Chapter 4
Computer Simulation

In this chapter, the computer program used to simulate the vehicle roll-plane model will be discussed. This discussion starts with an explanation of the main program, aided by a detailed flow chart. Next, the subroutines called within the main program are discussed. The subroutine called within the main body of the program depends on whether a passive or semiactive damper is simulated. For passive damping, no subroutine is called and a constant damper value is set. For semiactive damping, a subroutine is called and used to simulate on-off skyhook, continuous skyhook, on-off groundhook, or fuzzy logic damper control to obtain the damping value. To conclude the chapter, validation of the simulation program is accomplished.

4.1 Main Program Body

The simulation of the roll-plane vehicle model dynamics was accomplished through a MATLAB program [22]. The organization of the main body of this program is shown in the flow chart of Figure 4.1.

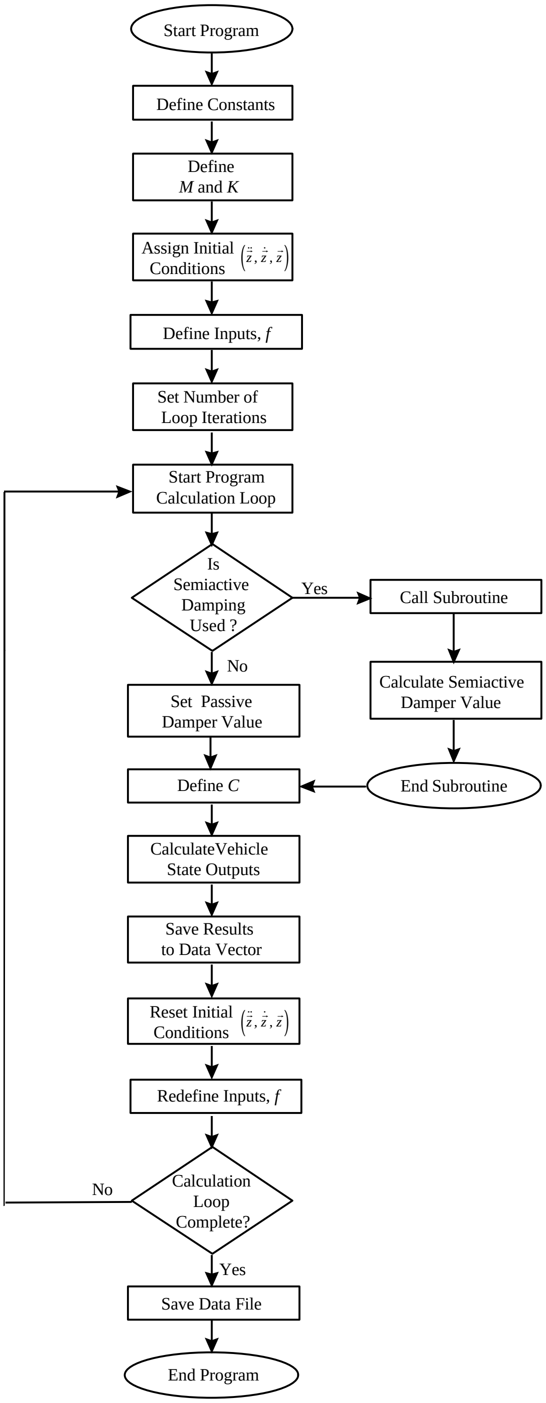
Start Program
Define Constants
Define M and K
Assign Initial Conditions $\left(\ddot{\vec{z}}, \dot{\vec{z}}, \vec{z}\right)$
Define Inputs, f
Set Number of Loop Iterations
Start Program Calculation Loop
Is Semiactive Damping Used ?
Yes
Call Subroutine
Calculate Semiactive Damper Value
End Subroutine
No
Set Passive Damper Value
Define C
CalculateVehicle State Outputs
Save Results to Data Vector
Reset Initial Conditions $\left(\ddot{\vec{z}}, \dot{\vec{z}}, \vec{z}\right)$
Redefine Inputs, f
No
Calculation Loop Complete?
Yes
Save Data File
End Program

Figure 4.1. Flow Chart for the Roll-Plane Simulation Model

As detailed in this flow chart, the program begins with a definition of the constants of the vehicle parameters of Table 3.1. These include the mass, spring stiffness, roll moment, and vehicle width dimensions. With these vehicle parameters defined, the M and K

matrices of Equation (3.5) are determined. Next, the initial conditions and inputs of the vehicle model are assigned. These include the displacement, velocity, and acceleration state vectors ($\bar{z}$, $\dot{\bar{z}}$, and $\ddot{\bar{z}}$) and the input force vector, f, of Equation (3.5). The time differential of each iteration and final time of the program are now defined, setting the total number of iterations to be performed by the calculation loop of the program. The calculation loop is now ready to begin execution.

