ME-8930 | LMIs for Robust & Optimal Control Design

Capstone Project [Full Scale Vehicle - Generic]

Team: Chinmay Samak and Tanmay Samak

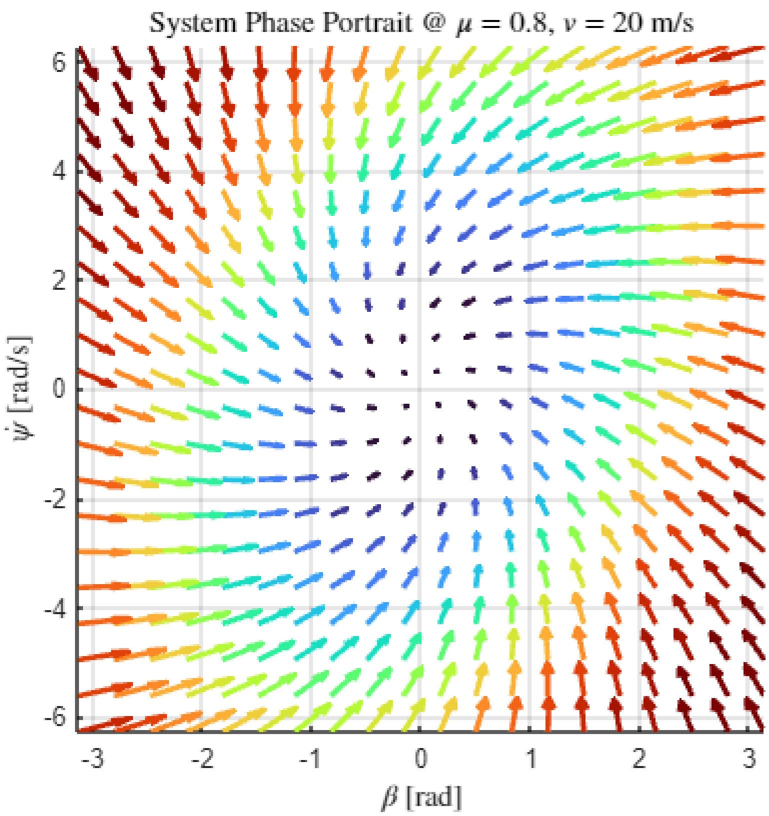
```
clear;
close;
clc;
```

System Modeling & Analysis

```
% System parameters

m = 1630; % Mass [kg]
J = 3853; % Yaw moment of inertia [kg-m^2]
l_f = 1.2313; % Distance from CG to front axel [m]
l_r = 1.4087; % Distance from CG to rear axel [m]
l = l_f + l_r; % Wheelbase [m]
C_fl = 50000; % Cornering stiffness of FL tire [N/rad]
C_fr = 50000; % Cornering stiffness of FR tire [N/rad]
C_rl = 50000; % Cornering stiffness of RL tire [N/rad]
C_rr = 50000; % Cornering stiffness of RR tire [N/rad]
```

