ASSIGNMENT - 6

PROBLEM 1: Linear Tire Model

PART – A

The simplified lateral dynamics of the bicycle model is given as :-

$$F_{YF} - F_Y + F_{YR} = 0$$

$$I_z \ddot{\psi} = -F_{YR} \cdot b + F_{YF} \cdot a$$

For a linear tire model,

$$F_{YF} = C_{\alpha F} \cdot \alpha_F = C_{\alpha F} \cdot \left[\delta - \frac{\dot{\psi} \cdot a}{V} - \beta \right]$$

$$F_{YR} = C_{\alpha R} \cdot \alpha_R = C_{\alpha R} \cdot \left[\frac{\dot{\psi} b}{V} - \beta \right]$$

Substituting these F_{YF} and F_{YR} in the EOMs, and rewriting them in state-space representation :-

→ Now, let $\dot{\psi} = r$ and re-arrange :

$$\dot{\beta} = \overbrace{-\left(\frac{C_{\alpha R} + C_{\alpha F}}{mV}\right)}^{a_{11}} \beta + \overbrace{\left(\frac{C_{\alpha R} \cdot b - C_{\alpha F} \cdot a}{mV^2} - 1\right)}^{a_{12}} r + \underbrace{\left(\frac{C_{\alpha F}}{mV}\right)}_{b_1} \delta$$

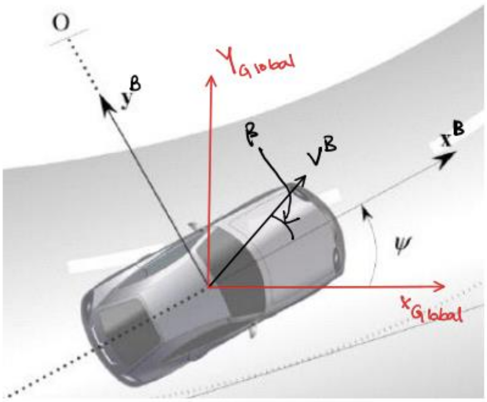
$$\dot{r} = \overbrace{+\left(\frac{C_{\alpha R}\, b - C_{\alpha F}\, a}{I_z}\right)}^{a_{21}} \beta \overbrace{-\left(\frac{C_{\alpha R} b^2 + C_{\alpha F} a^2}{I_z \cdot V}\right)}^{a_{22}} r + \underbrace{\left(\frac{C_{\alpha F} \cdot a}{I_z}\right)}_{b_2} \cdot \delta$$

→ state - space :

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \dot{x} = \underbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}}_{A} \cdot x + \underbrace{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}_{\underline{b}} \delta$$

Solving these equations at every time-step gives us $\dot{x}(t)$. So, we would need to integrate numerically to get $x(t+1)$.

For the location of cg in the global coordinates, we need to factor in the yaw angle ψ in addition to the body slip angle β



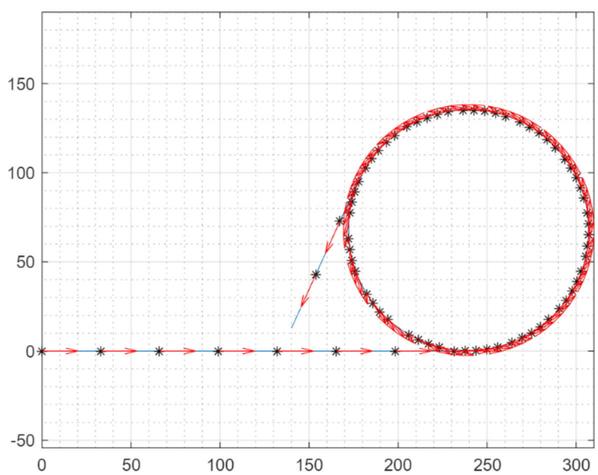
Therefore,

$$\dot{x}_{Global} = V\cos(\beta + \psi)$$

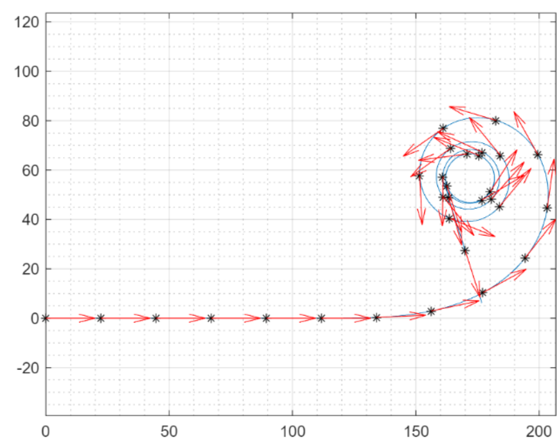
$$\dot{y}_{Global} = V\sin(\beta + \psi)$$

Like before, integrate numerically to get X_{Global} and Y_{global}.

Trajectory – Step Input



Trajectory - Fish hook Maneuver



PART – B

Lateral Load transfer

$$\Longrightarrow \underbrace{F_{zR}}_{\text{Dynamic wheel load}} = \underbrace{\frac{W}{2}}_{\text{Static}} + \underbrace{\frac{W A_y h}{t}}_{\text{Due to } A_y} \qquad \text{with } A_y = a_y/g$$

If true acceleration A_y used (not D'Alembert!), otherwise the signs change!

