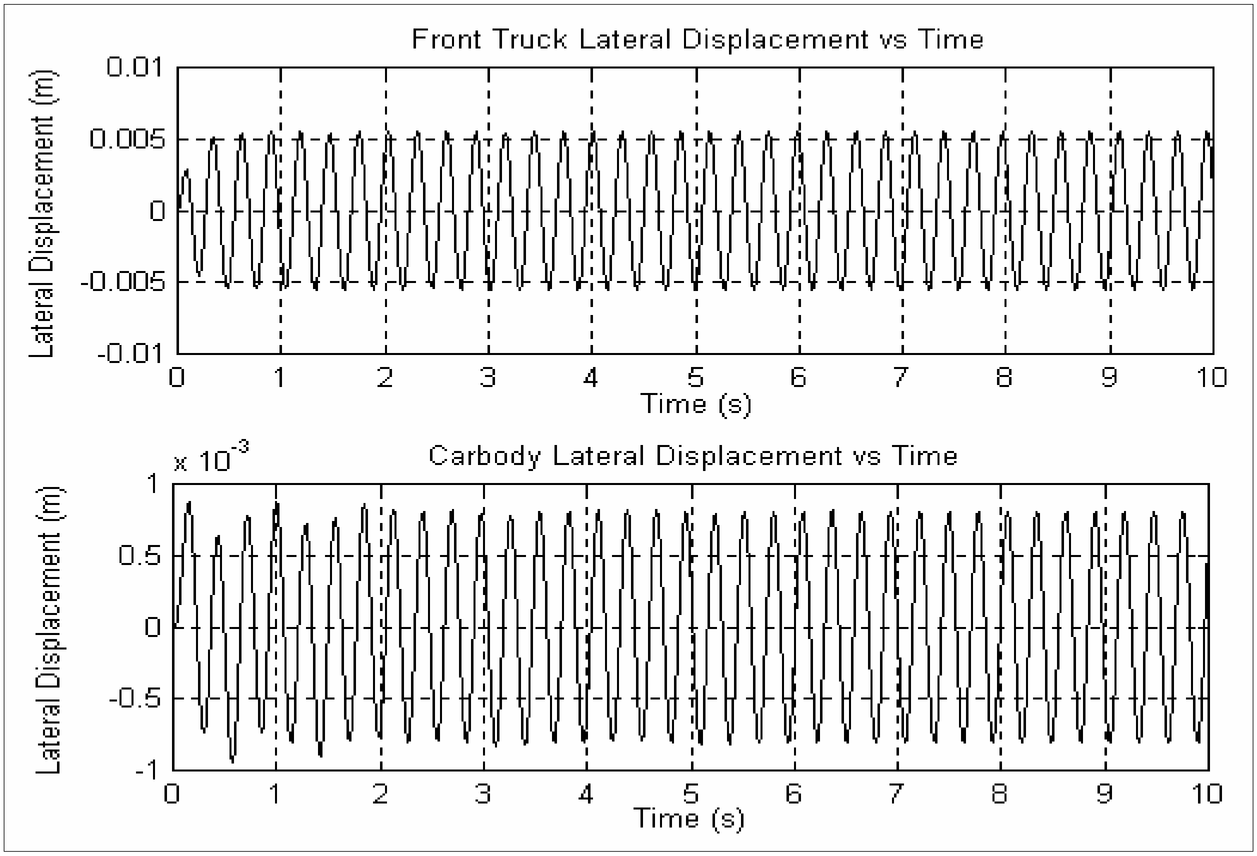


Figure 5-2 Front Truck and Carbody Response for 0.003 rad Yaw Threshold

5.3 Conclusions

This chapter discussed a methodology of improving hunting behavior in rail vehicles by semi-active control of suspension elements. This methodology focuses on increasing the critical velocity of hunting beyond the operational speed range.

Chapters 2, 3, and 4 have shown that the critical velocity is most sensitive to the primary longitudinal stiffness, K_{PX}. A higher primary longitudinal stiffness can significantly increase the critical hunting velocity of the rail vehicle. But, having high values of K_{PX} is largely undesirable, since it makes the wheelset suspension very rigid. Any disturbance to the wheels due to imperfections in the rails will result in a forcing function in the equations of motion, thereby facilitating the transfer of the recurrent wheelset oscillations to the carbody, leading to a poorer ride quality. As an alternative to using sustained high values of K_{PX} throughout the ride, a semi-active approach has been attempted, whereby, a nominal value of K_{PX} was assumed for the majority of the simulation time. On a need basis, this value was made to increase for limited portions of the oscillatory cycle.

As an initial approach, the value of K_{PX} was made to be a function of the wheelset lateral excursion. A moderate value of K_{PX} was assumed for the duration of time that the wheelset lateral excursions were below a preset threshold value. Whenever the lateral displacement exceeded this threshold limit, the value of K_{PX} was increased to 10 times the original value. It was thought that this approach would raise the critical hunting velocity, and drive the oscillations down towards the center. But simulation showed that this strategy barely improved the critical velocity.

A subsequent approach was to make the value of K_{PX} a function of the wheelset yaw displacement instead. This approach yielded a significant improvement in the critical velocity, as is shown in Table 5-2.

5.4 Recommendations for future work

The objective of this work was to quantify the sensitivity of the critical velocity of hunting to various primary and secondary suspension parameters. The suspension parameter that most influences the critical velocity was identified, and a semi-active control strategy was developed in order to increase the critical velocity of hunting beyond the typical operational speed range of rail vehicles.

This work was intended to be a starting point for investigating control strategies in order to improve hunting behavior in rail vehicles. The following recommendations are provided for future work based on this thesis:

➢ The simulations in this thesis were performed using a PC with a Windows 98 environment. Due to computing limitations, the equations of motion governing the rail vehicle were limited to the lateral and yaw degrees of freedom. While the lateral and the yaw degrees of freedom are most significant in a lateral stability model, it might be worthwhile to investigate dynamic behavior of the rail vehicle system with the vertical and the longitudinal degrees of freedom included.

- The semi-active control strategy developed for increasing the critical velocity of hunting involves an abrupt and significant increase in the primary longitudinal stiffness. Future work needs to aim at the feasibility of such an increase in the longitudinal stiffness, and must also be directed towards arriving at an optimal method for effecting this increase.

- The semi-active control strategy developed in this thesis involves a 1-step ten-fold increase in the primary longitudinal stiffness. The high resulting stiffness may adversely affect the ride quality, and other performance parameters. A thorough investigation must be carried out to clearly quantify the effect of such high primary longitudinal stiffness on these parameters.

- Based on the investigation as mentioned above, an optimal path (one that least affects the critical performance parameters) may need to be developed in order to effect the desired increase in the longitudinal stiffness. This path may involve a multi-step approach instead of the 1-step approach mentioned in this thesis. An approach of this kind will ensure a more gradual increase in the stiffness, and possibly shorter time periods that higher stiffness coefficients are in effect.

References

1. Nagurka, M.L., “Curving Performance of Rail Passenger Vehicles,” Ph.D. Thesis, Department of Mechanical Engineering, M.I.T., 1983.
2. “The Student Edition of MATLAB,” Version 5 User’s Guide, Prentice Hall, 1998.
3. Ahmadian, M., and Yang, S., “Hopf Bifurcation and Hunting Behavior in a Rail Wheelset with Flange Contact,” Nonlinear Dynamics, Vol. 15, Issue 1, January 1998, pp. 15-30.
4. Horak, D., and Wormley, D.N., “Nonlinear Stability and Tracking of Rail Passenger Trucks,” Journal of Dynamic Systems, Measurement, and Control, Vol. 104, September 1982, pp. 256-263.
5. Clark, R.A., Eickhoff, B.M., and Hunt, G.A., “Prediction of the Dynamic Response of Vehicles to Lateral Track Irregularities,” presented at the 7th IAVSD-IUTAM symposium on the Dynamics of Vehicles on Roads and Tracks, Cambridge, U.K., September 7-11, 1981.
6. Kalker, J.J., “Survey of Wheel-Rail Rolling Contact Theory,” Vehicle System Dynamics, Vol. 5, 1979, pp. 317-358.
7. Vermeulen, P.J., and Johnson, K.L., “Contact of Nonspherical Elastic Bodies Transmitting Tangential Forces,” Journal of Applied Mechanics, Vol. 86, June 1964, pp. 338-340.
8. De Pater, A.D., “The Approximate Determination of the Hunting Movement of a Railway Vehicle by aid of the Method of Krylov and Bogoljubow,” in proceedings of the 10th International Congress of Applied Mechanics, Stresa, Appl. Sci. Res. A. 10, 1960.
9. Matsudaira, T., Paper awarded prize in the competition sponsored by the Office of Research and Experiment of the International Union of Railways, Utrecht, 1960.
10. Wickens, A.H., “The Dynamic Stability of Railway Vehicle Wheelsets and Bogies having Profiled Wheels,” International Journal of Solids and Structures, Vol. 1, Issue 3, July 1965, pp. 319-341.
11. Cooperrider, N.K., “The Hunting Behavior of Conventional Railway Trucks,” ASME Journal of Eng. Industry, Vol. 94, 1972, pp. 752-762.

12. Huilgol, R.R., “Hopf-Friedrichs Bifurcation and the Hunting of a Railway Axle,” Quart. J. Appl. Math., Vol. 36, pp. 85-94.

13. Kaas-Peterson, C., “Chaos in a Railway Bogie,” Acta Mech., Vol. 61, pp. 89-107.

14. True, H., “Dynamics of a Rolling Wheelset,” Applied Mechanics Reviews, Vol. 46, No. 7, July 1993, pp. 438-444.

15. True, H., “Chaotic Motion of Railway Vehicles,” presented at the 11th IAVSD symposium on Vehicle System Dynamics in the Dynamics of Vehicles on Roads and Tracks, Amsterdam/Lisse, 1989, pp. 578-587.

16. True, H., and Kaas-Peterson, C., “A Bifurcation Analysis of Nonlinear Oscillations in Railway Vehicles,” presented at the 8th IAVSD symposium on Vehicle System Dynamics in the Dynamics of Vehicles on Roads and Tracks, 1984, pp. 655-665.

17. Kaas-Peterson, C., and True, H., “Periodic, Biperiodic, and Chaotic Dynamical Behavior of Railway Vehicles,” presented at the 9th IAVSD symposium on Vehicle System Dynamics in the Dynamics of Vehicles on Roads and Tracks, 1985, pp. 208-221.

18. Knudsen, C., Feldberg, R., and Jaschinski, A., “Non-Linear Dynamic Phenomena in the Behavior of a Railway Wheelset Model,” Nonlinear Dynamics, Vol. 2, 1991, pp. 389-404.

19. Knudsen, C., Slivsgaard, E., Rose, M., True, H., and Feldberg, R., “Dynamics of a Model of a Railway Wheelset,” Nonlinear Dynamics, Vol. 6, 1994, pp. 215-236.

20. Law, E.H., and Brand, R.S., “Analysis of the Nonlinear Dynamics of a Railway Vehicle Wheelset,” Journal of Dynamic Systems, Measurement, and Control, Series G 95, March 1973, pp. 28-35.

21. Scheffel, H., “The Influence of the Suspension on the Hunting Stability of Railway Vehicles,” Rail International, Vol. 10, No. 8, August 1979, pp. 662-696.

22. Ahmadian, M., and Yang, S., “Effect of Suspension Nonlinearities on Rail Vehicle Bifurcation and Stability,” ASME Rail Transportation, Vol. 13, 1997, pp. 97-106.

23. Yang, S., and Ahmadian, M., “The Hopf Bifurcation in a Rail Wheelset with Nonlinear Damping,” ASME Rail Transportation, Vol. 12, 1996, pp. 113-119.