The first part of the calculation loop determines the value of damping for each damper. If passive damping is applied, the damper is set to a constant value, not changing within the loop. If semiactive damping is applied, a subroutine is called to calculate the current damper value based on the control policy applied. The subroutines called within the program are explained in Section 4.2. Once the damping values for the left and right dampers, $c_{s,l}$ and $c_{s,r}$, have been determined, the C matrix of Equation (3.5) can be formulated, completing the definition of all components of the vehicle model.

The next state can now be computed from the differential equations. This computation is performed by what is commonly referred to as the "Newmark-Beta" method of differential equation solution [23-24]. In this method, the acceleration of the system is assumed to be constant. Therefore, the acceleration of the system at any time within the interval, $\ddot{\bar{z}}$, can be estimated by:

$$\ddot{\bar{z}} = \frac{\left(\ddot{\bar{z}}_0 + \ddot{\bar{z}}_1\right)}{2} \tag{4.1}$$

where the acceleration states at time zero and at the next time interval are respectively represented by $\ddot{\bar{z}}_0$ and $\ddot{\bar{z}}_1$. Equation (4.1) can then be integrated once to find the velocity state of the system, $\dot{\bar{z}}_1$, and twice to find the system displacement state, $\bar{z}_1$, at the next time interval. The results of these integrations are shown in Equations (4.2) and (4.3). In these equations, the change in time is represented by Δt.

$$\dot{\bar{z}}_1 = \dot{\bar{z}}_0 + \frac{\left[\left(\ddot{\bar{z}}_0 + \ddot{\bar{z}}_1\right)\Delta t\right]}{2} \tag{4.2}$$

$$\bar{z}_1 = \bar{z}_0 + \dot{\bar{z}}_0 \Delta t + \frac{\left[\left(\ddot{\bar{z}}_0 + \ddot{\bar{z}}_1\right)\Delta t^2\right]}{4} \tag{4.3}$$

In the simulation program, the states at time zero were defined by the initial conditions of the system. Since the acceleration state at the next time interval is unknown, the states at time zero and time one are first assumed to be equal. Equations (4.2) and (4.3) are then calculated using this assumption. The acceleration state of the system at the next time interval is then determined by solution of the system equation of Equation (3.5), shown by Equation (4.4).

$$\ddot{\bar{z}}_1 = M^{-1}\left(f - C\dot{\bar{z}}_1 - K\bar{z}_1\right) \tag{4.4}$$

Equations (4.2) and (4.3) are then recalculated with the new value of the acceleration. The initial conditions remain unchanged, and Equation (4.4) is recalculated. This process continues until the acceleration of Equation (4.4) converges to within 0.001 of the previous result. The initial conditions of the model states are reset to the new state values, and the force vector , f , is redefined according to the profile of the force input.

The next loop of the program is now ready to be calculated, continuing for the pre-defined number of iterations, until the calculation loop are completed and the program is halted. The data from each iteration calculation are saved to a data vector. The data files of these vectors are then saved at the completion of the calculation loop.

4.2 Program Subroutines

During execution of the main body of the program, one of four subroutines may be called. These subroutines are used to simulate the application of any one of four semiactive control policies studied. These semiactive control policies include on-off skyhook, continuous skyhook, on-off groundhook, and fuzzy logic damper control schemes, as explained in detail in Chapters 2 and 3. The flow chart summarizing the implementation of each semiactive subroutine will be discussed in the Sections 4.2.1 to 4.2.4.

4.2.1 On-Off Skyhook Control

One of the subroutines executed within the main program body was used to apply the concept of on-off skyhook semiactive control. The flow chart summarizing the subroutine applying the on-off skyhook policy of damping control is shown in Figure 4.2.

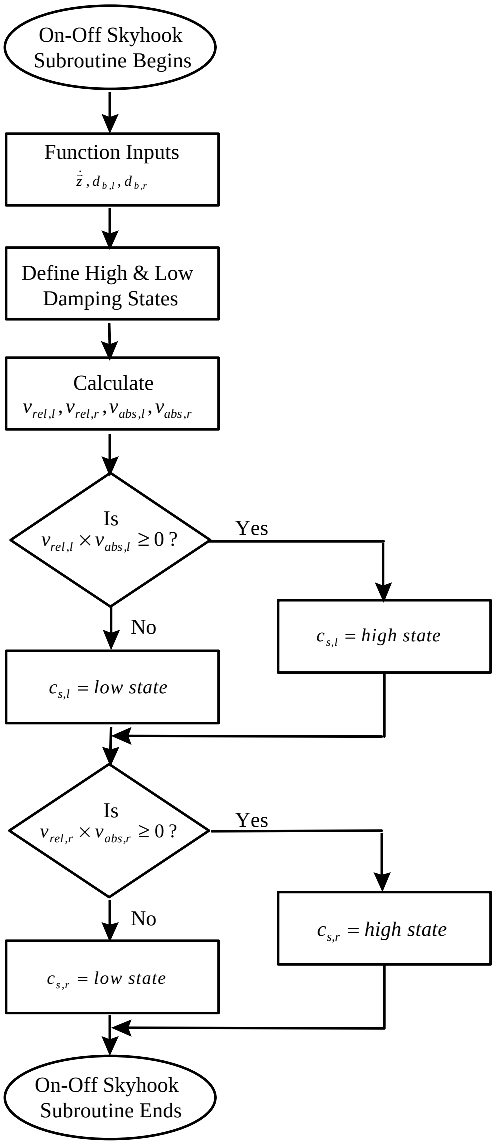
On-Off Skyhook Subroutine Begins
Function Inputs
$\dot{\vec{z}}, d_{b,l}, d_{b,r}$
Define High & Low Damping States
Calculate
$v_{rel,l}, v_{rel,r}, v_{abs,l}, v_{abs,r}$
Is $v_{rel,l} \times v_{abs,l} \geq 0$?
Yes
No
$c_{s,l} = high\ state$
$c_{s,l} = low\ state$
Is $v_{rel,r} \times v_{abs,r} \geq 0$?
Yes
No
$c_{s,r} = high\ state$
$c_{s,r} = low\ state$
On-Off Skyhook Subroutine Ends