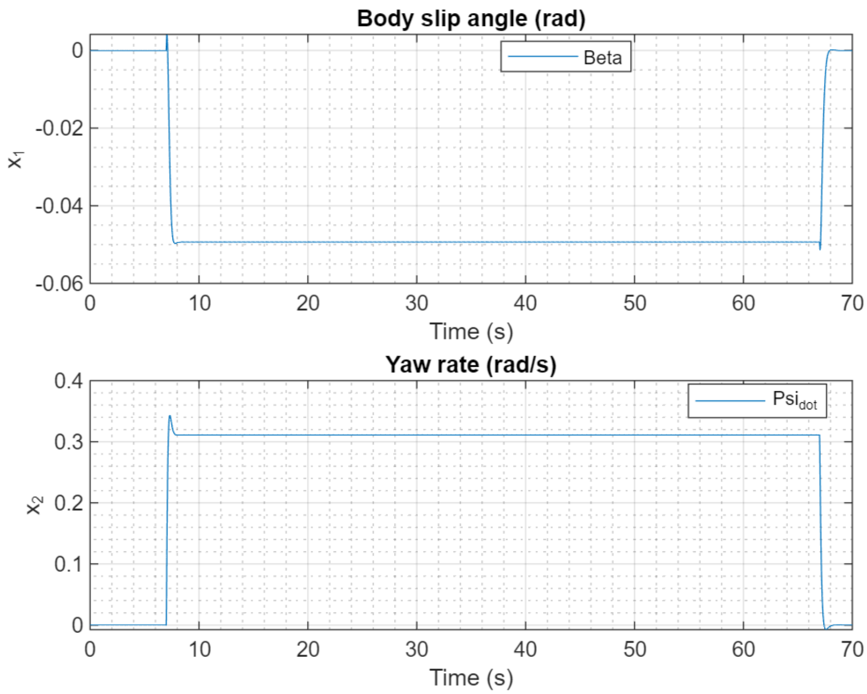
$$\Longrightarrow F_{zL} = \frac{W}{2} - \frac{W A_y h}{t}$$

$\Rightarrow$ Total load transferred in cornering is:

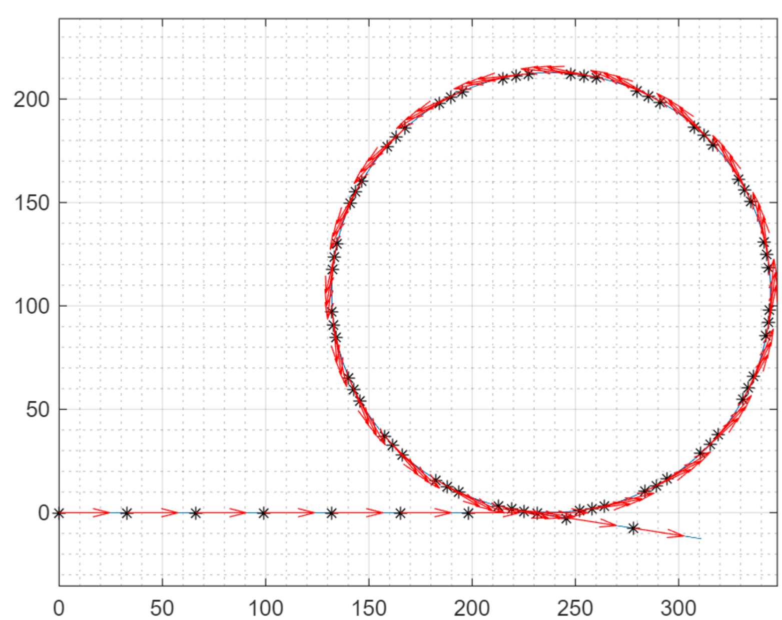
$$\boxed{\Delta W_y = \frac{W A_y h}{t}}$$

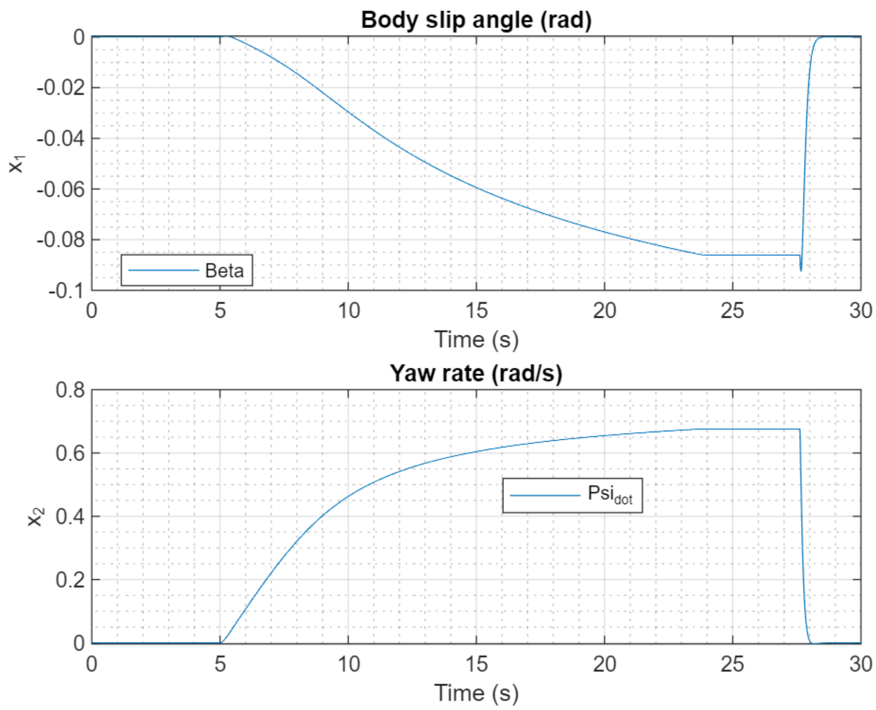
$$\Delta W_{y*} = \frac{W A_y H}{t} \left(\frac{K_{\phi *}}{\Sigma K_{\phi *}} \right) \leftarrow K_{\phi F} + K_{\phi R}$$