Appendix

PROGRAM FOR SIMULATING DYNAMIC BEHAVIOR OF A SINGLE WHEELSET

```
% M-file name: single_wheelset.m
% M-file type: Script (main) file

%   This program simulates the dynamic behavior of a single wheelset rolling on a straight track.
%   The wheelset is modeled using six degrees of freedom, namely, lateral and yaw wheelset
%   displacements, lateral and yaw wheelset velocities, and left and right rail displacements.

% Initial conditions

% x10: Initial Lateral Displacement of wheelset (m)
% x20: Initial Yaw Displacement of wheelset (rad)
% x30: Initial Lateral Velocity of wheelset (m/sec)
% x40: Initial Yaw Velocity of wheelset (rad/sec)
% x50: Initial Lateral Displacement of left rail (m)
% x60: Initial Lateral Displacement of right rail (m)

x10=0.00635;x20=0.0010;x30=0.10;x40=0.0;x50=0.0000;x60=0.0000;

% Globalizing all the variables

% Values for global variables need to be specified in the main file alone even though they may
% be used in several function files.

% Global variables

% V: Forward velocity of wheelset (m/sec)
% lambda: Wheel conicity
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)
% mw: Mass of wheelset (kg)
% Iwz: Yaw principal mass moment of inertia of wheelset (kg-m^2)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% g: Acceleration due to gravity (m/s^2)
% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)
% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-s/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-s/m)
% dp: Half of lateral distance between primary longitudinal springs (m)
% N: Axle load (N)

global V lambda a r0 yfc yfctol mw Iwz Iwy krail crail g muN f11 f12 f22 f33 kpx cpx kpy cpy
dp N;

V=20;lambda=0.125;a=0.716;r0=0.3556;yfc=0.0080;yfctol=0.0010;mw=1751;Iwz=761;Iwy=13
0;krail=14.6e7;crail=14.6e4;g=9.81;muN=12000;f11=9430000;f12=1.2e3;
f22=1e3;f33=10230000;kpx=9.12e5;cpx=8376.9;kpy=5.84e5;cpy=9048.2;dp=0.61;
N=100000;

% Solving the system of differential equations

%   The system of differential equations is integrated from t=0 sec to t=10 sec with the above
%   initial conditions. The state space variables to be solved are specified in the function file
%   ‘equations.m’. The solver automatically chooses a suitable time step for integration.

[t,x] = ode45('equations',0,50,[x10;x20;x30;x40;x50;x60]);

% Plotting time response and phase portraits

% Plotting lateral and yaw displacements vs. time

subplot(2,1,1),plot(t,x(:,1),'r')
xlabel('Time')
ylabel('Lateral Displacement')
title('Lateral Displacement vs Time')
grid on
subplot(2,1,2),plot(t,x(:,2),'r')
xlabel('Time')
ylabel('Yaw Displacement')
title('Yaw Displacement vs Time')
grid on

figure

% Plotting left and right rail displacements vs. time

subplot(2,1,1),plot(t,x(:,5),'r')
```

```
xlabel(’Time’)
ylabel(’Left Rail Displacement’)
title(’Left Rail Displacement vs Time’)
grid on
subplot(2,1,2),plot(t,x(:,6),’r’)
xlabel(’Time’)
ylabel(’Right Rail Displacement’)
title(’Right Rail Displacement vs Time’)
grid on

figure

% Plotting lateral displacement vs. lateral velocity

plot(x(:,1),x(:,3),’r’)
xlabel(’Lateral Displacement’)
ylabel(’Lateral Velocity’)
title(’Lateral Displacement - Velocity Phase Portrait’)
grid on

figure

% Plotting yaw displacement vs. yaw velocity

plot(x(:,2),x(:,4),’r’)
xlabel(’Yaw Displacement’)
ylabel(’Yaw Velocity’)
title(’Yaw Displacement - Velocity Phase Portrait’)
grid on


% M-file name: equations.m
% M-file type: Function file

%   This function file obtains suspension forces and moments from function file
%   ‘wheelset_suspension.m’ and uses them to solve the single wheelset dynamic equations by
%   invoking the function ‘wheelset’.

%   At the end of the simulation, MATLAB stores each degree of freedom as a column array and
%   the entire solution as a matrix. The solution matrix in this program has been named 'x'.
%   Hence, in this case, the solution matrix 'x' will have 6 columns. The number of rows of 'x'
%   will be equal to the total number of time steps required for the simulation. Throughout this
%   function file, the six degrees of freedom at any particular time step are denoted as:

% x(1): Lateral Displacement of wheelset
% x(2): Yaw Displacement of wheelset
% x(3): Lateral Velocity of wheelset
% x(4): Yaw Velocity of wheelset
% x(5): Lateral Displacement of Left Rail
% x(6): Lateral Displacement of Right Rail

% where x(n) represents the nth column of the solution matrix ’x’.
% The vector of time-derivatives within any time step for the solution vector ’x’ has been
% named ’xdot’. Hence, the time-derivative of variable x(n) would  be xdot(n).

% Nomenclature:

% Fsuspyw: Lateral suspension force on the wheelset
% Msuspzw: Vertical suspension moment on the wheelset
% phi: Wheelset roll angle

function [xdot] = equations(t,x)

% Initializing the vector of time-derivatives

xdot=zeros(6,1);

% Printing time at end of each time-step on the command screen

t

%   Obtaining wheelset suspension forces and moments from function
%   ’wheelset_suspension’

[Fsuspyw,Msuspzw] = wheelset_suspension(x(1),x(3),x(2),x(4));

%   Invoking function ’wheelset’ to solve for wheelset response at each time-step

[xdot(1),xdot(2),xdot(3),xdot(4),xdot(5),xdot(6),phi]=wheelset(x(1),x(2),
x(3),x(4),x(5),x(6),Fsuspyw,Msuspzw);


% M-file name: wheelset.m
% M-file type: Function file

%   This function file identifies the single-point (tread contact) and the two-point (tread and
%   flange contact) equations that constitute the mathematical model for the dynamics of a single
%   wheelset rolling on a straight track. The equations are written in state space form.

%   This function is called by the function file ‘equations.m’ for solving the differential equations
%   constituting the motion of wheelset. The wheelset lateral and yaw displacements and
%   velocities, and the left and right rail lateral displacements are provided as inputs to this
%  function in order to solve for the wheelset. Additionally, the wheelset suspension forces and
%  moments are provided as inputs.


%  This function calls the following functions -
%  ’rolling_radius’, ’contact_angle’, ’onept_creep’, ’onept_normal’, ’twopt_creep’, and
%  ’twopt_normal’.

function [xdot1,xdot2,xdot3,xdot4,xdot5,xdot6,phi] =
wheelset(x1,x2,x3,x4,x5,x6,Fsuspyw,Msuspzw)

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% lambda: wheel conicity
% a: wheel base (m)
% r0: Centered rolling radius of the wheel (m)
% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)
% mw: Mass of wheelset (kg)
% Iwz: Yaw principal mass moment of inertia of wheelset (kg-m^2)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% g: Acceleration due to gravity (m/s^2)

%   Indicating the global nature of the variables. This means that the value of the variables need
%  not be specified in this function file. This value is automatically obtained from the main file
%  ’single_wheelset.m’.

global V lambda a r0 yfc yfctol mw Iwz Iwy krail crail g;


% Nomenclature:

% rlt: Rolling radius at left wheel tread contact patch
% rlf: Rolling radius at left wheel flange contact patch
% rrt: Rolling radius at right wheel tread contact patch
% rrf: Rolling radius at right wheel flange contact patch
% deltalt: Contact angle at left wheel tread contact patch
% deltalf: Contact angle at left wheel flange contact patch
% deltart: Contact angle at right wheel tread contact patch
% deltarf: Contact angle at right wheel flange contact patch
% rl: Rolling radius at left wheel contact patch (for single-point contact)
% rr: Rolling radius at right wheel contact patch (for single-point contact)
% deltal: Effective contact angle at left wheel contact patch after compensating for wheelset roll
% angle (for single-point contact)
% deltatl: Effective contact angle at left wheel tread contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltafl: Effective contact angle at left wheel flange contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltar: Effective contact angle at right wheel contact patch after compensating for wheelset
% roll angle (for single-point contact)
% deltatr: Effective contact angle at right wheel tread contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltafr: Effective contact angle at right wheel flange contact patch after compensating for
% wheelset roll angle (for two-point contact)
% lambdal: Slope of tangent at left wheel contact patch (for single-point contact)
% lambdalt: Slope of tangent at left wheel tread contact patch (for two-point contact)
% lambdalf: Slope of tangent at left wheel flange contact patch (for two-point contact)
% lambdar: Slope of tangent at right wheel contact patch (for single-point contact)
% lambdart: Slope of tangent at right wheel tread contact patch (for two-point contact)
% lambdarf: Slope of tangent at right wheel flange contact patch (for two-point contact)
% phi: Wheelset roll angle
% dphi: Rate of change of wheelset roll angle
% etaxl: Longitudinal creepage at left wheel contact patch (for single-point contact)
% etaxlt: Longitudinal creepage at left wheel tread contact patch (for two-point contact)
% etaxlf: Longitudinal creepage at left wheel flange contact patch (for two-point contact)
% etayl: Lateral creepage at left wheel contact patch (for single-point contact)
% etaylt: Lateral creepage at left wheel tread contact patch (for two-point contact)
% etaylf: Lateral creepage at left wheel flange contact patch (for two-point contact)
% etaspl: Spin creepage at left wheel contact patch (for single-point contact)
% etasplt: Spin creepage at left wheel tread contact patch (for two-point contact)
% etasplf: Spin creepage at left wheel flange contact patch (for two-point contact)
% etaxr: Longitudinal creepage at right wheel contact patch (for single-point contact)
% etaxrt: Longitudinal creepage at right wheel tread contact patch (for two-point contact)
% etaxrf: Longitudinal creepage at right wheel flange contact patch (for two-point contact)
% etayr: Lateral creepage at right wheel contact patch (for single-point contact)
% etayrt: Lateral creepage at right wheel tread contact patch (for two-point contact)
% etayrf: Lateral creepage at right wheel flange contact patch (for two-point contact)
% etaspr: Spin creepage at right wheel contact patch (for single-point contact)
% etasprt: Spin creepage at right wheel tread contact patch (for two-point contact)
% etasprf: Spin creepage at right wheel flange contact patch (for two-point contact)
% Fcxl: Longitudinal creep force on left wheel contact patch (for single-point contact)
% Fcxlt: Longitudinal creep force on left wheel tread contact patch (for two-point contact)
% Fcxlf: Longitudinal creep force on left wheel flange contact patch (for two-point contact)
% Fcyl: Lateral creep force on left wheel contact patch (for single-point contact)
% Fcylt: Lateral creep force on left wheel tread contact patch (for two-point contact)
% Fcylf: Lateral creep force on left wheel flange contact patch (for two-point contact)
% Fczl: Vertical creep force on left wheel contact patch (for single-point contact)
% Fczlt: Vertical creep force on left wheel tread contact patch (for two-point contact)
% Fczlf: Vertical creep force on left wheel flange contact patch (for two-point contact)
% Mcxl: Longitudinal creep moment on left wheel contact patch (for single-point contact)
% Mcxlt: Longitudinal creep moment on left wheel tread contact patch (for two-point contact)
% Mcxlf: Longitudinal creep moment on left wheel flange contact patch (for two-point contact)
% Mcyl: Lateral creep moment on left wheel contact patch (for single-point contact)
% Mcylt: Lateral creep moment on left wheel tread contact patch (for two-point contact)
% Mcylf: Lateral creep moment on left wheel flange contact patch (for two-point contact)
% Mczl: Vertical creep moment on left wheel contact patch (for single-point contact)
% Mczlt: Vertical creep moment on left wheel tread contact patch (for two-point contact)
% Mczlf: Vertical creep moment on left wheel flange contact patch (for two-point contact)
% Fcxr: Longitudinal creep force on right wheel contact patch (for single-point contact)
% Fcxrt: Longitudinal creep force on right wheel tread contact patch (for two-point contact)
% Fcxrf: Longitudinal creep force on right wheel flange contact patch (for two-point contact)
% Fcyr: Lateral creep force on right wheel contact patch (for single-point contact)
% Fcyrt: Lateral creep force on right wheel tread contact patch (for two-point contact)
% Fcyrf: Lateral creep force on right wheel flange contact patch (for two-point contact)
% Fczr: Vertical creep force on right wheel contact patch (for single-point contact)
% Fczrt: Vertical creep force on right wheel tread contact patch (for two-point contact)
% Fczrf: Vertical creep force on right wheel flange contact patch (for two-point contact)
% Mcxr: Longitudinal creep moment on right wheel contact patch (for single-point contact)
% Mcxrt: Longitudinal creep moment on right wheel tread contact patch (for two-point contact)
% Mcxrf: Longitudinal creep moment on right wheel flange contact patch (for two-point contact)
% Mcyr: Lateral creep moment on right wheel contact patch (for single-point contact)
% Mcyrt: Lateral creep moment on right wheel tread contact patch (for two-point contact)
% Mcyrf: Lateral creep moment on right wheel flange contact patch (for two-point contact)
% Mczr: Vertical creep moment on right wheel contact patch (for single-point contact)
% Mczrt: Vertical creep moment on right wheel tread contact patch (for two-point contact)
% Mczrf: Vertical creep moment on right wheel flange contact patch (for two-point contact)
% Fnyl: Lateral normal force on left wheel contact patch (for single-point contact)
% Fnylt: Lateral normal force on left wheel tread contact patch (for two-point contact)
% Fnylf: Lateral normal force on left wheel flange contact patch (for two-point contact)
% Fnzl: Vertical normal force on left wheel contact patch (for single-point contact)
% Fnzlt: Vertical normal force on left wheel tread contact patch (for two-point contact)
% Fnzlf: Vertical normal force on left wheel flange contact patch (for two-point contact)
% Fnyr: Lateral normal force on right wheel contact patch (for single-point contact)
% Fnyrt: Lateral normal force on right wheel tread contact patch (for two-point contact)
% Fnyrf: Lateral normal force on right wheel flange contact patch (for two-point contact)
% Fnzr: Vertical normal force on right wheel contact patch (for single-point contact)
% Fnzrt: Vertical normal force on right wheel tread contact patch (for two-point contact)
% Fnzrf: Vertical normal force on right wheel flange contact patch (for two-point contact)

%   Obtaining rolling radii and contact angles at left and right wheel contact patches from
%   functions ’rolling_radius’ and ’contact_angle’

[rlt,rlf,rrt,rrf] = rolling_radius(x1,x1-x5,x6-x1);

[deltalt,deltalf,deltart,deltarf] = contact_angle(x1,x1-x5,x6-x1);
```

```
%   Choose set of differential equations to be solved depending on single-point or two-point
%   contact condition at end of previous time step

% Single-point contact equations - Right wheel flange contact and Left wheel tread contact

if (x6-x1) > (yfc+yfctol)

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlt;
    rr = rrf;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalt);
    lambdar = tan(deltarf);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rr)/(2*a);
    dphi = (lambdal+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalt+phi;
    deltar = deltarf-phi;

    % Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

    % Computing longitudinal, lateral, and spin creepages at right wheel flange contact patch

    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % ’onept_creep’

    [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
    onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);
    % Obtaining normal forces at left and right wheel contact patches from function
    % 'onept_normal'

    [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
    Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

    % The single-point equations in state space form

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-
          rl*tan(deltal))*(Fcyl+Fnyl)-(a-rr*tan(deltar))
          *(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Two-point contact equations - Right wheel two-point contact and Left wheel tread
% contact

elseif (x6-x1) > yfc

    % Assigning variable name to rolling radius at left wheel contact patch

    rl = rlt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalt);
    lambdart = tan(deltart);
    lambdarf = tan(deltarf);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rrt)/(2*a);
    dphi = (lambdal+lambdarf)* (x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalt+phi;
    deltatr = deltart-phi;
    deltafr = deltarf-phi;

    % Computing longitudinal, lateral, and spin creepages at right wheel tread and flange contact
    % patches
```

```
etaxrt = a*(x4/V)+(1-rrt/r0);
etayrt = ((x3/V)-x2*(rrt/r0)+rrt*dphi/V)/cos(deltatr);
etasprt = ((x4/V)+(phi/r0))*cos(deltatr)+(1/r0)*sin(deltatr);
etaxrf = a*(x4/V)+(1-rrf/r0);
etayrf = ((x3/V)-x2*(rrf/r0)+rrf*dphi/V)/cos(deltafr);
etasprf = ((x4/V)+(phi/r0))*cos(deltafr)+(1/r0)*sin(deltafr);

% Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

etaxl = -a*(x4/V)+(1-rl/r0);
etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

% Obtaining creep forces and moments at left and right wheel contact patches from function
% ’twopt_creep’

[Fcxrt,Fcyrt,Fczrt,Mcxrt,Mcyrt,Mczrt,Fcxrf,Fcyrf,Fczrf,Mcxrf,Mcyrf,Mczrf,Fcxl,Fcyl,Fczl,
Mcxl,Mcyl,Mczl] = twopt_creep(x1,etaxrt,etayrt,etasprt,
etaxrf,etayrf,etasprf,etaxl,etayl,etaspl,deltatr,deltafr,deltal);

% Correcting signs of the vertical creep force components when the vehicle has two-point
% right wheel contact.

Fczrt = -Fczrt;      Fczrf = -Fczrf;           Fczl = -Fczl;
Mcyrt = -Mcyrt;    Mcyrf = -Mcyrf;         Mcyl = -Mcyl;

%Reversing the signs of lateral creep force values to fit in the ’twopt_normal’ function under
% "going right" condition

yfc = -yfc;Fcyrt = -Fcyrt;Fcyrf = -Fcyrf;Fcyl = -Fcyl;

% Reversing the signs of x1, x2, x3, and x4 to allow for mirror effect of right and left
% traverse

x1 = -x1;x2 = -x2;x3 = -x3;x4 = -x4;

% Obtaining normal forces at left and right wheel contact patches from function
% ’twopt_normal’

[Fnyrt,Fnzrt,Fnyrf,Fnzrf,Fnyl,Fnzl] = twopt_normal(x1,x2,x3,x4,
deltatr,deltafr,deltal,rrt,rrf,rl,Fcxrt,Fcyrt,Fczrt,Fcxrf,Fcyrf,Fczrf,
Fcxl,Fcyl,Fczl,Mcyrt,Mcyrf,Mcyl);

% Correcting the signs of the lateral normal forces to adjust for the mirror effect and
% restoring the signs of the lateral creep forces
```

```
    Fnyrt = -Fnyrt; Fnyrf = -Fnyrf; Fnyl = -Fnyl; Fcyrt = -Fcyrt; Fcyrf = -Fcyrf; Fcyl = -Fcyl;

    % Restoring the signs of x1, x2, x3, and x4

    x1 = -x1;x2 = -x2;x3 = -x3;x4 = -x4;

    % The two-point equations in state space form for right wheel two-point contact

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyrt+Fcyrf+Fcyl+Fnyrt+Fnyrf+Fnyl+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)+a*(Fcxrt+Fcxrf-Fcxl)+x2*((a-rrt*
          tan(deltatr))*(Fcyrt+Fnyrt)+(a-rrf*tan(deltafr))*(Fcyrf+Fnyrf)-
          (a-rl*tan(deltal))*(Fcyl+Fnyl))+Mczrt+Mczrf+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyrt-Fnyrf-Fcyrt-Fcyrf-krail*x6);

% Single-point contact equations - Right wheel tread contact and Left wheel flange contact

elseif (x1-x5) > (yfc+yfctol)

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlf;
    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalf);
    lambdar = tan(deltart);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rr)/(2*a);
    dphi = (lambdal+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalf+phi;
    deltar = deltart-phi;

    % Computing longitudinal, lateral, and spin creepages at left wheel flange contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

    % Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch
    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % ’onept_creep’

    [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
    onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);

    % Obtaining normal forces at left and right wheel contact patches from function
    % ’onept_normal’

    [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
    Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

    % The single-point equations in state space form

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-rl*tan(deltal))
        *(Fcyl+Fnyl)-(a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Two point contact equations - Right wheel tread contact and Left wheel two-point
% contact

elseif (x1-x5) > yfc

    % Assigning variable name to rolling radius at right wheel contact patch

