1A. 60% front load distribution $\Rightarrow \frac{b}{l} = 0.6, \; \frac{a}{l} = 0.4$

$\Rightarrow b = 0.6l, \; a = 0.4l$

$$\begin{bmatrix} \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} -\left(\frac{C_r + C_f}{mV}\right) & \left(\frac{C_r \cdot b - C_f \cdot a}{mV^2} - 1\right) \\ \left(\frac{C_r \cdot b - C_f \cdot a}{I_z}\right) & -\left(\frac{C_r \cdot b^2 + C_f \cdot a^2}{I_z \cdot V}\right) \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \end{bmatrix} + \begin{bmatrix} \left(\frac{C_f}{mV}\right) \\ \left(\frac{C_f \cdot a}{I_z}\right) \end{bmatrix} \delta$$

$\rightarrow \dot{X} = AX + BU$

$\gamma = \dot{\psi}$

$\rightarrow \psi_{t+1} = \psi_t + \dot{\psi}\, dt$

$\rightarrow \begin{cases} V_x = V \cdot \cos(\beta + \psi) \\ V_y = V \cdot \sin(\beta + \psi) \end{cases}$

$\rightarrow \begin{cases} P_{x_{t+1}} = P_{x_t} + V_x\, dt \\ P_{y_{t+1}} = P_{y_t} + V_y\, dt \end{cases}$

1B. $C_\alpha = a F_z + b F_z^2$

$\Rightarrow \begin{bmatrix} F_z & F_z^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = C_\alpha$

$a_y = V(\dot{\beta} + \dot{\psi})$

$F_{z_{*\#}} = S_* \pm \underbrace{\frac{m a_y h}{t}\left(\frac{K_{\phi_*}}{\Sigma K_{\phi *}}\right)}_{\text{load transfer}}$

* front / rear

\# left / right

static load

(+) outer (right)

(−) inner (left)

$S_* = \frac{mg}{2}$ (load distrib)

$C_{\alpha_{*\#}} = a F_{z_{*\#}} + b\left(F_{z_{*\#}}\right)^2$

$C_{\alpha_*} = \Sigma C_{\alpha_{*\#}}$ → e.g. $C_{\alpha f} = C_{\alpha f l} + C_{\alpha f r}$

$$\psi_{t+1} = \psi_t + \dot{\psi}\,\Delta t$$

$$x = \begin{bmatrix}\beta \\ \dot{\psi}\end{bmatrix} \Rightarrow x(2) = \dot{\psi} = r$$

$$V_x = v\cdot\cos(\beta+\psi)$$

$$V_y = v\cdot\sin(\beta+\psi)$$

$$\dot{\beta} = \dot{x}(2) \leftarrow \dot{x} = \begin{bmatrix}\dot{\beta} \\ \ddot{\psi}\end{bmatrix} = \begin{bmatrix}\dot{\beta} \\ \dot{r}\end{bmatrix}$$

(Other eq^n's / concepts are similar to previous problem)

2A. $a_y = v(\dot{\beta} + \dot{\psi})$

$$F_{z_{*\#}} = S_* \pm \frac{m a_y h}{t}\left(\frac{K_{\phi *}}{\Sigma K_{\phi *}}\right)$$

* front / rear
\# left / right

$$\alpha_f = \delta - \frac{\dot{\psi} a}{v} - \beta$$

$\uparrow$ U

$$\alpha_r = \frac{\dot{\psi} b}{v} - \beta$$

ISO → Adapted ISO

$$F_{y_{*\#}} = -\,\mathrm{nonlintire}(\alpha_*, F_{z*\#}, v)$$

* front / rear
\# left / right

$$F_{y_*} = \sum F_{y_{*\#}} \rightarrow \text{e.g. } F_{yf} = F_{yfl} + F_{yfr}$$

$$\left.\begin{aligned}\dot{\beta} &= \frac{F_{yf} + F_{yr}}{mv} - \dot{\psi} \\ \ddot{\psi} &= \frac{F_{yf}\cdot a - F_{yr}\cdot b}{I_z}\end{aligned}\right\} \dot{x} = \begin{bmatrix}\dot{\beta} \\ \ddot{\psi}\end{bmatrix} = \begin{bmatrix}\dot{\beta} \\ \dot{r}\end{bmatrix}$$

(Other eq^n's / concepts are similar to previous problems)

2B1. Maneuver 1: We have const. R and const V

but, we know that

$$\left.\begin{aligned}\delta &= \delta_{Ack} + \alpha_f - \alpha_r \\ \delta &= \delta_{Ack} + UG\end{aligned}\right\} \Rightarrow UG = \frac{(\alpha_f - \alpha_r)}{a_y}$$

Maneuver 2: We have const V but variable R

$$UG = \left.\frac{\partial \delta}{\partial a_y}\right|_{R=const} \Rightarrow UG = \underbrace{\left.\frac{\partial \delta}{\partial a_y}\right|_{Test}}_{\text{(in linear range)}} - \frac{\partial \delta_{Ack}}{\partial a_y} = \frac{l}{V^2}$$

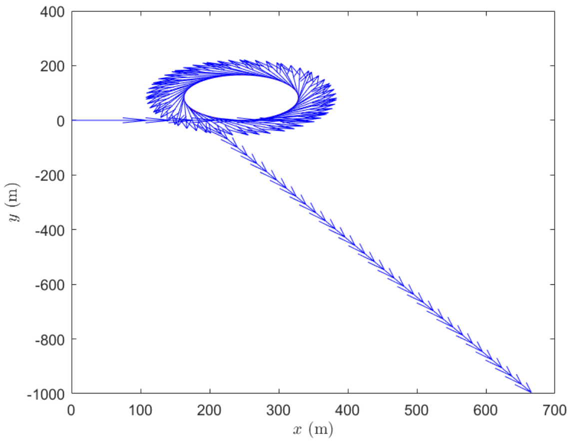
2B.2. → Similar approach to 2B.1, except account for load transfer & solve only for maneuver 2.

→ UG is larger (as expected) since load transfer to outside (right) will make vehicle more understeer.

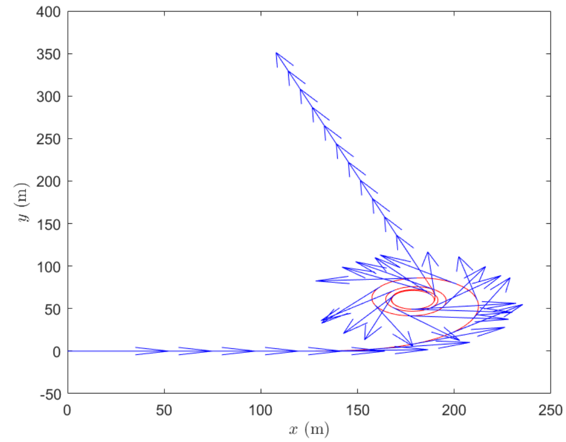
2B.3. Similar approach to 2B.2, but change "$k_{\phi f}$" from 0.6 to 0.3 & "$k_{\phi r}$" from 0.4 to 0.7

ANSWER 1A:

Maneuver 1



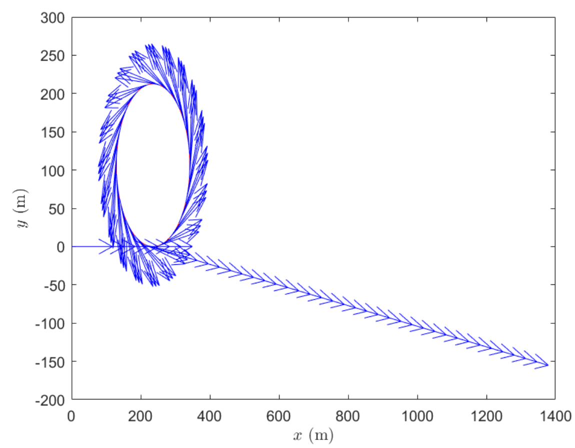
Maneuver 2

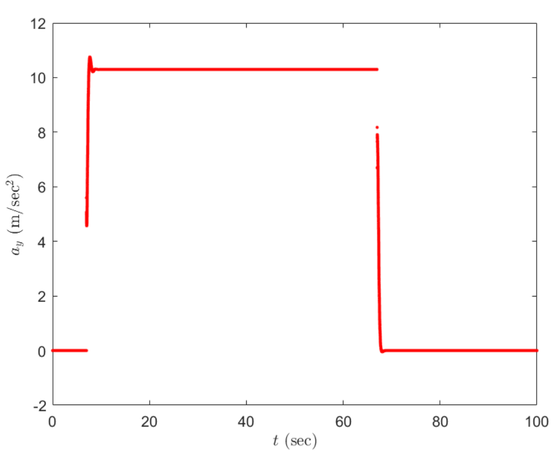


We can observe that the step input results in a straight-circle-straight maneuver and fishhook maneuver results in a straight-spiral-straight motion where the radius of spiral keeps decreasing.

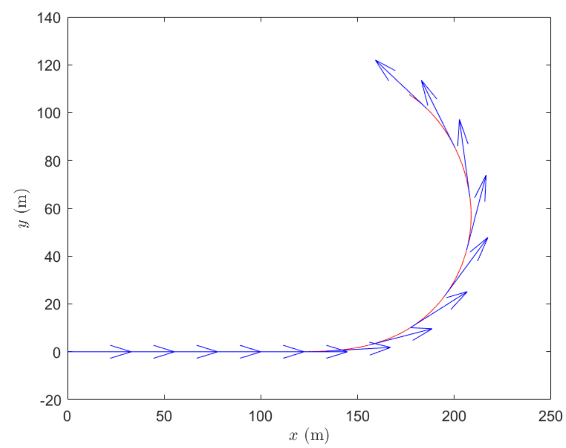
ANSWER 1B:

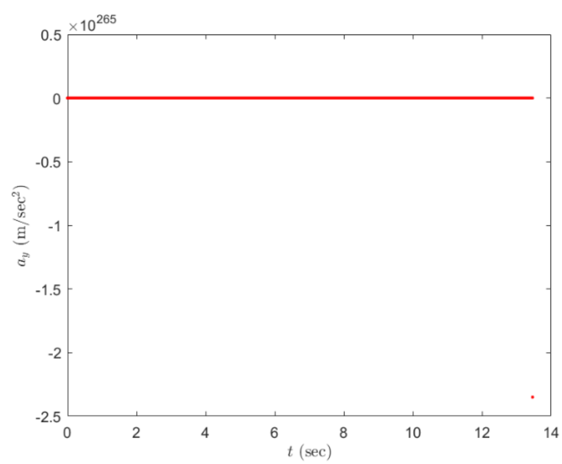
Maneuver 1





Maneuver 2

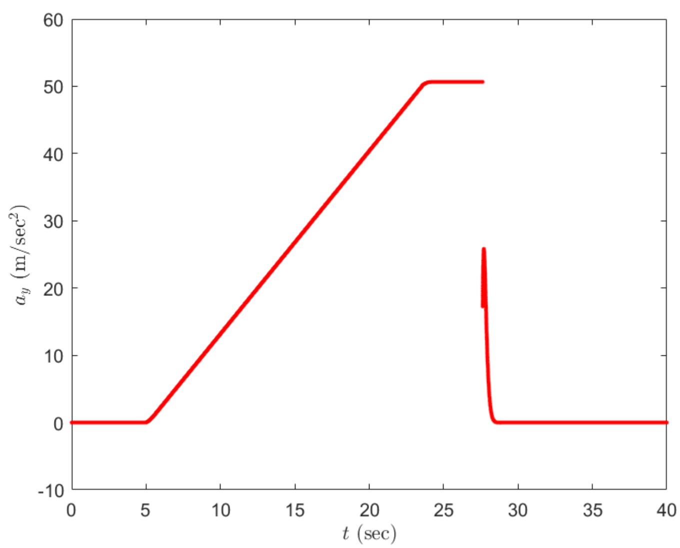




From the tabular data, the values of coefficients are, a = 25.6349 rad^{-1} and b = -0.0016178 $rad^{-1}N^{-1}$

We can observe that when we account for load transfer, the understeering behavior of the vehicle increases, resulting in corner with increased radius. However, we can see that since we are still using a linear tire model, this can lead to numerical instabilities in cases where the lateral acceleration is too high.

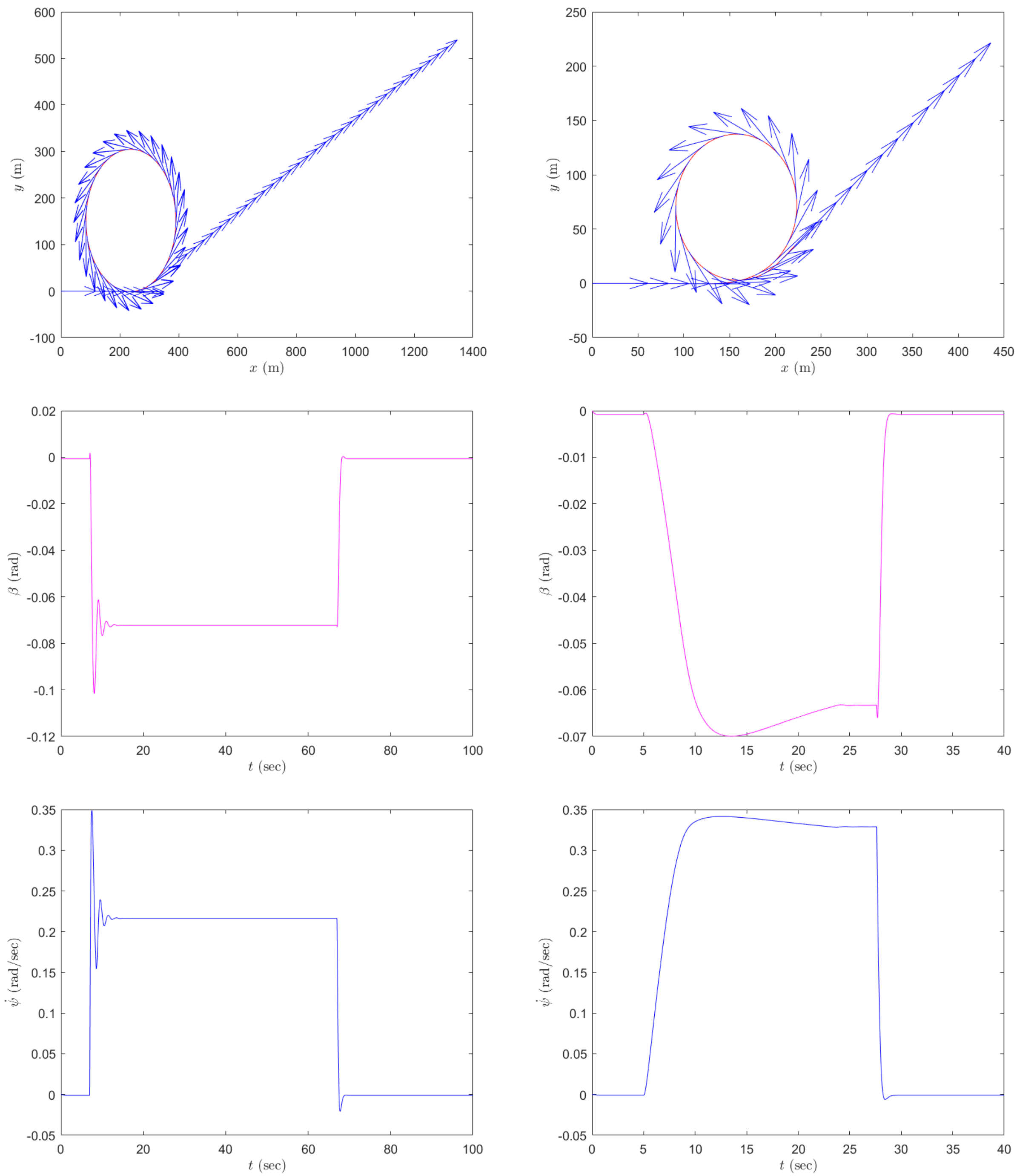
NOTE: If we look at lateral acceleration vs. time plots for both maneuvers, we can observe that for 1st maneuver, the maximum lateral acceleration achieved is slightly greater than 1g. However, for 2nd maneuver, the lateral acceleration becomes about -2.25e265 m/s^2 (which is due to numerical instability and not practical).



