

# Symbolic Formulation of Multibody Dynamic Equations for Wheeled Vehicle Systems on Three-Dimensional Roads

2010-01-0719

Published  
04/12/2010

William Bombardier, John McPhee and Chad Schmitke  
University of Waterloo, Canada

Copyright © 2010 SAE International

## ABSTRACT

A method to improve the computational efficiency of analyzing wheeled vehicle systems on three-dimensional (3-D) roads has been developed. This was accomplished by creating a technique to incorporate the tire on a 3-D road in a multibody dynamics model of the vehicle with an approach that formulates the governing equations using symbolic formulation. For general handling analysis performed on the vehicle, the tire forces and moments are determined using a tire model that represents the tire as a set of mathematical expressions. Since these expressions need numerical values to determine the forces and moments, a symbolic solution does not exist. Therefore, the evaluation of the tire forces and moments needs to be done during simulation. However, symbolic operations can be used when the governing equations are formulated to develop an efficient method to evaluate these forces.

A method to automatically construct the procedures necessary to evaluate the tire forces and moments has been developed. This approach includes a technique to automatically generate an optimized road model procedure that calculates the contact point between the tire and the ground. The method is based on the thin disk tire model with variable radius, resulting in two non-linear equations that define the point of contact. The road model procedure is developed and optimized during automated formulation of the equations of motion.

The road model procedure was implemented in the DynaFlexPro software package by creating a simulation code structure to evaluate the tire forces and moments during simulation. The structure was included in a tire/road component that is a linear graph representation of the tire and

a three-dimensional (3-D) road. A vehicle system was analyzed on two different road profiles performing a cornering maneuver using DynaFlexPro and repeated in MSC.ADAMS to test the accuracy of the approach. Good agreement was achieved between the symbolic computing method presented here, and the purely numerical algorithm along with significant improvements in the simulation time.

## INTRODUCTION

There are three steps to analyze multibody dynamic systems: model, formulate, and simulate [1]. A multibody system model is created by describing the topology of the system using idealized components, such as joints and rigid bodies. The governing equations are generated during the formulation stage and solved in the simulation phase using numerical techniques.

A symbolical or numerical formulation approach can be used during the formulation stage to generate the governing equations (the equations of motion and the non-linear algebraic constraint equations) that describe the system. A numerical approach will construct numerical matrices that represent the system at a given instant in time. These matrices are numerical in nature and thus need to be re-formulated at each time step. On the other hand, a symbolical formulation method will generate the governing equations in the form of symbolic mathematical expressions that describe the system for all time. Therefore, the equations only have to be formulated once and not at every time step; thus, permitting a reduction in the simulation time. Other advantages of a symbolic formulation approach include: the equations can be optimized/simplified before they are solved, the equations are portable, and they can be viewed to obtain some physical insight [2].

There have been a few symbolic formulation techniques developed in the past few years to generate the governing equations representing vehicle systems. Sayers and Han [1] created AutoSim to generate the equations of a generic four-wheeled vehicle. They used Kane's method with an emphasis on open-loop systems using joint coordinates to formulate the equations. The equations have been incorporated into the CarSim commercial software package [3]. Fisette et al. [4] developed ROBOTRAN to generate the governing equations based on d'Alembert's potential power principle. Postiau et al. [5] used ROBOTRAN to analyze vehicle systems with multi-link suspensions.

Neither AutoSim nor ROBOTRAN are implemented in a general purpose computer algebra software package and thus are only capable of performing a limited number of optimization techniques on the equations before they are simulated. Also, all symbolic operations have to be hard-coded and therefore are limited in the operations that can be used to develop the equations and post-process them. Additionally, they are limited to the mechanical domain only and can not analyze multi-domain systems. CarSim and veDYNA from Tesis DYNAware [6] use a pre-defined topology to represent the vehicle system, and thus can only analyze certain vehicle systems.

These shortcomings can be overcome using a symbolic formulation method to analyze universal multibody dynamic systems that are implemented in a general purpose computer algebra software package. This permits the user to define the vehicle he/she wishes to analyze while taking advantage of the symbolic simplification techniques of the general purpose computer algebra software package. The difficulty in analyzing wheeled vehicle systems is the incorporation of the pneumatic tire in the multibody dynamics model of the vehicle. For general handling analysis performed on the vehicle, the tire is modeled through a tire model that represents the tire as a set of mathematical expressions that were determined from physical test performed on the tire [7]. The tire model will determine the magnitude of the tire forces and moments based on a set of tire intermediate variables as input. The majority of tire models contain "if" statements that can only be evaluated during simulation once numerical values of the tire intermediate variables are known. Thus, in most cases, a symbolic solution to the tire forces and moments does not exist. However, a symbolic formulation approach can still be utilized to formulate the governing equations with a method that allows the tire forces and moments that appear in the generalized force matrix to be evaluated during simulation [8].