Figure 4.2. Flow Chart for On-Off Skyhook Control Policy Subroutine

When the subroutine function is called, specified values from the main program are inserted into the subroutine. For the on-off skyhook control subroutine, the necessary inputs include the state velocity vector, $\dot{\bar{z}}$, and the distances from each suspension to the center of gravity, $d_{b,l}$ and $d_{b,r}$. With these inputs, the next two steps of the subroutine may be performed. First, the high and low state damping values of $c_{s,l}$ and $c_{s,r}$ are defined. Then the relative velocity across the dampers, $v_{rel,l}$ and $v_{rel,r}$, and the absolute velocities of the vehicle body, $v_{abs,l}$ and $v_{abs,r}$, are determined. These values are calculated by Equations (3.6) and (3.7), using the velocity and dimension inputs $\dot{\bar{z}}$, $d_{b,l}$, and $d_{b,r}$.

Next, the on-off skyhook control policy is applied by the subroutine. First, the program evaluates the product of $v_{abs,l}$ and $v_{rel,l}$, and applies the high-state or low-state damping to the left damper, as formulated for the policy by Equation (3.8). If the product is positive or greater than zero, the damper is adjusted to its high state, and if the product is negative, it is adjusted to its low state. The program then evaluates the product of $v_{abs,r}$ and $v_{rel,r}$, and repeats the process of damper adjustment for the right damper, according to Equation (3.9). The subroutine has now been completed and the adjusted damper values for $c_{s,l}$ and $c_{s,r}$, based on the on-off skyhook control policy, are applied to the main program.

4.2.2 Continuous Skyhook Control

The subroutine applying the concept of continuous skyhook control is nearly identical to the subroutine used for the execution of the on-off skyhook control application of Section 4.2.1. The flow chart summarizing the continuous skyhook control policy subroutine is shown in Figure 4.3.

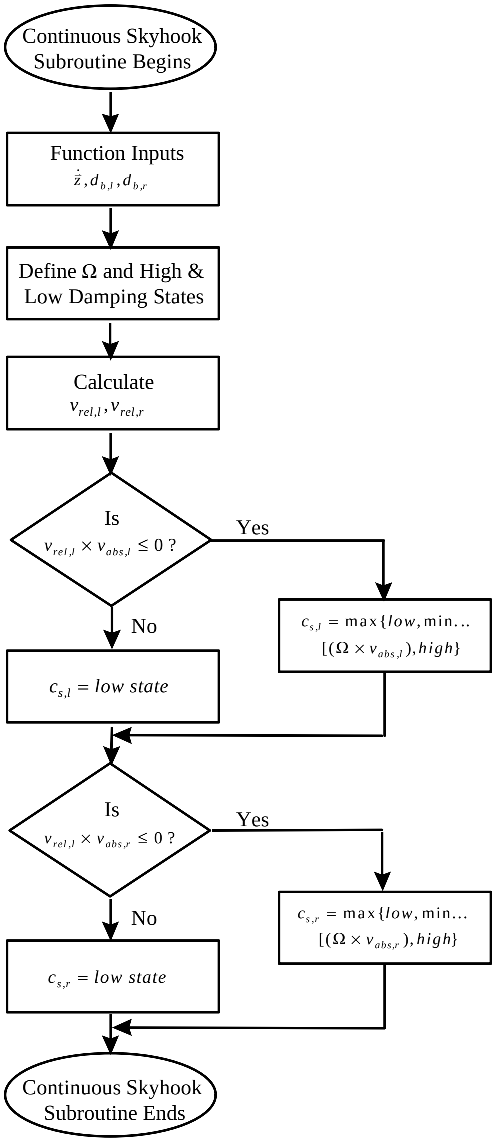
Continuous Skyhook Subroutine Begins
Function Inputs
$\dot{z}, d_{b,l}, d_{b,r}$
Define Ω and High & Low Damping States
Calculate
$v_{rel,l}, v_{rel,r}$
Is
$v_{rel,l} \times v_{abs,l} \leq 0$?
Yes
$c_{s,l} = \max\{low, \min\ldots$
$[(\Omega \times v_{abs,l}), high\}$
No
$c_{s,l} = low\ state$
Is
$v_{rel,l} \times v_{abs,r} \leq 0$?
Yes
$c_{s,r} = \max\{low, \min\ldots$
$[(\Omega \times v_{abs,r}), high\}$
No
$c_{s,r} = low\ state$
Continuous Skyhook Subroutine Ends