If we closely observe, for the 2nd maneuver (refer to plot above) without load transfer, we can see that the maneuver demands a peak lateral acceleration of more than 5g, which is high, but can still be handled by linear tire model (since we are not accounting for lateral acceleration anywhere in the linear steady state model). However, when we plug in the load transfer equations, as a_y keeps increasing there comes a point where the F_z for left (inner) tires becomes negative! This causes the cornering stiffness values to become negative, which in turn cause the state variables to become negative and ultimately makes a_y negative in next iteration. This continues until values blow up to NaN and as a result, the simulation stops without completion of maneuver.

ANSWER 2A:

Maneuver 1 **Maneuver 2**



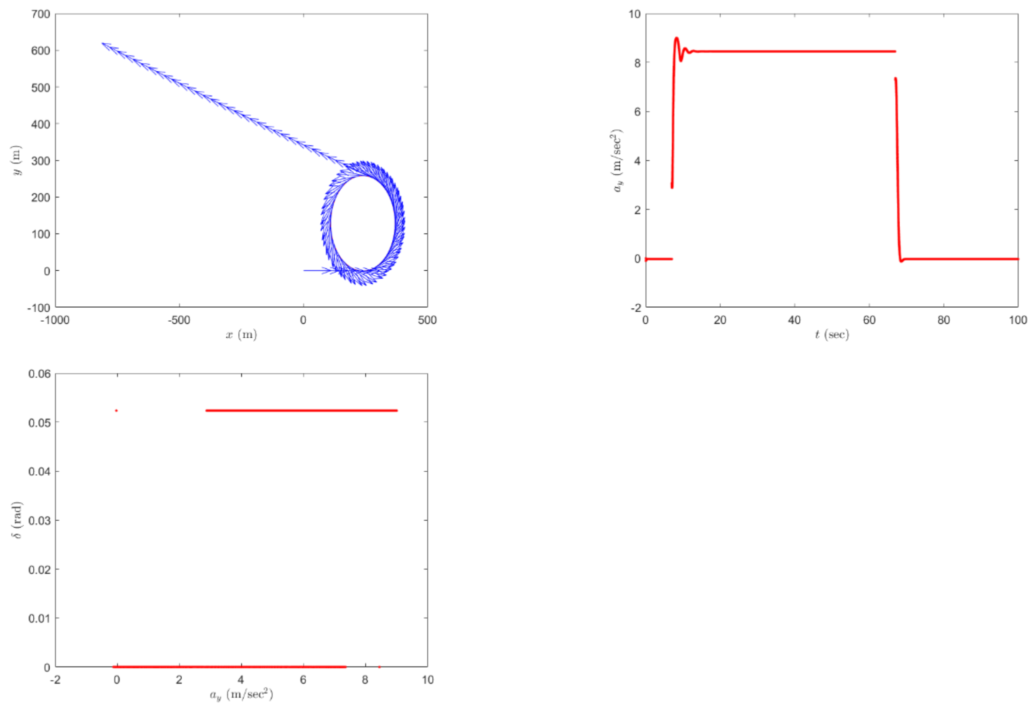
NOTE: The above plots were generated using the non-linear tire model while accounting load transfer assuming roll stiffness distribution of 60% front and 40% rear.

We can clearly observe that when we account for load transfer and use non-linear tire model, the understeering behavior of the vehicle increases further, resulting in corner with further increased radius.

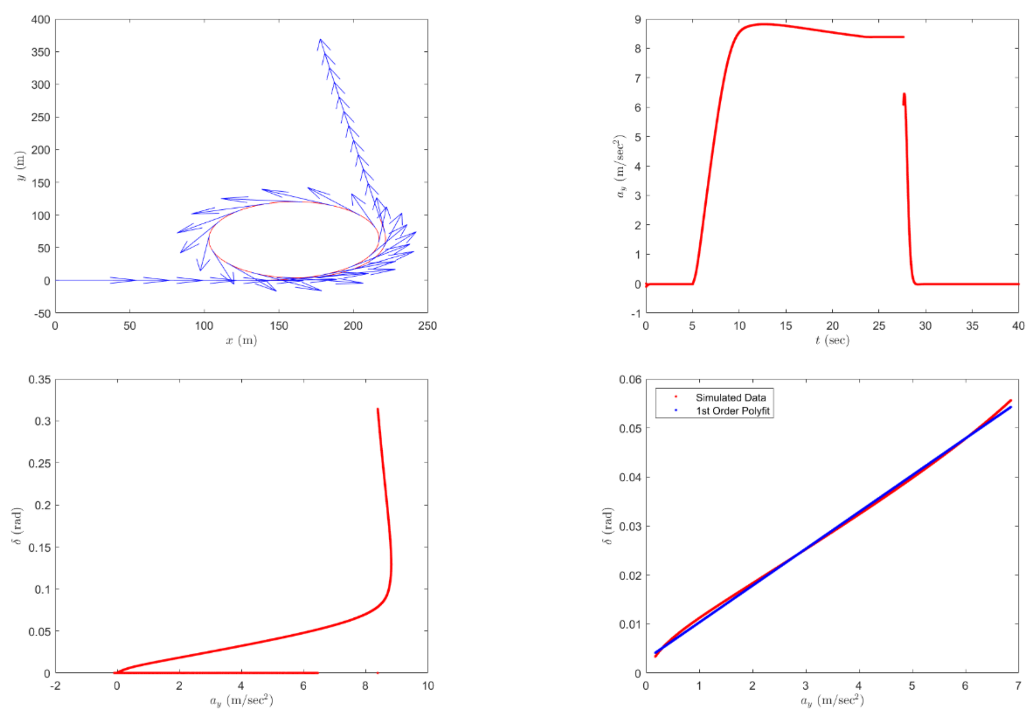
ANSWER 2B-1:

UG for Maneuver 1 with Non-Linear Tire Model is 0.003700 rad sec^2 m^{-1}
UG for Maneuver 2 with Non-Linear Tire Model is 0.002029 rad sec^2 m^{-1}

Maneuver 1



Maneuver 2

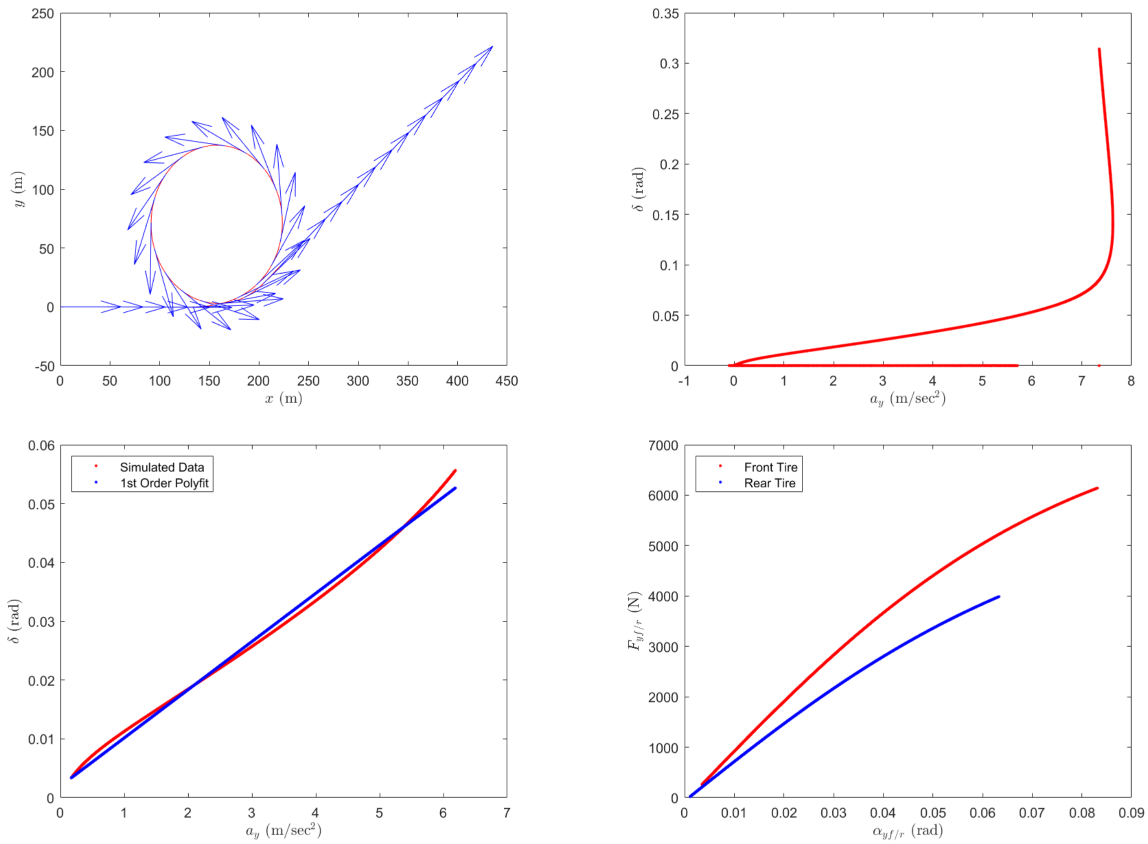


NOTE: The above plots were generated assuming roll stiffness distribution of 60% front and 40% rear. We can see that when we disregard load transfer but use non-linear tire model, the radius of the corner increases as compared to 1A.

ANSWER 2B-2:

UG for Maneuver 2 with Non-Linear Tire Model and Load Transfer (with roll stiffness distribution of 60% front and 40% rear) is 0.002719 rad sec^2 m^{-1}

Looking at variation of lateral force vs. slip angle plots for front & rear tires (with load transfer), and assuming same tire quality (i.e., same breakaway slip angle), we can observe that since the slope of the lateral force vs. slip angle plot in the linear region (i.e., cornering stiffness) of rear tire is less than that of the front tire, it will saturate (reach maximum lateral force) before the front tire, indicating that the vehicle will be limit oversteer.

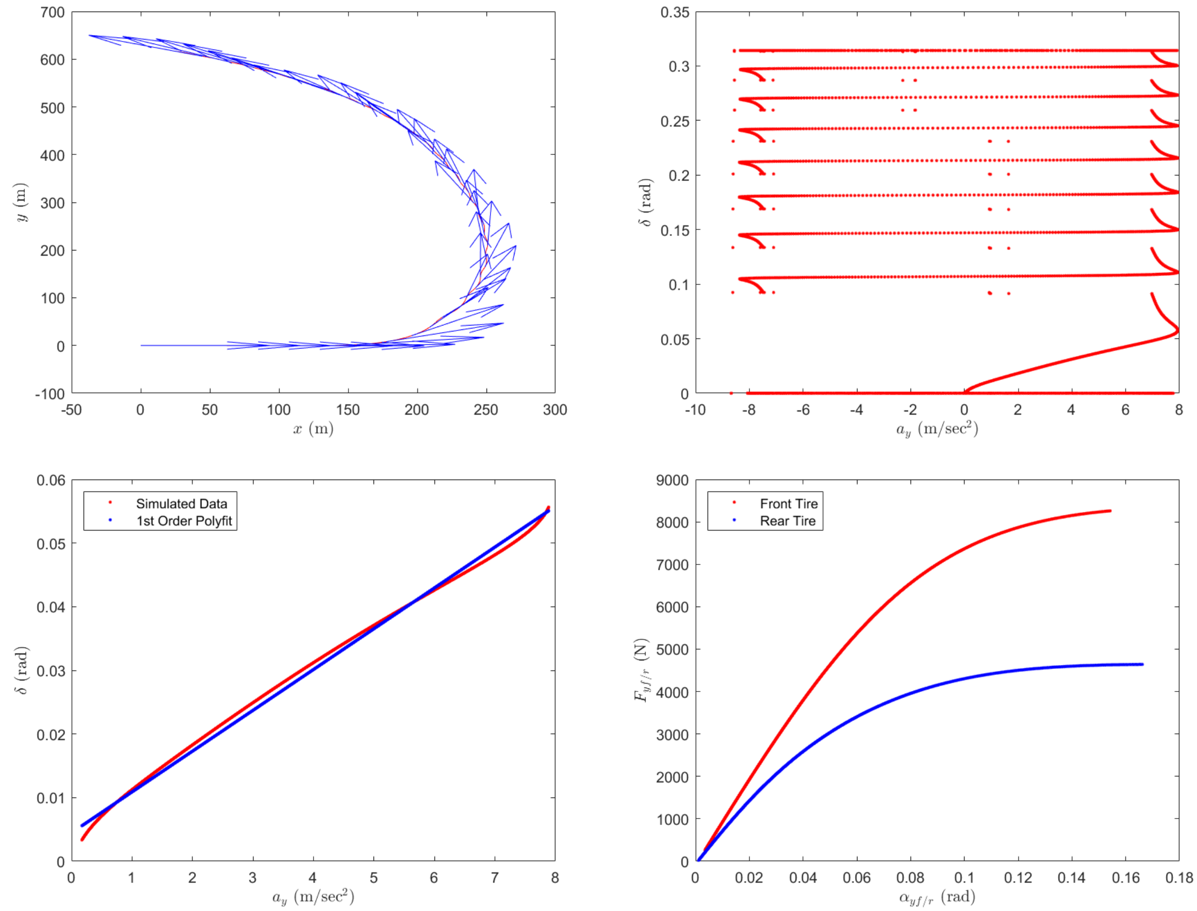


Since roll stiffness and load distribution are same, more (magnitude) load transfer will be resisted by more (magnitude) roll stiffness and hence we see expected behavior of increased understeering gradient (larger corner radius) as compared to 2B-1.

ANSWER 2B-3:

UG for Maneuver 2 with Non-Linear Tire Model and Load Transfer (with roll stiffness distribution of 30% front and 70% rear) is 0.000938 rad sec^2 m^{-1}

Looking at variation of lateral force vs. slip angle plots for front & rear tires (with load transfer), and assuming same tire quality (i.e., same breakaway slip angle), we can observe that since the slope of the lateral force vs. slip angle plot in the linear region (i.e., cornering stiffness) of rear tire is less than that of the front tire, it will saturate (reach maximum lateral force) before the front tire, indicating that the vehicle will be limit oversteer.



Since roll stiffness and load distributions are not same, rather the roll stiffness is less where there is more load and vice versa, more (magnitude) load transfer will be resisted by less (magnitude) roll stiffness and hence we see somewhat unstable vehicle motion.