    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdar = tan(deltart);
    lambdalt = tan(deltalt);
    lambdalf = tan(deltalf);

    % Computing wheelset roll angle and rate of change of wheelset roll angle
```

```
phi = (rlt-rr)/(2*a);
dphi = (lambdalf+lambdar)*(x3/(2*a));

% Computing effective contact angles at left and right wheel contact patches by
% compensating for wheelset roll angle

deltatl = deltalt+phi;
deltafl = deltalf+phi;
deltar = deltart-phi;

% Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch

etaxr = a*(x4/V)+(1-rr/r0);
etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

% Computing longitudinal, lateral, and spin creepages at left wheel tread and flange contact
% patches

etaxlt = -a*(x4/V)+(1-rlt/r0);
etaylt = ((x3/V)-x2*(rlt/r0)+rlt*dphi/V)/cos(deltatl);
etasplt = ((x4/V)+(phi/r0))*cos(deltatl)-(1/r0)*sin(deltatl);
etaxlf = -a*(x4/V)+(1-rlf/r0);
etaylf = ((x3/V)-x2*(rlf/r0)+rlf*dphi/V)/cos(deltafl);
etasplf = ((x4/V)+(phi/r0))*cos(deltafl)-(1/r0)*sin(deltafl);

% Obtaining creep forces and moments at left and right wheel contact patches from function
% ’twopt_creep’

[Fcxlt,Fcylt,Fczlt,Mcxlt,Mcylt,Mczlt,Fcxlf,Fcylf,Fczlf,Mcxlf,Mcylf,Mczlf,Fcxr,Fcyr,Fczr,M
cxr,Mcyr,Mczr] = twopt_creep(x1,etaxlt,etaylt,etasplt,
etaxlf,etaylf,etasplf,etaxr,etayr,etaspr,deltatl,deltafl,deltar);

% Obtaining normal forces at left and right wheel contact patches from function
% ’twopt_normal’

[Fnylt,Fnzlt,Fnylf,Fnzlf,Fnyr,Fnzr] = twopt_normal(x1,x2,x3,x4,deltatl,
deltafl,deltar,rlt,rlf,rr,Fcxlt,Fcylt,Fczlt,Fcxlf,Fcylf,Fczlf,Fcxr,Fcyr, Fczr,Mcylt,Mcylf,Mcyr);

% The two-point equations in state space form for left wheel two-point contact
xdot1 = x3;
xdot2 = x4;
xdot3 = 1/mw*(Fcylt+Fcylf+Fcyr+Fnylt+Fnylf+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxlt+Fcxlf-Fcxr)-x2*((a-rlt*
    tan(deltatl))*(Fcylt+Fnylt)+(a-rlf*tan(deltafl))*(Fcylf+Fnylf)-
    (a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczlt+Mczlf+Msuspzw);
    xdot5 = (1/crail)*(-Fnylt-Fnylf-Fcylt-Fcylf-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Single-point contact equations - Right wheel tread contact and Left wheel tread contact

else

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlt;
    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalt);
    lambdar = tan(deltart);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rr)/(2*a);
    dphi = (lambdal+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalt+phi;
    deltar = deltart-phi;

    % Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

    % Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch

    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % 'onept_creep'

    [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
    onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);
    % Obtaining normal forces at left and right wheel contact patches from function
    % 'onept_normal'

    [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
    Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

  % The single point equations in state space form

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-rl*tan(deltal))
        *(Fcyl+Fnyl)-(a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

end


% M-file name: onept_creep.m
% M-file type: Function file

%   This function file computes the creep forces and moments acting on the left and the right
%   wheels due to the interaction between wheel and rail. This calculation is valid for a single-
%   point contact condition between the wheel and the rail. The inputs to the function are the
%   longitudinal, lateral, and spin creepages and the contact angles. The outputs are the creep
%   forces and moments resolved in longitudinal, lateral, and vertical directions.

%   Kalker's creep theory is used and spin creep saturation has been taken into account.

%   The outputs given by this function are called by the function file 'wheelset.m'.

function [Fcx1,Fcy1,Fcz1,Mcx1,Mcy1,Mcz1,Fcx2,Fcy2,Fcz2,Mcx2,Mcy2,Mcz2] =
onept_creep(etax1,etay1,etasp1,etax2,etay2,etasp2,delta1,delta2)

% Parameters used for simulation

% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)
% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_wheelset.m’.

global muN f11 f12 f22 f33;

% Computing the creep forces and moments at the left wheel-rail contact patch

fcpx1 = -f33*etax1;
fcpy1 = -f11*etay1-f12*etasp1;
mcpz1 = f12*etay1-f22*etasp1;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx1^2+fcpy1^2);

if beta<=3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx1 = epsilon*fcpx1;
Fcpy1 = epsilon*fcpy1;
Mcpz1 = epsilon*mcpz1;

% Computing the creep forces and moments at the right wheel-rail contact patch

fcpx2 = -f33*etax2;
fcpy2 = -f11*etay2-f12*etasp2;
mcpz2 = f12*etay2-f22*etasp2;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx2^2+fcpy2^2);

if beta<=3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx2 = epsilon*fcpx2;
Fcpy2 = epsilon*fcpy2;
Mcpz2 = epsilon*mcpz2;
```

```
% The creep forces and moments calculated above are in the contact patch plane. These forces
% and moments are resolved in the longitudinal, lateral, and vertical directions in the track
% coordinate system. The variables calculated below are passed to the function 'wheelset'.

% Resolving creep forces and moments for the left wheel

Fcx1 = Fcpx1;
Fcy1 = Fcpy1*cos(delta1);
Fcz1 = Fcpy1*sin(delta1);
Mcx1 = 0;
Mcy1 = -Mcpz1*sin(delta1);
Mcz1 = Mcpz1*cos(delta1);

% Resolving creep forces and moments for the right wheel

Fcx2 = Fcpx2;
Fcy2 = Fcpy2*cos(delta2);
Fcz2 = -Fcpy2*sin(delta2);
Mcx2 = 0;
Mcy2 = Mcpz2*sin(delta2);
Mcz2 = Mcpz2*cos(delta2);


% M-file name: twopt_creep.m
% M-file type: Function file

%   This function file computes the creep forces and moments acting on the left and the right
%   wheels due to the interaction between wheel and rail. This calculation is valid for a two-point
%   contact condition between the wheel and the rail. The inputs to the function are the
%   longitudinal, lateral, and spin creepages and the contact angles. The outputs are the creep
%   forces and moments resolved in longitudinal, lateral, and vertical directions.

%   Kalker's creep theory is used and spin creep saturation has been taken into account.

%   The outputs given by this function are called by the function file 'wheelset.m'.

function [Fcx1,Fcy1,Fcz1,Mcx1,Mcy1,Mcz1,Fcx2,Fcy2,Fcz2,Mcx2,Mcy2,Mcz2,Fcx3,
Fcy3,Fcz3,Mcx3,Mcy3,Mcz3] = twopt_creep(x1,etax1,etay1,etasp1,etax2,etay2,
etasp2,etax3,etay3,etasp3,delta1,delta2,delta3)

% Parameters used for simulation

% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'single_wheelset.m'.

global muN f11 f12 f22 f33;

% Computing the creep forces and moments at contact patch 1. For two-point contact at the left
% wheel, contact patch 1 is the left wheel-rail tread contact patch. For two-point contact at the
% right wheel, contact patch 1 is the right wheel-rail tread contact patch.

fcpx1 = -f33*etax1;
fcpy1 = -f11*etay1-f12*etasp1;
mcpz1 = f12*etay1-f22*etasp1;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx1^2+fcpy1^2);

if beta<3
   epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
   epsilon = (1/beta);
end

Fcpx1 = epsilon*fcpx1;
Fcpy1 = epsilon*fcpy1;
Mcpz1 = epsilon*mcpz1;

% Computing the creep forces and moments at contact patch 2. For two-point contact at the left
% wheel, contact patch 2 is the left wheel-rail flange contact patch. For two-point contact at the
% right wheel, contact patch 2 is the right wheel-rail flange contact patch.

fcpx2 = -f33*etax2;
fcpy2 = -f11*etay2-f12*etasp2;
mcpz2 = f12*etay2-f22*etasp2;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx2^2+fcpy2^2);

if beta<3
   epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
```

```
  epsilon = (1/beta);
end

Fcpx2 = epsilon*fcpx2;
Fcpy2 = epsilon*fcpy2;
Mcpz2 = epsilon*mcpz2;

% Computing the creep forces and moments at contact patch 3. For two-point contact at the left
% wheel, contact patch 3 is the right wheel-rail contact patch. For two-point contact at the right
% wheel, contact patch 3 is the left wheel-rail flange contact patch.

fcpx3 = -f33*etax3;
fcpy3 = -f11*etay3-f12*etasp3;
mcpz3 = f12*etay3-f22*etasp3;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx3^2+fcpy3^2);

if beta<3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx3 = epsilon*fcpx3;
Fcpy3 = epsilon*fcpy3;
Mcpz3 = epsilon*mcpz3;

% The creep forces and moments calculated above are in the contact patch plane. These forces
% and moments are resolved in the longitudinal, lateral, and vertical directions in the track
% coordinate system. The variables calculated below are passed to the function ’wheelset’.

% Resolving creep forces and moments at contact patch 1

Fcx1 = Fcpx1;
Fcy1 = Fcpy1*cos(delta1);
Fcz1 = Fcpy1*sin(delta1);
Mcx1 = 0;
Mcy1 = -Mcpz1*sin(delta1);
Mcz1 = Mcpz1*cos(delta1);

% Resolving creep forces and moments at contact patch 2

Fcx2 = Fcpx2;
Fcy2 = Fcpy2*cos(delta2);
```

```
Fcz2 = Fcpy2*sin(delta2);
Mcx2 = 0;
Mcy2 = -Mcpz2*sin(delta2);
Mcz2 = Mcpz2*cos(delta2);

% Resolving creep forces and moments at contact patch 3

Fcx3 = Fcpx3;
Fcy3 = Fcpy3*cos(delta3);
Fcz3 = -Fcpy3*sin(delta3);
Mcx3 = 0;
Mcy3 = Mcpz3*sin(delta3);
Mcz3 = Mcpz3*cos(delta3);


% M-file name: onept_normal.m
% M-file type: Function file

%   This function file computes the normal forces acting on the left and the right wheels due to
%   the interaction between wheel and rail. This calculation is valid for a single-point contact
%   condition between the wheel and the rail. The inputs to the function are the longitudinal,
%   lateral, and vertical creep forces, the lateral creep moments, rolling radii and contact angles at
%   the left and the right wheel contact patches, and the wheelset yaw displacement and velocity.
%   The outputs are the normal forces on the left and the right wheels resolved in longitudinal,
%   lateral, and vertical directions.

%   The outputs given by this function are called by the function file 'wheelset.m'.

function [Fny1,Fnz1,Fny2,Fnz2] = onept_normal(x2,x4,delta1,delta2,r1,r2,
Fcx1,Fcx2,Fcy1,Fcy2,Fcz1,Fcz2,Mcy1,Mcy2)

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% mw: Mass of wheelset (kg)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% g: Acceleration due to gravity (m/s^2)
% N: Axle load (N)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'single_wheelset.m'.

global V a r0 mw Iwy g N;
```

```
% Computing the normal forces at the left and right wheel-rail contact patches

F = -Fcz1-Fcz2+mw*g+N;
M = a*(Fcz2-Fcz1)-r1*(Fcy1-x2*Fcx1)-r2*(Fcy2-x2*Fcx2)-x2*(Mcy1+Mcy2)-Iwy*(V/r0)*x4;
vl = F*(a*cos(delta2)-r2*sin(delta2))+M*cos(delta2);
vr = F*(a*cos(delta1)-r1*sin(delta1))-M*cos(delta1);
del1 = 2*a*cos(delta1)*cos(delta2)-r2*cos(delta1)*sin(delta2)-r1*sin(delta1)*cos(delta2);

Fnl = vl/del1;
Fnr = vr/del1;

% The normal forces calculated above are normal to the contact patch plane. These forces are
% resolved in the longitudinal, lateral, and vertical directions in the track coordinate system. The
% variables calculated below are passed to the function 'wheelset'.

% Resolving normal forces at the left and right wheel contact patches

Fny1 = -Fnl*sin(delta1);
Fnz1 = Fnl*cos(delta1);
Fny2 = Fnr*sin(delta2);
Fnz2 = Fnr*cos(delta2);


% M-file name: twopt_normal.m
% M-file type: Function file

%  This function file computes the normal forces acting on the left and the right wheels due to
%  the interaction between wheel and rail. This calculation is valid for a two-point contact
%  condition between the wheel and the rail. The inputs to the function are the longitudinal,
%  lateral, and vertical creep forces, the lateral creep moments, rolling radii and contact angles at
%  the left and the right wheel contact patches, and the wheelset lateral and yaw displacement
%  and velocity. The outputs are the normal forces on the left and the right wheels resolved in
%  longitudinal, lateral, and vertical directions.

%  The outputs given by this function are called by the function file 'wheelset.m'.

function [Fny1,Fnz1,Fny2,Fnz2,Fny3,Fnz3] = twopt_normal(x1,x2,x3,x4,delta1,
delta2,delta3,r1,r2,r3,Fcx1,Fcy1,Fcz1,Fcx2,Fcy2,Fcz2,Fcx3,Fcy3,Fcz3,Mcy1,Mcy2,Mcy3)

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% mw: Mass of wheelset (kg)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% g: Acceleration due to gravity (m/s^2)
% yfc: Flange clearance or flange width (m)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% N: Axle load (N)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'single_wheelset.m'.

global V a r0 mw Iwy g yfc krail crail N;

% Computing the normal forces at the left and right wheel-rail contact patches

F2 = -Fcz1-Fcz2-Fcz3+N+mw*g;
M1 = -a*(Fcz1+Fcz2-Fcz3)-r1*(Fcy1-x2*Fcx1)-r2*(Fcy2-x2*Fcx2)-r3*(Fcy3-x2*Fcx3)-
x2*(Mcy1+Mcy2+Mcy3)-Iwy*(V/r0)*x4;

Vt = F1*(2*a*cos(delta2)*cos(delta3)-r2*sin(delta2)*cos(delta3)-r3*
cos(delta2)*sin(delta3))+F2*(sin(delta2)*(a*cos(delta3)-r3*sin(delta3)))
+M1*sin(delta2)*cos(delta3);

vf = F1*(-2*a*cos(delta1)*cos(delta3)+r1*sin(delta1)*cos(delta3)+r3*
cos(delta1)*sin(delta3))-F2*(sin(delta1)*(a*cos(delta3)-r3*sin(delta3)))-
M1*sin(delta1)*cos(delta3);

v = F1*(r2*cos(delta1)*sin(delta2)-r1*sin(delta1)*cos(delta2))+F2*
(a*(cos(delta1)*sin(delta2)-sin(delta1)*cos(delta2))+(r2-r1)* sin(delta1)*sin(delta2))-
M1*(cos(delta1)*sin(delta2)-sin(delta1)*
cos(delta2));

del2 = (2*a*cos(delta3)-r3*sin(delta3))*(cos(delta1)*sin(delta2)-sin(delta1)*cos(delta2))+(r2-
r1)*sin(delta1)*sin(delta2)*cos(delta3);

Fnt = vt/del2;
Fnf = vf/del2;
Fn = v/del2;

% The normal forces calculated above are normal to the contact patch plane. These forces are
% resolved in the longitudinal, lateral, and vertical directions in the track coordinate system. The
% variables calculated below are passed to the function 'wheelset'.

% Resolving normal forces at the left and right wheel contact patches

Fny1 = -Fnt*sin(delta1);
Fnz1 = Fnt*cos(delta1);
```

```
Fny2 = -Fnf*sin(delta2);
Fnz2 = Fnf*cos(delta2);
Fny3 = Fn*sin(delta3);
Fnz3 = Fn*cos(delta3);


% M-file name: wheelset_suspension.m
% M-file type: Function file

%   This function file computes the suspension forces acting on the wheelset due to the primary
%   suspension components. The primary suspension consists of longitudinal and lateral springs
%   and dampers. The inputs to the function are the wheelset lateral and yaw displacements and
%   velocities. The outputs are the lateral suspension force and the vertical suspension moment
%   on the wheelset.

%   The outputs given by this function are called by the function file ’equations.m’.

function [Fsusp1,Msusp1] = wheelset_suspension(x1,x3,x2,x4)

% Parameters used for simulation

% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-sec/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-sec/m)
% dp: Half of lateral distance between primary longitudinal springs (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_wheelset.m’.

global kpx cpx kpy cpy dp;

% Computing the suspension forces and moments acting on the wheelset. These variables are
% passed to the function ’equations’.

Fsusp1 = -2*kpy*x1-2*cpy*x3;
Msusp1 = -2*dp^2*kpx*x2-2*dp^2*cpx*x4;


% M-file name: rolling_radius.m
% M-file type: Function file

%   This function file computes the rolling radii at the left and the right wheel-rail contact
%   patches as a function of the relative lateral displacement between the left and the right wheels
%   and rails. The inputs to the function are the relative lateral displacement between the left
%   wheel and the left rail and the right wheel and the right rail. The outputs are the rolling radii
%   at the left and the right tread and flange contact patches.

%   The outputs given by this function are called by the function file ’wheelset.m’.

function [rlefttread,rleftflange,rrighttread,rrightflange] = rolling_radius(y,yl,yr)

% Parameters used for simulation

% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)
% r0: Centered rolling radius of the wheel (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_wheelset.m’.

global yfc yfctol r0;

% Computing the rolling radii at the left and the right wheel-rail contact patches. These variables
% are passed to the function ‘wheelset’.

if y >= 0

    % Wheelset displacement is in the positive (or left) direction from the centerline (Right
    %   wheel tread contact)

    % Note that for a left movement of the wheelset from the centerline, the
    % right wheel does not flange. Hence, the zero value for the right wheel
    % flange radius is not real and is not used in calculations.

    rrighttread = r0 + 0.125*(yr);
    rrightflange = 0.000;

    if yl <= yfc

    % Left wheel tread contact. Note that the zero value for the left wheel flange radius is not
    % real and is not used in calculations.

    rlefttread = r0 + 0.125*yl;
    rleftflange = 0.000;

    elseif yl < (yfc+yfctol)

    % Left wheel flange contact
    rlefttread = r0+0.125*yl;
    rleftflange = 0.3566+13700*(yl-0.008)^2;

    else

    % Left wheel two-point contact

    rlefttread = r0+0.125*yl;
    rleftflange = 0.3703+2.5833*(yl-0.009);

    end

else

    % Wheelset displacement is in the negative (or right) direction from the centerline (Left
    % wheel tread contact)

    % Note that for a right movement of the wheelset from the centerline, the
    % left wheel does not flange. Hence, the zero value for the left wheel
    % flange radius is not real and is not used in calculations.

    rlefttread = r0 + 0.125*yl;
    rleftflange = 0.000;

    if yr <= yfc

    % Right wheel tread contact. Note that the zero value for the right
    % wheel flange radius is not real and is not used in calculations.

    rrighttread = r0 + 0.125*(yr);
    rrightflange = 0.000;

    elseif yr < (yfc+yfctol)

    % Right wheel flange contact

    rrighttread = r0 + 0.125*yr;
    rrightflange = 0.3566+13700*(yr-0.008)^2;

    else
  % Right wheel two-point contact

    rrighttread = r0+0.125*yr;
    rrightflange = 0.3703+2.5833*(yr-0.009);

    end
```

```
end


% M-file name: contact_angle.m
% M-file type: Function file

%   This function file computes the contact angles at the left and the right wheel-rail contact
%   patches as a function of the relative lateral displacement between the left and the right wheels
%   and rails. The inputs to the function are the relative lateral displacement between the left
%   wheel and the left rail and the right wheel and the right rail. The outputs are the contact
%   angles at the left and the right tread and flange contact patches.

%   The outputs given by this function are called by the function file ’wheelset.m’.

function [calefttread,caleftflange,carighttread,carightflange] = contact_angle(y,yl,yr)

% Parameters used for simulation

% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_wheelset.m’.

global yfc yfctol;

% Computing the rolling radii at the left and the right wheel-rail contact patches. These variables
% are passed to the function ‘wheelset’.

if y >= 0

    % Wheelset displacement is in the positive (or left) direction from the centerline (Right
    % wheel tread contact)

    % Note that for a left movement of the wheelset from the centerline, the
    % right wheel does not flange. Hence, the zero value for the right wheel
    % flange angle is not real and is not used in calculations.

    carighttread = atan(0.125);
    carightflange = 0.000;

    if yl <= yfc


    % Left wheel tread contact. Note that the zero value for the left wheel flange angle is not real
    % and is not used in calculations.
    calefttread = atan(0.125);
    caleftflange = 0.000;

    elseif yl < (yfc+yfctol)

    % Left wheel flange contact

    calefttread = atan(0.125);
    caleftflange = atan(0.125+2648*(yl-0.008));

    else

    % Left wheel two-point contact

    calefttread = atan(0.125);
    caleftflange = atan(2.748);

    end

else

    % Wheelset displacement is in the negative (or right) direction from the centerline (Left
    % wheel tread contact)

    % Note that for a right movement of the wheelset from the centerline, the
    % left wheel does not flange. Hence, the zero value for the left wheel
    % flange angle is not real and is not used in calculations.

    calefttread = atan(0.125);
    caleftflange = 0.000;

    if yr <= yfc

    % Right wheel tread contact. Note that the zero value for the right
    % wheel flange angle is not real and is not used in calculations.

    carighttread = atan(0.125);
    carightflange = 0.000;

    elseif yr < (yfc+yfctol)

    % Right wheel flange contact

    carighttread = atan(0.125);
    carightflange = atan(0.125+2648*(yr-0.008));
    else

    % Right wheel two-point contact

    carighttread = atan(0.125);
    carightflange = atan(2.748);

    end

end
```