Morency et al. [8] developed a simulation code structure that permits the tire forces and moments to be evaluated during simulation. The simulation code structure follows the steps defined by Sayers and Han [1] to evaluate the forces and moments through a series of expression evaluations and

function calls. Sayers and Han [1] outlined a number of steps required to incorporate the tire on a flat road in a multibody dynamics model of the vehicle. Morency [8,9] incorporated his method in the DynaFlexPro (DFP) software package, which is implemented in the Maple computer algebra system by Maplesoft and uses linear graph theory to formulate the governing equations. Morency et al. [8,9] used DFP to test different vehicle topologies in diverse scenarios.

Bombardier and McPhee [10] extended the approach created by Morency [9] to incorporate the tire rolling on a three-dimensional (3-D) road surface by updating the steps outlined by Sayers and Han [1] to include an additional step, which is the determination of the contact point between the tire and the ground. The contact point can not be assumed to be known when the road is free to vary. This added step requires an extra function call in the simulation code structure to a road model procedure that determines the contact point between the tire and the ground. Bombardier and McPhee [10] incorporated their approach into the DFP software package by hard-coding the equations that are necessary to determine the contact point into the road model procedure. The road model included the Newton-Raphson iteration scheme to solve the equations at each time step for every tire in the system. They showed the many advantages of an in-line Newton-Raphson method, analogous to the in-line integration approach used by Dymola [11]. However, the method has the limitation that the road model needs to be manually hard-coded for every road profile.

In the past decade there has been an increase in demand to improve the time to analyze vehicle systems. This includes the time required to formulate the governing equations and to simulate them. Thus, it is inefficient to develop the equations and to hard-code them in the road model procedure every time the road profile changes. Therefore, there is a need to develop an approach to automatically construct the road model procedure during the formulation stage using symbolic operations so that the result (procedure) can be utilized during simulation to determine the contact point directly without any equation generation. This paper presents a method to automatically generate the road model procedure during the formulation stage based on the road profile as input. The methods that describe the road model are presented first followed by a description of how the model is automatically formulated and an explanation of the simulation code structure. The approach is validated by comparing results obtained from DFP to results obtained from MSC.AMAMS for a vehicle performing a cornering maneuver on two different road profiles.

# PNEUMATIC TIRE ON A 3-d ROAD IN A MULTIBODY DYNAMICS CONTEXT

Sayers and Han [1] outlined a number of steps to incorporate the tire in a multibody dynamics model of the vehicle. Bombardier and McPhee [10] extended the steps to include a tire on a 3-D road, and they are as follows.

1. Define a point where the tire forces will act on the multibody model of the vehicle
2. Determine a method to locate the contact point between the tire and the ground
3. Select an expression for the normal force
4. Establish directions for the ISO unit vectors,  $\hat{u}_{ISOX}$ ,  $\hat{u}_{ISOY}$ , and  $\hat{u}_{ISOZ}$  [12]
5. Determine equations for the most common kinematic inputs, such as the camber angle and the slip parameters, required by tire models
6. Use a tire model, such as the Fiala [7] or Pacejka [13] tire model, to determine expressions for the magnitudes of the remaining tire forces and moments

The second step is the additional one required to analyze vehicle systems on 3-D roads. Bombardier and McPhee [10] used the thin disk tire model with variable radius approach to develop an in-line algorithm to determine the contact point between each tire and the road. The thin disk tire model with variable radius method was developed by Postiau et al. [5] based on a model proposed by Fisette and Samin [14]. The approach is shown in Fig. 1, and the assumptions to the method include the following.