Table of Contents

Clear Workspace

```
close all;
clear;
clc;
```

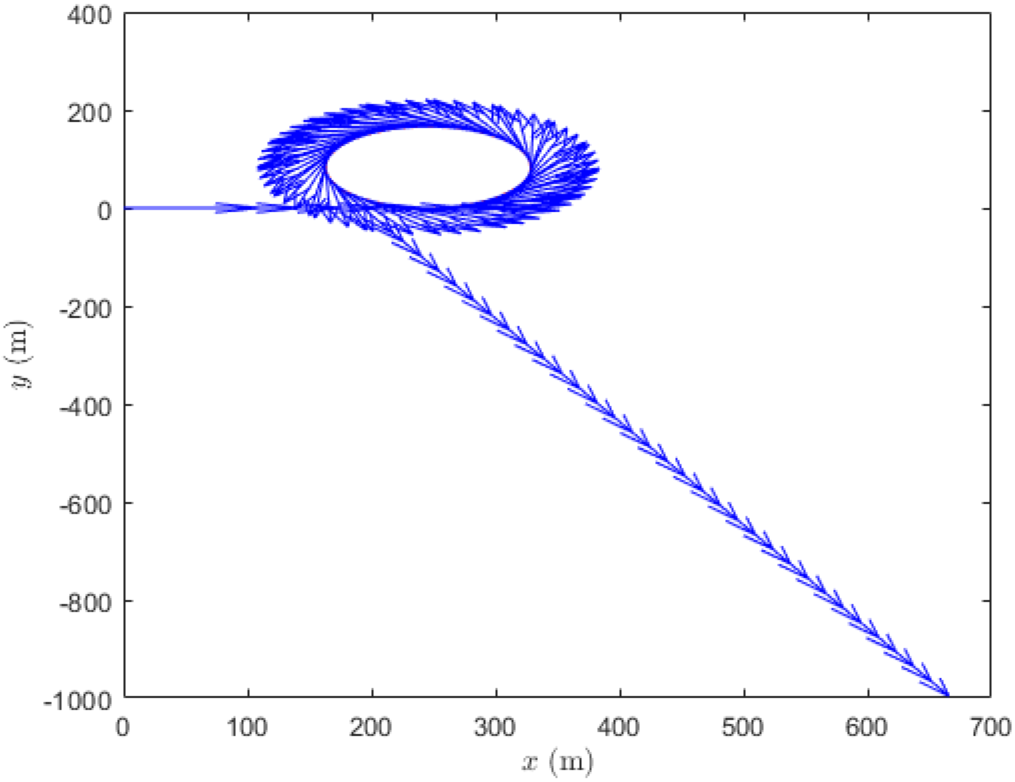
Given Data

```
m = 1637; % kg
Iz = 3326; % kg-m^2
l = 2.736; % m
t = 1.7; % m
a = 0.4*l; % 60% front load distribution
b = 0.6*l; % 60% front load distribution
Cf = 1500*(180/pi); % N/rad
Cr = 1146*(180/pi); % N/rad
G = 15;
h = 2.4/3.281; % m
```

Maneuver 1

```
v = 74/2.237; % m/s

% System matrix A
A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
     (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];

% Control matrix B
B = [Cf/(m*v); (Cf*a)/Iz];

% Simulation parameters
dt = 0.001; % s
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
V = [];
P = [0;0];
j = 1;

% Simulate from t=0 to t=7
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:7
    Xdot(:,j) = A*X(:,j)+B*U;
```

```
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    j = j+1;
end

% Simulate from t=7 to t=67
U = (45*(pi/180))/G; % Control input (@ 45 deg steering wheel angle)
for i = 7+dt:dt:67
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    j = j+1;
end

% Simulate from t=67 to t=100
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 67+dt:dt:100
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    j = j+1;
end

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(1)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')
```

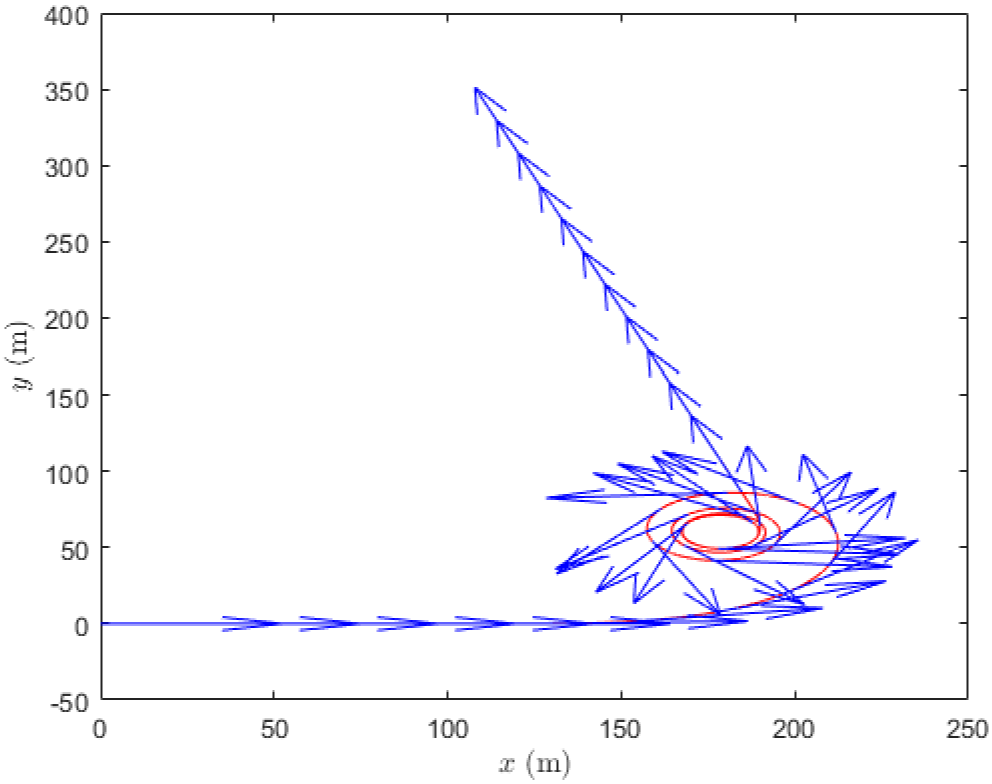


Maneuver 2

```
v = 50/2.237; % m/s

% System matrix A
A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
     (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];

% Control matrix B
B = [Cf/(m*v); (Cf*a)/Iz];

% Simulation parameters
dt = 0.001; % s
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
V = [];
P = [0;0];
j = 1;

% Simulate from t=0 to t=5
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:5
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
```

```
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    j = j+1;
end

% Simulate from t=5 to t=23.621
U = 0:((14.5*(pi/180))/G)*dt:(270*(pi/180))/G; % Control input (@ 0-270 deg
 steering wheel angle)
for i = 5+dt:dt:23.621
    Xdot(:,j) = A*X(:,j)+B*U(j-5000);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    j = j+1;
end

% Simulate from t=23.621 to t=27.621
U = (270*(pi/180))/G; % Control input (@ 270 deg steering wheel angle)
for i = 23.621+dt:dt:27.621
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    j = j+1;
end

% Simulate from t=27.621 to t=40
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 27.621+dt:dt:40
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    j = j+1;
end

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(2)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
```

```
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')
```



Published with MATLAB® R2022a

Table of Contents

Clear Workspace

```
close all;
clear;
clc;
```

Given Data

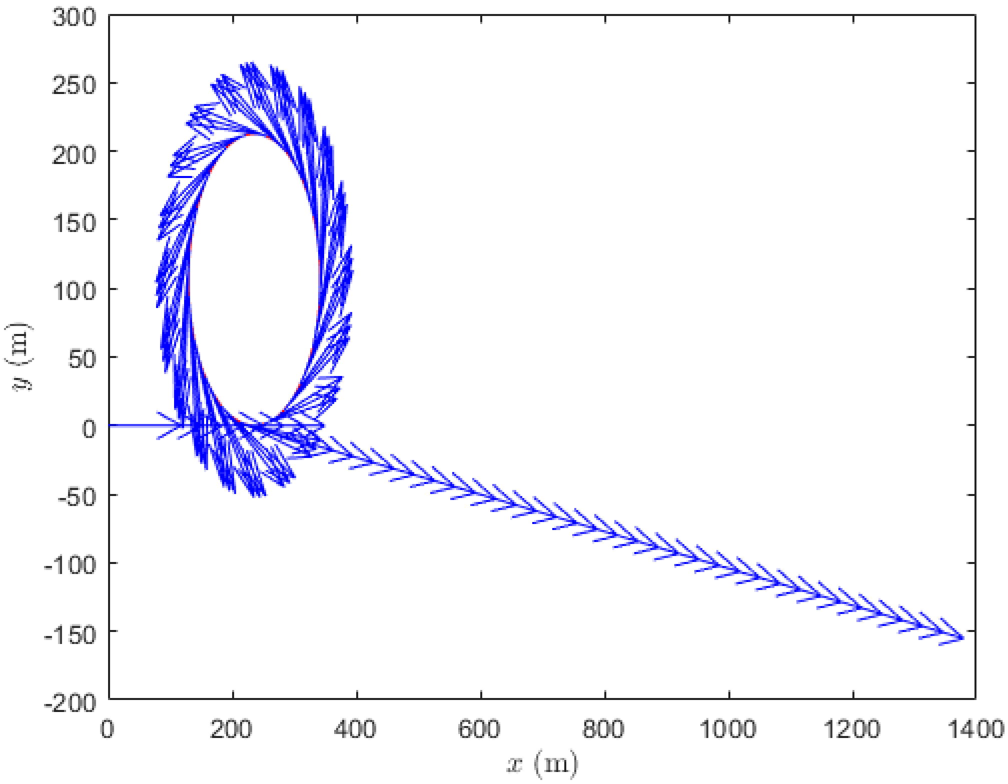
```
m = 1637; % kg
g = 9.81; % m/s^2
Sf = (m*g*0.6)/2; % 60% front (static) load distribution
Sr = (m*g*0.4)/2; % 60% front (static) load distribution
Iz = 3326; % kg-m^2
l = 2.736; % m
t = 1.7; % m
a = 0.4*l; % 60% front load distribution
b = 0.6*l; % 60% front load distribution
G = 15;
h = 2.4/3.281; % m

C_array = [848 972 1088 1195 1292 1381 1461 1531 1593 1645 1689 1723
 1748].*(180/pi); % N/rad
FZ_array = [2200 2600 3000 3400 3800 4200 4600 5000 5400 5800 6200 6600
 7000]; % N
FZ_matrix = [FZ_array' (FZ_array.^2)'];
C_matrix = C_array';
ab = FZ_matrix\C_matrix;
```

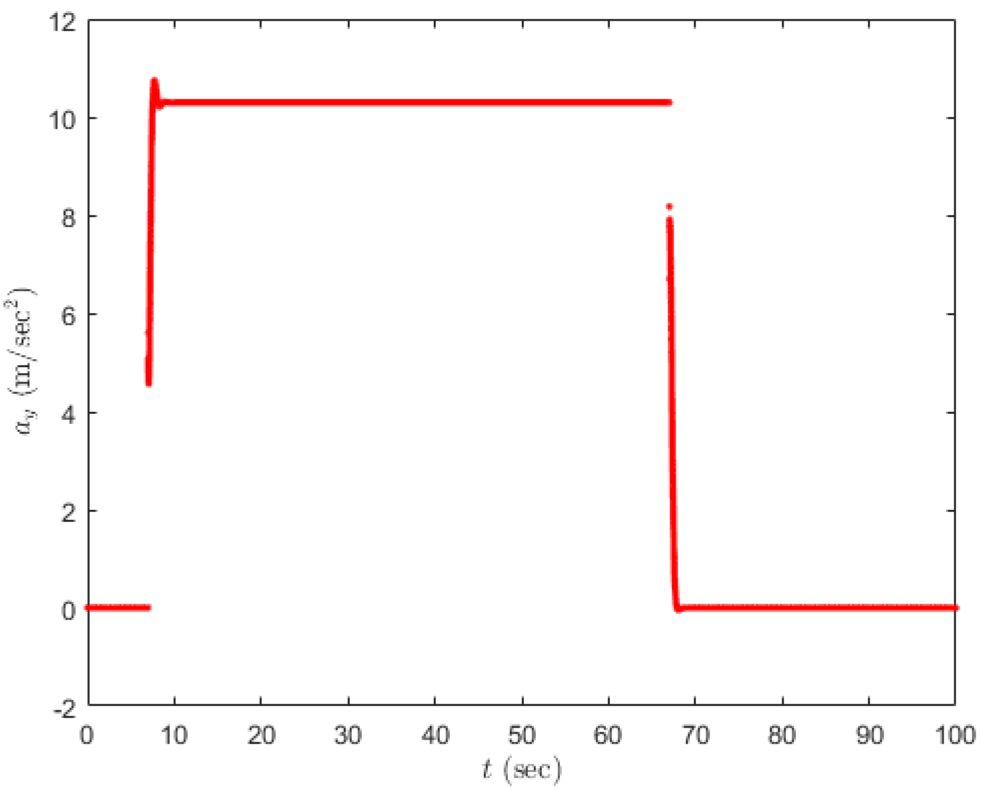
Maneuver 1

```
% Simulation parameters
v = 74/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
ay = [];
j = 1;
```

```matlab
% Simulate from t=0 to t=7
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:7
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    Cfl = ab(1)*FZfl + ab(2)*FZfl^2;
    Cfr = ab(1)*FZfr + ab(2)*FZfr^2;
    Cf = Cfl+Cfr;
    Crl = ab(1)*FZrl + ab(2)*FZrl^2;
    Crr = ab(1)*FZrr + ab(2)*FZrr^2;
    Cr = Crl+Crr;
    A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
         (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];
    B = [Cf/(m*v); (Cf*a)/Iz];
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=7 to t=67
U = (45*(pi/180))/G; % Control input (@ 45 deg steering wheel angle)
for i = 7+dt:dt:67
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    Cfl = ab(1)*FZfl + ab(2)*FZfl^2;
    Cfr = ab(1)*FZfr + ab(2)*FZfr^2;
    Cf = Cfl+Cfr;
    Crl = ab(1)*FZrl + ab(2)*FZrl^2;
    Crr = ab(1)*FZrr + ab(2)*FZrr^2;
    Cr = Crl+Crr;
    A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
         (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];
    B = [Cf/(m*v); (Cf*a)/Iz];
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=67 to t=100
U = 0; % Control input (@ 0 deg steering wheel angle)
```

```
for i = 67+dt:dt:100
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    Cfl = ab(1)*FZfl + ab(2)*FZfl^2;
    Cfr = ab(1)*FZfr + ab(2)*FZfr^2;
    Cf = Cfl+Cfr;
    Crl = ab(1)*FZrl + ab(2)*FZrl^2;
    Crr = ab(1)*FZrr + ab(2)*FZrr^2;
    Cr = Crl+Crr;
    A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
         (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];
    B = [Cf/(m*v); (Cf*a)/Iz];
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(1)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')

% Plot ay vs t
figure(2)
plot((0:dt:100),[0 ay],'.','color','red')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
```


y (m)
x (m)
300
250
200
150
100
50
0
-50
-100
-150
-200
0
200
400
600
800
1000
1200
1400