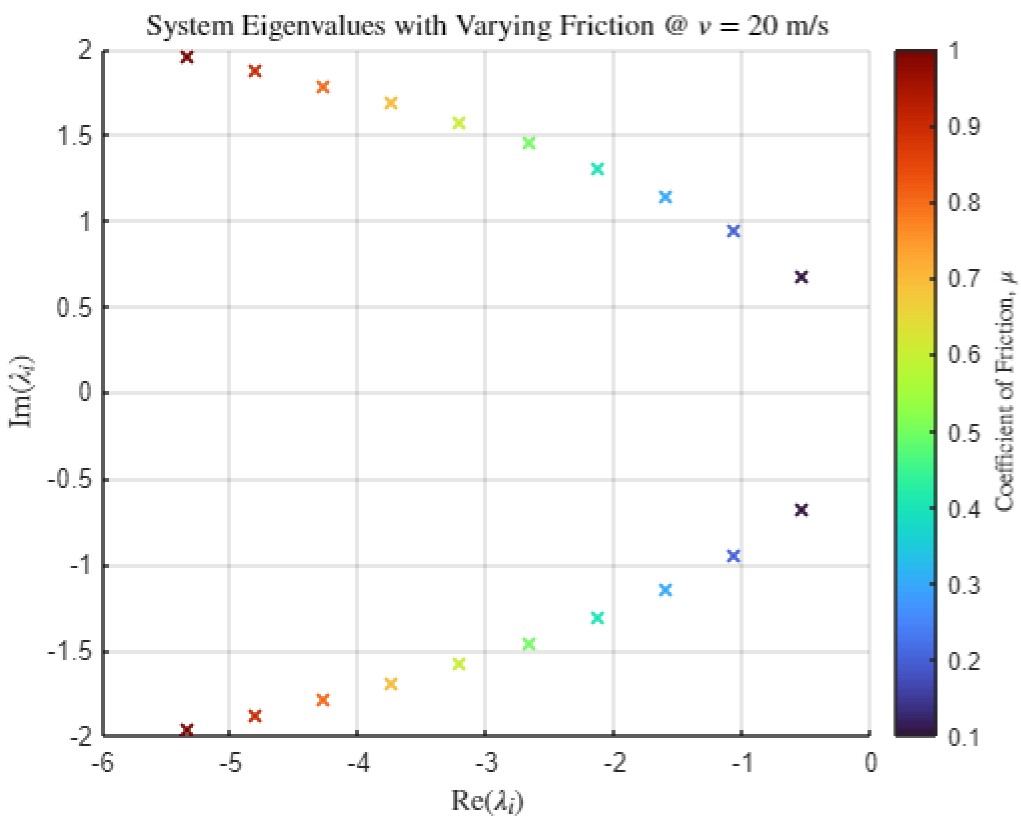
```
% System phase portrait

u_fl = 0.8; % Coefficient of friction for ground and FL tire interconnect
u_fr = 0.8; % Coefficient of friction for ground and FR tire interconnect
u_rl = 0.8; % Coefficient of friction for ground and RL tire interconnect
u_rr = 0.8; % Coefficient of friction for ground and RR tire interconnect
v = 20; % Vehicle velocity [m/s]

% Compute system matrix
a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr))) - 1;
a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) + l_r^2*(u_rl*C_rl+u_rr*C_rr));
A = [a11 a12;
     a21 a22];

% Define a range of initial conditions
X1_range = linspace(-3.14159, 3.14159, 20);
X2_range = linspace(-6.28319, 6.28319, 20);

% Create a meshgrid of initial conditions
[X1, X2] = meshgrid(X1_range, X2_range);

% Initialize vectors to store the derivatives
```

```
X1_dot = zeros(size(X1));
X2_dot = zeros(size(X2));

% Calculate the derivatives for each initial condition
for i = 1:numel(X1)
    X_dot = A * [X1(i); X2(i)];
    X1_dot(i) = X_dot(1);
    X2_dot(i) = X_dot(2);
end
Z = (sqrt(X1_dot.^2 + X2_dot.^2));
Z = Z/max(Z,[],'all');
Z = reshape(Z.',1,[]);
[sortedZ, sortedIdx] = sort(Z);
sortedIdx = reshape(sortedIdx,[20,20]).';
X1_s = X1(sortedIdx);
X2_s = X2(sortedIdx);
X1_dot_s = X1_dot(sortedIdx);
X2_dot_s = X2_dot(sortedIdx);

% Plot
fig = figure();
fig.Position(3:4) = [560, 420];
cmap = turbo(size(X1_s,1));
hold on;
for i = 1:size(X1_s,1)
    quiver(X1_s(i,:),X2_s(i,:),X1_dot_s(i,:),X2_dot_s(i,:), .2, 'Color', cmap(i,:),
'LineWidth',2)
end
hold off;
xlabel('X1');
ylabel('X2');
xlabel('$\beta$ [rad]','Interpreter','latex');
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex');
title('System Phase Portrait @ $\mu = 0.8$, $v = 20$ m/s','Interpreter','latex');
grid on;
axis tight;
pbaspect([1 1 1])
```



