Lateral Dynamics of the ITNS Vehicle

J. E. Anderson

Contents

1. Introduction

The subject of this paper is the derivation of the equations that define the lateral motion of an ***ITNS*** vehicle, a sketch of which is shown in Figure 6. For this analysis, the vehicle passes through a diverge section of guideway. The solution for lateral motion of the vehicle permits us to determine the stiffness of the lateral tires required for acceptable ride comfort, the forces on the wheels, and the required length of flared switch rails. Because there is little coupling between pitch motion and lateral motion and because in this analysis it is assumed that the running surfaces are smooth, we can treat three-dimensional lateral motion separately from three-dimensional pitch motion. However, in Appendix G[1], we analyze the most severe pitch motion, which results in a pitch angle of about 0.2^0 and 70% of the maximum weight on the rear wheels. The lateral degrees of freedom are yaw ψ, roll ϕ, and sidewise motion y_{mc}, i.e. the sidewise motion of the mass center of the vehicle. It will be assumed that the vehicle passes through the diverge section at constant speed V.

2. The Equations of Motion

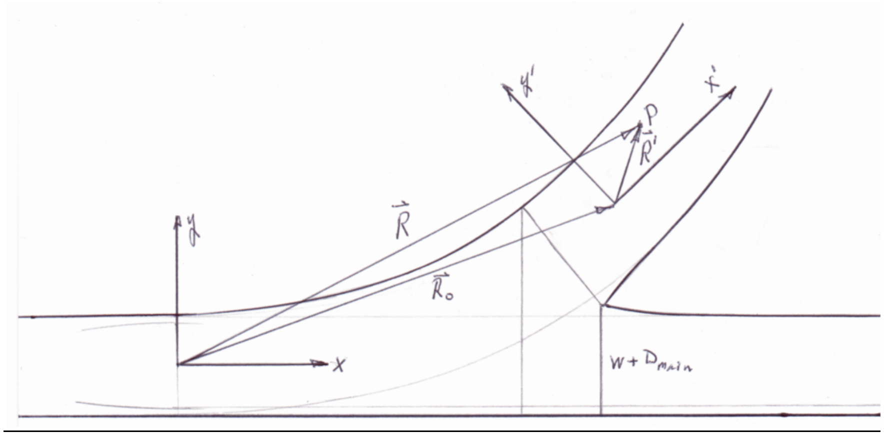

Figure 1. The Geometry of a Guideway Diverge.

Figure 1 and Appendix A define a fixed reference system x, y, z in which x and y are assumed to be in a horizontal plane, and a reference system x', y', z' centered in the guideway but moving with the center of mass of the vehicle, with both $x'$and y' in the same horizontal plane. The x' axis points in the local direction of the center line of the guideway, the y and y' axes are horizontal and point to the left, and the z, z' axis are vertical and points upward, giving three orthogonal axes consistent with the right-hand rule. The sidewise acceleration of the mass center of the vehicle (in the y' direction) is equal to the sum of the sidewise forces divided by the mass of the vehicle. The yaw acceleration $\ddot{\psi}$ of the vehicle is the sum of the moments about the z' axis at the mass center divided by the yaw moment of inertia of the vehicle about the z' axis. Similarly the roll acceleration $\ddot{\phi}$ is the sum of the moments about the x' axis divided by the roll moment of inertia of the vehicle about the x'axis. Thus, from Appendix A, we have

$$\frac{W_{vehicle}}{g}\left(\ddot{y}_{mc}+\frac{V^2}{R}\right)=F_{ufr}+F_{ufl}+F_{ubr}+F_{ubl}+F_{lfr}+F_{lfl}+F_{lbr}+F_{lbl}$$
$$+F_{sfr}+F_{sfl}+F_{sbr}+F_{sbl}+F_{wind}$$

[1] No Appendix G in the paper.

(2-1)

in which $\ddot{y}_{mc}$ is the vehicle acceleration in the y' direction, *V* is the constant vehicle speed, $\dot{\Psi}$ is time rate of change of the direction of the guideway, $W_{vehicle}$ is the weight of the vehicle, and *g* is the acceleration of gravity. The first eight forces shown in equation (2-1) are applied to the eight lateral support tires by the lateral running surfaces, the next four forces, designated by *s* as the first subscript, are the forces applied to the switch wheels by the switch rails, and the remaining force is applied to the cabin by a side wind. If a tire force would be calculated to be negative, it will be set to zero. Of the first eight forces, the first subscript *u* or *l* designates upper or lower side wheels. For the next four forces, the first subscript, *s*, designates a switch wheel. The second subscript *f* or *b* in all cases designates a front or back wheel. The third subscript *r* or *l* in all cases designates a right or left wheel.

The wind force is given by

$$F_{wind} = \frac{d_{air}}{2g} V_{wind}^2 C_D A$$

(2-2)

in which $d_{air} = 0.075$ lb weight/ft^3, $g = 32.174$ ft/sec^2, V_{wind} is the maximum wind speed in ft/sec, C_D is the dimensionless side drag coefficient, and *A* is the side area of the cabin in square feet, giving a wind force in pounds.

The two equations for the moments about the center of mass are

$$\frac{W_{vehicle}}{g} R_\psi^2 \ddot{\psi} = \sum Yaw\ moments, \quad \frac{W_{vehicle}}{g} R_\phi^2 \ddot{\phi} = \sum Roll\ moments$$

(2-3)

in which R_ψ, R_ϕ are the radii of gyration of the vehicle about the $z'$and x'axes, respectively.

3. The Orientation of the Vehicle with respect to the Guideway.

In Appendix A we defined a reference frame x', y', z' with $x' = 0$ at and moving at constant speed *V* with the center of mass of the vehicle. This reference frame is centered in the guideway so that x' is parallel to the guideway, $y' = 0$ at the center of the guideway and directed perpendicular to the guideway, positive to the left, and $z' = 0$ at the vertical position of the center of mass, positive upward. Because the position of the center of mass is only approximately known and will vary as the design proceeds, we establish a set of body axes x_b, y_b, z_b such that $x_b = 0$ at the center of the rear axle of the rear main-support wheels, $y_b = 0$ at the center of the chassis, and $z_b = 0$ at the center of the rear axle of the main-support wheels. The x_b axis points along the length of the chassis, y_b is to the left, and z_b is vertically upward parallel to the vertical chassis. The lateral motion of the vehicle with respect to the reference frame x', y', z' will be described by the lateral deflection y_{mc}, a yaw angle ψ between the coordinates x_b and x', and a roll angle

ϕ between the coordinates z_b and z'. Each of the angles is positive according to the right-hand rule. Thus the position of the mass center is $y_{mc}\hat{j}'$, its position in body coordinates is $X_{mc}\hat{\imath}_b + Z_{mc}\hat{k}_b$, and the position of any point on the chassis in body coordinates is $x_w\hat{\imath}_b + y_w\hat{j}_b + z_w\hat{k}_b$.

The *x, y, z* reference frame defined in Figure 1 and Appendix A is taken to be an inertial, i.e., fixed, reference frame. We need to know the position of any tire contact point in the vehicle in this fixed reference frame and the acceleration of the center of mass of the vehicle in the y'direction. From Figure 1 the vector distance from the origin of the *x, y, z* reference frame to a point on the vehicle is

$$\vec{R} = \vec{R}_0 + \vec{R}' = x(s)\hat{\imath} + y(s)\hat{j} + z(s)\hat{k} + y_{mc}\hat{j}' - X_{mc}\hat{\imath}_b - Z_{mc}\hat{k}_b + x_w\hat{\imath}_b + y_w\hat{j}_b + z_w\hat{k}_b \tag{3-1}$$

in which $x(s), y(s), z(s)$ are the coordinates of the origin of reference frame x', y', z' with respect to the fixed reference frame, y_{mc} is the lateral displacement of the mass center from the center of the guideway, X_{mc}, Z_{mc} are the coordinates of the mass center from the origin of the body coordinates, and x_w, y_w, z_w are the coordinates of the point of contact of an <u>undeflected</u> wheel contact point in body coordinates. We need to express all of these unit vectors in terms of the space-fixed unit vectors.

To do so, define the angle between the x and x' axes as Ψ, with Ψ positive if the x' axis has turned counterclockwise, as shown in Figure 1. Then, in matrix form, the angular orientation of the x', y', z' frame with respect to the x, y, z frame is

$$\begin{vmatrix} \hat{\imath}' \\ \hat{j}' \\ \hat{k}' \end{vmatrix} = \begin{bmatrix} C\Psi & S\Psi & 0 \\ -S\Psi & C\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{vmatrix} \hat{\imath} \\ \hat{j} \\ \hat{k} \end{vmatrix} \tag{3-2}$$

in which $C \equiv cos,\ S \equiv sin.$

To reach the body axes, first define an intermediate set of axes x_1, y_1, z_1, which rotate about the vertical through the yaw angle ψ, which is positive for yaw to the left. In terms of the corresponding intermediate set of unit vectors $\hat{\imath}_1, \hat{j}_1, \hat{k}_1$ the angular orientation of these unit vectors to the reference frame to the x', y', z' is

$$\begin{vmatrix} \hat{\imath}_1 \\ \hat{j}_1 \\ \hat{k}_1 \end{vmatrix} = \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{vmatrix} \hat{\imath}' \\ \hat{j}' \\ \hat{k}' \end{vmatrix}$$

Finally, rotate the $\hat{\imath}_1, \hat{\jmath}_1, \hat{k}_1$ unit vectors about the common $\hat{\imath}_1, \hat{\imath}_b$ axes to the right (positive with the right-hand rule) through the roll angle ϕ to reach the body axes. Thus

$$\begin{vmatrix} \hat{\imath}_b \\ \hat{\jmath}_b \\ \hat{k}_b \end{vmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{bmatrix} \begin{vmatrix} \hat{\imath}_1 \\ \hat{\jmath}_1 \\ \hat{k}_1 \end{vmatrix}$$

Then, by matrix multiplication,

$$\begin{vmatrix} \hat{\imath}_b \\ \hat{\jmath}_b \\ \hat{k}_b \end{vmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & S\phi \\ 0 & -S\phi & C\phi \end{bmatrix} \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{vmatrix} \hat{\imath}' \\ \hat{\jmath}' \\ \hat{k}' \end{vmatrix} = \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi C\phi & C\psi C\phi & S\phi \\ S\psi S\phi & -C\psi S\phi & C\phi \end{bmatrix} \begin{vmatrix} \hat{\imath}' \\ \hat{\jmath}' \\ \hat{k}' \end{vmatrix} \tag{3-3}$$

Then, using well-known trigonometric identities,

$$\begin{aligned} \begin{vmatrix} \hat{\imath}_b \\ \hat{\jmath}_b \\ \hat{k}_b \end{vmatrix} &= \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi C\phi & C\psi C\phi & S\phi \\ S\psi S\phi & -C\psi S\phi & C\phi \end{bmatrix} \begin{bmatrix} C\Psi & S\Psi & 0 \\ -S\Psi & C\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{vmatrix} \hat{\imath} \\ \hat{\jmath} \\ \hat{k} \end{vmatrix} \\ &= \begin{bmatrix} C(\Psi+\psi) & S(\Psi+\psi) & 0 \\ -S(\Psi+\psi)C\phi & C(\Psi+\psi)C\phi & S\phi \\ S(\Psi+\psi)S\phi & -C(\Psi+\psi)S\phi & C\phi \end{bmatrix} \begin{vmatrix} \hat{\imath} \\ \hat{\jmath} \\ \hat{k} \end{vmatrix} \end{aligned} \tag{3-4}$$

From equations (3-1), (3-2) and (3-4) the x and y coordinates of a wheel-contact point are

$$\begin{aligned} x &= x(s) - S\Psi y_{\mathrm{mc}} + C(\Psi+\psi)(x_w - X_{mc}) - S(\Psi+\psi)C\phi y_W + S(\Psi+\psi)S\phi(z_w - Z_{mc}) \\ y &= y(s) + C\Psi y_{\mathrm{mc}} + S(\Psi+\psi)(x_w - X_{mc}) + C(\Psi+\psi)C\phi y_W - C(\Psi+\psi)S\phi(z_w - Z_{mc}) \end{aligned} \tag{3-5}$$

4. Moments and Body Coordinates

The positions of application of the tire forces with respect to the body axes are shown in Figure 6. The side-wheel forces on the right side are either positive or zero, and on left side either negative or zero. The switch-wheel forces on the right side are either negative or zero, and on left side either positive or zero. The moments are as follows:

$$\begin{aligned} \sum Yaw\ Moments = &\left(F_{ufr} + F_{ufl}\right)\left(X_{uf} - X_{cg}\right) - \left(F_{ubr} + F_{ubl}\right)\left(X_{cg}\right) \\ &+ \left(F_{lfr} + F_{lfl}\right)\left(X_{lf} - X_{cg}\right) - \left(F_{lbr} + F_{lbl}\right)\left(X_{cg} - X_{lb}\right) \\ &- \left(F_{sfr} + F_{sfl}\right)\left(X_{sf} - X_{cg}\right) - \left(F_{sbr} + F_{sbl}\right)\left(X_{cg} - X_{sb}\right) + F_{wind}\left(X_{wind} - X_{cg}\right) \end{aligned} \tag{4-1}$$

$$\sum Roll\ moments = \left(F_{ufr} + F_{ufl} + F_{ubr} + F_{ubl}\right)\left(Z_{cg} - Z_u\right)$$

$$+\left(F_{lfr}+F_{lfl}+F_{lbr}+F_{lbl}\right)\left(Z_{cg}-Z_{l}\right)$$
$$-\left(F_{sfr}+F_{sfl}+F_{sbr}+F_{sbl}\right)\left(Z_{cg}-Z_{s}\right)-F_{wind}\left(Z_{wind}-Z_{cg}\right)-W_{p}Y_{p}$$
$$+\frac{1}{2}D_{mainwheels}\left(F_{main_L}-F_{main_R}\right)$$

(4-2)

in which $F_{wind} > 0$ if it blows to the left , X_{wind} and Z_{wind} define the position of application of its centroid, W_p is the weight of a passenger, Y_p is the sidewise displacement of the passenger, and $D_{mainwheels}$ is the distance between the centerlines of the right and left main support tires.

The coordinates of application of the forces are tentatively given in Table 1, in which the distances, which are shown in Figure 6, are given in the above-defined body reference frame. These values will likely change somewhat as the design proceeds.

Table 1. Coordinates of Application Points of the Forces, inches

u = upper wheel	f = front wheel	r = right wheel	s = switch wheel
l = lower wheel	b = back wheel	l = left wheel	
Position of cg	Xcg=41	Ycg=0	Zcg=26
Wind, Passenger	X_{wind}=41	Yp=20	Z_{wind}=57
Wheels			
Ufr	Xuf=82	Yr=-10.5	Zu=21
Ufl	Xuf=82	Yl=10.5	Zu=21
Ubr	Xub=0	Yr=-10.5	Zu=21
Ubl	Xub=0	Yl=10.5	Zu=21
Lfr	Xlf=68	Yr=-10.5	Zl=-4.2
Lfl	Xlf=68	Yr=10.5	Zl=-4.2
Lbr	Xlb=12	Yr=-10.5	Zl=-4.2
Lbl	Xlb=12	Yl=10.5	Zl=-4.2
Sfr	Xsf=72	Ysr=-6.6	Zs=8.4
Sfl	Xsf=72	Ysl=6.6	Zs=8.4
Sbr	Xsb=10	Ysr=-6.6	Zs=8.4
Sbl	Xsb=10	Ysl=6.6	Zs=8.4
LIM_f	X_{LIMf}=66	Y_{LIMf}=0	Z_{LIMf}=-6.6
LIM_b	X_{LIMb}=6	Y_{LIMb}=0	Z_{LIMb}=-6.6

5. Equations of the Center of the Curved Guideway.

Consider Figure 1. The position of an ***ITNS*** vehicle as it moves along the guideway is defined by the equation $s = Vt$, where V is the constant speed of the vehicle and t is time; however, we begin the simulation at a negative value of s in order to permit the motion to settle before the diverge point is reached. We define $s = 0$ at a point $x = 0$ in Figure 1 where the guideway begins to curve. To find the side-tire deflections, we need the x and y coordinates of each of the running surfaces for a given value of s. This calculation begins with the equations of the center of the curved guideway. We can derive the *x(s)*, *y*(*s*) curve for the centerline of the curved guideway from *Transit Systems Theory*, Chapter 3. For *s* < 0 this is a straight line at $y = 0$. At *s* = 0 it begins to curve to the left first at a constant rate of change of curvature, and then when the lateral acceleration has reached the comfort level a_n at constant curvature. The guideway at any point makes an angle Ψ with the *x* axis. In the constant-rate-of-change-of-curvature region the solution is

$$\frac{d^2\Psi}{ds^2} = \frac{J_n}{V^3}, \qquad \frac{d\Psi}{ds} = \frac{J_n s}{V^3} = \frac{1}{R} = \frac{a}{V^2}, \qquad s = V\frac{a}{J_n}, \qquad \Psi = \frac{J_n s^2}{2V^3}$$

(5-1)

in which J_n is the comfort level of lateral jerk, *V* is the speed along the curved guideway, *R* is the radius of curvature, and *a* is lateral acceleration. When *a* reaches the maximum comfort value a_n *s* reaches a point we call point 1 where

$$s = s_1 = V\frac{a_n}{J_n} \text{ and } \Psi = \Psi_1 = \frac{a_n^2}{2VJ_n}$$

(5-2)

If for example *V* = 30 mph or 44 ft/sec, J_n = 0.25 g/sec, $a = a_n$ = 0.2g and g = 32.174 ft/sec^2, then $s_1 = 44\frac{0.2}{0.25} = 35.2$ ft and $\Psi_1 = \frac{(0.2g)^2}{88(0.25g)} = 0.0585$ radians or 3.35 degrees.

The coordinates of the guideway centerline between *s* = 0 and $s = s_1$ are

$$\frac{dx}{ds} = cos\Psi, \qquad x \cong \int_0^s \left(1 - \frac{\Psi^2}{2}\right) ds = s\left(1 - \frac{1}{10}\Psi^2\right)$$

$$\frac{dy}{ds} = \ sin\Psi, \qquad y \cong \int_0^s \left(\Psi - \frac{\Psi^3}{6}\right) ds = \frac{s\Psi}{3}\left(1 - \frac{1}{14}\Psi^2\right)$$

(5-3)

At $s = s_1$we have, for the values chosen above, $x_1 = 35.2(1 - 0.0003)$ ft, and $y_1 = \frac{35.2(0.0585)}{3}(1 - 0.0002) = 0.686$ ft = 8.24 in. No te from Table 1 that the horizontal distance from the guideway centerline to the side running surface is 10.5 in, which is greater than y_1, a fact that is needed in Section 10.

