FURTHER DEVELOPMENTS IN THE SIMULATION OF AUTOMOBILE HANDLING

by

D. L. Wilson

R. A. Scott

Technical Report, Department of Mechanical Engineering and Applied Mechanics

UM-MEAM-82-2

TABLE OF CONTENTS

Chapter 1. Introduction

Recently , under contract No. DOT-HS7-01715, the authors were involved in the creation of some all-digital simulations for both open and closed-loop automobile maneuvers. Specific achievements of that project were the development of:

(1) A vehicle module IDSFC (Improved Digital Simulation Fully Comprehensive).

(2) A driver module DRIVER, which involved several mathematical models of the human driver. Such a module is required for closed loop maneuvers

Overall features of these modules are as follows. Full details can be found in Refs. [1.1] to [1.9].

(i) IDSFC. The vehicle model IDSFC involves the following degrees of freedom:

Sprung Mass. Specification of the sprung mass requires 3 translational and 3 rotational degrees of freedom.

Front Unsprung Masses. The degrees of freedom allowed are 2 wheel hops, 2 wheel spins, 2 wheel rotations about the kingpins and 1 steering connecting rod displacement. To reduce costs, the steering system is handled statically.

Rear Unsprung Masses.

A. Solid Rear Axle. The degrees of freedom allowed are 1 rear suspension deflection, 1 rear axle roll and 2 wheel spins.

B. Independent Rear Suspensions. The degrees of freedom allowed are 2 rear suspension deflections and 2 wheel spins.

The mathematical representation of the vehicle model involves 30 first order nonlinear differential equations and approximately 250 algebraic equations. The digital program contains 30 subroutines and both single precision and double precision versions are available.

The vehicle simulation capabilities are basically as follows:

(1) Straight line braking/acceleration, cornering without braking/acceleration and cornering with braking/ acceleration are allowed.

(2) Maneuvers up to and including the limit range can be studied in that (i) Nonlinear terms in the kinematics are retained. (ii) These terms are activated by model level switches and can be deleted for less severe maneuvers, thereby decreasing running costs. These switches can also be employed if the user wishes to do studies on the effects of various nonlinearities. The tire and suspension forces and moments are modeled into the nonlinear range.

(3) For system and user flexibility, two methods are provided for computing tire forces and moments, namely (i) The APL-CALSPAN model which is based on curves fitted to the measured data. (ii) A Partial Data Deck model which directly uses the measured data.

(4) An antilock capacity, which can be activated by a model level switch, is available.

(5) Both solid rear axle and independent rear suspensions are allowed.

(6) Front wheel drive, rear wheel drive and four wheel drive are available.

(7) Separate braking at each wheel is permissible.

(8) An interactive capability is provided, which is activated by a model level switch.

(ii) A driver module (DRIVER), involving several mathematical models of human driving behavior, was also developed. The main features of DRIVER are as follows:

(1) DRIVER controls steering, braking and drive torque inputs to the vehicle model.

(2) There are 5 pre-programmed open-loop maneuvers available, namely:

(a) Sinusoidal steer with trapezoidal braking.

(b) Trapezoidal steer with trapezoidal braking.

(c) Double trapezoidal steer with trapezoidal braking.

(d) Trapezoidal steering with a sinusoidal perturbation with trapezoidal braking.

(e) Sinusoidal steering sweep with no braking.

In addition the driver module will accept:

(i) Any open-loop maneuver supplied by the user in tabular form.

(ii) Any open-loop maneuver specified by a user supplied subroutine.

(3) The driver module can operate in a closed-loop

mode following a desired path. Four control strategies are available, namely:

(a) A "cross-over" model for a straight line path.

(b) A "cross-over" model for an arbitrary path.

(c) A preview-predictor model which uses a geometric predictor.

(d) A preview-predictor model which uses a three-degree-of-freedom vehicle model as a predictor.

(4) The driver module permits a mixed-mode operation which allows combined open and closed loop control.

(5) An obstacle avoidance strategy using the preview-predictor models is available.

Since the completion of that contract, additional work has been done in the areas of: (i) developing a simpler vehicle model, (ii) vehicle asymmetry, and (iii) driver modeling. More specifically: (a) A three-degree-of-freedom vehicle model incorporating certain asymmetries has been developed and a digital program has been written for it. (b) IDSFC has been modified to take into account certain vehicle asymmetries. (c) The module DRIVER has been modified to allow it to interface with the three-degree-of-freedom vehicle model. Also, an improved cross-over driver model has been implemented in it.

The purpose of this report is to provide documentation to enable users of IDSFC and DRIVER to incorporate the above additions/changes. In Chapter 2 a description is given of the mechanical modeling involved in the three-degree-of-freedom vehicle model. Then a discussion of the numerical strategy used is given, as well as a Fortran program listing of the computer code. In Chapter 3, the alterations in the seventeen-degree-of-freedom vehicle model IDSFC that are required to allow for certain asymmetries are presented. Both equation and program changes are given. Chapter 4 is concerned with the module DRIVER. The program alterations required to interface it with the three-degree-of-freedom vehicle model are documented. Also, an extended cross-over model of human driving is described and the program changes required to implement it are detailed.

REFERENCES FOR CHAPTER 1

1.1 "Improvement of Mathematical Models for Simulation of Vehicle Handling, Vol. 4. User's Guide for the General Simulation", W. R. Garrott, D. L Wilson A. M. White and R. A. Scott. Final Report DOT-HS-7-01715, July 1979.

1.2 "Improvement of Mathematical Models for Simulation of Vehicle Handling, Vol. 5, Programmer's Guide for the Vehicle Model, Part 1. Documentation Through Subroutine IOUT. Part 2. Documentation for Subroutines PRTCID through KNMTC. Part 3. Documentation for Subroutines SLPCLC Through OUTPUT and a Double Precision Listing", W. R. Garrott, A. M. White and R. A. Scott. Final Report DOT-HS-7-01715, December, 1979.

1.3 "Improvement of Mathematical Models for Simulation of Vehicle Handling, Vol. 6. Programmer's Guide for the Driver Module", W. R. Garrott, D. Wilson, A. M. White and R. A. Scott. Final Report DOT-HS-7-01715, December 1979.

1.4 "Improvement of Mathematical Models for Simulation of Vehicle Handling, Vol. 7. Technical Manual for the General Simulation", W. R. Garrott and R. A. Scott. Final Report DOT-HS-7-01715, March 1980.

1.5 "Improvement of Mathematical Models for Simulation of Vehicle Handling, Vol. 8. Summary Report", W. R. Garrott and R. A. Scott. Final Report DOT-HS-7-01715, April 1980.

1.6 "A Note on the Calculation of Tire Side Forces", W. R. Garrott and R. A. Scott. Journal of Sound and Vibration, Vol. 78, 1981, pp. 306-309.

1.7 "An All-Digital Simulation for Open and Closed-Loop Automobile Maneuvers", W. R. Garrott, D. L. Wilson and R. A. Scott. Simulation, Sept. 1981, pp. 83-91.

1.8 "Closed-Loop Automobile Maneuvers Using Describing Function Methods", W. R. Garrott, D. L Wilson and R. A. Scott. SAE Paper, No. 820305, Feb. 1982.

1.9 "Closed-Loop Automobile Maneuvers Using Preview-Predictor Models", W. R. Garrott D. L. Wilson and R. A. Scott. SAE Paper, No. 820306, Feb. 1982.

Chapter 2. Three-Degree-of-Freedom Vehicle Model

2.1 Equations of Motion for the Model

A relatively simple mathematical model of a four-wheel vehicle has been developed as an inexpensive and easily manipulated simulation tool. It can be applied to both symmetric, and certain types of unsymmetric, vehicles.

The model possesses three degrees of freedom; namely, translation in a plane (X-Y plane) and rotation about an axis (the Z axis) perpendicular to that plane. The vehicle is considered to be acted upon by gravity, by air resistance, and by contact forces and moments at the four tire-road contact points.

The following asymmetries can be treated:

(1) Addition of a "payload" at any arbitrary location in the vehicle.

(2) Independent tire properties at each wheel.

(3) Torque transfer from the chassis to the driven axle(s).

(4) Asymmetrical brake-torque and drive-torque distribution.

(5) Independent steering system compliance at each front wheel. In addition, the model includes the following features which are particularly important to the asymmetry mechanisms.

(i) Non-linear tire models.

(ii) Variable lateral weight transfer at front and rear (corresponding to front-rear roll stiffness ratio).

Payloads are incorporated by considering a system consisting of two rigidly connected arbitrary rigid bodies. Body 1 will ultimately be interpreted as the nominal (unloaded) vehicle, and Body 2 will be interpreted as a payload.

Let Body 1 be a rigid body (mass m_1) and let xyz be a set of axes fixed in Body 1 with the origin 0 at the center of mass.* Let $[I^1]$ denote the inertia tensor of Body 1 with respect to these axes.

Let Body 2 be another rigid body of mass m_2 which is rigidly fastened to Body 1 such that the center of mass of Body 2 lies at point P which has coordinates (x_p, y_p, z_p). The position of P with respect to 0 is given by

$$\vec{\rho}_p = (x_p\vec{i}+y_p\vec{j}+z_p\vec{k}) \tag{2.1}$$

Define a set of axes $x_2y_2z_2$ with origin at P, which will be parallel to xyz as shown in Fig. 2.1. Let $[I^2]$ denote the inertia tensor of Body 2 with respect to axes $x_2y_2z_2$.

Assume that the only forces and moments acting on Body 2 are gravity and the forces and moments $\vec{F}_{21}$ and $\vec{M}_{21}$ exerted by the connection to Body 2 from Body 1. Then, Newton's laws give

*A List of Symbols is given at the end of the chapter.

$$m_2\vec{a}_2 = m_2\vec{g} + \vec{F}_{21} \tag{2.2}$$

$$\dot{\vec{H}}^2 = \vec{M}_{21} \tag{2.3}$$

The forces and moments acting on Body 1 are gravity, the forces and moments $\vec{F}_{12}$ and $\vec{M}_{12}$ exerted by the connection from Body 2, and $\vec{F}_0$ and $\vec{M}_0$ which are the resultants at 0 of all other forces and moments acting on Body 1 as shown in Fig. 2.2. Then

$$m_1\vec{a}_1 = m_1\vec{g} + \vec{F}_{12} + \vec{F}_0 \tag{2.4}$$

$$\dot{\vec{H}}^1 = \vec{M}_{12} + \vec{M}_0 + \vec{\rho}_p X \vec{F}_{12} \tag{2.5}$$

Noting that

$$\vec{F}_{12} = -\vec{F}_{21} = -m_2\vec{a}_2 + m_2\vec{g},$$

$$\vec{M}_{12} = -\vec{M}_{21} = -\dot{\vec{H}}^2,$$

(2.4) and (2.5) become, using (2.2),

$$m_1\vec{a}_1 = (m_1+m_2)\vec{g} - m_2\vec{a}_2 + \vec{F}_0 \tag{2.6}$$

$$\dot{\vec{H}}^1 = -\dot{\vec{H}}^2 + \vec{\rho}_p X(-m_2\vec{a}_2+m_2\vec{g}) + \vec{M}_0 \tag{2.7}$$

The equation of translational motion will be developed first, from (2.6). The equation of rotational motion will then be developed from (2.7).

Let $\vec{\omega}$ be the angular velocity of Body 1. Then

$$\vec{a}_2 = \vec{a}_1 + \dot{\vec{\omega}} x \vec{\rho}_p + \vec{\omega} x (\vec{\omega} x \vec{\rho}_p) \tag{2.8}$$

Substitution of (2.8) into (2.6) gives the equation of translational motion

$$(m_1+m_2)\vec{a}_1 + m_2 \dot{\vec{\omega}} x \vec{\rho}_p + m_2 \vec{\omega} x (\vec{\omega} x \vec{\rho}_p) = (m_1+m_2)\vec{g} + \vec{F}_0. \tag{2.9}$$

Now specialization is made to the three-degree-of-freedom model, for which

$$\vec{\omega} = r\vec{k} \tag{2.10}$$

$$\vec{g} = g\vec{k} \tag{2.11}$$

Recalling that

$$\left(\frac{d}{dt}\right) = \left(\frac{d}{dt}\right)_{rel} + \vec{\omega} x \tag{2.12}$$

where "rel" stands for relative to the moving frame, it follows that

$$\vec{a}_1 = (\dot{\vec{v}}_1)_{rel} + \vec{\omega} \times \vec{v}_1 \tag{2.13}$$

where $\vec{v}_1$ is the velocity of the point 0, namely,

$$\vec{v}_1 = u\vec{i} + v\vec{j} \tag{2.14}$$

Using (2.10), (2.11), (2.13) and (2.14), (2.9) gives, in component form,

$$\dot{u}(m_1+m_2) - \dot{r}y_p m_2 = (m_1+m_2)vr + m_2 r^2 x_p + F_x \tag{2.15}$$

$$\dot{v}(m_1+m_2) + \dot{r}x_p m_2 = -(m_1+m_2)ur + m_2 r^2 y_p + F_y \tag{2.16}$$

$$0 = (m_1+m_2)g + F_z \tag{2.17}$$

For the model at hand, angular momenta are given by

$$H_n^\gamma = I_{nz}^\gamma \omega_z = I_{nz}^\gamma r, \quad \gamma=1,2,\ n=1,2,3 \tag{2.18}$$

Noting this and the relation (2.12), the component version of (2.7) is, using (2.10) and (2.11)

$$I^S{}_{xz}\dot{r} - I^S{}_{yz}r^2 = m_2(\dot{v}z_p+\dot{r}x_pz_p+urz_p$$

$$-r^2y_pz_p+y_pg) + M_x \qquad (2.19)$$

$$I^S{}_{xz}r^2 + I^S{}_{yz}\dot{r} = m_2(\dot{r}y_pz_p-\dot{u}z_p+vrz_p$$

$$+r^2x_pz_p-x_pg) + M_y \qquad (2.20)$$

$$I^S{}_{zz}\dot{r} = m_2(-\dot{v}x_p-\dot{r}x_p{}^2+\dot{u}y_p-\dot{r}y_p{}^2+urx_p$$

$$-vry_p) + M_z \qquad (2.21)$$

where

$$I^S_{mn} \equiv I^1_{mn} + I^2_{mn} \qquad (2.22)$$

The forces and moments acting on the vehicle are tire forces and moments, aerodynamic effects, and gravity (which is already included in the formulation). Then

$$F_x = \sum_{i=1}^{4} F_{xi} + F_{xaero} \qquad (2.23)$$

$$F_y = \sum_{i=1}^{4} F_{yi} + F_{yaero} \qquad (2.24)$$

$$F_z = \sum_{i=1}^{4} F_{zi} + F_{zaero} \qquad (2.25)$$

$$M_x = \sum_{i=1}^{4} (Y_{ti}F_{zi}-z_{ti}F_{yi}+M_{xi}) + M_{xaero} \qquad (2.26)$$

$$M_y = \sum_{i=1}(z_{ti}F_{xi} - x_{ti}F_{zi} + M_{yi}) + M_{yaero} \qquad (2.27)$$

$$M_z = \sum_{i=1}(x_{ti}F_{yi} - y_{ti}F_{xi} + M_{zi}) + M_{zaero} \qquad (2.28)$$

where F_{xi}, F_{yi}, F_{zi}, M_{xi}, M_{yi}, M_{zi} are the tire forces at, and moments about, the tire contact points; F_{xaero}, F_{yaero}, F_{zaero}, M_{xaero}, M_{yaero}, M_{zaero} are the forces at, and moments about, the vehicle c.m. due to aerodynamic effects, and x_{ti}, y_{ti}, z_{ti} are the coordinates of the tire contact points.

The tire is assumed to operate with zero camber angle in the three-degree-of-freedom model, so F_{xi}, F_{yi}, M_{xi}, M_{yi}, M_{zi} are functions only of F_{zi}, tire drive torque, and slip angle. F_{xaero}, ..., M_{zaero} are functions of vehicle velocity. Equations (2.15),(2.16),(2.17),(2.19),(2.20) and (2.21) thus provide six equations in the seven variables $(\dot{u},\dot{v},\dot{r},F_{z1},F_{z2},F_{z3},F_{z4})$.

A seventh equation is obtained by the following argument. Consider a system consisting of a rigid body which pivots on two solid axles, with torsional springs at the pivots, as in Fig. 2.3. Assume that the axles are parallel, and that there is some angular deflection (roll) of the body with respect to the axles. In addition to the moments caused by the springs, let there be torques T_1 and T_2 transmitted between the body and axles which are independent of θ.

A free-body-diagram of the front axle is shown in Fig. 2.4. If the axle is taken to be massless, the equation of rotational motion about the x-axis becomes

$$F_{z1}y_{t1} + F_{z2}y_{t2} - F_{y1}h_F - F_{y2}h_F + M_{x1} + M_{x2} - T_1 = -k_1\theta \tag{2.29}$$

Note that $y_{t2} < 0$.

The analogous equation for the rear axle is

$$F_{z3}y_{t3} + F_{z4}y_{t4} - F_{y3}h_R - F_{y4}h_R + M_{x3} + M_{x4} - T_2 = -k_2\theta \tag{2.30}$$

Define

$$K = k_1 + k_2 \tag{2.31}$$

$$\lambda_{RS} = k_1/K. \tag{2.32}$$

Then

$$k_1 = \lambda_{RS}K \tag{2.33}$$

$$k_2 = (1-\lambda_{RS})K \tag{2.34}$$

Substitution of (2.33) into (2.30) and (2.34) into (2.31) yields

$$(F_{z1}y_{t1}+F_{z2}y_{t2}-F_{y1}h_F-F_{y2}h_F+M_{x1}+M_{x2}-T_1)/\lambda_{RS} = -K\theta \tag{2.35}$$

$$(F_{z3}Y_{t3}+F_{z4}Y_{t4}-F_{y3}h_R-F_{y4}h_R+M_{x3}+M_{x4}-T_2)/(1-\lambda_{RS}) = -K\theta \tag{2.36}$$

Equating (2.35) and (2.36) yields

$$(1-\lambda_{RS})(F_{z1}Y_{t1}+F_{z2}Y_{t2}) - \lambda_{RS}(F_{z3}Y_{t3}+F_{z4}Y_{t4})$$

$$= (1-\lambda_{RS})(T_1+F_{y1}h_F+F_{y2}h_F-M_{x1}-M_{xz})$$

$$- \lambda_{RS}\ (T_2+F_{y3}h_R+F_{y3}h_R+F_{y4}h_R-M_{x3}-M_{x4}) \tag{2.37}$$

Equation (2.37) holds for all values of K. In the limit as K becomes infinitely large, the two-axle/body system will behave as a rigid body in plane motion (i.e. no rolling). Eq. (2.37) can thus be used to remove the indeterminacy in F_Z for the three-degree-of-freedom model.

The steering system will now be addressed. The front tire forces depend in part on the steer angles of the wheels. The steering system is shown schematically in Fig. 2.5. It is assumed to be massless, so that the deflection of the compliant members can be computed statically. The input to the system is the steering-wheel displacement. The linkage geometry can be specified to yield either parallel steer angles or Ackerman steer angles, as shown in Fig. 2.6. In either case, the simplifying assumption is made that the torque acting at the pitman arm is the sum of the kingpin torques.

The steering system is described by the following equations, where δ_{SW} is the steering-wheel displacement, δ_P is the pitman arm displacement, δ_1 and δ_2 are the front-wheel steer angles, TQ_{ST1} and TQ_{ST2} are the steering torques about the kingpins, NG is the steering ratio, C_{ST1}, C_{ST2}, and C_{ST3} are the steering-system compliances as shown in Fig. 2.5, and δ_{TOE1} AND δ_{TOE2} are toe-in angles.

$$\delta_P = \left\{\delta_{Sw}+C_{ST3}(TQ_{ST1}+TQ_{ST2})/NG\right\}/NG \qquad (2.38)$$

For parallel steer linkage geometry,

$$\delta_1 = \delta_P + C_{ST1}TQ_{ST1} + \delta_{TOE1} \qquad (2.39)$$

$$\delta_2 = \delta_P + C_{ST2}TQ_{ST2} + \delta_{TOE2} \qquad (2.40)$$

For Ackerman steering linkage geometry (assuming a perfectly stiff linkage) the following relations are obtained from geometry, where the wheelbase L and rear-axle center turning radius R_r are defined as in Fig. 2.6.

$$\delta_P = \tan^{-1}(L/R_r) \qquad (2.41)$$

$$\delta_1 = \tan^{-1}(L/(R_r-y_{t1})) \qquad (2.42)$$

$$\delta_2 = \tan^{-1}(L/(R_r-y_{t2})) \qquad (2.43)$$

(note that $y_{t2} < 0$).