PROGRAM FOR SIMULATING DYNAMIC BEHAVIOR OF A SINGLE TRUCK

```
% M-file name: single_truck.m
% M-file type: Script (main) file

%  This program simulates the dynamic behavior of a two-axle truck and bolster moving on a
%  straight track. The model of the truck is based on Cooperider's complex truck model, in
%  which the front and the rear wheelsets have independent degrees of freedom. The wheelsets
%  are connected to the truck frame through primary suspension elements. The truck is modeled
%  using sixteen degrees of freedom, namely, the lateral and yaw displacements and velocities
%  of the front and the rear wheelsets, the left and the right rail displacements, and the lateral
%  and yaw displacements and velocities of the truck. In addition, the bolster is modeled with
%  two degrees of freedom, namely, the yaw displacement and velocity.

% Initial conditions

% x10: Initial Lateral Displacement of front wheelset (m)
% x20: Initial Yaw Displacement of front wheelset (rad)
% x30: Initial Lateral Velocity of front wheelset (m/sec)
% x40: Initial Yaw Velocity of front wheelset (rad/sec)
% x50: Initial Lateral Displacement of left front rail (m)
% x60: Initial Lateral Displacement of right front rail (m)
% x70: Initial Lateral Displacement of rear wheelset (m)
% x80: Initial Yaw Displacement of rear wheelset (rad)
% x90: Initial Lateral Velocity of rear wheelset (m/sec)
% x100: Initial Yaw Velocity of rear wheelset (rad/sec)
% x110: Initial Lateral Displacement of left rear rail (m)
% x120: Initial Lateral Displacement of right rear rail (m)
% x130: Initial Lateral Displacement of truck frame (m)
% x140: Initial Yaw Displacement of truck frame (rad)
% x150: Initial Lateral Velocity of truck frame (m/sec)
% x160: Initial Yaw Velocity of truck frame (rad/sec)
% x170: Initial Yaw Displacement of bolster (rad)
% x180: Initial Yaw Velocity of bolster (rad/sec)

x10=0.00635;x20=0.0010;x30=0.010;x40=0.0;x50=0.0000;x60=0.0000;x70=0.0030;
x80=0.0010;x90=0.010;x100=0.0;x110=0.0000;x120=0.0000;x130=0.0000;x140=0.0010;x150=
0.10;x160=0.0;x170=0.0;x180=0.0;

% Globalizing all the variables

% Values for global variables need to be specified in the main file alone even though they may
% be used in several function files.

% Global variables
% V: Forward velocity of wheelset (m/sec)
% lambda: Wheel conicity
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation
% mw: Mass of wheelset (kg)
% Iwz: Yaw principal mass moment of inertia of wheelset (kg-m^2)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% g: Acceleration due to gravity (m/s^2)
% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)
% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-s/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-s/m)
% dp: Half of lateral distance between primary longitudinal springs (m)
% mf: Mass of truck frame (kg)
% mb: Mass of bolster (kg)
% Ifz: Yaw principal mass moment of inertia of truck frame (kg-m^2)
% Ibz: Yaw principal mass moment of inertia of bolster (kg-m^2)
% b: Half of wheelbase (m)
% N: Axle load (N)
% ksy: Secondary lateral stiffness coefficient (N/m)
% csy: Secondary lateral damping coefficient (N-sec/m)
% ksphi: Secondary yaw stiffness coefficient (N/rad)
% csphi: Secondary yaw damping coefficient (N-sec/rad)
% T0: Centerplate breakaway torque (N-m)
% C0: Coulomb viscous yaw damping coefficient (N-sec/rad)

global V lambda a r0 yfc yfctol mw Iwz Iwy krail crail g muN f11 f12 f22 f33 kpx cpx kpy cpy
dp mf mb Ifz Ibz b N ksy csy ksphi csphi T0 C0;

V=20;lambda=0.125;a=0.716;r0=0.3556;yfc=0.0080;yfctol=0.0010;mw=1751;Iwz=761;Iwy=13
0;krail=14.6e7;crail=14.6e4;g=9.81;muN=12000;f11=9430000;f12=1.2e3;
f22=1e3;f33=10230000;kpx=9.12e5;cpx=8376.9;kpy=5.84e5;cpy=9048.2;dp=0.61;
mf=4041;mb=365;Ifz=3371;Ibz=337;b=1.295;N=100000;ksy=3.5e5;csy=1.75e4;
ksphi=3.8e8;csphi=2.5e7;T0=10168;C0=3.5e7;

% Specify initial time, final time, and number of time steps
```

```
tspan=linspace(0,1.5,1500);

% Solve the system of differential equations

% The system of differential equations is integrated from t=0 sec to t=1.5 sec with the above
% initial conditions. The function file ‘equations.m’ points to the system of differential equations
% which are contained within the function files ‘wheelset.m’ and ‘truck.m’.

[t,x] = ode23('equations',tspan,[x10;x20;x30;x40;x50;x60;x70;x80;x90;x100;
x110;x120;x130;x140;x150;x160;x170;x180]);

% Plotting time response

% Plotting lateral and yaw displacements for the front wheelset vs. time

subplot(2,1,1),plot(t,x(:,1),'r')
xlabel('Time')
ylabel('Lateral Displacement of front wheelset')
title('Lateral Displacement vs Time for front wheelset')
grid on
subplot(2,1,2),plot(t,x(:,2),'r')
xlabel('Time')
ylabel('Yaw Displacement of front wheelset')
title('Yaw Displacement vs Time for front wheelset')
grid on

figure

% Plotting lateral and yaw displacements for the rear wheelset vs. time

subplot(2,1,1),plot(t,x(:,7),'r')
xlabel('Time')
ylabel('Lateral Displacement of rear wheelset')
title('Lateral Displacement vs Time for rear wheelset')
grid on
subplot(2,1,2),plot(t,x(:,8),'r')
xlabel('Time')
ylabel('Yaw Displacement of rear wheelset')
title('Yaw Displacement vs Time for rear wheelset')
grid on

figure

% Plotting lateral and yaw displacements for the truck frame vs. time

subplot(2,1,1),plot(t,x(:,13),'r')
```

```
xlabel(’Time’)
ylabel(’Lateral Displacement of truck frame’)
title(’Lateral Displacement vs Time for truck frame’)
grid on
subplot(2,1,2),plot(t,x(:,14),’r’)
xlabel(’Time’)
ylabel(’Yaw Displacement of truck frame’)
title(’Yaw Displacement vs Time for truck frame’)
grid on

figure

% Plotting displacement vs. time for front left and right rails

subplot(2,1,1),plot(t,x(:,5),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of front left rail’)
title(’Lateral Displacement vs Time for front left rail’)
grid on
subplot(2,1,2),plot(t,x(:,6),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of front right rail’)
title(’Lateral Displacement vs Time for front right rail’)
grid on

figure

% Plotting displacement vs. time for rear left and right rails

subplot(2,1,1),plot(t,x(:,11),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of rear left rail’)
title(’Lateral Displacement vs Time for rear left rail’)
grid on
subplot(2,1,2),plot(t,x(:,12),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of rear right rail’)
title(’Lateral Displacement vs Time for rear right rail’)
grid on


% M-file name: equations.m
% M-file type: Function file
%  This function file obtains suspension forces and moments from function files
%  ‘wheelset_suspension.m’ and ‘truck_suspension.m’ and uses them to solve the wheelset and
%  truck dynamic equations by invoking the functions ‘wheelset’ and ‘truck’.

%  At the end of the simulation, MATLAB stores each degree of freedom as a column array and
%  the entire solution as a matrix. The solution matrix in this program has been named 'x'.
%  Hence, in this case, the solution matrix 'x' will have 16 columns. The number of rows of 'x'
%  will be equal to the total number of time steps required for the simulation. Throughout this
%  function file, the sixteen degrees of freedom at any particular time step are denoted as:

%  x(1): Lateral Displacement of front wheelset
%  x(2): Yaw Displacement of front wheelset
%  x(3): Lateral Velocity of front wheelset
%  x(4): Yaw Velocity of front wheelset
%  x(5): Lateral Displacement of left front rail
%  x(6): Lateral Displacement of right front rail
%  x(7): Lateral Displacement of rear wheelset
%  x(8): Yaw Displacement of rear wheelset
%  x(9): Lateral Velocity of rear wheelset
%  x(10): Yaw Velocity of rear wheelset
%  x(11): Lateral Displacement of left rear rail
%  x(12): Lateral Displacement of right rear rail
%  x(13): Lateral Displacement of truck frame
%  x(14): Yaw Displacement of truck frame
%  x(15): Lateral Velocity of truck frame
%  x(16): Yaw Velocity of truck frame
%  x(17): Yaw Displacement of bolster
%  x(18): Yaw Velocity of bolster

%  where x(n) represents the nth column of the solution matrix 'x'.
%  The vector of time-derivatives within any time step for the solution vector 'x' has been named
%  'xdot'. Hence, the time-derivative of variable x(n) would  be xdot(n).

function [xdot] = equations(t,x)

% Nomenclature:

% Fsusp1w1: Lateral suspension force on the front wheelset
% Fsusp1w2: Lateral suspension force on the rear wheelset
% Msusp1w1: Vertical suspension moment on the front wheelset
% Msusp1w2: Vertical suspension moment on the rear wheelset
% Fsuspt1: Lateral suspension force on the truck frame
% Msuspt1: Vertical suspension moment on the truck frame
% Msuspb1: Vertical suspension moment on the bolster
% phiw1: Roll angle of front wheelset
% phiw2: Roll angle of rear wheelset
% Initializing the vector of time-derivatives

xdot = zeros(18,1);

% Printing time at end of each time-step on the command screen

t

% Obtaining front wheelset suspension forces from function ’wheelset_suspension’. Note that the
% variable Fsusp1w2 is merely a dummy and is not passed to the function ‘wheelset’.

[Fsusp1w1,Fsusp1w2,Msusp1w1] =
wheelset_suspension(x(1),x(3),x(2),x(4),x(13),x(14),x(15),x(16));

% The function ‘wheelset’ is invoked in order to solve the differential equations for the front
% wheelset.

[xdot(1),xdot(2),xdot(3),xdot(4),xdot(5),xdot(6),phiw1] =
wheelset(x(1),x(2),x(3),x(4),x(5),x(6),Fsusp1w1,Msusp1w1);

% Obtaining rear wheelset suspension forces from function 'wheelset_suspension'. Note that the
% variable Fsusp1w1 is merely a dummy and is not passed to the function ‘wheelset’.

[Fsusp1w1,Fsusp1w2,Msusp1w2] =
wheelset_suspension(x(7),x(9),x(8),x(10),x(13),x(14),x(15),x(16));

% The function ‘wheelset’ is invoked in order to solve the differential equations for the rear
% wheelset.

[xdot(7),xdot(8),xdot(9),xdot(10),xdot(11),xdot(12),phiw2] =
wheelset(x(7),x(8),x(9),x(10),x(11),x(12),Fsusp1w2,Msusp1w2);

% Obtaining truck frame and bolster suspension forces from function 'truck_suspension'

[Fsuspt1,Msuspt1,Msuspb1]=truck_suspension(x(1),x(2),x(3),x(4),x(7),x(8),x(9),x(10),x(13),x(1
4),x(15),x(16),x(17),x(18));

% The function ‘truck’ is invoked in order to solve the differential equations for the truck frame.

[xdot(13),xdot(14),xdot(15),xdot(16),xdot(17),xdot(18)]=truck(x(13),x(14),
x(15),x(16),x(17),x(18),phiw1,phiw2,Fsuspt1,Msuspt1,Msuspb1);


% M-file name: wheelset.m
% M-file type: Function file
%  This function file identifies the single-point (tread contact) and the two-point (tread and
%  flange contact) equations that constitute the mathematical model for the dynamics of a single
%  wheelset rolling on a straight track. The equations are written in state space form.

%  This function is called by the function file ‘equations.m’ for solving the differential equations
%  constituting the motion of wheelset. The wheelset lateral and yaw displacements and
%  velocities, and the left and right rail lateral displacements are provided as inputs to this
%  function in order to solve for the wheelset. Additionally, the wheelset suspension forces and
%  moments are provided as inputs.

%  This function calls the following functions -
%  'rolling_radius', 'contact_angle', 'onept_creep', 'onept_normal', 'twopt_creep', and
%  'twopt_normal'.

function [xdot1,xdot2,xdot3,xdot4,xdot5,xdot6,phi] =
wheelset(x1,x2,x3,x4,x5,x6,Fsuspyw,Msuspzw)

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% lambda: Wheel conicity
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)
% mw: Mass of wheelset (kg)
% Iwz: Yaw principal mass moment of inertia of wheelset (kg-m^2)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% g: Acceleration due to gravity (m/s^2)

%   Indicating the global nature of the variables. This means that the value of the variables need
%  not be specified in this function file. This value is automatically obtained from the main file
%  'single_truck.m'.

global V lambda a r0 yfc yfctol mw Iwz Iwy krail crail g;

% Nomenclature:

% rlt: Rolling radius at left wheel tread contact patch
% rlf: Rolling radius at left wheel flange contact patch
% rrt: Rolling radius at right wheel tread contact patch
% rrf: Rolling radius at right wheel flange contact patch
% deltalt: Contact angle at left wheel tread contact patch
% deltalf: Contact angle at left wheel flange contact patch
% deltart: Contact angle at right wheel tread contact patch
% deltarf: Contact angle at right wheel flange contact patch
% rl: Rolling radius at left wheel contact patch (for single-point contact)
% rr: Rolling radius at right wheel contact patch (for single-point contact)
% deltal: Effective contact angle at left wheel contact patch after compensating for wheelset roll
% angle (for single-point contact)
% deltatl: Effective contact angle at left wheel tread contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltafl: Effective contact angle at left wheel flange contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltar: Effective contact angle at right wheel contact patch after compensating for wheelset
% roll angle (for single-point contact)
% deltatr: Effective contact angle at right wheel tread contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltafr: Effective contact angle at right wheel flange contact patch after compensating for
% wheelset roll angle (for two-point contact)
% lambdal: Slope of tangent at left wheel contact patch (for single-point contact)
% lambdalt: Slope of tangent at left wheel tread contact patch (for two-point contact)
% lambdalf: Slope of tangent at left wheel flange contact patch (for two-point contact)
% lambdar: Slope of tangent at right wheel contact patch (for single-point contact)
% lambdart: Slope of tangent at right wheel tread contact patch (for two-point contact)
% lambdarf: Slope of tangent at right wheel flange contact patch (for two-point contact)
% phi: Wheelset roll angle
% dphi: Rate of change of wheelset roll angle
% etaxl: Longitudinal creepage at left wheel contact patch (for single-point contact)
% etaxlt: Longitudinal creepage at left wheel tread contact patch (for two-point contact)
% etaxlf: Longitudinal creepage at left wheel flange contact patch (for two-point contact)
% etayl: Lateral creepage at left wheel contact patch (for single-point contact)
% etaylt: Lateral creepage at left wheel tread contact patch (for two-point contact)
% etaylf: Lateral creepage at left wheel flange contact patch (for two-point contact)
% etaspl: Spin creepage at left wheel contact patch (for single-point contact)
% etasplt: Spin creepage at left wheel tread contact patch (for two-point contact)
% etasplf: Spin creepage at left wheel flange contact patch (for two-point contact)
% etaxr: Longitudinal creepage at right wheel contact patch (for single-point contact)
% etaxrt: Longitudinal creepage at right wheel tread contact patch (for two-point contact)
% etaxrf: Longitudinal creepage at right wheel flange contact patch (for two-point contact)
% etayr: Lateral creepage at right wheel contact patch (for single-point contact)
% etayrt: Lateral creepage at right wheel tread contact patch (for two-point contact)
% etayrf: Lateral creepage at right wheel flange contact patch (for two-point contact)
% etaspr: Spin creepage at right wheel contact patch (for single-point contact)
% etasprt: Spin creepage at right wheel tread contact patch (for two-point contact)
% etasprf: Spin creepage at right wheel flange contact patch (for two-point contact)
% Fcxl: Longitudinal creep force on left wheel contact patch (for single-point contact)
% Fcxlt: Longitudinal creep force on left wheel tread contact patch (for two-point contact)
% Fcxlf: Longitudinal creep force on left wheel flange contact patch (for two-point contact)
% Fcyl: Lateral creep force on left wheel contact patch (for single-point contact)
% Fcylt: Lateral creep force on left wheel tread contact patch (for two-point contact)
% Fcylf: Lateral creep force on left wheel flange contact patch (for two-point contact)
% Fczl: Vertical creep force on left wheel contact patch (for single-point contact)
% Fczlt: Vertical creep force on left wheel tread contact patch (for two-point contact)
% Fczlf: Vertical creep force on left wheel flange contact patch (for two-point contact)
% Mcxl: Longitudinal creep moment on left wheel contact patch (for single-point contact)
% Mcxlt: Longitudinal creep moment on left wheel tread contact patch (for two-point contact)
% Mcxlf: Longitudinal creep moment on left wheel flange contact patch (for two-point contact)
% Mcyl: Lateral creep moment on left wheel contact patch (for single-point contact)
% Mcylt: Lateral creep moment on left wheel tread contact patch (for two-point contact)
% Mcylf: Lateral creep moment on left wheel flange contact patch (for two-point contact)
% Mczl: Vertical creep moment on left wheel contact patch (for single-point contact)
% Mczlt: Vertical creep moment on left wheel tread contact patch (for two-point contact)
% Mczlf: Vertical creep moment on left wheel flange contact patch (for two-point contact)
% Fcxr: Longitudinal creep force on right wheel contact patch (for single-point contact)
% Fcxrt: Longitudinal creep force on right wheel tread contact patch (for two-point contact)
% Fcxrf: Longitudinal creep force on right wheel flange contact patch (for two-point contact)
% Fcyr: Lateral creep force on right wheel contact patch (for single-point contact)
% Fcyrt: Lateral creep force on right wheel tread contact patch (for two-point contact)
% Fcyrf: Lateral creep force on right wheel flange contact patch (for two-point contact)
% Fczr: Vertical creep force on right wheel contact patch (for single-point contact)
% Fczrt: Vertical creep force on right wheel tread contact patch (for two-point contact)
% Fczrf: Vertical creep force on right wheel flange contact patch (for two-point contact)
% Mcxr: Longitudinal creep moment on right wheel contact patch (for single-point contact)
% Mcxrt: Longitudinal creep moment on right wheel tread contact patch (for two-point contact)
% Mcxrf: Longitudinal creep moment on right wheel flange contact patch (for two-point contact)
% Mcyr: Lateral creep moment on right wheel contact patch (for single-point contact)
% Mcyrt: Lateral creep moment on right wheel tread contact patch (for two-point contact)
% Mcyrf: Lateral creep moment on right wheel flange contact patch (for two-point contact)
% Mczr: Vertical creep moment on right wheel contact patch (for single-point contact)
% Mczrt: Vertical creep moment on right wheel tread contact patch (for two-point contact)
% Mczrf: Vertical creep moment on right wheel flange contact patch (for two-point contact)
% Fnyl: Lateral normal force on left wheel contact patch (for single-point contact)
% Fnylt: Lateral normal force on left wheel tread contact patch (for two-point contact)
% Fnylf: Lateral normal force on left wheel flange contact patch (for two-point contact)
% Fnzl: Vertical normal force on left wheel contact patch (for single-point contact)
% Fnzlt: Vertical normal force on left wheel tread contact patch (for two-point contact)
% Fnzlf: Vertical normal force on left wheel flange contact patch (for two-point contact)
% Fnyr: Lateral normal force on right wheel contact patch (for single-point contact)
% Fnyrt: Lateral normal force on right wheel tread contact patch (for two-point contact)
% Fnyrf: Lateral normal force on right wheel flange contact patch (for two-point contact)
% Fnzr: Vertical normal force on right wheel contact patch (for single-point contact)
% Fnzrt: Vertical normal force on right wheel tread contact patch (for two-point contact)
% Fnzrf: Vertical normal force on right wheel flange contact patch (for two-point contact)
%   Obtaining rolling radii and contact angles at left and right wheel contact patches from
%   functions ’rolling_radius’ and ’contact_angle’

[rlt,rlf,rrt,rrf] = rolling_radius(x1,x1-x5,x6-x1);

[deltalt,deltalf,deltart,deltarf] = contact_angle(x1,x1-x5,x6-x1);

%   Choose set of differential equations to be solved depending on single-point or two-point
%   contact condition at end of previous time step

% Single-point contact equations - Right wheel flange contact and Left wheel tread contact

if (x6-x1) > (yfc+yfctol)

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlt;
    rr = rrf;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalt);
    lambdar = tan(deltarf);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rr)/(2*a);
    dphi = (lambdal+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalt+phi;
    deltar = deltarf-phi;

    % Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

    % Computing longitudinal, lateral, and spin creepages at right wheel flange contact patch

    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);
    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % ’onept_creep’

    [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
    onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);

    % Obtaining normal forces at left and right wheel contact patches from function
    % ’onept_normal’

    [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
    Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

    % The single-point equations in state space form

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-
          rl*tan(deltal))*(Fcyl+Fnyl)-(a-rr*tan(deltar))
          *(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Two-point contact equations - Right wheel two-point contact and Left wheel tread
% contact

elseif (x6-x1) > yfc

    % Assigning variable name to rolling radius at left wheel contact patch

    rl = rlt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalt);
    lambdart = tan(deltart);
    lambdarf = tan(deltarf);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rrt)/(2*a);
    dphi = (lambdal+lambdarf)* (x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle
```

```
deltal = deltalt+phi;
deltatr = deltart-phi;
deltafr = deltarf-phi;

% Computing longitudinal, lateral, and spin creepages at right wheel tread and flange contact
% patches

etaxrt = a*(x4/V)+(1-rrt/r0);
etayrt = ((x3/V)-x2*(rrt/r0)+rrt*dphi/V)/cos(deltatr);
etasprt = ((x4/V)+(phi/r0))*cos(deltatr)+(1/r0)*sin(deltatr);
etaxrf = a*(x4/V)+(1-rrf/r0);
etayrf = ((x3/V)-x2*(rrf/r0)+rrf*dphi/V)/cos(deltafr);
etasprf = ((x4/V)+(phi/r0))*cos(deltafr)+(1/r0)*sin(deltafr);

% Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

etaxl = -a*(x4/V)+(1-rl/r0);
etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

% Obtaining creep forces and moments at left and right wheel contact patches from function
% ’twopt_creep’

[Fcxrt,Fcyrt,Fczrt,Mcxrt,Mcyrt,Mczrt,Fcxrf,Fcyrf,Fczrf,Mcxrf,Mcyrf,Mczrf,Fcxl,Fcyl,Fczl,
Mcxl,Mcyl,Mczl] = twopt_creep(x1,etaxrt,etayrt,etasprt,
etaxrf,etayrf,etasprf,etaxl,etayl,etaspl,deltatr,deltafr,deltal);

% Correcting signs of the vertical creep force components when the vehicle has two-point
% right wheel contact.

Fczrt = -Fczrt;     Fczrf = -Fczrf;          Fczl = -Fczl;
Mcyrt = -Mcyrt;    Mcyrf = -Mcyrf;        Mcyl = -Mcyl;

%Reversing the signs of lateral creep force values to fit in the ’twopt_normal’ function under
% "going right" condition

yfc = -yfc;Fcyrt = -Fcyrt;Fcyrf = -Fcyrf;Fcyl = -Fcyl;

% Reversing the signs of x1, x2, x3, and x4 to allow for mirror effect of right and left
% traverse

x1 = -x1;x2 = -x2;x3 = -x3;x4 = -x4;

% Obtaining normal forces at left and right wheel contact patches from function
% ’twopt_normal’
```

```
    [Fnyrt,Fnzrt,Fnyrf,Fnzrf,Fnyl,Fnzl] = twopt_normal(x1,x2,x3,x4,
    deltatr,deltafr,deltal,rrt,rrf,rl,Fcxrt,Fcyrt,Fczrt,Fcxrf,Fcyrf,Fczrf,
    Fcxl,Fcyl,Fczl,Mcyrt,Mcyrf,Mcyl);

    % Correcting the signs of the lateral normal forces to adjust for the mirror effect and
    % restoring the signs of the lateral creep forces

    Fnyrt = -Fnyrt; Fnyrf = -Fnyrf; Fnyl = -Fnyl; Fcyrt = -Fcyrt; Fcyrf = -Fcyrf; Fcyl = -Fcyl;

    % Restoring the signs of x1, x2, x3, and x4

    x1 = -x1;x2 = -x2;x3 = -x3;x4 = -x4;

    % The two-point equations in state space form for right wheel two-point contact

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyrt+Fcyrf+Fcyl+Fnyrt+Fnyrf+Fnyl+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)+a*(Fcxrt+Fcxrf-Fcxl)+x2*((a-rrt*
          tan(deltatr))*(Fcyrt+Fnyrt)+(a-rrf*tan(deltafr))*(Fcyrf+Fnyrf)-
          (a-rl*tan(deltal))*(Fcyl+Fnyl))+Mczrt+Mczrf+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyrt-Fnyrf-Fcyrt-Fcyrf-krail*x6);

% Single-point contact equations - Right wheel tread contact and Left wheel flange contact

elseif (x1-x5) > (yfc+yfctol)

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlf;
    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalf);
    lambdar = tan(deltart);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rr)/(2*a);
    dphi = (lambdal+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle
    deltal = deltalf+phi;
    deltar = deltart-phi;

    % Computing longitudinal, lateral, and spin creepages at left wheel flange contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

    % Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch
    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % 'onept_creep'

    [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
    onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);

    % Obtaining normal forces at left and right wheel contact patches from function
    % 'onept_normal'

    [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
    Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

    % The single-point equations in state space form

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-rl*tan(deltal))
        *(Fcyl+Fnyl)-(a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Two point contact equations - Right wheel tread contact and Left wheel two-point
% contact

elseif (x1-x5) > yfc

    % Assigning variable name to rolling radius at right wheel contact patch

    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches
```

```
lambdar = tan(deltart);
lambdalt = tan(deltalt);
lambdalf = tan(deltalf);

% Computing wheelset roll angle and rate of change of wheelset roll angle

phi = (rlt-rr)/(2*a);
dphi = (lambdalf+lambdar)*(x3/(2*a));

% Computing effective contact angles at left and right wheel contact patches by
% compensating for wheelset roll angle

deltatl = deltalt+phi;
deltafl = deltalf+phi;
deltar = deltart-phi;

% Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch

etaxr = a*(x4/V)+(1-rr/r0);
etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

% Computing longitudinal, lateral, and spin creepages at left wheel tread and flange contact
% patches

etaxlt = -a*(x4/V)+(1-rlt/r0);
etaylt = ((x3/V)-x2*(rlt/r0)+rlt*dphi/V)/cos(deltatl);
etasplt = ((x4/V)+(phi/r0))*cos(deltatl)-(1/r0)*sin(deltatl);
etaxlf = -a*(x4/V)+(1-rlf/r0);
etaylf = ((x3/V)-x2*(rlf/r0)+rlf*dphi/V)/cos(deltafl);
etasplf = ((x4/V)+(phi/r0))*cos(deltafl)-(1/r0)*sin(deltafl);

% Obtaining creep forces and moments at left and right wheel contact patches from function
% ’twopt_creep’

[Fcxlt,Fcylt,Fczlt,Mcxlt,Mcylt,Mczlt,Fcxlf,Fcylf,Fczlf,Mcxlf,Mcylf,Mczlf,Fcxr,Fcyr,Fczr,M
cxr,Mcyr,Mczr] = twopt_creep(x1,etaxlt,etaylt,etasplt,
etaxlf,etaylf,etasplf,etaxr,etayr,etaspr,deltatl,deltafl,deltar);

% Obtaining normal forces at left and right wheel contact patches from function
% ’twopt_normal’

[Fnylt,Fnzlt,Fnylf,Fnzlf,Fnyr,Fnzr] = twopt_normal(x1,x2,x3,x4,deltatl,
deltafl,deltar,rlt,rlf,rr,Fcxlt,Fcylt,Fczlt,Fcxlf,Fcylf,Fczlf,Fcxr,Fcyr, Fczr,Mcylt,Mcylf,Mcyr);

% The two-point equations in state space form for left wheel two-point contact
```

```
    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcylt+Fcylf+Fcyr+Fnylt+Fnylf+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxlt+Fcxlf-Fcxr)-x2*((a-rlt*
        tan(deltatl))*(Fcylt+Fnylt)+(a-rlf*tan(deltafl))*(Fcylf+Fnylf)-
        (a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczlt+Mczlf+Msuspzw);
    xdot5 = (1/crail)*(-Fnylt-Fnylf-Fcylt-Fcylf-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Single-point contact equations - Right wheel tread contact and Left wheel tread contact

else

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlt;
    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalt);
    lambdar = tan(deltart);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rr)/(2*a);
    dphi = (lambdal+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalt+phi;
    deltar = deltart-phi;

    % Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

    % Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch

    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);
    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % 'onept_creep'

    [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
    onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);

    % Obtaining normal forces at left and right wheel contact patches from function
    % 'onept_normal'

    [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
    Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

   % The single point equations in state space form

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-rl*tan(deltal))
        *(Fcyl+Fnyl)-(a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

end


% M-file name: truck.m
% M-file type: Function file

%  This function file identifies the differential equations for the mathematical model of a two-
%  axle truck and bolster. The front and the rear wheelsets are connected to the truck frame
%  through primary suspension elements. The equations are written in state space form.

%  This function is called by the function file 'equations.m' for solving the differential equations
%  constituting the motion of the truck frame. Inputs to this function are the lateral and yaw
%  displacements and velocities of the wheelsets, the roll angles of the wheelsets, and the lateral
%  and yaw displacements and velocities of the truck frame, and the truck frame suspension
%  forces and moments.

function [ydot13,ydot14,ydot15,ydot16,ydot17,ydot18] =
truck(y13,y14,y15,y16,y17,y18,phi1,phi2,Fsuspyf,Msuspzf,Msuspzb)

% Parameters used for simulation

% mf: Mass of truck frame (kg)
% Ifz: Yaw principal mass moment of inertia of truck frame (kg-m^2)
% mb: Mass of bolster (kg)
% Ibz: Yaw principal mass moment of inertia of bolster (kg-m^2)
% g: Acceleration due to gravity (m/s^2)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'single_truck.m'.

global mf mb Ifz Ibz g;

% The truck frame equations in state space form

ydot13 = y15;
ydot14 = y16;
ydot15 = 1/(mf+mb)*(-0.5*(mf+mb)*g*(phi1+phi2)+Fsuspyf);
ydot16 = 1/Ifz*Msuspzf;
ydot17 = y18;
ydot18 = 1/Ibz*Msuspzb;


% M-file name: onept_creep.m
% M-file type: Function file

%   This function file computes the creep forces and moments acting on the left and the right
%   wheels due to the interaction between wheel and rail. This calculation is valid for a single-
%   point contact condition between the wheel and the rail. The inputs to the function are the
%   longitudinal, lateral, and spin creepages and the contact angles. The outputs are the creep
%   forces and moments resolved in longitudinal, lateral, and vertical directions.

%   Kalker's creep theory is used and spin creep saturation has been taken into account.

%   The outputs given by this function are called by the function file 'wheelset.m'.

function [Fcx1,Fcy1,Fcz1,Mcx1,Mcy1,Mcz1,Fcx2,Fcy2,Fcz2,Mcx2,Mcy2,Mcz2] =
onept_creep(etax1,etay1,etasp1,etax2,etay2,etasp2,delta1,delta2)

% Parameters used for simulation

% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)
% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_truck.m’.

global muN f11 f12 f22 f33;

% Computing the creep forces and moments at the left wheel-rail contact patch

fcpx1 = -f33*etax1;
fcpy1 = -f11*etay1-f12*etasp1;
mcpz1 = f12*etay1-f22*etasp1;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx1^2+fcpy1^2);

if beta<=3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx1 = epsilon*fcpx1;
Fcpy1 = epsilon*fcpy1;
Mcpz1 = epsilon*mcpz1;

% Computing the creep forces and moments at the right wheel-rail contact patch

fcpx2 = -f33*etax2;
fcpy2 = -f11*etay2-f12*etasp2;
mcpz2 = f12*etay2-f22*etasp2;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx2^2+fcpy2^2);

if beta<=3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx2 = epsilon*fcpx2;
Fcpy2 = epsilon*fcpy2;
Mcpz2 = epsilon*mcpz2;
```

```
% The creep forces and moments calculated above are in the contact patch plane. These forces
% and moments are resolved in the longitudinal, lateral, and vertical directions in the track
% coordinate system. The variables calculated below are passed to the function 'wheelset'.

% Resolving creep forces and moments for the left wheel

Fcx1 = Fcpx1;
Fcy1 = Fcpy1*cos(delta1);
Fcz1 = Fcpy1*sin(delta1);
Mcx1 = 0;
Mcy1 = -Mcpz1*sin(delta1);
Mcz1 = Mcpz1*cos(delta1);

% Resolving creep forces and moments for the right wheel

Fcx2 = Fcpx2;
Fcy2 = Fcpy2*cos(delta2);
Fcz2 = -Fcpy2*sin(delta2);
Mcx2 = 0;
Mcy2 = Mcpz2*sin(delta2);
Mcz2 = Mcpz2*cos(delta2);
```

```
% M-file name: twopt_creep.m
% M-file type: Function file

%   This function file computes the creep forces and moments acting on the left and the right
%   wheels due to the interaction between wheel and rail. This calculation is valid for a two-point
%   contact condition between the wheel and the rail. The inputs to the function are the
%   longitudinal, lateral, and spin creepages and the contact angles. The outputs are the creep
%   forces and moments resolved in longitudinal, lateral, and vertical directions.

%   Kalker's creep theory is used and spin creep saturation has been taken into account.

%   The outputs given by this function are called by the function file 'wheelset.m'.

function [Fcx1,Fcy1,Fcz1,Mcx1,Mcy1,Mcz1,Fcx2,Fcy2,Fcz2,Mcx2,Mcy2,Mcz2,Fcx3,
Fcy3,Fcz3,Mcx3,Mcy3,Mcz3] = twopt_creep(x1,etax1,etay1,etasp1,etax2,etay2,
etasp2,etax3,etay3,etasp3,delta1,delta2,delta3)

% Parameters used for simulation

% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'single_truck.m'.

global muN f11 f12 f22 f33;

% Computing the creep forces and moments at contact patch 1. For two-point contact at the left
% wheel, contact patch 1 is the left wheel-rail tread contact patch. For two-point contact at the
% right wheel, contact patch 1 is the right wheel-rail tread contact patch.

fcpx1 = -f33*etax1;
fcpy1 = -f11*etay1-f12*etasp1;
mcpz1 = f12*etay1-f22*etasp1;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx1^2+fcpy1^2);

if beta<3
   epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
   epsilon = (1/beta);
end

Fcpx1 = epsilon*fcpx1;
Fcpy1 = epsilon*fcpy1;
Mcpz1 = epsilon*mcpz1;

% Computing the creep forces and moments at contact patch 2. For two-point contact at the left
% wheel, contact patch 2 is the left wheel-rail flange contact patch. For two-point contact at the
% right wheel, contact patch 2 is the right wheel-rail flange contact patch.

fcpx2 = -f33*etax2;
fcpy2 = -f11*etay2-f12*etasp2;
mcpz2 = f12*etay2-f22*etasp2;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx2^2+fcpy2^2);

if beta<3
   epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
```

```
  epsilon = (1/beta);
end

Fcpx2 = epsilon*fcpx2;
Fcpy2 = epsilon*fcpy2;
Mcpz2 = epsilon*mcpz2;

% Computing the creep forces and moments at contact patch 3. For two-point contact at the left
% wheel, contact patch 3 is the right wheel-rail contact patch. For two-point contact at the right
% wheel, contact patch 3 is the left wheel-rail flange contact patch.

fcpx3 = -f33*etax3;
fcpy3 = -f11*etay3-f12*etasp3;
mcpz3 = f12*etay3-f22*etasp3;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx3^2+fcpy3^2);

if beta<3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx3 = epsilon*fcpx3;
Fcpy3 = epsilon*fcpy3;
Mcpz3 = epsilon*mcpz3;

% The creep forces and moments calculated above are in the contact patch plane. These forces
% and moments are resolved in the longitudinal, lateral, and vertical directions in the track
% coordinate system. The variables calculated below are passed to the function ’wheelset’.

% Resolving creep forces and moments at contact patch 1

Fcx1 = Fcpx1;
Fcy1 = Fcpy1*cos(delta1);
Fcz1 = Fcpy1*sin(delta1);
Mcx1 = 0;
Mcy1 = -Mcpz1*sin(delta1);
Mcz1 = Mcpz1*cos(delta1);

% Resolving creep forces and moments at contact patch 2

Fcx2 = Fcpx2;
Fcy2 = Fcpy2*cos(delta2);
```

```
Fcz2 = Fcpy2*sin(delta2);
Mcx2 = 0;
Mcy2 = -Mcpz2*sin(delta2);
Mcz2 = Mcpz2*cos(delta2);

% Resolving creep forces and moments at contact patch 3

Fcx3 = Fcpx3;
Fcy3 = Fcpy3*cos(delta3);
Fcz3 = -Fcpy3*sin(delta3);
Mcx3 = 0;
Mcy3 = Mcpz3*sin(delta3);
Mcz3 = Mcpz3*cos(delta3);


% M-file name: onept_normal.m
% M-file type: Function file

%   This function file computes the normal forces acting on the left and the right wheels due to
%   the interaction between wheel and rail. This calculation is valid for a single-point contact
%   condition between the wheel and the rail. The inputs to the function are the longitudinal,
%   lateral, and vertical creep forces, the lateral creep moments, rolling radii and contact angles at
%   the left and the right wheel contact patches, and the wheelset yaw displacement and velocity.
%   The outputs are the normal forces on the left and the right wheels resolved in longitudinal,
%   lateral, and vertical directions.

%   The outputs given by this function are called by the function file 'wheelset.m'.

function [Fny1,Fnz1,Fny2,Fnz2] = onept_normal(x2,x4,delta1,delta2,r1,r2,
Fcx1,Fcx2,Fcy1,Fcy2,Fcz1,Fcz2,Mcy1,Mcy2)

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% mw: Mass of wheelset (kg)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% g: Acceleration due to gravity (m/s^2)
% N: Axle load (N)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'single_truck.m'.

global V a r0 mw Iwy g N;
```