For $s > s_1$ the curvature is constant at

$$\frac{d\Psi}{ds} = \frac{1}{R} = \frac{a_n}{V^2}$$

(5-4)

The coordinates of the center of curvature are

$$x_c = x_1 - Rsin\Psi_1, \qquad y_c = y_1 + Rcos\Psi_1$$

(5-5)

and the coordinates of any point in the constant curvature region are

$$x = x_c + Rsin\Psi, \qquad y = y_c - Rcos\Psi$$

(5-6)

in which

$$\Psi = \Psi_1 + \frac{s - s_1}{R}$$

(5-7)

and

$$s_1 = V\frac{a_n}{J_n}, \qquad R = \frac{V^2}{a_n}, \qquad \Psi_1 = \frac{a_n^2}{2VJ_n}, \qquad x_1 = s_1\left(1 - \frac{\Psi_1^2}{10}\right), \qquad y_1 = \frac{a_n^3}{6J_n^2}\left(1 - \frac{\Psi_1^2}{14}\right)$$

(5-8)

Summarizing,

$$\text{if } 0 < s < s_1 \quad \Psi = \frac{J_n s^2}{2V^3}, \qquad x = s\left(1 - \frac{1}{10}\Psi^2\right), \qquad y = \frac{s\Psi}{3}\left(1 - \frac{1}{14}\Psi^2\right)$$

$$\text{if } s \geq s_1 \quad \Psi = \Psi_1 + \frac{s - s_1}{R}, \qquad x = x_c + Rsin\Psi, \qquad y = y_c - Rcos\Psi$$

(5-9)

6. Equations of the Outer Running Surfaces

Facing the direction of motion of the vehicle, which is to the right in Figure 1, call the "outer running surfaces" the extreme left and right surfaces. Call the "inner running surface" the intermediate left and right surfaces after a vehicle has diverged, i.e., the right running surface of the left segment of guideway and the left running surface of the right segment of guideway. The side wheels run against these surfaces. Facing the direction of motion, the right outer running surface is parallel to the *x*-axis and at the position $y = -0.5w,$ where *w* is the distance between the left and right main running surfaces. The left outer running surface is found by adding $\frac{0.5w}{\cos\Psi}$ to *y* in equations (5-9). As mentioned, from Table 1 0.5*w* = 10.5 in.

7. Equations of the Switch Rail Running Surfaces

The switch rails are present from $s = -L_{swx}$ to $s = s_{start} + L_{main} + L_{swx}$, in which L_{swx} is the length of the flare section of the switch rails, L_{main} is the length of the main rail flared section at the diverge point (see Figure 1), and s_{start}, derived in Section 8, is the value of arc-length *s* along the curved guideway of Figure 1 at the point where the inner running surfaces start. Based on the analysis given in Appendix B, to account for variations in the positions of the switch wheels due to external forces the switch rails must be flared at the entry and exit points according to a cubic equation, which is a section of constant rate of change of curvature. Let D_{swx} be the lateral distance the initial end of the flared section lies from a point where it would be if there were no flare. Assume the flared section ends when $s = 0$, which is the point at which the constant rate of change of curvature of the left running surface starts.

If $-L_{swx} \leq s \leq 0$, the equation of the left switch rail is then

$$y_{swx_{left}} = 0.5w - w_{swx} + D_{swx}\left(\frac{s}{L_{swx}}\right)^3 \tag{7-1}$$

in which $w_{swx} = 4.5$ inch is the gap between the main left running surface and the switch-rail running surface at $s = 0$. When $s > 0$ y differs from the main left running surface only in that $0.5w/cos\Psi$ must be replaced by $(0.5w - w_{swx})/cos\Psi$.

Using the notation of Section 8, when $s_{start} + L_{main} \leq s \leq s_{start} + L_{main} + L_{swx}$ the equation of the downstream end of the left switch rail is

$$y_{swx_{left}} = \left[0.5w - w_{swx} - D_{swx}\left(\frac{s - s_{start} - L_{main}}{L_{swx}}\right)^3\right]/cos\Psi$$

Similarly, if – $L_{swx} \leq s \leq 0$ the equation of the right switch rail is

$$y_{swx_{right}}(x) = -0.5w + w_{swx} - D_{swx}\left(\frac{s}{L_{swx}}\right)^3$$

if $0 \leq s < s_{start} + L_{main}$

$$y_{swx_{right}} = -0.5w + w_{swx}$$

and if $s_{start} + L_{main} \leq s \leq s_{start} + L_{main} + L_{swx}$

$$y_{swx_{right}}(x) = -0.5w + w_{swx} + D_{swx}\left(\frac{s - s_{start} - L_{main}}{L_{swx}}\right)^3 \tag{7-2}$$

8. **Equations of the Inner Running Surfaces.**

The flared inner running surfaces are shown in Figure 1. Let x_{start} be the x-coordinate of the point at which these running surfaces start. The y-distance from the right outer running surface to the point where the two flared surfaces intersect, i.e., where the flared inner running surfaces start, is $w + D_{main}$, where D_{main} is the lateral distance between the end of the flare and the right inner running surface after the flare ends. Appendix E shows that the y-distance at x_{start} from the right outer running surface to the point of intersection to the left branch's right running surface if there were no flare is $w + D_{main}\ (1 + 1/cos\Psi)$. This value is also given by the right-most equation in equations (5-9). Thus

$$w + D_{main}(1 + 1/cos\Psi_{start}) = y_c - Rcos\Psi_{start} \tag{8-1}$$

in which the subscript *"start"* designates the value of Ψ at the start of the inner running surfaces. Multiply equation (8-1) by $cos\Psi_{start}$ and rearranged in the standard form of a quadratic equation. Then

$$Rcos^2\Psi_{start} - \mathrm{Q}\cos\Psi_{start} + D_{main} = 0$$

in which $Q = y_c - w - D_{main}$. Thus

$$cos\Psi_{start} = \frac{1}{2\mathrm{R}}\left[Q \pm \sqrt{Q^2 - 4RD_{main}}\right]$$

As $D_{main} \to 0$ equation (8-1) shows that $cos\Psi_{start}$ goes to $\frac{y_c - w}{R}$, which corresponds to the + sign, which is therefore the correct one. The value of $cos\Psi$ at the start of the inner surfaces is therefore

$$cos\Psi_{start} = \frac{1}{2\mathrm{R}}\left[\mathrm{Q} + \sqrt{Q^2 - 4RD_{main}}\right] \tag{8-2}$$

From equations (5-9), the arc length at the starting point is

$$s_{start} = s_1 + R[cos^{-1}(cos\Psi_{start}) - \Psi_1] \tag{8-3}$$

Also from equations (5-9) and a well-known trigonometric identity the value of *x* at the start point is,

$$x_{start} = x_c + R\sqrt{1 - cos^2\Psi_{start}} \tag{8-4}$$

The equation for the main flare on the inside running surfaces is

$$y_{flare}(s) = D_{main}\left(\frac{L_{main} + s_{start} - s}{L_{main}}\right)^3 \text{ if } s_{start} \le s \le s_{start} + L_{main} \text{else } y_{flare} = 0 \tag{8-5}$$

in which L_{main} is the length of the flared section. For the left running surface of the right branch, the sign of y_{flare} is positive and $\Psi = 0$.

The equation for the left running surface of the right branch of the diverge is

$$y(s) = 0.5w + y_{flare}(s) \tag{8-6}$$

The equations for the right running surface of the left branch of the diverge are, from equations (5-9), starting at $s = s_{start} > s_1$,

$$\Psi = \Psi_1 + \frac{s - s_1}{R}\,, \qquad x(s) = x_c + Rsin\Psi$$
$$y(s) = y_c - Rcos\Psi - \left(0.5w + y_{flare}(s)\right)/cos\Psi \tag{8-7}$$

in which R and the values with subscript "1" are defined by equations (5-8).

9. Tire Force-Deflection Relationships for the Tires

In Appendix C we show that the force on each of our solid polyurethane side tires is proportional to the 1.5 power of the deflection, but the force on the pneumatic main-support tires is proportional to the first power of the deflection. In Appendix D we derive a simple relationship for the energy loss as tire first compresses and then decompresses. We use this relationship in the program of Appendix E.

10. The Deflection of the Side Tires

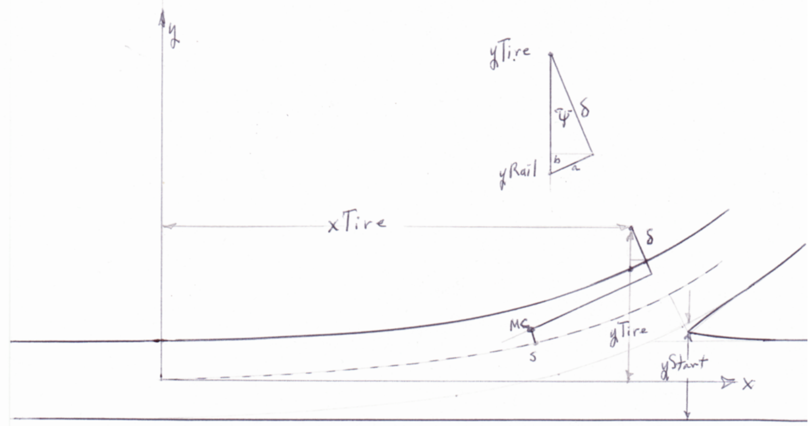

Figure 2. The Geometry of a Tire Deflection.

10.1 Deflection of the side tires running against the outer main running surfaces.

The s-position of the mass center of the vehicle is at the point marked "s" in Figure 2. We designate the coordinates of this point as x(s), $y(s)$. At this point, the guideway centerline for the **left** outer main running surface lies at an angle Ψ with the x-axis. The chassis mass center is displace a distance $y_{mc}\hat{\jmath}'$ and lies at a small angle ψ with respect to the guideway centerline. Using equations (3-5) the x and y coordinates of a tire-contact point are

$$xTire = x(s) - S\Psi y_{mc} + C(\Psi+\psi)(x_w - X_{mc}) - S(\Psi+\psi)[C\phi y_w - S\phi(z_w - Z_{mc})]$$
$$yTire = y(s) + C\Psi \mathrm{y_{mc}} + S(\Psi+\psi)(x_w - X_{mc}) + C(\Psi+\psi)[C\phi y_w - S\phi(z_w - Z_{mc})]$$
(10.1-1)

in which x_w, y_w, z_w are the coordinates of a tire contact point measured in body axes, obtained from Table 1.

With $y = 0$ at $s = 0$ for the guideway-center curve, we have from Section 6 and equations (5-9) the equations of the left outer running surface:

if $s \leq 0$ $y = 0.5w$

If $0 < s < s_1$

$$\Psi = \frac{J_n s^2}{2V^3}, \qquad x = s\left(1 - \frac{1}{10}\Psi^2\right), \qquad y = 0.5w/cos\Psi + \frac{s\Psi}{3}\left(1 - \frac{1}{14}\Psi^2\right)$$
(10.1-2)

If $s \geq s_1$

$$\Psi = \Psi_1 + \frac{s - s_1}{R}, \qquad x = x_c + Rsin\Psi, \qquad y = 0.5w/cos\Psi + y_c - Rcos\Psi$$
(10.1-3)

After calculating *xTire* and *yTire* from equations (10.1-1), we need to find the correspond value of *y* = *yRail* for the left-outer running surface. Having this value, we see from Figure 2 that the deflection of the front-left tire is

$$\delta = (yTire - yRail)cos\Psi$$
(10.1-4)

in which we must determine both *yRail* and $cos\Psi$.

If $xTire \leq 0$ then $cos\Psi = 1$ and $yRail = 0.5w$.

If $0 < xTire < x_1$ the needed values of Ψ and *yRail* come from equations (10.1-2). To find these values, we need to solve the equation for x(s) inversely for *s*, which would require solution of a fifth-order polynomial. Unfortunately, there is no known exact solution for such an equation. Hence, we must turn to a numerical solution. Let the first guess be $s = s_{1st} = x_{1st}$ and the second guess $s_{2nd} = s_{1st} + ds$, where *ds* is a given small value, from the numerical work after equation (5-3) about 0.01 ft. The value of *x* calculated from s_{2nd} is x_{2nd}. Then, since the difference between *x* and *s* is very small, we take as the correct value of *s* as

$$s = s_{2nd} + ds\left(\frac{xTire - x_{2nd}}{x_{2nd} - x_{1st}}\right)$$

(10.1-5)

With this value of s, we have from equations (10.1-2)

$$\Psi = \frac{J_n s^2}{2V^3} \text{ and } yRail = 0.5w/cos\Psi + \frac{s\Psi}{3}\left(1 - \frac{1}{14}\Psi^2\right)$$

(10.1-6)

If $xTire \geq x_1$ we have an exact solution. From the second of equations (10.1-3) we calculate

$$sin\Psi = \frac{xTire - x_c}{R}$$

(10.1-7)

Then by using a well-known trigonometric identity we can calculate

$$cos\Psi = \sqrt{1 - sin^2\Psi}$$

(10.1-9)

Then, from the third of equations (10.1-3),

$$yRail = 0.5w/cos\Psi + y_c - Rcos\Psi$$

(10.1-10)

For the right outer running surface, $x(s) = s, y(s) = -0.5w$. With these values, equation (10.1-1) with $\Psi = 0$ gives *xTire, yTire*. The tire deflection is then

$$\delta = -yTire - 0.5w \; if > 0 \; else \; 0. \qquad (10.1\text{-}11)$$

10.2 <u>Deflection of the switch tires running against the switch surfaces.</u>

The switch rails run from $s = -L_{swx}$ to $s = s_{start} + L_{main} + L_{swx}$, in which these lengths are defined in Sections 7 and 8. To find the position of the **left** switch tires, note that in equations (10.1-1) for the switch wheels on the **left** side, substitute for x_w, y_w, z_w from Table 1 the values *Xsf/b – Xcg, Ysl, Zs – Zcg*. This gives *xTire* and *yTire* for the left switch tires. To find *yRail* at *xTire*, we have the following equations:

From equation (7-1), if $-L_{swx} \leq s < 0$

$$yRail_{left} = 0.5w - w_{swx} + D_{swx}\left(\frac{xTire}{L_{swx}}\right)^3$$

$$yRail_{right} = -0.5w + w_{swx} - D_{swx}\left(\frac{xTire}{L_{swx}}\right)^3$$

(10.2-1)

For the left switch rail, if $0 \leq s < s_1$ then we must use the same numerical method used in Section 10.1. In this case from equations (10.1-2), using equation (10.1-5) to find *s(xRail),* we have for $s < s_1$

$$\Psi = \frac{J_n s^2}{2V^3}, \qquad xRail = s\left(1 - \frac{1}{10}\Psi^2\right), \qquad yRail = (0.5w - w_{swx})/cos\Psi + \frac{s\Psi}{3}\left(1 - \frac{1}{14}\Psi^2\right)$$

(10.2-2)

If $s \geq s_1$we have

$$yRail_{left} = (0.5w - w_{swx})/cos\Psi + y_c - Rcos\Psi - SwxFlare(s)/cos\Psi$$

where

$$cos\Psi = \sqrt{1 - \left(\frac{xTire - x_c}{R}\right)^2}$$

in which in the region $s = sStart + L_{main}$ to $sStart + L_{main} + L_{swx}$

$$SwxFlare(s) = D_{swx}\left(\frac{s - S_{start} - L_{main}}{L_{swx}}\right)^3 \text{ otherwise 0.}$$

(10.2-3)

Then, similar to equation (10.1-4), for positive deflection for the left switch rail we have

$$\delta = (yRail - yTire)cos\Psi \text{ if} > 0 \; else \; 0.$$

(10.2-3)

For the right switch rail and $s > 0$

$$yRail_{right} = -0.5w + w_{swx} + SwxFlare(s)$$

(10.2-4)

i.e. set $\Psi = 0$ and change the sign.

10.3 Deflection of the side tires running against the inner main running surface.

First, the right side of the left guideway.

The values of *xTire* and *yTire* are computed from equations (10.1-1) with $Y_w = Yr$ from Table 1. From equation (8-7) we have for $s \geq s_{start}$

$$cos\Psi = \sqrt{1 - \left(\frac{xTire - x_c}{R}\right)^2}$$

$$yRail_{right} = -0.5w/cos\Psi + y_c - Rcos\Psi - y_{flare}(s)/cos\Psi$$

(10.2-5)

in which

$$y_{flare}(s) = D_{main}\left(\frac{s_{start} + L_{main} - s}{L_{main}}\right)^3 \text{ if } s_{start} \leq s < s_{start} + L_{main} \text{ else } 0.$$

The positive deflection is then given by equation (10.2-3).

For the left side of the right guideway, to get $xTire$ & $yTire, set\ x(s) = s, y(s) = 0, y_w = Yl$ from Table 1.

$$yRail_{left} = 0.5w + y_{flare}(s)$$

Then

$$\delta = yTire - yRail_{left}.$$

11. Deflection of the Main Tires

If the four main support tires that run on the bottom horizontal surface are pneumatic[2], the force on each tire from Appendix C is proportional to the first power of the deflection. Thus

$$\delta_{FL} + \delta_{FR} + \delta_{BL} + \delta_{BR} = \frac{W_{veh} + W_{pass}}{k}$$

(9-5)

in which *k* is the main tire stiffness, W_{veh} is the empty weight of the vehicle, and W_{pass} is the weight of the passenger (see Appendix C). If the vehicle has tilted to the right by the angle ϕ, the deflection of the right tire is greater than the deflection of the left tire by

$$\delta_{FR} = \delta_{FL} + D_{mainwheels}\phi, \qquad \delta_{BR} = \delta_{BL} + D_{mainwheels}\phi$$

(9-6)

in which $D_{mainwheels}$ is the separation between the left and right tire contact points.

If X_{mc} is the distance between the axles of the rear wheels and the mass center of the vehicle, X_{pass} is the horizontal distance of the passenger to the rear wheels and *WB* is the wheelbase of the vehicle, then by taking moments about the rear wheels we get

[2] They may be the new airless tires that have the same properties as pneumatic tires.

$$\delta_{FR} + \delta_{FL} = \frac{W_{veh}X_{mc} + W_{pass}X_{pass}}{kWB}$$

(9-7)

Substituting into equation (9-5),

$$\begin{aligned}\delta_{BR} + \delta_{BL} &= \frac{W_{veh} + W_{pass}}{k} - \frac{W_{veh}X_{mc} + W_{pass}X_{pass}}{kWB} \\ &= \frac{W_{veh}(WB - X_{mc}) + W_{pass}(WB - X_{pass})}{kWB}\end{aligned}$$

(9-8)

Then, using equations (9-6) we get

$$\delta_{FR} = \delta_{FL} + D_{mainwheels}\phi, \qquad \delta_{BR} = \delta_{BL} + D_{mainwheels}\phi$$

$$\delta_{FL} = \frac{1}{2}\left(\frac{W_{veh}X_{mc} + W_{pass}X_{pass}}{kWB} - D\phi\right), \qquad \delta_{FR} = \frac{1}{2}\left(\frac{W_{veh}X_{mc} + W_{pass}X_{pass}}{kWB} + D\phi\right)$$

$$\delta_{BL} = \frac{1}{2}\left[\frac{W_{veh}(WB-X_{mc})+W_{pass}(WB-X_{pass})}{kWB} - D\phi\right], \quad \delta_{BR} = \frac{1}{2}\left[\frac{W_{veh}(WB-X_{mc})+W_{pass}(WB-X_{pass})}{kWB} + D\phi\right]$$

(9-9)

where $D = D_{mainwheels}$.