Figure 4.3. Flow Chart for Continuous Skyhook Control Policy Subroutine

As with on-off skyhook control, the function inputs of $\dot{z}$, $d_{b,l}$, and $d_{b,r}$ are read into the subroutine, the high and low state damping values for $c_{s,l}$ and $c_{s,r}$ are defined, and the relative damper velocities, $v_{rel,l}$ and $v_{rel,r}$, and the absolute body velocities, $v_{abs,l}$ and $v_{abs,r}$, are calculated according to Equations (3.6) and (3.7). With continuous skyhook control, however, the definition of the damper gain value, Ω, is also defined prior to the policy execution.

After defining and calculating the necessary parameters, the next step in the continuous skyhook control policy is the execution of the subroutine decision blocks, which apply Equations (3.10) and (3.11) for adjustment of the vehicle model dampers. First, the product of $v_{abs,l}$ and $v_{rel,l}$ is calculated. As with the on-off skyhook control policy, if the product is less than zero, the low state value of damping is applied by the left damper, $c_{s,l}$. If, however, this product is not negative, then the damper constant is set equal to the gain of the damper, Ω, times $v_{abs,l}$, constrained within the high and low state limits. The value of damping to be applied to the right damper, $c_{s,r}$, is then determined by execution of the final decision block. The product of $v_{abs,r}$ and $v_{rel,r}$ is calculated, and the same logic applied for adjustment of the left damper is used. With the completion of this step, the subroutine is finished and the adjusted damper values for $c_{s,l}$ and $c_{s,r}$, based on the continuous skyhook control policy, are applied to the main program.

4.2.3 On-Off Groundhook Control

Figure 4.4 shows the on-off groundhook control policy.

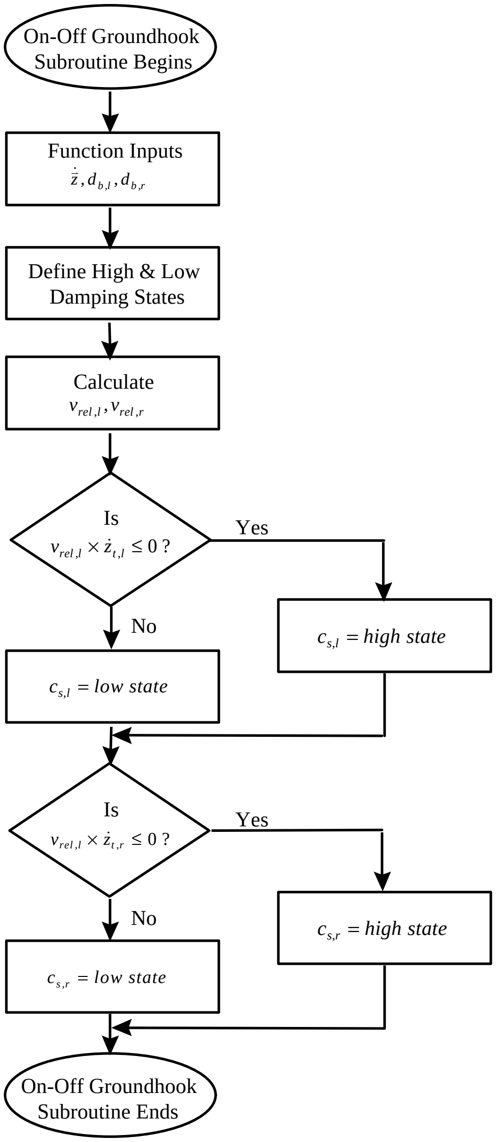
On-Off Groundhook Subroutine Begins
Function Inputs
$\dot{\vec{z}}, d_{b,l}, d_{b,r}$
Define High & Low Damping States
Calculate
$v_{rel,l}, v_{rel,r}$
Is
$v_{rel,l} \times \dot{z}_{t,l} \leq 0$?
Yes
No
$c_{s,l}$ = *high state*
$c_{s,l}$ = *low state*
Is
$v_{rel,l} \times \dot{z}_{t,r} \leq 0$?
Yes
No
$c_{s,r}$ = *high state*
$c_{s,r}$ = *low state*
On-Off Groundhook Subroutine Ends

Figure 4.4. Flow Chart for On-Off Groundhook Control Policy Subroutine

The first three steps in the execution of this subroutine are similar to those used in the on-off and continuous skyhook control policy subroutines. As with these, in the first step, the function inputs of $\dot{\vec{z}}$, $d_{b,l}$, and $d_{b,r}$ are read into the subroutine. The high-state and low-state damping for $c_{s,l}$ and $c_{s,r}$ are then defined, and imperative velocities $v_{rel,l}$ and $v_{rel,r}$ are calculated according to Equation (3.6).