% Computing the normal forces at the left and right wheel-rail contact patches

```
F = -Fcz1-Fcz2+mw*g+N;
M = a*(Fcz2-Fcz1)-r1*(Fcy1-x2*Fcx1)-r2*(Fcy2-x2*Fcx2)-x2*(Mcy1+Mcy2)-Iwy*(V/r0)*x4;
vl = F*(a*cos(delta2)-r2*sin(delta2))+M*cos(delta2);
vr = F*(a*cos(delta1)-r1*sin(delta1))-M*cos(delta1);
del1 = 2*a*cos(delta1)*cos(delta2)-r2*cos(delta1)*sin(delta2)-r1*sin(delta1)*cos(delta2);

Fnl = vl/del1;
Fnr = vr/del1;
```

% The normal forces calculated above are normal to the contact patch plane. These forces are
% resolved in the longitudinal, lateral, and vertical directions in the track coordinate system. The
% variables calculated below are passed to the function ’wheelset’.

% Resolving normal forces at the left and right wheel contact patches

```
Fny1 = -Fnl*sin(delta1);
Fnz1 = Fnl*cos(delta1);
Fny2 = Fnr*sin(delta2);
Fnz2 = Fnr*cos(delta2);
```

% M-file name: **twopt_normal.m**
% M-file type: Function file

% This function file computes the normal forces acting on the left and the right wheels due to
% the interaction between wheel and rail. This calculation is valid for a two-point contact
% condition between the wheel and the rail. The inputs to the function are the longitudinal,
% lateral, and vertical creep forces, the lateral creep moments, rolling radii and contact angles at
% the left and the right wheel contact patches, and the wheelset lateral and yaw displacement
% and velocity. The outputs are the normal forces on the left and the right wheels resolved in
% longitudinal, lateral, and vertical directions.

% The outputs given by this function are called by the function file ’wheelset.m’.

```
function [Fny1,Fnz1,Fny2,Fnz2,Fny3,Fnz3] = twopt_normal(x1,x2,x3,x4,delta1,
delta2,delta3,r1,r2,r3,Fcx1,Fcy1,Fcz1,Fcx2,Fcy2,Fcz2,Fcx3,Fcy3,Fcz3,Mcy1,Mcy2,Mcy3)
```

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% mw: Mass of wheelset (kg)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)

```
% g: Acceleration due to gravity (m/s^2)
% yfc: Flange clearance or flange width (m)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% N: Axle load (N)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_truck.m’.

global V a r0 mw Iwy g yfc krail crail N;

% Computing the normal forces at the left and right wheel-rail contact patches

F2 = -Fcz1-Fcz2-Fcz3+N+mw*g;
M1 = -a*(Fcz1+Fcz2-Fcz3)-r1*(Fcy1-x2*Fcx1)-r2*(Fcy2-x2*Fcx2)-r3*(Fcy3-x2*Fcx3)-
x2*(Mcy1+Mcy2+Mcy3)-Iwy*(V/r0)*x4;

Vt = F1*(2*a*cos(delta2)*cos(delta3)-r2*sin(delta2)*cos(delta3)-r3*
cos(delta2)*sin(delta3))+F2*(sin(delta2)*(a*cos(delta3)-r3*sin(delta3)))
+M1*sin(delta2)*cos(delta3);

vf = F1*(-2*a*cos(delta1)*cos(delta3)+r1*sin(delta1)*cos(delta3)+r3*
cos(delta1)*sin(delta3))-F2*(sin(delta1)*(a*cos(delta3)-r3*sin(delta3)))-
M1*sin(delta1)*cos(delta3);

v = F1*(r2*cos(delta1)*sin(delta2)-r1*sin(delta1)*cos(delta2))+F2*
(a*(cos(delta1)*sin(delta2)-sin(delta1)*cos(delta2))+(r2-r1)* sin(delta1)*sin(delta2))-
M1*(cos(delta1)*sin(delta2)-sin(delta1)*
cos(delta2));

del2 = (2*a*cos(delta3)-r3*sin(delta3))*(cos(delta1)*sin(delta2)-sin(delta1)*cos(delta2))+(r2-
r1)*sin(delta1)*sin(delta2)*cos(delta3);

Fnt = vt/del2;
Fnf = vf/del2;
Fn = v/del2;

% The normal forces calculated above are normal to the contact patch plane. These forces are
% resolved in the longitudinal, lateral, and vertical directions in the track coordinate system. The
% variables calculated below are passed to the function ’wheelset’.

% Resolving normal forces at the left and right wheel contact patches

Fny1 = -Fnt*sin(delta1);
Fnz1 = Fnt*cos(delta1);
```

```
Fny2 = -Fnf*sin(delta2);
Fnz2 = Fnf*cos(delta2);
Fny3 = Fn*sin(delta3);
Fnz3 = Fn*cos(delta3);


% M-file name: wheelset_suspension.m
% M-file type: Function file

%   This function file computes the suspension forces acting on the front and the rear wheelsets
%   due to the primary suspension components. The primary suspension consists of longitudinal
%   and lateral springs and dampers. The inputs to the function are the wheelset lateral and yaw
%   displacements and velocities. The outputs are the lateral suspension force and the yaw
%   suspension moment on the wheelset.

%   The outputs given by this function are called by the function file 'equations.m'.

function [Fsusp1,Fsusp2,Msusp1] = wheelset_suspension(y1,y3,y2,y4,y13,y14,y15,y16)

% Parameters used for simulation

% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-sec/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-sec/m)
% dp: Half of lateral distance between primary longitudinal springs (m)
% b: Half of wheelbase (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'single_truck.m'.

global kpx cpx kpy cpy dp b;

% Computing the suspension forces and moments acting on the wheelset. These variables are
% passed to the function 'equations'.

Fsusp1 = -2*kpy*y1+2*kpy*y13+2*b*kpy*y14-2*cpy*y3+2*cpy*y15+2*b*cpy*y16;
Fsusp2 = -2*kpy*y1+2*kpy*y13-2*b*kpy*y14-2*cpy*y3+2*cpy*y15-2*b*cpy*y16;
Msusp1 = -2*dp^2*kpx*y2+2*dp^2*kpx*y14-2*dp^2*cpx*y4+2*dp^2*cpx*y16;


% M-file name: truck_suspension.m
% M-file type: Function file
%  This function file computes the suspension forces acting on the truck frame and bolster due
%  to the primary and secondary suspension components. The primary suspension consists of
%  longitudinal and lateral springs and dampers. The secondary suspension consists of a lateral
%  spring and damper, a yaw spring and damper, and a Coulomb friction arrangement. The
%  inputs to the function are the wheelset lateral and yaw displacements and velocities, the truck
%  frame lateral and yaw displacements and velocities, and the bolster yaw displacement and
%  velocity. The outputs are the lateral suspension force and the vertical suspension moment on
%  the truck frame, and the vertical suspension moment on the bolster.

%  The outputs given by this function are called by the function file ’equations.m’.

function [Fsusp1,Msusp1,Msusp2] =
truck_suspension(x1,x2,x3,x4,x7,x8,x9,x10,x13,x14,x15,x16,x17,x18)

% Parameters used for simulation

% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-sec/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-sec/m)
% dp: Half of lateral distance between primary longitudinal springs (m)
% b: Half of wheelbase (m)
% ksy: Secondary lateral stiffness coefficient (N/m)
% csy: Secondary lateral damping coefficient (N-sec/m)
% ksphi: Secondary yaw stiffness coefficient (N/rad)
% csphi: Secondary yaw damping coefficient (N-sec/rad)
% T0: Centerplate breakaway torque (N-m)
% C0: Coulomb viscous yaw damping coefficient (N-sec/rad)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_truck.m’.

global kpx cpx kpy cpy dp b ksy csy ksphi csphi T0 C0;

% Computing the suspension forces and moments acting on the truck frame and bolster. These
% variables are passed to the function ’equations’.

% Truck frame / Bolster Coulomb friction characteristic

if (x16-x18) >= T0/C0
  Tcoul = T0;
elseif (x16-x18) <= -T0/C0
   Tcoul = -T0;
else
   Tcoul = C0*(x16-x18);
```

```
end

Fsusp1 = 2*kpy*x1+2*kpy*x7+(-4*kpy-2*ksy)*x13+2*cpy*x3+2*cpy*x9+(-4*cpy-
2*csy)*x15;
Msusp1 = 2*b*kpy*x1+2*dp^2*kpx*x2-2*b*kpy*x7+2*dp^2*kpx*x8+(-4*dp^2*kpx-
4*b^2*kpy)*x14+2*b*cpy*x3+2*dp^2*cpx*x4-2*b*cpy*x9+2*dp^2*cpx*x10+(-4*dp^2*cpx-
4*b^2*cpy)*x16-Tcoul;
Msusp2 = -ksphi*x17-csphi*x18+Tcoul;
```

```
% M-file name: rolling_radius.m
% M-file type: Function file

%   This function file computes the rolling radii at the left and the right wheel-rail contact
%   patches as a function of the relative lateral displacement between the left and the right wheels
%   and rails. The inputs to the function are the relative lateral displacement between the left
%   wheel and the left rail and the right wheel and the right rail. The outputs are the rolling radii
%   at the left and the right tread and flange contact patches.

%   The outputs given by this function are called by the function file ’wheelset.m’.

function [rlefttread,rleftflange,rrighttread,rrightflange] = rolling_radius(y,yl,yr)

% Parameters used for simulation

% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)
% r0: Centered rolling radius of the wheel (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_truck.m’.

global yfc yfctol r0;

% Computing the rolling radii at the left and the right wheel-rail contact patches. These variables
% are passed to the function ‘wheelset’.

if y >= 0

    % Wheelset displacement is in the positive (or left) direction from the centerline (Right
    %   wheel tread contact)

    % Note that for a left movement of the wheelset from the centerline, the
    % right wheel does not flange. Hence, the zero value for the right wheel
    % flange radius is not real and is not used in calculations.
    rrighttread = r0 + 0.125*(yr);
    rrightflange = 0.000;

    if yl <= yfc

    % Left wheel tread contact. Note that the zero value for the left wheel flange radius is not
    % real and is not used in calculations.

    rlefttread = r0 + 0.125*yl;
    rleftflange = 0.000;

    elseif yl < (yfc+yfctol)

    % Left wheel flange contact

    rlefttread = r0+0.125*yl;
    rleftflange = 0.3566+13700*(yl-0.008)^2;

    else

    % Left wheel two-point contact

    rlefttread = r0+0.125*yl;
    rleftflange = 0.3703+2.5833*(yl-0.009);

    end

else

    % Wheelset displacement is in the negative (or right) direction from the centerline (Left
    % wheel tread contact)

    % Note that for a right movement of the wheelset from the centerline, the
    % left wheel does not flange. Hence, the zero value for the left wheel
    % flange radius is not real and is not used in calculations.

    rlefttread = r0 + 0.125*yl;
    rleftflange = 0.000;

    if yr <= yfc

    % Right wheel tread contact. Note that the zero value for the right
    % wheel flange radius is not real and is not used in calculations.

    rrighttread = r0 + 0.125*(yr);
    rrightflange = 0.000;
    elseif yr < (yfc+yfctol)

    % Right wheel flange contact

    rrighttread = r0 + 0.125*yr;
    rrightflange = 0.3566+13700*(yr-0.008)^2;

    else

  % Right wheel two-point contact

    rrighttread = r0+0.125*yr;
    rrightflange = 0.3703+2.5833*(yr-0.009);

    end

end


% M-file name: contact_angle.m
% M-file type: Function file

%   This function file computes the contact angles at the left and the right wheel-rail contact
%   patches as a function of the relative lateral displacement between the left and the right wheels
%   and rails. The inputs to the function are the relative lateral displacement between the left
%   wheel and the left rail and the right wheel and the right rail. The outputs are the contact
%   angles at the left and the right tread and flange contact patches.

%   The outputs given by this function are called by the function file ’wheelset.m’.

function [calefttread,caleftflange,carighttread,carightflange] = contact_angle(y,yl,yr)

% Parameters used for simulation

% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’single_truck.m’.

global yfc yfctol;

% Computing the rolling radii at the left and the right wheel-rail contact patches. These variables
% are passed to the function ‘wheelset’.
```

```
if y >= 0

    % Wheelset displacement is in the positive (or left) direction from the centerline (Right
    % wheel tread contact)

    % Note that for a left movement of the wheelset from the centerline, the
    % right wheel does not flange. Hence, the zero value for the right wheel
    % flange angle is not real and is not used in calculations.

    carighttread = atan(0.125);
    carightflange = 0.000;

    if yl <= yfc


    % Left wheel tread contact. Note that the zero value for the left wheel flange angle is not real
    % and is not used in calculations.


    calefttread = atan(0.125);
    caleftflange = 0.000;

    elseif yl < (yfc+yfctol)

    % Left wheel flange contact

    calefttread = atan(0.125);
    caleftflange = atan(0.125+2648*(yl-0.008));

    else

    % Left wheel two-point contact

    calefttread = atan(0.125);
    caleftflange = atan(2.748);

    end

else

    % Wheelset displacement is in the negative (or right) direction from the centerline (Left
    % wheel tread contact)

    % Note that for a right movement of the wheelset from the centerline, the
    % left wheel does not flange. Hence, the zero value for the left wheel
    % flange angle is not real and is not used in calculations.

    calefttread = atan(0.125);
    caleftflange = 0.000;

    if yr <= yfc

    % Right wheel tread contact. Note that the zero value for the right
    % wheel flange angle is not real and is not used in calculations.

    carighttread = atan(0.125);
    carightflange = 0.000;

    elseif yr < (yfc+yfctol)

    % Right wheel flange contact

    carighttread = atan(0.125);
    carightflange = atan(0.125+2648*(yr-0.008));

    else

    % Right wheel two-point contact

    carighttread = atan(0.125);
    carightflange = atan(2.748);

    end

end
```