12. A Limitation on the Tire Force-Deflection Relationship

In Section 9 it is shown that the force-deflection relationship for the side tires is

$$Force = k\delta^{3/2}$$

(10-1)

where δ is the deflection and *k* is a constant that must be low enough to meet ride-comfort standards, but not so low that the chassis will rub against the top edge of the cover. Thus, we must know the distance between the top edge of the cover and the centerline of the main support wheels. We must calculate the quantity

$$y_{cover} = [y_{mc} - (Z_{cover} - Z_{mc})\phi]_{max}$$

where $Z_{cover} = 31.6$ inches. (10-2)

13. Passenger Motion

Our prime interest is in the lateral acceleration of the passenger. We model the passenger as a concentrated mass located a distance *D_{pass}* above the seat and subject to the lateral acceleration of the seat. It takes about 5 lb to move the passenger's mass center sideways one inch, giving a spring constant of

$$k_{pass} = 5\frac{lb}{in} \times \frac{1\,in}{0.0254\,m} \times \frac{4.448\,N}{lb} = 875 N/m$$

The equation of motion is

$$\frac{W_{pass}}{g}\ddot{y}_{pass} = -k_{pass}y_{pass} - c\dot{y}_{pass} + \frac{W_{pass}}{g}\ddot{y}_{seat}$$

in which *c* is a damping constant. In standard form this equation is

$$\ddot{y}_{pass} + 2\zeta\omega_n\dot{y}_{pass} + \omega_n^2 y_{pass} = \ddot{y}_{seat}$$

in which

$$\omega_n^2 = \frac{k_{pass}g}{W_{pass}}, \qquad 2\zeta\omega_n = \frac{cg}{W_{pass}}$$

We will assume that $\zeta \cong 0.7$.

14. Motion of the LIM Bogie

Two LIMs are mounted side by side and ride on a set of four 4-in diameter wheels, two in the front and two in the back. This bogie is the tug that propels the vehicle and it is guided sideways via attachment points on the chassis, which are located in body axes at points given in Table 1. From equation (3-1), the vector distance to these points is

$$\vec{R} = \vec{R}_0 + y_{mc}\hat{\jmath}' + \Delta X\hat{\imath}_b + \Delta Z\hat{k}_b \tag{14-1}$$

where

$$\Delta X = X_{LIM_{f,b}} - X_{mc}, \qquad \Delta Z = Z_{LIM} - Z_{mc} \tag{14-2}$$

But

$$X_{LIM_b} - X_{mc} = -\left(X_{LIM_f} - X_{mc}\right)$$

Thus, let $\Delta X \equiv X_{LIM_f} - X_{mc} > 0$.

(14-2a)

The LIM Bogie front and back attachment points will follow the chassis attachment points through a pair of springs of spring constant k_{LIM}.

Side motion of the LIM Bogie is defined by two parameters: y_{LIMmc}, which is the sidewise motion of the LIM-bogie mass center; and ψ_{LIM}, the angular motion about the mass center referenced to the direction of the guideway. Thus, the vector distance to these attachment points is

$$\vec{R}_{LIM} = \vec{R}_0 + y_{LIMmc}\hat{j}' \pm \Delta X\hat{\imath}_b \pm \Delta X(\psi_{LIM} - \psi)\hat{j}_b + \Delta Z\hat{k}_b \tag{14-3}$$

Thus

$$\Delta\vec{R} = \vec{R}_{LIM} - \vec{R} = (y_{LIMmc} - y_{mc})\hat{j}' \pm \Delta X(\psi_{LIM} - \psi)\hat{j}_b \tag{14-4}$$

in which, from equations (3-3),

$$\hat{j}_b = -S\psi C\phi\hat{\imath}' + C\psi C\phi\hat{j}' + S\phi\hat{k}'$$

Thus, since the angles are very small, the LIM-bogie attachment points are displaced laterally (in the $\hat{j}'$direction) from the corresponding chassis attachment point by the amounts

$$\Delta = y_{LIMmc} - y_{mc} \pm \Delta X(\psi_{LIM} - \psi) \tag{14-5}$$

Thus the force exerted by the chassis on the bogie is

$$F = -k_{LIM}\Delta \tag{14-6}$$

and by the bogie on the chassis $+k_{LIM}\Delta$.

When the bogie experiences a $\hat{j}'$ component of velocity there will be a friction force from the running surface on the bogie tires. To find it, differentiate equation (14-3). Thus

$$\frac{d\vec{R}_{LIM}}{dt} = \vec{V} + \dot{y}_{LIM_{mc}}\hat{j}' + y_{LIMmc}\frac{d\hat{j}'}{dt} \pm \Delta X\frac{d\hat{\imath}_b}{dt} \pm \Delta X(\dot{\psi}_{LIM} - \dot{\psi})\hat{j}_b \pm \Delta X(\psi_{LIM} - \psi)\frac{d\hat{j}_b}{dt}$$
$$+ \Delta Z\frac{d\hat{k}_b}{dt}$$

But

$$d\hat{j}' = -d\Psi\hat{\imath}', \qquad d\hat{\imath}_b = d\psi\hat{j}_b, \qquad d\hat{j}_b = -d\psi\hat{\imath}_b + d\phi\hat{k}_b, \qquad d\hat{k}_b = -d\phi\hat{j}_b$$

Thus,

$$\vec{V}_{LIM} = \vec{V} + \dot{y}_{LIM_{mc}}\hat{j}' - y_{LIMmc}\dot{\Psi}\hat{\imath}' \pm \Delta X\dot{\psi}\hat{j}_b \pm \Delta X(\dot{\psi}_{LIM} - \dot{\psi})\hat{j}_b$$
$$\pm \Delta X(\psi_{LIM} - \psi)(-\dot{\psi}\hat{\imath}_b + \dot{\phi}\hat{k}_b) - \Delta Z\dot{\phi}\hat{j}_b$$

$$
\begin{aligned}
&= V\hat{\imath}' + \dot{y}_{LIM_{mc}}\hat{\jmath}' - y_{LIMmc}\dot{\psi}\hat{\imath}' + \left(\pm\Delta X\dot{\psi}_{LIM} - \Delta Z\dot{\phi}\right)\left(-S\psi C\phi\hat{\imath}' + C\psi C\phi\hat{\jmath}' + S\phi\hat{k}'\right) \\
&\qquad\qquad \pm \Delta X(\psi_{LIM} - \psi)\left[-\dot{\psi}(C\psi\hat{\imath}' + S\psi\hat{\jmath}') + \dot{\phi}\left(S\psi S\phi\hat{\imath}' - C\psi S\phi\hat{\jmath}' + C\phi\hat{k}'\right)\right]
\end{aligned}
\tag{14-7}
$$

Dropping products of small angles, the transverse component (in the $\hat{\jmath}'$ direction) is

$$\dot{y}_{LIM_{mc}} \pm \Delta X\dot{\psi}_{LIM} - \Delta Z\dot{\phi}$$

Note that ΔZ is negative. When ψ_{LIM} is not in the direction of the bogie velocity vector, the LIM wheels will be subject to a side-friction force proportional to the difference between ψ_{LIM} and the direction the velocity vector, the downward force on the bogie wheels, and the coefficient of rolling friction μ. This force on the LIM-bogie wheels is in the direction of the sign of $\psi_{LIM} - \frac{\dot{y}_{LIM_{mc}} \pm \Delta X\dot{\psi}_{LIM} - \Delta Z\dot{\phi}}{V}$. Thus the friction force on the LIM bogie is

$$F_{friction_{f,b}} = \frac{1}{2}\mu F_{normal}\left(\psi_{LIM} - \frac{\dot{y}_{LIM_{mc}} \pm \Delta X\dot{\psi}_{LIM} - \Delta Z\dot{\phi}}{V}\right) \tag{14-8}$$

When the LIMs are operating, a normal force is produced closely equal to the thrust, which, when speed is constant, is equal to the sum of air drag and rolling resistance. Thus,

$$F_{normal} = W_{LIM} + AirDrag + RollingResistance$$

in which (see equation (2-2))

$$AirDrag = CV^2, \qquad RollingResistance = F_{normal}(a + bV)$$

in which *a* and *b* are constants. Thus

$$F_{normal} = \frac{W_{LIM} + AirDrag}{1 - (a + bV)}$$

Therefore,

$$F_{friction_{f,b}} = \frac{1}{2}\mu\left[\frac{W_{LIM} + AirDrag}{1 - (a + bV)}\right]\left(\psi_{LIM} - \frac{\dot{y}_{LIM_{mc}} \pm \Delta X\dot{\psi}_{LIM} - \Delta Z\dot{\phi}}{V}\right) \tag{14-9}$$

Thus, using equations (14-5, 6), the sum of the forces on the bogie is

$$\sum Forces = -2k_{LIM}(y_{LIMmc} - y_{mc}) + \mu\left[\frac{W_{LIM} + AirDrag}{1 - (a + bV)}\right]\left(\psi_{LIM} - \frac{\dot{y}_{LIM_{mc}} - \Delta Z\dot{\phi}}{V}\right)$$

The sum of the yaw moments on the bogie about the LIM mass center is

$$\sum Moments = -2k_{LIM}(\psi_{LIM} - \psi)\Delta X^2 - \mu\left[\frac{W_{LIM} + AirDrag}{1-(a+bV)}\right]\frac{\dot{\psi}_{LIM}}{V}\Delta X^2$$

The equations of motion of the LIM bogie are

$$\frac{W_{LIM}}{g}\ddot{y}_{LIMmc} = \sum Forces$$
$$\frac{W_{LIM}}{g}r_{LIM}^2\ddot{\psi}_{LIM} = \sum Moments$$

The side force produced by the LIM bogie on the chassis is

$$2k_{LIM}(y_{LIMmc} - y_{mc})$$

The yaw moment produced by the LIM bogie on the chassis is

$$2k_{LIM}(\psi_{LIM} - \psi)\Delta X^2$$

The roll moment produced by the LIM bogie on the chassis is

$$2k_{LIM}(y_{LIMmc} - y_{mc})(-\Delta Z)$$

15. Numerical Solution of the Equations of Motion

Each of the three second-order differential equations of Section 2 can be written in general as a pair of first-order differential equations.

$$\frac{du}{dt} = f(u,t), \qquad \frac{dx}{dt} = u$$

(11-1)

From the paper "A Practical Method for Numerical Solution of Differential Equations" we take as the solution of equations (11-1)

$$u_{n+1} = u_n + 0.5\delta t(3f_n - f_{n-1}), \qquad x_{n+1} = x_n + 0.5\delta t(u_{n+1} + u_n)$$

(11-2)

in which δt is preset to reduce numerical errors to an acceptable level. With use of double precision numbers, experience has shown that with a value of δt low enough to keep truncation errors to an acceptable level, round errors will be negligible.

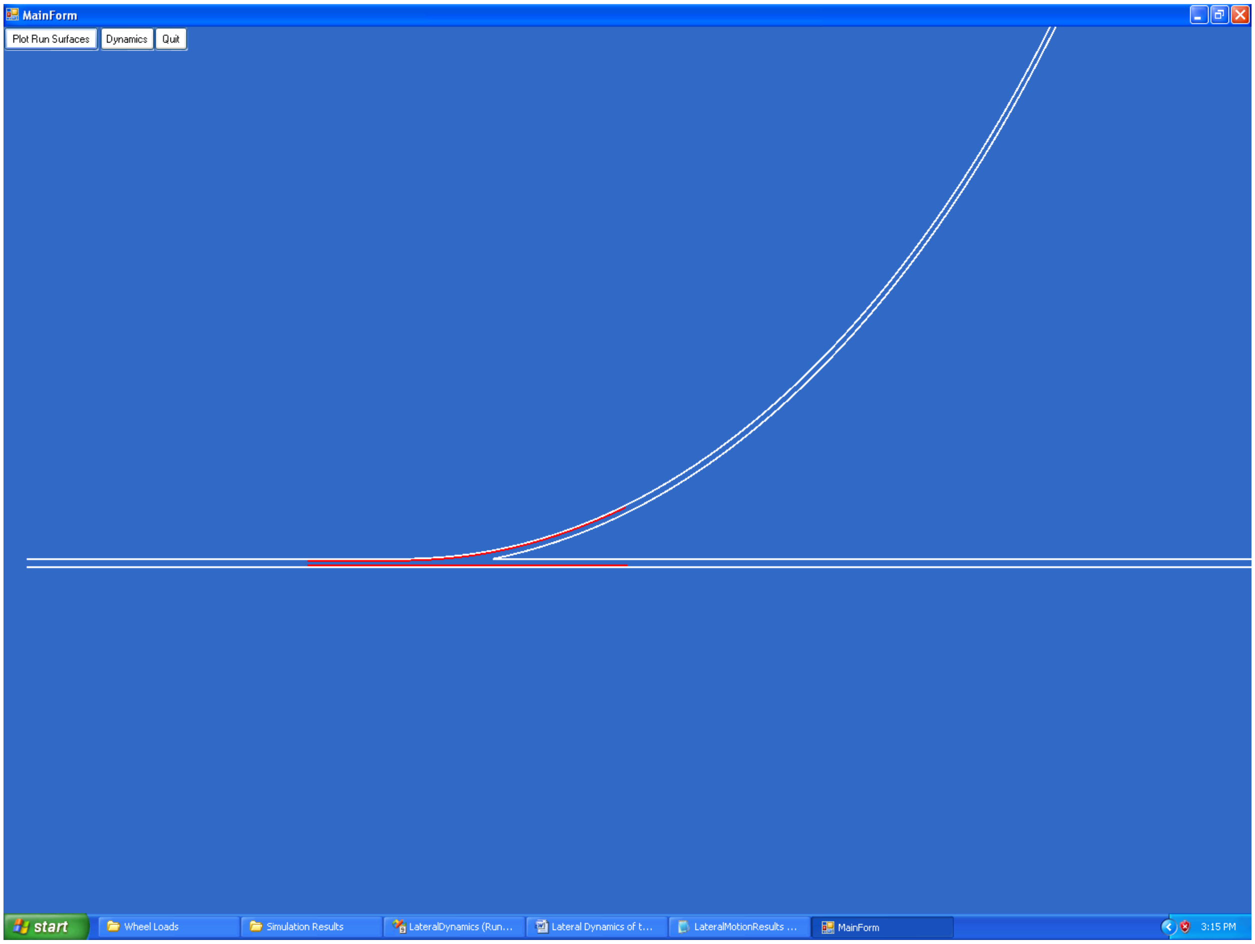

Figure 3. The running surfaces in a diverge section of guideway. Switch rails in red.

16. Results and Discussion

Figure 3 shows the surfaces against which the side wheels run. The switch rails, as they are flared in and out, appear as red lines. Motion of the vehicle, which is calculated using the program of Appendix G, is from left to right. The object of the simulation is to determine the maximum tire loads and tire stiffnesses, and the parameters that will make 1) the lateral displacement and acceleration of the passenger acceptable and 2) the lateral displacement at the cover acceptable.

In Figure 4, motion is again from left to right. The figure shows the wheel loads, lateral displacement of the chassis at the guideway cover, and the lateral displacement of the passenger when a 500 lb passenger is displaced one seat width (20 in) to the right, i.e., in the direction that will add to the moments generated by the centrifugal force, and a wind force on the vehicle produced by a wind speed of 30 mph. The two horizontal red lines in Figure 4 correspond to a force of ±2000 lb and the two horizontal yellow lines correspond to an acceleration of 0.2g. The long vertical white line is the point in the guideway when the vehicle mass center reaches *s* = 0. The series of short vertical lines are spaced ten feet apart, and the taller vertical white line marks the point at which the vehicle mass center reaches the diverge-point junction. In the following runs, the line speed is 15.6 m/s or 35 mph. Seven curves are shown in Figure 4: The lateral displacement at the slot at the top of the covers, the forward and rear switch forces on the left side, the forward and rear lower lateral forces on the left side, and the forward and rear upper lateral wheel forces on the right side. Table 2 gives the details of one run.

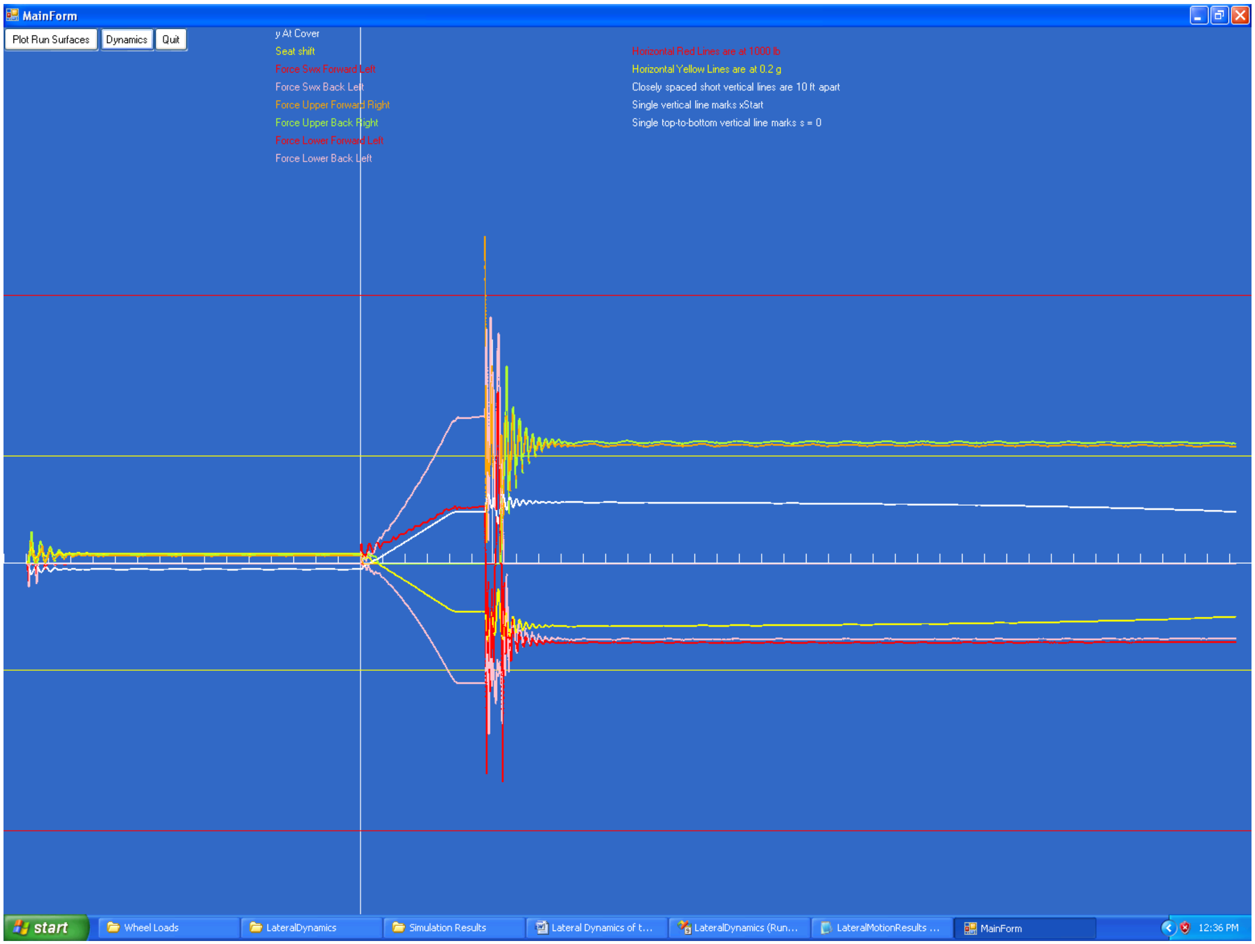

Figure 4. Tire forces and passenger acceleration.