Substitution of (2.41) into (2.42) and (2.43) and addition of initial toe-in angle and steering compliance effects yields

$$\delta_1 = \tan^{-1}[L\tan\delta_p/(L-y_{t1}\tan\delta_p)] + C_{ST1}TQ_{ST1}+\delta_{TOE1} \quad (2.44)$$

$$\delta_2 = \tan^{-1}[L\tan\delta_p/(L-y_{t2}\tan\delta_p)] + C_{ST2}TQ_{ST2}+\delta_{TOE2} \quad (2.45)$$

where δ_p is given by (2.38).

Tire force and moment equations will now be considered. It is assumed that the tires are capable of generating side forces, circumferential forces (along the intersection of the ground plane with the wheel plane), and aligning torques (moments about the z-axis) in addition to the normal forces (perpendicular to the ground plane). The tire forces are calculated using a simplified CALSPAN tire model in which the tire forces and moments depend only on the normal force, slip angle, and drive/brake torque [2.1].

The slip angle at each tire is computed by the following equations.

$$\alpha_i = \beta_i - \delta_i \qquad i=1,2 \quad (2.46)$$

$$\alpha_i = \beta_i \qquad i=3,4 \quad (2.47)$$

where

$$\beta_i = \tan^{-1}[(v+x_{ti}r)/(u-y_{ti}r)] \quad (2.48)$$

The tire circumferential and side forces and aligning torques are resolved into the vehicle axis system by the following equations, where FC_i, FS_i, and TQ_{ALi} refer to circumferential force, side force, and aligning torque,

$$F_{xi} = -FS_i \sin\delta_i + FC_i \cos\delta_i \qquad i=1,2 \qquad (2.49)$$

$$F_{yi} = FS_i \cos\delta_i + FC_i \sin\delta_i \qquad i=1,2 \qquad (2.50)$$

$$F_{xi} = FC_i \qquad i=3,4 \qquad (2.51)$$

$$F_{yi} = FS_i \qquad i=3,4 \qquad (2.52)$$

$$M_{zi} = TQ_{ALi} \qquad i=1,2,3,4 \qquad (2.53)$$

The steering torques TQ_{STi} are computed by the following equation, where C_x is the caster offset, as shown in Fig. 2.7.

$$TQ_{STi} = TQ_{ALi} - C_x FS_i \qquad i=1,2 \qquad (2.54)$$

Tire modeling will be addressed now. The force and moment generating capabilities of the tires in the vehicle model in this report are represented by a tire model developed by investigators at the CALSPAN Corporation. The model has evolved over the last 15 years; in its current form it supplies a fairly complete representation of the non-linear steady-state behavior of the pneumatic automobile tire [2.3, 2.4]. Such a representation is felt to be necessary even with simple vehicle models, if realistic results are to be obtained. It includes the following features:

(1) non-linear cornering stiffness/normal force relation

(2) non-linear camber stiffness/normal force relation

(3) non-linear side force/slip angle relation including side force saturation

(4) non-linear circumferential force/normal force relation

(5) side force roll-off as a function of longitudinal slip

(6) aligning moments and overturning moments as non-linear functions of normal load, side force, and camber angle.

The CALSPAN tire model is based on a representation of experimentally measured data curves with polynomial expressions. A total of 29 descriptive parameters plus a side force roll-off versus longitudinal slip table are required. The independent variables for the model are slip angle, camber angle, radial deflection, and longitudinal slip. The dependent variables are radial force, side force, circumferential force, aligning moment, and overturning moment

A simplified version of the CALSPAN tire model for use with the 3-DOF vehicle model will now be described. The simplifications arise since no tire deflection is allowed and wheel-spin dynamics are not included. As a result, tire deflection and longitudinal slip cannot be independent variables. Instead, the normal force and drive/brake torque are taken as input variables. Also, the camber angle is always zero, so the camber angle dependence is removed. In addition, the overturning moment was felt to be unimportant for this vehicle model, so it was taken to be zero. In summary, for this simplified model the independent variables are slip angle, normal force, and drive/brake torque; the dependent variables are side force, circumferential force, and aligning moment. Longitudinal slip is calculated as an intermediate variable so that the standard CALSPAN side force roll-off calculation, which depends on longitudinal slip, can be used.

The simplified CALSPAN tire model can be broken down into the following steps. Details of the calculations are given later.

(1) The circumferential force necessary to give the existing drive/brake torque is calculated.

(2) The corresponding longitudinal slip is computed, which requires the calculation of several frictional properties of the tire and road surface.

(3) The circumferential force computed in (1) is compared with the maximum tire capabilities and

reduced if necessary.

(4) Rolling resistance effects are added.

(5) The side force is calculated for the existing slip angle and normal force as if the tire were free-rolling (no longitudinal slip).

(6) The side force computed in (5) is modified if the longitudinal slip computed in (2) is non-zero.

(7) The aligning moment is computed based on the normal force and side force.

These steps are carried out by the following sequence of calculations.

The circumferential force for the i^{th} tire is first assumed to have the following form

$$FC_{0i} = TQ_{ti}/RT_i \tag{2.55}$$

The coefficient of sliding friction in braking (at longitudinal slip = 1.0) as a function of normal load is represented by

$$\mu_{Si} = S_{0i} + S_{1i}FN_i + S_{2i}FN_i^2 \tag{2.56}$$

The peak coefficient of friction for zero slip angle as a function of normal load is represented by

$$\mu_{Pi} = P_{0i} + P_{1i}FN_i + P_{2i}FN_i^2 \tag{2.57}$$

The longitudinal slip at which μ_{Pi} occurs is given by

$$SI_i = -R_{0i} - R_{1i}FN_i \qquad (2.58)$$

Note that in the CALSPAN data R_{0i} AND R_{1i} have negative values so that (2.58) gives positive value for SI_i.

An effective coefficient of sliding friction (longitudinal slip = 1) for the existing slip angle and road surface skid number is given by

$$\mu_{1i} = \mu_{Si}\cos(\alpha_i)SN_i \qquad (2.59)$$

An effective peak coefficient of friction for the existing slip angle and road surface skid number is given by

$$\mu_{Mi} = \mu_{Pi}(1-57.3B_{Ci}\alpha_i)SN_i \qquad (2.60a)$$

provided this expression is greater than μ_{1i}. Otherwise,

$$\mu_{Mi} = \mu_{1i} \qquad (2.60b)$$

The longitudinal slip corresponding to this level of FN_i and FC_{0i} is given by

$$S_i = 1.0, \quad \text{for } FC_{0i} < -\mu_{Mi}FN_i \qquad (2.61a)$$

$$S_i = -(FC_{Oi}/FN_i)(SI_i/\mu_{Mi}), \text{ for } FC_{Oi} \leq \mu_{Mi}FN_i \quad (2.61b)$$

$$S_i = -1.0, \text{ for } FC_{Oi} > \mu_{Mi}FN_i \quad (2.61c)$$

Note that if $|FC_{Oi}| \leq \mu_{Mi}FN_i$, a value of slip S_i will be obtained such that $-SI_i \leq S_i \leq SI_i$.

The circumferential force generated by the tire is then given by

$$FC_{Ai} = FC_{Oi}, \text{ for } FC_{Oi} \leq \mu_{Mi}FN_i \quad (2.62a)$$

$$= \mu_{1i}FN_i, \text{ for } FC_{Oi} > \mu_{Mi}FN_i \quad (2.62b)$$

Finally, an additional circumferential force, proportional to the normal load, is added opposing the direction of motion to simulate the effects of rolling resistance [2.5]. Thus,

$$FC_i = FC_{Ai} - K_{RRi}FN_i \quad (2.63)$$

where K_{RRi} is the rolling resistance proportionality factor.

Some remarks should be made concerning the possibility of tire spin due to drive torque ($S_i = -1$), even though maneuvers involving tire spin were not included in this investigation. The limit on circumferential force in this tire model leads to a corresponding limit on the drive torque which can be utilized, given by

$$TL_i = FC_{Ai}RT_i \tag{2.64}$$

This utilized drive torque would be less than TQ_{ti} in the case of $S_i = -1$. For a two-wheel-drive vehicle with a standard differential in the driven axle the drive torque applied to the wheels on that axle is the same and is limited by the torque which can be utilized by either tire. To accurately simulate post-tire-spin behavior of such a vehicle, TL_i should be taken as the input drive torque to the non-spinning driven wheel, rather than TQ_{ti}. This provision is included in the 3-DOF model.

The side force generatd by the tire is computed by first calculating the force which would be generated by a free-rolling tire operating at the same slip angle, then modidying this to account for the longitudinal slip. The side force for a free-rolling tire is calculated by

$$FS_{0i} = -G_{\alpha i}\mu_{yi}FN_i \tag{2.65}$$

where μ_{yi} is the peak lateral friction coefficient for the existing normal load, given by

$$\mu_{yi} = (B_{1i}FN_i+B_{3i}+B_{4i}FN_i^2)SN_i \tag{2.66}$$

$G_{\alpha i}$ is a side force shaping function, given by

$$G_{\alpha i} = 1.0 \text{ for } \bar{\alpha}_i > 3.0 \tag{2.67}$$

$$= \bar{\alpha}_i - 1/3\bar{\alpha}_i|\bar{\alpha}_i| + (1/27)\bar{\alpha}_i^3, \text{ for } |\bar{\alpha}_i| \leq 3.0 \tag{2.68}$$

$$= -1.0, \text{ for } \bar{\alpha}_i < -3.0 \tag{2.69}$$

where $\bar{\alpha}_i$ is a non-dimensional slip angle defined by

$$\bar{\alpha}_i = -\alpha_i C_{\alpha i}/(\mu_{yi} FN_i) \tag{2.70}$$

and $C_{\alpha i}$ is the low-slip angle cornering stiffness, represented by

$$C_{\alpha i} = A_{0i} + A_{1i}FN_i - (A_{1i}/A_{2i})FN_i^2,$$

$$\text{for } FN_i \leq A_{2i} \tag{2.71}$$

$$= A_{0i}, \text{ for } FN_i > A_{2i} \tag{2.72}$$

It may be noted that at low slip angles this free-rolling tire model behaves like a linear tire, reducing to $FS_{0i} = C_{\alpha i}\alpha_i$; at extremely high slip angles it saturates and behaves like a sliding tire, reducing to $FS_{0i} = \mu_{yi}FN_i$.

The effects of longitudinal slip are accounted for by assuming that the side force of a side-slipping longitudinally-slipping tire can be broken down into two components: a "rolling" side force and a "sliding" side force. A side force roll-off factor f_i is defined where $f_i = 0$ corresponds to a free-rolling tire ($S_i = 0$), and $f_i = 1$ corresponds to a sliding tire ($S_i = 1.0$). f_i is given by linear interpolation on S_i in a lookup table.

The final value of the side force is then given by

$$FS_i = FS_{0i}(1-f_i) + FN_i \mu_{Si} \left|\sin\alpha_i\right| f_i \,\mathrm{sgn}(FS_{0i}) SN_i \qquad (2.73)$$

where the first term is the "rolling" component and the second term is the "sliding" component.

The aligning torque is assumed to be a function of both normal load and side force, and is given by

$$TQ_{ALi} = (K_{1i}FN_i + K_{2i}\left|FS_i\right|)FS_i \qquad (2.74)$$

Aerodynamic effects are treated in the standard fashion. They are represented by the following equations, where all forces are taken to act at the center of mass of the vehicle. The longitudinal drag, which affects the drive thrust requirement, is given by (2.75) where C_D is the drag coefficient, A_{PF} is the projected frontal area, ρ_A is the density of air, and u is the forward velocity. This assumes the vehicle is moving through still air with constant velocity.

$$F_{xaero} = C_D A_{PF} \rho_A u \left|u\right|/2 \qquad (2.75)$$

$$F_{yaero} = F_{zaero} = M_{xaero} = M_{yaero} = M_{zaero} = 0 \quad (2.76)$$

The terms which are taken to be zero in (2.76) will be retained in the model for completeness.

The drive-brake torque at tire i is given by

$$TQ_{ti} = T_{QD} R_A \lambda_{TQi} - P_{FL} B_{RKi} \quad (2.77)$$

where T_{QD} is drive-line torque, R_A is the drive axle ratio, λ_{TQi} are torque distribution parameters, P_{FL} is brake-line pressure, and B_{RKi} are brake torque coefficients.

Chassis-drive axle torque transfer is given by

$$T_i = T_{QD} \lambda_{DTi}, \quad i = 1,2 \quad (2.78)$$

Differential equations for inertial coordinates X, Y, and heading angle ψ are given by

$$\dot{X} = u \cos\psi - v \sin\psi \quad (2.79)$$

$$\dot{Y} = u \sin\psi + v \cos\psi \quad (2.80)$$

$$\dot{\psi} = r \quad (2.81)$$

2.2 Solution Procedure for the Equations for the Three-Degree-of-Freedom Model.

Equations (2.15), (2.16), (2.17), (2.19), (2.20), (2.21) and (2.37) describe a set of coupled non-linear first order differential equations in the variables u, v, and r. To integrate these equations, it is convenient to put them into the form

$$\dot{u} = \dot{u}(u,v,r)$$

$$\dot{v} = \dot{v}(u,v,r)$$

$$\dot{r} = \dot{r}(u,v,r)$$

This cannot be done immediately due to the implicit nature of the tire force relations, the presence of compliance in the steering system, and the fact that the normal force-side force relation is not one-to-one. The desired form is obtained by the following method.

Equations (2.23) through (2.28) are substituted into (2.15), (2.16), (2.17), (2.19) and (2.21). In matrix form, they, plus (2.37), (2.44), and (2.45) can then be written

$$[C]\left\{a\right\} = \left\{b\right\} \qquad (2.82)$$

where

$$a_1 = \dot{u}$$

$$a_2 = \dot{v}$$

$$a_3 = \dot{r}$$

$$a_4 = F_{z1}$$

$$a_5 = F_{z2}$$

$$a_6 = F_{z3}$$

$$a_7 = F_{z4}$$

$$a_8 = \delta_p \tag{2.83}$$

and

$c_{ij} = 0$, except:

$$c_{11} = c_{22} = m_1 + m_2$$

$$c_{13} = c_{61} = -m_2 y_p$$

$$c_{23} = c_{62} = m_2 x_p$$

$$c_{34} = c_{35} = c_{36} = c_{37} = -1$$

$$c_{42} = -m_2 z_p$$

$$C_{43} = I^S{}_{xz} - m_2 x_p z_p$$

$$C_{44} = -y_{t1} \ , \ C_{45} = -y_{t2} \ , \ C_{46} = -y_{t3} \ , \ C_{47} = -y_{t4}$$

$$C_{51} = m_2 z_p$$

$$C_{53} = I^S{}_{yz} - m_2 y_p z_p$$

$$C_{54} = x_{t1} \ , \ C_{55} = x_{t2} \ , \ C_{56} = x_{t3} \ , \ C_{57} = x_{t4}$$

$$C_{63} = I^S{}_{zz} + m_2 (x_p{}^2 + y_p{}^2)$$

$$C_{74} = y_{t1}(1-\lambda_{RS})$$

$$C_{75} = y_{t2}(1-\lambda_{RS})$$

$$C_{76} = -y_{t3}\lambda_{RS}$$

$$C_{77} = -y_{t4}\lambda_{RS}$$

$$C_{88} = 1 \qquad (2.84)$$

$$b_1 = (m_1+m_2)vr + m_2 r^2 x_p + \sum_{i=1}^{4} F_{xi} + F_{xaero}$$

$$b_2 = (m_1+m_2)ur + m_2 r^2 y_p + \sum_{i=1}^{4} F_{yi} + F_{yaero}$$

$$b_3 = (m_1+m_2)g+F_{zaero}$$

$$b_4 = I^S_{yz}r^2+m_2(urz_p-r^2y_pz_p+y_pg)-\sum_{i=1}^{4} z_{ti}F_{yi}+M_{xaero}$$

$$b_5 = -I^S_{xz}r^2+m_2(vrz_p+r^2x_pz_p-x_pg)+\sum_{i=1}^{4} z_{ti}F_{xi}+M_{yaero}$$

$$b_6 = m_2(urx_p-vry_p)+\sum_{i=1}^{4}[(x_{ti}F_{yi}-y_{ti}F_{xi})+M_{zi}]+M_{zaero}$$

$$b_7 = (1-\lambda_{RS})T_1-\lambda_{RS}T_2$$

$$b_8 = \{\delta_{Sw} + C_{ST3}\ (TQ_{ST1} + TQ_{ST2})/NG\}\ /NG \qquad (2.85)$$

{b} is a function of u, v, r, F_{xi}, F_{yi} M_{zi}, and TQ_{STi} ($\vec{F}_{aero}$, $\vec{M}_{aero}$, are functions of u, v, r). F_{xi}, F_{yi}, M_{zi} and TQ_{STi} depend on F_{zi}, which are elements of {a}. It is thus necessary to solve (2.82) simultaneously with the following non-linear equation

$$\{b\} = \{b\}(u,v,r,\{a\}) \qquad (2.86)$$

The solution to (2.82) and (2.86) is obtained iteratively. At a given time, u, v, and r are known. An estimate of F_{zi} is made from a previous time step. (The first estimate is made using the static weight distribution.) An initial $\{b_0\}$ can be computed, and the following algorithm applied.

$$\{a_j\} = [C]^{-1}\{b_j\} \tag{2.87}$$

$$\{b_{j+1}\} = \{b\}(u,v,r,\{a_j\}) \tag{2.88}$$

This process is continued until

$$|a_{i,j}-a_{i,j-1}| < \varepsilon_i \qquad i=1,\ldots 8 \tag{2.89}$$

where ε_i are assigned convergence parameters whose magnitudes are related to the expected magnitudes of a_i.

2.3 Fortran Computer Program for the Three-Degree-of-Freedom Model

The differential equations presented in 2.1 were integrated by a fourth-order predictor-corrector method. A pre-programmed code, the HPCG subroutine in the IBM SSP package [2.6] was used to implement the scheme. The source code for HPCG is not given in the following program listing since it is widely available.

The 3-DOF model, as programmed, requires a set of "driver" subroutines, named DRINPT, DRINIT, DRIOUT, and DRIVER. This was done to interface this model with the driver module described in Reference [2.7] and Chapter 4. The call statements for these subroutines may be removed if alternate provisions are made for supplying values of

DELSW, TQD, and PFL in subroutine FCT.