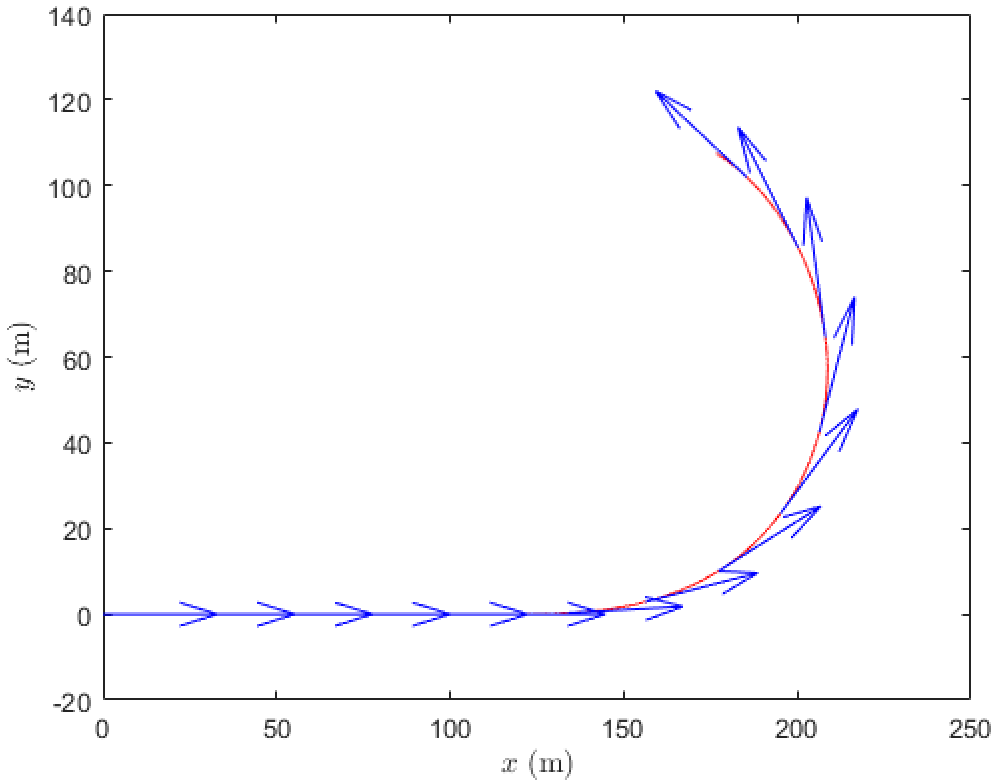

a_y (m/sec^2)
t (sec)
12
10
8
6
4
2
0
-2
0
10
20
30
40
50
60
70
80
90
100

Maneuver 2

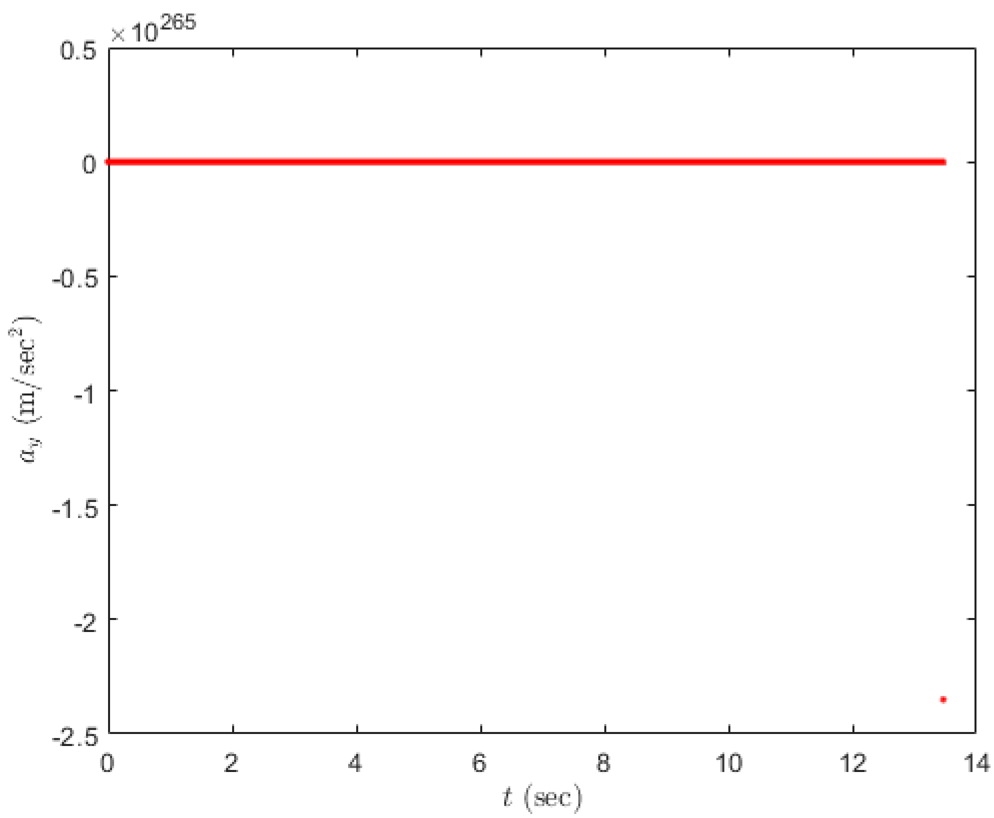
```
% Simulation parameters
v = 50/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
ay = [];
j = 1;

% Simulate from t=0 to t=5
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:5
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    Cfl = ab(1)*FZfl + ab(2)*FZfl^2;
    Cfr = ab(1)*FZfr + ab(2)*FZfr^2;
    Cf = Cfl+Cfr;
    Crl = ab(1)*FZrl + ab(2)*FZrl^2;
    Crr = ab(1)*FZrr + ab(2)*FZrr^2;
    Cr = Crl+Crr;
    A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
         (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];
    B = [Cf/(m*v); (Cf*a)/Iz];
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=5 to t=23.621
U = 0:((14.5*(pi/180))/G)*dt:(270*(pi/180))/G; % Control input (@ 0-270 deg
 steering wheel angle)
for i = 5+dt:dt:23.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    Cfl = ab(1)*FZfl + ab(2)*FZfl^2;
    Cfr = ab(1)*FZfr + ab(2)*FZfr^2;
    Cf = Cfl+Cfr;
```

```
    Crl = ab(1)*FZrl + ab(2)*FZrl^2;
    Crr = ab(1)*FZrr + ab(2)*FZrr^2;
    Cr = Crl+Crr;
    A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
         (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];
    B = [Cf/(m*v); (Cf*a)/Iz];
    Xdot(:,j) = A*X(:,j)+B*U(j-5000);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=23.621 to t=27.621
U = (270*(pi/180))/G; % Control input (@ 270 deg steering wheel angle)
for i = 23.621+dt:dt:27.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    Cfl = ab(1)*FZfl + ab(2)*FZfl^2;
    Cfr = ab(1)*FZfr + ab(2)*FZfr^2;
    Cf = Cfl+Cfr;
    Crl = ab(1)*FZrl + ab(2)*FZrl^2;
    Crr = ab(1)*FZrr + ab(2)*FZrr^2;
    Cr = Crl+Crr;
    A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
         (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];
    B = [Cf/(m*v); (Cf*a)/Iz];
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=27.621 to t=40
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 27.621+dt:dt:40
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    Cfl = ab(1)*FZfl + ab(2)*FZfl^2;
    Cfr = ab(1)*FZfr + ab(2)*FZfr^2;
    Cf = Cfl+Cfr;
    Crl = ab(1)*FZrl + ab(2)*FZrl^2;
    Crr = ab(1)*FZrr + ab(2)*FZrr^2;
```

```
    Cr = Crl+Crr;
    A = [-((Cf+Cr)/(m*v)) ((Cr*b-Cf*a)/(m*v^2)-1);
         (Cr*b-Cf*a)/Iz   -((Cr*b^2+Cf*a^2)/(Iz*v))];
    B = [Cf/(m*v); (Cf*a)/Iz];
    Xdot(:,j) = A*X(:,j)+B*U;
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(3)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')

% Plot ay vs t
figure(4)
plot((0:dt:40),[0 ay],'.','color','red')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
```



y (m)
x (m)



$\times 10^{265}$
a_y (m/sec^2)
t (sec)

Published with MATLAB® R2022a

Table of Contents

Clear Workspace

```
close all;
clear;
clc;
```

Given Data

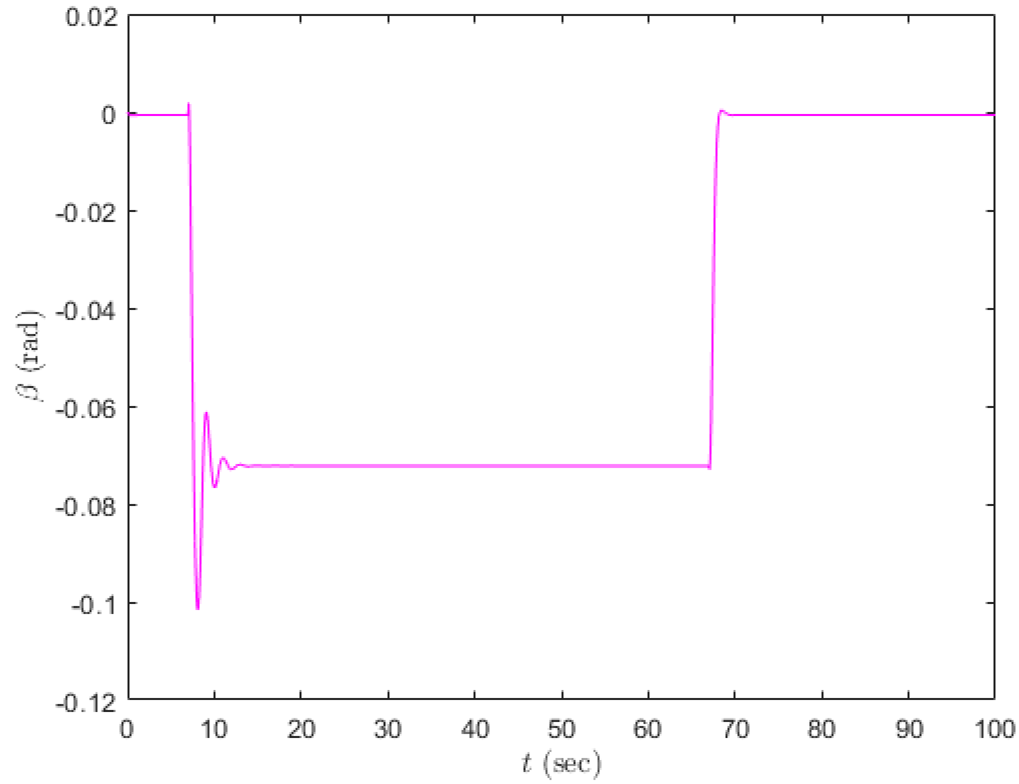
```
m = 1637; % kg
g = 9.81; % m/s^2
Iz = 3326; % kg-m^2
l = 2.736; % m
t = 1.7; % m
a = 0.4*l; % 60% front load distribution
b = 0.6*l; % 60% front load distribution
G = 15;
h = 2.4/3.281; % m
```

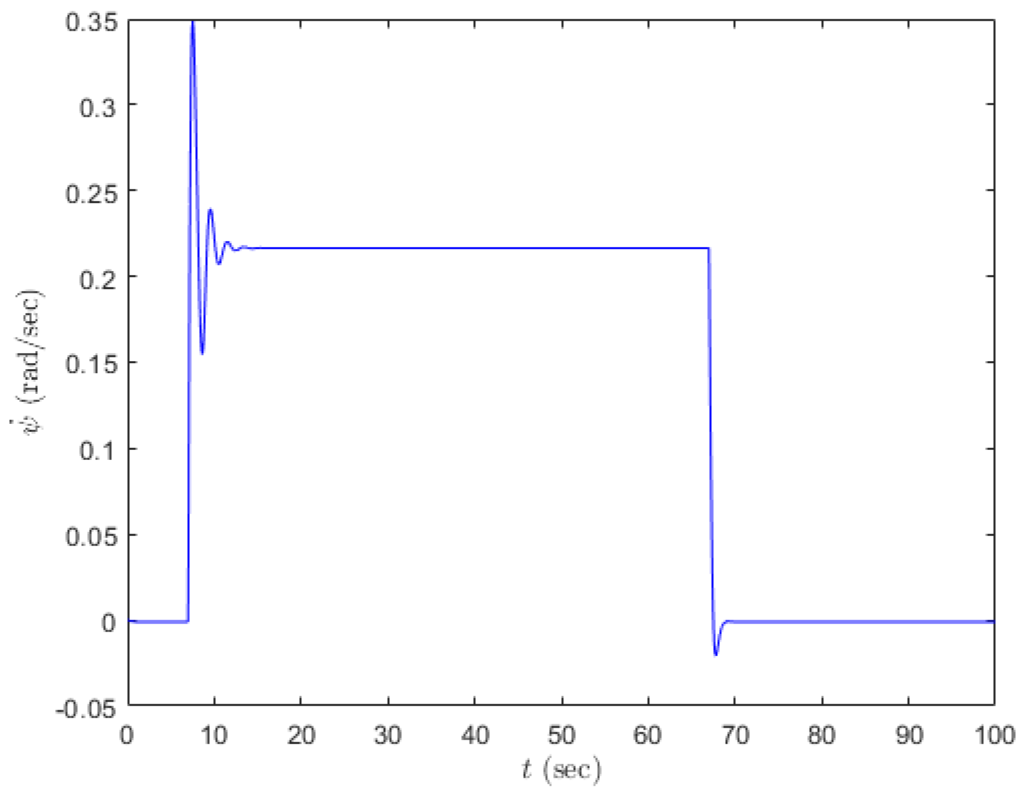
Maneuver 1

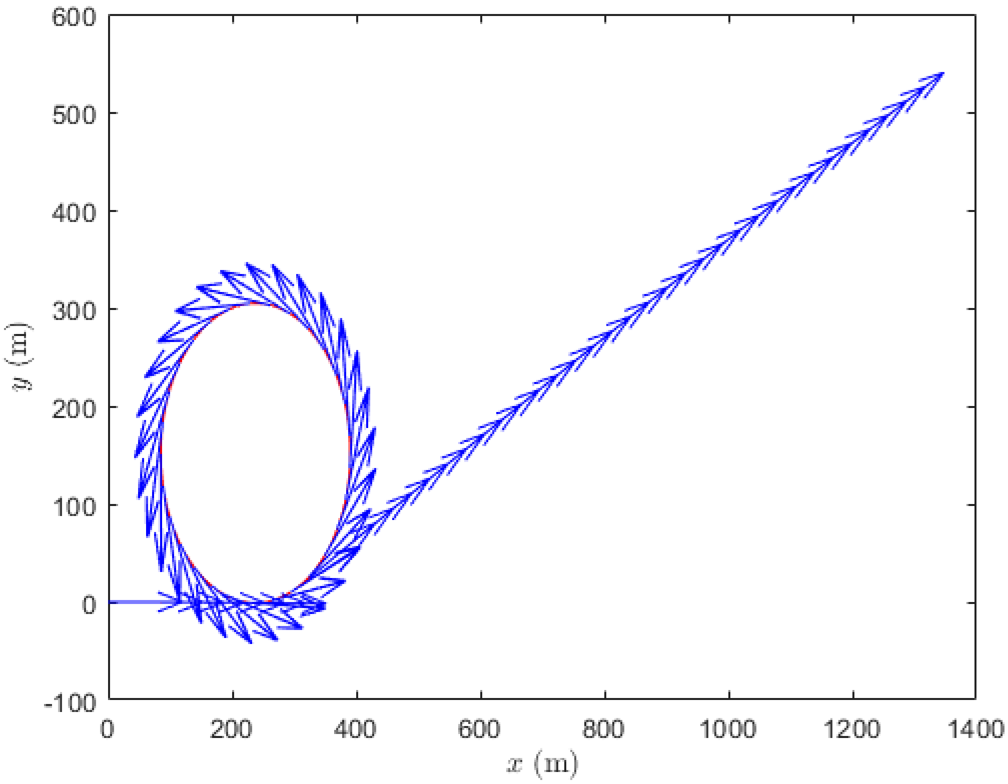
```
% Simulation parameters
v = 74/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
Sf = (m*g*0.6)/2; % 60% front (static) load distribution
Sr = (m*g*0.4)/2; % 60% front (static) load distribution
j = 1;

% Simulate from t=0 to t=7
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:7
    ay = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay*h)/t)*0.6;
    FZfr = Sf + ((m*ay*h)/t)*0.6;
    FZrl = Sr - ((m*ay*h)/t)*0.4;
    FZrr = Sr + ((m*ay*h)/t)*0.4;
    af = U - (X(2,j)*a)/v - X(1,j);
```

```
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=7 to t=67
U = (45*(pi/180))/G; % Control input (@ 45 deg steering wheel angle)
for i = 7+dt:dt:67
    ay = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay*h)/t)*0.6;
    FZfr = Sf + ((m*ay*h)/t)*0.6;
    FZrl = Sr - ((m*ay*h)/t)*0.4;
    FZrr = Sr + ((m*ay*h)/t)*0.4;
    af = U - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=67 to t=100
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 67+dt:dt:100
    ay = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay*h)/t)*0.6;
    FZfr = Sf + ((m*ay*h)/t)*0.6;
    FZrl = Sr - ((m*ay*h)/t)*0.4;
    FZrr = Sr + ((m*ay*h)/t)*0.4;
    af = U - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
```

```
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Plot states
figure(1)
plot((0:dt:100),X(1,:),'color','magenta')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${\beta}$ (rad)','interpreter','latex')
figure(2)
plot((0:dt:100),X(2,:),'color','blue')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${\dot\psi}$ (rad/sec)','interpreter','latex')

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(3)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')
```


0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
β (rad)
0
10
20
30
40
50
60
70
80
90
100
t (sec)


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
$\dot{\psi}$ (rad/sec)
0
10
20
30
40
50
60
70
80
90
100
t (sec)