Table 2. A Basic Set of Results

```
LATERAL MOTION RESULTS
 Date: 11/14/2015 3:07:12 PM Computational distance step: 0.002
 Positive directions: forward, left, up
 Guideway design speed and vehicle speed, mi/hr 35
 Tolerance, i.e., distance between tire and rail, in 0
 sStart @ Diverge Junction, ft:59.25
```

```
sPositiveDeflection, ft:89.58
Tire stiffnesses. Main lb/in, Side lb/in^1.5
kmain: 3,000.0, kUpper: 60,000, kLower: 110,000, kSwitch: 120,000
Passenger weight, lb: 500, kSeat: 400
Vehicle weight, lb: 1200, LIM Weight: 400
MainFlareLength, ft 36, MainFlareOffSet, in 1.5
SwxFlareLength, ft 24, SwxFlareOffSet, in 1
Wind Speed, ft/s -44, Passenger Offset,in -20
Energy lost in side tires: 25%
Tire friction coefficient: 0.25
Centrifugal Force on if OnOff = 1, off if OnOff = 0, OnOff = 1
Maximum Force between LIM bogie and Chassis: 27.86 lb.

Max Roll Angle, deg: 0.054, sRollMax, ft: 65.9
Max Yaw Angle, deg: 0.072, sYawMax, ft: 56.1
Max yMC, in: 0.1, syMCMax, ft: 57.3

Deflections, inches
MaxDeflUFL     MaxDeflUBL     MaxDeflLFL     MaxDeflLBL
   0.000           0.000           0.051           0.033
MaxDeflUFR     MaxDeflUBR     MaxDeflLFR     MaxDeflLBR
   0.099           0.045           0.000           0.000
MaxDeflSFL     MaxDeflSBL
   0.033           0.045

Maximum Forces,lb
MaxForceUFL      MaxForceUBL      MaxForceLFL      MaxForceLBL
   0.0           0.0           -1,254.6           -661.8
MaxForceUFR      MaxForceUBR      MaxForceLFR      MaxForceLBR
   1,875.8           565.1           0.0           0.0
MaxForceSFL      MaxForceSBL
   735.4           1,162.8

s at MaxForceUFR 55.4, s at MaxForceSBL 59.0

yMCAccelMax, g's   MaxPassAccel, g's   MaxSeatShift, in
   1.563              0.000                0.17
yCoverMax, in           s at yCoverMax, ft
   0.145                    57.26
UFR deflection at sStart, in: -1.446, s when UFR tire hits, ft 89.58
```

The parameters oscillate before *s* = 0 because it takes about half a second for the vehicle to settle down after being suddenly struck by the side wind and the off-set passenger. The curves on the right side of *s* = 0 are of significance. Detailed results of one run are tabulated in Table 2. Table 3 gives the results of a series of runs. The forces on the right side vanish as the vehicle pulls away from the right-side running surface and moves down the curved guideway. During this period, the vehicle retains its vertical position as a result of the moment developed by forces to the left on the left switch wheels and forces to the right on the lower lateral wheels on the left side of the chassis. When the forward right upper wheel reaches the diverge junction, it impacts the flared right side of the left, or curved, guideway. This occurs in the run shown in Table 2 when the vehicles center of mass reaches s = 89.6 feet or slightly more than two seconds from *s* = 0. The force on this wheel suddenly jumps to its maximum value of 1876 lb in the best set of runs in Table 3. The upper-back-right wheel, however, reaches a maximum force of only 738 lb. The force on the left-rear switch wheel peaks slightly after engagement of the upper-forward-right wheel and then vanishes in about a third of a second as vehicle support is picked up by the wheels on the right side. The discontinuities in the force curves are due to tire hysteresis produced when the deflection on the tire stops decreasing and suddenly must increase.

Table 3. Some Results of Parameter Variations

Swx	Main	k	k	k	k	EI	Weight	Damping	Energy	Pass	Max	Max	Max	Max
Flare	Flare	Main	Upper	Lower	Swx	Seat	Pass	Coeff.	Loss	Offset	Y	Seat	SideTire	SwxTire
											Cover	Shift	Force	Force
ft	ft	lb/in	lb/in^1.5	lb/in^1.5	lb/in^1.5	lb-in^2	lb		%	in	in	in	lb	lb
24	36	3000	60,000	110,000	120,000	2000	500	0.7	25	20	0.145	0.170	1876	1163
12											0.145	0.170	1876	1163
	18										0.145	0.170	1876	1163
	9										0.149	0.178	1876	1163
6											0.149	0.178	1875	1162
4											0.149	0.178	1875	1162
	5										0.145	0.172	1875	1162
	3										0.166	0.215	1875	1162
3	4										0.161	0.208	1874	1161
		6000									0.140	0.158	1809	1056
			40,000								0.134	0.148	1206	955
				100,000							0.135	0.148	1207	933
				90,000							0.137	0.149	1208	939
					110,000						0.138	0.148	1223	922
						1000					0.138	0.148	1223	922
						4000					0.138	0.148	1223	922
							200				0.142	0.154	1192	907
							50				0.143	0.158	1177	900
							500				0.141	0.152	1184	889
											0.138	0.148	1222	909
								0.5			0.138	0.148	1222	922
									10		0.147	0.160	1220	1215
									40		0.133	0.142	1223	828
									25	0	0.144	0.159	1173	898
								0.7		20	0.138	0.148	1222	922

Table 4. The Best Results.

```
LATERAL MOTION RESULTS
 Date: 11/17/2015 12:34:52 PM Computational distance step: 0.002
 Positive directions: forward, left, up
 Guideway design speed and vehicle speed, mi/hr 35
 Tolerance, i.e., distance between tire and rail, in 0
 sStart @ Diverge Junction, ft:59.25
 sPositiveDeflection, ft:65.98
 Tire stiffnesses. Main lb/in, Side lb/in^1.5
 kmain: 6,000.0, kUpper: 40,000.0, kLower: 90,000.0, kswitch: 110,000.0
 Passenger weight, lb: 500, kSeat: 400
 Vehicle weight, lb: 1200 LIM Weight: 400
 MainFlareLength, ft 4, MainFlareOffSet, in 1.5
 SwxFlareLength, ft 3, SwxFlareOffSet, in 1
 Wind Speed, ft/s -44, Passenger Offset,in -20
 Energy lost in side tires: 25%
 Tire friction coefficient: 0.25
 Centrifugal Force on if OnOff = 1, off if OnOff = 0, OnOff = 1
 Maximum Force between LIM bogie and Chassis: 27.06

 Max Roll Angle, deg:0.052, sRollMax, ft: 61.5
 Max  Yaw Angle, deg:0.064, sYawMax, ft: 56.2
 Max yMC, in:0.1, syMCMax, ft: 63.4

 Deflections, in
 MaxDeflUFL     MaxDeflUBL     MaxDeflLFL     MaxDeflLBL
    0.000           0.000           0.043           0.037
 MaxDeflUFR     MaxDeflUBR     MaxDeflLFR     MaxDeflLBR
    0.098           0.070           0.000           0.001
 MaxDeflSFL     MaxDeflSBL
    0.032           0.041

 Maximum Forces,lb
 MaxForceUFL      MaxForceUBL      MaxForceLFL      MaxForceLBL
    0.0           0.0           -813.4           -635.8
 MaxForceUFR      MaxForceUBR      MaxForceLFR      MaxForceLBR
    1,221.9           738.3           0.9           3.5
 MaxForceSFL      MaxForceSBL
    640.2           922.3

 s at MaxForceUFR 55.4 s at MaxForceSBL 58.0

 yMCAccelMax, g's   MaxPassAccel, g's   MaxSeatShift, in
    1.018            0.000            0.148
 yCoverMax, in           s at yCoverMax, ft
    0.138               63.42
 UFR deflection at sStart, in: -1.451, s when UFR tire hits, ft 65.98
```

```
   Forces at end of run,lb
   ForceUFL      ForceUBL      ForceLFL      ForceLBL
      0.0           0.0           -291.5           -278.6
   ForceUFR      ForceUBR      ForceLFR      ForceLBR
      438.0           449.8           0.0           0.0
   ForceSFL      ForceSBL
0.0           0.0
```

Table 3 reports on a series of runs aimed at finding the best values of the flare lengths and tire stiffnesses, and to note effect of changing the amount of damping and the energy loss in the tires. These are of course only sample runs and with assumed values of the vehicle weight and moments of inertia about the roll and yaw axes. The vehicle designer will correct these assumptions as well as the assumptions about placement of the wheels and other parameters and will need to make may runs to become satisfied with the parameters chosen.

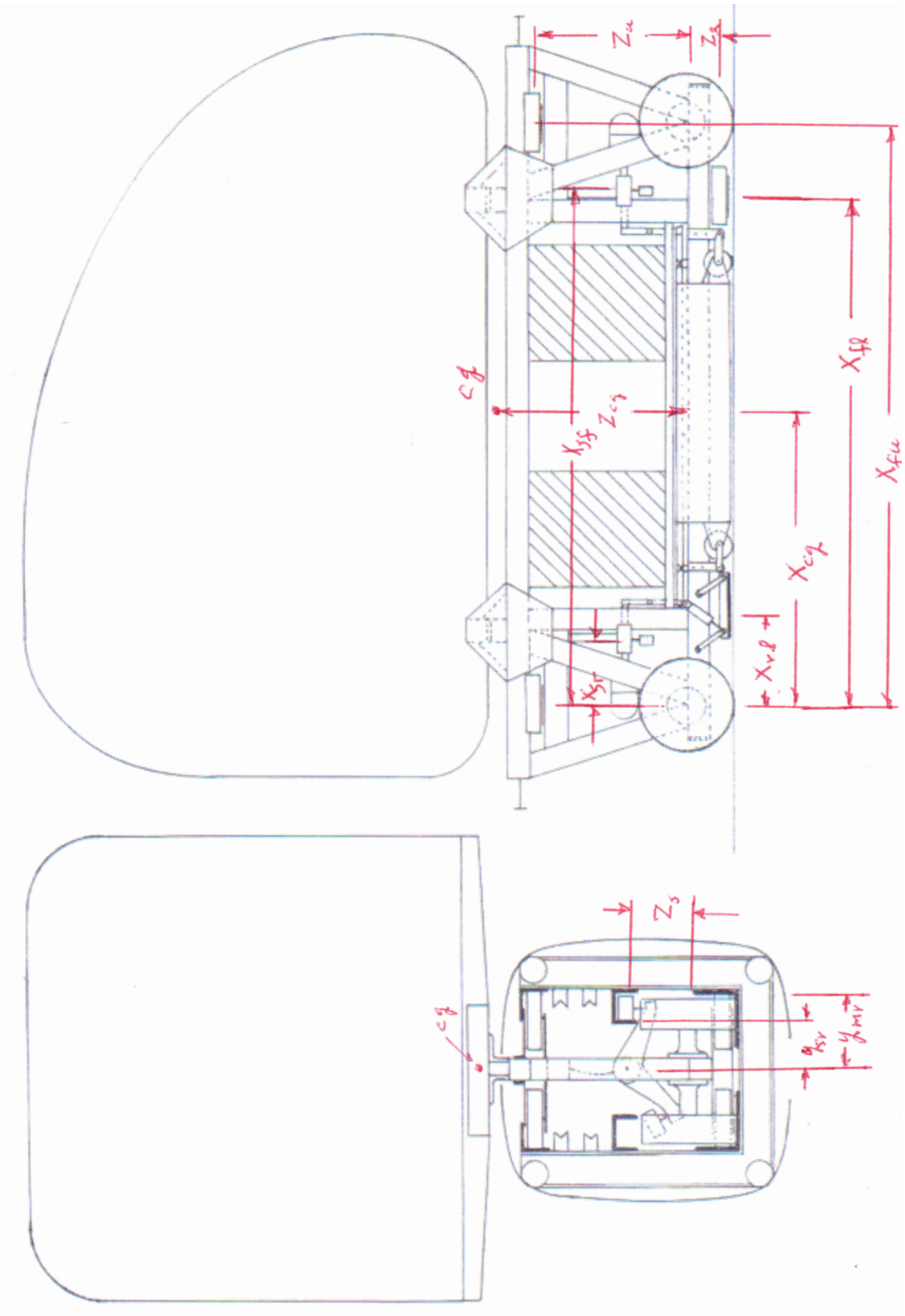

Figure 5. The Vehicle Dimensioned.

Appendix A. The Equations of Motion in a Rotating Reference Frame

Figure 1 shows the reference frames used in the analysis. The reference frame *x, y, z* is fixed with respect to the earth and we assume for this calculation, as well as is done for many calculations of motion, that the earth's rotation is sufficiently small that the basic laws of motion are valid fixed to the earth. Let a reference frame x', y', z' move with the vehicle as it moves along a curved guideway and let it be centered in the center of the guideway. Let the x' axis be in the local direction of the centerline of the guideway and the y' axis be in the transverse direction to the left. In accordance with the right-hand rule an orthogonal z' coordinate will then point upward. In this reference frame, we take the x'-component of the velocity of the center of mass of the vehicle *V* to be constant. The angle between *x* and x' has in Section 3 been called Ψ, which is greater than zero if the rotation of the x', y', z' reference frame is counterclockwise as shown in Figure 1, i.e., according to the right-hand rule. The vehicle has the three degrees of freedom y_{mc}, ψ, ϕ with respect to the reference frame x', y', z'. ψ and ϕ are positive according to the right-hand rule.

Designate as the vector $\vec{R}$ the position of a point *P* fixed in the vehicle measured from the origin of the *x, y, z* reference frame and as the vector $\overrightarrow{R'}$with respect to the origin of x', y', z' . Let a vector from the origin of the *x, y, z* frame to the origin of the x', y', z' frame be called $\overrightarrow{R_0}$. Then

$$\vec{R} = \overrightarrow{R_0} + \overrightarrow{R'}$$

We will take point P as a wheel-contact point. This contact point has the fixed body coordinates x_w, y_w, z_w where the origin of body coordinates is at the center of the axel of the rear wheels. The mass center of the vehicle is a distance X_{mc} ahead of the origin of body coordinates and a distance Z_{mc} above it. The vector $\vec{R}'$ can usefully be broken up into three vectors: the vector distance from the guideway center at the vehicle mass center to the vehicle mass center, the vector distance from the vehicle mass center to the origin of body coordinates, and the vector distance from the origin of body coordinates to point P. Thus

$$\vec{R}' = y_{mc}\,\hat{\jmath}' - X_{mc}\hat{\imath}_b - Z_{mc}\hat{k}_b + x_w\hat{\imath}_b + y_w\hat{\jmath}_b + z_w\hat{k}_b$$

in which x_w, y_w, z_w are the coordinates of a wheel contact point with respect to the origin of the body coordinates, i.e., the values given in Table 1.

Thus the vector from the origin of the fixed reference frame to a wheel contact point is

$$\vec{R} = \vec{R}_0 + y_{mc}\hat{\jmath}' + \left(x_w - X_{cg}\right)\hat{\imath}_b + y_w\hat{\jmath}_b + \left(z_w - Z_{cg}\right)\hat{k}_b$$

in which $\hat{\imath}, \hat{\jmath}$ will be unit vectors in the *x, y* reference frame, $\hat{\jmath}'$ is a unit vector in the x', y' reference frame, and the unit vectors designated by subscript $_b$ are unit vectors in body axes.

The velocity of the mass center of the vehicle is then

$$\vec{V}_{mc} = \frac{d\vec{R}}{dt} = V\hat{\imath}' + \dot{y}_{mc}\hat{\jmath}' + y_{mc}\frac{d\hat{\jmath}'}{dt}$$

But

$$\frac{d\hat{\imath}'}{dt} = \dot{\Psi}\hat{\jmath}', \quad \frac{d\hat{\jmath}'}{dt} = -\dot{\Psi}\hat{\imath}'$$

Therefore,

$$\vec{V}_{mc} = \left(V - y_{mc}\dot{\Psi}\right)\hat{\imath}' + \dot{y}_{mc}\hat{\jmath}'$$

The acceleration of the mass center of the vehicle is

$$\vec{A}_{mc} = \frac{d\vec{V}_{mc}}{dt} = V\dot{\Psi}\hat{\jmath}' + \ddot{y}_{mc}\hat{\jmath}' - 2\dot{y}_{mc}\dot{\Psi}\hat{\imath}' - y_{mc}\ddot{\Psi}\hat{\imath}' - y_{mc}\dot{\Psi}^2\hat{\jmath}'$$

The $\hat{\jmath}'$component of acceleration of the mass center of the vehicle is equal to the sum of the lateral forces on the vehicle divided by the mass of the vehicle. Thus

$$\ddot{y}_{mc} + \dot{\Psi}\left(V - y_{mc}\dot{\Psi}\right) = \frac{\sum LateralForces}{GrossMass}$$

But $\dot{\Psi} = \frac{d\Psi}{dt} = \frac{ds}{dt}\frac{d\Psi}{ds} = V\frac{d\Psi}{ds} = \frac{V}{R}$, where *R* is the radius of curvature of the guideway. Thus

$$\ddot{y}_{mc} = -\frac{V^2}{R}\left(1 - \frac{y_{mc}}{R}\right) + \frac{\sum LateralForces}{GrossMass} \cong -\frac{V^2}{R} + \frac{\sum LateralForces}{GrossMass}$$

in which the factor $\frac{y_{mc}}{R}$ is a small fraction of an inch divided by an *R* of upwards of 60 feet.