A Fortran listing of the program is given on the following pages, followed by a list of program variables and a typical data set.

```
C   MAIN SUBROUTINE TRANS35
C   MAIN PROGRAM FOR THE 3DOF MODEL.  MAIN READS VEHICLE AND TIRE
C    DATA AND CONTROLS THE ITERATIVE SOLUTION OF THE STEADY-STATE
C EQUATIONS
C   THIS VERSION OF MAIN REQUIRES SUBROUTINES F35,TIRE3, MINV,
C   HPCG, FCT, OUT1. (OUT2 INCLUDED FOR EXTENDED OUTPUT)
C   DEVELOPED BY DOUGLAS L. WILSON, 8/30/81
      LOGICAL*1 VEHCON(6),TIRCON(6),ICSET(6)
      REAL K1,K2, KD
      REAL*8 T,DT,T1,DTPRNT
      DIMENSION C(7,7),PRMT(5),Y(6),DERY(6),AUX(16,6),JUNK1(7),JUNK2(7)
      EXTERNAL FCT, OUT1
      COMMON /T3DATA/ FRO(4,10,2),  A0(4), A1(4), A2(4), B1(4), B3(4),
     1 B4(4), RT(4),  P0(4), P1(4), P2(4), S0(4), S1(4), S2(4),
     2 R0(4), R1(4), K1(4), K2(4), BC(4), SN(4), FRR(4)
      COMMON /V3D/ CI(7,7),ALAMT(4),EC(5),XT(4),YT(4),DTDE(2),TAXL(2),
     1 AXLR,VC,G,ALAMRS,VM,VIZZ,VIYZ,VIZX,XPL,YPL,ZPL,PLM,PLIZZ,PLIYZ,
     2 PLIZX,CST1,CST2,CST3,SR,XC(2),BRK(4),HF,HR,CD,PFA,RHOA,IACKER
      COMMON /FOUT/ FN(4),ALPHAT(4),TQT(4),FS(4),FC(4),S(4),DELT(2),
     1 TQST(2)
      COMMON /OUTPT/ DSWOUT,TQDOUT,PFLOUT
      COMMON /PRNT/ DTPRNT,T1
      COMMON /VPR/ DSWMAX,TQDMAX,PFLMAX,KD,DSWO,TQDO,PFLO
      COMMON /FINFO/ IFIRST
C
C   READ VEHICLE AND TIRE DATA
      READ (5,105) VEHCON
      READ (5,101) (XT(I), YT(I), I=1,4)
      READ (5,101) DTDE(1), DTDE(2)
      READ (5,101) ALAMT(1), ALAMT(2)
      READ (5,101) ALAMT(3), ALAMT(4)
      READ (5,101) TAXL(1), TAXL(2)
      READ (5,100) VC, VIZZ, VIYZ, VIZX, VM, AXLR, ALAMRS, G
      READ (5,100) XPL, YPL, ZPL, PLM, PLIZZ, PLIYZ, PLIZX
      READ (5,100) CST1, CST2, CST3, SR, HF, HR
      READ (5,101) XC(1), XC(2)
      READ (5,101) BRK(1), BRK(2)
      READ (5,101) BRK(3), BRK(4)
      READ (5,100) CD, PFA, RHOA
      READ (5,99) IACKER
      READ (5,100) DSWMAX,TQDMAX,PFLMAX,KD

      READ (5,105) TIRCON
      READ (5,102) (RT(I), I=1,4)
      READ (5,102) (A0(I), I=1,4)
      READ (5,102) (A1(I), I=1,4)
      READ (5,102) (A2(I), I=1,4)
      READ (5,102) (B1(I), I=1,4)
      READ (5,102) (B3(I), I=1,4)
      READ (5,102) (B4(I), I=1,4)
      READ (5,102) (P0(I), I=1,4)
      READ (5,102) (P1(I), I=1,4)
      READ (5,102) (P2(I), I=1,4)
      READ (5,102) (S0(I), I=1,4)
      READ (5,102) (S1(I), I=1,4)
```

```
      READ (5,102) (S2(I), I=1,4)
      READ (5,102) (R0(I), I=1,4)
      READ (5,102) (R1(I), I=1,4)
      READ (5,102) (K1(I),I=1,4)
      READ (5,102) (K2(I),I=1,4)
      READ (5,102) (BC(I), I=1,4)
      READ (5,102) (SN(I), I=1,4)
      READ (5,102) (FRR(I),I=1,4)
      READ (5,104) ((FRD(I,J,1),FRD(I,J,2), I=1,4),J=1,10)
C
      READ (4,105) ICSET
      READ (4,100) X0,Y0,PSI0,U0,V0,PSID0,DSW0,TQD0,PFL0,DT,DTPRNT
      READ (4,100) TMAX
      READ (4,103) (EC(I), I=1,5)
C     WRITE (1,199) VEHCON, TIRCON, ICSET
  199 FORMAT ('1VEHICLE CONFIGURATION:',6A1/' TIRE CONFIGURATION:',6A1/
     *' INITIAL CONDITIONS SET:',6A1/
     *' T',11X,'DELSW',7X,'X',11X,'Y',11X,'PSI',9X,'V',11X,
     *'R',11X,'AY')
   99 FORMAT (I6)
  100 FORMAT (F12.5)
  101 FORMAT (2F12.5)
  102 FORMAT (4E16.6)
  103 FORMAT (5F12.5)
  104 FORMAT (8F12.5)
  105 FORMAT (6A1)
C
C SET UP THE COEFFICIENT MATRIX
      DO 10 I=1,7
      DO 10 J=1,7
   10 C(I,J) = 0.0
C
      C(1,1) = VM + PLM
      C(1,3) = -YPL*PLM
      C(2,2) = C(1,1)
      C(2,3) = XPL*PLM
C
      C(3,4) = 1.0
      C(3,5) = 1.0
      C(3,6) = 1.0
      C(3,7) = 1.0
C
      C(4,2) = -ZPL*PLM
      C(4,3) = -PLM*XPL*ZPL + VIZX +
      C(4,4) = -YT(1)
      C(4,5) = -YT(2)
      C(4,6) = -YT(3)
      C(4,7) = -YT(4)
C
      C(5,1) = ZPL*PLM
      C(5,3) = VIYZ + PLIYZ - YPL*ZPL*PLM
      C(5,4) = XT(1)
      C(5,5) = XT(2)
      C(5,6) = XT(3)
      C(5,7) = XT(4)
```

```
C
      C(6,1) = C(1,3)
      C(6,2) = C(2,3)
      C(6,3) = (PLM*(XPL**2 + YPL**2) + PLIZZ + VIZZ)
C
      C(7,4) = YT(1)*(1.0 - ALAMRS)
      C(7,5) = YT(2)*(1.0 - ALAMRS)
      C(7,6) = -YT(3)*ALAMRS
      C(7,7) = -YT(4)*ALAMRS
C
C CALCULATE THE INVERSE
      DO 20 I=1,7
      DO 20 J=1,7
   20 CI(I,J) = C(I,J)
C
      CALL MINV(CI,7,D,JUNK1,JUNK2)
      IF (ABS(D) .LT. 0.001) GO TO 70
      T1 = -DTPRNT
      IFIRST = 1
      PRMT(1) = 0.0
      PRMT(2) = TMAX
      PRMT(3) = DT
      PRMT(4) = 0.1
C
      Y(1) = X0
      Y(2) = Y0
      Y(3) = PSI0
      Y(4) = U0
      Y(5) = V0
      Y(6) = PSID0
      DERY(1) = 0.16666
      DERY(2) = 0.16665
      DERY(3) = 0.16667
      DERY(4) = 0.16667
      DERY(5) = 0.16667
      DERY(6) = 0.16667
C
      CALL DRINPT
      CALL DRINIT
      CALL DRIOUT(1)
      CALL HPCG (PRMT, Y, DERY, 6, IHLF, FCT, OUT1, AUX)
C
      IF (IHLF .LT. 11) GO TO 30
C   ERROR RETURN FROM HPCG
      WRITE (6,151) IHLF
  151 FORMAT ('0ERROR RETURN FROM HPCG; IHLF =',I3)
      STOP
   30 IF (PRMT(5) .NE. 0.0) GO TO 40
C
C   TMAX HAS BEEN EXCEEDED
      WRITE (6,152)
  152 FORMAT ('0TMAX HAS BEEN EXCEEDED')
      STOP
C
C   VEHICLE HAS STOPPED
```

```
        40 WRITE (6,153)
       153 FORMAT('OVEHICLE HAS STOPPED')
           STOP
     C
     C COEFFICIENT MATRIX SINGULAR
        70 WRITE (6,210) D
       210 FORMAT ('OCOEFFICIENT MATRIX SINGULAR; D =',F20.10)
           STOP
           END
 *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
 *OPTIONS IN EFFECT*  NAME = MAIN    , LINECNT =       57
 *STATISTICS*    SOURCE STATEMENTS =      133,PROGRAM SIZE =      5078
 *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
C
C     SUBROUTINE OUT1 CHECKS THE VEHICLE STOPPING CRITERION
C     AND WRITES OUT INTERMEDIATE VALUES.
C
          SUBROUTINE OUT1 (T,Y,DY,IHLF,NEQ,PRMT)
          REAL*8 DTPRNT,T1
          DIMENSION PRMT(5), Y(6), DY(6)
          COMMON /OUTPT/ DSWOUT,TODOUT,PFLOUT
          COMMON /PRNT/ DTPRNT,T1
C
          IF (((Y(4)**2 + Y(5)**2) .GT. 0.1) .OR. (TOD .NE. 0.0)) GO TO 10
          PRMT(5) = 1.0
          RETURN
C
C     PRINT OUT INTERMEDIATE VALUES AT INTERVALS DTPRNT
       10 IF (T-T1+0.0001 .LT. DTPRNT) GO TO 20
          T1 = T
          VEL = SQRT(Y(3)**2 + Y(4)**2)
          AY = DY(5) + Y(4)*Y(6)
C         WRITE(1,201) T,DSWOUT,Y(1),Y(2),Y(3),VEL,TODOUT,Y(4),Y(5),Y(6),
C        1 AY,PFLOUT,DY(4),DY(5),DY(6)
      201 FORMAT('OT =',F12.4,5X,'DELSW=',F12.4,5X,'X     =',F12.4,5X,
         *'Y     =',F12.4,5X,'PSI  =',F12.4,5X,'VEL  =',F12.4/
         *21X,'TOD  =',F12.4,5X,'U     =',F12.4,5X,'V     =',F12.4,5X,
         *'R     =',F12.4,5X,'AY    =',F12.4/
         *21X,'PFL  =',F12.4,5X,'UDOT =',F12.4,5X,'VDOT =',F12.4,5X,
         *'RDOT =',F12.4)
          WRITE (2,202) T,DSWOUT,Y(1),Y(2),Y(3),Y(5),Y(6),AY,Y(4)
      202 FORMAT(9F12.5)
       20 RETURN
          END
*OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT*  NAME = OUT1    , LINECNT =        57
*STATISTICS*    SOURCE STATEMENTS =       17,PROGRAM SIZE =       936
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
           C
           C  SUBROUTINE OUT2 PRINTS OUT TIRE FORCES AND SLIPS
           C  OUT2 IS NOT CALLED IN THIS VERSION OF MAIN PROGRAM*
           C
0001              SUBROUTINE OUT2
0002              COMMON /FOUT/ FN(4),ALPHAT(4),TQT(4),FS(4),FC(4),S(4),DELT(2),
                 1 TQST(2)
           C  20 WRITE (1,205)
           C      WRITE (1,206) (I,FN(I),I,FS(I),I,FC(I),I=1,4)
           C      WRITE (1,207)
           C      WRITE (1,208) (I,S(I),I,ALPHAT(I),I=1,4)
           C      WRITE (1,209) DELT(1),TQST(1),DELT(2),TQST(2)
0003          205 FORMAT ('ONORMAL FORCES AT TIRES',8X,'SIDE FORCES AT TIRES',10X,
                 *'CIRCUMFERENTIAL FORCES AT TIRES')
0004          206 FORMAT (' FN(',I1,') =',F12.5,11X,'FS(',I1,') =',F12.5,11X,
                 *'FC(',I1,') =',F12.5)
0005          207 FORMAT ('OSLIPS AT TIRES',16X,'SLIP ANGLES AT TIRES')
0006          208 FORMAT (' S(',I1,') =',F12.5,12X,'ALPHAT(',I1,') =',F12.5)
0007          209 FORMAT('OFRONT TIRE STEER ANGLES:  DELT(1) =',F12.5,8X,
                 * 'KINGPIN STEERING TORQUES:  TQST(1) =',F12.5/27X,
                 * 'DELT(2) =',F12.5,35X,'TQST(2) =',F12.5)
0008              RETURN
0009              END
 *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
 *OPTIONS IN EFFECT*  NAME = OUT2     , LINECNT =       57
 *STATISTICS*     SOURCE STATEMENTS =         9,PROGRAM SIZE =      590
 *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
      C
      C
      C   SUBROUTINE FCT SERVES AS AN INTERFACE BETWEEN HPCG AND F35
      C
            SUBROUTINE FCT(T,Y,DY)
            DIMENSION Y(6), DY(6)
            COMMON /FINFO/ IFIRST
            COMMON /FOUT/ FN(4), ALPHAT(4),TQT(4),FS(4),FC(4),S(4),DELT(2),
           1 TQST(2)
            COMMON /OUTPT/ DSWOUT,TQDOUT,PFLOUT
      C
            DEL1 = DELT(1)
            DEL2 = DELT(2)
            CALL DRIVER(DELSW,PFL,TQD,D3,D4,DY,T,Y,DEL1,DEL2)
            CALL F35(DELSW,TQD,PFL,Y(4),Y(5),Y(6),DY(4),DY(5),DY(6),IFIRST,T)
            DY(1) = Y(4)*COS(Y(3)) - Y(5)*SIN(Y(3))
            DY(2) = Y(4)*SIN(Y(3)) + Y(5)*COS(Y(3))
            DY(3) = Y(6)
            DSWOUT = DELSW
            TQDOUT = TQD
            PFLOUT = PFL
            RETURN
            END
 *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
 *OPTIONS IN EFFECT*  NAME = FCT     , LINECNT =       57
 *STATISTICS*    SOURCE STATEMENTS =       17,PROGRAM SIZE =      766
 *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
C    SUBROUTINE F35 COMPUTES THE TIME DERIVATIVES FOR A SIMPLE 3-DOF
C    VEHICLE MODEL.  THE ONLY FORCES ACTING ON THE VEHICLE ARE TIRE
C    FORCES, GRAVITY, AND AIR RESISTANCE
C    F35 REQUIRES SUBROUTINE TIRE3
C    DEVELOPED BY DOUGLAS L. WILSON 8/30/81
        SUBROUTINE F35(DELSW,TQD,PFL,U,V,R,UDOT,VDOT,RDOT,IFIRST,T)
        DIMENSION B(7), BO(7), FX(4), FY(4), FZ(4), THETAT(4),
      *TOAL(4), UT(4), VT(4),TL(4)
        COMMON /V3D/ CI(7,7),ALAMT(4),FC(5),XT(4),YT(4),DTOE(2),TAXL(2),
      1 AXLR,VC,G,ALAMRS,VM,VIZZ,VIYZ,VIZX,XPL,YPL,ZPL,PLM,PLIZZ,PLIYZ,
      2 PLIZX,CST1,CST2,CST3,SR,XC(2),BRK(4),HF,HR,CD,PFA,RHOA,IACKER
        COMMON /FOUT/ FN(4),ALPHAT(4),TQT(4),FS(4),FC(4),S(4),DELT(2),
      1 TQST(2)
C
C    INITIALIZE FZ AND TQST ON FIRST CALL OF F35
        IF (IFIRST .NE. 1) GO TO 10
        FZ(1) = (VM*XT(3) + PLM*(XT(3)-XPL))/(XT(1)-XT(3))*G/2.0
        FZ(2) = FZ(1)
        FZ(3) = -(VM*XT(1) + PLM*(XT(1)-XPL))/(XT(1)-XT(3))*G/2.0
        FZ(4) = FZ(3)
        TQST(1) = 0.0
        TQST(2) = 0.0
        IFIRST = 0
C
C    COMPUTE THE PART OF B WHICH IS INDEPENDENT OF TIRE FORCES
     10 BO(1) = (VM+PLM)*V*R + PLM*XPL*R**2 - CD*PFA*RHOA*0.5*U*ABS(U)
        BO(2) = -(VM + PLM)*U*R + PLM*YPL*R**2
        BO(3) = -(VM + PLM)*G
        BO(4) =  (VIYZ + PLIYZ)*R**2 + PLM*(U*R*ZPL - YPL*ZPL*R**2)
      1         + PLM*YPL*G
        BO(5) = -(VIZX + PLIZX)*R**2 + PLM*(V*R*ZPL + XPL*ZPL*R**2)
      1         -PLM*XPL*G
        BO(6) = PLM*R*(XPL*U - YPL*V)
        BO(7) = (1.0 - ALAMRS)*TAXL(1)*TQD - ALAMRS*TAXL(2)*TQD
C
C    COMPUTE TIRE PATCH VELOCITIES
        PIO2 = 1.57079633
        DO 20 I=1,4
          UT(I) = U - YT(I)*R
          VT(I) = V + XT(I)*R
          IF (ABS(UT(I)) .LT. 0.001) GO TO 15
          THETAT(I) = ATAN(VT(I)/UT(I))
          GO TO 20
     15   THETAT(I) = PIO2
          IF (VT(I) .LT. 0.0) THETAT(I) = -PIO2
         IF (ABS(VT(I)) .LT. 0.001) THETAT(I) = 0.0
     20 CONTINUE
C
C    COMPUTE DRIVE TORQUE DISTRIBUTION
        DO 30 I=1,4
        TQT(I) = TQD*ALAMT(I)*AXLR
     30 TQT(I) = TQT(I) - BRK(I)*PFL
C
C    ITERATE TO FIND SOLUTION TO AX=B
C
```

```
          K = 0
       40 K=K+1
C
C   SAVE THE OLD VALUES OF UDOT, VDOT, RDOT, FZ(I), DELT(I)
C
          UDOTO = UDOT
          VDOTO = VDOT
          RDOTO = RDOT
          FZ10 = FZ(1)
          FZ20 = FZ(2)
          FZ30 = FZ(3)
          FZ40 = FZ(4)
          DELT10 = DELT(1)
          DELT20 = DELT(2)
C
C   COMPUTE TIRE FORCES
          DO 50 I=1,4
       50 FN(I) = -FZ(I)
C
C   CHECK FOR TIRE LIFTOFF
          DO 60 I=1,4
          IF (FN(I) .GE. 0.001) GO TO 60
          WRITE (6,202) (J,FN(J),J=1,4)
          STOP
       60 CONTINUE
C
C   COMPUTE STEERING ANGLES
          DELTP = DELSW/SR + CST3*(TQST(1) + TQST(2))/SR**2
C   CHECK FOR STEERING TYPE
          IF (IACKER .EQ. 1) GO TO 70
C   COMPUTE WAGON-TYPE STEERING ANGLES
          DELT(1) = DELTP + CST1*TQST(1)
          DELT(2) = DELTP + CST2*TQST(2)
          GO TO 75
C
C   COMPUTE ACKERMAN STEER ANGLES
       70 XL = XT(1) - XT(3)
          TP = TAN(DELTP)
          DELT(1) = DTOE(1)+ATAN(XL*TP/(XL-TP*YT(1)))+CST1*TQST(1)
          DELT(2) = DTOE(2)+ATAN(XL*TP/(XL-TP*YT(2)))+CST2*TQST(2)
C
C   COMPUTE TIRE SLIP ANGLES
       75 DO 80 I=1,2
            ALPHAT(I) = THETAT(I) - DELT(I)
       80   ALPHAT(I+2) = THETAT(I+2)
C
C   COMPUTE TIRE FORCES
C
          DO 85 I=1,4
          CALL TIRE3(FN,ALPHAT,TQT,FS,FC,TQAL,S,TL,I)
C   CHECK FOR TIRE SPIN
          IF (S(I) .GT. -0.999) GO TO 85
          IF (I .EQ. 1) IOTHER = 2
          IF (I .EQ. 2) IOTHER = 1
          IF (I .EQ. 3) IOTHER = 4
```

```
0070            IF (I .EQ. 4) IOTHER = 3
0071            TQT(IOTHER) = TL(I)
0072            IF (IOTHER .EQ. 1 .OR. IOTHER .EQ. 3)
               * CALL TIRE3(FN,ALPHAT,TAT,FS,FC,TQAL,S,TL,IOTHER)
0073         85 CONTINUE
       C
       C   RESOLVE TIRE FORCES ALONG VEHICLE AXES
       C
0074            FX(1) = -FS(1)*SIN(DELT(1)) + FC(1)*COS(DELT(1))
0075            FY(1) =  FS(1)*COS(DELT(1)) + FC(1)*SIN(DELT(1))
0076            FX(2) = -FS(2)*SIN(DELT(2)) + FC(2)*COS(DELT(2))
0077            FY(2) =  FS(2)*COS(DELT(2)) + FC(2)*SIN(DELT(2))
0078            FX(3) = FC(3)
0079            FY(3) = FS(3)
0080            FX(4) = FC(4)
0081            FY(4) = FS(4)
0082            TQST(1) = TQAL(1) - XC(1)*FS(1)
0083            TQST(2) = TQAL(2) - XC(2)*FS(2)
       C
       C
       C   COMPUTE THE TIRE-FORCE DEPENDENT PART OF B
       C
0084            B(1) = BO(1) + FX(1) + FX(2) + FX(3) + FX(4)
0085            B(2) = BO(2) + FY(1) + FY(2) + FY(3) + FY(4)
0086            B(6) = BO(6) - FX(1)*YT(1)-FX(2)*YT(2)-FX(3)*YT(3)-FX(4)*YT(4)
              1              + FY(1)*XT(1)+FY(2)*XT(2)+FY(3)*XT(3)+FY(4)*XT(4)
              2             + TQAL(1)+TQAL(2)+TQAL(3)+TQAL(4)
0087            B(3) = BO(3)
0088            B(4) = BO(4) - VC*(FY(1)+FY(2)+FY(3)+FY(4))
0089            B(5) = BO(5) + VC*(FX(1)+FX(2)+FX(3)+FX(4))
0090            B(7) = BO(7)+(1.0-ALAMRS)*HF*(FY(1)+FY(2))-ALAMRS*HR*(FY(3)+FY(4))
       C
       C   COMPUTE NEW VALUES FOR UDOT,VDOT, RDOT, FZ(I)
       C
0091            UDOT = 0.0
0092            VDOT = 0.0
0093            RDOT = 0.0
0094            FZ(1) = 0.0
0095            FZ(2) = 0.0
0096            FZ(3) = 0.0
0097            FZ(4) = 0.0
0098            DO 90 J = 1,7
0099            UDOT = UDOT + CI(1,J)*B(J)
0100            VDOT = VDOT + CI(2,J)*B(J)
0101            RDOT = RDOT + CI(3,J)*B(J)
0102            FZ(1) = FZ(1) + CI(4,J)*B(J)
0103            FZ(2) = FZ(2) + CI(5,J)*B(J)
0104            FZ(3) = FZ(3) + CI(6,J)*B(J)
0105         90 FZ(4) = FZ(4) + CI(7,J)*B(J)
       C
       C   CHECK FOR CONVERGENCE
       C
0106            IF((UDOT-UDOTO)**2 .LE. EC(1) .AND. (VDOT-VDOTO)**2 .LE. EC(2)
              1 .AND. (RDOT-RDOTO)**2 .LE. EC(3) .AND. (FZ(1)-FZ1O)**2 .LE. EC(4)
              2 .AND. (FZ(2)-FZ2O)**2 .LE. EC(4) .AND. (FZ(3)-FZ3O)**2 .LE. EC(4)
```

```
                3 .AND. (FZ(4)-FZ40)**2 .LE. EC(4) .AND. (DELT(1)-DELT10)**2 .LT.
                4 EC(5) .AND. (DELT(2)-DELT20)**2 .LT. EC(5)) RETURN
          C
          C   CHECK WHETHER ITERATION LIMIT HAS BEEN EXCEEDED
          C
0107              IF (K .LT. 20) GO TO 40
0108              WRITE (6,201) T
0109          201 FORMAT ('ONO CONVERGENCE AFTER 20 ITERATIONS IN SUBROUTINE F32',
                * ' AT T=',F12.5)
0110          202 FORMAT ('ONEGATIVE NORMAL TIRE FORCE--TIRE LIFTOFF'/
                *        (' FN(',I1,') =',F12.5))
          C
0111              RETURN
0112              END
 *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
 *OPTIONS IN EFFECT*  NAME = F35     , LINECNT =       57
 *STATISTICS*    SOURCE STATEMENTS =       112,PROGRAM SIZE =     4088
 *STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
C     SUBROUTINE TIRE3 COMPUTES TIRE SIDE FORCE USING THE CALSPAN MODEL.
C     CIRCUMFERENTIAL FORCES ARE COMPUTED BY FC=TQT/RT.
C     THIS MODEL INCLUDES SIDE-FORCE FRICTION ROLL-OFF AS A FUNCTION
C     OF SLIP, WHICH IS COMPUTED FROM THE CIRCUMFERENTIAL FORCE.
C     DEVELOPED BY DOUGLAS L. WILSON, 8/30/81
      SUBROUTINE TIRE3(FN,ALPHAT,TQT,FS,FC,TQAL,S,TL,I)
      REAL K1,K2
      DIMENSION FN(4),ALPHAT(4),TQT(4),FS(4),FC(4),TQAL(4),S(4),TL(4)
      COMMON /T3DATA/ FRO(4,10,2),  A0(4), A1(4), A2(4), B1(4), B3(4),
     1 B4(4), RT(4),  P0(4), P1(4), P2(4), S0(4), S1(4), S2(4),
     2 R0(4), R1(4), K1(4), K2(4), BC(4), SN(4), FRR(4)
C
      FMAX = (B1(I)*FN(I) + B3(I) + B4(I)*FN(I)**2)*FN(I)*SN(I)
      CALPHA = A0(I) + A1(I)*FN(I) - A1(I)*FN(I)**2/A2(I)
      IF (FN(I) .GT. A2(I)) CALPHA = A0(I)
      ALFBAR = -CALPHA * ALPHAT(I) / FMAX
      DALF = ABS(ALFBAR)
      G = 1.0
      IF (ALFBAR .LT. 0.0) G = -G
      IF (DALF .LT. 3.0) G = ALFBAR - ALFBAR * DALF / 3.0
     1+ ALFBAR**3 / 27.0
      FS(I) = G * FMAX
      FC(I) = TQT(I)/RT(I)
      UP = P0(I) + P1(I)*FN(I) + P2(I)*FN(I)**2
      US = S0(I) + S1(I)*FN(I) + S2(I)*FN(I)**2
      SI = -R0(I) - R1(I)*FN(I)
      U1 = US*ABS(COS(ALPHAT(I)))/SN(I)
C   XM1 IS THE SLOPE AT LOW SLIP NUMBERS
      XM1 = UP*(1.0-57.3*BC(I)*ABS(ALPHAT(I)))*SN(I)/SI
      IF (XM1 .GE. U1/SI) GO TO 20
      XM1 = U1/SI
C   S IS THE SLIP NUMBER
   20 S(I) = -(FC(I)/FN(I))/XM1
      IF (S(I) .GT. SI) S(I) = 1.0
      IF (S(I) .LT. -SI) S(I) = -1.0
      IF (ABS(S(I)) .LE. 0.999) GO TO 30
      FC(I) = U1*FN(I)
      IF (S(I) .GT. 0.0) FC(I) = -FC(I)
      IF (S(I) .LE. -0.999) TL(I) = FC(I)*RT(I)
C    ADD ROLLING RESISTANCE FORCES
   30 FC(I) = FC(I) - FRR(I)*FN(I)
C    COMPUTE SIDE-FORCE FRICTION ROLL-OFF-
      DO 40 J=1,9
      IF (ABS(S(I)) .LE. FRO(I,J+1,1)) GO TO 50
   40 CONTINUE
      F = 1.0
      GO TO 60
   50 F = FRO(I,J,2) + (ABS(S(I)) - FRO(I,J,1))
     1  *(FRO(I,J+1,2) - FRO(I,J,2))/(FRO(I,J+1,1)-FRO(I,J,1))
   60 L = 1
      IF (FS(I) .LT. 0.0) L = -1
   70 FS(I) = FS(I)*(1.0-F) + FN(I)*US*ABS(SIN(ALPHAT(I)))
     1          *F*L*SN(I)
C
   80 TQAL(I) = (K1(I)*FN(I) + K2(I)*ABS(FS(I)))*FS(I)
```

```
0040                RETURN
0041                END
 *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
 *OPTIONS IN EFFECT*  NAME = TIRE3    , LINECNT =        57
 *STATISTICS*    SOURCE STATEMENTS =        41,PROGRAM SIZE =       2056
 *STATISTICS*  NO DIAGNOSTICS GENERATED

NO STATEMENTS FLAGGED IN THE ABOVE COMPILATIONS.
```

```
C NAASA  2.1.020 MINV      FTN   06-24-75      THE UNIV OF MICH COMP CTR
C
C     .............................................................
C
C        SUBROUTINE MINV
C
C        PURPOSE
C           INVERT A MATRIX
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C               RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C           THE MATRIX IS SINGULAR.
C
C     .............................................................
C
0001      SUBROUTINE MINV(A,N,D,L,M)
0002      DIMENSION A(1),L(1),M(1)
C
C        ..........................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C        STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,D,BIGA,HOLD
C
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C        ROUTINE.
C
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C        10 MUST BE CHANGED TO DABS.
C
C        ..........................................................
C
C        SEARCH FOR LARGEST ELEMENT
```

```
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVO
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
```

```
C         REDUCE MATRIX
C
          DO 65 I=1,N
          IK=NK+I
          HOLD=A(IK)
          IJ=I-N
          DO 65 J=1,N
          IJ=IJ+N
          IF(I-K) 60,65,60
       60 IF(J-K) 62,65,62
       62 KJ=IJ-I+K
          A(IJ)=HOLD*A(KJ)+A(IJ)
       65 CONTINUE
C
C         DIVIDE ROW BY PIVOT
C
          KJ=K-N
          DO 75 J=1,N
          KJ=KJ+N
          IF(J-K) 70,75,70
       70 A(KJ)=A(KJ)/BIGA
       75 CONTINUE
C
C         PRODUCT OF PIVOTS
C
          D=D*BIGA
C
C         REPLACE PIVOT BY RECIPROCAL
C
          A(KK)=1.0/BIGA
       80 CONTINUE
C
C         FINAL ROW AND COLUMN INTERCHANGE
C
          K=N
      100 K=(K-1)
          IF(K) 150,150,105
      105 I=L(K)
          IF(I-K) 120,120,108
      108 JQ=N*(K-1)
          JR=N*(I-1)
          DO 110 J=1,N
          JK=JQ+J
          HOLD=A(JK)
          JI=JR+J
          A(JK)=-A(JI)
      110 A(JI) =HOLD
      120 J=M(K)
          IF(J-K) 100,100,125
      125 KI=K-N
          DO 130 I=1,N
          KI=KI+N
          HOLD=A(KI)
          JI=KI-K+J
          A(KI)=-A(JI)
```

```
0087          130 A(JI) =HOLD
0088              GO TO 100
0089          150 RETURN
0090              END
 *OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
 *OPTIONS IN EFFECT*  NAME = MINV    , LINECNT =       57
 *STATISTICS*    SOURCE STATEMENTS =       90,PROGRAM SIZE =     2084
 *STATISTICS*  NO DIAGNOSTICS GENERATED

NO STATEMENTS FLAGGED IN THE ABOVE COMPILATIONS.
```