PROGRAM FOR SIMULATING DYNAMIC BEHAVIOR OF A RAIL VEHICLE

```
% M-file name: full_vehicle.m
% M-file type: Script (main) file

%  This program simulates the dynamic behavior of a complete rail vehicle moving on a straight
%  track. The rail vehicle model consists of a front and a rear conventional truck, a front and a
%  rear bolster, and a carbody. The model has a total of 42 degrees of freedom. Each
%  conventional truck consists of a leading and a trailing wheelset (with independent degrees of
%  freedom) connected to the truck frame through primary suspension elements. Each truck
%  frame is connected to a bolster through a Coulomb friction element. The bolster and the truck
%  frame share the same lateral degree of freedom, but have independent yaw degrees of
%  freedom. Each bolster is connected to the carbody through secondary lateral, yaw, and
%  vertical springs and dampers.

% Initial conditions

% x10: Initial Lateral Displacement of leading front wheelset (m)
% x20: Initial Yaw Displacement of leading front wheelset (rad)
% x30: Initial Lateral Velocity of leading front wheelset (m/sec)
% x40: Initial Yaw Velocity of leading front wheelset (rad/sec)
% x50: Initial Lateral Displacement of left leading front rail (m)
% x60: Initial Lateral Displacement of right leading front rail (m)
% x70: Initial Lateral Displacement of trailing front wheelset (m)
% x80: Initial Yaw Displacement of trailing front wheelset (rad)
% x90: Initial Lateral Velocity of trailing front wheelset (m/sec)
% x100: Initial Yaw Velocity of trailing front wheelset (rad/sec)
% x110: Initial Lateral Displacement of left trailing front rail (m)
% x120: Initial Lateral Displacement of right trailing front rail (m)
% x130: Initial Lateral Displacement of front truck frame (m)
% x140: Initial Yaw Displacement of front truck frame (rad)
% x150: Initial Lateral Velocity of front truck frame (m/sec)
% x160: Initial Yaw Velocity of front truck frame (rad/sec)
% x170: Initial Yaw Displacement of front bolster (rad)
% x180: Initial Yaw Velocity of front bolster (rad/sec)
% x190: Initial Lateral Displacement of leading rear wheelset (m)
% x200: Initial Yaw Displacement of leading rear wheelset (rad)
% x210: Initial Lateral Velocity of leading rear wheelset (m/sec)
% x220: Initial Yaw Velocity of leading rear wheelset (rad/sec)
% x230: Initial Lateral Displacement of left leading rear rail (m)
% x240: Initial Lateral Displacement of right leading rear rail (m)
% x250: Initial Lateral Displacement of trailing rear wheelset (m)
% x260: Initial Yaw Displacement of trailing rear wheelset (rad)
% x270: Initial Lateral Velocity of trailing rear wheelset (m/sec)
% x280: Initial Yaw Velocity of trailing rear wheelset (rad/sec)
% x290: Initial Lateral Displacement of left trailing rear rail (m)
% x300: Initial Lateral Displacement of right trailing rear rail (m)
% x310: Initial Lateral Displacement of rear truck frame (m)
% x320: Initial Yaw Displacement of rear truck frame (rad)
% x330: Initial Lateral Velocity of rear truck frame (m/sec)
% x340: Initial Yaw Velocity of rear truck frame (rad/sec)
% x350: Initial Yaw Displacement of rear bolster (rad)
% x360: Initial Yaw Velocity of rear bolster (rad/sec)
% x370: Initial Lateral Displacement of carbody (m)
% x380: Initial Yaw Displacement of carbody (rad)
% x390: Initial Roll Displacement of carbody (rad)
% x400: Initial Lateral Velocity of carbody (m/sec)
% x410: Initial Yaw Velocity of carbody (rad/sec)
% x420: Initial Roll Velocity of carbody (rad/sec)

x10=0.00635;x20=0.0010;x30=0.0100;x40=0.0000;x50=0.0000;x60=0.0000;x70=0.0030;x80=0.
0010;x90=0.0100;x100=0.0000;x110=0.0000;x120=0.0000;x130=0.0000;x140=0.0010;x150=0.
1000;x160=0.0000;x170=0.0000;x180=0.0000;x190=0.00635;x200=0.0010;x210=0.0100;x220=
0.0000;x230=0.0000;x240=0.0000;x250=0.0030;x260=0.0010;x270=0.0100;x280=0.0000;x290
=0.0000;x300=0.0000;x310=0.0000;x320=0.0010;x330=0.10000;x340=0.0000;x350=0.0000;x3
60=0.0000;x370=0.0000;x380=0.0000;x390=0.0000;x400=0.0000;x410=0.0000;x420=0.0000;

% Globalizing all the variables

% Values for global variables need to be specified in the main file alone even though they may
% be used in several function files.

% Global variables

% V: Forward velocity of wheelset (m/sec)
% lambda: Wheel conicity
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation
% mw: Mass of wheelset (kg)
% Iwz: Yaw principal mass moment of inertia of wheelset (kg-m^2)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% g: Acceleration due to gravity (m/s^2)
% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)
% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-s/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-s/m)
% dp: Half of Lateral distance between primary longitudinal springs (m)
% mf: Mass of truck frame (kg)
% mb: Mass of bolster (kg)
% Ifz: Yaw principal mass moment of inertia of truck frame (kg-m^2)
% Ibz: Yaw principal mass moment of inertia of bolster (kg-m^2)
% b: Half of wheelbase (m)
% N: Axle load (N)
% ksy: Secondary lateral stiffness coefficient (N/m)
% csy: Secondary lateral damping coefficient (N-sec/m)
% ksphi: Secondary yaw stiffness coefficient (N/rad)
% csphi: Secondary yaw damping coefficient (N-sec/rad)
% T0: Centerplate breakaway torque (N-m)
% C0: Coulomb viscous yaw damping coefficient (N-sec/rad)
% mc: Mass of carbody (kg)
% Icx: Roll principal mass moment of inertia of carbody (kg-m^2)
% Icz: Yaw principal mass moment of inertia of carbody (kg-m^2)
% ksz: Secondary vertical stiffness coefficient (N/m)
% csz: Secondary vertical damping coefficient (N-sec/m)
% ds: Half of Lateral distance between secondary vertical springs (m)
% ls: Longitudinal distance between C.G. of carbody to C.G. of either truck (m)
% hcs: Vertical distance between C.G. of carbody to secondary lateral springs (m)

global V lambda a r0 yfc yfctol mw Iwz Iwy krail crail g muN f11 f12 f22 f33 kpx cpx kpy cpy
dp mf mb Ifz Ibz b N ksy csy ksphi csphi T0 C0 mc Icx Icz ksz csz ds ls hcs;

V=20;lambda=0.125;a=0.716;r0=0.3556;yfc=0.0080;yfctol=0.0010;mw=1751;Iwz=761;Iwy=13
0;krail=14.6e7;crail=14.6e4;g=9.81;muN=12000;f11=9430000;f12=1.2e3;
f22=1e3;f33=10230000;kpx=9.12e5;cpx=8376.9;kpy=5.84e5;cpy=9048.2;dp=0.61;
mf=4041;mb=365;Ifz=3371;Ibz=337;b=1.295;N=100000;ksy=3.5e5;csy=1.75e4;
ksphi=3.8e8;csphi=2.5e7;T0=10168;C0=3.5e7;mc=31900;Icx=1.185e5;Icz=2.414e6;
ksz=3.8e5;csz=2.5e4;ds=1.13;ls=7.239;hcs=0.88;

% Specify initial time, final time, and number of time steps

tspan=linspace(0,10,10000);

% Solve the system of differential equations

% The system of differential equations is integrated from t=0 sec to t=10 sec with the above
% initial conditions. The function file ‘equations.m’ points to the system of different equations
% which are contained within the function files ‘wheelset.m’, ‘truck.m’, and ‘carbody.m’
```

```
[t,x] = ode23(’equations’,tspan,[x10;x20;x30;x40;x50;x60;x70;x80;x90;x100;
x110;x120;x130;x140;x150;x160;x170;x180;x190;x200;x210;x220;x230;x240;x250;
x260;x270;x280;x290;x300;x310;x320;x330;x340;x350;x360;x370;x380;x390;x400;
x410;x420]);

% Plotting time response

% Plotting lateral and yaw displacements for the leading front wheelset vs. time

subplot(2,1,1),plot(t,x(:,1),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of leading front wheelset’)
title(’Lateral Displacement vs Time for leading front wheelset’)
grid on
subplot(2,1,2),plot(t,x(:,2),’r’)
xlabel(’Time’)
ylabel(’Yaw Displacement of leading front wheelset’)
title(’Yaw Displacement vs Time for leading front wheelset’)
grid on

figure

% Plotting lateral and yaw displacements for the trailing front wheelset vs. time

subplot(2,1,1),plot(t,x(:,7),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of trailing front wheelset’)
title(’Lateral Displacement vs Time for trailing front wheelset’)
grid on
subplot(2,1,2),plot(t,x(:,8),’r’)
xlabel(’Time’)
ylabel(’Yaw Displacement of trailing front wheelset’)
title(’Yaw Displacement vs Time for trailing front wheelset’)
grid on

figure

% Plotting lateral and yaw displacements for the front truck frame vs. time

subplot(2,1,1),plot(t,x(:,13),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of front truck frame’)
title(’Lateral Displacement vs Time for front truck frame’)
grid on
subplot(2,1,2),plot(t,x(:,14),’r’)
xlabel(’Time’)
```

```
ylabel(’Yaw Displacement of front truck frame’)
title(’Yaw Displacement vs Time for front truck frame’)
grid on

figure

% Plotting lateral and yaw displacements for the leading rear wheelset vs. time

subplot(2,1,1),plot(t,x(:,19),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of leading rear wheelset’)
title(’Lateral Displacement vs Time for leading rear wheelset’)
grid on
subplot(2,1,2),plot(t,x(:,20),’r’)
xlabel(’Time’)
ylabel(’Yaw Displacement of leading rear wheelset’)
title(’Yaw Displacement vs Time for leading rear wheelset’)
grid on

figure

% Plotting lateral and yaw displacements for the trailing rear wheelset vs. time

subplot(2,1,1),plot(t,x(:,25),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of trailing rear wheelset’)
title(’Lateral Displacement vs Time for trailing rear wheelset’)
grid on
subplot(2,1,2),plot(t,x(:,26),’r’)
xlabel(’Time’)
ylabel(’Yaw Displacement of trailing rear wheelset’)
title(’Yaw Displacement vs Time for trailing rear wheelset’)
grid on

figure

% Plotting lateral and yaw displacements for the rear truck frame vs. time

subplot(2,1,1),plot(t,x(:,31),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of rear truck frame’)
title(’Lateral Displacement vs Time for rear truck frame’)
grid on
subplot(2,1,2),plot(t,x(:,32),’r’)
xlabel(’Time’)
ylabel(’Yaw Displacement of rear truck frame’)
```

```
title(’Yaw Displacement vs Time for rear truck frame’)
grid on

figure

% Plotting lateral, yaw, and roll displacements for the carbody vs. time

subplot(3,1,1),plot(t,x(:,37),’r’)
xlabel(’Time’)
ylabel(’Lateral Displacement of carbody’)
title(’Lateral Displacement vs Time for carbody’)
grid on
subplot(3,1,2),plot(t,x(:,38),’r’)
xlabel(’Time’)
ylabel(’Yaw Displacement of carbody’)
title(’Yaw Displacement vs Time for carbody’)
grid on
subplot(3,1,3),plot(t,x(:,39),’r’)
xlabel(’Time’)
ylabel(’Roll Displacement of carbody’)
title(’Roll Displacement vs Time for carbody’)
grid on


% M-file name: equations.m
% M-file type: Function file

%   This function file obtains suspension forces and moments from function files
%   ‘wheelset_suspension.m’, ‘truck_suspension.m’, and ‘carbody_suspension’, and uses them to
%   solve the wheelset, truck, and carbody dynamic equations by invoking the functions
%   ‘wheelset’, ‘truck’, and ‘carbody’.

%   At the end of the simulation, MATLAB stores each degree of freedom as a column array and
%   the entire solution as a matrix. The solution matrix in this program has been named 'x'.
%   Hence, in this case, the solution matrix 'x' will have 42 columns. The number of rows of 'x'
%   will be equal to the total number of time steps required for the simulation. Throughout this
%   function file, the sixteen degrees of freedom at any particular time step are denoted as:

%   x(1): Lateral Displacement of leading front wheelset
%   x(2): Yaw Displacement of leading front wheelset
%   x(3): Lateral Velocity of leading front wheelset
%   x(4): Yaw Velocity of leading front wheelset
%   x(5): Lateral Displacement of left leading front rail
%   x(6): Lateral Displacement of right leading front rail
%   x(7): Lateral Displacement of trailing front wheelset
%   x(8): Yaw Displacement of trailing front wheelset
%   x(9): Lateral Velocity of trailing front wheelset
%   x(10): Yaw Velocity of trailing front wheelset
%   x(11): Lateral Displacement of left trailing front rail
%   x(12): Lateral Displacement of right trailing front rail
%   x(13): Lateral Displacement of front truck frame
%   x(14): Yaw Displacement of front truck frame
%   x(15): Lateral Velocity of front truck frame
%   x(16): Yaw Velocity of front truck frame
%   x(17): Yaw Displacement of front bolster
%   x(18): Yaw Velocity of front bolster
%   x(19): Lateral Displacement of leading rear wheelset
%   x(20): Yaw Displacement of leading rear wheelset
%   x(21): Lateral Velocity of leading rear wheelset
%   x(22): Yaw Velocity of leading rear wheelset
%   x(23): Lateral Displacement of left leading rear rail
%   x(24): Lateral Displacement of right leading rear rail
%   x(25): Lateral Displacement of trailing rear wheelset
%   x(26): Yaw Displacement of trailing rear wheelset
%   x(27): Lateral Velocity of trailing rear wheelset
%   x(28): Yaw Velocity of trailing rear wheelset
%   x(29): Lateral Displacement of left trailing rear rail
%   x(30): Lateral Displacement of right trailing rear rail
%   x(31): Lateral Displacement of rear truck frame
%   x(32): Yaw Displacement of rear truck frame
%   x(33): Lateral Velocity of rear truck frame
%   x(34): Yaw Velocity of rear truck frame
%   x(35): Yaw Displacement of rear bolster
%   x(36): Yaw Velocity of rear bolster
%   x(37): Lateral Displacement of carbody
%   x(38): Yaw Displacement of carbody
%   x(39): Roll Displacement of carbody
%   x(40): Lateral Velocity of carbody
%   x(41): Yaw Velocity of carbody
%   x(42): Roll Velocity of carbody

%   where x(n) represents the nth column of the solution matrix ’x’.
%   The vector of time-derivatives within any time step for the solution vector ’x’ has been named
%   ’xdot’. Hence, the time-derivative of variable x(n) would  be xdot(n).

function [xdot] = equations(t,x)

% Nomenclature:

% Fsusp1w1: Lateral suspension force on the leading front wheelset
% Fsusp1w2: Lateral suspension force on the trailing front wheelset
% Fsusp1w3: Lateral suspension force on the leading rear wheelset
% Fsusp1w4: Lateral suspension force on the trailing rear wheelset
% Msusp1w1: Vertical suspension moment on the leading front wheelset
% Msusp1w2: Vertical suspension moment on the trailing front wheelset
% Msusp1w3: Vertical suspension moment on the leading rear wheelset
% Msusp1w4: Vertical suspension moment on the trailing rear wheelset
% Fsuspt1: Lateral suspension force on the front truck frame
% Fsuspt2: Lateral suspension force on the rear truck frame
% Msuspt1: Vertical suspension moment on the front truck frame
% Msuspt2: Vertical suspension moment on the rear truck frame
% Msuspb1: Vertical suspension moment on the front bolster
% Msuspb2: Vertical suspension moment on the rear bolster
% Fsuspyc1: Lateral suspension force on the carbody due to front truck
% Fsuspyc2: Lateral suspension force on the carbody due to rear truck
% Msuspzc1: Vertical suspension moment on the carbody due to front truck
% Msuspzc2: Vertical suspension moment on the carbody due to rear truck
% Msuspxc1: Longitudinal suspension moment on the carbody due to front truck
% Msuspxc2: Longitudinal suspension moment on the carbody due to rear truck
% phiw1: Roll angle of leading front wheelset
% phiw2: Roll angle of trailing front wheelset
% phiw3: Roll angle of leading rear wheelset
% phiw4: Roll angle of trailing rear wheelset

% Initializing the vector of time-derivatives

xdot = zeros(42,1);

% Printing time at end of each time-step on the command screen

t

% Obtaining leading front wheelset suspension forces from function ’wheelset_suspension’. Note
% that the variable Fsusp1w2 is merely a dummy and is not passed to the function ‘wheelset’.

[Fsusp1w1,Fsusp1w2,Msusp1w1] =
wheelset_suspension(x(1),x(3),x(2),x(4),x(13),x(14),x(15),x(16));

% The function ‘wheelset’ is invoked in order to solve the differential equations for the leading
% front wheelset.

[xdot(1),xdot(2),xdot(3),xdot(4),xdot(5),xdot(6),phiw1] =
wheelset(x(1),x(2),x(3),x(4),x(5),x(6),Fsusp1w1,Msusp1w1);

% Obtaining trailing front wheelset suspension forces from function 'wheelset_suspension'. Note
% that the variable Fsusp1w1 is merely a dummy and is not passed to the function ‘wheelset’.
```

```
[Fsusp1w1,Fsusp1w2,Msusp1w2] =
wheelset_suspension(x(7),x(9),x(8),x(10),x(13),x(14),x(15),x(16));

% The function ‘wheelset’ is invoked in order to solve the differential equations for the trailing
% front wheelset.

[xdot(7),xdot(8),xdot(9),xdot(10),xdot(11),xdot(12),phiw2] =
wheelset(x(7),x(8),x(9),x(10),x(11),x(12),Fsusp1w2,Msusp1w2);

% Obtaining front truck frame and bolster suspension forces from function 'truck_suspension'.
% Note that the variable Fsuspt2 is merely a dummy and is not passed to the function ‘truck’.

[Fsuspt1,Fsuspt2,Msuspt1,Msuspb1]=truck_suspension(x(1),x(2),x(3),x(4),x(7),
x(8),x(9),x(10),x(13),x(14),x(15),x(16),x(37),x(38),x(40),x(41),x(17),x(18));

% The function ‘truck’ is invoked in order to solve the differential equations for the front truck
% frame.

[xdot(13),xdot(14),xdot(15),xdot(16),xdot(17),xdot(18)]=truck(x(13),x(14),
x(15),x(16),x(17),x(18),phiw1,phiw2,Fsuspt1,Msuspt1,Msuspb1);

% Obtaining leading rear wheelset suspension forces from function 'wheelset_suspension'. Note
% that the variable Fsusp1w4 is merely a dummy and is not passed to the function ‘wheelset’.

[Fsusp1w3,Fsusp1w4,Msusp1w3] =
wheelset_suspension(x(19),x(21),x(20),x(22),x(31),x(32),x(33),x(34));

% The function ‘wheelset’ is invoked in order to solve the differential equations for the leading
% rear wheelset.

[xdot(19),xdot(20),xdot(21),xdot(22),xdot(23),xdot(24),phiw3] =
wheelset(x(19),x(20),x(21),x(22),x(23),x(24),Fsusp1w3,Msusp1w3);

% Obtaining trailing rear wheelset suspension forces from function 'wheelset_suspension'. Note
% that the variable Fsusp1w3 is merely a dummy and is not passed to the function ‘wheelset’.

[Fsusp1w3,Fsusp1w4,Msusp1w4] =
wheelset_suspension(x(25),x(27),x(26),x(28),x(31),x(32),x(33),x(34));

% The function ‘wheelset’ is invoked in order to solve the differential equations for the trailing
% rear wheelset.

[xdot(25),xdot(26),xdot(27),xdot(28),xdot(29),xdot(30),phiw4] =
wheelset(x(25),x(26),x(27),x(28),x(29),x(30),Fsusp1w4,Msusp1w4);
```

```
% Obtaining rear truck frame and bolster suspension forces from function ’truck_suspension’.
% Note that the variable Fsuspt1 is merely a dummy and is not passed to the function ‘truck’.

[Fsuspt1,Fsuspt2,Msuspt2,Msuspb2]=truck_suspension(x(19),x(20),x(21),x(22),
x(25),x(26),x(27),x(28),x(31),x(32),x(33),x(34),x(37),x(38),x(40),x(41),x(35),x(36));

% The function ‘truck’ is invoked in order to solve the differential equations for the rear truck
% frame.

[xdot(31),xdot(32),xdot(33),xdot(34),xdot(35),xdot(36)]=truck(x(31),x(32),
x(33),x(34),x(35),x(36),phiw3,phiw4,Fsuspt2,Msuspt2,Msuspb2);

% Obtaining carbody suspension forces from function 'carbody_suspension'.

[Fsuspyc1,Fsuspyc2,Msuspzc1,Msuspzc2,Msuspxc1,Msuspxc2] =
carbody_suspension(x(13),x(15),x(17),x(18),x(31),x(33),x(35),x(36),x(37),
x(38),x(39),x(40),x(41),x(42));

% The function ‘carbody’ is invoked in order to solve the differential equations for the carbody.

[xdot(37),xdot(38),xdot(39),xdot(40),xdot(41),xdot(42)] =
carbody(x(39),x(40),x(41),x(42),Fsuspyc1,Fsuspyc2,Msuspzc1,Msuspzc2,Msuspxc1,Msuspxc2);


% M-file name: wheelset.m
% M-file type: Function file

%  This function file identifies the single-point (tread contact) and the two-point (tread and
%  flange contact) equations that constitute the mathematical model for the dynamics of a single
%  wheelset rolling on a straight track. The equations are written in state space form.

%  This function is called by the function file ‘equations.m’ for solving the differential equations
%  constituting the motion of wheelset. The wheelset lateral and yaw displacements and
%  velocities, and the left and right rail lateral displacements are provided as inputs to this
%  function in order to solve for the wheelset. Additionally, the wheelset suspension forces and
%  moments are provided as inputs.

%  This function calls the following functions -
%  'rolling_radius', 'contact_angle', 'onept_creep', 'onept_normal', 'twopt_creep', and
%  'twopt_normal'.

function [xdot1,xdot2,xdot3,xdot4,xdot5,xdot6,phi] =
wheelset(x1,x2,x3,x4,x5,x6,Fsuspyw,Msuspzw)

% Parameters used for simulation
% V: Forward velocity of wheelset (m/sec)
% lambda: Wheel conicity
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)
% mw: Mass of wheelset (kg)
% Iwz: Yaw principal mass moment of inertia of wheelset (kg-m^2)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% g: Acceleration due to gravity (m/s^2)

%   Indicating the global nature of the variables. This means that the value of the variables need
%  not be specified in this function file. This value is automatically obtained from the main file
%  ’full_vehicle.m’.

global V lambda a r0 yfc yfctol mw Iwz Iwy krail crail g;

% Nomenclature:

% rlt: Rolling radius at left wheel tread contact patch
% rlf: Rolling radius at left wheel flange contact patch
% rrt: Rolling radius at right wheel tread contact patch
% rrf: Rolling radius at right wheel flange contact patch
% deltalt: Contact angle at left wheel tread contact patch
% deltalf: Contact angle at left wheel flange contact patch
% deltart: Contact angle at right wheel tread contact patch
% deltarf: Contact angle at right wheel flange contact patch
% rl: Rolling radius at left wheel contact patch (for single-point contact)
% rr: Rolling radius at right wheel contact patch (for single-point contact)
% deltal: Effective contact angle at left wheel contact patch after compensating for wheelset roll
% angle (for single-point contact)
% deltatl: Effective contact angle at left wheel tread contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltafl: Effective contact angle at left wheel flange contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltar: Effective contact angle at right wheel contact patch after compensating for wheelset
% roll angle (for single-point contact)
% deltatr: Effective contact angle at right wheel tread contact patch after compensating for
% wheelset roll angle (for two-point contact)
% deltafr: Effective contact angle at right wheel flange contact patch after compensating for
% wheelset roll angle (for two-point contact)
% lambdal: Slope of tangent at left wheel contact patch (for single-point contact)
% lambdalt: Slope of tangent at left wheel tread contact patch (for two-point contact)
% lambdalf: Slope of tangent at left wheel flange contact patch (for two-point contact)
% lambdar: Slope of tangent at right wheel contact patch (for single-point contact)
% lambdart: Slope of tangent at right wheel tread contact patch (for two-point contact)
% lambdarf: Slope of tangent at right wheel flange contact patch (for two-point contact)
% phi: Wheelset roll angle
% dphi: Rate of change of wheelset roll angle
% etaxl: Longitudinal creepage at left wheel contact patch (for single-point contact)
% etaxlt: Longitudinal creepage at left wheel tread contact patch (for two-point contact)
% etaxlf: Longitudinal creepage at left wheel flange contact patch (for two-point contact)
% etayl: Lateral creepage at left wheel contact patch (for single-point contact)
% etaylt: Lateral creepage at left wheel tread contact patch (for two-point contact)
% etaylf: Lateral creepage at left wheel flange contact patch (for two-point contact)
% etaspl: Spin creepage at left wheel contact patch (for single-point contact)
% etasplt: Spin creepage at left wheel tread contact patch (for two-point contact)
% etasplf: Spin creepage at left wheel flange contact patch (for two-point contact)
% etaxr: Longitudinal creepage at right wheel contact patch (for single-point contact)
% etaxrt: Longitudinal creepage at right wheel tread contact patch (for two-point contact)
% etaxrf: Longitudinal creepage at right wheel flange contact patch (for two-point contact)
% etayr: Lateral creepage at right wheel contact patch (for single-point contact)
% etayrt: Lateral creepage at right wheel tread contact patch (for two-point contact)
% etayrf: Lateral creepage at right wheel flange contact patch (for two-point contact)
% etaspr: Spin creepage at right wheel contact patch (for single-point contact)
% etasprt: Spin creepage at right wheel tread contact patch (for two-point contact)
% etasprf: Spin creepage at right wheel flange contact patch (for two-point contact)
% Fcxl: Longitudinal creep force on left wheel contact patch (for single-point contact)
% Fcxlt: Longitudinal creep force on left wheel tread contact patch (for two-point contact)
% Fcxlf: Longitudinal creep force on left wheel flange contact patch (for two-point contact)
% Fcyl: Lateral creep force on left wheel contact patch (for single-point contact)
% Fcylt: Lateral creep force on left wheel tread contact patch (for two-point contact)
% Fcylf: Lateral creep force on left wheel flange contact patch (for two-point contact)
% Fczl: Vertical creep force on left wheel contact patch (for single-point contact)
% Fczlt: Vertical creep force on left wheel tread contact patch (for two-point contact)
% Fczlf: Vertical creep force on left wheel flange contact patch (for two-point contact)
% Mcxl: Longitudinal creep moment on left wheel contact patch (for single-point contact)
% Mcxlt: Longitudinal creep moment on left wheel tread contact patch (for two-point contact)
% Mcxlf: Longitudinal creep moment on left wheel flange contact patch (for two-point contact)
% Mcyl: Lateral creep moment on left wheel contact patch (for single-point contact)
% Mcylt: Lateral creep moment on left wheel tread contact patch (for two-point contact)
% Mcylf: Lateral creep moment on left wheel flange contact patch (for two-point contact)
% Mczl: Vertical creep moment on left wheel contact patch (for single-point contact)
% Mczlt: Vertical creep moment on left wheel tread contact patch (for two-point contact)
% Mczlf: Vertical creep moment on left wheel flange contact patch (for two-point contact)
% Fcxr: Longitudinal creep force on right wheel contact patch (for single-point contact)
% Fcxrt: Longitudinal creep force on right wheel tread contact patch (for two-point contact)
% Fcxrf: Longitudinal creep force on right wheel flange contact patch (for two-point contact)
% Fcyr: Lateral creep force on right wheel contact patch (for single-point contact)
% Fcyrt: Lateral creep force on right wheel tread contact patch (for two-point contact)
% Fcyrf: Lateral creep force on right wheel flange contact patch (for two-point contact)
% Fczr: Vertical creep force on right wheel contact patch (for single-point contact)
% Fczrt: Vertical creep force on right wheel tread contact patch (for two-point contact)
% Fczrf: Vertical creep force on right wheel flange contact patch (for two-point contact)
% Mcxr: Longitudinal creep moment on right wheel contact patch (for single-point contact)
% Mcxrt: Longitudinal creep moment on right wheel tread contact patch (for two-point contact)
% Mcxrf: Longitudinal creep moment on right wheel flange contact patch (for two-point contact)
% Mcyr: Lateral creep moment on right wheel contact patch (for single-point contact)
% Mcyrt: Lateral creep moment on right wheel tread contact patch (for two-point contact)
% Mcyrf: Lateral creep moment on right wheel flange contact patch (for two-point contact)
% Mczr: Vertical creep moment on right wheel contact patch (for single-point contact)
% Mczrt: Vertical creep moment on right wheel tread contact patch (for two-point contact)
% Mczrf: Vertical creep moment on right wheel flange contact patch (for two-point contact)
% Fnyl: Lateral normal force on left wheel contact patch (for single-point contact)
% Fnylt: Lateral normal force on left wheel tread contact patch (for two-point contact)
% Fnylf: Lateral normal force on left wheel flange contact patch (for two-point contact)
% Fnzl: Vertical normal force on left wheel contact patch (for single-point contact)
% Fnzlt: Vertical normal force on left wheel tread contact patch (for two-point contact)
% Fnzlf: Vertical normal force on left wheel flange contact patch (for two-point contact)
% Fnyr: Lateral normal force on right wheel contact patch (for single-point contact)
% Fnyrt: Lateral normal force on right wheel tread contact patch (for two-point contact)
% Fnyrf: Lateral normal force on right wheel flange contact patch (for two-point contact)
% Fnzr: Vertical normal force on right wheel contact patch (for single-point contact)
% Fnzrt: Vertical normal force on right wheel tread contact patch (for two-point contact)
% Fnzrf: Vertical normal force on right wheel flange contact patch (for two-point contact)

%   Obtaining rolling radii and contact angles at left and right wheel contact patches from
%   functions ’rolling_radius’ and ’contact_angle’

[rlt,rlf,rrt,rrf] = rolling_radius(x1,x1-x5,x6-x1);

[deltalt,deltalf,deltart,deltarf] = contact_angle(x1,x1-x5,x6-x1);

%   Choose set of differential equations to be solved depending on single-point or two-point
%   contact condition at end of previous time step

% Single-point contact equations - Right wheel flange contact and Left wheel tread contact

if (x6-x1) > (yfc+yfctol)

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlt;
    rr = rrf;

    % Computing tangent slopes at left and right wheel contact patches
```

```
lambdal = tan(deltalt);
lambdar = tan(deltarf);

% Computing wheelset roll angle and rate of change of wheelset roll angle

phi = (rl-rr)/(2*a);
dphi = (lambdal+lambdar)*(x3/(2*a));

% Computing effective contact angles at left and right wheel contact patches by
% compensating for wheelset roll angle

deltal = deltalt+phi;
deltar = deltarf-phi;

% Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

etaxl = -a*(x4/V)+(1-rl/r0);
etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

% Computing longitudinal, lateral, and spin creepages at right wheel flange contact patch

etaxr = a*(x4/V)+(1-rr/r0);
etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

% Obtaining creep forces and moments at left and right wheel contact patches from function
% ’onept_creep’

[Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);

% Obtaining normal forces at left and right wheel contact patches from function
% ’onept_normal’

[Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

% The single-point equations in state space form

xdot1 = x3;
xdot2 = x4;
xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-
        rl*tan(deltal))*(Fcyl+Fnyl)-(a-rr*tan(deltar))
        *(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
```

```
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Two-point contact equations - Right wheel two-point contact and Left wheel tread
% contact

elseif (x6-x1) > yfc

    % Assigning variable name to rolling radius at left wheel contact patch

    rl = rlt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalt);
    lambdart = tan(deltart);
    lambdarf = tan(deltarf);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rrt)/(2*a);
    dphi = (lambdal+lambdarf)* (x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalt+phi;
    deltatr = deltart-phi;
    deltafr = deltarf-phi;

    % Computing longitudinal, lateral, and spin creepages at right wheel tread and flange contact
    % patches

    etaxrt = a*(x4/V)+(1-rrt/r0);
    etayrt = ((x3/V)-x2*(rrt/r0)+rrt*dphi/V)/cos(deltatr);
    etasprt = ((x4/V)+(phi/r0))*cos(deltatr)+(1/r0)*sin(deltatr);
    etaxrf = a*(x4/V)+(1-rrf/r0);
    etayrf = ((x3/V)-x2*(rrf/r0)+rrf*dphi/V)/cos(deltafr);
    etasprf = ((x4/V)+(phi/r0))*cos(deltafr)+(1/r0)*sin(deltafr);

    % Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);
```

```
% Obtaining creep forces and moments at left and right wheel contact patches from function
% ’twopt_creep’

[Fcxrt,Fcyrt,Fczrt,Mcxrt,Mcyrt,Mczrt,Fcxrf,Fcyrf,Fczrf,Mcxrf,Mcyrf,Mczrf,Fcxl,Fcyl,Fczl,
Mcxl,Mcyl,Mczl] = twopt_creep(x1,etaxrt,etayrt,etasprt,
etaxrf,etayrf,etasprf,etaxl,etayl,etaspl,deltatr,deltafr,deltal);

% Correcting signs of the vertical creep force components when the vehicle has two-point
% right wheel contact.

Fczrt = -Fczrt;       Fczrf = -Fczrf;            Fczl = -Fczl;
Mcyrt = -Mcyrt;    Mcyrf = -Mcyrf;         Mcyl = -Mcyl;

%Reversing the signs of lateral creep force values to fit in the ’twopt_normal’ function under
% "going right" condition

yfc = -yfc;Fcyrt = -Fcyrt;Fcyrf = -Fcyrf;Fcyl = -Fcyl;

% Reversing the signs of x1, x2, x3, and x4 to allow for mirror effect of right and left
% traverse

x1 = -x1;x2 = -x2;x3 = -x3;x4 = -x4;

% Obtaining normal forces at left and right wheel contact patches from function
% ’twopt_normal’

[Fnyrt,Fnzrt,Fnyrf,Fnzrf,Fnyl,Fnzl] = twopt_normal(x1,x2,x3,x4,
deltatr,deltafr,deltal,rrt,rrf,rl,Fcxrt,Fcyrt,Fczrt,Fcxrf,Fcyrf,Fczrf,
Fcxl,Fcyl,Fczl,Mcyrt,Mcyrf,Mcyl);

% Correcting the signs of the lateral normal forces to adjust for the mirror effect and
% restoring the signs of the lateral creep forces

Fnyrt = -Fnyrt; Fnyrf = -Fnyrf; Fnyl = -Fnyl; Fcyrt = -Fcyrt; Fcyrf = -Fcyrf; Fcyl = -Fcyl;

% Restoring the signs of x1, x2, x3, and x4

x1 = -x1;x2 = -x2;x3 = -x3;x4 = -x4;

% The two-point equations in state space form for right wheel two-point contact

xdot1 = x3;
xdot2 = x4;
xdot3 = 1/mw*(Fcyrt+Fcyrf+Fcyl+Fnyrt+Fnyrf+Fnyl+Fsuspyw-mw*g*lambda*x1/a);
xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)+a*(Fcxrt+Fcxrf-Fcxl)+x2*((a-rrt*
       tan(deltatr))*(Fcyrt+Fnyrt)+(a-rrf*tan(deltafr))*(Fcyrf+Fnyrf)-
```

```
        (a-rl*tan(deltal))*(Fcyl+Fnyl))+Mczrt+Mczrf+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyrt-Fnyrf-Fcyrt-Fcyrf-krail*x6);

% Single-point contact equations - Right wheel tread contact and Left wheel flange contact

elseif (x1-x5) > (yfc+yfctol)

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlf;
    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdal = tan(deltalf);
    lambdar = tan(deltart);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rl-rr)/(2*a);
    dphi = (lambdal+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltal = deltalf+phi;
    deltar = deltart-phi;

    % Computing longitudinal, lateral, and spin creepages at left wheel flange contact patch

    etaxl = -a*(x4/V)+(1-rl/r0);
    etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
    etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

    % Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch
    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % ’onept_creep’

    [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
    onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);
    % Obtaining normal forces at left and right wheel contact patches from function
    % ’onept_normal’

    [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
    Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

    % The single-point equations in state space form

    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-rl*tan(deltal))
        *(Fcyl+Fnyl)-(a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
    xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

% Two point contact equations - Right wheel tread contact and Left wheel two-point
% contact

elseif (x1-x5) > yfc

    % Assigning variable name to rolling radius at right wheel contact patch

    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches

    lambdar = tan(deltart);
    lambdalt = tan(deltalt);
    lambdalf = tan(deltalf);

    % Computing wheelset roll angle and rate of change of wheelset roll angle

    phi = (rlt-rr)/(2*a);
    dphi = (lambdalf+lambdar)*(x3/(2*a));

    % Computing effective contact angles at left and right wheel contact patches by
    % compensating for wheelset roll angle

    deltatl = deltalt+phi;
    deltafl = deltalf+phi;
    deltar = deltart-phi;

    % Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch

    etaxr = a*(x4/V)+(1-rr/r0);
    etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
    etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

    % Computing longitudinal, lateral, and spin creepages at left wheel tread and flange contact
    % patches

    etaxlt = -a*(x4/V)+(1-rlt/r0);
    etaylt = ((x3/V)-x2*(rlt/r0)+rlt*dphi/V)/cos(deltatl);
    etasplt = ((x4/V)+(phi/r0))*cos(deltatl)-(1/r0)*sin(deltatl);
    etaxlf = -a*(x4/V)+(1-rlf/r0);
    etaylf = ((x3/V)-x2*(rlf/r0)+rlf*dphi/V)/cos(deltafl);
    etasplf = ((x4/V)+(phi/r0))*cos(deltafl)-(1/r0)*sin(deltafl);

    % Obtaining creep forces and moments at left and right wheel contact patches from function
    % ’twopt_creep’

    [Fcxlt,Fcylt,Fczlt,Mcxlt,Mcylt,Mczlt,Fcxlf,Fcylf,Fczlf,Mcxlf,Mcylf,Mczlf,Fcxr,Fcyr,Fczr,M
    cxr,Mcyr,Mczr] = twopt_creep(x1,etaxlt,etaylt,etasplt,
    etaxlf,etaylf,etasplf,etaxr,etayr,etaspr,deltatl,deltafl,deltar);

    % Obtaining normal forces at left and right wheel contact patches from function
    % ’twopt_normal’

    [Fnylt,Fnzlt,Fnylf,Fnzlf,Fnyr,Fnzr] = twopt_normal(x1,x2,x3,x4,deltatl,
    deltafl,deltar,rlt,rlf,rr,Fcxlt,Fcylt,Fczlt,Fcxlf,Fcylf,Fczlf,Fcxr,Fcyr, Fczr,Mcylt,Mcylf,Mcyr);

    % The two-point equations in state space form for left wheel two-point contact
    xdot1 = x3;
    xdot2 = x4;
    xdot3 = 1/mw*(Fcylt+Fcylf+Fcyr+Fnylt+Fnylf+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
    xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxlt+Fcxlf-Fcxr)-x2*((a-rlt*
        tan(deltatl))*(Fcylt+Fnylt)+(a-rlf*tan(deltafl))*(Fcylf+Fnylf)-
        (a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczlt+Mczlf+Msuspzw);
    xdot5 = (1/crail)*(-Fnylt-Fnylf-Fcylt-Fcylf-krail*x5);
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);
```