The lateral acceleration at the passenger level is

$$A_{passenger} = \ddot{y}_{mc} - \ddot{\phi}\left(Z_{passenger} - Z_{cg}\right)$$

Appendix B. Force-Deflection Relationships for Tires

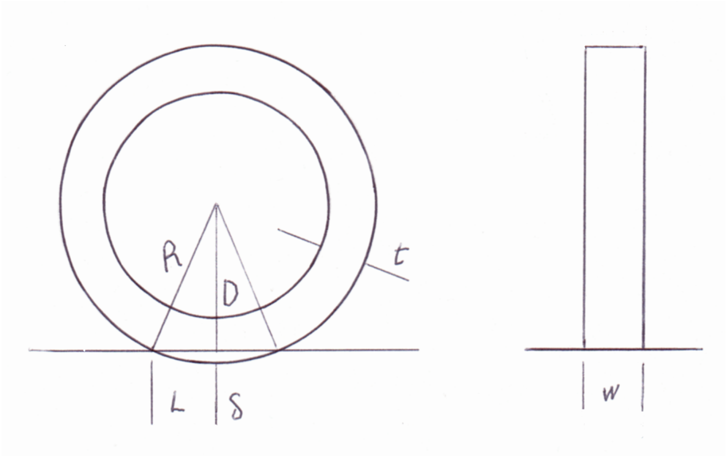

Figure 1. A Deflected Tire.

Consider a wheel with an outside radius *R* and tire thickness *t* and width *w*. The deflection of the tire with respect to the running surface is δ. The distance from the centerline to the point at which deflection is zero is *L*. Then the distance *D* is given by

$$D^2 + L^2 = R^2, D = \sqrt{R^2 - L^2}$$

Hence the deflection is

$$\delta = R - D = R - \sqrt{R^2 - L^2}$$

and

$$L^2 = R^2 - (R - \delta)^2 = 2R\delta - \delta^2, \qquad \frac{L}{R} = \sqrt{2\frac{\delta}{R} - \frac{\delta^2}{R^2}} = \sqrt{\frac{\delta}{R}}\sqrt{2 - \frac{\delta}{R}}$$

Let an *x* coordinate be placed along the running surface with the origin at the center point of the wheel and a *y* coordinate be placed vertically along the centerline of the tire, with its origin at the running surface. In terms of these coordinates, the equation of the outer tire surface is

$$x^2 + (y - D)^2 = R^2, \qquad y = D \pm \sqrt{R^2 - x^2}$$

The plus sign corresponds to a point near the top of the tire and the minus sign, which we want, corresponds to a point near the bottom of the tire. Thus the deflection of the tire at any point is

$$\delta(x) = -y = \sqrt{R^2 - x^2} - D = \sqrt{R^2 - x^2} - \sqrt{R^2 - L^2}$$

The strain ϵ at any point *x* is

$$\epsilon(x) = \frac{\delta(x)}{t}$$

The stress at the same point is

$$\sigma(x) = E\epsilon = E\frac{\delta(x)}{t}$$

where E is the modulus of elasticity.

If the contact surface is rectangular, as it will be if the tire is a flexible solid, the total force on the tire is

$$F = 2\int_0^L \sigma(x)w dx = 2\frac{Ew}{t}\int_0^L \delta(x)\, dx = 2\frac{Ew}{t}\int_0^L \left[\sqrt{R^2 - x^2} - \sqrt{R^2 - L^2}\right] dx$$

$$= 2\frac{Ew}{t}\left[\frac{L}{2}\sqrt{R^2 - L^2} + \frac{R^2}{2}sin^{-1}\left(\frac{L}{R}\right) - \sqrt{R^2 - L^2}L\right] = \frac{Ew}{t}\left[R^2 sin^{-1}\left(\frac{L}{R}\right) - L\sqrt{R^2 - L^2}\right]$$

$$= E\frac{wR^2}{t}\left[sin^{-1}\left(\frac{L}{R}\right) - \frac{L}{R}\sqrt{1 - \left(\frac{L}{R}\right)^2}\right]$$

where

$$\frac{L}{R} = \sqrt{\frac{\delta}{R}}\sqrt{2 - \frac{\delta}{R}} = 2\alpha^{1/2}(1-\alpha)^{1/2}$$

where $\alpha = \delta/2R$.

We need to expand the function $F(\delta)$ into a power series in α. The expansion of $f(\alpha) = (1-\alpha)^{1/2}$ about $\alpha = 0$ is Maclaurin's series and in general is given by

$$f(x) = f(0) + f'(0)\frac{x}{1!} + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} + \cdots$$

In our case

$$f' = -\frac{1}{2}(1-\alpha)^{-1/2}, f'' = -\frac{1}{4}(1-\alpha)^{-3/2}, f''' = -\frac{3}{8}(1-\alpha)^{-5/2}$$

Therefore,

$$(1-\alpha)^{1/2} = 1 - \frac{1}{2}\alpha - \frac{1}{8}\alpha^2 - \frac{1}{16}\alpha^3 - \cdots$$

The series expansion of $sin^{-1}x$ is

$$sin^{-1}x = x + \frac{x^3}{6} + \frac{3x^5}{40} + \cdots$$

Therefore

$$\begin{aligned} sin^{-1}\left(\frac{L}{R}\right) &\approx 2\alpha^{1/2}(1-\alpha)^{1/2} + \frac{4}{3}\alpha^{3/2}(1-\alpha)^{3/2} \\ &\approx 2\alpha^{1/2}\left(1-\frac{1}{2}\alpha-\frac{1}{8}\alpha^2\right)\left[1+\frac{2}{3}\alpha(1-\alpha)\right] \\ &\approx 2\alpha^{1/2}\left(1+\frac{1}{6}\alpha-\frac{9}{8}\alpha^2\right) \end{aligned}$$

$$\frac{L}{R}\sqrt{1-\left(\frac{L}{R}\right)^2} = 2\alpha^{1/2}(1-\alpha)^{1/2}\sqrt{1-4\alpha(1-\alpha)} = 2\alpha^{1/2}(1-\alpha)^{1/2}(1-2\alpha)$$

$$\approx 2\alpha^{1/2}(1-2\alpha)\left(1-\frac{1}{2}\alpha-\frac{1}{8}\alpha^2\right) = 2\alpha^{1/2}\left(1-\frac{5}{2}\alpha+\frac{7}{8}\alpha^2\right)$$

$$F = E\frac{wR^2}{t}2\alpha^{1/2}\left(1+\frac{1}{6}\alpha-\frac{9}{8}\alpha^2-1+\frac{5}{2}\alpha-\frac{7}{8}\alpha^2\right) = E\frac{wR^2}{t}2\alpha^{1/2}\left(\frac{8}{3}\alpha-2\alpha^2\right)$$

$$\approx \frac{16}{3}E\frac{wR^2}{t}\alpha^{3/2} = \frac{16}{3}E\frac{wR^2}{t}\left(\frac{\delta}{2R}\right)^{3/2} = \frac{8}{3\sqrt{2}}E\frac{w}{t}R^{1/2}\delta^{3/2} = K\delta^{3/2}$$

So, we find that the force on the tire is close to proportionality to the three halves power of the deflection.

Round Pneumatic Tire

If the tire is air filled with a pressure *p*, the contact area is an ellipse. The length of the contact area, as calculated above, is

$$L = \delta^{1/2}(2R-\delta)^{1/2}$$

With a tire of width *w*, the width *b* of the contact area, by similar analysis, is

$$b = 2\delta^{1/2}(w-\delta)^{1/2}$$

The area of the elliptical contact area is

$$A = \pi L\frac{b}{2} = \pi\delta(2R-\delta)^{1/2}(w-\delta)^{1/2} \approx \pi\delta(2R)^{1/2}w^{1/2}$$

The force on the tire *F* is the tire pressure *p* multiplied by the area *A*. Thus

$$F = p\left[\pi\delta(2R)^{1/2}w^{1/2}\right] = k\delta$$

Appendix C. Energy Loss in a Tire

If the deflection $\delta > 0$ the force-deflection relationship for the tires on the left side when the deflection is increasing is

$$F = -k\delta^{1.5}$$

in which *F* is the force, *k* is a constant and δ is the deflection. Let the maximum deflection be labeled δ_{max}. When δ is decreasing assume

$$F = -k_r\delta^{\beta}$$

in which $\beta > 1.5$. At the maximum deflection these forces are equal. Thus

$$k\delta_{max}^{1.5} = k_r\delta_{max}^{\beta}$$

So

$$k_r = \frac{k}{\delta_{max}^{\beta-1.5}}$$

The energy lost is

$$\Delta E = \int_0^{\delta_{max}} \left(k\delta^{1.5} - k_r\delta^{\beta}\right) d\delta = k\frac{\delta_{max}^{2.5}}{2.5} - k_r\frac{\delta_{max}^{\beta+1}}{\beta+1} = k\left(\frac{\delta_{max}^{2.5}}{2.5} - \frac{\delta_{max}^{2.5}}{\beta+1}\right)$$

Thus

$$\frac{\Delta E}{E_{in}} = 1 - \frac{2.5}{\beta+1}$$

Hence

$$\beta = \frac{2.5}{1 - \frac{\Delta E}{E_{in}}} - 1$$

For example, if $\frac{\Delta E}{E_{in}} = 0.2$ then $\beta = 2.125$. Or, if $\beta = 2$ then $\frac{\Delta E}{E_{in}} = 0.167$.

A program to find the force would go as follows:

Input: k, $\frac{\Delta E}{E_{in}}$

Compute $\beta = \frac{2.5}{1-\frac{\Delta E}{E_{in}}} - 1$

In the main program when a Defl is to be calculated, first let

```
DeflpreviousP = Deflprevious
Deflprevious = Defl
Defl = (calculation formula)
```

Then the force is calculated as follows

```
Function Force(Defl, Deflprevious, DeflpreviousP, Deflmax)
    If Defl > 0 then
        If Defl >= Deflprevious then
            $\text{Force} = k*\text{Defl}^{1.5}$
        Else
            If Delfprevious >= DeflpreviousP then Deflmax = Deflprevious
            (save Deflmax)
            $\text{kr} = k/\text{Deflmax}^{\text{Beta}-1.5}$
            $\text{Force} = \text{kr}*\text{Defl}^{\text{Beta}}$
        End if
    Else
        Force = 0
    End if
End Function
```

Appendix D. The Starting Point of the Diverge Guideways.

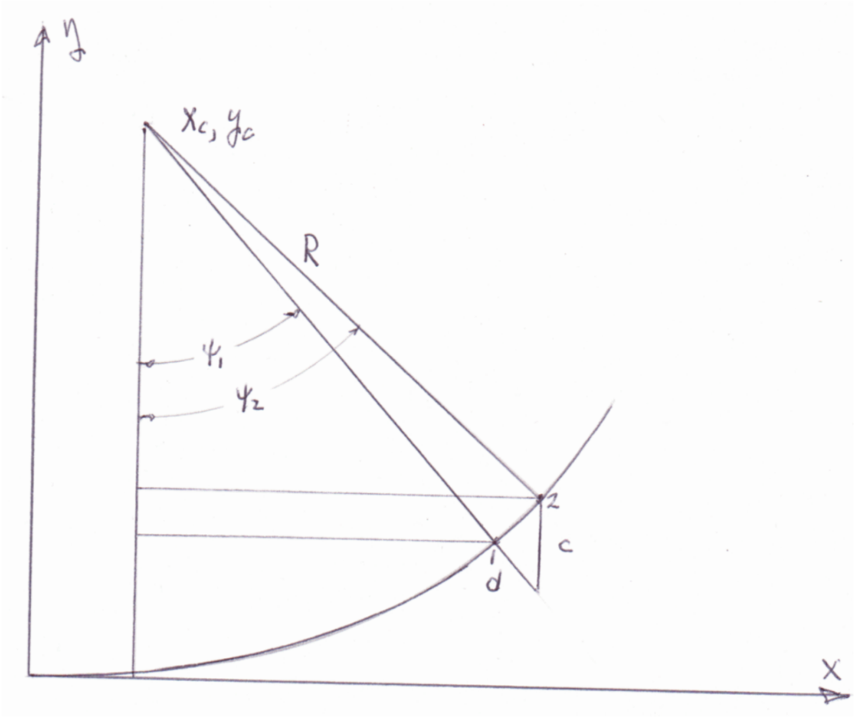

Figure D-1. Geometry of a curved guideway.

The y-distance from the x-axis to the point at which the flared guideways start is the point in Figure E-1 a distance *c* below the point 2. The distance *d* in Figure E-1 is the distance D_{main} in Section 8. Given *d* and Ψ_2, we need to find *c*. The coordinates of the points 1 and 2 are

$$x_1 = x_c + Rsin\Psi_1, y_1 = y_c - Rcos\Psi_1;\ x_2 = x_c + Rsin\Psi_2, y_2 = y_c - Rcos\Psi_2$$

$$c = y_c - Rcos\Psi_2 - [y_c - Rcos\Psi_1 - dcos\Psi_1] = R(cos\Psi_1 - cos\Psi_2) + dcos\Psi_1$$

$$dsin\Psi_1 = x_c + Rsin\Psi_2 - (x_c + Rsin\Psi_1) = R(sin\Psi_2 - sin\Psi_1)$$

Let $\Psi_1 = \Psi_2 - \Delta\Psi$. Then, using trigonometric identities,

$$c = R(cos\Psi_2 cos\Delta\Psi + \sin\Psi_2 \sin\Delta\Psi - cos\Psi_2) + d(cos\Psi_2 cos\Delta\Psi + \sin\Psi_2 \sin\Delta\Psi)$$

$$d(sin\Psi_2 cos\Delta\Psi - \cos\Psi_2 sin\Delta\Psi) = R(sin\Psi_2 - sin\Psi_2 cos\Delta\Psi + \cos\Psi_2 sin\Delta\Psi)$$

Solve the second of these equations for *R* and substitute into the first, letting $C \equiv cos, S \equiv Sin$. Then

$$\frac{c}{d} = (C\Psi_2 C\Delta\Psi + S\Psi_2 S\Delta\Psi - C\Psi_2)\frac{(S\Psi_2 C\Delta\Psi - C\Psi_2 S\Delta\Psi)}{(S\Psi_2 - S\Psi_2 C\Delta\Psi + C\Psi_2 S\Delta\Psi)} + C\Psi_2 C\Delta\Psi + S\Psi_2 S\Delta\Psi$$

$$
\begin{aligned}
&= \frac{C\Psi_2 C\Delta\Psi(S\Psi_2 C\Delta\Psi - \mathrm{C}\Psi_2 S\Delta\Psi) + \mathrm{S}\Psi_2 S\Delta\Psi(S\Psi_2 C\Delta\Psi - \mathrm{C}\Psi_2 S\Delta\Psi)}{S\Psi_2 - S\Psi_2 C\Delta\Psi + \mathrm{C}\Psi_2 S\Delta\Psi} \\
&\quad + \frac{-S\Psi_2 C\Delta\Psi(C\Psi_2 \mathrm{C}\Delta\Psi + \mathrm{S}\,\Psi_2 \mathrm{S}\Delta\Psi) + \mathrm{C}\Psi_2 S\Delta\Psi(C\Psi_2 \mathrm{C}\Delta\Psi + \mathrm{S}\,\Psi_2 \mathrm{S}\Delta\Psi)}{S\Psi_2 - S\Psi_2 C\Delta\Psi + \mathrm{C}\Psi_2 S\Delta\Psi} \\
&\quad + \frac{-\mathrm{C}\Psi_2(S\Psi_2 C\Delta\Psi - \mathrm{C}\Psi_2 S\Delta\Psi) + S\Psi_2(C\Psi_2 \mathrm{C}\Delta\Psi + \mathrm{S}\,\Psi_2 \mathrm{S}\Delta\Psi)}{S\Psi_2 - S\Psi_2 C\Delta\Psi + \mathrm{C}\Psi_2 S\Delta\Psi} \\
&= \frac{S\Delta\Psi}{S\Psi_2(1 - C\Delta\Psi) + \mathrm{C}\Psi_2 S\Delta\Psi} \cong \frac{1}{\mathrm{C}\Psi_2 + \dfrac{S\Psi_2 \Delta\Psi^2}{2\Delta\Psi\left(1 - \frac{1}{6}\Delta\Psi^2\right)}} \cong \frac{1}{\mathrm{C}\Psi_2 + \frac{1}{2} S\Psi_2 \Delta\Psi}
\end{aligned}
$$

Note that

$$
d = R\frac{(sin\Psi_2 - sin\Psi_2 cos\Delta\Psi + \cos\Psi_2 sin\Delta\Psi)}{(sin\Psi_2 cos\Delta\Psi - \cos\Psi_2 sin\Delta\Psi)} \cong R\frac{sin\Psi_2 \frac{1}{2}\Delta\Psi^2 + \cos\Psi_2 \Delta\Psi}{sin\Psi_2 - \Delta\Psi\cos\Psi_2} \cong \frac{R\Delta\Psi}{tan\Psi_2}
$$

in which $R = \frac{V^2}{A_l}$. Assume V = 13.5 m/s and A_l = 0.2g = 1.96 m/s^2. Then R = 93 m. We take d = 0.1m. Then

$$
\Delta\Psi = \frac{d}{R} tan\Psi_2 = 0.00108 tan\Psi_2
$$

We are interested in Ψ_2 at the point in a diverge where the inner running surfaces start. At this point, Ψ_2 is well under 45°, hence $\Delta\Psi < 0.001\ rad$. If $\Psi_2$were as much as 10° we have

$$
\frac{c}{d} = \frac{1}{\mathrm{C}\Psi_2 + \frac{1}{2} S\Psi_2 \Delta\Psi} = \frac{1}{0.985 + (0.087)(0.001)} \cong \frac{1}{0.985 + 0.0001}
$$

Thus, we can, with little error, take

$$
c = \frac{d}{cos\Psi_2},
$$

which is the result we need.