Front or Rear

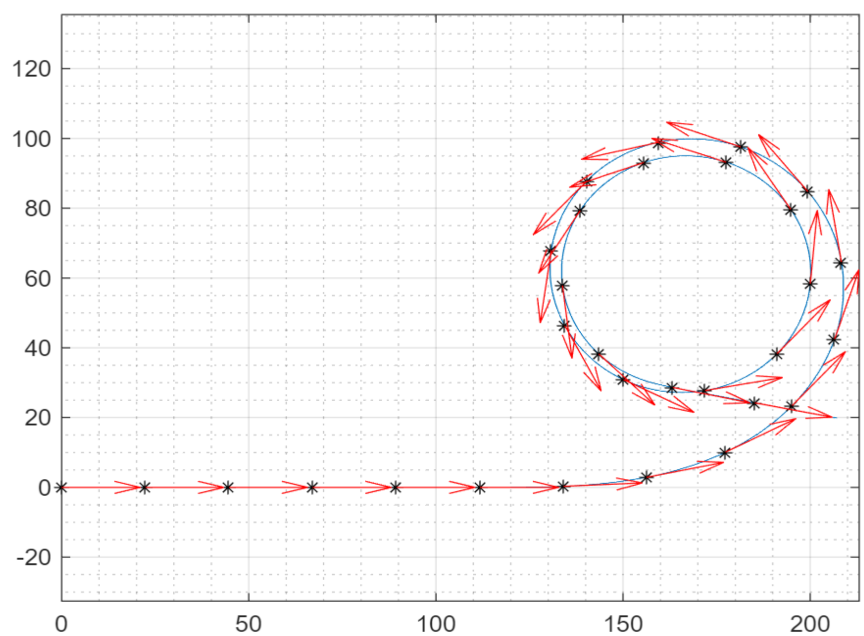


Trajectory – Step Input





Trajectory – Fishhook Maneuver



PROBLEM – 2: Non-Linear Tire model

We know that the EOMs for the bicycle model is as below:

$$F_{YF} - F_Y + F_{YR} = 0$$

$$I_z \ddot{\psi} = -F_{YR} \cdot b + F_{YF} \cdot a$$

Now that we are considering a non-linear tire model, we need to use the nonlintire.m file to get F_{YF} and F_{YR} for these equations. The nonlintire.m takes in F_z, α and v_{wheel} for input and outputs F_{ywheel}.

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

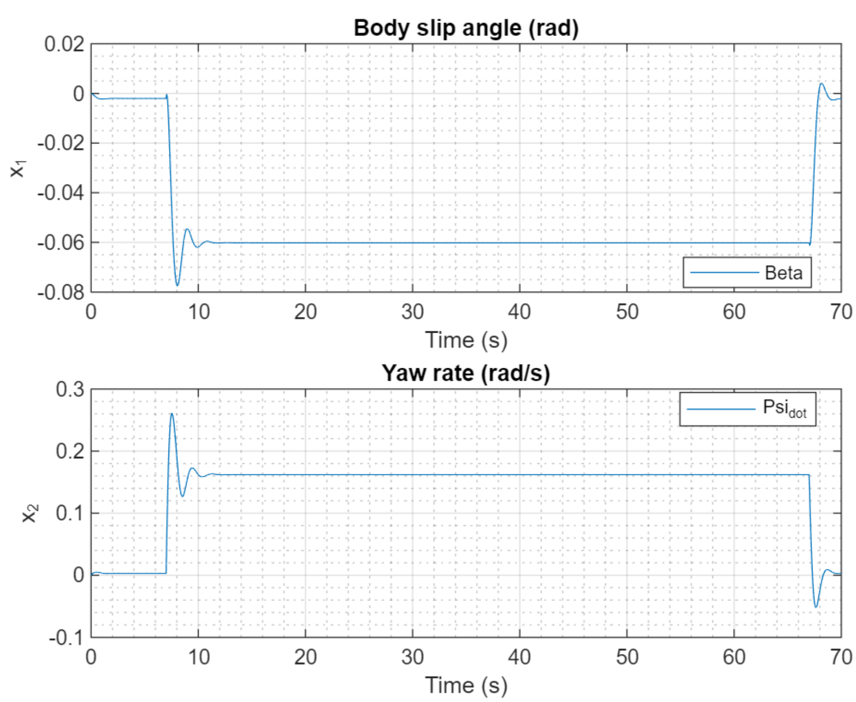
$$\alpha_R = \frac{\dot{\psi} b}{v} - \beta$$