The on-off groundhook control policy is then executed according to Equations (3.12) and (3.13). First, adjustment of the left damper, $c_{s,l}$, is determined by the result of the product of the left tire heave, $\dot{z}_{t,l}$, with the relative velocity across the left damper, $v_{rel,l}$. If this product is less than or equal to zero, high-state damping is applied. However, if this product is positive, the damper is adjusted to its low state. This logic is then applied to the right damper, $c_{s,r}$, for the result of the product of the right tire heave, $\dot{z}_{t,r}$, with the relative velocity across the right damper, $v_{rel,r}$. With the completion of this step, the damper values for $c_{s,l}$ and $c_{s,r}$, based on the on-off groundhook control policy, are applied to the main program and the subroutine is finished.

4.2.4 Fuzzy Logic Control

One of the subroutines that may be called by the main program determines the damper values of the vehicle model through the use of a fuzzy logic control policy. The flow chart explaining this subroutine is shown in Figure 4.5.

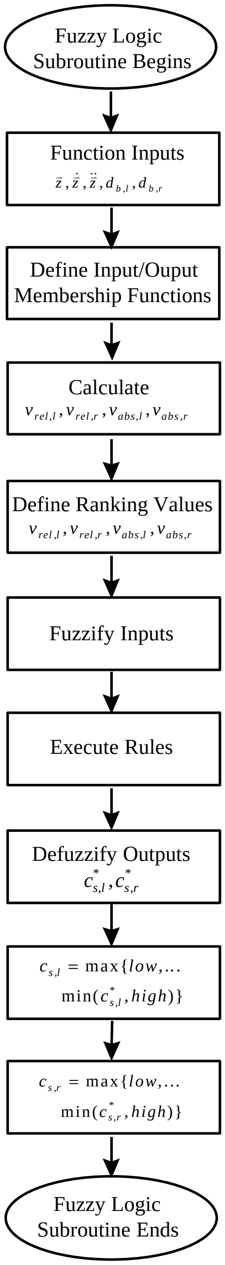
Fuzzy Logic Subroutine Begins
Function Inputs
$\bar{z}, \dot{\bar{z}}, \ddot{\bar{z}}, d_{b,l}, d_{b,r}$
Define Input/Ouput Membership Functions
Calculate
$v_{rel,l}, v_{rel,r}, v_{abs,l}, v_{abs,r}$
Define Ranking Values
$v_{rel,l}, v_{rel,r}, v_{abs,l}, v_{abs,r}$
Fuzzify Inputs
Execute Rules
Defuzzify Outputs
$c^*_{s,l}, c^*_{s,r}$
$c_{s,l} = \max\{low,\ldots \min(c^*_{s,l}, high)\}$
$c_{s,r} = \max\{low,\ldots \min(c^*_{s,r}, high)\}$
Fuzzy Logic Subroutine Ends

Figure 4.5. Flow Chart for Fuzzy Logic Control Policy Subroutine

As with all the subroutines, the first step is to read in the necessary values from the program main body. For the fuzzy logic subroutine, these values included the state vectors, $\bar{z}$, $\dot{\bar{z}}$, and $\ddot{\bar{z}}$, and the vehicle width dimensions, $d_{b,l}$ and $d_{b,r}$. With these inputs, the absolute body and relative damper velocities $v_{rel,l}$, $v_{rel,r}$, $v_{abs,l}$ and $v_{abs,r}$ are calculated. Next, the linguistic variable maximums of the membership function inputs and outputs of Figures 3.3, 3.4, and 3.5 are defined, along with the ranking functions of Equations (3.16) and (3.17).

With the definition of all the necessary parameters complete, the next section of the subroutine is to apply the three fuzzy logic controller steps: fuzzify the controller inputs, execute the rules of the controller, and defuzzify the outputs to be applied to the dampers being controlled. In the first step, fuzzification of the controller inputs is performed. These inputs and their membership functions were discussed in detail in Section 3.3.1 Explained by the example in Section 2.4.1, in fuzzification, each input is assigned a weighting value for each linguistic variable that it intersects. In the subroutine, this objective is performed with the use of similar triangles, as illustrated in Figure 4.6.