Appendix E. The Program.

```
Module InputData
    'This module inputs data needed to study the lateral dynamics of an ITNS
vehicle.
    'Units are feet, pounds, seconds
    Public Const c_g As Double = 32.174               'acceleration of
gravity, ft/sec^2
    Public Const c_DegperRad As Double = 180 / Math.PI
    Public Const c_Speed As Double = 35 * (88 / 60)   'line speed, ft/s
    Public Const c_Jn As Double = c_g / 4             'comfort jerk
    Public Const c_Bank As Double = 0                 'superelevation angle
    Public Const c_Al As Double = c_g / 5             'comfort lateral
acceleration
    Public Const c_J2V3 As Double = 0.5 * c_Jn / c_Speed ^ 3

    Public Const c_ChannelWidth As Double = 21 / 12   'distance between left
and right running surfaces
    Public Const c_HalfChWidth As Double = 0.5 * c_ChannelWidth
    Public Const c_SwxRailGap As Double = 4.5 / 12    'distance between main
and switch running surfaces
    Public Const c_SwxFlareLength As Double = 24      'length of flared
section of swith rail
    Public Const c_MainFlareLength As Double = 36     'length of flared
section in inner rail surfaces
    Public Const c_SwxFlareOffSet As Double = 1 / 12  'offset of end of
switch flare section
    Public Const c_mainFlareOffSet As Double = 1.5 / 12 'offset of end of
main flare section
    Public Const c_VehicleWeight As Double = 1200     'lb

    Public Const c_LIMWeight As Double = 400          'lb
    Public Const c_LIMRadiusGyration As Double = 1    'ft
    Public Const c_LIMYawInertia As Double = c_LIMWeight *
c_LIMRadiusGyration ^ 2 / c_g

    Public Const c_PassengerWeight As Double = 500    'lb
    Public Const c_PassStiffness As Double = 4800     'spring constant of
passenger suspension, lb/ft
    Public Const c_PassDamp As Double = 0.8           'dimensionless damping
constant of passenger system
    Public Const c_PassengerOffset As Double = -20 / 12   'ft

    Public Const c_YawRadiusGyration As Double = 2.5  'ft
    Public Const c_RollRadiusGyration As Double = 2.0 'ft
    Public Const c_YawInertia As Double = c_VehicleWeight *
c_YawRadiusGyration ^ 2 / c_g
    Public Const c_RollInertia As Double = c_VehicleWeight *
c_RollRadiusGyration ^ 2 / c_g
    'See paper "Deflection of Running Surface"
    Public Const c_Guage As Double = (22 - 2 * 3.75) / 12  'distance between
main-tire loads, ft
    Public Const c_Friction As Double = 0.25          'fraction of normal
force
    Public Const c_RadiusMainTire As Double = 0.5 * 13.25 / 12  'ft
    Public Const c_AirDensity As Double = 0.075     'weight density of air,
lb/ft^3
    Public Const c_WindDirection As Double = -1     'dimensionless
    Public Const c_WindSpeed As Double = 44 * c_WindDirection   'ft/sec
    Public Const c_CdFront As Double = 0.7
    Public Const c_CdSide As Double = 0.8
    Public Const c_SideArea As Double = 40       'ft^2
    Public Const c_FrontArea As Double = 25      'ft^2
    Public Const C_AirDrag As Double = (c_AirDensity / 2 / c_g) * c_Speed ^ 2
* c_CdFront * c_FrontArea  'lb
    Public Const c_WindForce As Double = c_WindDirection * (c_AirDensity / 2
/ c_g) * c_WindSpeed ^ 2 * c_CdSide * c_SideArea  'lb
    Public Const c_Zwind As Double = 57.375 / 12        'ft
    Public Const c_aRoad As Double = 0.005    'rolling resistance,
dimensionless
    Public Const c_bRoad As Double = 0.0005   'rolling resistance
proportional to speed, s/m

    'In these body coordinates x points forward, y points to the left, and z
points upward
    'x = 0 at the rear main axle, y = 0 at the center of the vehicle, and z =
0 at the height of the main axles.
    Public Const c_WB = 82 / 12                    'the distance between
front and rear main-wheel axles
    Public Const c_Xcg As Double = 0.45 * c_WB     'x-position of cg of empty
vehicle forward of rear main axle
    Public Const c_Xpass As Double = 0.2 * c_WB    'x-position of the
passenger
    Public Const c_Zcg As Double = 27.375 / 12     'z-position of cg of empty
vehicle above main axle
    Public Const c_Zcover As Double = 25 / 12      'z-position of upper edge
of cover above main axle
    Public Const c_Zpassenger As Double = 57 / 12  'z-position of passenger
midsection above main axle

    'Positions of main side wheels from main rear axle
    Public Const c_Xuf As Double = c_WB          'x-position of upper forward
wheel (Wheel Base)
    Public Const c_Xub As Double = 0             'x-position of upper back
wheel
    Public Const c_Xlf As Double = 72 / 12       'x-position of lower forward
wheel
    Public Const c_Xlb As Double = 10 / 12       'x-position of lower back
wheel

    Public Const c_Yl As Double = c_HalfChWidth  'y-position of left side
wheels
    Public Const c_Yr As Double = -c_Yl          'y-position of right side
wheels

    Public Const c_Zu As Double = 21.375 / 12    'z-position of upper side
wheels above main axle
    Public Const c_Zl As Double = -4.625 / 12    'z-position of lower side
wheels

    'Positions of switch wheels
    Public Const c_Xsf As Double = 72 / 12       'x-position of forward
switch wheels
    Public Const c_Xsb As Double = 10 / 72       'x-position of back switch
wheels
    Public Const c_Ysl As Double = c_Yl - c_SwxRailGap  'y-position of left
switch wheels
    Public Const c_Ysr As Double = -c_Ysl        'y-position of right switch
wheels
    Public Const c_Zs As Double = 10.375 / 12    'z-position of switch wheels
above main axle

    'Positions of LIM attachments
    Public Const c_XLIMf As Double = 61 / 12     'ft ahead of the rear axle
    Public Const c_XLIMb As Double = 21 / 12
    Public Const c_Zlim As Double = -4.625 / 12
    Public Const c_DZ As Double = c_Zlim - c_Zcg
    Public Const c_DX As Double = c_XLIMf - c_WB / 2

    Public Const c_kmain As Double = 3000 * 12             'lb/ft
    Public Const c_kUpper As Double = 60000 * 12 ^ 1.5     'lb/ft^1.5
    Public Const c_kLower As Double = 110000 * 12 ^ 1.5    'lb/ft^1.5
    Public Const c_kswitch As Double = 120000 * 12 ^ 1.5   'lb/ft^1.5
    Public Const c_kLIM As Double = 2400                   'lb/ft
    Public Const c_Lseat As Double = 17 / 12               'seat height, ft
    Public Const c_EI As Double = 400                      'column stiffness,
lb-ft^2
    Public Const c_kSeat As Double = 3 * c_EI / c_Lseat ^ 3 'seat stiffness,
lb/ft
    Public Const c_seatDamping As Double = 0.7
    Public Const c_seatFrequencySq As Double = c_kSeat * c_g /
c_PassengerWeight
    Public Const c_EnergyLoss As Double = 0.25
    Public Const c_Beta As Double = 2.5 / (1 - c_EnergyLoss) - 1
    Public Const c_Gamma As Double = 2 / (1 - c_EnergyLoss) - 1
    Public Const c_Tolerance As Double = 0 '0.04 / 12   'slop between side
tire and running surface

    Public Const c_sBegin As Double = -150

End Module

Imports System
Imports System.Diagnostics

Public Class MainForm
    Public scaleX As Single = 2.5  'sets the size of the action on the screen
    Public scaleY As Single = scaleX
    Public scaleYRS As Single = 2 * scaleY  'to expand run surface
    Public scaleF As Single = 0.05F
    Public scaleA As Single = 600
    Public scaleR As Single = 2000
    Public scaleC As Single = 5000
    Public x0 As Single = 400 'locates the action in the x-direction
    Public y0 As Single = 600 'locates the action in the y-direction
    Public xGraph As Single   'x graph coordinate
    Public yGraph As Single   'y graph coordinate
    Public An As Double = c_g * Math.Tan(c_Bank) + c_Al / Math.Cos(c_Bank)
'comfort horizontal acceleration
    Public s1 As Double = c_Speed * An / c_Jn
    Public Psi1 As Double = 0.5 * c_Jn * s1 ^ 2 / c_Speed ^ 3
    Public y1 As Double = (1 - Psi1 ^ 2 / 14) * s1 * Psi1 / 3
    Public x1 As Double = (1 - Psi1 ^ 2 / 10) * s1
    Public R As Double = c_Speed ^ 2 / An     'radius of curvature in
constant-curvature section
    Public xc As Double = x1 - R * Math.Sin(Psi1)
    Public yc As Double = y1 + R * Math.Cos(Psi1)
    Public Q As Double = yc - c_ChannelWidth - c_mainFlareOffSet
    Public cosThStart As Double = (Q + Math.Sqrt(Q ^ 2 - 4 * R *
c_mainFlareOffSet)) / 2 / R
    Public xStartInnerSurface As Double = xc + R * Math.Sqrt(1 - cosThStart ^
2)
    Public sStartInnerSurface As Double = s1 + R * (Math.Acos(cosThStart) -
Psi1)
    Public sMax As Double = s1 + R * (Math.PI / 4 - Psi1)  'up to Psi = 45
deg
    Public sEnd As Double = sStartInnerSurface + c_MainFlareLength +
c_SwxFlareLength
    Public yLIMr As Double  'sidewise motion at rear axle of LIM bogie, m
    Public yLIMf As Double  'sidewise motion at front axle of LIM bogie, m
    Public s As Double
    Public ds As Double = 0.002

    Dim objGraphics As System.Drawing.Graphics
    Sub RunSurface()
        Dim Psi, x, y As Double

        s = c_sBegin
        objGraphics = Me.CreateGraphics
        Do
            'Left Main Running Surface
            CurvedGuideway(Psi, x, y)
            y = y + c_HalfChWidth / Math.Cos(Psi)
            xGraph = x0 + scaleX * x
            yGraph = y0 - scaleYRS * y
            objGraphics.FillEllipse(Brushes.White, xGraph, yGraph, 2, 2)

            'Right Main Running Surface
            x = s
            y = -c_HalfChWidth
            xGraph = x0 + scaleX * x
            yGraph = y0 - scaleYRS * y
            objGraphics.FillEllipse(Brushes.White, xGraph, yGraph, 2, 2)

            'Left Switch Rail Running Surface
            CurvedGuideway(Psi, x, y)
            y = y + (c_HalfChWidth - c_SwxRailGap - SwxFlare(s)) /
Math.Cos(Psi)
            xGraph = x0 + scaleX * x
            yGraph = y0 - scaleYRS * y
            If s > -c_SwxFlareLength And s <= sEnd Then
                objGraphics.FillEllipse(Brushes.Red, xGraph, yGraph, 2, 2)
            End If
            'Right Switch Rail Running Surface
            x = s
            y = -c_HalfChWidth + c_SwxRailGap + SwxFlare(s)
            xGraph = x0 + scaleX * x
            yGraph = y0 - scaleYRS * y
            If s > -c_SwxFlareLength And s <= xStartInnerSurface +
c_MainFlareLength + c_SwxFlareLength Then
                objGraphics.FillEllipse(Brushes.Red, xGraph, yGraph, 2, 2)
            End If

            'Right Run Surface of Left Branch
            CurvedGuideway(Psi, x, y)
            y = y - (c_HalfChWidth + MainFlare(s)) / Math.Cos(Psi)
            xGraph = x0 + scaleX * x
            yGraph = y0 - scaleYRS * y
            If s >= sStartInnerSurface Then
                objGraphics.FillEllipse(Brushes.White, xGraph, yGraph, 2, 2)
            End If

            'Left Run Surface of Right Branch
            x = s
            If s >= sStartInnerSurface Then
                y = c_HalfChWidth + MainFlare(s)
            Else
                y = 0
            End If
            xGraph = x0 + scaleX * x
            yGraph = y0 - scaleYRS * y
            If x > xStartInnerSurface And y > 0 Then
                objGraphics.FillEllipse(Brushes.White, xGraph, yGraph, 2, 2)
            End If

            s = s + ds
        Loop Until s > sMax + 200
        objGraphics.Dispose()
    End Sub

    Sub LateralMotion()
        Dim OnOff As Double = 1

        Dim Psi As Double
        Dim dt = ds / c_Speed
        Dim DeflUFL, DeflUFR, DeflUBL, DeflUBR, DeflLFL, DeflLFR, DeflLBL,
DeflLBR As Double
        Dim DeflSFL, DeflSBL As Double
        Dim DuflP, DuflPP, DufrP, DufrPP, DublP, DublPP, DubrP, DubrPP As
Double
        Dim DlflP, DlflPP, DlfrP, DlfrPP, DlblP, DlblPP, DlbrP, DlbrPP As
Double
        Dim DsflP, DsflPP, DsblP, DsblPP As Double
        Dim mfrP, mfrPP, mflP, mflPP, mbrP, mbrPP, mblP, mblPP, MaxMfr,
MaxMfl, MaxMbr, MaxMbl As Double
        Dim MaxDufl, MaxDufr, MaxDubl, MaxDubr, MaxDlfl, MaxDlfr, MaxDlbl,
MaxDlbr As Double
        Dim MaxDsfl, MaxDsbl As Double
        Dim DeflMFR, DeflMFL, DeflMBR, DeflMBL As Double
        Dim ForceUFL, ForceUFR, ForceUBL, ForceUBR, ForceLFL, ForceLFR,
ForceLBL, ForceLBR As Double
        Dim ForceSFL, ForceSFR, ForceSBL, ForceSBR As Double    'Switch wheel
forces
        Dim ForceMFL, ForceMFR, ForceMBL, ForceMBR As Double    'Main support
tire forces.
        Dim LeftSideTireForces, RightSideTireForces, SwitchTireForces As
Double
        Dim SideTireForces, sideVelocity, updownCrabAngle As Double
        Dim MainTireFrictionForces, FrictionF, FrictionB As Double
        Dim BogieFrictionForce, BogieChassisForce As Double

        Dim YawMoments, RollMoments As Double
        Dim YawMuf, YawMub, YawMlf, YawMlb, YawMsf, YawMsb, YawMfriction As
Double
        Dim RollMl, RollMr, RollMs, RollMmain, RollMfriction,
RollMSideFriction, RollExternal As Double

        Dim Switch As String            'Direction switch is thrown, Left or
Right

        Dim yMCAccel As Double = 0      'Lateral acceleration of vehicle
c.g., + to left
        Dim yMCAccelMax As Double = 0   'Maximum lateral acceleration.
        Dim yMCAccelOld As Double = 0   'Previous value
        Dim yMCRate As Double = 0       'Lateral velocity of vehicle
        Dim yMCRateOld As Double = 0    'Previous value
        Dim yMC As Double = 0           'Lateral positon of vehicle MC with
respect to guideway
        Dim yMCmax As Double            'Maximum lateral motion
        Dim syMCMax As Double = 0       'Value of s at yMCMax
        Dim xGdwyCenter As Double       'x-coordinate of position of center
of guideway at s
        Dim yGdwyCenter As Double       'y-coordinate of position of center
of guideway at s

        Dim YawAccel As Double = 0      'Yaw acceleration of vehicle, +
counterclockwise
        Dim YawAccelOld As Double = 0   'Previous value
        Dim YawRate As Double = 0       'Yaw velocity of vehicle
        Dim YawRateOld As Double = 0    'Previous value
        Dim Yaw As Double = 0           'Angle of vehicle with respect to
guideway, + to left
        Dim YawMax As Double = 0        'Maximum yaw angle
        Dim sYawMax As Double = 0       'Value of s at YawMax

        Dim RollAccel As Double = 0     'Roll acceleration of vehicle, + to
right
        Dim RollAccelOld As Double = 0  'Previous vallue
        Dim RollRate As Double = 0      'Roll rate of vehicle
        Dim RollRateOld As Double = 0   'Previous value
        Dim Roll As Double = 0          'Roll angle of vehicle with respect
to guideway, + to right
        Dim RollMax As Double = 0
        Dim sRollMax As Double = 0
        Dim yCover As Double = 0         'Lateral movement of chassis at
position of top of guideway cover
        Dim yCoverMax As Double = 0      'Maximum value of yCover
        Dim sAtCoverMax As Double        's at yCoverMax

        Dim yPassAccel As Double = 0     'lateral acceleration of passenger
        Dim yPassAccelOld As Double = 0
        Dim yPassAccelinG As Double = 0
        Dim yPassRate As Double = 0
        Dim yPassRateOld As Double = 0
        Dim yPass As Double = 0
        Dim SeatShift As Double = 0
        Dim MaxSeatShift As Double = 0
        Dim RadFreq As Double = Math.Sqrt(c_PassStiffness * c_g /
c_PassengerWeight)

        Dim yLIMAccel As Double = 0
        Dim yLIMAccelOld As Double = 0
        Dim yLIMRate As Double = 0
        Dim yLIMRateOld As Double = 0
        Dim yLIMmc As Double = 0
        Dim YawLIMAccel As Double = 0
        Dim YawLIMAccelOld As Double = 0
        Dim YawLIMRate As Double = 0
        Dim YawLIMRateOld As Double = 0
        Dim YawLIM As Double = 0

        Dim Counter As Integer = 0
        Dim Flag As Integer = 0
        Dim sPositiveDeflection As Double   'Value of s when upper right
forward tire hits running surface.

        Dim DeflUFLmax As Double = 0     'Tire deflections
        Dim DeflUBLmax As Double = 0
        Dim DeflLFLmax As Double = 0
        Dim DeflLBLmax As Double = 0

        Dim DeflUFRmax As Double = 0
        Dim DeflUBRmax As Double = 0
        Dim DeflLFRmax As Double = 0
        Dim DeflLBRmax As Double = 0

        Dim DeflSFLmax As Double = 0
        Dim DeflSBLmax As Double = 0

        Dim ForceUFLmax As Double = 0    'Tire forces
        Dim ForceUBLmax As Double = 0
        Dim ForceLFLmax As Double = 0
        Dim ForceLBLmax As Double = 0