$$v_f = v_{cg} + \dot{\psi} a$$

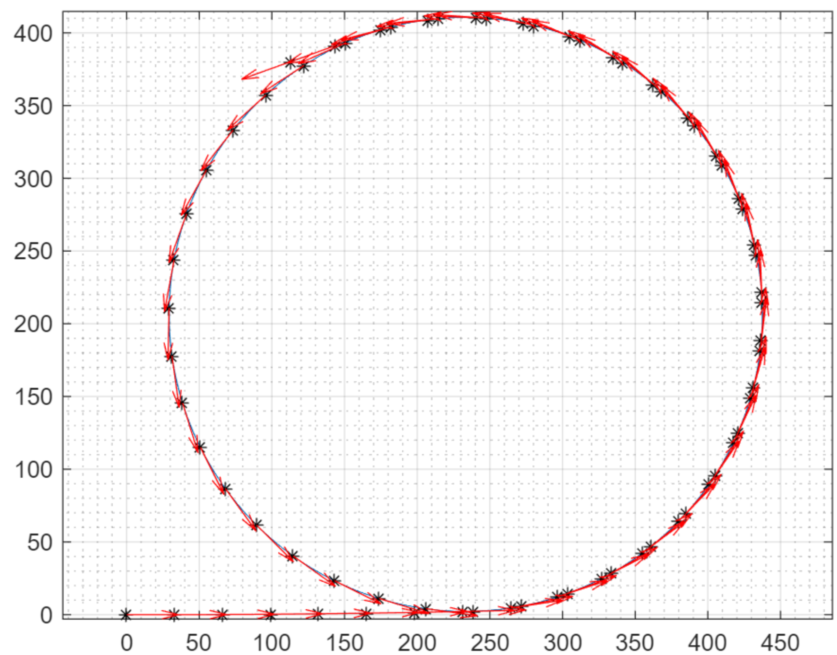
$$v_R = v_{cg} - \dot{\psi} b$$

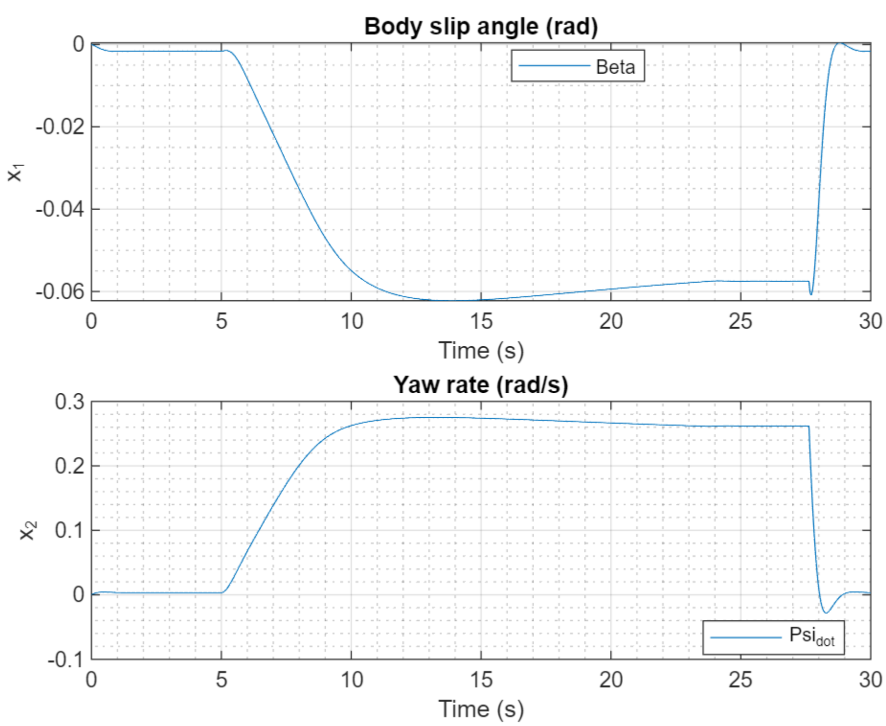
Solve for $\dot{x}$ and integrate to get x.

PART- A

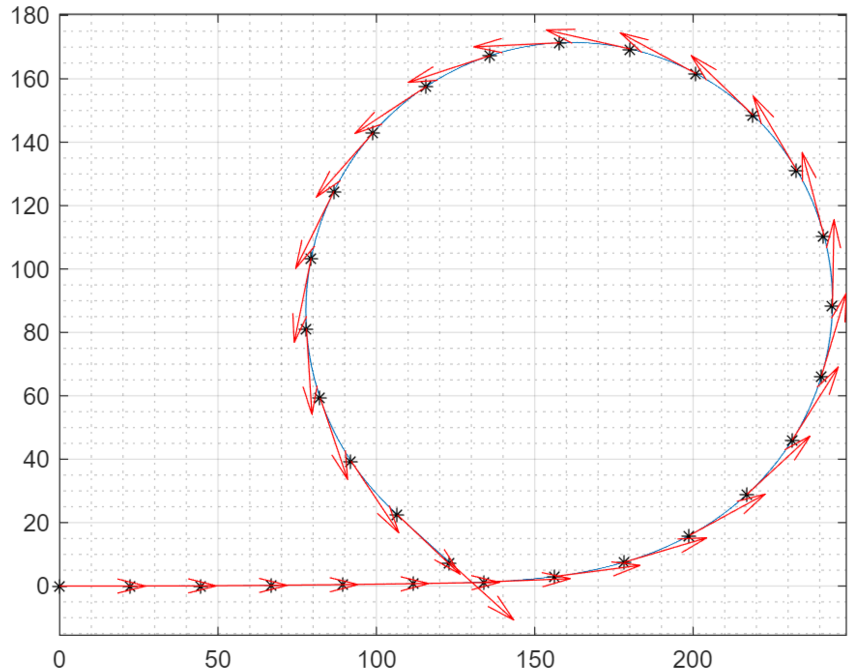


Trajectory – Step Input





Trajectory – Fishhook Maneuver



PART - B

Question 1:

```
The understeer gradient of step input is 7.393066e-03 rad/g
The understeer gradient of fishhook maneuver is 3.195825e-02 rad/g
```

Question 2:

```
The understeer gradient of fishhook maneuver considering load transfer is 2.385783e-02 rad/g
```

Comment:

It can be seen that the understeer gradient, decreases from the previous case, where we did not consider load transfer. In this case, the dynamic loads of the vehicle changes due to the lateral load transfer.