2.4 Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
ALAMRS	λ_{RS}	Fraction of total roll stiffness at front axle
ALAMT(I)	λ_{TQi}	Fraction of total drive torque applied at wheel I. Note that $\Sigma\lambda_{TQi} = 1$
ALFBAR	$\overline{\alpha}_i$	Normalized slip angle
ALPHAT(I) I=1,4	α_i	Slip angle at tire i
AUX		Auxiliary variable required by subroutine HPCG
AXLR	R_A	Axle drive ratio, same at front and rear if both driven
AY		Lateral acceleration
AO(I),A1(I), A2(I) I=1,4	A_{0i}, A_{1i}, A_{2i}	Coefficients in expression for low-slip cornering stiffness
B(I) I=1,7	$\{b\}$	Vector of force-type quantities in iterative solution to $\{a\}$
BC(I) I=1,4	B_{Ci}	Tire parameters which give the influence of slip angles on circumferential force

*Program variables are not listed for Subroutines MINV or HPCG

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
BRK(I) I=1,4	B_{RKi}	Brake torque coefficient for wheel i
B0(I) I=1,7		Part of {b} which does not depend on tire forces
B1(I),B3(I) B4(I),I=1,4	B_{1i}, B_{3i}, B_{4i}	Coefficients in peak lateral friction coefficient expression
C(I,J) I=1,7,J=1,7	C_{ij}	Matrix of inertia-type quantities in iterative solution to {a}
CALPHA	$C\alpha_i$	Low-slip angle tire cornering stiffness
CD	C_D	Aerodynamic drag coefficient
CI(I,J) I=1,7,J=1,7		Inverse of C_{ij}
CST1,CST2 CST3	C_{ST1}, C_{ST2} C_{ST3}	Steering system compliances (right, left, steering column)
D		Determinant of C_{ij}
DALF		Absolute value of $\bar{\alpha}$
DELSW	δ_{Sw}	Steering wheel angle

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
DELT(I)	δ_1, δ_2	Front wheel steering angles (right, left)
DELTP	p	Pitman arm angle
DELT10,DELT20		Values of DELT(I) from previous iteration
DEL1,DEL2		Arguments for DRIVER; correspond to DELT(I)
DERY(I) I=1,6		Input variable for HPCG
DSWMAX		Maximum allowable value for DELSW; required by DRIVER
DSWOUT		Output variable; corresponds to DELSW
DSW0		Initial value of DELSW; required by DRIVER
DT		Integration time step
DTOE(I) I=1,2	S_{TOE1}, S_{TOE2}	Toe-in angles of right and left front wheels, positive for positive rotation about z - axis
DTPRNT		Time increment at which output is printed
D3,D4		Dummy arguments; required by DRIVER

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
EC(I) I=1,5	e_i	Convergence criteria for the iterative solution to ¼a¶
F	f_i	Side force roll-off factor
FC(I) I=1,4	FC_i	Circumferential force at tire i
FMAX		Intermediate variable in tire side force calculation
FN(I) I=1,4	FN_i	Normal force at tire i
FRO(I,J,K) I=1,4 J=1,10 K=1,2		Lookup table for computation of f_i. K=1 gives S, K=2 gives f_i, I=tire number, J gives tabular values
FRR(I) I=1,4	K_{RRi}	Rolling resistance proportionality factors
FS(I) I=1,4	FS_i	Side force at tire I
FX(I),FY(I), FZ(I) I=1,4	F_{xi},F_{yi} F_{zi}	Components of tire forces at tire contact points
FZ10,FZ20 FZ30,FZ40		Values of FZ(I) from previous iteration

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
G	g	Gravity
G (subroutine TIRE3)	$G\alpha_i$	Tire side force saturation function
HF,HR	h_F, h_R	Height of roll center above ground plane, front and rear axles
IACKER		Steering type indicator; IACKER = 0: Parallel IACKER = 1: Ackerman
ICSET		Label for initial conditions data set (6 characters)
IFIRST		Indicator for first integration time step
IHLF		Error indicator for HPCG
IOTHER		Index identifying laterally opposite tire
JUNK1(I),JUNK2(I) I=1,7		Dummy vectors required by MINV
KD		Understeer factor; required by DRIVER
K1(I),K2(I) I=1,4	K_{1i}, K_{2i}	Coefficients in tire aligning torque calculations

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
NEQ		Number of equations integrated by HPCG
PFA	A_{PF}	Projected frontal area of vehicle
PFL	P_{FL}	Brake-line pressure
PFLMAX		Maximum allowable value of PFL; required by DRIVER
PFLOUT		Output variable; corresponds to PFL
PFL0		Initial value of PFL; required by DRIVER
PI02		$\pi/2$
PLIZZ,PLIYZ, PLIZX	$I^2{}_{zz}, I^2{}_{yz}$ $I^2{}_{zx}$	Movements of inertia of payload w.r.t. $x_2y_2z_2$ axes
PLM	m_2	Mass of payload
PRMT(I) I=1,4		Control parameters required by HPCG
PSI0		Initial value of
P0(I),P1(I) P2(I) I=1,4	$P_{0i}, P1_i$ P_{2i}	Coefficients in the peak coefficient of friction versus normal force expression

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
R	r	Angular velocity of vehicle about z-axis
RDOT	r	d(r)/dt
RHOA	ρ_A	Density of air
RT(I) I=1,4	RT_i	Rolling radius of tire i
R0(I),R1(I) I=1,4	R_{0i}, R_{1i}	Coefficients in longitudinal slip versus normal load expression
S(I) I=1,4	S_i	Longitudinal slip at tire I
SI	SI_i	Longitudinal slip at which peak coefficient of friction occurs
SN(I) I=1,4	SN_i	Skid number ratio: present surface/ measurement surface
SR	NG	Steering ratio
S0(I),S1(I) S2(I) I=1,4	S_{0i}, S_{1i} S_{2i}	Coefficients in expression for the sliding friction coefficient
T	t	Time
TAXL(I)	DT_1, DT_2	Drive torque transfer

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
I=1,2		parameter: fraction of drive torque acting at axle I (front,rear) which would cause axle roll relative to chassis. TAXL↑0 corresponds to negative axle roll about x-axis
THETAT(I) I=1,4	i	Angle between x-axis and tire contact point velocity vector at tire i
TIRCON		Label for tire data set (6 characters)
TL(I) I=1,4	TL_i	Drive torque limit. When driven wheel i spins, only generating TL_i, then TL_i is drive torque input to wheel IOTHER
TMAX		Simulation stopping time
TP		tan (δ_P)
TQAL(I) I=1,4	TQ_{ALi}	Aligning torque at tire i
TQD		Drive torque
TQDMAX		Maximum allowable value of TQD; required by DRIVER
TQDOUT		Output variable; cor-

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
		responds to TQD
TQDO		Initial value of TQD; required by DRIVER
TQST(I) I=1,2	TQ_{ST1}, TQ_{ST2}	Steering torque about the kingpin due to tire forces and moments (right,left)
TQT(I) I=1,4	TQ_{ti}	Drive/brake torque at tire i
T1		Time at which output was most recently printed
U,V	u,v	Components of vehicle velocity along x and y axes
UDOT,VDOT	u,v	du/dt, dv/dt
UP	u_{Pi}	Peak coefficient of friction
US	Si	Coefficient of sliding friction
UT(I),VT(I) I=1,4		Components of tire contact point velocity along x and y axes
U0,V0		Initial values of u, v
U1	1i	Effective coefficient of sliding friction

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
VC	z_{ti}	Height of vehicle center of mass above ground plane
VEHCON		Label for vehicle data set (6 characters)
VEL		Speed of vehicle
VIZZ,VIYZ VIZX	I^1_{zz}, I^1_{yz} I^1_{zx}	Moments of inertia of vehicle w.r.t. xyz axes
VM	m_1	Vehicle mass
XC(I) I=1,2	C_x	Caster offset
XL	L	Wheelbase
XM1	Mi	Effective maximum coefficient of friction
XPL,YPL,ZPL	x_p, y_p, z_p	Coordinates of center of mass of payload w.r.t. xyz axes
XT(I),YT(I) I=1,4	x_{ti}, y_{ti}	Coordinates of tire contact points w.r.t. xyz axes
X0,Y0		Initial values of inertial position X and Y

Program Variable Definitions for Three-Degree-of-Freedom Model

Program Variable	Analytic Symbol	Definition
Y(I) I=1,6		State vector: Y(I) = X Y(2) = Y Y(3) = ψ Y(4) = u Y(5) = v Y(6) = r

2.5. Three-Degree-of-Freedom Vehicle Data For Base Configuration

x_{ti}, y_{ti}, z_{ti}	48.0	30.75	21.74
(coordinates of tire	48.0	-30.75	21.74
contact points), in	-61.0	30.50	21.74
	-61.0	-30.50	21.74
δ_{TOE1}, δ_{TOE2} (toe angles), rad	0.0	0.0	
drive torque distri-	0.0	0.0	
bution parameters (fraction of T_{QD} at each wheel)	0.5	0.5	
λ_{TT} (drive torque distribution parameter: front, rear)	0.0	0.0	
λ_{RS} (roll-stiffness ratio, front/total)	0.66		

rear axle ratio	2.79		
g, in/sec^2	386.4		
m_1, lb-sec^2-in	9.403		
$I^1{}_{zz}$, lb-sec^2-in	22500.0		
$I^1{}_{yz}$, lb	0.0		
$I^1{}_{zx}$, lb-sec^2-in	-230.0		
x_p, y_p, z_p (payload coordinates), in	-6.0	0.0	-6.0
m_2, lb-sec^2/in	0.83		
$I^2{}_{zz}$, lb-sec^2-in	0.0		
$I^2{}_{yz}$, lb-sec^2-in	0.0		
$I^2{}_{zx}$, lb-sec^2-in	0.0		

C_{ST1} (right steering linkage compliance), rad/(in-lb)	0.000005404	
C_{ST2} (left steering linkage compliance), rad/(in-lb)	0.000005404	
C_{ST3} (steering column compliance), rad/(in-lb)	0.001389	
SR (overall steering ratio)	17.5	
h_F (height of front roll center), in	0.0	
h_R (height of rear roll center), in	0.0	
C_{xi} (caster offset), in	0.0	0.0
BK_i (brake torque coefficient), in-lb/(lb/in^2) = in^3	30.0 20.0	30.0 20.0
C_D (drag coefficient)	0.45	

A_p (projected frontal area), in^2	3100.0
ρ_A (air density), $lb\text{-}sec^2/in^4$	0.000000115

Convergence criteria used in solution:

$$\varepsilon_{convg} = 0.05$$
$$\varepsilon_1 = 0.001 \text{ in/sec}^2$$
$$\varepsilon_2 = 0.001 \text{ in/sec}^2$$
$$\varepsilon_3 = 0.0001 \text{ rad/sec}^2$$
$$\varepsilon_4 = \varepsilon_5 = \varepsilon_6 = \varepsilon_7 = 0.1 \text{ lb}$$
$$\varepsilon_8 = 0.001 \text{ rad}$$