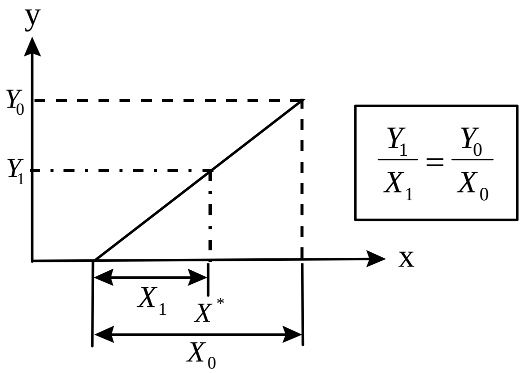

Figure 4.6. Similar Triangle Solution Example

In this figure, a section of the input membership functions of Figures 3.3 and 3.4 is represented. The values of the x axis represent the linguistic variable values, and the values of the y axis represent the weighting functions. The "X_0" and "Y_0" values can be determined from the given shape of the membership function. For all functions used for this research, the "Y_0" value is equal to one. The value for "X_1" can then be calculated from the given input value, X^*, and the definition of the membership function. Thus, the value of the weighting function, Y_1, is calculated based on the idea of similar triangles, according to [25], as shown by:

$$Y_1 = \frac{X_1}{X_0} \tag{4.5}$$

In the second step of the fuzzy logic controller, the rules of the controller are executed. These rules are shown in tabular form in Figures 3.6 and 3.7. Actual execution of the controller rules is performed in the subroutine program through the use of “IF-THEN” statements, as was discussed in detail in Section 2.4.2. In the third and final step of the controller, defuzzification of the outputs is performed. The crisp output value of the left damper, $c_{s,l}^*$, is calculated by Equation (3.16), and the crisp output value of the right damper, $c_{s,r}^*$, is calculated by Equation (3.17). To conclude the fuzzy logic control subroutine, the crisp outputs are adjusted, if necessary, to be maintained within the high- and low-state limits of the dampers. Thus, the results of crisp output adjustment performed by Equations (3.18) and (3.19) determine the actual values of the left and right dampers, $c_{s,l}$ and $c_{s,r}$, to be applied by the fuzzy logic control policy to the main body of the program.

4.3 Simulation Validation

In the computer simulation program used for this research, the response of the system is generated using the Newmark-Beta integration method discussed in Section 4.1. As a method of validation, the response characteristics of the theoretical uncoupled system were determined using a procedure detailed by Ahmadian [26-27]. These estimated characteristics are then compared with those of the actual system response. The peak time, settling time, percent overshoot, and steady-state value of the responses to a step input were included in this comparison.

4.3.1 Formulation of the Uncoupled System

In this section, a method for uncoupling the differential equations of the modeled system is presented. The transfer function of the uncoupled system will then be developed. When the theoretical characteristics of the uncoupled system are determined from the transfer function of the uncoupled system, they will contain some error when compared with the parameters of the actual system response. The reason for this error will be explained during the uncoupling procedure.

Taking Equation (3.5):

$$M\ddot{\bar{z}} + C\dot{\bar{z}} + K\bar{z} = f \tag{4.6}$$

and premultiplying it by the inverse of the mass matrix, M^{-1}, yields:

$$\ddot{\bar{z}} + M^{-1}C\dot{\bar{z}} + M^{-1}K\bar{z} = M^{-1}f \tag{4.7}$$

Premultiplying Equation (4.7) by Φ^{-1} and using the assumption that $\bar{z} = \Phi\bar{x}$ then results in:

$$\ddot{\bar{x}} + \Phi^{-1}M^{-1}C\Phi\dot{\bar{x}} + \Phi^{-1}M^{-1}K\Phi\bar{x} = \Phi^{-1}M^{-1}f \tag{4.8}$$

where Φ represents the eigenvector matrix of $M^{-1}K$ [28].

Using the following substitutions :

Λ_c = diagonal elements of " $\Phi^{-1}M^{-1}C\Phi$ "

Λ_k = eigenvalues of " $M^{-1}K$ "

the approximated uncoupled system equation is determined as:

$$\ddot{\bar{x}} + \Lambda_c\dot{\bar{x}} + \Lambda_k\bar{x} = \Phi^{-1}M^{-1}f \tag{4.9}$$

In Equation (4.9), the off-diagonal elements of $\Phi^{-1}M^{-1}C\Phi$ are ignored in order to obtain an uncoupled set of differential equations.

Solving Equation (4.9) for $\vec{x}$ yields:

$$\vec{x} = \frac{\Phi^{-1}M^{-1}f}{s^2 + \Lambda_c s + \Lambda_k} \tag{4.10}$$

Recalling that $\vec{z} = \Phi\vec{x}$, the transfer function of the estimated uncoupled response is determined by

$$\vec{z} = \frac{M^{-1}f}{s^2 + \Lambda_c s + \Lambda_k} \tag{4.11}$$

4.3.2 Derivation of the Theoretical and Actual Response Parameters

In this section, the transient response characteristics, such as peak time, settling time, percent overshoot, and steady-state value, to a step input are formulated for the uncoupled system. The specifications of these characteristics are then given as they pertain to an actual response.