→ Cornering forces of pneumatic tires depend on load (see section on tires)
→ In hard cornering: loads typically might change to lower values on the inside wheel and higher values on the outside wheel
→ Average force from both tires will be reduced
→ Tires will have to assume a greater slip angle to maintain the lateral force necessary for turn
→ If this happens in the front —> car ploughs outwards —> understeer
→ If this happens in the rear —> rear slips outward —> oversteer

Question 3:

```
The understeer gradient of fishhook maneuver considering load transfer with 70% rear and 30% front roll
distribution is -1.305388e-02 rad/g
```

Comment:

From one of the previously shown figures, the load transfer depends on the roll stiffness distribution, for this case 30 % front stiffness causes less load transfer at the front and more at the rear, leading to rear tire saturation. Hence oversteer (negative UG value).

Appendix:

```
clc; clear; close all;
maneuver = 1;
tire_model = 0;


[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_wllt(maneuver,tire_model);


% Trajectory plot
clf;
delta_t = 0.001;
plot(x(3,:),x(4,:))
hold on
plot(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),'k*')
quiver(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-
1),xDot(3,1:(1/delta_t):end),xDot(4,2:(1/delta_t):end),0.5,'Color','r')
hold off
grid on; grid minor;
axis equal
```

```
%% Maneuver 2
maneuver = 2;
tire_model = 0;


[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_wllt(maneuver,tire_model);


% Trajectory plot
delta_t = 0.001;
clf;
plot(x(3,:),x(4,:))
hold on
plot(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),'k*')
quiver(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-
1),xDot(3,1:(1/delta_t):end),xDot(4,2:(1/delta_t):end),0.5,'Color','r')
hold off
grid on; grid minor;
axis equal
```

PART - B

```
%MANUEVER 1
maneuver = 1;
tire_model = 0;




[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_llt(maneuver,tire_model,1);
```

```
%% Plots


% Body slip angle and yaw rate vs time


clf;
subplot(2,1,1);
plot(time,x(1,:));
grid on; grid minor;
xlabel('Time (s)');
ylabel('x_1');
legend('Beta','Location','Best');
title('Body slip angle (rad)')
subplot(2,1,2);
plot(time,x(2,:));
grid on; grid minor;
xlabel('Time (s)');
ylabel('x_2');
legend('Psi_{dot}','Location','Best');
title('Yaw rate (rad/s)')


% Trajectory plot
clf;
plot(x(3,:),x(4,:))
hold on
plot(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),'k*')
quiver(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),xDot(3,1:(1/delta_t):end-
1),xDot(4,2:(1/delta_t):end-1),0.5,'Color','r')
hold off
grid on; grid minor;
axis equal


% MANEUVER -2
maneuver = 2;
tire_model = 0;




%% Simulating model
[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_llt(maneuver,tire_model,1);


%% Plots
% Body slip angle and yaw rate vs time


clf;
subplot(2,1,1);
plot(time,x(1,:));
grid on; grid minor;
```

```
xlabel('Time (s)');
ylabel('x_1');
legend('Beta','Location','Best');
title('Body slip angle (rad)')
subplot(2,1,2);
plot(time,x(2,:));
grid on; grid minor;
xlabel('Time (s)');
ylabel('x_2');
legend('Psi_{dot}','Location','Best');
title('Yaw rate (rad/s)')


% Trajectory plot


clf;
plot(x(3,:),x(4,:))
hold on
plot(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),'k*')
quiver(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),xDot(3,1:(1/delta_t):end-
1),xDot(4,2:(1/delta_t):end-1),0.5,'Color','r')
hold off
grid on; grid minor;
axis equal
```

```
maneuver = 1;
tire_model = 1;
k=0;


%Simulation
[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_wllt(maneuver,tire_model);




% Body slip angle and yaw rate vs time
clf;
subplot(2,1,1);
plot(time,x(1,:));
grid on; grid minor;
xlabel('Time (s)');
ylabel('x_1');
legend('Beta','Location','Best');
title('Body slip angle (rad)')
subplot(2,1,2);
plot(time,x(2,:));
grid on; grid minor;
xlabel('Time (s)');
ylabel('x_2');
legend('Psi_{dot}','Location','Best');
title('Yaw rate (rad/s)')
```

```matlab
% Trajectory plot
clf;
plot(x(3,:),x(4,:))
hold on
plot(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),'k*')
quiver(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-
1),xDot(3,1:(1/delta_t):end),xDot(4,2:(1/delta_t):end),0.5,'Color','r')
hold off
grid on; grid minor;
axis equal


% Understeer gradient
f2=polyfit(ay,deltavec(1,2:end),1);
l = 2.736; %m 2.66
v = 74*0.44704; %m/s
UG1=f2(1)-l/(v^2);


%% Maneuver 2
maneuver = 2;
tire_model = 1;


%simulation
[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_wllt(maneuver,tire_model);




% Body slip angle and yaw rate vs time


clf;
subplot(2,1,1);
plot(time,x(1,:));
grid on; grid minor;
xlabel('Time (s)');
ylabel('x_1');
legend('Beta','Location','Best');
title('Body slip angle (rad)')
subplot(2,1,2);
plot(time,x(2,:));
grid on; grid minor;
xlabel('Time (s)');
ylabel('x_2');
legend('Psi_{dot}','Location','Best');
title('Yaw rate (rad/s)')
% Trajectory plot


clf;
plot(x(3,:),x(4,:))
hold on
plot(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-1),'k*')
```

```
quiver(x(3,1:(1/delta_t):end-1),x(4,1:(1/delta_t):end-
1),xDot(3,1:(1/delta_t):end),xDot(4,2:(1/delta_t):end),0.5,'Color','r')
hold off
grid on; grid minor;
axis equal


% Understeer gradient
f2=polyfit(ay,deltavec(1,1:end-1),1);
l = 2.736; %m 2.66
v = 50*0.44704;
UG2=f2(1)-l/(v^2);
```