        Dim ForceUFRmax As Double = 0
        Dim ForceUBRmax As Double = 0
        Dim ForceLFRmax As Double = 0
        Dim ForceLBRmax As Double = 0

        Dim ForceSFLmax As Double = 0
        Dim ForceSBLmax As Double = 0
        Dim YawLIMmoment As Double = 0
        Dim RollLIMmoment As Double = 0
        Dim BogieMoment As Double = 0
        Dim LIMNormalForceFactor As Double = 0
        Dim MaxChassisForce As Double = 0


        Dim sAtMaxUFR, sAtMaxSBL As Double      'positions of maximum forces
        Dim DeflUFRsStart As Double

        Dim yPassAccelMax As Double = 0
        Dim MaxSeatAccel As Double = 0
        Dim startInnerSurface As Integer = scaleX * xStartInnerSurface
        Dim xTire, yTire, yRail As Double

        Dim textOut As New System.IO.StreamWriter("c:/Simulation
Results/LateralMotionResults.txt")

        objGraphics = Me.CreateGraphics
        objGraphics.DrawLine(Pens.White, x0, 3 * y0, x0, 0)
'ordinate
        objGraphics.DrawLine(Pens.White, x0 - 600, y0, x0 + 1000, y0)
'absissa
        objGraphics.DrawLine(Pens.White, x0 + startInnerSurface, y0, x0 +
startInnerSurface, y0 - 250)   'marker at xStart
        objGraphics.DrawLine(Pens.Yellow, x0 - 600, y0 - scaleA / 5, x0 +
1000, y0 - scaleA / 5)         '0.25g line
        objGraphics.DrawLine(Pens.Yellow, x0 - 600, y0 + scaleA / 5, x0 +
1000, y0 + scaleA / 5)         '0.25g line
        objGraphics.DrawLine(Pens.Red, x0 - 600, y0 - scaleF * 2000, x0 +
1000, y0 - scaleF * 2000)      '3000 lb line
        objGraphics.DrawLine(Pens.Red, x0 - 600, y0 + scaleF * 2000, x0 +
1000, y0 + scaleF * 2000)      '3000 lb line
        For i = -200 To 400 Step 10
            objGraphics.DrawLine(Pens.White, x0 + scaleX * i, y0, x0 + scaleX
* i, y0 - 10)
        Next
        objGraphics.DrawString(" y At Cover ", Me.Font,
System.Drawing.Brushes.White, 300, 0)
        objGraphics.DrawString(" y Seat Acceleration ", Me.Font,
System.Drawing.Brushes.Thistle, 300, 20)
        objGraphics.DrawString(" y Passenger Acceleration ", Me.Font,
System.Drawing.Brushes.Yellow, 300, 40)
        objGraphics.DrawString(" Force Swx Front Left ", Me.Font,
System.Drawing.Brushes.Red, 300, 60)
        objGraphics.DrawString(" Force Swx Back Left ", Me.Font,
System.Drawing.Brushes.Pink, 300, 80)
        objGraphics.DrawString(" Force Upper Forward Right ", Me.Font,
System.Drawing.Brushes.Orange, 300, 100)
        objGraphics.DrawString(" Force Upper Back Right ", Me.Font,
System.Drawing.Brushes.GreenYellow, 300, 120)
        objGraphics.DrawString(" Force Lower Forward Left ", Me.Font,
System.Drawing.Brushes.Turquoise, 300, 140)
        objGraphics.DrawString(" Force Lower Back Left ", Me.Font,
System.Drawing.Brushes.Wheat, 300, 160)
```

```
        objGraphics.DrawString(" Roll ", Me.Font,
System.Drawing.Brushes.PaleVioletRed, 300, 180)

        objGraphics.DrawString(" Horizontal Red Lines are at 2000 lb ",
Me.Font, System.Drawing.Brushes.Red, 700, 20)
        objGraphics.DrawString(" Horizontal Yellow Lines are at 0.2 g ",
Me.Font, System.Drawing.Brushes.Yellow, 700, 40)
        objGraphics.DrawString(" Closely spaced short vertical lines are 10
ft apart ", Me.Font, System.Drawing.Brushes.White, 700, 60)
        objGraphics.DrawString(" Single vertical line marks xStart", Me.Font,
System.Drawing.Brushes.White, 700, 80)
        objGraphics.DrawString(" Single top-to-bottom vertical line marks s =
0", Me.Font, System.Drawing.Brushes.White, 700, 100)


        Switch = "Left"     'following code based on switching left as worst
case
        s = c_sBegin

        Do
            CurvedGuideway(Psi, xGdwyCenter, yGdwyCenter)

            'Tire deflections:
            'Upper front left tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xuf,
c_Yl, c_Zu)
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xuf,
c_Yl, c_Zu)
            yRail = yCL(xTire) + c_HalfChWidth / cosPsi(xTire) - c_Tolerance
            DuflPP = DuflP
            DuflP = DeflUFL
            DeflUFL = (yTire - yRail) * cosPsi(xTire)  '0 if < 0

            'Upper back left tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xub,
c_Yl, c_Zu)
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xub,
c_Yl, c_Zu)
            yRail = yCL(xTire) + c_HalfChWidth / cosPsi(xTire) - c_Tolerance
            DublPP = DublP
            DublP = DeflUBL
            DeflUBL = (yTire - yRail) * cosPsi(xTire)  '0 if < 0

            'Lower front left tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xlf,
c_Yl, c_Zl)
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xlf,
c_Yl, c_Zl)
            yRail = yCL(xTire) + c_HalfChWidth / cosPsi(xTire) - c_Tolerance
            DlflPP = DlflP
            DlflP = DeflLFL
            DeflLFL = (yTire - yRail) * cosPsi(xTire)  '0 if < 0

            'Lower back left tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xlb,
c_Yl, c_Zl)
```

```
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xlb,
c_Yl, c_Zl)
            yRail = yCL(xTire) + c_HalfChWidth / cosPsi(xTire) - c_Tolerance
            DlblPP = DlblP
            DlblP = DeflLBL
            DeflLBL = (yTire - yRail) * cosPsi(xTire)  '0 if < 0

            'Upper front right tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xuf,
c_Yr, c_Zu)
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xuf,
c_Yr, c_Zu)
            If xTire < xStartInnerSurface Then
                yRail = -c_HalfChWidth + c_Tolerance
            Else
                yRail = yc - R * cosPsi(xTire) - (c_HalfChWidth +
MainFlare(s)) / cosPsi(xTire) + c_Tolerance
            End If
            DufrPP = DufrP
            DufrP = DeflUFR
            DeflUFR = (yRail - yTire) * cosPsi(xTire)
            If s >= sStartInnerSurface And s < sStartInnerSurface + ds Then
                DeflUFRsStart = DeflUFR
            End If
            If s > sStartInnerSurface And DeflUFR > 0 And Flag = 0 Then
                sPositiveDeflection = s + c_Xuf - c_Xcg
                Flag = 1
            End If

            'Upper back right tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xub,
c_Yr, c_Zu)
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xub,
c_Yr, c_Zu)
            If xTire < xStartInnerSurface Then
                yRail = -c_HalfChWidth + c_Tolerance
            Else
                yRail = yc - R * cosPsi(xTire) - (c_HalfChWidth +
MainFlare(s)) / cosPsi(xTire) + c_Tolerance
            End If
            DubrPP = DubrP
            DubrP = DeflUBR
            DeflUBR = (yRail - yTire) * cosPsi(xTire)

            'Lower front right tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xlf,
c_Yr, c_Zl)
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xlf,
c_Yr, c_Zl)
            If xTire < xStartInnerSurface Then
                yRail = -c_HalfChWidth + c_Tolerance
            Else
                yRail = yc - R * cosPsi(xTire) - (c_HalfChWidth +
MainFlare(s)) / cosPsi(xTire) + c_Tolerance
            End If
            DlfrPP = DlfrP
            DlfrP = DeflLFR
            DeflLFR = (yRail - yTire) * cosPsi(xTire)

            'Lower back right tire
            yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xlb,
c_Yr, c_Zl)
            xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xlb,
c_Yr, c_Zl)
            If xTire < xStartInnerSurface Then
                yRail = -c_HalfChWidth + c_Tolerance
            Else
                yRail = yc - R * cosPsi(xTire) - (c_HalfChWidth +
MainFlare(s)) / cosPsi(xTire) + c_Tolerance
            End If
            DlbrPP = DlbrP
            DlbrP = DeflLBR
            DeflLBR = (yRail - yTire) * cosPsi(xTire)

            'Switch front left tire
            If s >= -c_SwxFlareLength And s < sEnd Then
                yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xsf,
c_Ysl, c_Zs)
                xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xsf,
c_Ysl, c_Zs)
                yRail = yCL(xTire) + (c_HalfChWidth - c_SwxRailGap -
SwxFlare(s)) / cosPsi(xTire) + c_Tolerance
                DsflPP = DsflP
                DsflP = DeflSFL
                DeflSFL = (yRail - yTire) * cosPsi(xTire)
            Else
                DeflSFL = 0
            End If

            'Switch back left tire
            If s >= -c_SwxFlareLength And s < sEnd Then
                yTire = yGdwyCenter + yTireLocal(Psi, yMC, Yaw, Roll, c_Xsb,
c_Ysl, c_Zs)
                xTire = xGdwyCenter + xTireLocal(Psi, yMC, Yaw, Roll, c_Xsb,
c_Ysl, c_Zs)
                yRail = yCL(xTire) + (c_HalfChWidth - c_SwxRailGap -
SwxFlare(s)) / cosPsi(xTire) + c_Tolerance
                DsblPP = DsblP
                DsblP = DeflSBL
                DeflSBL = (yRail - yTire) * cosPsi(xTire)
            Else
                DeflSBL = 0
            End If

            If s > 0 Then
                If DeflUFL > DeflUFLmax Then DeflUFLmax = DeflUFL
                If DeflUBL > DeflUBLmax Then DeflUBLmax = DeflUBL
                If DeflLFL > DeflLFLmax Then DeflLFLmax = DeflLFL
                If DeflLBL > DeflLBLmax Then DeflLBLmax = DeflLBL

                If DeflUFR > DeflUFRmax Then DeflUFRmax = DeflUFR
                If DeflUBR > DeflUBRmax Then DeflUBRmax = DeflUBR
                If DeflLFR > DeflLFRmax Then DeflLFRmax = DeflLFR
                If DeflLBR > DeflLBRmax Then DeflLBRmax = DeflLBR
                If DeflSFL > DeflSFLmax Then DeflSFLmax = DeflSFL
                If DeflSBL > DeflSBLmax Then DeflSBLmax = DeflSBL
            End If

            'Main Vehicle-Support Tire deflections
            'Deflection of front right tire
            DeflMFR = 0.5 * c_g * ((c_VehicleWeight * c_Xcg +
c_PassengerWeight * c_Xpass) / c_WB / c_kmain + c_Guage * Roll)
            'Deflection of front left tire
            DeflMFL = 0.5 * c_g * ((c_VehicleWeight * c_Xcg +
c_PassengerWeight * c_Xpass) / c_WB / c_kmain - c_Guage * Roll)
            'Defection of back right tire
            DeflMBR = 0.5 * c_g * ((c_VehicleWeight * (c_WB - c_Xcg) +
c_PassengerWeight * (c_WB - c_Xpass)) / c_WB / c_kmain + c_Guage * Roll)
            'Defection of back left tire
            DeflMBL = 0.5 * c_g * ((c_VehicleWeight * (c_WB - c_Xcg) +
c_PassengerWeight * (c_WB - c_Xpass)) / c_WB / c_kmain - c_Guage * Roll)

            'Side Tire forces on the vehicle, left -, right +
            If DeflUFL > 0 Then     'Upper forward left
                ForceUFL = -SideTireForce(c_kUpper, DeflUFL, DuflP, DuflPP,
MaxDufl)
            Else
                ForceUFL = 0
            End If

            If DeflUBL > 0 Then     'Upper back left
                ForceUBL = -SideTireForce(c_kUpper, DeflUBL, DublP, DublPP,
MaxDubl)
            Else
                ForceUBL = 0
            End If

            If DeflLFL > 0 Then     'Lower forward left
                ForceLFL = -SideTireForce(c_kLower, DeflLFL, DlflP, DlflPP,
MaxDlfl)
            Else
                ForceLFL = 0
            End If

            If DeflLBL > 0 Then     'Lower back left
                ForceLBL = -SideTireForce(c_kLower, DeflLBL, DlblP, DlblPP,
MaxDlbl)
            Else
                ForceLBL = 0
            End If

            If DeflUFR > 0 Then     'Upper forward right
                ForceUFR = SideTireForce(c_kUpper, DeflUFR, DufrP, DufrPP,
MaxDufr)
            Else
                ForceUFR = 0
            End If

            If DeflUBR > 0 Then     'Upper back right
```

```
                ForceUBR = SideTireForce(c_kUpper, DeflUBR, DubrP, DubrPP,
MaxDubr)
            Else
                ForceUBR = 0
            End If

            If DeflLFR > 0 Then     'Lower forward right
                ForceLFR = SideTireForce(c_kLower, DeflLFR, DlfrP, DlfrPP,
MaxDlfr)
            Else
                ForceLFR = 0
            End If

            If DeflLBR > 0 Then     'Lower back right
                ForceLBR = SideTireForce(c_kLower, DeflLBR, DlbrP, DlbrPP,
MaxDlbr)
            Else
                ForceLBR = 0
            End If

            'Switch Tire Forces, Left +, Right -
            If DeflSFL > 0 Then     'Switch forward left
                ForceSFL = SideTireForce(c_kswitch, DeflSFL, DsflP, DsflPP,
MaxDsfl)
            Else
                ForceSFL = 0
            End If

            If DeflSBL > 0 Then     'Switch back left
                ForceSBL = SideTireForce(c_kswitch, DeflSBL, DsblP, DsblPP,
MaxDsbl)
            Else
                ForceSBL = 0
            End If

            If s >= -c_SwxFlareLength Then
                If ForceUFL < ForceUFLmax Then ForceUFLmax = ForceUFL
                If ForceUBL < ForceUBLmax Then ForceUBLmax = ForceUBL
                If ForceLFL < ForceLFLmax Then ForceLFLmax = ForceLFL
                If ForceLBL < ForceLBLmax Then ForceLBLmax = ForceLBL
                If ForceUFR > ForceUFRmax Then
                    ForceUFRmax = ForceUFR
                    sAtMaxUFR = s
                End If
                If ForceUBR > ForceUBRmax Then ForceUBRmax = ForceUBR
                If ForceLFR > ForceLFRmax Then ForceLFRmax = ForceLFR
                If ForceLBR > ForceLBRmax Then ForceLBRmax = ForceLBR

                If ForceSFL > ForceSFLmax Then ForceSFLmax = ForceSFL
                If ForceSBL > ForceSBLmax Then
                    ForceSBLmax = ForceSBL
                    sAtMaxSBL = s
                End If
            End If

            'Main Tire Forces
            ForceMFL = MainTireForce(c_kmain, DeflMFL, mflP, mflPP, MaxMfl)
            ForceMFR = MainTireForce(c_kmain, DeflMFR, mfrP, mfrPP, MaxMfr)
            ForceMBL = MainTireForce(c_kmain, DeflMBL, mblP, mblPP, MaxMbl)
            ForceMBR = MainTireForce(c_kmain, DeflMBR, mbrP, mbrPP, MaxMbr)

            'Forward Main Tire Friction Forces
            sideVelocity = yMCRate + YawRate * (c_Xuf - c_Xcg) + RollRate *
(c_Zcg + c_RadiusMainTire)
            FrictionF = -c_Friction * (ForceMFL + ForceMFR) * (sideVelocity /
c_Speed + Yaw)

            'Back Main Tire Friction Forces
            sideVelocity = yMCRate + YawRate * (c_Xub - c_Xcg) + RollRate *
(c_Zcg + c_RadiusMainTire)
            FrictionB = -c_Friction * (ForceMBL + ForceMBR) * (sideVelocity /
c_Speed + Yaw)

            'Side Tire Friction Forces roll the vehicle
            updownCrabAngle = c_HalfChWidth * RollRate / c_Speed
            LeftSideTireForces = ForceUFL + ForceUBL + ForceLFL + ForceLBL
            RightSideTireForces = ForceUFR + ForceUBR + ForceLFR + ForceLBR
            RollMSideFriction = -c_HalfChWidth * c_Friction * updownCrabAngle
* (LeftSideTireForces + RightSideTireForces)

            'Forces on LIM Bogie
            LIMNormalForceFactor = c_Friction * (c_LIMWeight + C_AirDrag) /
(1 - c_aRoad - c_bRoad * c_Speed)
            BogieFrictionForce = LIMNormalForceFactor * (YawLIM - (yLIMRate -
c_DZ * RollRate) / c_Speed)
            BogieChassisForce = -2 * c_kLIM * (yLIMmc - yMC)
            BogieMoment = -(2 * c_kLIM * (YawLIM - Yaw) +
LIMNormalForceFactor * YawLIMRate / c_Speed) * c_DX ^ 2
            If Math.Abs(BogieChassisForce) > MaxChassisForce Then
MaxChassisForce = Math.Abs(BogieChassisForce)

            'Equations of motion
            SwitchTireForces = ForceSFL + ForceSBL + ForceSFR + ForceSBR
            SideTireForces = LeftSideTireForces + RightSideTireForces +
SwitchTireForces
            MainTireFrictionForces = FrictionF + FrictionB

            yMCAccelOld = yMCAccel
            yMCAccel = -c_Speed ^ 2 * Curvature() * OnOff
            yMCAccel = yMCAccel + (SideTireForces + MainTireFrictionForces +
c_WindForce - BogieChassisForce) * c_g / c_VehicleWeight
            If s > 0 And Math.Abs(yMCAccel) > yMCAccelMax Then yMCAccelMax =
Math.Abs(yMCAccel)
            yMCRateOld = yMCRate
            yMCRate = yMCRate + 0.5 * dt * (3 * yMCAccel - yMCAccelOld)
            yMC = yMC + 0.5 * dt * (yMCRate + yMCRateOld)
            If s > 0 And Math.Abs(yMCAccel) > yMCAccelMax Then yMCAccelMax =
Math.Abs(yMCAccel)
            If s > 0 And Math.Abs(yMC) > yMCmax Then
                yMCmax = Math.Abs(yMC)
                syMCMax = s
            End If
            YawMuf = (ForceUFR + ForceUFL) * (c_Xuf - c_Xcg)