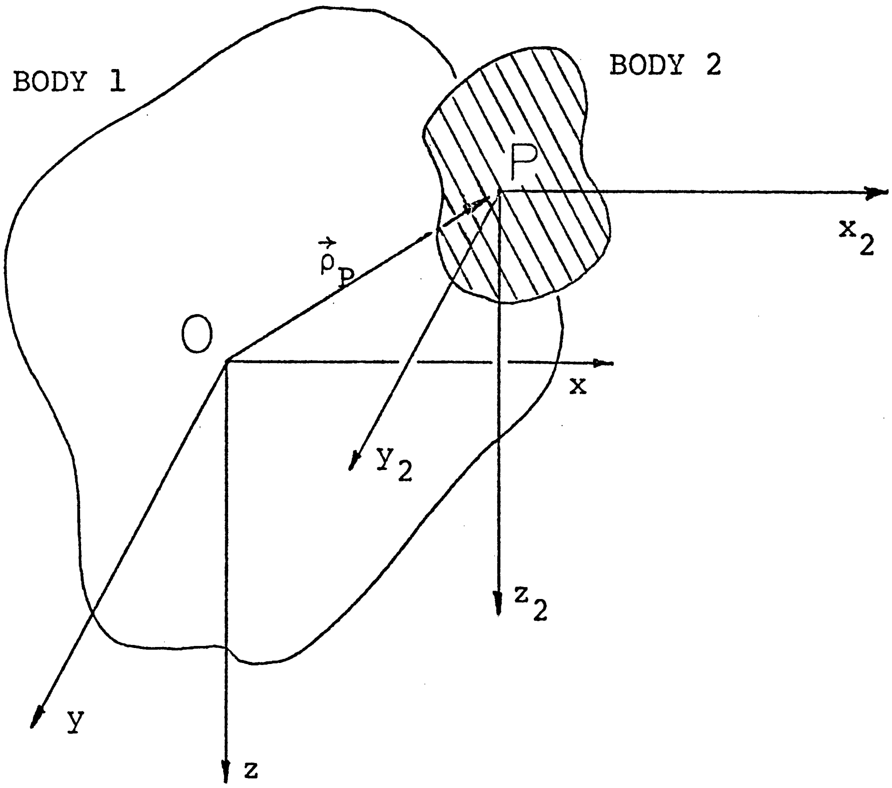


FIGURE 2.1 BODY-FIXED AXIS SYSTEMS

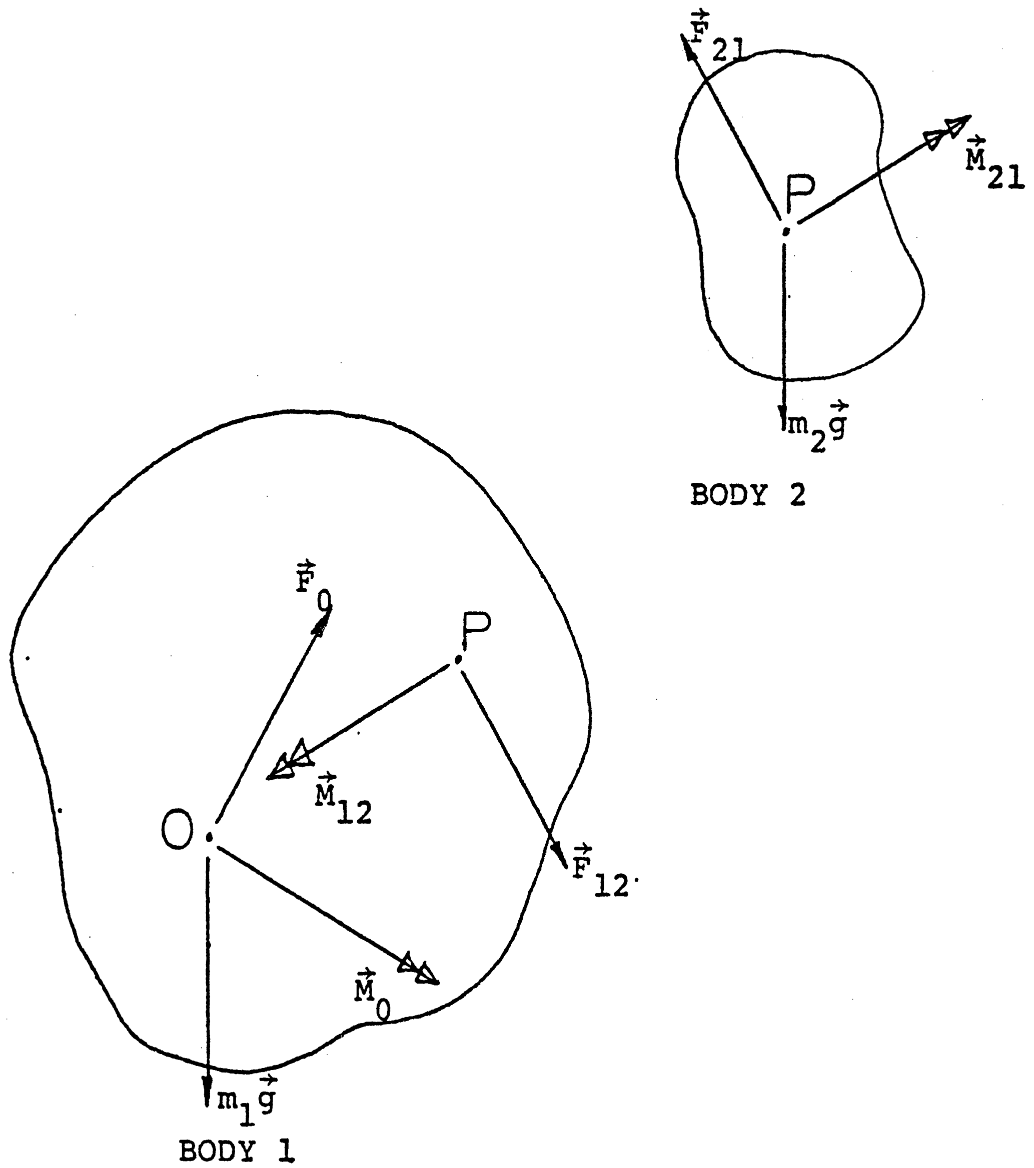


FIGURE 2.2 FREE-BODY DIAGRAMS OF BODY 1 AND BODY 2

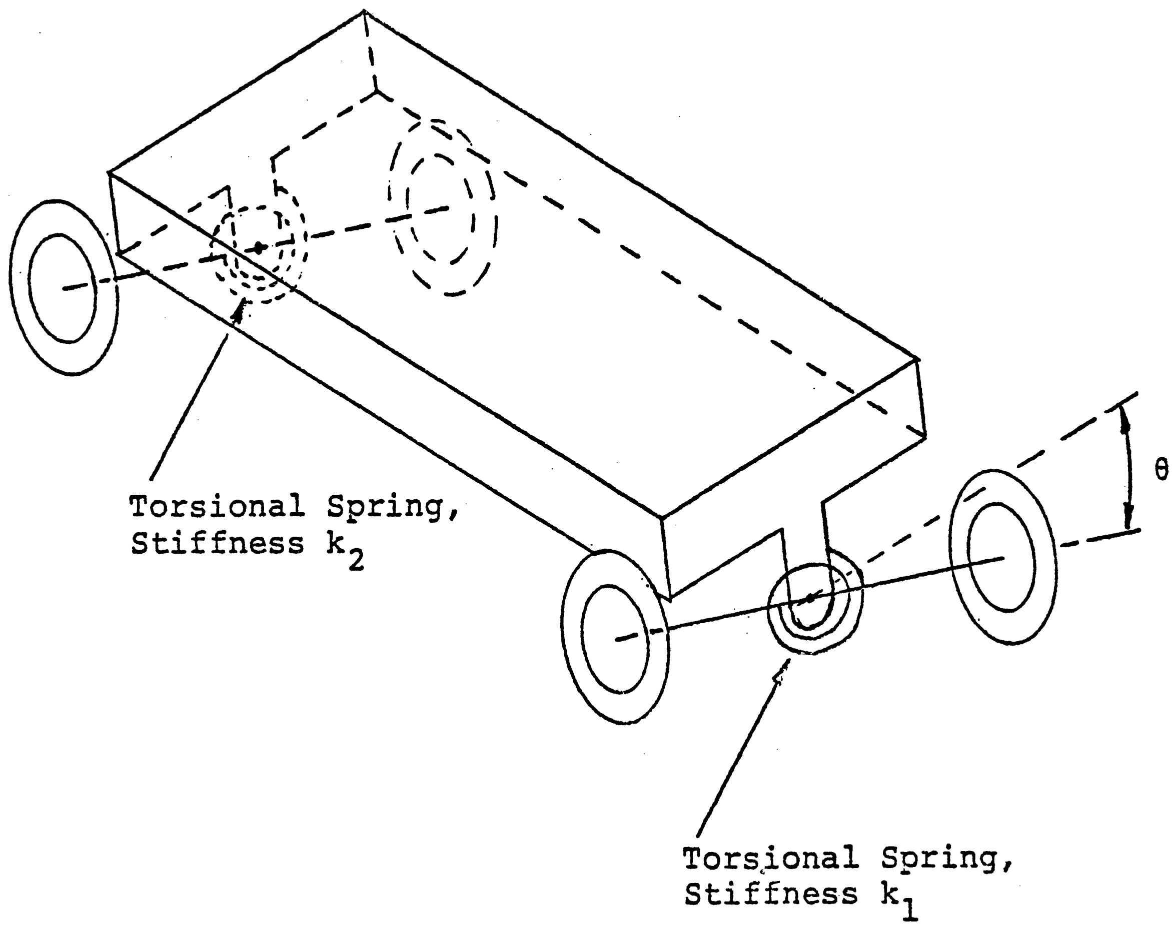


FIGURE 2.3 CONCEPTUAL MODEL FOR NORMAL FORCE DISTRIBUTION

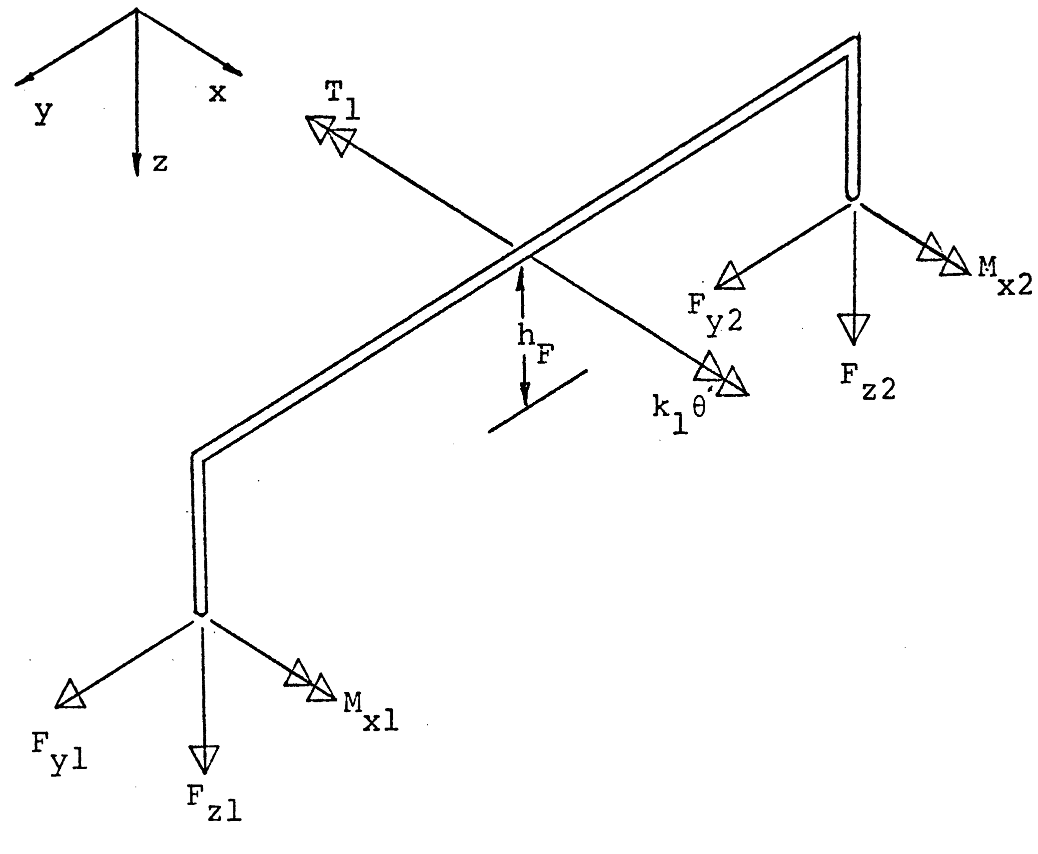


FIGURE 2.4 FREE-BODY DIAGRAM OF FRONT AXLE OF CONCEPTUAL MODEL

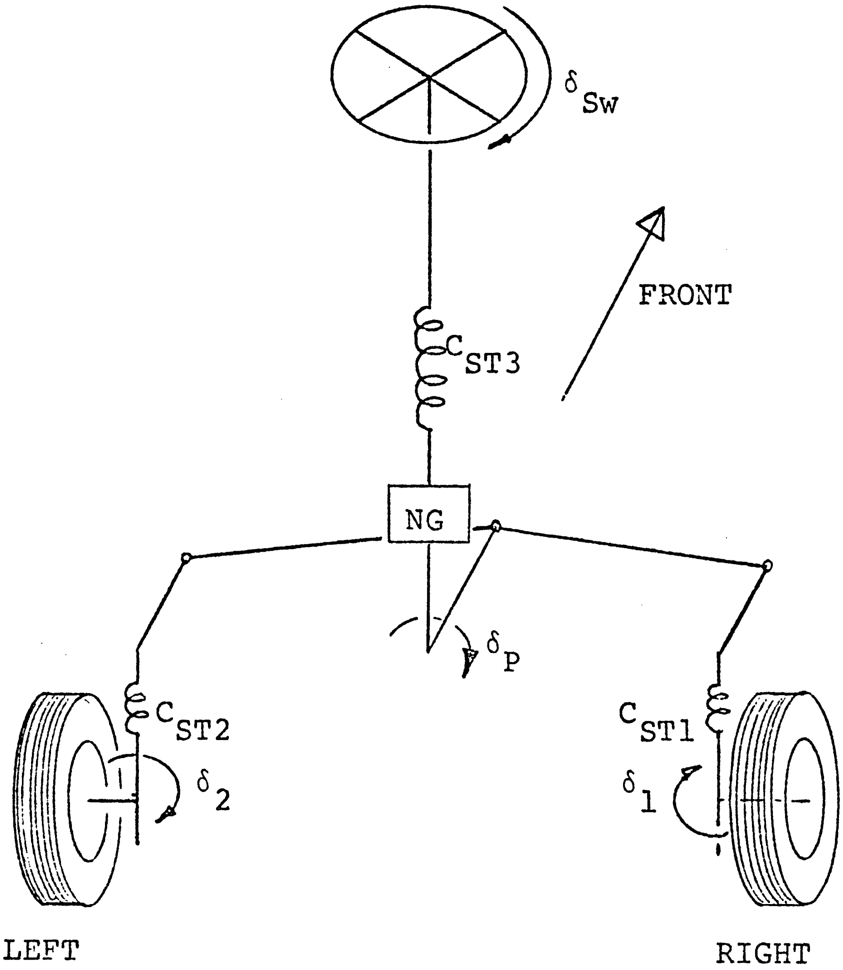


FIGURE 2.5 SCHEMATIC DIAGRAM OF STEERING SYSTEM, 3-DOF MODEL

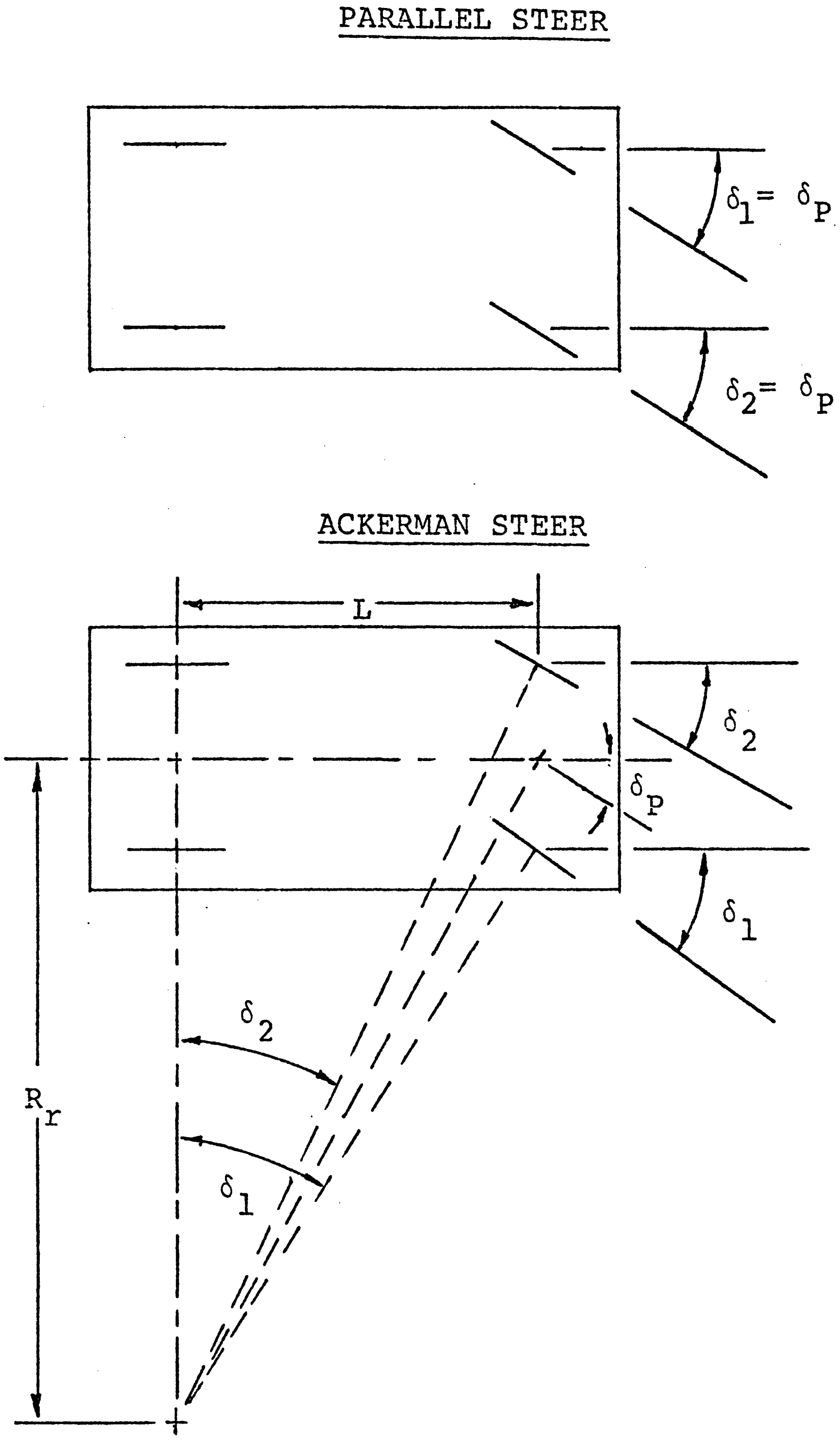
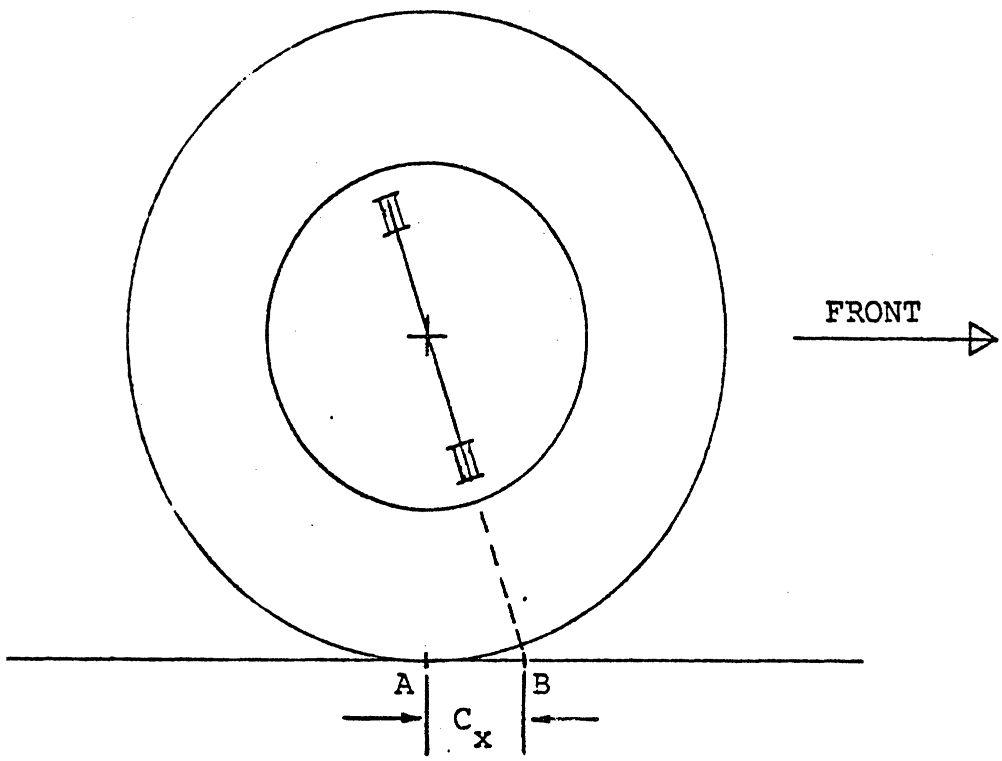


FIGURE 2.6 STEERING LINKAGE GEOMETRIES: PARALLEL AND ACKERMAN



A = Tire contact point

B = Intersection of kingpin axis with ground plane

C_x = Caster Offset

FIGURE 2.7 SCHEMATIC DIAGRAM OF CASTER OFFSET

REFERENCES FOR CHAPTER 2

2.1 "Reserch on the Influence of Tire Properties on Vehicle Handling", Calspan Corporation, Rept. No. DOT-HS-053-3-727, 1976.