PART - B

Question 1:

```
fprintf('The understeer gradient of step input is %d rad/g',UG1);
fprintf('The understeer gradient of fishhook maneuver is %d rad/g',UG2);
```

Question 2:

```
clear all
delta_t = 0.001;
maneuver = 2;
tire_model = 1;


%% Simulating model


[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_llt(maneuver,tire_model,1);


f2=polyfit(ay,deltavec,1);
l = 2.736; %m 2.66
v = 50*0.44704;
UG=f2(1)-l/(v^2);
fprintf('The understeer gradient of fishhook maneuver considering load transfer is %d
rad/g',UG);
```

Question 3:

```
maneuver = 2;
tire_model = 1;


%% Simulating model
[x,xDot,alphaf,alphar,ay,deltavec,time] =simulation_llt(maneuver,tire_model,2);


f1=polyfit(ay,deltavec,1);
l = 2.736; %m 2.66
v = 50*0.44704;
UGb=f1(1)-l/(v^2);
fprintf('The understeer gradient of fishhook maneuver considering load transfer with 70%%
rear and 30%% front roll distribution is %d rad/g',UGb);
```

```
function [x,xDot,alphaf,alphar,ay,deltavec,time] = simulation_wllt(maneuver,tire_model)
```

```
% Given
m = 1637; %kg 1150
Iz = 3326; %kg-m^2 1850
l = 2.736; %m 2.66
a = 0.4*l; %m 1.064
b = 0.6*l; %m 1.596
h=2*0.3048; % m height of Cg
t=1.7; % m trackwidth
cf = 2*1500*180/pi; %N/rad 2*1312*180/pi
cr = 2*1146*180/pi; %N/rad 2*984*180/pi
v = 50*0.44704; %m/s, maneuver1=70mph; maneuver2=50mph
Fzf = m*9.81*b/l; %N, Front tire normal load
Fzr = m*9.81-Fzf; %N, Rear tire normal load


if maneuver == 1 % Step input
    % Time Vector
    T = 70; % Maneuver1=67 + 60 seconds; Maneuver2=29 seconds;
    delta_t = 0.001;
    time = linspace(0,T,(T/delta_t)+1);
    delta_sw = zeros(1,(T/delta_t)+1);
    index1 = (7/delta_t)+1;
    index2 = (67/delta_t)+1;
    index3= (125/delta_t)+1;
    delta_sw(index1+1:index2) = 45/15;
    %delta_sw(index2+5000:index3) = -45/15;
    deltavec = delta_sw*pi/180;
    v = 74*0.44704; %m/s, maneuver1=74mph;
else % Fish hook maneuver
    % Time Vector
    T = 30;
    delta_t = 0.001;
    time = linspace(0,T,(T/delta_t)+1);
    % indices for steering angle vector
    index0 = (5/delta_t) + 1; % 5 sec
    index1 = int16((5+270/14.5)/delta_t) + 1; % next 18 sec
    index2 = int16((5+270/14.5+4)/delta_t) + 1; % next 4 sec
    delta_sw = zeros(1,(T/delta_t)+1); % initialization of Steering angle vector
    delta_sw(1,index0+1:index1) = 14.5 * time(1,1:(index1-index0)); % linear increase in
steering angle
    delta_sw(1,index1+1:index2) = 270;
    delta_sw(1,index2+1:end) = 0;
    deltavec = deg2rad((delta_sw./15)); % converting handwheel angle in deg to road wheel
angle in rad
    v = 50*0.44704; %m/s,  maneuver2=50mph
end
x=zeros(5,1);
xDot=zeros(5,length(x));
for i = 1:(T/delta_t)
    if tire_model == 0
        xdot = bicyclemodel_linear(x(:,i),deltavec(i),v);
    else
        xdot = bicyclemodel_nonlinear(x(:,i),deltavec(i),v);
```

```
    end
 x(:,i+1) = x(:,i) + delta_t * xdot;
 xDot(:,i)=xdot;
end


alphaf=deltavec-x(2,:)*a/v-x(1,:);
alphar=x(2,:)*b/v-x(1,:);
ay=v*(xDot(1,:)+x(2,1:end-1));


end




function [x,xDot,alphaf,alphar,ay,deltavec,time] = simulation_llt(maneuver,tire_model,f)


% Given
m = 1637; %kg 1150
Iz = 3326; %kg-m^2 1850
l = 2.736; %m 2.66
a = 0.4*l; %m 1.064
b = 0.6*l; %m 1.596
h=2*0.3048; % m height of Cg
t=1.7; % m trackwidth
cf = 2*1500*180/pi; %N/rad 2*1312*180/pi
cr = 2*1146*180/pi; %N/rad 2*984*180/pi
v = 50*0.44704; %m/s, maneuver1=70mph; maneuver2=50mph
Fzf = m*9.81*b/l; %N, Front tire normal load
Fzr = m*9.81-Fzf; %N, Rear tire normal load
%var=f;


if maneuver == 1 % Step input
    T = 70; % Maneuver1=67 + 60 seconds; Maneuver2=29 seconds;
    delta_t = 0.001;
    time = linspace(0,T,(T/delta_t)+1);
    delta_sw = zeros(1,(T/delta_t)+1);
    index1 = (7/delta_t)+1;
    index2 = (67/delta_t)+1;
    index3= (125/delta_t)+1;
    delta_sw(index1+1:index2) = 45/15;