The dynamic characteristics of a second-order differential equation, namely the natural frequency (ω_n) and damping ratio (ζ) of the uncoupled response of Equation (4.6), can be determined as [29]:

$$\omega_n = \sqrt{\Lambda_k} \tag{4.12}$$

$$\zeta = \frac{\Lambda_c}{2\omega_n} \tag{4.13}$$

Furthermore, the peak time (T_p), percent overshoot ($\%OS$), settling time (T_s), and the steady state value of the system transient response to a step input ($\bar{z}_{SS}$) can be calculated as

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \tag{4.14}$$

$$\%OS = e^{-\left(\zeta\pi \big/ \sqrt{1-\zeta^2} \right)} \times 100 \tag{4.15}$$

$$T_s = \frac{4}{\zeta\omega_n} \tag{4.16}$$

$$\bar{z}_{SS} = \lim_{s \to 0} \left[\frac{M^{-1} f}{s^2 + \Lambda_c s + \Lambda_k} \right] = \frac{M^{-1} f}{\Lambda_k} \tag{4.17}$$

Figure 4.7 shows a system transient response, $y(t)$, to a step input.

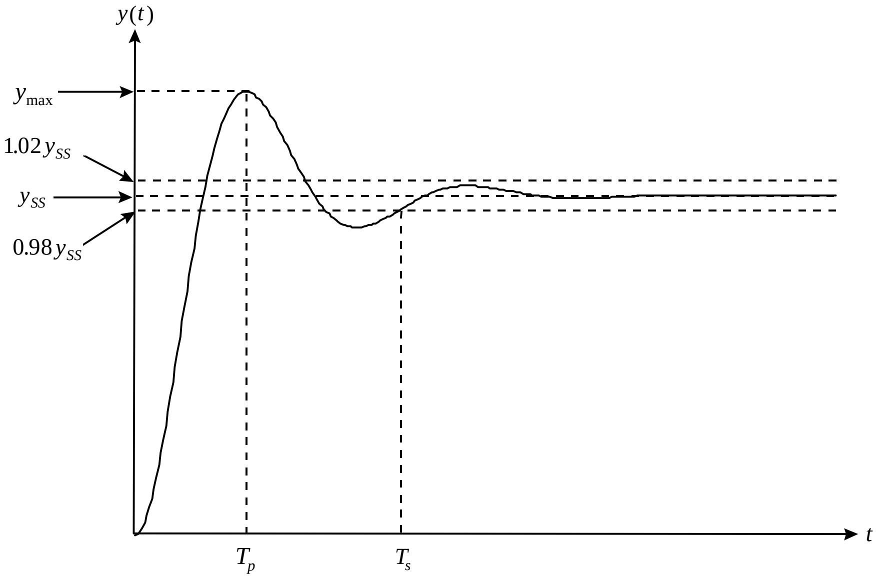

Figure 4.7. Step Input Response of a Damped System

The peak time is the amount of time it takes the response to reach its maximum value, $y_{\max}$. Settling time is the time it takes the response to stay within $\pm 2\%$ of the steady-state value, y_{SS}. The percent overshoot of the actual response is determined by

$$\%OS = \left[\frac{y_{\max} - y_{SS}}{y_{SS}} \right] \times 100 \tag{4.18}$$

4.3.3 Comparison of the Theoretical and Actual System Response Parameters

In Section 4.3.2, the characteristics of a transient response were derived for both the theoretical uncoupled system and the actual simulation response. In this section, the values of these characteristics are calculated and compared.

The vehicle parameters used for the comparison of the theoretical and actual responses are given in Table 4.1.

Table 4.1. Vehicle Parameters Used for Response Comparison

Symbol	Value	Units
$c_{s,l}$	55	lb-sec/in
$c_{s,r}$	55	lb-sec/in
$d_{b,l}$	48	inch
$d_{b,r}$	48	inch
I_{xx}	21,540	slug-in^2
$k_{s,l}$	1200	lb/in
$k_{s,r}$	1200	lb/in
$k_{t,l}$	7450	lb/in
$k_{t,r}$	7450	lb/in
m_b	26.4	slug
$m_{t,l}$	2.33	slug
$m_{t,r}$	2.33	slug

With the remaining components set equal to zero, one component of the force vector, f , was held to a constant value, simulating a step input. The values of the inputs were:

- Body Heave Input, $F_b = 2000\ lb$
- Body Roll Input, $T_b = 50{,}000\ in - lb$
- Road Input, Left Tire, $z_{in,l} = 1\ inch$
- Road Input, Right Tire, $z_{in,r} = 1\ inch$

When the body heave or body roll force inputs were applied, the response of the corresponding body heave or body roll was studied; when the road inputs were applied to the left and right tires, the corresponding tire heave response was examined. The results of these four responses, for both the uncoupled and actual systems, are shown in Figure 4.8.