- The road is modeled as a 3-D mathematical function ( $z = f(x, y)$ )
- There is only one point of contact between the tire and the ground
- The ground is rigid and smooth
- The tire is assumed to be a planar disk with variable radius

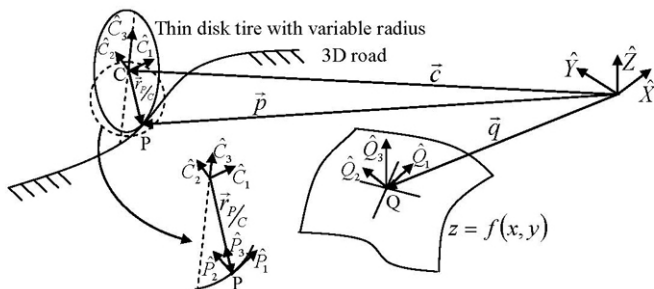


**Figure 1. Nomenclature of the Thin Disk Tire Model with Variable Radius**

where,  $\vec{c}$  locates the tire center,  $\vec{p}$  establishes the contact point, Q is the ground reference frame, P is the frame at the contact point, and C is the frame at the tire center.  $\hat{P}_1$  is in the wheel plane and tangent to the wheel,  $\hat{P}_3$  is also in the wheel plane and acts along the variable radius, and  $\hat{P}_2$  is chosen to make the coordinate frame right-handed.  $\hat{C}_2$  is equivalent to the symmetry axis ( $\hat{u}_{SymAxis}$ ) and the spin axis of the tire,  $\hat{C}_3$  is in the wheel plane and used as a reference to locate the variable radius ( $\vec{r}_{P/C}$ ), and  $\hat{C}_1$  is chosen to make the frame right-handed.  $\hat{X}$ ,  $\hat{Y}$ , and  $\hat{Z}$  are the unit vectors that represent the ground reference frame. Q is chosen to be equivalent to the ground reference frame and is only required to ensure that the tire forces and moments are in balance.

The tire forces and moments are applied to the center of the tire (point C in Fig. 1) [7], and the tire is modeled as a single spring and damper in the normal direction to determine the normal force. This is valid when the model is used to perform handling analysis on the vehicle, and not durability analysis [7,8,9,10,11,12,13,14,15]. For complete details of each of the steps refer to Bombardier and McPhee [10]. The contact point is determined by varying the radius of the planar disk ( $w$ ) and its orientation ( $\theta$ ) until it comes into contact with the ground (refer to Fig. 2). This is accomplished by solving Eq. 1 and Eq. 2.

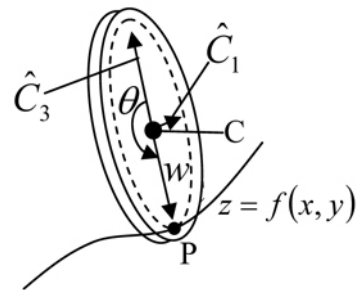


**Figure 2. Determination of the Contact Point**

$$Eq_1 = \vec{p} \cdot \hat{Z} - f(\vec{p} \cdot \hat{X}, \vec{p} \cdot \hat{Y}) = 0 \quad (1)$$

$$Eq_2 = \hat{P}_1 \cdot \hat{u}_{ISOZ} = 0 \quad (2)$$

where, the road is represented as a single elevation function ( $z = f(x, y)$ ), and  $\hat{u}_{ISOZ}$  is the ISO unit vector in the Z direction, which is equivalent to the surface normal at the point of contact.

The first equation (1) states that the thin disk tire must be in contact with the ground, and the second (2) that the tire must be tangent to the road surface at the contact point. The two equations can be expressed in terms of the magnitude of the variable radius,  $w$ , and its orientation,  $\theta$ , which is the angle in the wheel plane between the variable radius and  $\hat{C}_3$  (refer to Fig. 2), using geometrical relations. The updated equations are Eq. 3 and Eq. 4.

$$\left( \begin{Bmatrix} \{c\} + [R_c]^T \begin{Bmatrix} -w \sin(\theta) \\ 0 \\ w \cos(\theta) \end{Bmatrix} \end{Bmatrix} \right) \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} - f \left( \begin{Bmatrix} \{c\} + [R_c]^T \begin{Bmatrix} -w \sin(\theta) \\ 0 \\ w \cos(\theta) \end{Bmatrix} \end{Bmatrix} \right) \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} = 0$$

$$\left( \begin{Bmatrix} \{c\} + [R_c]^T \begin{Bmatrix} -w \sin(\theta) \\ 0 \\ w \cos(\theta) \end{Bmatrix} \end{Bmatrix} \right) \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} = 0$$

(3)

$$\left( [R_c]^T \begin{Bmatrix} -\cos(\theta) \\ 0 \\ -\sin(\theta) \end{Bmatrix} \right) \left( \frac{1}{\sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 + 1}} \begin{Bmatrix} -\frac{\partial z}{\partial x} \\ -\frac{\partial z}{\partial y} \\ 1 \end{Bmatrix} \right) = 0$$

(4)

where  $[R_c]^T$  is the rotation matrix that is used to transform a vector expressed in the tire frame (frame C) to a vector expressed in the ground reference frame. Note that the ISO Z unit vector can be determined because it is equivalent to the surface normal that can be evaluated using vector calculus since the road is defined as a mathematical function (refer to Eq. 4). The current position ( $\{c\}$ ) and orientation ( $[R_c]^T$ ) of the tire are known from the vehicle model and used as inputs to the two non-linear equations in order to solve them. These two equations can be developed using symbolic operations during the formulation stage and used to automatically create the road model procedure.