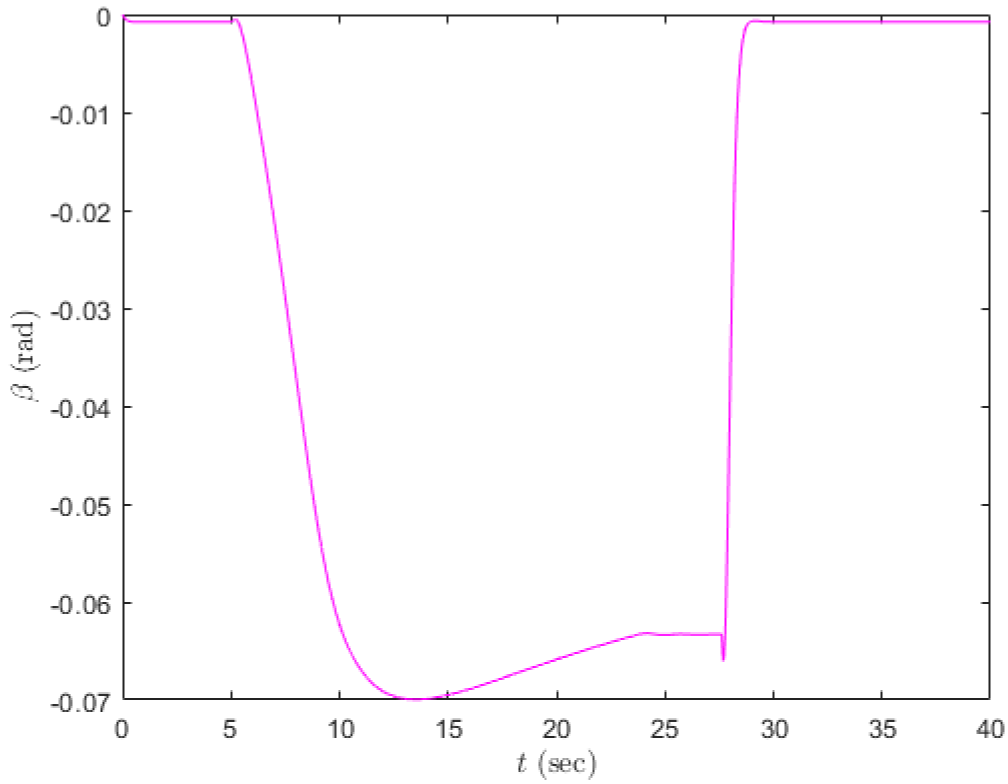
Maneuver 2

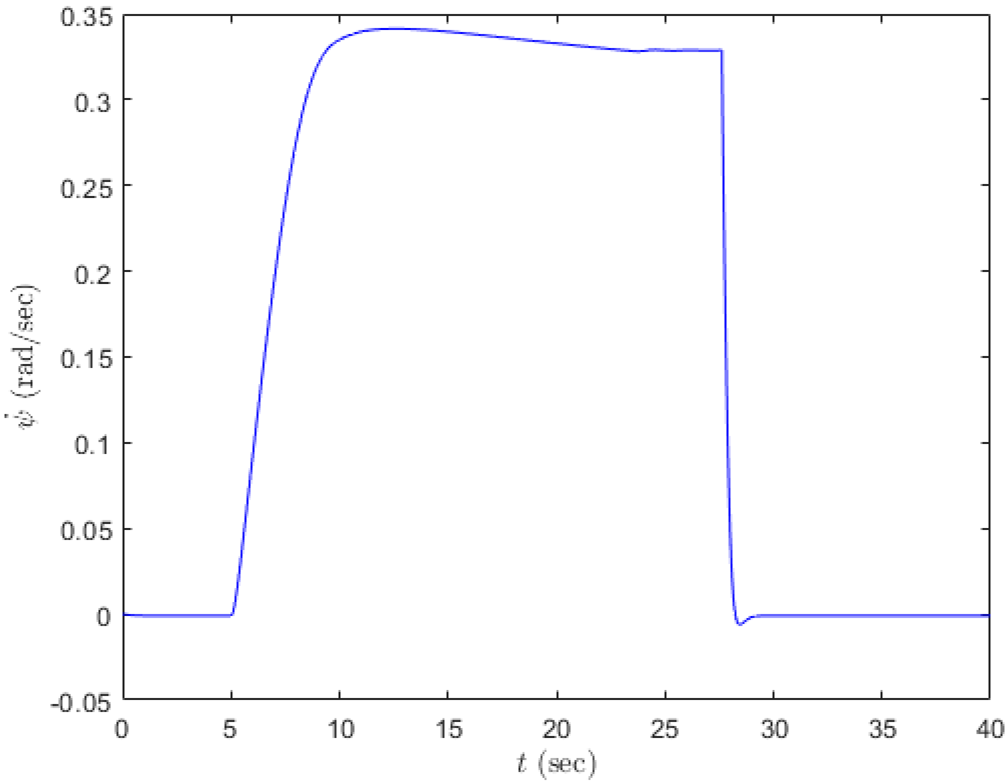
```
% Simulation parameters
v = 50/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
Sf = (m*g*0.6)/2; % 60% front (static) load distribution
Sr = (m*g*0.4)/2; % 60% front (static) load distribution
j = 1;

% Simulate from t=0 to t=5
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:5
    ay = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay*h)/t)*0.6;
    FZfr = Sf + ((m*ay*h)/t)*0.6;
    FZrl = Sr - ((m*ay*h)/t)*0.4;
    FZrr = Sr + ((m*ay*h)/t)*0.4;
    af = U - (X(2,j)*a)/v - X(1,j);
```

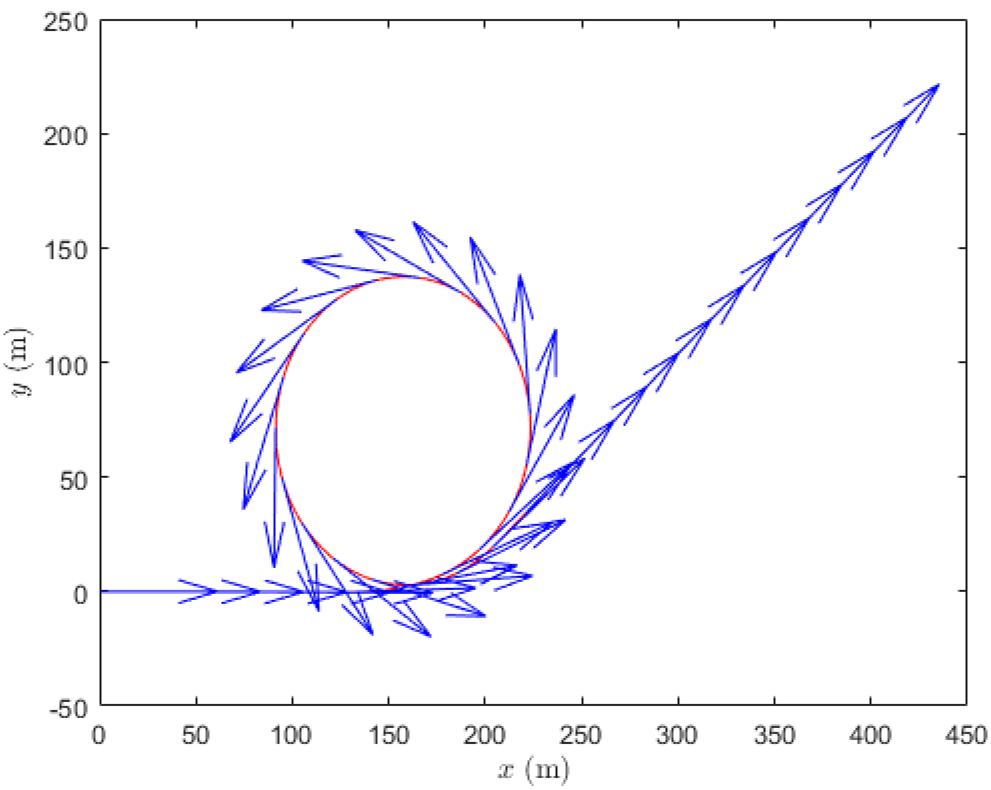
```
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=5 to t=23.621
U = 0:((14.5*(pi/180))/G)*dt:(270*(pi/180))/G; % Control input (@ 0-270 deg
 steering wheel angle)
for i = 5+dt:dt:23.621
    ay = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay*h)/t)*0.6;
    FZfr = Sf + ((m*ay*h)/t)*0.6;
    FZrl = Sr - ((m*ay*h)/t)*0.4;
    FZrr = Sr + ((m*ay*h)/t)*0.4;
    af = U(j-5000) - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=23.621 to t=27.621
U = (270*(pi/180))/G; % Control input (@ 270 deg steering wheel angle)
for i = 23.621+dt:dt:27.621
    ay = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay*h)/t)*0.6;
    FZfr = Sf + ((m*ay*h)/t)*0.6;
    FZrl = Sr - ((m*ay*h)/t)*0.4;
    FZrr = Sr + ((m*ay*h)/t)*0.4;
    af = U - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
```

```
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=27.621 to t=40
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 27.621+dt:dt:40
    ay = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay*h)/t)*0.6;
    FZfr = Sf + ((m*ay*h)/t)*0.6;
    FZrl = Sr - ((m*ay*h)/t)*0.4;
    FZrr = Sr + ((m*ay*h)/t)*0.4;
    af = U - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Plot states
figure(4)
plot((0:dt:40),X(1,:),'color','magenta')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${\beta}$ (rad)','interpreter','latex')
figure(5)
plot((0:dt:40),X(2,:),'color','blue')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${\dot\psi}$ (rad/sec)','interpreter','latex')

% Plot trajectory
PosX = P(1,:);
```

```
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(6)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')
```




0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
$\dot{\psi}$ (rad/sec)
0
5
10
15
20
25
30
35
40
t (sec)


250
200
150
100
50
0
-50
y (m)
0
50
100
150
200
250
300
350
400
450
x (m)

Published with MATLAB® R2022a

Table of Contents

Clear Workspace

```
close all;
clear;
clc;
```

Given Data

```
m = 1637; % kg
g = 9.81; % m/s^2
Iz = 3326; % kg-m^2
l = 2.736; % m
t = 1.7; % m
a = 0.4*l; % 60% front load distribution
b = 0.6*l; % 60% front load distribution
G = 15;
h = 2.4/3.281; % m
```

Maneuver 1

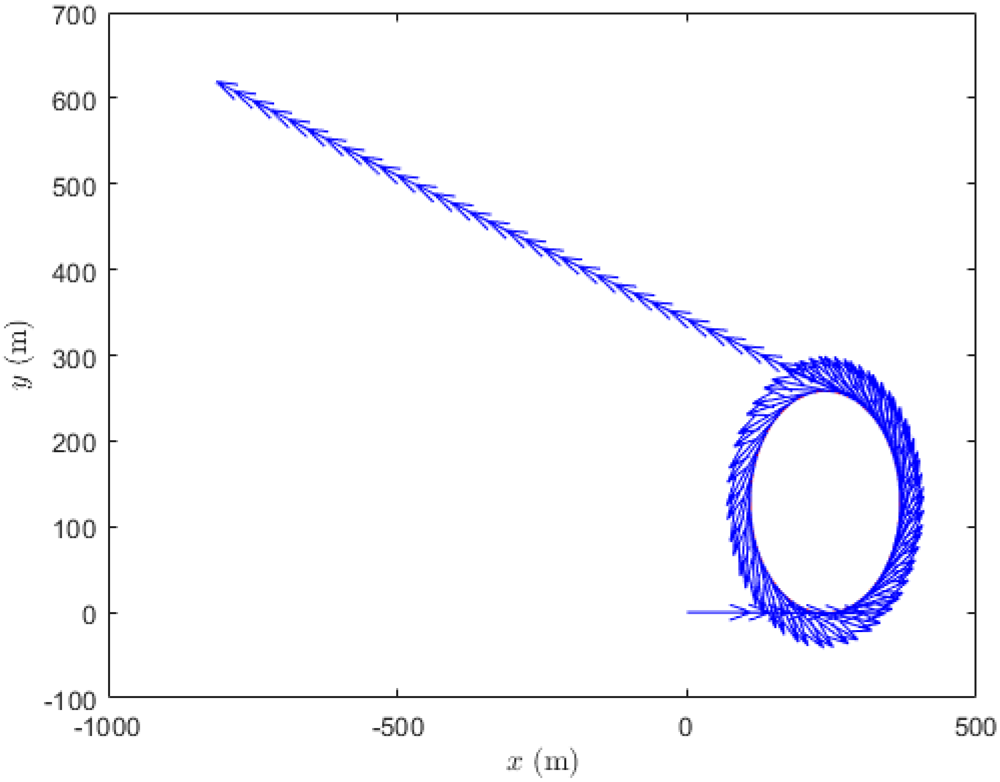
```
% Simulation parameters
v = 74/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
Sf = (m*g*0.6)/2; % 60% front (static) load distribution
Sr = (m*g*0.4)/2; % 60% front (static) load distribution
ay = [];
j = 1;

% Simulate from t=0 to t=7
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:7
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf;
    FZfr = Sf;
```

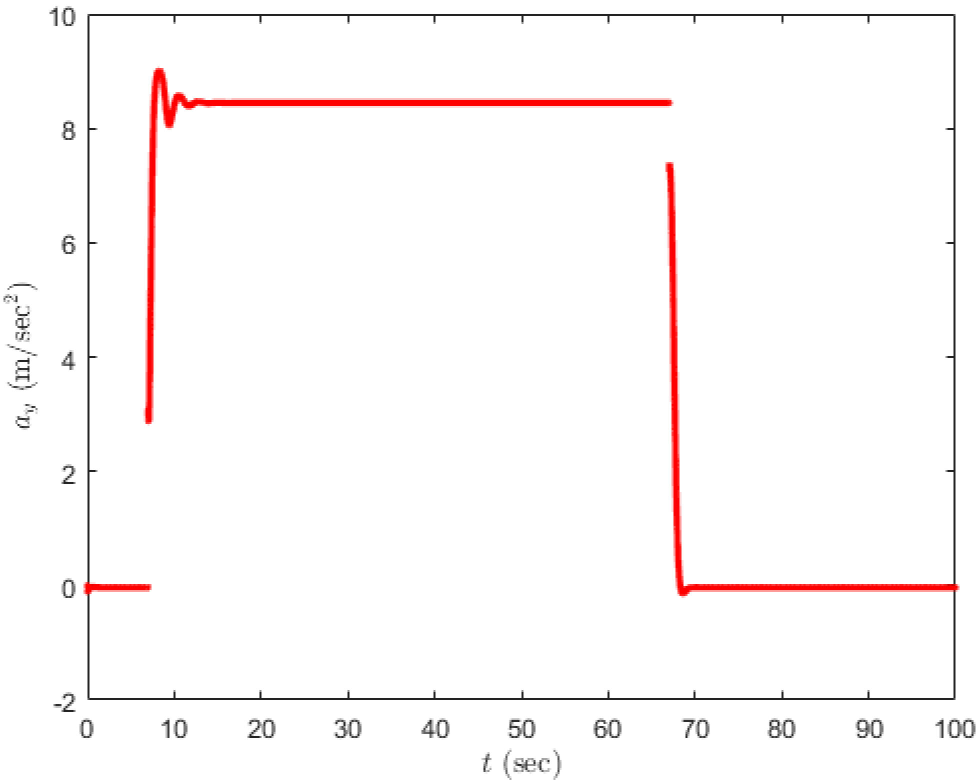
```
    FZrl = Sr;
    FZrr = Sr;
    af(j) = U - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=7 to t=67
U = (45*(pi/180))/G; % Control input (@ 45 deg steering wheel angle)
for i = 7+dt:dt:67
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf;
    FZfr = Sf;
    FZrl = Sr;
    FZrr = Sr;
    af(j) = U - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=67 to t=100
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 67+dt:dt:100
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf;
    FZfr = Sf;
    FZrl = Sr;
    FZrr = Sr;
```

```
    af(j) = U - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Concatenate array of steering angle inputs
d = [zeros(size(0:dt:7)) ones(size(7+dt:dt:67))*(45*(pi/180))/G
 zeros(size(67+dt:dt:100))];

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(1)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')

% Plot ay vs t
figure(2)
plot((0:dt:100),[0 ay],'.','color','red')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${a_y}$ (m/sec$^2$)','interpreter','latex')

% Plot d vs ay
figure(3)
plot([0 ay],d,'.','color','red')
xlabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
ylabel('${\delta}$ (rad)','interpreter','latex')

% Calculate UG (for steady state ay, af, ar values)
ay = ay(50000);
af = af(50000);
```

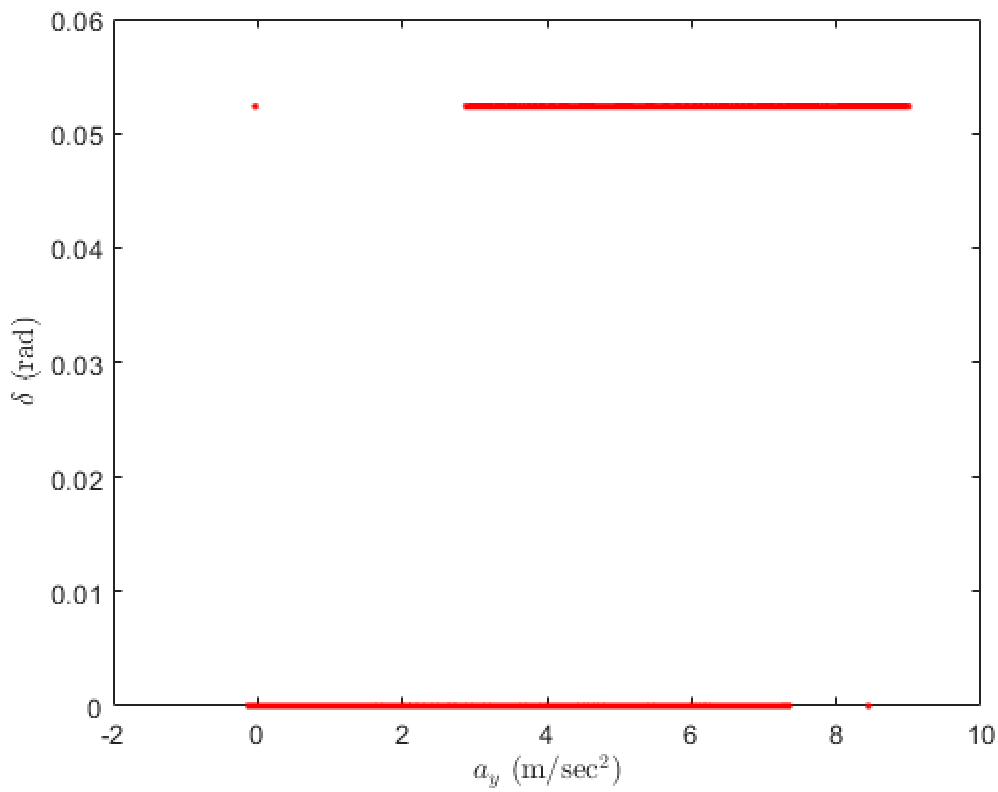
```
ar = ar(50000);
UG = (af-ar)/ay;
fprintf('UG for Maneuver 1 with Non-Linear Tire Model is %f rad sec^2
 m^-1\n',UG)
```