    %delta_sw(index2+5000:index3) = -45/15;
    deltavec = delta_sw*pi/180;
    v = 74*0.44704; %m/s, maneuver1=74mph;
else % Fish hook maneuver
    T = 30;
    delta_t = 0.001;
    time = linspace(0,T,(T/delta_t)+1);
    % indices for steering angle vector
    index0 = (5/delta_t) + 1; % 5 sec
    index1 = int16((5+270/14.5)/delta_t) + 1; % next 18 sec
    index2 = int16((5+270/14.5+4)/delta_t) + 1; % next 4 sec
    delta_sw = zeros(1,(T/delta_t)+1); % initialization of Steering angle vector
```

```
    delta_sw(1,index0+1:index1) = 14.5 * time(1,1:(index1-index0)); % linear increase in
steering angle
    delta_sw(1,index1+1:index2) = 270;
    delta_sw(1,index2+1:end) = 0;
    deltavec = deg2rad((delta_sw./15)); % converting handwheel angle in deg to road wheel
angle in rad
    v = 50*0.44704; %m/s,  maneuver2=50mph
end


x=zeros(5,1);
xDot=zeros(5,length(time));
for i = 1:(T/delta_t)
    if tire_model == 0
        xdot = bicyclemodel_linear_llt(x(:,i),deltavec(i),v,xDot(:,i),f);
    else % For roll stiffness distribution of 30-70 Please change the...
        % value of f from 1 to 2
        xdot = bicyclemodel_nonlinear_llt(x(:,i),deltavec(i),v,xDot(:,i),f);
    end
 x(:,i+1) = x(:,i) + delta_t * xdot;
 xDot(:,i)=xdot;
end
alphaf=deltavec-x(2,:)*a/v-x(1,:);
alphar=x(2,:)*b/v-x(1,:);
ay=v*(xDot(1,:)+x(2,:));




end
%Functions
```

```
%% Non linear tire model with llt
function [xdot1] = bicyclemodel_nonlinear_llt(x,delta,v,xDot,f)
% Creating a function to calculate differential equation values with input
% ... x - 5 states of (Vehicle Slip angle, yaw velocity, longitudinal
% velocity of CG, lateral velocity of CG and yaw angle) Steering angle in
% rad, xDot and parameter for roll stiffness distribution

m = 1637; %kg 1150
Iz = 3326; %kg-m^2 1850
l = 2.736; %m 2.66
a = 0.4*l; %m 1.064
b = 0.6*l; %m 1.596
g=9.81; %m/s^2 acceleration due to gravity
h=2.4*0.3048; % m height of Cg
t=1.7; % m trackwidth
roll_parameters =[0.6 0.3]; % roll stiffness distribution values
kf=roll_parameters(f); % front roll stiffness distribution in percentage
kr=1-kf; % rear roll stiffness distribution in percentage
% Static loads
Wf= m*g*b/l; % Static load on front axle in N
Wr=m*g*a/l; % Static load on rear axle in N
% acceleration in lateral direction
ay=v*(xDot(1)+x(2)); % Taking lateral acceleration from betadot and yaw velocity value
% Dynamic loads

Fzfl=Wf/2-(m*ay*h/t)*kf; % Dynamic load on front left in N
```

```
Fzfr=Wf/2+(m*ay*h/t)*kf; % Dynamic load on front right in N
Fzrl=Wr/2-(m*ay*h/t)*kr; % Dynamic load on rear left in N
Fzrr=Wr/2+(m*ay*h/t)*kr; % Dynamic load on rear right in N


beta=x(1);
r=x(2);
%  xcg=x(3);
%  ycg=x(4);
 psi=x(5);

 %Slip angles
 alphaf = delta - (r*a/v) - beta;
 alphar = (r*b/v) - beta;

 %Wheel velocities
 vf = v + r*a;
 vr = v - r*b;

 %Lateral force for each tire
 Fyfl = -1*nonlintire(alphaf,Fzfl,vf); % front left
 Fyrl = -1*nonlintire(alphar,Fzrl,vr); % rear left
 Fyfr = -1*nonlintire(alphaf,Fzfr,vf); % front right
 Fyrr = -nonlintire(alphar,Fzrr,vr); % rear right

 % lateral force for each axle
 Fyf = Fyfl+Fyfr; % front axle
 Fyr = Fyrl + Fyrr; % rear axle


 %State derivatives
 rdot = (-Fyr*b + Fyf*a)/Iz;
 betadot = (Fyf+Fyr)/m/v - r;
 xcgdot = v*cos(beta + psi);
 ycgdot = v*sin(beta + psi);
 psidot = r;

 xdot1 = [betadot;...
 rdot;...
 xcgdot;...
 ycgdot;...
 psidot];
end


function [xdot1] = bicyclemodel_nonlinear(x,delta,v)

m = 1637; %kg 1150
Iz = 3326; %kg-m^2 1850
l = 2.736; %m 2.66
a = 0.4*l; %m 1.064
b = 0.6*l; %m 1.596
Fzf = m*9.81*b/l; %N, Front tire normal load
Fzr = m*9.81-Fzf; %N, Rear tire normal load
beta=x(1);
r=x(2);
%  xcg=x(3);
%  ycg=x(4);
 psi=x(5);

 %Slip angles
```

```
 alphaf = delta - (r*a/v) - beta;
 alphar = (r*b/v) - beta;

 %Wheel velocities