## AUTOMATIC GENERATION OF THE ROAD MODEL DURING FORMULATION

Symbolic operations can be used during the formulation stage to develop the equations that are employed in the Newton-Raphson iteration method. The Newton-Raphson iteration scheme is chosen to solve the two non-linear equations (Eq. 3 and Eq. 4) because it is quadratically convergent [16]. This permits the iteration scheme to be utilized directly during simulation without requiring any symbolic operations to generate the expressions. In order to accomplish this, Eq. 3

and Eq. 4 need to be expressed into a form that permits them to be incorporated into the Newton-Raphson iteration directly (refer to Eq. 5 and Eq. 6). This is realized using the definition of the scheme based on the known solution of the Jacobian [16] to construct  $\Delta w$  and  $\Delta \theta$ , which are shown in Eq. 8 and Eq. 9.

$$w_{i+1} = w_i + \Delta w$$

(5)

$$\theta_{i+1} = \theta_i + \Delta \theta$$

(6)

$$\det J = \left( \frac{\partial Eq_1}{\partial w} \right) \left( \frac{\partial Eq_2}{\partial \theta} \right) - \left( \frac{\partial Eq_1}{\partial \theta} \right) \left( \frac{\partial Eq_2}{\partial w} \right)$$

(7)

$$\Delta w = \left[ \frac{\partial Eq_1 / \partial \theta}{\det J} \right] Eq_2 - \left[ \frac{\partial Eq_2 / \partial \theta}{\det J} \right] Eq_1$$

(8)

$$\Delta \theta = \left[ \frac{\partial Eq_2 / \partial w}{\det J} \right] Eq_1 - \left[ \frac{\partial Eq_1 / \partial w}{\det J} \right] Eq_2$$

(9)

The Newton-Raphson iteration (Eq. 5 and Eq. 6) needs an initial guess and an appropriate criterion for convergence. The criteria can be the value of each of the equations or the result for both  $w$  and  $\theta$  by monitoring the difference between the results obtained during the current iteration and from the previous one. The latter approach will be used in the present formulation with a tolerance of 0.01. The frame at the tire center (C) is a frame on the geometrical wheel of the tire, and thus will tilt and yaw with the actual tire but will not spin with it. Therefore, an appropriate initial guess is  $w = -R_w$  and  $\theta = 0$ , since  $\hat{C}_3$  is likely pointing in the same direction as the global Z axis (refer to Fig. 1 and 2).

The main disadvantage with the Newton-Raphson iteration scheme is that it requires the computation of the Jacobian, and it can become ill-conditioned if its determinant is close to zero. The Jacobian will become ill-conditioned if the value of  $\theta$  is such that  $\hat{P}_1$  is close to parallel to  $\hat{u}_{ISOZ}$ . It was found that if the value of the determinant of the Jacobian ( $\det J$ ) is smaller than 0.8, then the Newton-Raphson iteration will take more than 3 iterations to converge or may even diverge. This was determined by performing a series of simulations by varying the initial guess for  $\theta$  and monitoring the value of the determinant of the Jacobian and the number of iterations required to achieve convergence. Thus, the determinant of the Jacobian should be monitored and the initial guess for  $\theta$

changed by  $90^\circ$  when  $J \leq 0.8$  so that the system will converge in 3 iterations or less.

The iterative scheme needs to be built around the two non-linear equations obtained using symbolic operations (Eq. 8 and Eq. 9), and constructed into a road model procedure that is automatically generated during formulation. This permits the contact point to be calculated for each tire at each time step during simulation so that the tire forces and moments can be evaluated. The steps to automatically create the road model procedure during the formulation stage are as follows:

- Compute  $\Delta w$ ,  $\Delta \theta$  and  $\det J$  based on the road function  $z = f(x, y)$ , and join the results into a single column vector by assigning each equation to a component of NR\_OUT; refer to Fig. 2
- Calculate the ISO unit vectors, the variable radius ( $\bar{r}_{F/C}$ ), and the loaded radius ( $R_l$ ), and combine the results into a single column vector by assigning each equation to a component of NRIteration; refer to Fig. 3
- Generate the first part of the middle of the procedure as a string by denoting that all three of the components of the first column vector (all three of the NR\_OUTs) be assigned to their appropriate equation; refer to s\_middle1 in Fig. 4
- Create the second segment of the middle of the procedure as a string by indicating the criteria for the determinant to ensure that the Jacobian is not ill-conditioned; refer to s\_middle2 in Fig. 4
- Produce the next portion as a string by creating the Newton-Raphson method utilizing the result for  $\Delta w$  and  $\Delta \theta$ , and by specifying the criteria for convergence; refer to s\_middle3 in Fig. 4
- Construct the final section of the middle part as a string by specifying that all of the components in the second column vector (all of the NRIterations) be assigned to their appropriate equation; refer to s\_middle4 in Fig. 4
- Create the top portion of the procedure as a string by assigning the position and orientation of the tire center to their appropriate variables and by initializing a guess for  $w$  and  $\theta$ ; refer to s\_top in Fig. 4
- Generate the end part of the procedure as a string by ending the procedure; refer to s\_bottom in Fig. 4
- Parse all of the steps together to create the final road model procedure; refer to Fig. 4

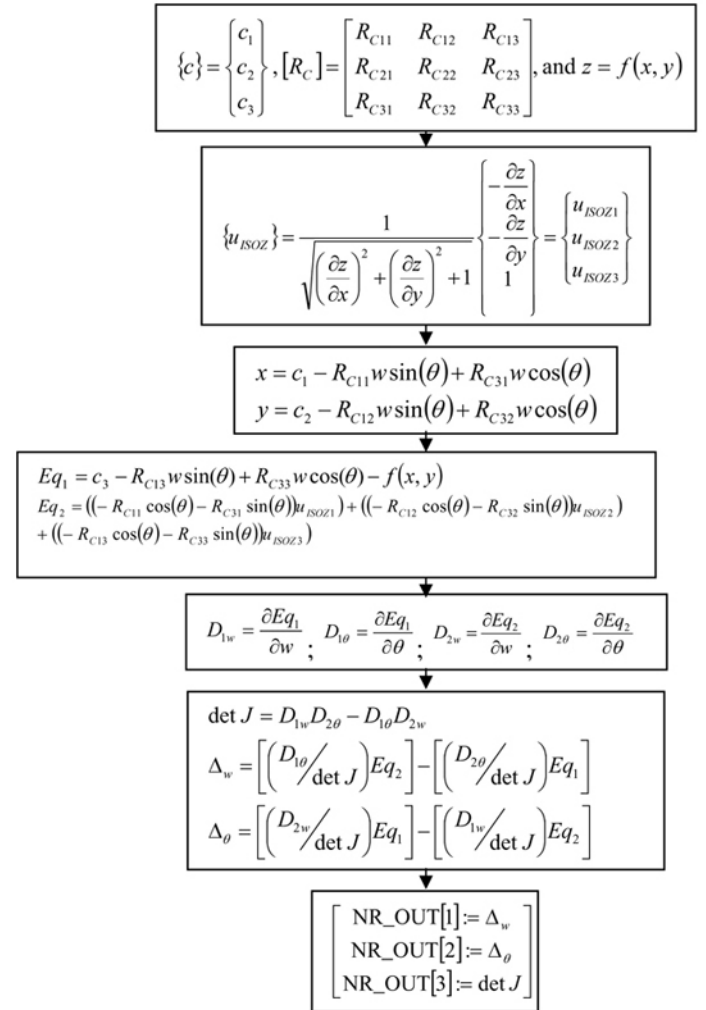


**Figure 2. The Flow Diagram to Build the Two Non-Linear Equations Using Symbolic Operations**

One of the major advantages of a symbolic formulation approach is that symbolic simplification and optimization techniques can be used to improve the computational cost associated with evaluating expressions/procedures. Maple has a CodeGeneration package that will identify repeated terms within a procedure and store them as temporary variables allowing them to only have to be evaluated once and used several times [17]. Also, during the optimization, variables that are evaluated somewhere in the procedure that do not contribute to the evaluation of the output (or outputs) are dropped.

Symbolic simplification techniques are utilized while the road model procedure is being generated to optimize the evaluation of the expressions in the procedure. Therefore the NR\_OUTs and NRIterations are combined into a single column vector and the CodeGeneration package used to optimize the evaluation of each of the components. The road model procedure is automatically formulated around the optimized equations instead of around the NR\_OUTs and