% **Single-point contact equations - Right wheel tread contact and Left wheel tread contact**

```
else

    % Assigning variable names to rolling radii at left and right wheel contact patches

    rl = rlt;
    rr = rrt;

    % Computing tangent slopes at left and right wheel contact patches
```

```
  lambdal = tan(deltalt);
  lambdar = tan(deltart);

  % Computing wheelset roll angle and rate of change of wheelset roll angle

  phi = (rl-rr)/(2*a);
  dphi = (lambdal+lambdar)*(x3/(2*a));

  % Computing effective contact angles at left and right wheel contact patches by
  % compensating for wheelset roll angle

  deltal = deltalt+phi;
  deltar = deltart-phi;

  % Computing longitudinal, lateral, and spin creepages at left wheel tread contact patch

  etaxl = -a*(x4/V)+(1-rl/r0);
  etayl = ((x3/V)-x2*(rl/r0)+rl*dphi/V)/cos(deltal);
  etaspl = ((x4/V)+(phi/r0))*cos(deltal)-(1/r0)*sin(deltal);

  % Computing longitudinal, lateral, and spin creepages at right wheel tread contact patch

  etaxr = a*(x4/V)+(1-rr/r0);
  etayr = ((x3/V)-x2*(rr/r0)+rr*dphi/V)/cos(deltar);
  etaspr = ((x4/V)+(phi/r0))*cos(deltar)+(1/r0)*sin(deltar);

  % Obtaining creep forces and moments at left and right wheel contact patches from function
  % ’onept_creep’

  [Fcxl,Fcyl,Fczl,Mcxl,Mcyl,Mczl,Fcxr,Fcyr,Fczr,Mcxr,Mcyr,Mczr] =
  onept_creep(etaxl,etayl,etaspl,etaxr,etayr,etaspr,deltal,deltar);

  % Obtaining normal forces at left and right wheel contact patches from function
  % ’onept_normal’

  [Fnyl,Fnzl,Fnyr,Fnzr] = onept_normal(x2,x4,deltal,deltar,rl,rr,Fcxl,
  Fcxr,Fcyl,Fcyr,Fczl,Fczr,Mcyl,Mcyr);

% The single point equations in state space form

  xdot1 = x3;
  xdot2 = x4;
  xdot3 = 1/mw*(Fcyl+Fcyr+Fnyl+Fnyr+Fsuspyw-mw*g*lambda*x1/a);
  xdot4 = 1/Iwz*((-Iwy*V/r0*dphi)-a*(Fcxl-Fcxr)-x2*((a-rl*tan(deltal))
      *(Fcyl+Fnyl)-(a-rr*tan(deltar))*(Fcyr+Fnyr))+Mczl+Mczr+Msuspzw);
  xdot5 = (1/crail)*(-Fnyl-Fcyl-krail*x5);
```

```
    xdot6 = (1/crail)*(-Fnyr-Fcyr-krail*x6);

end


% M-file name: truck.m
% M-file type: Function file

%  This function file identifies the differential equations for the mathematical model of a two-
%  axle truck and bolster. The front and the rear wheelsets are connected to the truck frame
%  through primary suspension elements. The equations are written in state space form.

%  This function is called by the function file 'equations.m' for solving the differential equations
%  constituting the motion of the truck frame. Inputs to this function are the lateral and yaw
%  displacements and velocities of the wheelsets, the roll angles of the wheelsets, and the lateral
%  and yaw displacements and velocities of the truck frame, and the truck frame suspension
%  forces and moments.

function [ydot13,ydot14,ydot15,ydot16,ydot17,ydot18] =
truck(y13,y14,y15,y16,y17,y18,phi1,phi2,Fsuspyf,Msuspzf,Msuspzb)

% Parameters used for simulation

% mf: Mass of truck frame (kg)
% Ifz: Yaw principal mass moment of inertia of truck frame (kg-m^2)
% mb: Mass of bolster (kg)
% Ibz: Yaw principal mass moment of inertia of bolster (kg-m^2)
% g: Acceleration due to gravity (m/s^2)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'full_vehicle.m'.

global mf mb Ifz Ibz g;

% The truck frame equations in state space form

ydot13 = y15;
ydot14 = y16;
ydot15 = 1/(mf+mb)*(-0.5*(mf+mb)*g*(phi1+phi2)+Fsuspyf);
ydot16 = 1/Ifz*Msuspzf;
ydot17 = y18;
ydot18 = 1/Ibz*Msuspzb;


% M-file name: carbody.m
% M-file type: Function file

%  This function file identifies the differential equations for the mathematical model of the
%  carbody. The front and the rear trucks are connected to the carbody through secondary
%  suspension elements. The equations are written in state space form.

%  This function is called by the function file ‘equations.m’ for solving the differential equations
%  constituting the motion of the carbody. Inputs to this function are the carbody roll
%  displacement, the carbody lateral, yaw, and roll velocities, and the carbody suspension forces
%  and moments.

function [xdot37,xdot38,xdot39,xdot40,xdot41,xdot42] =
carbody(x39,x40,x41,x42,Fsuspy1,Fsuspy2,Msuspz1,Msuspz2,Msuspx1,Msuspx2);

% Parameters used for simulation

% mc: Mass of carbody (kg)
% Icz: Yaw principal mass moment of inertia of carbody (kg-m^2)
% Icx: Roll principal mass moment of inertia of carbody (kg-m^2)
% g: Acceleration due to gravity (m/s^2)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'full_vehicle.m'.

global mc Icz Icx g;

% The carbody equations in state space form

xdot37 = x40;
xdot38 = x41;
xdot39 = x42;
xdot40 = 1/mc*(-mc*g*x39+Fsuspy1+Fsuspy2);
xdot41 = 1/Icz*(Msuspz1+Msuspz2);
xdot42 = 1/Icx*(Msuspx1+Msuspx2);


% M-file name: onept_creep.m
% M-file type: Function file

%  This function file computes the creep forces and moments acting on the left and the right
%  wheels due to the interaction between wheel and rail. This calculation is valid for a single-
%  point contact condition between the wheel and the rail. The inputs to the function are the
%  longitudinal, lateral, and spin creepages and the contact angles. The outputs are the creep
%  forces and moments resolved in longitudinal, lateral, and vertical directions.
%   Kalker’s creep theory is used and spin creep saturation has been taken into account.

%   The outputs given by this function are called by the function file ’wheelset.m’.

function [Fcx1,Fcy1,Fcz1,Mcx1,Mcy1,Mcz1,Fcx2,Fcy2,Fcz2,Mcx2,Mcy2,Mcz2] =
onept_creep(etax1,etay1,etasp1,etax2,etay2,etasp2,delta1,delta2)

% Parameters used for simulation

% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global muN f11 f12 f22 f33;

% Computing the creep forces and moments at the left wheel-rail contact patch

fcpx1 = -f33*etax1;
fcpy1 = -f11*etay1-f12*etasp1;
mcpz1 = f12*etay1-f22*etasp1;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx1^2+fcpy1^2);

if beta<=3
   epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
   epsilon = (1/beta);
end

Fcpx1 = epsilon*fcpx1;
Fcpy1 = epsilon*fcpy1;
Mcpz1 = epsilon*mcpz1;

% Computing the creep forces and moments at the right wheel-rail contact patch

fcpx2 = -f33*etax2;
fcpy2 = -f11*etay2-f12*etasp2;
```

```
mcpz2 = f12*etay2-f22*etasp2;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx2^2+fcpy2^2);

if beta<=3
   epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
   epsilon = (1/beta);
end

Fcpx2 = epsilon*fcpx2;
Fcpy2 = epsilon*fcpy2;
Mcpz2 = epsilon*mcpz2;

% The creep forces and moments calculated above are in the contact patch plane. These forces
% and moments are resolved in the longitudinal, lateral, and vertical directions in the track
% coordinate system. The variables calculated below are passed to the function ’wheelset’.

% Resolving creep forces and moments for the left wheel

Fcx1 = Fcpx1;
Fcy1 = Fcpy1*cos(delta1);
Fcz1 = Fcpy1*sin(delta1);
Mcx1 = 0;
Mcy1 = -Mcpz1*sin(delta1);
Mcz1 = Mcpz1*cos(delta1);

% Resolving creep forces and moments for the right wheel

Fcx2 = Fcpx2;
Fcy2 = Fcpy2*cos(delta2);
Fcz2 = -Fcpy2*sin(delta2);
Mcx2 = 0;
Mcy2 = Mcpz2*sin(delta2);
Mcz2 = Mcpz2*cos(delta2);


% M-file name: twopt_creep.m
% M-file type: Function file

%   This function file computes the creep forces and moments acting on the left and the right
%   wheels due to the interaction between wheel and rail. This calculation is valid for a two-point
%   contact condition between the wheel and the rail. The inputs to the function are the
%  longitudinal, lateral, and spin creepages and the contact angles. The outputs are the creep
%  forces and moments resolved in longitudinal, lateral, and vertical directions.

%  Kalker’s creep theory is used and spin creep saturation has been taken into account.

%  The outputs given by this function are called by the function file ’wheelset.m’.

function [Fcx1,Fcy1,Fcz1,Mcx1,Mcy1,Mcz1,Fcx2,Fcy2,Fcz2,Mcx2,Mcy2,Mcz2,Fcx3,
Fcy3,Fcz3,Mcx3,Mcy3,Mcz3] = twopt_creep(x1,etax1,etay1,etasp1,etax2,etay2,
etasp2,etax3,etay3,etasp3,delta1,delta2,delta3)

% Parameters used for simulation

% muN: Product of coefficient of friction between wheel and rail (mu) and the normal load on
% the axle (N)
% f11: Lateral creep coefficient (N)
% f12: Lateral/Spin creep coefficient (N-m)
% f22: Spin creep coefficient (N-m^2)
% f33: Longitudinal creep coefficient (N)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global muN f11 f12 f22 f33;

% Computing the creep forces and moments at contact patch 1. For two-point contact at the left
% wheel, contact patch 1 is the left wheel-rail tread contact patch. For two-point contact at the
% right wheel, contact patch 1 is the right wheel-rail tread contact patch.

fcpx1 = -f33*etax1;
fcpy1 = -f11*etay1-f12*etasp1;
mcpz1 = f12*etay1-f22*etasp1;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx1^2+fcpy1^2);

if beta<3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx1 = epsilon*fcpx1;
Fcpy1 = epsilon*fcpy1;
```

```
Mcpz1 = epsilon*mcpz1;

% Computing the creep forces and moments at contact patch 2. For two-point contact at the left
% wheel, contact patch 2 is the left wheel-rail flange contact patch. For two-point contact at the
% right wheel, contact patch 2 is the right wheel-rail flange contact patch.

fcpx2 = -f33*etax2;
fcpy2 = -f11*etay2-f12*etasp2;
mcpz2 = f12*etay2-f22*etasp2;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx2^2+fcpy2^2);

if beta<3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx2 = epsilon*fcpx2;
Fcpy2 = epsilon*fcpy2;
Mcpz2 = epsilon*mcpz2;

% Computing the creep forces and moments at contact patch 3. For two-point contact at the left
% wheel, contact patch 3 is the right wheel-rail contact patch. For two-point contact at the right
% wheel, contact patch 3 is the left wheel-rail flange contact patch.

fcpx3 = -f33*etax3;
fcpy3 = -f11*etay3-f12*etasp3;
mcpz3 = f12*etay3-f22*etasp3;

% Creep force saturation

beta = (1/(muN))*sqrt(fcpx3^2+fcpy3^2);

if beta<3
  epsilon = (1/beta)*(beta-beta^2/3+beta^3/27);
else
  epsilon = (1/beta);
end

Fcpx3 = epsilon*fcpx3;
Fcpy3 = epsilon*fcpy3;
Mcpz3 = epsilon*mcpz3;
```

```
% The creep forces and moments calculated above are in the contact patch plane. These forces
% and moments are resolved in the longitudinal, lateral, and vertical directions in the track
% coordinate system. The variables calculated below are passed to the function ’wheelset’.

% Resolving creep forces and moments at contact patch 1

Fcx1 = Fcpx1;
Fcy1 = Fcpy1*cos(delta1);
Fcz1 = Fcpy1*sin(delta1);
Mcx1 = 0;
Mcy1 = -Mcpz1*sin(delta1);
Mcz1 = Mcpz1*cos(delta1);

% Resolving creep forces and moments at contact patch 2

Fcx2 = Fcpx2;
Fcy2 = Fcpy2*cos(delta2);
Fcz2 = Fcpy2*sin(delta2);
Mcx2 = 0;
Mcy2 = -Mcpz2*sin(delta2);
Mcz2 = Mcpz2*cos(delta2);

% Resolving creep forces and moments at contact patch 3

Fcx3 = Fcpx3;
Fcy3 = Fcpy3*cos(delta3);
Fcz3 = -Fcpy3*sin(delta3);
Mcx3 = 0;
Mcy3 = Mcpz3*sin(delta3);
Mcz3 = Mcpz3*cos(delta3);


% M-file name: onept_normal.m
% M-file type: Function file

%  This function file computes the normal forces acting on the left and the right wheels due to
%  the interaction between wheel and rail. This calculation is valid for a single-point contact
%  condition between the wheel and the rail. The inputs to the function are the longitudinal,
%  lateral, and vertical creep forces, the lateral creep moments, rolling radii and contact angles at
%  the left and the right wheel contact patches, and the wheelset yaw displacement and velocity.
%  The outputs are the normal forces on the left and the right wheels resolved in longitudinal,
%  lateral, and vertical directions.

%  The outputs given by this function are called by the function file ’wheelset.m’.
```

```
function [Fny1,Fnz1,Fny2,Fnz2] = onept_normal(x2,x4,delta1,delta2,r1,r2,
Fcx1,Fcx2,Fcy1,Fcy2,Fcz1,Fcz2,Mcy1,Mcy2)

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% mw: Mass of wheelset (kg)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% g: Acceleration due to gravity (m/s^2)
% N: Axle load (N)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% 'full_vehicle.m'.

global V a r0 mw Iwy g N;

% Computing the normal forces at the left and right wheel-rail contact patches

F = -Fcz1-Fcz2+mw*g+N;
M = a*(Fcz2-Fcz1)-r1*(Fcy1-x2*Fcx1)-r2*(Fcy2-x2*Fcx2)-x2*(Mcy1+Mcy2)-Iwy*(V/r0)*x4;
vl = F*(a*cos(delta2)-r2*sin(delta2))+M*cos(delta2);
vr = F*(a*cos(delta1)-r1*sin(delta1))-M*cos(delta1);
del1 = 2*a*cos(delta1)*cos(delta2)-r2*cos(delta1)*sin(delta2)-r1*sin(delta1)*cos(delta2);

Fnl = vl/del1;
Fnr = vr/del1;

% The normal forces calculated above are normal to the contact patch plane. These forces are
% resolved in the longitudinal, lateral, and vertical directions in the track coordinate system. The
% variables calculated below are passed to the function 'wheelset'.

% Resolving normal forces at the left and right wheel contact patches

Fny1 = -Fnl*sin(delta1);
Fnz1 = Fnl*cos(delta1);
Fny2 = Fnr*sin(delta2);
Fnz2 = Fnr*cos(delta2);
```