 vf = v + r*a;
 vr = v - r*b;

 %Lateral force
 Fyf = -1*nonlintire(alphaf,Fzf,vf);
 Fyr = -1*nonlintire(alphar,Fzr,vr);

 %State derivatives
 rdot = (-Fyr*b + Fyf*a)/Iz;
 betadot = (Fyf+Fyr)/m/v - r;
 xcgdot = v*cos(beta + psi);
 ycgdot = v*sin(beta + psi);
 psidot = r;

 xdot1 = [betadot;...
 rdot;...
 xcgdot;...
 ycgdot;...
 psidot];
end
```

```
%% linear tire model with llt
function xdot=bicyclemodel_linear_llt(x,delta,v,xDot,f)
% Creating a function to calculate differential equation values with input
% ... x - 5 states of (Vehicle Slip angle, yaw velocity, longitudinal
% velocity of CG, lateral velocity of CG and yaw angle) Steering angle in
% rad, xDot and parameter for roll stiffness distribution

 m = 1637; %kg 1150
 Iz = 3326; %kg-m^2 1850
 l = 2.736; %m 2.66
 a = 0.4*l; %m 1.064
 b = 0.6*l; %m 1.596
 g=9.81; % accelereation due to gravity in m/s^2
 h=2.4*0.3048; % m height of Cg
 t=1.7; % m trackwidth
 roll_parameters =[0.6 0.3]; % roll stiffness distribution values
 kf=roll_parameters(f); % front roll stiffness distribution in percentage
 kr=1-kf; % rear roll stiffness distribution in percentage
 % Static loads
 Wf= m*g*b/l; % Static load on front axle in N
 Wr=m*g*a/l; % Static load on rear axle in N
 % acceleration in lateral direction
 ay=v*(xDot(1)+x(2)); % Taking lateral acceleration from betadot and yaw velocity value
 % Dynamic loads

 Fzfl=Wf/2-(m*ay*h/t)*kf; % Dynamic load on front left in N
 Fzfr=Wf/2+(m*ay*h/t)*kf; % Dynamic load on front right in N
 Fzrl=Wr/2-(m*ay*h/t)*kr; % Dynamic load on rear left in N
 Fzrr=Wr/2+(m*ay*h/t)*kr; % Dynamic load on rear right in N

 % cornering stifnesses
 cfl=cornering_stifness(Fzfl); % Cornering stiffness of front left in N/rad
 cfr=cornering_stifness(Fzfr); % Cornering stiffness of front left in N/rad
```

```
 crl=cornering_stifness(Fzrl); % Cornering stiffness of front left in N/rad
 crr=cornering_stifness(Fzrr); % Cornering stiffness of front left in N/rad

 cf=cfl+cfr; % cornering stiffness of front axle;

 cr= crl+crr; % Cornering stiffness of rear axle;
 % bicycle model parameters
 a11 = -((cr + cf)/(m*v));
 a12 = ((cr*b - cf*a)/(m*v^2))-1;
 a21 = (cr*b - cf*a)/Iz;
 a22 = -(cr*(b^2) + cf*(a^2))/(Iz*v);
 b11 = cf/(m*v);
 b21 = (cf*a)/Iz;
 beta=x(1);
 r=x(2);
%  xcg=x(3);
%  ycg=x(4);
 psi=x(5);
 betadot = a11*beta + a12*r + b11*delta;
 rdot = a21*beta + a22*r + b21*delta;
 xcgdot = v*cos(beta + psi);
 ycgdot = v*sin(beta + psi);
 psidot = r;

 xdot = [betadot;...
 rdot;
 xcgdot;
 ycgdot;
 psidot];
end



function xdot=bicyclemodel_linear(x,delta,v)
 m = 1637; %kg 1150
 Iz = 3326; %kg-m^2 1850
 l = 2.736; %m 2.66
 a = 0.4*l; %m 1.064
 b = 0.6*l; %m 1.596
 cf = 2*1500*180/pi; %N/rad 2*1312*180/pi
 cr = 2*1146*180/pi; %N/rad 2*984*180/pi
 a11 = -((cr + cf)/(m*v));
 a12 = ((cr*b - cf*a)/(m*v^2))-1;
 a21 = (cr*b - cf*a)/Iz;
 a22 = -(cr*(b^2) + cf*(a^2))/(Iz*v);
 b11 = cf/(m*v);
 b21 = (cf*a)/Iz;
 beta=x(1);
 r=x(2);
%  xcg=x(3);
%  ycg=x(4);
 psi=x(5);
 betadot = a11*beta + a12*r + b11*delta;
 rdot = a21*beta + a22*r + b21*delta;
 xcgdot = v*cos(beta + psi);
 ycgdot = v*sin(beta + psi);
 psidot = r;

 xdot = [betadot;...
 rdot;
 xcgdot;
```

```
 ycgdot;
 psidot];
end

function CS = cornering_stifness(Fz)

%% Cornering Stiffness Parameters
% Uncomment these to find the a and b values
% N1=[2200 2600 3000 3400 3800 4200 4600 5000 5400 5800 6200 6600 7000]; % Normal load in N
% CS1=[848 972 1088 1195 1292 1381 1461 1531 1593 1645 1689 1723 1748]; % Cornering stiffness
in N/deg
% q1=fittype(@ (a,b,c,x) a*x.^2+b*x+c, 'problem','c');
% f1 = fit( N1', CS1', q1, 'problem', 0 );
% a1=f1.b; % coefficient of Fz
% b1=f1.a; % coefficeint of Fz^2

a1=0.4474; % coefficient of Fz
b1=-2.8236e-05; % coefficeint of Fz^2

CS1=a1*Fz+b1*Fz^2;
CS=CS1*180/pi;

end
```