2.2 "Tire Parameter Determination", D. J. Schuring, DOT-HS-4-00923, 1975.

2.3 "The Dynamics of Pneumatic Tires", Motor Vehicle Performance - Measurement and Prediction", Highway Safety Research Institute, Univ. of Michigan, 1974.

2.4 "Analysis of Tire Properties", H. Pacejka, Chapter 9, Mechanics of Pneumatic Tires (S. K. Clark, Ed.) NHTSA, 1982.

2.5 "A Handbook for the Rolling Resistance of Pneumatic Tires", S. K. Clark and R. N. Dodge, DOT-TSC-1031, 1978.

2.6 System/360 Scientific Subroutine Package, Version III, Programmer's Manual, IBM Corporation, 1970.

2.7 "Improvement of Mathematical Models for Simulation of Vehicle Handling", Vol. 6. Programmer's Guide for the DRIVER MODULE, W. R. Garrott, D. L. Wilson, A. M. White and R. A. Scott. Final Report DOT-HS-7-01715, December 1979.

List of Symbols

The nomenclature used in Chapter 2 is presented below.

Symbol	Definition
$\vec{a}_1$, $\vec{a}_2$	Acceleration of body 1 and body 2, respectively. Body 1 is the vehicle. Body 2 is the payload.
$\{a\}$, a_i	Vector of kinematic derivatives, normal tire forces, and steering angle.
A_{PF}	Projected frontal area of vehicle.
A_{0i}, A_{1i}, A_{2i}	Coefficients in expression for low-slip-angle cornering stiffness.
B_{1i}, $B_{3i} B_{4i}$	Coefficients of curves fitted to the peak lateral friction coefficients.
B_{Ci}	Tire parameters which give the influence of slip angles on circumferential tire forces.
B_{RKi}	Brake torque coefficient for wheel i.
$\{b\}$, b_i	Vector of force type terms.
C_D	Aerodynamic drag coefficient.
C_{ij}	Matrix of inertia-type quantities in iterative solution to $\{a\}$.
C_{ST1}, C_{ST2}, C_{ST3}	Steering system compliances.
C_X	Caster offset.

$C_{\alpha i}$	Low-slip-angle tire cornering stiffness.
f_i	Side force roll-off factor for tire i.
$\vec{F}_o$	Net force acting on body 1, excluding gravity and $\vec{F}_{12}$
F_x, F_y, F_z	Components of $\vec{F}_o$
$F_{xaero}, F_{yaero}, F_{zaero}$	Components of the aerodynamic force that acts at the vehicle center of mass. These are designated as ΣF_{xs}, ΣF_{ys}, ΣF_{zs} in Ref [1.1].
$\vec{F}_{12}$	Force on body 1 exerted by the connection from body 2.
$\vec{F}_{21}$	Force on body 2 exerted by the connection from body 1.
F_{xi}, F_{yi}, F_{zi}	Components of the tire forces at the tire contact points.
FC_i	Circumferential force at tire i.
FC_{Ai}, FC_{Oi}	Intermediate variables in tire circumferential force calcultions.
FN_i	Normal force at tire i.
FS_i	Side force at tire i.
$\vec{g}$	The acceleration due to gravity.
G_{α_i}	Function that arises in tire side force saturation.

$h_F h_R$ — Height of roll center above ground plane, front and rear axles, respectively.

$\vec{H}^1, \vec{H}^2$ — Angular momentum of body 1 and body 2, respectively.

H_n^γ — Angular momentum components.

$[I^1],[I^2]$ — Inertial tensor of body 1 and body 2, respectively.

I^1_{nz}, I^2_{nz} — Moments of inertia of bodies 1 and 2 w.r.t. the z-axis

Note: $I^2_{xy}, I^2_{yz}, I^2_{xz}$ are defined such that the inertial tensor $[I^2]$ is given by

$$[I^2] = \begin{bmatrix} I^2_{xx} & -I^2_{xy} & -I^2_{xz} \\ -I^2_{xy} & I^2_{yy} & -I^2_{yz} \\ -I^2_{xz} & -I^2_{yz} & I^2_{zz} \end{bmatrix}$$

I^S_{mn} — Sum of corresponding moments of inertia of bodies 1 and 2.

K — Sum of torsional spring constants k_1 and k_2.

k_1, k_2 — Roll stiffness used in determining the normal tire force distribution.

K_{1i}, K_{2i} — Coefficients in tire aligning torque calculation.

K_{RRi} — Rolling resistance proportionality factors.

L — Wheel base.

$\vec{M}_o$ — Resultant external moment acting

	on body 1, excluding $\vec{m}_{12}$
M_x, M_y, M_z	Components of M_o.
$M_{x1}, M_{x2}, M_{x3}, M_{x4}$	X components of moments generated by the tires.
$\vec{M}_{12}$	Moment on body 1 exerted by the connection from body 2.
$\vec{M}_{21}$	Moment on body 2 exerted by the connection from body 1.
m_1, m_2	Mass of body 1 and body 2, respectively.
$M_{xaero}, M_{yaero}, M_{zaero}$	Components of the aerodynamic moment at the vehicle center of mass.
NG	Steering ratio.
P_{0i}, P_{1i}, P_{2i}	Coefficients in the peak coefficient of friction versus normal force relations.
P_{FL}	Brake line pressure.
r	Angular velocity of body 1 about the z-axis.
R_A	Axle drive ratio.
R_r	Rear axle center turning radius.
R_{0i}, R_{1i}	Coefficients in longitudinal slip versus normal load relations.
RT_i	Rolling radius of tire i.

S_i	Longitudinal slip of wheel i.
S_{0i}, S_{1i}, S_{2i}	Coefficients in the sliding friction coefficient versus normal load relations.
SI_i	Longitudinal slips for which the peak coefficients of friction occur.
SN_i	Skid number ratio: present surface/measurement surface.
t	Time.
T_1, T_2	Torque about the x-axis transmitted from the vehicle body to the solid front and rear axles, respectively, as a result of drive torque.
TL_i	Drive torque which can be utilized by tire i.
TQ_{ALi}	Tire aligning moments.
T_{QD}	Drive line torque.
TQ_{ST1}, TQ_{ST2}	Steering torque about the kingpin due to tire forces and moments (right, left).
TQ_{ti}	Drive/brake torque at tire i.
u	Component of $\vec{v}_1$ along the x-axis.
v	Component of $\vec{v}_1$ along the y-axis.
X,Y,Z	Inertial coordinate system. Z is directed downward

x, y, z	Body-fixed coordinate system. x and z are directed towards the front and bottom of the vehicle, respectively.
x_p, y_p, z_p	Coordinates of P (center of the payload) with respect to the xyz axes.
x_{ti}, y_{ti}	Coordinates of tire contact points with respect to the xyz axes.
x_2, y_2, z_2	Body-fixed coordinate system with origin at P (center of mass of payload), parallel to the xyz axes.
α_i	Slip angle at tire i.
$\bar{\alpha}_i$	Normalized slip angle at tire i.
β_i	Angle between x-axis and tire contact point velocity vector at tire i.
δ_p	Pitman arm angle.
δ_{Sw}	Steering wheel angle.
$\delta_{TOE1}, \delta_{TOE2}$	Toe-in angles of right and left front wheels, positive for positive (i.e., right-hand) rotation about the Z-axis.
δ_1, δ_2	Front wheel steering angles (right, left).
$\gamma (=1, 2)$	An index designating body 1 or body 2.

ε_i	Convergence criteria for the iterative solution to {a}.
θ	Roll of vehicle body about an axis parallel to the x-axis.
λ_{RS}	Fraction of total roll stiffness at front axle in conceptual model for normal tire force distribution.
λ_{TQi}	Fraction of drive torque applied at wheel i.
$\lambda_{DT1}, \lambda_{DT2}$	Drive torque transfer parameter (front, rear).
μ_{Mi}	Effective maximum coefficient of friction at tire i.
μ_{Pi}	Peak coefficient of friction for tire i.
μ_{Si}	Coefficient of sliding friction at tire i.
μ_{yi}	Peak lateral friction coefficents for tire i.
μ_{1i}	Effective coefficient of sliding friction at tire i.
ρ_A	Density of air.
$\vec{\rho}_P$	Position vector from vehicle center of mass to payload center of mass.
ψ	Heading angle; angle between inertial X axis and vehicle x axis.

$\vec{\omega}$ Angular velocity of body 1.

Chapter 3. Modifications Required in IDSFC to Simulate Asymmetric Vehicles

3.1 Changes in the Equations of Motion

The equations of motion of the vehicle in IDSFC are written

$$[M]\{x\} = \{F\} \tag{3.1}$$

where {x} is the state vector consisting of the ten degrees of freedom described by differential equations (the remaining seven "degrees of freedom" are handled using algebraic approximations), [M] is a matrix of inertial-like quantities, which depends on {x} through suspension deflections, and {F} is a vector of forcing functions which also depends on {x}, and is found through a numerical iterative procedure.

IDSFC was modified to include the options of adding a rigidly attached payload of arbitrary inertial properties at an arbitrary point in the vehicle sprung mass, and of transmitting an arbitrary portion of the drive torque from the chassis to the rear axle in the case of a driven solid rear axle. The equations below detail the changes necessary to implement the options above; the new terms are underlined. The "Technical Manual" referred to below is Ref. [3.1]. Variables m_2, I^2_{mn}, x_P, y_P, z_P, and λ_{TT} are defined in section 2.1.

Changes Necessary in {F}

Equations (2.2S) and (2.2I) in Technical Manual:

$$F(1) = (vr-wq-g\sin\theta)(\Sigma M+\underline{m_2}) + \Sigma F_{xs} + \Sigma F_{xu}$$

$$+ \underline{m_2(q^2x_p+r^2x_p-qpy_p-rpz_p)}$$

$$+ \text{ terms involving assumption switches } O_{ijkl} \qquad (3.2)$$

Equations (2.3S) and (2.3I) in Technical Manual:

$$F(2) = (wp-ur+g\cos\theta\sin\phi)(\Sigma M+\underline{m_2})$$

$$+ \underline{m_2(r^2y_p+p^2y_p-rqz_p-pqx_p)}$$

$$+ \Sigma F_{ys} + \Sigma F_{yu} + \text{ terms involving } O_{ijkl}. \qquad (3.3)$$

Equations (2.4S) and (2.4I) in Technical Manual:

$$F(3) = (uq-vp+g\cos\theta\cos\phi)(M_S+\underline{m_2})$$

$$+ \underline{m_2(p^2z_p+q^2z_p-prx_p-qry_p)}$$

$$+ \Sigma F_{zs} - \sum_{i=1}^{4} S_i \qquad (3.4)$$

Equation (2.5S) in Technical Manual:

$$F(4) = \gamma_2'(ur-wp-g\cos\theta\sin\phi) + \Sigma N_{\phi s} + \Sigma N_{\phi u}$$

$$+ \underline{N_{\phi 2}} + \underline{T_2} + \text{terms involving } O_{ijkl} \qquad (3.5)$$

Equation (2.5I) in Technical Manual:

$$F(4) = \gamma_2'(ur-wp-g\cos\theta\sin\phi) + \Sigma N_{\phi s} + \Sigma N_{\phi u}$$

$$+ \underline{N_{\phi 2}} + \text{terms involving } O_{ijkl} \qquad (3.6)$$

where

$$N_{\phi 2} = I^2{}_{xz}pq + I^2{}_{yz}q^2 - I^2{}_{zz}rq - I^2{}_{xy}pr + I^2{}_{yy}qr - I^2{}_{yz}r^2$$

$$+ m_2(uqy_p-vpy_p-prx_py_p+q^2z_py_p-qry_p{}^2$$

$$+urz_p-wpz_p+rqz_p{}^2-r^2y_pz_p+pqx_pz_p$$

$$+y_pg\cos\theta\cos\phi-z_pg\cos\theta\sin\phi) \qquad (3.7)$$

and

$$T_2 = TQD\lambda_{TT} \qquad (3.8)$$

Equations (2.6S) and (2.6I) in Technical Manual:

$$F(5) = \gamma_2'(vr-wq-g\sin\theta) + \Sigma N_{\theta s} + \Sigma N_{\theta u} + \underline{N_{\theta 2}}$$

$+ \text{ terms involving } O_{ijkl}$ (3.9)

where

$$N_{\theta 2} = -I^2_{xx}pr + I^2_{xy}qr + I^2_{xz}r^2 - I^2_{xz}p^2 - I^2_{yz}qp$$

$$+ I^2_{zz}rp + m_2(vrz_p - wqz_p - qpy_pz_p + r^2x_pz_p - rpz_p^2$$

$$+vpx_p - uqx_p + prx_p^2 - p^2z_px_p + qry_px_p$$

$$-z_pg\sin\theta - x_pg\cos\theta\cos\phi) \qquad (3.10)$$

Equations (2.7S) and (2.7I) in Technical Manual:

$$F(6) = \gamma_1(wp - ur + g\cos\theta\sin\phi) + \Sigma N_{\psi s} + \Sigma N_{\psi u} + \underline{N_{\psi 2}}$$

$+ \text{ terms involving } O_{ijkl}$ (3.11)

where

$$N_{\psi 2} = I^2_{xy}p^2 - I^2_{yy}pq + I^2_{yz}rp + I^2_{xx}pq - I^2_{xy}q^2 - I^2_{xz}rq$$

$$+ m_2(wpx_p - urx_p - rqz_px_p + p^2y_px_p - pqx_p^2$$

$$+wqy_p - vry_p + qpy_p^2 - q^2x_py_p + rpz_py_p$$

$$+x_p g\cos\theta\sin\phi + y_p g\sin\theta) \quad (3.12)$$

Equation (2.11S) in Technical Manual:

$$F(10) = \Sigma N_{\phi R} - \underline{T_2} + \text{terms involving } 0_{ijkl} \quad (3.13)$$

Changes Necessary in [M]

Equation (2.16) in Technical Manual:

$$M(1,1) = \Sigma M + \underline{m_2} \quad (3.14)$$

$$M(2,2) = \Sigma M + \underline{m_2} \quad (3.15)$$

Equation (2.22) in Technical Manual:

$$M(3,3) = M_S + \underline{m_2} \quad (3.16)$$

Equation (2.15) in Technical Manual:

$$M(3,4) = \underline{m_2 y_p} \quad (3.17)$$

$$M(3,5) = \underline{m_2 x_p} \quad (3.18)$$

$$M(4,3) = \underline{m_2 y_p} \quad (3.19)$$

$$M(4,5) = \underline{-I^2_{xy} - m_2 x_p y_p} \quad (3.20)$$

$$M(5,3) = \underline{-m_2 x_p} \qquad (3.21)$$

$$M(5,4) = \underline{-I_{xy}^2 - m_2 x_p y_p} \qquad (3.22)$$

Equations (2.17S) and (2.17I) in Technical Manual:

$$M(5,1) = \gamma_2' + \underline{m_2 z_p} + \text{terms involving } 0_{ijkl} \qquad (3.23)$$

Equations (2.18S) and (2.18I) in Technical Manual:

$$M(6,1) = \underline{-m_2 y_p} + \text{terms involving } 0_{ijkl} \qquad (3.24)$$

Equations (2.19) in Technical Manual:

$$M(4,2) = -\gamma_2' - \underline{m_2 z_p} + \text{terms involving } 0_{ijkl} \qquad (3.25)$$

Equation (2.20) in Technical Manual:

$$M(2,6) = \gamma_1 + \underline{m_2 x_p} \qquad (3.26)$$

$$M(6,2) = \gamma_1 + \underline{m_2 x_p} \qquad (3.27)$$

Equations (2.26S) and (2.26I) in Technical Manual:

$$M(2,4) = -\gamma_2' - \underline{m_2 z_p} + \text{terms involving } 0_{ijkl} \qquad (3.28)$$

$$M(1,5) = \gamma_2' + \underline{m_2 z_p} + \text{terms involving } 0_{ijkl} \qquad (3.29)$$

Equations (2.27S) and (2.27I) in Technical Manual:

$$M(4,4) = I_x + I_x' + \underline{I_{xx}^2 + m_2(y_p^2 + z_p^2)}$$

$$+ \text{terms involving } 0_{ijkl} \qquad (3.30)$$

Equations (2.28S) and (2.28I) in Technical Manual:

$$M(6,4) = -I_{xz} - I_{xz}' - \underline{I_{xz}^2 - m_2 x_p z_p}$$

$$+ \text{terms involving } 0_{ijkl} \qquad (3.31)$$

Equations (2.31S) and (2.31I) in Technical Manual:

$$M(5,5) = I_y + I_y' + \underline{I_{yy}^2 + m_2(x_p^2 + z_p^2)}$$

$$+ \text{terms involving } 0_{ijkl} \qquad (3.32)$$

Equations (2.32S) and (2.32I) in Technical Manual:

$$M(6,5) = \underline{-I_{yz}^2 - m_2 y_p z_p} + \text{terms involving } 0_{ijkl} \qquad (3.33)$$

Equations (2.36S) and (2.36I) in Technical Manual:

$$M(1,6) = \underline{-m_2 y_p} + \text{terms involving } 0_{ijkl} \qquad (3.34)$$

Equations (2.37S) and (2.37I) in Technical Manual:

$$M(4,6) = -I_{xz} - I'_{xz} - \underline{I^2_{xz} - m_2 x_p z_p}$$

$$+ \text{ terms involving } 0_{ijkl} \tag{3.35}$$

Equations (2.38S) and (2.38I) in Technical Manual:

$$M(5,6) = \underline{-I^2_{yz} - m_2 y_p z_p} + \text{ terms involving } 0_{ijkl} \tag{3.36}$$

Equation (2.39S) in Technical Manual:

$$M(6,6) = I_R + I_z + M_{uF}(a^2+T_F{}^2/4) + M_{uR}b^2$$

$$+ \underline{I^2_{zz} + m_2(x_p{}^2+y_p{}^2)}$$

$$+ \text{ terms involving } 0_{ijkl} \tag{3.37}$$

Equation (2.39I) in Technical Manual:

$$M(6,6) = I_z + M_{uF}(a^2+I_F{}^2/4) + M_{uR}(b^2+I_R{}^2/4)$$

$$+ \underline{I^2_{zz} + m_2\ (x_p{}^2 + y_p{}^2)} \tag{3.38}$$

3.2 Running the IDSFCAS Simulation

The modified version of IDSFC described by the equations above is referred to as IDSFCAS (IDSFC-Asymmetric). The following changes in simulation operating instructions are necessary as a result of the modifications:

1) Payload data must be read in on I/O unit 9 in the format and order described below.