% M-file name: **twopt_normal.m**
% M-file type: Function file

```
%  This function file computes the normal forces acting on the left and the right wheels due to
%  the interaction between wheel and rail. This calculation is valid for a two-point contact
%  condition between the wheel and the rail. The inputs to the function are the longitudinal,
%  lateral, and vertical creep forces, the lateral creep moments, rolling radii and contact angles at
%  the left and the right wheel contact patches, and the wheelset lateral and yaw displacement
%  and velocity. The outputs are the normal forces on the left and the right wheels resolved in
%  longitudinal, lateral, and vertical directions.

%  The outputs given by this function are called by the function file ’wheelset.m’.

function [Fny1,Fnz1,Fny2,Fnz2,Fny3,Fnz3] = twopt_normal(x1,x2,x3,x4,delta1,
delta2,delta3,r1,r2,r3,Fcx1,Fcy1,Fcz1,Fcx2,Fcy2,Fcz2,Fcx3,Fcy3,Fcz3,Mcy1,Mcy2,Mcy3)

% Parameters used for simulation

% V: Forward velocity of wheelset (m/sec)
% a: Half of track gage (m)
% r0: Centered rolling radius of the wheel (m)
% mw: Mass of wheelset (kg)
% Iwy: Pitch principal mass moment of inertia of wheelset (kg-m^2)
% g: Acceleration due to gravity (m/s^2)
% yfc: Flange clearance or flange width (m)
% krail: Effective lateral stiffness of rail (N/m)
% crail: Effective lateral damping of rail (N/m)
% N: Axle load (N)

% Indicating the global nature of the variables. This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global V a r0 mw Iwy g yfc krail crail N;

% Computing the normal forces at the left and right wheel-rail contact patches

F2 = -Fcz1-Fcz2-Fcz3+N+mw*g;
M1 = -a*(Fcz1+Fcz2-Fcz3)-r1*(Fcy1-x2*Fcx1)-r2*(Fcy2-x2*Fcx2)-r3*(Fcy3-x2*Fcx3)-
x2*(Mcy1+Mcy2+Mcy3)-Iwy*(V/r0)*x4;

Vt = F1*(2*a*cos(delta2)*cos(delta3)-r2*sin(delta2)*cos(delta3)-r3*
cos(delta2)*sin(delta3))+F2*(sin(delta2)*(a*cos(delta3)-r3*sin(delta3)))
+M1*sin(delta2)*cos(delta3);

vf = F1*(-2*a*cos(delta1)*cos(delta3)+r1*sin(delta1)*cos(delta3)+r3*
cos(delta1)*sin(delta3))-F2*(sin(delta1)*(a*cos(delta3)-r3*sin(delta3)))-
M1*sin(delta1)*cos(delta3);
```

```
v = F1*(r2*cos(delta1)*sin(delta2)-r1*sin(delta1)*cos(delta2))+F2*
(a*(cos(delta1)*sin(delta2)-sin(delta1)*cos(delta2))+(r2-r1)* sin(delta1)*sin(delta2))-
M1*(cos(delta1)*sin(delta2)-sin(delta1)*
cos(delta2));

del2 = (2*a*cos(delta3)-r3*sin(delta3))*(cos(delta1)*sin(delta2)-sin(delta1)*cos(delta2))+(r2-
r1)*sin(delta1)*sin(delta2)*cos(delta3);

Fnt = vt/del2;
Fnf = vf/del2;
Fn = v/del2;

% The normal forces calculated above are normal to the contact patch plane. These forces are
% resolved in the longitudinal, lateral, and vertical directions in the track coordinate system. The
% variables calculated below are passed to the function ’wheelset’.

% Resolving normal forces at the left and right wheel contact patches

Fny1 = -Fnt*sin(delta1);
Fnz1 = Fnt*cos(delta1);
Fny2 = -Fnf*sin(delta2);
Fnz2 = Fnf*cos(delta2);
Fny3 = Fn*sin(delta3);
Fnz3 = Fn*cos(delta3);


% M-file name: wheelset_suspension.m
% M-file type: Function file

%   This function file computes the suspension forces acting on the front and the rear wheelsets
%   due to the primary suspension components. The primary suspension consists of longitudinal
%   and lateral springs and dampers. The inputs to the function are the wheelset lateral and yaw
%   displacements and velocities. The outputs are the lateral suspension force and the yaw
%   suspension moment on the wheelset.

%   The outputs given by this function are called by the function file ’equations.m’.

function [Fsusp1,Fsusp2,Msusp1] = wheelset_suspension(y1,y3,y2,y4,y13,y14,y15,y16)

% Parameters used for simulation

% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-sec/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-sec/m)
% dp: Half of lateral distance between primary longitudinal springs (m)
% b: Half of wheelbase (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global kpx cpx kpy cpy dp b;

% Computing the suspension forces and moments acting on the wheelset. These variables are
% passed to the function ’equations’.

Fsusp1 = -2*kpy*y1+2*kpy*y13+2*b*kpy*y14-2*cpy*y3+2*cpy*y15+2*b*cpy*y16;
Fsusp2 = -2*kpy*y1+2*kpy*y13-2*b*kpy*y14-2*cpy*y3+2*cpy*y15-2*b*cpy*y16;
Msusp1 = -2*dp^2*kpx*y2+2*dp^2*kpx*y14-2*dp^2*cpx*y4+2*dp^2*cpx*y16;


% M-file name: truck_suspension.m
% M-file type: Function file

%   This function file computes the suspension forces acting on the truck frame and bolster due
%   to the primary and secondary suspension components. The primary suspension consists of
%   longitudinal and lateral springs and dampers. The secondary suspension consists of a lateral
%   spring and damper, a yaw spring and damper, and a Coulomb friction arrangement. The
%   inputs to the function are the wheelset lateral and yaw displacements and velocities, the truck
%   frame lateral and yaw displacements and velocities, and the bolster yaw displacement and
%   velocity. The outputs are the lateral suspension force and the vertical suspension moment on
%   the truck frame, and the vertical suspension moment on the bolster.

%   The outputs given by this function are called by the function file ’equations.m’.

function [Fsusp1,Fsusp2,Msusp1,Msusp2] =
truck_suspension(x1,x2,x3,x4,x7,x8,x9,x10,x13,x14,x15,x16,x37,x38,x40,x41, x17,x18)

% Parameters used for simulation

% kpx: Primary longitudinal stiffness coefficient (N/m)
% cpx: Primary longitudinal damping coefficient (N-sec/m)
% kpy: Primary lateral stiffness coefficient (N/m)
% cpy: Primary lateral damping coefficient (N-sec/m)
% dp: Half of lateral distance between primary longitudinal springs (m)
% b: Half of wheelbase (m)
% ksy: Secondary lateral stiffness coefficient (N/m)
% csy: Secondary lateral damping coefficient (N-sec/m)
% ksphi: Secondary yaw stiffness coefficient (N/rad)
% csphi: Secondary yaw damping coefficient (N-sec/rad)
% T0: Centerplate breakaway torque (N-m)
% C0: Coulomb viscous yaw damping coefficient (N-sec/rad)
% ls: Longitudinal distance between C.G. of carbody to C.G. of either truck (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global kpx cpx kpy cpy dp b ksy csy ksphi csphi T0 C0 ls;

% Computing the suspension forces and moments acting on the truck frame and bolster. These
% variables are passed to the function ’equations’.

% Truck frame / Bolster Coulomb friction characteristic

if (x16-x18) >= T0/C0
  Tcoul = T0;
elseif (x16-x18) <= -T0/C0
   Tcoul = -T0;
else
   Tcoul = C0*(x16-x18);
end

Fsusp1 = 2*kpy*x1+2*kpy*x7+(-4*kpy-2*ksy)*x13+2*ksy*x37+
2*ksy*ls*x38+2*cpy*x3+2*cpy*x9+(-4*cpy-2*csy)*x15+2*csy*x40+2*csy*ls*x41;
Fsusp2 = 2*kpy*x1+2*kpy*x7+(-4*kpy-2*ksy)*x13+2*ksy*x37-2*ksy*ls*x38+
2*cpy*x3+2*cpy*x9+(-4*cpy-2*csy)*x15+2*csy*x40-2*csy*ls*x41;
Msusp1 = 2*b*kpy*x1+2*dp^2*kpx*x2+(-2*b*kpy)*x7+2*dp^2*kpx*x8+(-4*dp^2*kpx-
4*b^2*kpy)*x14+2*b*cpy*x3+2*dp^2*cpx*x4+(-2*b*cpy)*x9+2*dp^2*cpx*x10+(-
4*dp^2*cpx-4*b^2*cpy)*x16-Tcoul;
Msusp2 = -ksphi*x17+ksphi*x38-csphi*x18+csphi*x41+Tcoul;


% M-file name: carbody_suspension.m
% M-file type: Function file

%  This function file computes the suspension forces acting on the carbody due to the secondary
%  suspension components. The secondary suspension consists of lateral, yaw, and vertical
%  springs and dampers. The inputs to the function are the lateral displacement and velocity of
%  the front and the rear truck, the yaw displacement and velocity of the front and the rear
%  bolster, and the lateral, yaw, and roll displacements and velocities of the carbody. The
%  outputs are the lateral suspension force, and the vertical and longitudinal suspension
%  moments on the carbody.

%  The outputs given by this function are called by the function file ’equations.m’.
```

```
function [Fsuspy1,Fsuspy2,Msuspz1,Msuspz2,Msuspx1,Msuspx2] =
carbody_suspension(y13,y15,y17,y18,y31,y33,y35,y36,y37,y38,y39,y40,y41,y42)

% Parameters used for simulation

% ksy: Secondary lateral stiffness coefficient (N/m)
% csy: Secondary lateral damping coefficient (N-sec/m)
% ksphi: Secondary yaw stiffness coefficient (N/rad)
% csphi: Secondary yaw damping coefficient (N-sec/rad)
% ksz: Secondary vertical stiffness coefficient (N/m)
% csz: Secondary vertical damping coefficient (N-sec/m)
% ds: Half of Lateral distance between secondary vertical springs (m)
% ls: Longitudinal distance between C.G. of carbody to C.G. of either truck (m)
% hcs: Vertical distance between C.G. of carbody to secondary lateral springs (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global ksy csy ksphi csphi ksz csz ds ls hcs;

% Computing the suspension forces and moments acting on the carbody. These variables are
% passed to the function ’equations’.

Fsuspy1 = 2*ksy*y13-2*ksy*y37+2*csy*y15-2*csy*y40-2*hcs*ksy*y39-2*hcs*csy*y42;
Fsuspy2 = 2*ksy*y31-2*ksy*y37+2*csy*y33-2*csy*y40-2*hcs*ksy*y39-2*hcs*csy*y42;
Msuspz1 = (2*ls*ksy)*y13+(ksphi)*y17+(-2*ls^2*ksy-ksphi)*y38+
(2*ls*csy)*y15+(csphi)*y18+(-2*ls^2*csy-csphi)*y41;
Msuspz2 = (-2*ls*ksy)*y31+(ksphi)*y35+(-2*ls^2*ksy-ksphi)*y38+
(-2*ls*csy)*y33+(csphi)*y36+(-2*ls^2*csy-csphi)*y41;
Msuspx1 = Fsuspy1*hcs-2*ksz*ds^2*y39-2*csz*ds^2*y42;
Msuspx2 = Fsuspy2*hcs-2*ksz*ds^2*y39-2*csz*ds^2*y42;



% M-file name: rolling_radius.m
% M-file type: Function file

%   This function file computes the rolling radii at the left and the right wheel-rail contact
%   patches as a function of the relative lateral displacement between the left and the right wheels
%   and rails. The inputs to the function are the relative lateral displacement between the left
%   wheel and the left rail and the right wheel and the right rail. The outputs are the rolling radii
%   at the left and the right tread and flange contact patches.

%   The outputs given by this function are called by the function file ’wheelset.m’.

function [rlefttread,rleftflange,rrighttread,rrightflange] = rolling_radius(y,yl,yr)
```

```
% Parameters used for simulation

% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)
% r0: Centered rolling radius of the wheel (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global yfc yfctol r0;

% Computing the rolling radii at the left and the right wheel-rail contact patches. These variables
% are passed to the function ‘wheelset’.

if y >= 0

    % Wheelset displacement is in the positive (or left) direction from the centerline (Right
    %   wheel tread contact)

    % Note that for a left movement of the wheelset from the centerline, the
    % right wheel does not flange. Hence, the zero value for the right wheel
    % flange radius is not real and is not used in calculations.

    rrighttread = r0 + 0.125*(yr);
    rrightflange = 0.000;

    if yl <= yfc

    % Left wheel tread contact. Note that the zero value for the left wheel flange radius is not
    % real and is not used in calculations.

    rlefttread = r0 + 0.125*yl;
    rleftflange = 0.000;

    elseif yl < (yfc+yfctol)

    % Left wheel flange contact

    rlefttread = r0+0.125*yl;
    rleftflange = 0.3566+13700*(yl-0.008)^2;

    else

    % Left wheel two-point contact
    rlefttread = r0+0.125*yl;
    rleftflange = 0.3703+2.5833*(yl-0.009);

    end

else

    % Wheelset displacement is in the negative (or right) direction from the centerline (Left
    % wheel tread contact)

    % Note that for a right movement of the wheelset from the centerline, the
    % left wheel does not flange. Hence, the zero value for the left wheel
    % flange radius is not real and is not used in calculations.

    rlefttread = r0 + 0.125*yl;
    rleftflange = 0.000;

    if yr <= yfc

    % Right wheel tread contact. Note that the zero value for the right
    % wheel flange radius is not real and is not used in calculations.

    rrighttread = r0 + 0.125*(yr);
    rrightflange = 0.000;

    elseif yr < (yfc+yfctol)

    % Right wheel flange contact

    rrighttread = r0 + 0.125*yr;
    rrightflange = 0.3566+13700*(yr-0.008)^2;

    else

  % Right wheel two-point contact

    rrighttread = r0+0.125*yr;
    rrightflange = 0.3703+2.5833*(yr-0.009);

    end

end


% M-file name: contact_angle.m
% M-file type: Function file
%   This function file computes the contact angles at the left and the right wheel-rail contact
%   patches as a function of the relative lateral displacement between the left and the right wheels
%   and rails. The inputs to the function are the relative lateral displacement between the left
%   wheel and the left rail and the right wheel and the right rail. The outputs are the contact
%   angles at the left and the right tread and flange contact patches.

%   The outputs given by this function are called by the function file ’wheelset.m’.

function [calefttread,caleftflange,carighttread,carightflange] = contact_angle(y,yl,yr)

% Parameters used for simulation

% yfc: Flange clearance or flange width (m)
% yfctol: Lateral tolerance added to yfc in order to facilitate numerical simulation (m)

% Indicating the global nature of the variables This means that the value of the variables need
% not be specified in this function file. This value is automatically obtained from the main file
% ’full_vehicle.m’.

global yfc yfctol;

% Computing the rolling radii at the left and the right wheel-rail contact patches. These variables
% are passed to the function ‘wheelset’.

if y >= 0

    % Wheelset displacement is in the positive (or left) direction from the centerline (Right
    % wheel tread contact)

    % Note that for a left movement of the wheelset from the centerline, the
    % right wheel does not flange. Hence, the zero value for the right wheel
    % flange angle is not real and is not used in calculations.

    carighttread = atan(0.125);
    carightflange = 0.000;

    if yl <= yfc


    % Left wheel tread contact. Note that the zero value for the left wheel flange angle is not real
    % and is not used in calculations.


    calefttread = atan(0.125);
    caleftflange = 0.000;

    elseif yl < (yfc+yfctol)
    % Left wheel flange contact

    calefttread = atan(0.125);
    caleftflange = atan(0.125+2648*(yl-0.008));

    else

    % Left wheel two-point contact

    calefttread = atan(0.125);
    caleftflange = atan(2.748);

    end

else

    % Wheelset displacement is in the negative (or right) direction from the centerline (Left
    % wheel tread contact)

    % Note that for a right movement of the wheelset from the centerline, the
    % left wheel does not flange. Hence, the zero value for the left wheel
    % flange angle is not real and is not used in calculations.

    calefttread = atan(0.125);
    caleftflange = 0.000;

    if yr <= yfc

    % Right wheel tread contact. Note that the zero value for the right
    % wheel flange angle is not real and is not used in calculations.

    carighttread = atan(0.125);
    carightflange = 0.000;

    elseif yr < (yfc+yfctol)

    % Right wheel flange contact

    carighttread = atan(0.125);
    carightflange = atan(0.125+2648*(yr-0.008));

    else

    % Right wheel two-point contact

    carighttread = atan(0.125);
    carightflange = atan(2.748);

    end

end
```

Vita

Anant Mohan was born on July 18, 1975 in Pune, India. He grew up in Hyderabad, India where he did all his schooling. He graduated with a B.S. in Mechanical Engineering from Osmania University, India in May 1996.

Soon after graduation, Anant moved to the United States to pursue his further education. He finished his course work at Virginia Tech with an emphasis on Nonlinear Dynamics. He started working full-time in the industry in August 1998. Since then, he has worked in the Aerospace, Automotive, and Nuclear industry.

Anant currently works as a nuclear fuel development engineer for Framatome ANP in Lynchburg, Virginia.