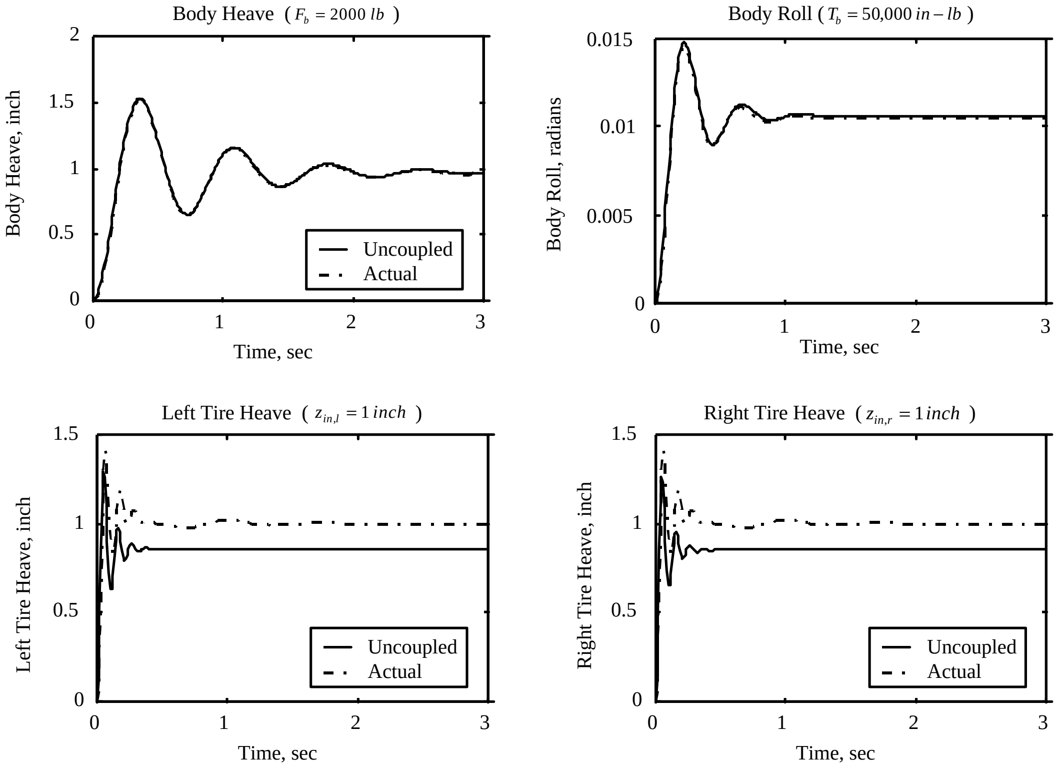

Figure 4.8. Theoretical and Actual Responses to a Step Input

From the responses of the theoretical and actual systems of Figure 4.8, the values for the peak time, settling time, percent overshoot, and steady-state were estimated using the "actual" techniques described in Section 4.3.2 and compared with the calculated "theoretical" results. The percent error between the theoretical and actual system responses is used to quantify the comparison.

$$\%Error = \left[\frac{|Theoretical - Actual|}{Theoretical} \right] \times 100 \tag{4.19}$$

The results are shown in Table 4.2.

Table 4.2. Comparison of the Theoretical and Actual Response Characteristics

	Input	T_p **(sec)**	T_s **(sec)**	$\bar{z}_{SS}$	**%*OS***
Theoretical Actual	$F_h = 2000\ lb$ $F_b = 2000\ lb$	0.3611 0.3700	2.6153 2.2900	0.9709 (in) 0.9700 (in)	57.56 57.48
%Error		***2.46***	***12.44***	***0.09***	***0.14***
Theoretical Actual	$T_h = 50{,}000\ in - lb$ $T_b = 50{,}000\ in - lb$	0.2216 0.2300	0.9455 0.8975	0.0106 (rad) 0.0105 (rad)	39.16 38.00
%Error		***3.79***	***5.08***	***0.94***	***2.96***
Theoretical Actual	$z_{in\ l} = 1\ inch$ $z_{in,l} = 1\ inch$	0.0526 0.0700	0.3236 0.7540	0.8583 (in) 1.0000 (in)	52.23 41.74
%Error		***33.08***	***133.00***	***16.28***	***20.08***
Theoretical Actual	$z_{in\ r} = 1\ inch$ $z_{in,r} = 1\ inch$	0.0526 0.0700	0.2972 0.7540	0.8526 (in) 1.0000 (in)	49.28 41.74
%Error		***33.08***	***153.70***	***17.65***	***15.30***

In Table 4.2, the body heave and roll step input responses are found to agree relatively well with the theoretical values. A maximum error of 12.44% is obtained, with the next largest error at only 5.08%. The errors of the tire heave responses were not as promising. The error for the settling time of the tire responses averages to 143.35%, with the remaining errors lingering between 15.30% and 33.08%. These large errors are expected, however, due to the fact that the tire masses are significantly smaller than the body mass.

Thus, the dynamic coupling that was ignored during the uncoupling process has a much greater effect on the axles than on the vehicle body.