UG for Maneuver 1 with Non-Linear Tire Model is 0.003700 rad sec^2 m^-1





10
8
6
4
2
0
-2
a_y (m/sec^2)
0
10
20
30
40
50
60
70
80
90
100
t (sec)



0.06
0.05
0.04
0.03
0.02
0.01
0
δ (rad)
-2
0
2
4
6
8
10
a_y (m/sec^2)

Maneuver 2

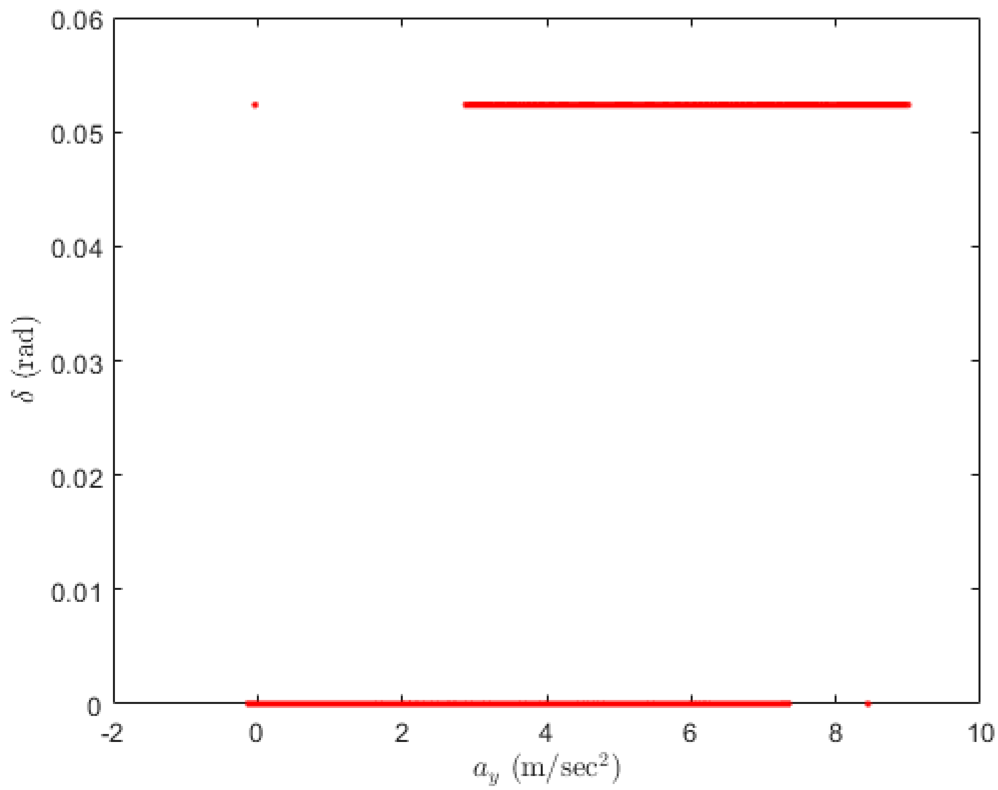
```
% Simulation parameters
v = 50/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
Sf = (m*g*0.6)/2; % 60% front (static) load distribution
Sr = (m*g*0.4)/2; % 60% front (static) load distribution
ay = [];
j = 1;

% Simulate from t=0 to t=5
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:5
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf;
    FZfr = Sf;
    FZrl = Sr;
    FZrr = Sr;
    af = U - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=5 to t=23.621
U = 0:((14.5*(pi/180))/G)*dt:(270*(pi/180))/G; % Control input (@ 0-270 deg
 steering wheel angle)
for i = 5+dt:dt:23.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf;
    FZfr = Sf;
    FZrl = Sr;
    FZrr = Sr;
    af = U(j-5000) - (X(2,j)*a)/v - X(1,j);
```

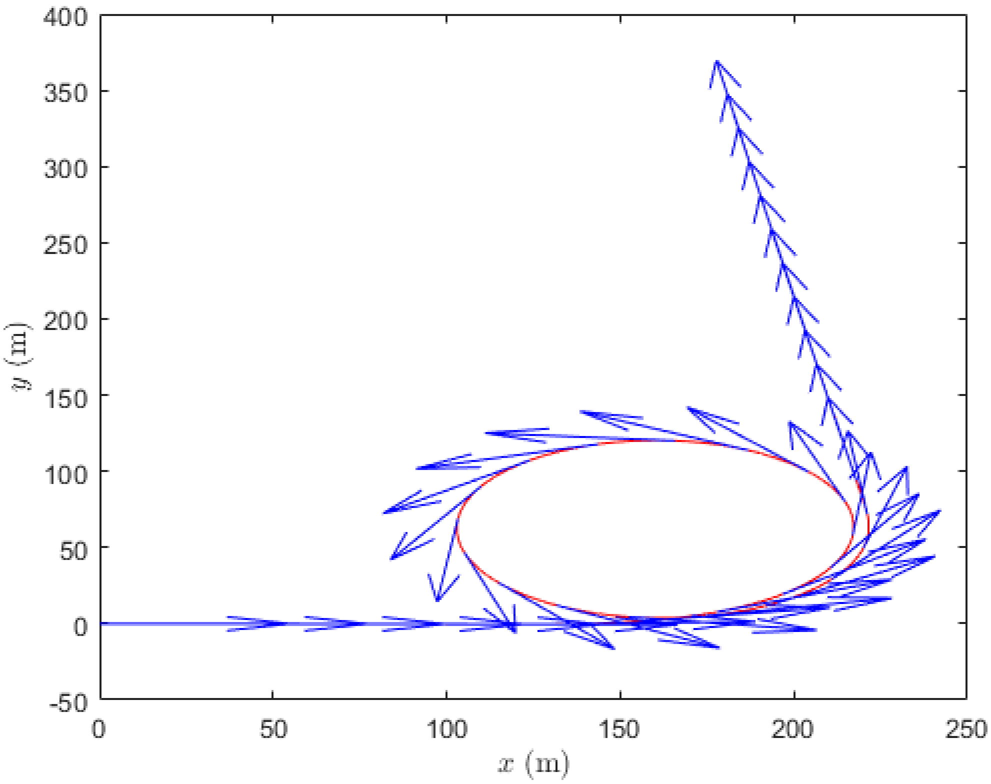
```
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=23.621 to t=27.621
U = (270*(pi/180))/G; % Control input (@ 270 deg steering wheel angle)
for i = 23.621+dt:dt:27.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf;
    FZfr = Sf;
    FZrl = Sr;
    FZrr = Sr;
    af = U - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=27.621 to t=40
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 27.621+dt:dt:40
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf;
    FZfr = Sf;
    FZrl = Sr;
    FZrr = Sr;
    af = U - (X(2,j)*a)/v - X(1,j);
    ar = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af,FZfl,v);
```

```
    FYfr = -nonlintire(af,FZfr,v);
    FYf = FYfl+FYfr;
    FYrl = -nonlintire(ar,FZrl,v);
    FYrr = -nonlintire(ar,FZrr,v);
    FYr = FYrl+FYrr;
    Xdot(1,j) = (FYf+FYr)/(m*v) - X(2,j);
    Xdot(2,j) = (FYf*a-FYr*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Concatenate array of steering angle inputs
d = [zeros(size(0:dt:5)) ones(size(5+dt:dt:23.621))*0:((14.5*(pi/180))/
G)*dt:(270*(pi/180))/G ones(size(23.621+dt:dt:27.621))*(270*(pi/180))/G
 zeros(size(27.621+dt:dt:40))];

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(4)
plot(PosX,PosY,'color','red')
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')

% Plot ay vs t
figure(5)
plot((0:dt:40),[0 ay],'.','color','red')
xlabel('${t}$ (sec)','interpreter','latex')
ylabel('${a_y}$ (m/sec$^2$)','interpreter','latex')

% Plot d vs ay
figure(6)
plot([0 ay],d,'.','color','red')
xlabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
ylabel('${\delta}$ (rad)','interpreter','latex')

% Calculate UG (from linear region of d vs. ay plot)
d_ay = polyfit(ay(5020:8300),d(5020:8300),1);
figure(7)
plot(ay(5200:8300),d(5200:8300),'.','color','red')
hold on
```

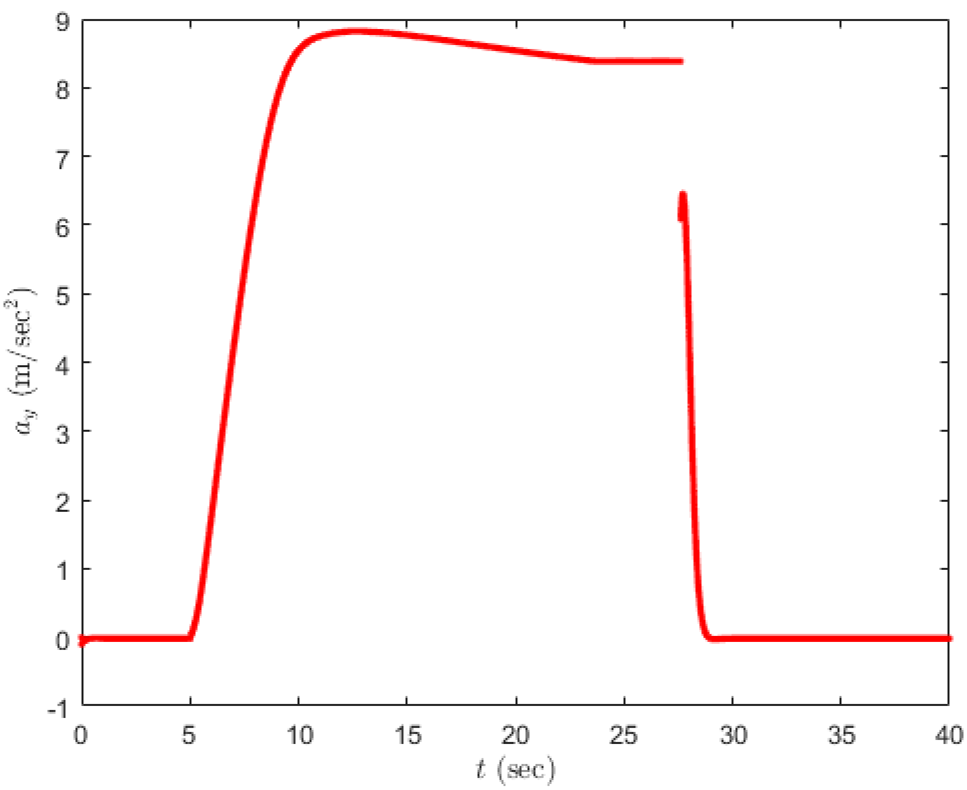
```
plot(ay(5200:8300),polyval(d_ay,ay(5200:8300)),'.','color','blue')
legend('Simulated Data','1st Order Polyfit','Location','NW')
xlabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
ylabel('${\delta}$ (rad)','interpreter','latex')
UG = polyder(d_ay)-(l/v^2);
fprintf('UG for Maneuver 2 with Non-Linear Tire Model is %f rad sec^2
 m^-1\n',UG)
```

UG for Maneuver 2 with Non-Linear Tire Model is 0.002029 rad sec^2 m^-1

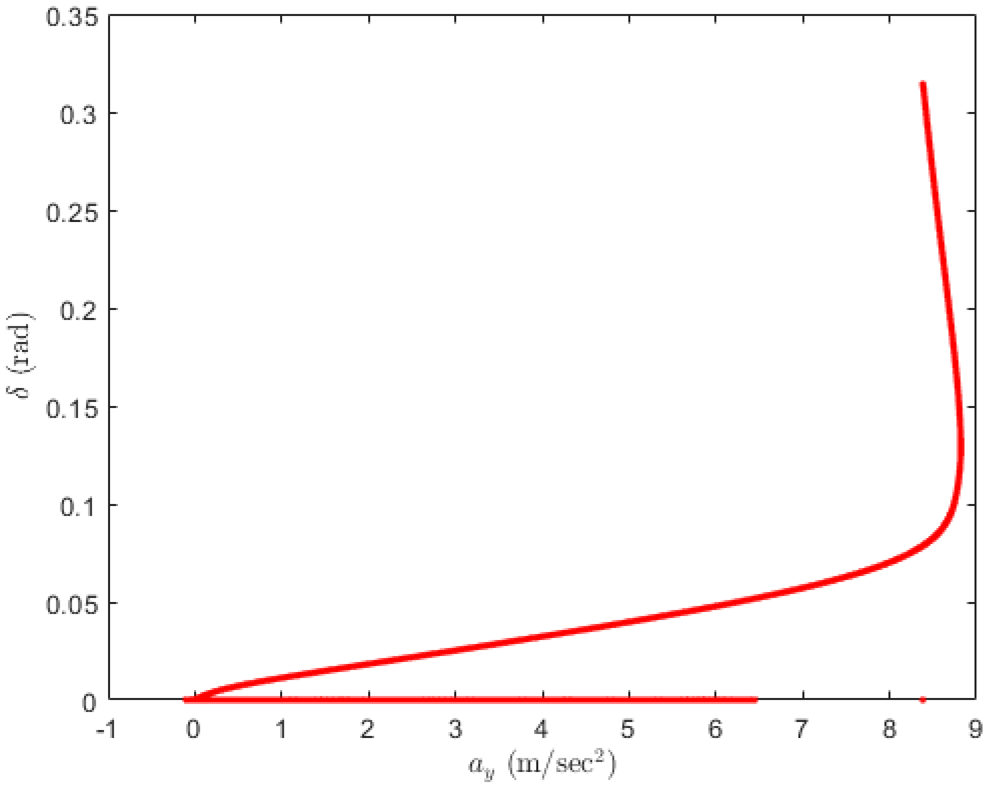


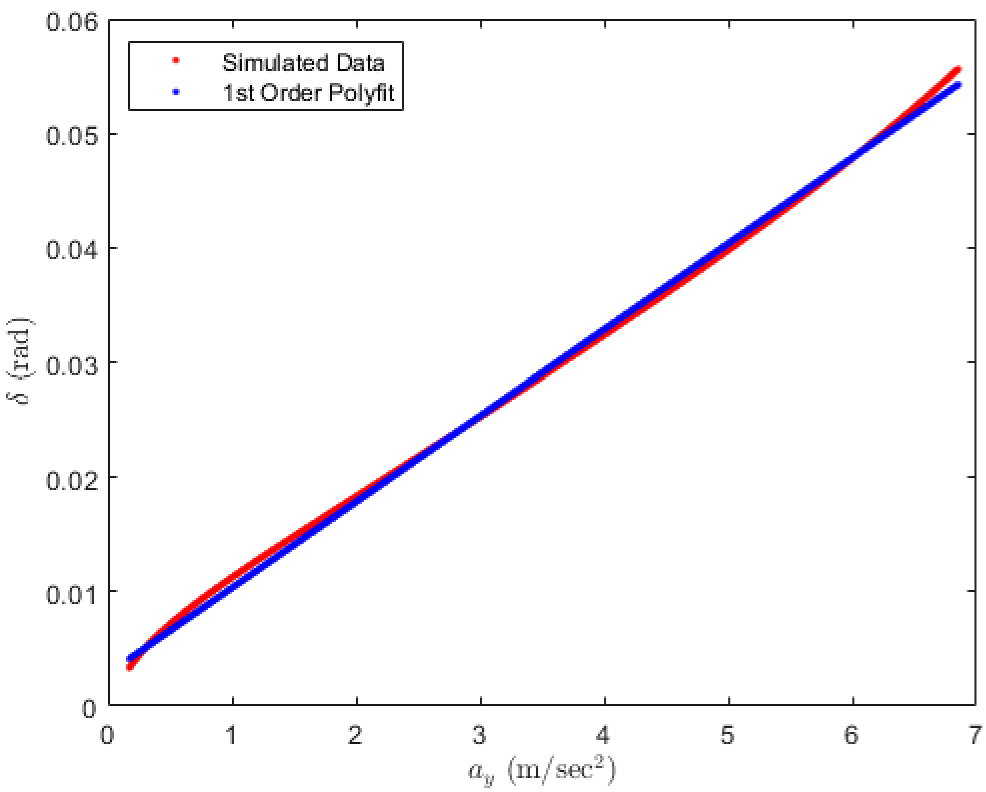


400
350
300
250
200
150
100
50
0
-50
0
50
100
150
200
250
y (m)
x (m)