Format for payload data deck: (card number, data) in format (I4,1X,F20.10)

Order of payload data deck:

Card Number	Variable to be Read
1	m_2(lb-sec^2/in)
2	x_p(in)
3	y_p(in)
4	z_p(in)
5	I_{xx}^2(lb-sec^2-in)
6	I_{yy}^2(lb-sec^2-in)
7	I_{zz}^2(lb-sec^2-in)
8	J_{xy}^2(lb-sec^2-in)
9	J_{yz}^2(lb-sec^2-in)
10	I_{xz}^2(lb-sec^2-in)
11	λ_{TT}(dimensionless)

2) Initial suspension deflections (which are required by the simulation) corresponding to the static equilibrium position of the vehicle plus payload must be determined by

one of two procedures: a calculation based on the given force/suspension-deflection data, or a brief simulation run. Suspension forces in IDSFCAS are given by interpolation from tabular data as a function of suspension deflection. For payloads which are sufficiently small that the suspension deflections remain within the initial linear segment of the force/deflection table the equations below can be used to compute the initial deflections. After the deflections have been computed they should be checked against the force/deflection table to insure that the deflections remained within the initial segment.

Equations for the loaded equilibrium position were derived by minimizing a potential energy expression in terms of suspension and tire deflections. Tire compliances are in general large enough that they must be included in the formulation.

The potential energy of the loaded vehicle with respect to the unloaded equilibrium position is given by equation (3.39), where δ_{si} is the suspension deflection for wheel i, measured at the wheel for an independent suspension and at the spring mount for a solid axle, δ_{ti} is the deflection of tire i, k_{si} is the stiffness of the suspension for wheel i, effective at the wheel for independent suspension and at the spring mount for solid axle suspension, k_{ti} is the tire stiffness, k_F and k_R are auxiliary front and rear roll stiffness, T_F and T_R are front and rear track widths, and T_{sF} and T_{sR} are the front and rear spring mount spacings (which are equal to T_F and T_R, respectively, for independent suspensions). Deflections are taken to be positive in elongation.

$$PE = \sum_{i=1}^{4} (k_{si}\delta_{si}^2 + k_{ti}\delta_{ti}^2)/2 + k_F(\delta_{s1}-\delta_{s2})^2/T_{sF}^2$$

$$+ k_R(\delta_{s3}-\delta_{s4})^2/T_{sR}^2 + m_2 g \Delta h_p \qquad (3.39)$$

where

$$\Delta h_p = (\delta_{s1}+\delta_{s2}+\delta_{t1}+\delta_{t2})(b+x_p)/[2(a+b)]$$

$$+ (\delta_{s3}+\delta_{s4}+\delta_{t3}+\delta_{t4})(a-x_p)/[2(a+b)]$$

$$+ [(\delta_{s1}-\delta_{s2})/T_{sF}+(\delta_{t1}-\delta_{t2})/T_F]y_p \qquad (3.40)$$

The frame and body are assumed to be rigid, so that the body roll is the same at the front and rear. This constraint is expressed by equation (3.41).

$$(\delta_{s1}-\delta_{s2})/T_{sF} + (\delta_{t1}-\delta_{t2})/T_F$$

$$- (\delta_{s3}-\delta_{s4})/T_{sR} + (\delta_{t3}-\delta_{t3})/T_R = 0 \qquad (3.41)$$

A constrained potential energy expression PE' is formed by multiplying the left-hand side of (3.41) by a Lagrange multiplier λ and adding it to (3.39). The equilibrium equations are then given by

$$PE'/\partial\xi_i = 0 \quad i = 1,9 \tag{3.42}$$

where

$$\xi_i = (\delta_{s1}, \delta_{s2}, \delta_{s3}, \delta_{s4}, \delta_{t1}, \delta_{t2}, \delta_{t3}, \delta_{t4}, \lambda)$$

After some algebra the following expressions are obtained for the suspension deflections δ_{si}.

$$[K] \begin{Bmatrix} \delta_{s1} \\ \\ \delta_{s3} \end{Bmatrix} = -m_2 g \begin{Bmatrix} A+C-F/T_{sF}+2Ak_F/(k_{s2}T_{sF}^2) \\ \\ B+F/T_{sR}+2Bk_R/(k_{s4}T_{sR}^2) \end{Bmatrix} \tag{3.43}$$

$$\delta_{s2} = (-2m_2gA-k_{s1}\delta_{s1})/k_{s2} \tag{3.44}$$

$$\delta_{s4} = (-2m_2gB-k_{s3}\delta_{s3})/K_{s4} \tag{3.45}$$

where

$$K_{11} = k_{s1} + (k_F+E)(1+k_{s1}/k_{s2})/T_{sF}^2$$

$$K_{12} = -E(1+k_{s3}/k_{s4})/(T_{sF}T_{sR})$$

$$K_{21} = -E(1+k_{s1}/k_{s2})/(T_{sF}T_{sR})$$

$$K_{22} = k_{s3} + (k_R+E)(1+k_{s3}/k_{s4})/T_{sR}^2$$

$$A = (b+x_p)/(2a+2b)$$

$$B = (a-x_p)/(2a+2b)$$

$$C = -y_p/T_{sF}$$

$$D = -y_p/T_F$$

$$E = 1/[1/(k_{t1}T_F^2)+1/(k_{t2}T_F^2)$$

$$+1/(k_{t3}T_R^2)+1/(k_{t4}T_R^2)]$$

$$F = E[(1/k_{t1}-1/k_{t2})A/T_F-(1/k_{t3}-1/k_{t4})B/T_R$$

$$+(1/k_{t1}+1/k_{t2})D/T_F-2A/(k_{s2}T_{sF})+2B/(k_{s4}T_{sR})]$$

For solid rear axle suspensions the initial conditons which must be specified are δ_R and ϕ_R. These are computed by the following equations.

$$\delta_R = (\delta_{s3}+\delta_{s4})/2 \qquad (3.46)$$

$$\phi_R = (\delta_{s3}-\delta_{s4})/T_{sR} \qquad (3.47)$$

Initial conditions can also be determined by performing a brief simulation run with either zero steering angle input or closed-loop driver control. The zero steering angle procedure is appropriate for symmetric vehicle configurations with payloads which are sufficiently large that the suspension operates outside of the initial linear range (or simply as an alternative to Equations (3.43)-(3.47)). A three-to-five second run at moderate speed (e.g., 400 in/sec) should be sufficient to determine the loaded equilibrium position. Extremely low speeds (below 100 in/sec) are not recommended because the wheel spin equations are very sensitive at such speeds.

Some asymmetric configurations, such as those involving geometrically asymmetric center of gravity location in a vehicle with significant roll steer, will not travel in a straight line for a zero steer angle. A closed-loop driver model, such as the "Straight Line Crossover Model" in the IDSFC Driver Module, may be used to control the steering angle to obtain straight line motion. (Driver time delay may be set to zero to improve convergence to a straight path.) This method will provide the steering wheel path angle necessary for straight line motion and the resultant vehicle side-slip angle as well as suspension deflections.

It should be noted that initial conditions computed from equations (3.43) - (3.47) will in general not agree exactly with those obtained from a simulation run. This discrepancy is largely due to coulomb friction in the suspension (which is not included in the static analysis), with small contributions coming from approximations to the trigonometric functions and other minor sources.

REFERENCE FOR CHAPTER 3

3.1 "Improvement of Mathematical Models for Simulation of Vehicle Handling, Vol. 7: Technical Manual", W. R. Garrott and R. A. Scott. Final Report DOT-HS-7-01715. March 1980

Chapter 4. Modifications in Driver Module

4.1. Technical Changes Required to Implement Extended Cross-Over Model

Based on a cross-over model of human driving, Garrott et al [4.1], implemented, in the Driver Module, the following control law for steering regulation while maintaining a constant forward speed*.

$$\delta_{Sw}(t) = K_\psi \psi_{err} + K_\psi T_{eq}\dot{\psi}_{err}$$

$$+ K_\psi K_Y Y_{err} + K_\psi K_Y T_{eq} Y_{err} \qquad (4.1)$$

However, this control law is inadequate for curved paths. If the vehicle is perfectly positioned, i.e. $\psi_{err} = \dot{\psi}_{err} = Y_{err} = \dot{Y}_{err} = 0$, then $\delta_{Sw} = 0$, which is not correct. To remedy this, following Garrott et al [4.2], it is assumed that the driver is able to perceive path curvature and vehicle speed u and is able to determine and implement the steering angle necessary to cause a steady turn of that curvature. This is achieved by the addition of the term,

$$(NG)\,[1+(KD)u^2]\,L\,\kappa(t-\tau)$$

A further extension has also been performed, following a proposal of Allen [4.3]. This extension involves the

*A list of symbols is given at the end of the chapter.

addition of an "integral trim" term. Thus, the latest version of the control law in the Driver Module, is

$$\delta_{Sw}(t) = K_{\psi}\psi_{err}(t-\tau) + K_{\psi}T_{eq}\,\dot{\psi}_{err}(t-\tau)$$

$$+ K_{\psi}K_{Y}Y_{err}(t-\tau) + K_{\psi}K_{Y}\,T_{eq}\,Y_{err}(t-\tau)$$

$$+ K\int^{t-\tau}\Big\{K_{\psi}\psi_{err}(\xi) + K_{\psi}T_{eq}\,\dot{\psi}_{err}(\xi)$$

$$+ K_{\psi}K_{Y}Y_{err}(\xi) + K_{\psi}K_{Y}T_{eq}\,Y_{err}(\xi)\Big\}\,d\xi$$

$$+ (GR)\,[1 + (KD)u^{2}]\,L\;\;\kappa(t-\tau) \qquad (4.2)$$

The following change was made in the desired heading gle computation in the Driver module.

ψ_{des} = linear interpolation between ψ_i and ψ_j, where ψ_i and ψ_j are the average path angle at the two end points of the closest road segment ($j = i+1$), defined by

$$\psi_i = (\psi_A + \psi_B)/2 \qquad (4.3)$$

$$\psi_j = (\psi_B + \psi_C)/2 \qquad (4.4)$$

$$\psi_A = \tan^{-1} (Y_{RPi} - Y_{RPi-1})/(X_{RPi} - X_{RPi-1}) \qquad (4.5)$$

$$\psi_B = \tan^{-1} (Y_{RPj} - Y_{RPi})/(X_{RPj} - X_{RPi}) \qquad (4.6)$$

$$\psi_C = \tan^{-1} (Y_{RPj+1} - Y_{RPj})/(X_{RPj+1} - Y_{RPj}) \qquad (4.7)$$

$$\text{If} \quad i = 1 \qquad \psi_A = \psi_B \qquad (4.8)$$

$$\text{If} \quad j = N_{RDPT} \quad \psi_C - \psi_B \qquad (4.9)$$

4.2 Programming Changes Required to Implement Extended Cross-over Model

The extended cross-over model and improved path curvature calculation described in section 4.1 were programmed into the Driver Module. Alterations were made in subroutines DRINPT, DRIOUT, DRINIT, DCROV, and DCRERR. Listings of the modified Fortran code for these subroutines are given on the following pages.

For subroutines DRINPT, DRIOUT, and DRINIT only the portions of the code affected by the changes are listed. New lines are denoted by ">". For subroutines DRCROV and DCRERR the entire subroutine is listed.

Changes in subroutine DRINPT

```
C     ***************************************C
C
C     SUBROUTINE DRINPT
C     THIS SUBROUTINE READS THE DRIVER DATADECK INTO THE SIMULATION.
      SUBROUTINE DRINPT
      IMPLICIT REAL(A - H,O - Z)
      INTEGER DRMODE
      LOGICAL ATEND, PASTOB, VIEWOB
      COMMON /DRDMAP/ RDPT(3,300), IPT1, NRDPT
      COMMON /DROLCM/ OLCOM(6,300), OLTIM(2,10), PPPARM(14),
     1       IOLCOM(300), NINT, NMAX, NOLP
      COMMON /DROPMD/ DRMODE
      COMMON /DRPAR1/ DRPR1(9), PE(12), WTAC(10), WTST(10),
     1       NPRED, NOPERR
      COMMON /DRPAR2/ DGAIN(7), TAU, VDES, DELINT, ACCINT, TLCR
      COMMON /DRPAR3/ ACCSW, ACFRAC(2)
      COMMON /DOBST/ OBCLR, OBBRK, TPPTO, IOBMOD, IOP, IVP,
     1       PASTOB, VIEWOB
      DIMENSION LIT1(3), LIT2(4), LIT3(8), NAME(4)
      DATA IBLNK /'0000'/
```

```
C
C     CROSSOVER MODEL DRIVER PARAMETER SECTION
C
  200 LIT3(4) = IBLNK
C
      DO 210 K = 1, 5
        READ (3,420) J, DGAIN(K)
  210 CALL DRORD(I, J, 1, NAME)
      READ (3,420) J,DGAIN(6)
      CALL DRORD(I,J,1,NAME)
      READ (3,420) J,DGAIN(7)
      CALL DRORD(I,J,1,NAME)
C
      READ (3,420) J, TAU
      CALL DRORD(I, J, 1, NAME)
      READ (3,420) J, VDES
      CALL DRORD(I, J, 1, NAME)
      CALL DRENDS(I, NAME)
      GO TO 60
C
C     PREVIEW-PREDICTOR MODEL DRIVER PARAMETER SECTION
C
```

Changes in subroutine DRIOUT

```
C*****************************************************************
C
C      SUBROUTINE DRIOUT
C
C      PURPOSE:
C      TO PRINT OUT DRIVER MODULE PARAMETERS
C
       SUBROUTINE DRIOUT(PRNTCN)
       IMPLICIT REAL(A - H,O - Z)
       INTEGER DRMODE, PRNTCN
       LOGICAL PASTOB, VIEWOB
       COMMON /DOBST/ OPCLR, OBBRK, TPPTO, IOBMOD, IOP, IVP,
      1       PASTOB, VIEWOB
       COMMON /DRDMAP/ RDPT(3,300), IPT1, NRDPT
       COMMON /DROLCM/ OLCOM(6,300), OLTIM(2,10), PPPARM(14),
      1       IOLCOM(300), NINT, NMAX, NOLP
       COMMON /DROPMD/ DRMODE
       COMMON /DRPAR1/ DRPR1(9), PE(12), WTAC(10), WTST(10),
      1       NPRED, NOPERR
       COMMON /DRPAR2/ DGAIN(7), TAU, VDES, DELINT, ACCINT, TLCR
       COMMON /DRPAR3/ ACCSW, ACFRAC(2)
       IF (PRNTCN .LT. 1) RETURN
       IF (DRMODE .GT. 0) GO TO 40
C      PRINT OUT OPEN-LOOP COMMAND PARAMETERS
       WRITE (1,270)
       WRITE (1,280)
       L = 0
    10 I = 1


       J = DRMODE - 10 * I
       IF (J .LE. 2) GO TO 210
       PRINT OUT DESCRIBING-FUNCTION(CROSSOVER) MODEL PARAMETERS
       IF (J .EQ. 3) WRITE (1,560)
       IF (J .EQ. 4) WRITE (1,570)
       WRITE (1,580) DGAIN(1), DGAIN(2)
       WRITE (1,590) DGAIN(3), DGAIN(4)
       WRITE (1,600) DGAIN(5), TAU
       WRITE (1,605) DGAIN(6)
       WRITE (1,606) DGAIN(7)
       IF (J .EQ. 3) GO TO 220
       WRITE (1,610) VDES
       RETURN
C      PRINT OUT PREVIEW PREDICTOR MODEL PARAMETERS
   210 IF (J .EQ. 1) WRITE (1,620)
       IF (J .EQ. 2) WRITE (1,630)
       WRITE (1,640) (DRPR1(K),K=1,5)
       WRITE (1,650) (DRPR1(K),K=6,9), NPRED
       WRITE (1,660) ACFRAC(1), ACFRAC(2), ACCSW


   570 FORMAT ('0', 'DRIVER PARAMETERS FOR THE STRAIGHT LINE '
      1       , 'CROSSOVER MODEL')
   580 FORMAT ('0', 'DRIVER GAIN ON Y    =', G12.5,
      1       ' ,    DRIVER ', 'GAIN ON V       =', G12.5)
   590 FORMAT (' ', 'DRIVER GAIN ON PSI =', G12.5,
      1       ' ,    DRIVER ', 'GAIN ON PSIDOT =', G12.5)
   600 FORMAT (' ', 'DRIVER GAIN ON U  =', G12.5,
      1       ' ,    DRIVER ', 'TIME LAG TAU    =', G12.5)
   605 FORMAT (' DRIVER GAIN ON STEERING ERROR INTEGRAL =',G12.5)
   606 FORMAT (' DRIVER GAIN ON VELOCITY ERROR INTEGRAL =',G12.5)
   610 FORMAT (' ', 'DESIRED VELOCITY   =', G12.5)
   620 FORMAT ('0', 'DRIVER PARAMETERS FOR THE PREVIEW-PRED',
      1       'ICTOR DRIVER MODEL USING THE GEOMETRIC PRED',
      2       'ICTOR')
```