            YawMub = (ForceUBR + ForceUBL) * (c_Xub - c_Xcg)
            YawMlf = (ForceLFR + ForceLFL) * (c_Xlf - c_Xcg)
            YawMlb = (ForceLBR + ForceLBL) * (c_Xlb - c_Xcg)
            YawMsf = (ForceSFR + ForceSFL) * (c_Xsf - c_Xcg)
            YawMsb = (ForceSBR + ForceSBL) * (c_Xsb - c_Xcg)
            YawMfriction = FrictionF * (c_Xuf - c_Xcg) - FrictionB * (c_Xub -
c_Xcg)
            YawLIMmoment = 2 * c_kLIM * (YawLIM - Yaw) * c_DX ^ 2
            YawMoments = YawMuf + YawMub + YawMlf + YawMlb + YawMsf + YawMsb
+ YawMfriction + YawLIMmoment
            YawAccelOld = YawAccel
            YawAccel = YawMoments / c_YawInertia
            YawRateOld = YawRate
            YawRate = YawRate + 0.5 * dt * (3 * YawAccel - YawAccelOld)
            Yaw = Yaw + 0.5 * dt * (YawRate + YawRateOld)
            If s > 0 And Math.Abs(Yaw) > YawMax Then
                YawMax = Math.Abs(Yaw)
                sYawMax = s
            End If

            RollMl = (ForceUFL + ForceUBL) * (c_Zcg - c_Zu) + (ForceLFL +
ForceLBL) * (c_Zcg - c_Zl)
            RollMr = (ForceUFR + ForceUBR) * (c_Zcg - c_Zu) + (ForceLFR +
ForceLBR) * (c_Zcg - c_Zl)
            RollMs = SwitchTireForces * (c_Zcg - c_Zs)
            RollMmain = (ForceMFL - ForceMFR + ForceMBL - ForceMBR) * c_Guage
/ 2
            RollMfriction = MainTireFrictionForces * (c_Zcg +
c_RadiusMainTire)
            RollExternal = -c_WindForce * (c_Zwind - c_Zcg) -
c_PassengerWeight * c_PassengerOffset
            RollLIMmoment = -BogieChassisForce * c_DZ
            RollMoments = RollMl + RollMr + RollMs + RollMmain +
RollMfriction + RollMSideFriction + RollExternal + RollLIMmoment
            RollAccelOld = RollAccel
            RollAccel = RollMoments / c_RollInertia
            RollRateOld = RollRate
            RollRate = RollRate + 0.5 * dt * (3 * RollAccel - RollAccelOld)
            Roll = Roll + 0.5 * dt * (RollRate + RollRateOld)
            If s > 0 And Math.Abs(Roll) > RollMax Then
                RollMax = Math.Abs(Roll)
                sRollMax = s
            End If

            yCover = yMC + (c_Zcg - c_Zcover) * Roll
            If s >= 0 And Math.Abs(yCover) > yCoverMax Then
                yCoverMax = Math.Abs(yCover)
                sAtCoverMax = s
            End If

            yPassAccelOld = yPassAccel
            yPassAccel = -c_seatFrequencySq * (yMC - (c_Zpassenger - c_Zcg) *
Roll) - 2 * c_seatDamping * Math.Sqrt(c_seatFrequencySq) * yPassRate
            yPassRateOld = yPassRate
            yPassRate = yPassRate + 0.5 * dt * (3 * yPassAccel -
yPassAccelOld)
```

```
            yPass = yPass + 0.5 * dt * (yPassRate + yPassRateOld)
            SeatShift = yPass - yMC + (c_Zpassenger - c_Zcg) * Roll
            If s > 0 And Math.Abs(SeatShift) > MaxSeatShift Then MaxSeatShift
= Math.Abs(SeatShift)
            If s > 0 And Math.Abs(yPassAccel) > yPassAccelMax Then
yPassAccelMax = Math.Abs(yPassAccel)

            yLIMAccelOld = yLIMAccel
            yLIMAccel = (BogieChassisForce + BogieFrictionForce) * c_g /
c_LIMWeight
            yLIMRateOld = yLIMRate
            yLIMRate = yLIMRate + 0.5 * dt * (3 * yLIMAccel - yLIMAccelOld)
            yLIMmc = yLIMmc + 0.5 * dt * (yLIMRate + yLIMRateOld)

            YawLIMAccelOld = YawLIMAccel
            YawLIMAccel = BogieMoment / c_LIMYawInertia
            YawLIMRateOld = YawLIMRate
            YawLIMRate = YawLIMRate + 0.5 * dt * (3 * YawLIMAccel -
YawLIMAccelOld)
            YawLIM = YawLIM + 0.5 * dt * (YawLIMRate + YawLIMRateOld)

            xGraph = x0 + scaleX * s
            yGraph = y0 - scaleC * yCover
            objGraphics.FillEllipse(Brushes.White, xGraph, yGraph, 2, 2)
            yGraph = y0 - scaleA * yMCAccel / c_g
            'objGraphics.FillEllipse(Brushes.Gold, xGraph, yGraph, 2, 2)
            yGraph = y0 - scaleC * SeatShift / c_g
            objGraphics.FillEllipse(Brushes.Green, xGraph, yGraph, 2, 2)
            yGraph = y0 - scaleA * yPassAccel / c_g
            'objGraphics.FillEllipse(Brushes.Yellow, xGraph, yGraph, 2, 2)
            yGraph = y0 - scaleF * ForceLFL
            'objGraphics.FillEllipse(Brushes.Turquoise, xGraph, yGraph, 2, 2)
'LFL
            yGraph = y0 - scaleF * ForceLBL
            'objGraphics.FillEllipse(Brushes.Wheat, xGraph, yGraph, 2, 2)
'LBL
            yGraph = y0 - scaleF * ForceUFL
            'objGraphics.FillEllipse(Brushes.Gold, xGraph, yGraph, 2, 2)
'UFL
            yGraph = y0 - scaleF * ForceUBL
            'objGraphics.FillEllipse(Brushes.LightCyan, xGraph, yGraph, 2, 2)
'UBL
            yGraph = y0 - scaleF * ForceLBR
            'objGraphics.FillEllipse(Brushes.Violet, xGraph, yGraph, 2, 2)
'LBR
            yGraph = y0 - scaleF * ForceLFR
            'objGraphics.FillEllipse(Brushes.Teal, xGraph, yGraph, 2, 2)
'LFR
            yGraph = y0 - scaleF * ForceUFR
            'objGraphics.FillEllipse(Brushes.Orange, xGraph, yGraph, 2, 2)
'UFR
            yGraph = y0 - scaleF * ForceUBR
            'objGraphics.FillEllipse(Brushes.GreenYellow, xGraph, yGraph, 2,
2)   'UBR
            yGraph = y0 - scaleF * ForceSFL
            'objGraphics.FillEllipse(Brushes.Red, xGraph, yGraph, 2, 2)
'SFL
```

```
            yGraph = y0 - scaleF * ForceSBL
            'objGraphics.FillEllipse(Brushes.Pink, xGraph, yGraph, 2, 2)
'SBL
            yGraph = y0 - scaleR * Roll
            'objGraphics.FillEllipse(Brushes.PaleVioletRed, xGraph, yGraph,
2, 2)   'Roll
            yGraph = y0 - scaleR * Yaw
            'objGraphics.FillEllipse(Brushes.Teal, xGraph, yGraph, 2, 2)
'Yaw
            yGraph = y0 - scaleR * yMC
            'objGraphics.FillEllipse(Brushes.Salmon, xGraph, yGraph, 2, 2)
'yMC
            yGraph = y0 - 10 * scaleR * DeflUFR
            If DeflUFR > 0 Then
                'objGraphics.FillEllipse(Brushes.Red, xGraph, yGraph, 2, 2)
'yMC
            End If
            s = s + ds
        Loop Until s > sMax + 50

        textOut.WriteLine(" LATERAL MOTION RESULTS")
        textOut.WriteLine(" Date: " & Date.Now & " Computational distance
step: " & ds)
        textOut.WriteLine(" Positive directions: forward, left, up")
        textOut.WriteLine(" Guideway design speed and vehicle speed, mi/hr "
& c_Speed * 15 / 22)
        textOut.WriteLine(" Tolerance, i.e., distance between tire and rail,
in " & c_Tolerance * 12)
        textOut.WriteLine(" sStart @ Diverge Junction, ft:" &
FormatNumber(sStartInnerSurface, 2))
        textOut.WriteLine(" sPositiveDeflection, ft:" &
FormatNumber(sPositiveDeflection, 2))
        textOut.WriteLine(" Tire stiffnesses. Main lb/in, Side lb/in^1.5")
        textOut.WriteLine(" kmain: " & FormatNumber(c_kmain / 12, 1) & ",
kUpper: " & FormatNumber(c_kUpper / 12 ^ 1.5, 1) & ", kLower: " &
FormatNumber(c_kLower / 12 ^ 1.5, 1) & ", kswitch: " & FormatNumber(c_kswitch
/ 12 ^ 1.5, 1))
        textOut.WriteLine(" Passenger weight, lb: " & c_PassengerWeight & ",
kSeat: " & c_PassStiffness / 12)
        textOut.WriteLine(" Vehicle weight, lb: " & c_VehicleWeight & " LIM
Weight: " & c_LIMWeight)
        textOut.WriteLine(" MainFlareLength, ft " & c_MainFlareLength & ",
MainFlareOffSet, in " & c_mainFlareOffSet * 12)
        textOut.WriteLine(" SwxFlareLength, ft " & c_SwxFlareLength & ",
SwxFlareOffSet, in " & c_SwxFlareOffSet * 12)
        textOut.WriteLine(" Wind Speed, ft/s " & c_WindSpeed & ", Passenger
Offset,in " & c_PassengerOffset * 12)
        textOut.WriteLine(" Energy lost in side tires: " & c_EnergyLoss * 100
& "%")
        textOut.WriteLine(" Tire friction coefficient: " & c_Friction)
        textOut.WriteLine(" Centrifugal Force on if OnOff = 1, off if OnOff =
0, OnOff = " & OnOff)
        textOut.WriteLine(" Maximum Force between LIM bogie and Chassis: " &
FormatNumber(MaxChassisForce, 2))
        textOut.WriteLine()
        textOut.WriteLine(" Max Roll Angle, deg:" & FormatNumber(RollMax *
c_DegperRad, 3) & ", sRollMax, ft: " & FormatNumber(sRollMax, 1))
```

```
        textOut.WriteLine(" Max  Yaw Angle, deg:" & FormatNumber(YawMax *
c_DegperRad, 3) & ", sYawMax, ft: " & FormatNumber(sYawMax, 1))
        textOut.WriteLine(" Max yMC, in:" & FormatNumber(yMCmax * 12, 1) & ",
syMCMax, ft: " & FormatNumber(syMCMax, 1))
        textOut.WriteLine()
        textOut.WriteLine(" Deflections, in")
        textOut.WriteLine(" MaxDeflUFL     MaxDeflUBL     MaxDeflLFL
MaxDeflLBL ")
        textOut.WriteLine("    " & FormatNumber(DeflUFLmax * 12, 3) & "
" & FormatNumber(DeflUBLmax * 12, 3) & "           " &
FormatNumber(DeflLFLmax * 12, 3) & "           " & FormatNumber(DeflLBLmax *
12, 3))
        textOut.WriteLine(" MaxDeflUFR     MaxDeflUBR     MaxDeflLFR
MaxDeflLBR ")
        textOut.WriteLine("    " & FormatNumber(DeflUFRmax * 12, 3) & "
" & FormatNumber(DeflUBRmax * 12, 3) & "           " &
FormatNumber(DeflLFRmax * 12, 3) & "           " & FormatNumber(DeflLBRmax *
12, 3))
        textOut.WriteLine(" MaxDeflSFL     MaxDeflSBL ")
        textOut.WriteLine("    " & FormatNumber(DeflSFLmax * 12, 3) & "
" & FormatNumber(DeflSBLmax * 12, 3))
        textOut.WriteLine()
        textOut.WriteLine(" Maximum Forces,lb")
        textOut.WriteLine(" MaxForceUFL      MaxForceUBL      MaxForceLFL
MaxForceLBL ")
        textOut.WriteLine("    " & FormatNumber(ForceUFLmax, 1) & "
" & FormatNumber(ForceUBLmax, 1) & "           " & FormatNumber(ForceLFLmax,
1) & "           " & FormatNumber(ForceLBLmax, 1))
        textOut.WriteLine(" MaxForceUFR      MaxForceUBR      MaxForceLFR
MaxForceLBR ")
        textOut.WriteLine("    " & FormatNumber(ForceUFRmax, 1) & "
" & FormatNumber(ForceUBRmax, 1) & "           " & FormatNumber(ForceLFRmax,
1) & "           " & FormatNumber(ForceLBRmax, 1))
        textOut.WriteLine(" MaxForceSFL      MaxForceSBL ")
        textOut.WriteLine("    " & FormatNumber(ForceSFLmax, 1) & "
" & FormatNumber(ForceSBLmax, 1))
        textOut.WriteLine()
        textOut.WriteLine(" s at MaxForceUFR " & FormatNumber(sAtMaxUFR, 1) &
" s at MaxForceSBL " & FormatNumber(sAtMaxSBL, 1))
        textOut.WriteLine()
        textOut.WriteLine(" yMCAccelMax, g's   MaxPassAccel, g's
MaxSeatShift, in")
        textOut.WriteLine("    " & FormatNumber(yMCAccelMax / c_g, 3) & "
" & FormatNumber(yPassAccelMax / c_g, 3) & "            " &
FormatNumber(MaxSeatShift * 12, 2))
        textOut.WriteLine(" yCoverMax, in           s at yCoverMax, ft")
        textOut.WriteLine("    " & FormatNumber(yCoverMax * 12, 3) & "
" & FormatNumber(sAtCoverMax, 2))
        textOut.WriteLine(" UFR deflection at sStart, in: " &
FormatNumber(DeflUFRsStart * 12, 3) & ", s when UFR tire hits, ft " &
FormatNumber(sPositiveDeflection, 2))
        textOut.WriteLine()
        textOut.Close()
        objGraphics.Dispose()
    End Sub
    Sub CurvedGuideway(ByRef Psi As Double, ByRef x As Double, ByRef y As
Double)
        If s < 0 Then
            x = s
            y = 0
            Psi = 0
        ElseIf s < s1 Then
            Psi = 0.5 * c_Jn * s ^ 2 / c_Speed ^ 3
            x = s * (1 - Psi ^ 2 / 10)
            y = s * Psi / 3 * (1 - Psi ^ 2 / 14)
        Else
            Psi = Psi1 + (s - s1) / R
            x = xc + R * Math.Sin(Psi)
            y = yc - R * Math.Cos(Psi)
        End If
    End Sub

    Function SwxFlare(ByVal s As Double) As Double
        Dim y As Double = 0
        If s >= -c_SwxFlareLength And s < 0 Then
            y = c_SwxFlareOffSet * (s / c_SwxFlareLength) ^ 2
        ElseIf s >= 0 And s < sStartInnerSurface + c_MainFlareLength Then
            y = 0
        ElseIf s >= sStartInnerSurface + c_MainFlareLength And s < sEnd Then
            y = c_SwxFlareOffSet * ((s - sStartInnerSurface -
c_MainFlareLength) / c_SwxFlareLength) ^ 2
        Else
            y = 0
        End If
        SwxFlare = y
    End Function

    Function MainFlare(ByVal s As Double) As Double
        Dim y As Double
        If s >= sStartInnerSurface And s < sStartInnerSurface +
c_MainFlareLength Then
            y = c_mainFlareOffSet * ((sStartInnerSurface + c_MainFlareLength
- s) / c_MainFlareLength) ^ 2
        Else
            y = 0
        End If
        MainFlare = y
    End Function

    Function yTireLocal(ByVal Psi As Double, ByVal yMC As Double, ByVal Yaw
As Double, ByVal Roll As Double, ByVal xw As Double, ByVal yw As Double,
ByVal zw As Double) As Double
        yTireLocal = yMC * Math.Cos(Psi) + Math.Sin(Psi + Yaw) * (xw - c_Xcg)
+ Math.Cos(Psi + Yaw) * (Math.Cos(Roll) * yw - Math.Sin(Roll) * (zw - c_Zcg))
    End Function

    Function xTireLocal(ByVal Psi As Double, ByVal yMC As Double, ByVal Yaw
As Double, ByVal Roll As Double, ByVal xw As Double, ByVal yw As Double,
ByVal zw As Double) As Double
        xTireLocal = -yMC * Math.Sin(Psi) + Math.Cos(Psi + Yaw) * (xw -
c_Xcg) - Math.Sin(Psi + Yaw) * (Math.Cos(Roll) * yw - Math.Sin(Roll) * (zw -
c_Zcg))
```

```
    End Function

    Function yCL(ByVal x As Double) 'y at centerline
        Dim sg1, xg1, sg2, xg2, sAns, PsiAns, cosPsi As Double
        If x <= 0 Then
            yCL = 0
        ElseIf x < x1 Then
            sg1 = x
            xg1 = sg1 * (1 - (c_J2V3 * sg1 ^ 2) ^ 2 / 10)
            sg2 = sg1 + ds
            xg2 = sg2 * (1 - (c_J2V3 * sg2 ^ 2) ^ 2 / 10)
            sAns = sg2 + ds * (x - xg2) / (xg2 - xg1)
            PsiAns = c_J2V3 * sAns ^ 2
            yCL = (1 - PsiAns ^ 2 / 14) * sAns * PsiAns / 3
        Else
            cosPsi = Math.Sqrt(1 - ((x - xc) / R) ^ 2)
            yCL = yc - R * cosPsi
        End If
    End Function
    Function cosPsi(ByVal x As Double) As Double
        Dim sAns, xg1, xg2, sg1, sg2, sinPsi As Double
        If x <= 0 Then
            sinPsi = 0
        ElseIf x < x1 Then
            sg1 = x
            xg1 = sg1 * (1 - (c_J2V3 * sg1 ^ 2) ^ 2 / 10)
            sg2 = sg1 + ds
            xg2 = sg2 * (1 - (c_J2V3 * sg2 ^ 2) ^ 2 / 10)
            sAns = sg2 + ds * (x - xg2) / (xg2 - xg1)
            sinPsi = Math.Sin(c_J2V3 * sAns ^ 2)
        Else
            sinPsi = (x - xc) / R
        End If
        cosPsi = Math.Sqrt(1 - sinPsi ^ 2)
    End Function

    Function SideTireForce(ByVal k As Double, ByVal Defl As Double, ByVal
previousD As Double, ByVal ppreviousD As Double, ByRef MaxD As Double) As
Double
        Dim kr As Double
        If Defl > 0 Then
            If Defl >= previousD Then
                SideTireForce = k * Defl ^ 1.5
            Else
                If previousD >= ppreviousD Then
                    MaxD = previousD
                End If
                If MaxD > 0 Then
                    kr = k / MaxD ^ (c_Beta - 1.5)
                    SideTireForce = kr * Defl ^ c_Beta
                Else
                    SideTireForce = 0
                End If
            End If
        Else
            SideTireForce = 0
        End If
    End Function
    Function MainTireForce(ByVal k As Double, ByVal Defl As Double, ByVal
previousD As Double, ByVal ppreviousD As Double, ByRef MaxD As Double) As
Double
        Dim kr As Double
        If Defl > 0 Then
            If Defl >= previousD Then
                MainTireForce = k * Defl
            Else
                If previousD >= ppreviousD Then
                    MaxD = previousD
                End If
                If MaxD > 0 Then
                    kr = k / MaxD ^ (c_Gamma - 1)
                    MainTireForce = kr * Defl ^ c_Gamma
                Else
                    Stop
                    MainTireForce = 0
                End If
            End If
        Else
            MainTireForce = 0
        End If
    End Function
    Function Curvature() As Double
        If s < 0 Then
            Curvature = 0
        ElseIf s < s1 Then
            Curvature = c_Jn * s / c_Speed ^ 3
        Else
            Curvature = 1 / R
        End If
    End Function

    Private Sub Outline_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Outline.Click
        RunSurface()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RunDynamics.Click
        LateralMotion()
    End Sub

    Private Sub Quit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnQuit.Click
        Me.Close()
    End Sub
End Class
```