9
8
7
6
5
4
3
2
1
0
-1
0
5
10
15
20
25
30
35
40
a_y (m/sec^2)
t (sec)


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
δ (rad)
-1
0
1
2
3
4
5
6
7
8
9
a_y (m/sec^2)


Simulated Data
1st Order Polyfit
0.06
0.05
0.04
0.03
0.02
0.01
0
δ (rad)
0
1
2
3
4
5
6
7
a_y (m/sec^2)

Published with MATLAB® R2022a

Table of Contents

Clear Workspace

```
close all;
clear;
clc;
```

Given Data

```
m = 1637; % kg
g = 9.81; % m/s^2
Iz = 3326; % kg-m^2
l = 2.736; % m
t = 1.7; % m
a = 0.4*l; % 60% front load distribution
b = 0.6*l; % 60% front load distribution
G = 15;
h = 2.4/3.281; % m
```

Maneuver 2

```
% Simulation parameters
v = 50/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
Sf = (m*g*0.6)/2; % 60% front (static) load distribution
Sr = (m*g*0.4)/2; % 60% front (static) load distribution
ay = [];
j = 1;

% Simulate from t=0 to t=5
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:5
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
```

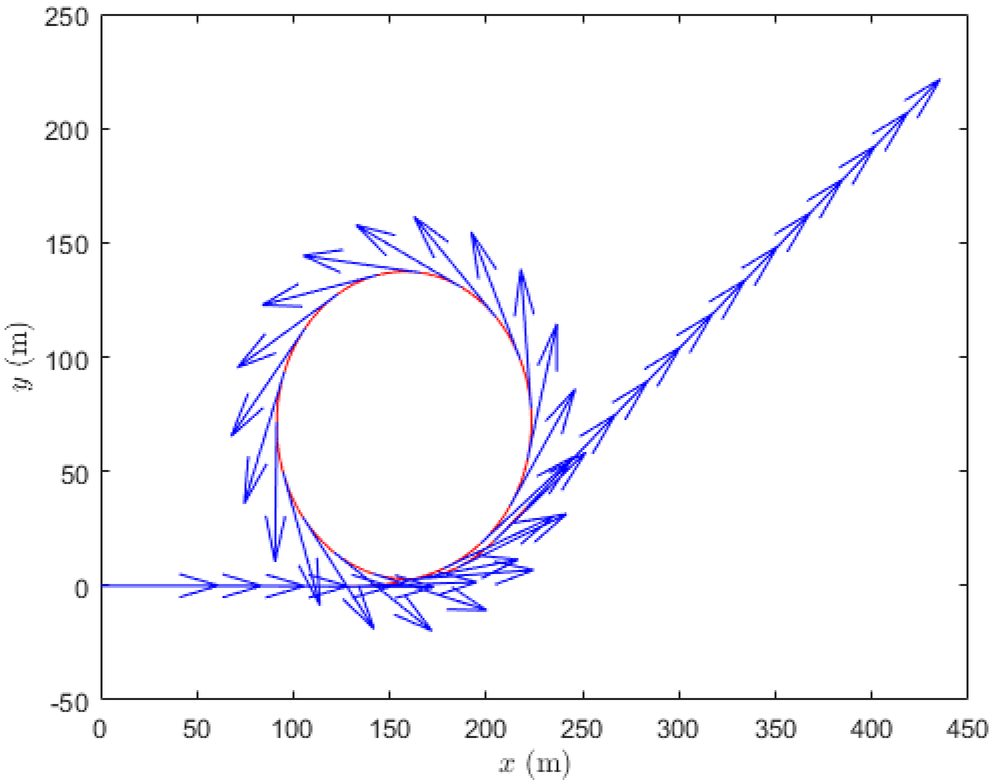
```
    af(j) = U - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=5 to t=23.621
U = 0:((14.5*(pi/180))/G)*dt:(270*(pi/180))/G; % Control input (@ 0-270 deg
 steering wheel angle)
for i = 5+dt:dt:23.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    af(j) = U(j-5000) - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=23.621 to t=27.621
U = (270*(pi/180))/G; % Control input (@ 270 deg steering wheel angle)
for i = 23.621+dt:dt:27.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    af(j) = U - (X(2,j)*a)/v - X(1,j);
```

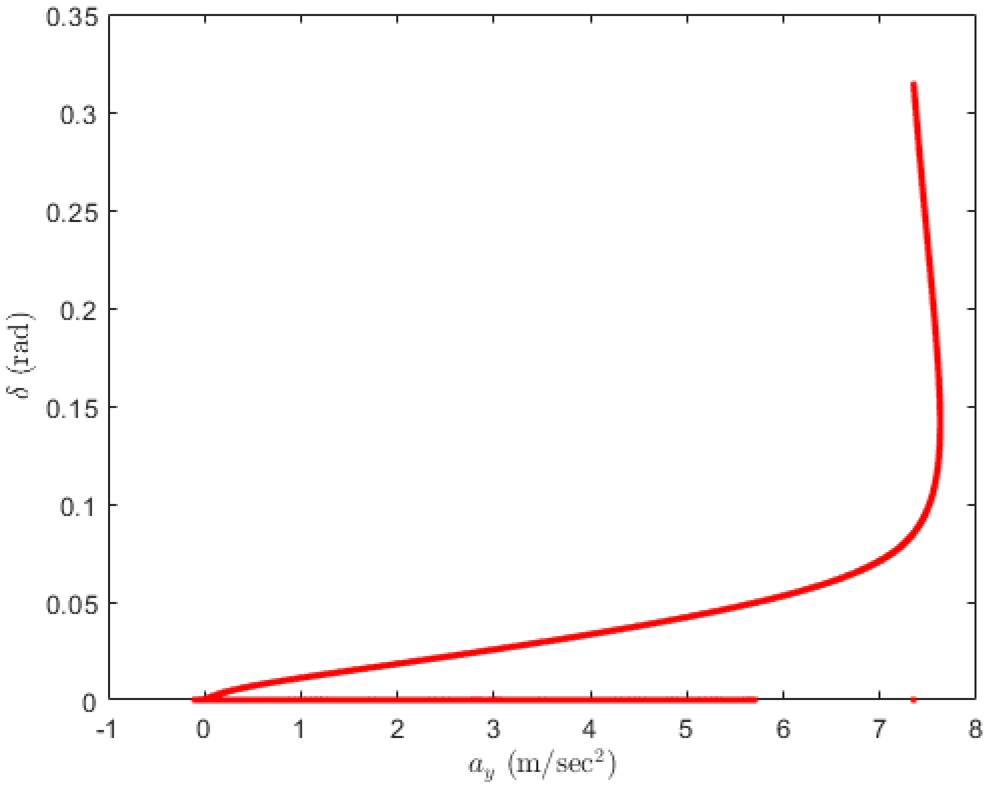
```
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=27.621 to t=40
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 27.621+dt:dt:40
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.6;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.6;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.4;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.4;
    af(j) = U - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(1)
plot(PosX,PosY,'color','red')
```

```
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')

% Concatenate array of steering angle inputs
d = [zeros(size(0:dt:5)) ones(size(5+dt:dt:23.621))*0:((14.5*(pi/180))/
G)*dt:(270*(pi/180))/G ones(size(23.621+dt:dt:27.621))*(270*(pi/180))/G
 zeros(size(27.621+dt:dt:40))];

% Plot d vs ay
figure(2)
plot([0 ay],d,'.','color','red')
xlabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
ylabel('${\delta}$ (rad)','interpreter','latex')

% Calculate sublimit UG (from linear region of d vs. ay plot)
d_ay = polyfit(ay(5020:8300),d(5020:8300),1);
figure(3)
plot(ay(5200:8300),d(5200:8300),'.','color','red')
hold on
plot(ay(5200:8300),polyval(d_ay,ay(5200:8300)),'.','color','blue')
legend('Simulated Data','1st Order Polyfit','Location','NW')
xlabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
ylabel('${\delta}$ (rad)','interpreter','latex')
UG = polyder(d_ay)-(l/v^2);
fprintf('UG for Maneuver 2 with Non-Linear Tire Model and Load Transfer is %f
 rad sec^2 m^-1\n',UG)

% Analyze af and ar to comment on limit behavior
figure(4)
plot(af(5200:8300),FYf(5200:8300),'.','color','red')
hold on
plot(ar(5200:8300),FYr(5200:8300),'.','color','blue')
legend('Front Tire','Rear Tire','Location','NW')
xlabel('${\alpha_{yf/r}}$ (rad)','interpreter','latex')
ylabel('${F_{yf/r}}$ (N)','interpreter','latex')
fprintf(['\nLooking at variation of lateral force vs. slip angle plots for
 front & rear tires (with load transfer), and assuming\n' ...
        'same tire quality (i.e., same breakaway slip angle), we can observe
 that since the slope of the lateral force vs. slip\n' ...
        'angle plot in the linear region (i.e., cornering stiffness) of rear
 tire is less than that of the front tire, it will\n' ...
        'saturate (reach maximum lateral force) before the front tire,
 indicating that the vehicle will be limit oversteer.\n\n'])
```

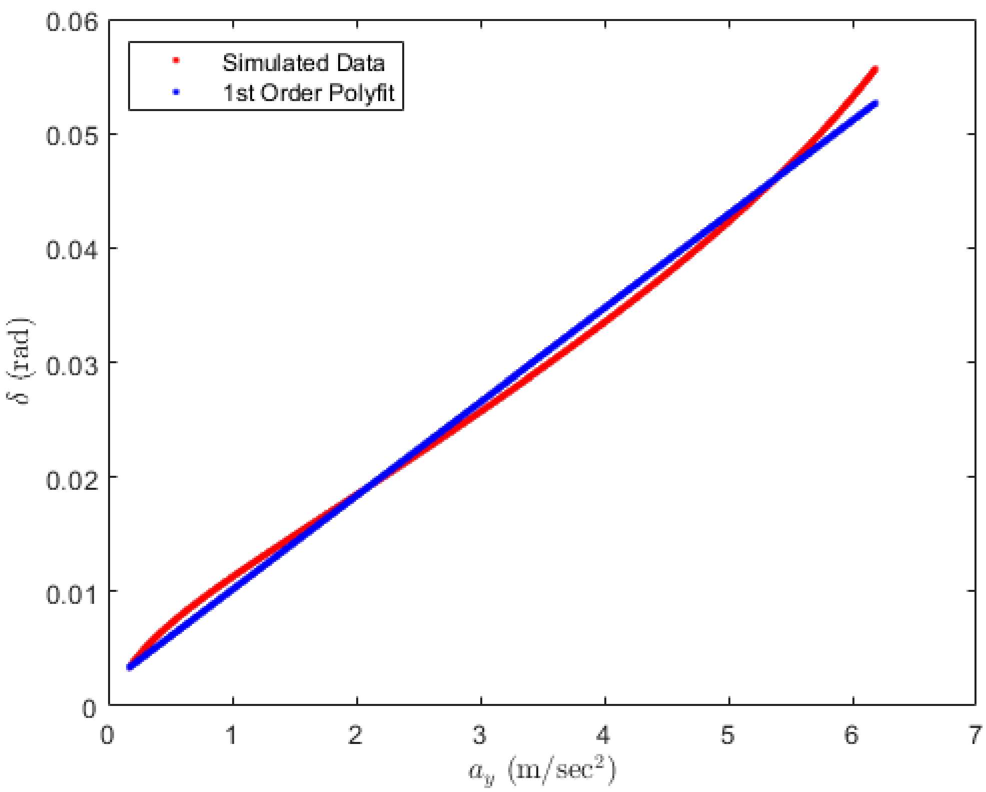
```
UG for Maneuver 2 with Non-Linear Tire Model and Load Transfer is 0.002719 rad
 sec^2 m^-1

Looking at variation of lateral force vs. slip angle plots for front & rear
 tires (with load transfer), and assuming
same tire quality (i.e., same breakaway slip angle), we can observe that since
 the slope of the lateral force vs. slip
```

angle plot in the linear region (i.e., cornering stiffness) of rear tire is less than that of the front tire, it will saturate (reach maximum lateral force) before the front tire, indicating that the vehicle will be limit oversteer.

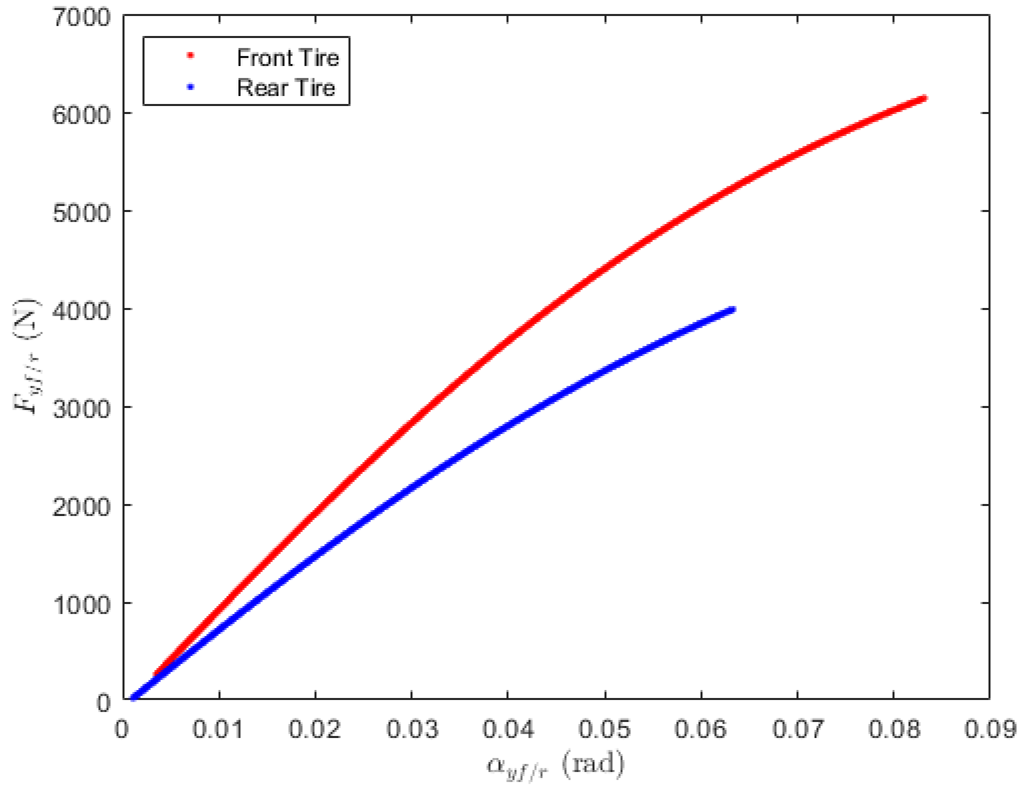




δ (rad)
a_y (m/sec^2)



Simulated Data
1st Order Polyfit
δ (rad)
a_y (m/sec^2)



Published with MATLAB® R2022a

Table of Contents

Clear Workspace

```
close all;
clear;
clc;
```

Given Data

```
m = 1637; % kg
g = 9.81; % m/s^2
Iz = 3326; % kg-m^2
l = 2.736; % m
t = 1.7; % m
a = 0.4*l; % 60% front load distribution
b = 0.6*l; % 60% front load distribution
G = 15;
h = 2.4/3.281; % m
```