Changes in subroutine DRINIT

```
      COMMON /DRINT1/ TLAST
      COMMON /DRINT2/ NPRINT
      COMMON /DRINT3/ ACCO, DELSWO, GP, TSTART
      COMMON /DROLCM/ OLCOM(6,300), OLTIM(2,10), PPPARM(14),
     1        IOLCOM(300), NINT, NMAX, NOLP
      COMMON /DROPMD/ DRMODE
      COMMON /DRMIX/ T2, JOLOLD
      COMMON /DRPAR1/ DRPR1(9), PE(12), WTAC(10), WTST(10),
     1        NPREC, NOPERR
      COMMON /DRPAR2/ DGAIN(7),TAU,VDES,DELINT,ACCINT,TLCR
      COMMON /DRPAR3/ ACCSW, ACFRAC(2)
      COMMON /DOBST/ CBCLR, OBBRK, TPPTO, IOBMOD, IOP, IVP,
     1        PASTOB, VIEWOB
      NINT = 1
C
C     CHECK FOR PURE OPEN-LOOP CONTROL
C
      IF (DRMODE .LE. 10) RETURN
      IPT1 = 1
      NPRINT = 0
      IF (DRMODE .EQ. 14) GO TO 20
      IF (DRMODE .EQ. 24) GO TO 20
C
C     CHECK FOR ERRORS IN ROAD-POINT DATA
C
      IF (NRDPT .LT. 2) GO TO 90
C
      DO 10 I = 2, NRDPT
        DIST2 = (RDPT(1,I - 1) - RDPT(1,I)) ** 2 + (RDPT(2,
     1  I - 1) - RDPT(2,I)) ** 2
        IF (DIST2 .LT. 1.E-08) GO TO 90
   10 CONTINUE
C
   20 CONTINUE
C     CHECK FOR NON-ZERO PERCEPTION ERROR
      NOPERR = 1
C
      DO 30 I = 1, 12
        IF (PE(I) .GE. 1.E-4) GO TO 40
   30 CONTINUE
C
      NOPERR = 0
      GO TO 50
C     INITIALIZE PERCEPTION-ERROR RANDOMIZATION SCHEME
   40 CALL DRAND1
C     OBTAIN NECESSARY VEHICLE PARAMETERS
   50 CALL DRVEH
C     DETERMINE CLOSED-LOOP DRIVER MODEL
      I = DRMODE / 10
      J = DRMODE - 10 * I
      IF (J .LE. 2) GO TO 60
C     INITIALIZE DESCRIBING-FUNCTION MODEL RESPONSE MATRIX
      NUSED = 2
      DELINT = 0.0
      ACCINT = 0.0
      TLCR = TSTART
      RES(1,1) = TSTART
      RES(2,1) = DELSWO
      RES(3,1) = ACCO
      RES(1,2) = TSTART + TAU - 1.E-04
```

```
C*****************************************************************************
C
C      SUBROUTINE DRCROV
C      DRCROV IS THE CONTROL SUBROUTINE FOR THE DESCRIBING FUNCTION MODEL
C
       SUBROUTINE DRCROV(ACC, DELSW, T, Y)
       IMPLICIT REAL(A - H,O - Z)
       INTEGER DRMODE
       REAL KD
       COMMON /DRDRMD/ DRMODE
       COMMON /DRPAR2/ DGAIN(7),TAU,VDES,DELINT,ACCINT,TLCR
       COMMON /DCRVEH/ A, B, GR, KD
       DIMENSION ERR(5), Y(6)
       DELNEW = 0.0
       IF (DRMODE .EQ. 13) GO TO 10
       IF (DRMODE .EQ. 23) GO TO 10
C      STRAIGHT LINE DESCRIBING FUNCTION ERROR COMPUTATION
       ERR(1) = -Y(2)
       ERR(2) = -Y(4) * SIN(Y(3)) - Y(5) * COS(Y(3))
       ERR(3) = -Y(3)
       ERR(4) = -Y(6)
       ERR(5) = VDES - Y(4)
       GO TO 20
C      GENERAL PATH DESCRIBING FUNCTION ERROR COMPUTATION
    10 CALL DCRERR(ERR, T, Y, RDCRV)
C      COMPUTE STEADY-TURNING STEER ANGLE
       RDES = Y(6) + ERR(4)
       VEL2 = Y(4)**2 + Y(5)**2
       DELNEW = (1.D+KD*VEL2)*(A+B)*RDES/SQRT(VEL2)/GR
C      ADD IN ERROR CORRECTIONS
    20 DELCOR = 0.0
       DO 30 I = 1, 4
    30 DELCOR = DGAIN(I) * ERR(I) + DELCOR
       DELNEW = DELNEW + DELCOR + DELINT*DGAIN(6)
C      COMPUTE STEERING ERROR INTEGRAL
       DELINT = DELINT + DELCOR*(T-TLCR)
C      COMPUTE ACCELERATION COMMAND
       ACCDES = DGAIN(5)*ERR(5)
       ACCNEW = ACCDES + DGAIN(7)*ACCINT
C      COMPUTE ACCELERATION INTEGRAL
       ACCINT = ACCINT + ACCDES*(T-TLCR)
C      UPDATE TLCR
       TLCR = T
       CALL DCRMF(ACC, ACCNEW, DELNEW, DELSW, T, TAU)
       RETURN
       END
C
```

```
C***********************************************************************
C
C      SUBROUTINE DCRERR
C      DCRERR CALCULATES THE FEEDBACK ERRORS FOR THE DESCRIBING-FUNCTION
C      MODEL
C
       SUBROUTINE DCRERR(ERR, T, Y, RDCRV)
       IMPLICIT REAL(A - H,O - Z)
       COMMON /DRPTER/ UINT, XINT, YINT, I2, IPT
       COMMON /DRDMAP/ RDPT(3,300), IPT1, NRDPT
       DIMENSION ERR(5), Y(6), YP(6,1)
C
       DO 10 I = 1, 6
    10 YP(I,1) = Y(I)
C      DETERMINE POSITION AND VELOCITY ERRORS FOR CURRENT POSITION
       CALL DPTERR(DIST, U2ERR, YP, 1, T)
C      ERR(1) IS THE LATERAL POSITION ERROR
       ERR(1) = -DIST
C      IPT AND I2 ARE THE INDICES OF THE ROAD POINTS
C      ON EITHER END OF THE NEAREST ROAD SEGMENT
       IF (IPT .LT. I2) I = IPT
       IF (IPT .GT. I2) I = I2
       J = I + 1
       DIST1 = SQRT((XINT-RDPT(1,I))**2 + (YINT-RDPT(2,I))**2)
       DISTOT = SQRT((RDPT(1,J) - RDPT(1,I))**2 + (RDPT(2,J)
      1- RDPT(2,I))**2)
C      COMPUTE DESIRED PATH ANGLE BY LINEAR INTERPOLATION
       PSIB = ATAN2((RDPT(2,J)-RDPT(2,I)),(RDPT(1,J)-RDPT(1,I)))
       IF (I .NE. 1) GO TO 20
       PSIA = PSIB
       GO TO 30
    20 K = I - 1
       PSIA = ATAN2((RDPT(2,I)-RDPT(2,K)),(RDPT(1,I)-RDPT(1,K)))
    30 IF (J .NE. NRDPT) GO TO 40
       PSIC = PSIB
       GO TO 50
    40 K = J + 1
       PSIC = ATAN2((RDPT(2,K)-RDPT(2,J)),RDPT(1,K)-RDPT(1,J)))
    50 PSI1 = (PSIA + PSIB)*0.5
       PSI2 = (PSIB + PSIC)*0.5
       PSIDES = PSI1 + DIST1*(PSI2 - PSI1)/DISTOT
C      ERR(2) IS THE LATERAL VELOCITY ERROR
       ERR(2) = -Y(4)*SIN(Y(3)-PSIDES) - Y(5)*COS(Y(3)-PSIDES)
       PI = 3.14159265359
    60 IF (Y(3) .LE. PI) GO TO 70
       Y(3) = Y(3) - 2.0*PI
       GO TO 60
    70 IF (Y(3) .GT. (-PI)) GO TO 80
       Y(3) = Y(3) + 2.0*PI
       GO TO 70
C      ERR(3) IS THE HEADING ANGLE ERROR
    80 ERR(3) = PSIDES - Y(3)
       IF (ERR(3) .GT. PI) ERR(3) = ERR(3) - 2.0*PI
       IF (ERR(3) .LT. (-PI)) ERR(3) = ERR(3) + 2.0*PI
C      COMPUTE ROAD CURVATURE AT POINT OF PERPENDICULAR INTERSECTION
C      BY INTERPOLATING BETWEEN CURVATURE AT SEGMENT ENDPOINTS
       IF (I .NE. 1) GO TO 90
       RDCRV1 = 0.0
       GO TO 100
    90 K = I - 1
       CALL DRCURV(RDPT(1,K), RDPT(2,K), RDPT(1,I),
      1     RDPT(2,I), RDPT(1,J), RDPT(2,J), RDCRV1, IFLG)
       IF (PSIDES .LT. PSI1) RDCRV1 = -RDCRV1
   100 IF (J .NE. NRDPT) GO TO 110
       RDCRV2 = 0.0
       GO TO 120
   110 K = J + 1
       CALL DRCURV(RDPT(1,I), RDPT(2,I), RDPT(1,J),
      1     RDPT(2,J),RDPT(1,K),RDPT(2,K), RDCRV2, IFLG)
       IF (PSI2 .LT. PSIDES) RDCRV2 = -RDCRV2
   120 RDCRV = RDCRV1 + DIST1 * (RDCRV2 - RDCRV1) / DISTOT
       VEL = SQRT(Y(4)**2 + Y(5)**2)
C      ERR(4) IS THE YAW-RATE ERROR
       ERR(4) = VEL * RDCRV - Y(6)
C      ERR(5) IS THE FORWARD VELOCITY ERROR
       ERR(5) = UINT - VEL
       RETURN
       END
C
```

4.3 Programming Changes Required to Interface Driver Module with The Three-Degree-of-Freedom Model

Four subroutines in the Driver Module are specific for the vehicle model being used, namely, DRFAC 1, DRFAC 2, (which includes an entry for DRFAC 3), DRFAC 4, and DRVEH. Versions of these subroutines have been written to interface the Driver Module with the three-degree-of-freedom vehicle model describe in Chapter 2. The Fortran code for these subroutines is listed on the following pages.

```
C ********************************************************************
C
C     SUBROUTINE DRFAC4
C     THIS SUBROUTINE COMPUTES DELTA AND ACC
C     FROM VEHICLE CONTROL VARIABLES
C
      SUBROUTINE DRFAC4(ACC, DELTA, DELSW, DOUT1, DOUT2,
     1                  DOUT3, DOUT4, JOLCOM)
      IMPLICIT REAL(A - H,O - Z)
      COMMON /DRVEH4/ BRKCON, DRVCON, GR, BRKTAB(2,11)
C     CONVERT STEERING WHEEL ANGLE TO FRONT WHEEL ANGLE
      DELTA = DELSW * GR
      IF (JOLCOM .EQ. 1) RETURN
C     CONVERT DRIVE TORQUE AND BRAKE LINE PRESSURE TO ACCELERATION
      DO 10 I = 2, 10
        IF (DOUT1 .LE. BRKTAB(1,I)) GO TO 20
   10 CONTINUE
C
      I = 11
   20 FRAC = (BRKTAB(1,I) - DOUT1) / (BRKTAB(1,I) - BRKTAB(
     11,I - 1))
      TOB = BRKTAB(2,I) - FRAC * (BRKTAB(2,I) - BRKTAB(2,I -
     1 1))
      ACC = TOB / BRKCON + DOUT2 / DRVCON
      RETURN
      END
C
C**********************************************************************
C
C     SUBROUTINE DRFAC1
C     DRFAC1 CONVERTS VEHICLE KINEMATIC VARIABLES TO DRIVER FORMAT
C
      SUBROUTINE DRFAC1(DY, DYVEH, Y, YVEH, DEL1, DEL2, JDR,
     1                  JOLCOM)
      IMPLICIT REAL(A - H,O - Z)
      DIMENSION DY(1), DYVEH(1), Y(1), YVEH(1)
      Y(1) = YVEH(1)
      Y(2) = YVEH(2)
      Y(3) = YVEH(3)
      Y(4) = YVEH(4)
      Y(5) = YVEH(5)
      Y(6) = YVEH(6)
      DY(1) = DYVEH(1)
      DY(2) = DYVEH(2)
      DY(3) = DYVEH(3)
      DY(4) = DYVEH(4)
      DY(5) = DYVEH(5)
      DY(6) = DYVEH(6)
      RETURN
      END
C
C**********************************************************************
```

```
C ****************************************************************
C
C       SUBROUTINE DREAC2
C       DREAC2 CONVERTS THE DRIVER OUTPUT TO VEHICLE FORMAT
C
        SUBROUTINE DREAC2(ACC, DELTA, DELSW, DOUT1, DOUT2,
      1              DOUT3, DOUT4, JOLCOM)
        IMPLICIT REAL(A - H,O - Z)
        COMMON /DRVEH4/ BRKCON, DRVCON, GR, BRKTAB(2,11)
        COMMON /FOUT/ FN(4),ALPHAT(4),TOT(4),FS(4),FC(4),S(4),DELT(2),
      1 TQST(2)
        COMMON /V3D/ CI(7,7),ALAMT(4),EC(5),XT(4),YT(4),DTOE(2),TAXL(2),
      1 AXLR,VC,G,ALAMRS,VM,VIZZ,VIYY,VIZX,XPL,YPL,ZPL,PLM,PLIZZ,PLIYZ,
      2 PLIZX,CST1,CST2,CST3,SR,XC(2),BRK(4),CD,PFA,RHOA,IACKER
C       JOLCOM .EQ. 8 MEANS PREDICTOR OPERATING
C       WHILE STILL IN TOTAL OPEN-LOOP CONTROL
        IF (JOLCOM .EQ. 8) RETURN
        IF (JOLCOM .EQ. 1) GO TO 10
C       COMPENSATE FOR FLEXIBILITY OF STEERING SYSTEM
        A1 = TQST(1)*CST1
        A2 = TQST(2)*CST2
        TEMP1 = SR*(A1+A2)/2.0
        TEMP2 = (TQST(1) + TQST(2))*CST3/SR
        DELSW = DELTA*SR - TEMP1 - TEMP2
     10 CONTINUE
C       BRAKE/DRIVE TORQUE OUTPUT COMMAND COMPUTATION
        ENTRY DREAC3(ACC,DOUT1,DOUT2,DOUT3,DOUT4,JOLCOM)
        IF (JOLCOM .NE. 5) DOUT3 = 0.0
        IF (JOLCOM .NE. 6) DOUT4 = 0.0
        IF (JOLCOM .EQ. 2) RETURN
        IF (JOLCOM .EQ. 3) GO TO 50
        IF (JOLCOM .EQ. 4) GO TO 80
        IF (ACC .GT. 0.0) GO TO 40
C       SPECIFY DOUT1 (BRAKELINE PRESSURE) BY INTERPOLATION
C       FROM BRAKE TABLE
        TQB = BRKCON * ACC
C
        DO 20 I = 2, 10
          IF (TQB .LE. BRKTAB(2,I)) GO TO 30
     20 CONTINUE
C
        I = 11
     30 FRAC = (BRKTAB(2,I) - TQB) / (BRKTAB(2,I) - BRKTAB(2,
      1I - 1))
        DOUT1 = BRKTAB(1,I) - FRAC * (BRKTAB(1,I) - BRKTAB(1,
      1I - 1))
        DOUT2 = 0.0
        RETURN
C       COMPUTE DOUT2 FOR CLOSED-LOOP MODES
     40 DOUT1 = 0.0
        DOUT2 = DRVCON * ACC
        RETURN
C       COMPUTE DOUT2 WHEN DOUT1 IS OPEN-LOOP CONTROLLED
     50 DO 60 I = 2, 10
          IF (DOUT1 .LE. BRKTAB(1,I)) GO TO 70
     60 CONTINUE
C
        I = 11
     70 FRAC = (BRKTAB(1,I) - DOUT1) / (BRKTAB(1,I) - BRKTAB(
      11,I - 1))
        TQB = BRKTAB(2,I) - FRAC * (BRKTAB(2,I) - BRKTAB(2,I -
      1 1))
        ACCPR = TQB / BRKCON
        ACCMOD = ACC - ACCPR
        DOUT2 = DRVCON * ACCMOD
        IF (DOUT2 .LT. 0.0) DOUT2 = 0.0
        RETURN
C       COMPUTE DOUT1 WHEN DOUT2 IS OPEN-LOOP CONTROLLED
     80 ACCPR = DOUT2 / DRVCON
        ACCMOD = ACC - ACCPR
        IF (ACCMOD .LT. 0.0) GO TO 90
        DOUT1 = 0.0
        RETURN
     90 TQB = BRKCON * ACCMOD
```

```
C
      DO 100 I = 2, 10
        IF (TOB .LE. BRKTAB(2,I)) GO TO 110
  100 CONTINUE
C
      I = 11
  110 FRAC = (BRKTAB(2,I) - TOB) / (BRKTAB(2,I) - BRKTAB(2,
     1I - 1))
      DOUT1 = BRKTAB(1,I) - FRAC * (BRKTAB(1,I) - BRKTAB(1,
     1I - 1))
      RETURN
      END
```

```
C*************************************************************************
C
C       SUBROUTINE DRVEH
C       DRVEH OBTAINS NECESSARY VEHICLE PARAMETERS FOR THE DRIVER MODULE
C       DRVEH IS SPECIFIC FOR THE TRANS35 3-DOF VEHICLE MODEL
C
        SUBROUTINE DRVEH
        IMPLICIT REAL(A-H,O-Z)
        INTEGER DRMODE,TMODE
        REAL KD
        LOGICAL FIRST
        COMMON /DCRVEH/ CRA, CRB, CRGR, CRKD
        COMMON /DRINT3/ ACCO, DELSWO, GR2, TSTART
        COMMON /DROPMD/ DRMODE
        COMMON /DRPAR3/ ACCSW, ACFRAC(2)
        COMMON /DRTDAT/ DRTR(24)
        COMMON /DRVEH1/ DRV1(3)
        COMMON /DRVEH2/ ACCMAX, ACCMIN, DELMAX
        COMMON /DRVEH3/ SIDACC
        COMMON /DRVEH4/ BRKCON, DRVCON, GR, BRKTAB(2,11)
        COMMON /DRVEH5/ ABM, AI, AM, C, DRG, TOE(2), XA(4),
      1         XW(4)
        COMMON /VPR/ DSWMAX,TQDMAX,PFLMAX,KD,DSWO,TQDO,PFLO
        COMMON /V3D/ CI(7,7),ALAMT(4),EC(5),XT(4),YT(4),DTOE(2),TAXL(2),
      1 AXLR,VC,G,ALAMRS,VM,VIZZ,VIYZ,VIZX,XPL,YPL,ZPL,PLM,PLIZZ,PLIYZ,
      2 PLIZX,CST1,CST2,CST3,SR,XC(2),BRK(4),CD,PFA,RHOA,IACKER
        COMMON /T3DATA/ FRO(4,10,2),  A0(4), A1(4), A2(4), B1(4), B3(4),
      1 B4(4), RT(4),  P0(4), P1(4), P2(4), S0(4), S1(4), S2(4),
      2 R0(4), R1(4), K1(4), K2(4), BC(4), SN(4), FRR(4)
        DIMENSION FR(4),FCMN(4),FSMX(4)
C
        A = XT(1)
        B = -XT(3)
        AI = VIZZ + PLIZZ + PLM*(XPL**2 + YPL**2)
        AM = VM+PLM
        ABM = AM/(A+B)
        C = (VC*VM + PLM*(VC - ZPL))/AM
        DELMAX = DSWMAX/SR
        DELSWO = DSWO
        DRG = G
C
C       ASSIGN TIRE PROPERTIES FOR 3DOF MODEL PREDICTOR
        DO 90 I=1,4
        DRTR(I) = A0(I)
        DRTR(I+4) = A1(I)
        DRTR(I+8) = A2(I)
        DRTR(I+12) = B1(I)
        DRTR(I+16) = B3(I)
     90 DRTR(I+20) = B4(I)
C    COMPUTE MAXIMUM CIRCUMFERENTIAL AND SIDE FORCES
        FCMNT = 0.0
        FSMXT = 0.0
        FR(1) = A*ABM
        FR(2) = B*ABM
        DO 70 I = 1, 4
          J = (I + 1) / 2
          FCMN(I) = -FR(J) * (P0(I) + FR(J)*(P1(I) + FR(J)*
      1   P2(I))) * SN(I)
          FCMNT = FCMNT + FCMN(I)
          FSMX(I) = FR(J) * (B3(I) + FR(J)*(B1(I) + FR(J)*B4(
      1   I))) * SN(I)
     70 FSMXT = FSMXT + FSMX(I)
C
```

```
C     COMPUTE MAXIMUM ALLOWABLE BRAKING ACCELERATION AND LATERAL ACCELERATION
         ACCMIN = ACFRAC(2) * FSMNT / AM
         SIDACC = FSMXT / AM
C     COMPUTE MAXIMUM ALLOWABLE ACCELERATION DUE TO DRIVE TORQUE
         DRVCON = AM/AXLR/(ALAMT(1)/RT(1) + ALAMT(2)/RT(2)
        1          + ALAMT(3)/RT(3) + ALAMT(4)/RT(4))
         ACCMAX = ACFRAC(1)*TODMAX*DRVCON
         DRV1(1) = A
         DRV1(2) = B
         DRV1(3) = KD
         TSTART = 0.0
         GR = 1.0/SR
         GR2 = GR
         DO 100 I=1,4
         XA(I) = XT(I)
  100 XW(I) = YT(I)
C
C     COMPUTE ENTRIES FOR DRIVER-MODEL BRAKE TABLE
         BRKCON = AM*(RT(1) + RT(2)+RT(3)+RT(4))*0.25
         BRKTAB(1,1) = 0.0
         BRKTAB(2,1) = 0.0
         BRKTAB(1,2) = PFLMAX + 1.0
         BRKTAB(2,2) = BRKTAB(1,2)*(BRK(1)+BRK(2)+BRK(3)+BRK(4))*0.25
C
         ACCO = TODO/DRVCON - PFLO/(PFLMAX+1.0)*BRKTAB(2,2)/BRKCON
         TOE(1) = DTOE(1)
         TOE(2) = DTOE(2)
         CRA = A
         CRB = B
         CRGR = GR
         CRKD = KD
         RETURN
         END
```

List Of Symbols For Chapter 4

Symbol	Definition
GR	Steering gear box ratio
K'	Driver trim integrator gain
K_Y	Factor converting position error to heading angle error
K_ψ	Driver gain on heading angle error
KD	Understeer factor for steady turning
L	Vehicle wheelbase
N_{RDPT}	Number of road points
T_{eq}	Equivalent yaw rate time constant
u	Velocity component along the x-axis
Y_{err}	Distance from vehicle position to desired path
X_{RPi}, Y_{RPi}	Coordinates of the road point i
δ_{SW}	Steering wheel angle
ψ_A, ψ_B, ψ_C	Intermediate variables defined by eqs. (4.5), (4.6), (4.7)

Symbol	Definition
ψ_{des}	Tangent angle to the desired path at the closest point on the road path
ψ_{err}	Difference between current and desired heading angle
ψ_i, ψ_j	Average path angles
κ	Curvature of the desired path
τ	Driver time delay

REFERENCES FOR CHAPTER 4

4.1 "Improvement of Mathematical Models for Simulation of Vehicle Handling, Vol. 7. Technical Manual for the General Simulation", W. R. Garrott and R. A. Scott. Final Report DOT-HS-7-01715, March 1980.

4.2 "Closed-Loop Automobile Maneuvers Using Describing Function Methods", W. R. Garrott, D. L. Wilson and R. A. Scott. SAE Paper, No. 820305, Feb. 1982.

4.3 "Stability and Performance Analysis of Automobile Driver Steering Control," R. W. Allen, SAE Paper No. 820303, Feb. 1982.