Maneuver 2

```
% Simulation parameters
v = 50/2.237; % m/s
dt = 0.001; % s
X = [];
X(:,1) = [0;0];
Xdot = [];
Xint(:,1) = [0;0];
Bdot = 0;
V = [];
P = [0;0];
Sf = (m*g*0.6)/2; % 60% front (static) load distribution
Sr = (m*g*0.4)/2; % 60% front (static) load distribution
ay = [];
j = 1;

% Simulate from t=0 to t=5
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 0+dt:dt:5
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.3;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.3;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.7;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.7;
```

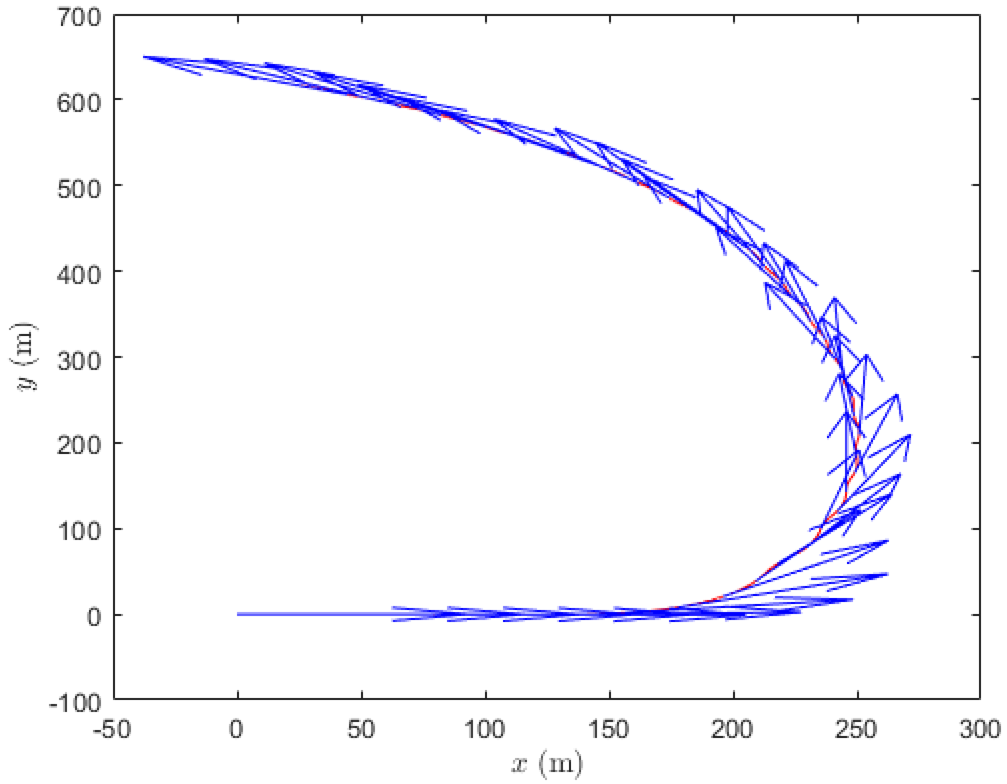
```
    af(j) = U - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=5 to t=23.621
U = 0:((14.5*(pi/180))/G)*dt:(270*(pi/180))/G; % Control input (@ 0-270 deg
 steering wheel angle)
for i = 5+dt:dt:23.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.3;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.3;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.7;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.7;
    af(j) = U(j-5000) - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=23.621 to t=27.621
U = (270*(pi/180))/G; % Control input (@ 270 deg steering wheel angle)
for i = 23.621+dt:dt:27.621
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.3;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.3;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.7;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.7;
    af(j) = U - (X(2,j)*a)/v - X(1,j);
```

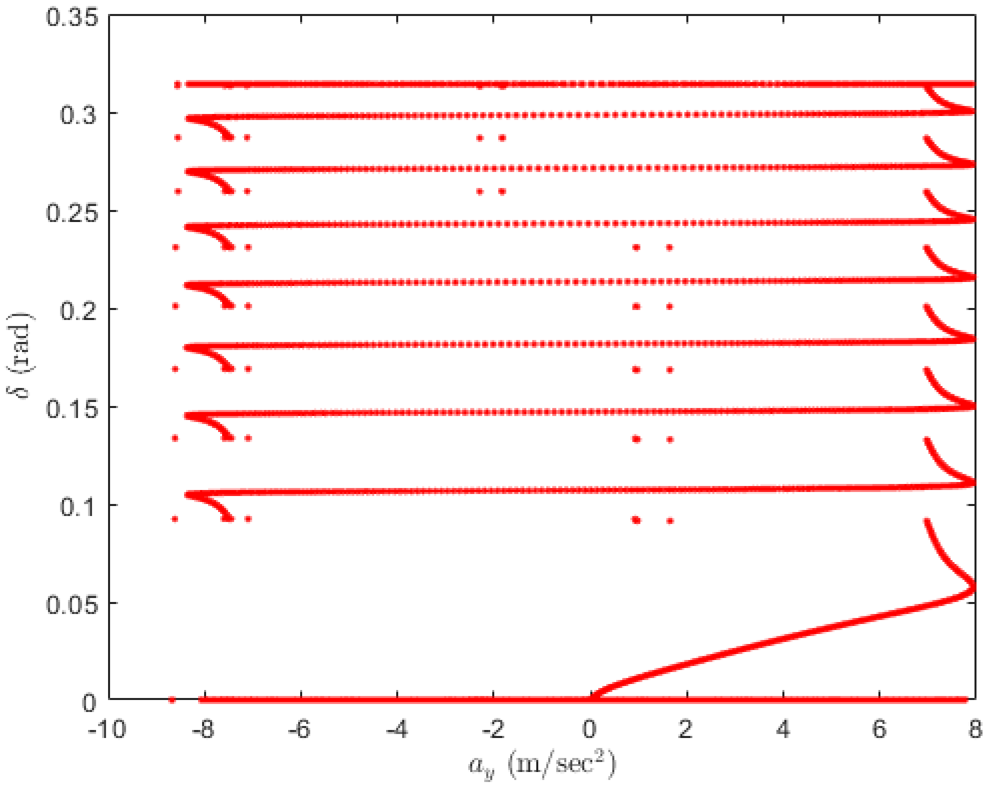
```
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Simulate from t=27.621 to t=40
U = 0; % Control input (@ 0 deg steering wheel angle)
for i = 27.621+dt:dt:40
    ay(j) = v*(Bdot+X(2,j));
    FZfl = Sf - ((m*ay(j)*h)/t)*0.3;
    FZfr = Sf + ((m*ay(j)*h)/t)*0.3;
    FZrl = Sr - ((m*ay(j)*h)/t)*0.7;
    FZrr = Sr + ((m*ay(j)*h)/t)*0.7;
    af(j) = U - (X(2,j)*a)/v - X(1,j);
    ar(j) = (X(2,j)*b)/v - X(1,j);
    FYfl = -nonlintire(af(j),FZfl,v);
    FYfr = -nonlintire(af(j),FZfr,v);
    FYf(j) = FYfl+FYfr;
    FYrl = -nonlintire(ar(j),FZrl,v);
    FYrr = -nonlintire(ar(j),FZrr,v);
    FYr(j) = FYrl+FYrr;
    Xdot(1,j) = (FYf(j)+FYr(j))/(m*v) - X(2,j);
    Xdot(2,j) = (FYf(j)*a-FYr(j)*b)/(Iz);
    X(:,j+1) = X(:,j) + Xdot(:,j)*dt;
    Xint(:,j+1) = Xint(:,j) + X(:,j)*dt;
    V(:,j) = [v*cos(X(1,j)+Xint(2,j)); v*sin(X(1,j)+Xint(2,j))];
    P(:,j+1) = P(:,j) + V(:,j)*dt;
    Bdot = Xdot(1,j);
    j = j+1;
end

% Plot trajectory
PosX = P(1,:);
PosX_sec = PosX(1:1000:end-1000);
PosY = P(2,:);
PosY_sec = PosY(1:1000:end-1000);
VelX = V(1,:);
VelX_sec = VelX(1:1000:end);
VelY = V(2,:);
VelY_sec = VelY(1:1000:end);
figure(1)
plot(PosX,PosY,'color','red')
```

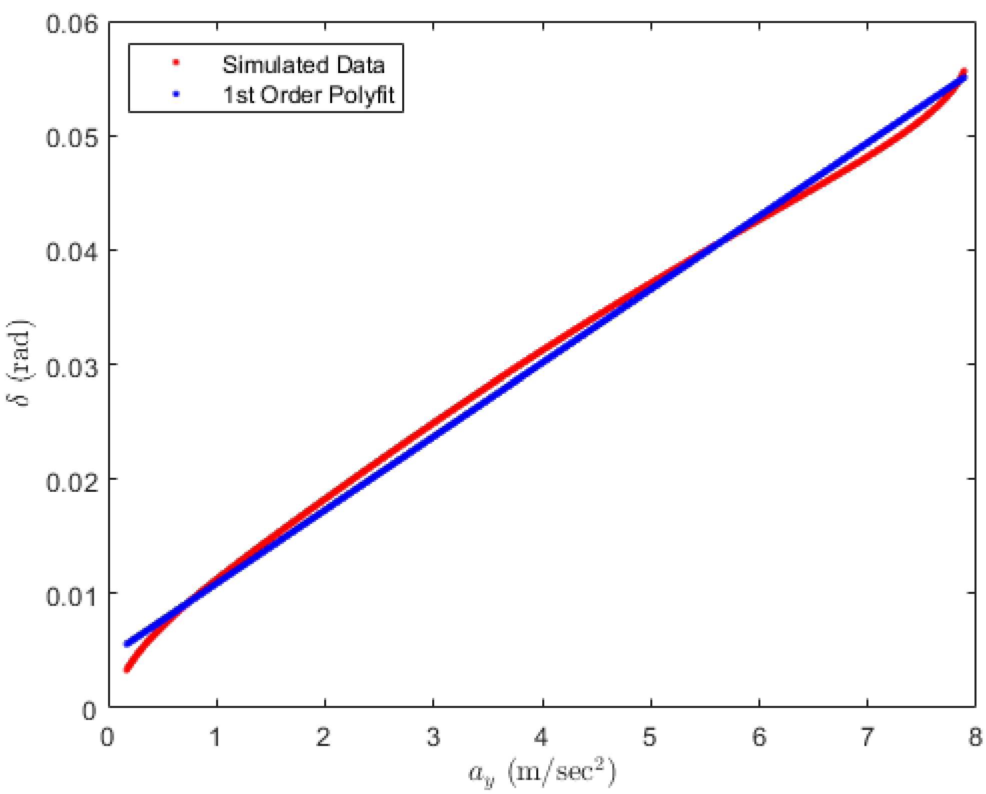
```
%comet(PosX,PosY)
hold on
quiver(PosX_sec,PosY_sec,VelX_sec,VelY_sec,'color','blue')
xlabel('${x}$ (m)','interpreter','latex')
ylabel('${y}$ (m)','interpreter','latex')

% Concatenate array of steering angle inputs
d = [zeros(size(0:dt:5)) ones(size(5+dt:dt:23.621))*0:((14.5*(pi/180))/
G)*dt:(270*(pi/180))/G ones(size(23.621+dt:dt:27.621))*(270*(pi/180))/G
 zeros(size(27.621+dt:dt:40))];

% Plot d vs ay
figure(2)
plot([0 ay],d,'.','color','red')
xlabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
ylabel('${\delta}$ (rad)','interpreter','latex')

% Calculate sublimit UG (from linear region of d vs. ay plot)
d_ay = polyfit(ay(5020:8300),d(5020:8300),1);
figure(3)
plot(ay(5200:8300),d(5200:8300),'.','color','red')
hold on
plot(ay(5200:8300),polyval(d_ay,ay(5200:8300)),'.','color','blue')
legend('Simulated Data','1st Order Polyfit','Location','NW')
xlabel('${a_y}$ (m/sec$^2$)','interpreter','latex')
ylabel('${\delta}$ (rad)','interpreter','latex')
UG = polyder(d_ay)-(l/v^2);
fprintf('UG for Maneuver 2 with Non-Linear Tire Model and Load Transfer is %f
 rad sec^2 m^-1\n',UG)

% Analyze af and ar to comment on limit behavior
figure(4)
plot(af(5200:8300),FYf(5200:8300),'.','color','red')
hold on
plot(ar(5200:8300),FYr(5200:8300),'.','color','blue')
legend('Front Tire','Rear Tire','Location','NW')
xlabel('${\alpha_{yf/r}}$ (rad)','interpreter','latex')
ylabel('${F_{yf/r}}$ (N)','interpreter','latex')
fprintf(['\nLooking at variation of lateral force vs. slip angle plots for
 front & rear tires (with load transfer), and assuming\n' ...
        'same tire quality (i.e., same breakaway slip angle), we can observe
 that since the slope of the lateral force vs. slip\n' ...
        'angle plot in the linear region (i.e., cornering stiffness) of rear
 tire is less than that of the front tire, it will\n' ...
        'saturate (reach maximum lateral force) before the front tire,
 indicating that the vehicle will be limit oversteer.\n\n'])
```

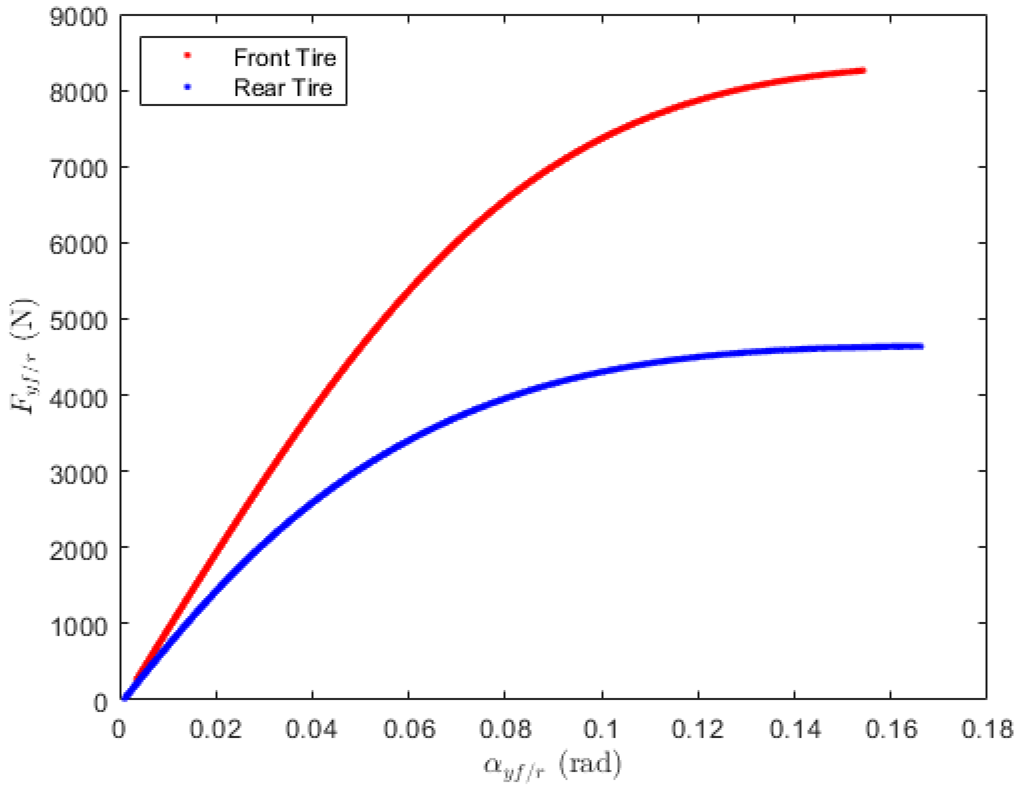
```
UG for Maneuver 2 with Non-Linear Tire Model and Load Transfer is 0.000938 rad
 sec^2 m^-1

Looking at variation of lateral force vs. slip angle plots for front & rear
 tires (with load transfer), and assuming
same tire quality (i.e., same breakaway slip angle), we can observe that since
 the slope of the lateral force vs. slip
```

angle plot in the linear region (i.e., cornering stiffness) of rear tire is less than that of the front tire, it will saturate (reach maximum lateral force) before the front tire, indicating that the vehicle will be limit oversteer.




0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-10
-8
-6
-4
-2
0
2
4
6
8
δ (rad)
a_y (m/sec^2)


Simulated Data
1st Order Polyfit
0.06
0.05
0.04
0.03
0.02
0.01
0
0
1
2
3
4
5
6
7
8
δ (rad)
a_y (m/sec^2)



Published with MATLAB® R2022a