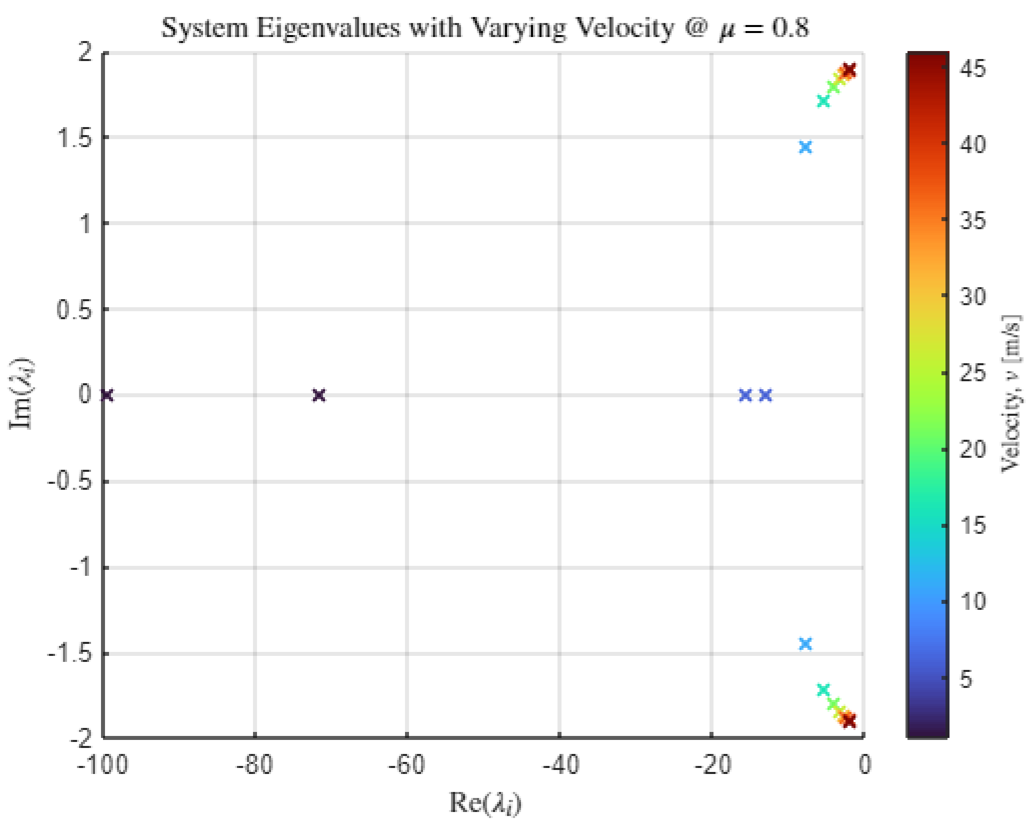

```
% System eignvalues with varying friction

EV = []; % Array to store system eigenvalues
v = 20; % Vehicle velocity [m/s]
for u=0.1:0.1:1
    % Set friction value
    u_fl = u;
    u_fr = u;
    u_rl = u;
    u_rr = u;
    % Update system matrix
    a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
    a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr))) -
1;
    a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
    a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) + l_r^2*(u_rl*C_rl+u_rr*C_rr));
    A = [a11 a12;
         a21 a22];
    % Compute eigenvalues
    EV(:,end+1) = eig(A);
end

% Plot
u=0.1:0.1:1;
fig = figure();
fig.Position(3:4) = [560, 420];
```

```
scatter(real(EV(1,:)),imag(EV(1,:)),[],u,'LineWidth',1.5,'Marker','x')
hold on;
scatter(real(EV(2,:)),imag(EV(2,:)),[],u,'LineWidth',1.5,'Marker','x')
cb = colorbar();
ylabel(cb,'Coefficient of Friction, $\mu$','Interpreter','latex');
colormap turbo;
hold off;
xlabel('Re$(\lambda_i)$','Interpreter','latex');
ylabel('Im$(\lambda_i)$','Interpreter','latex');
title('System Eigenvalues with Varying Friction @ $v = 20$ m/
s','Interpreter','latex');
grid on;
```



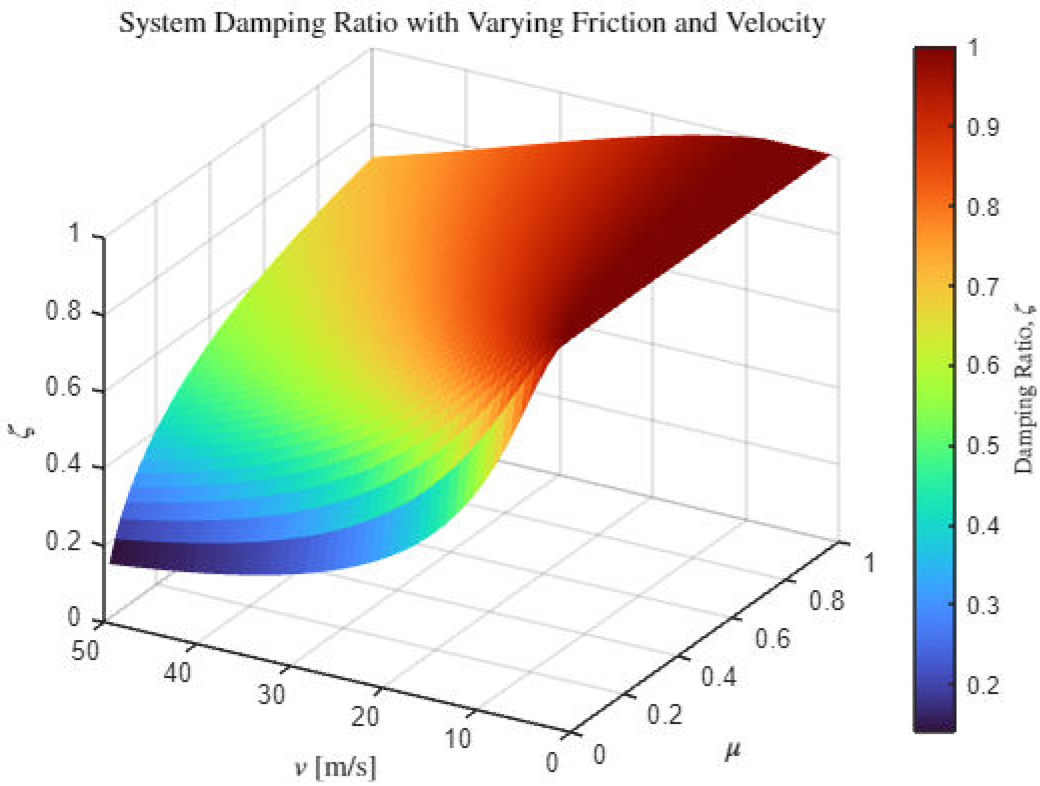

```
% System eignvalues with varying velocity

EV = []; % Array to store system eigenvalues
u = 0.8; % Coefficient of friction for ground and tire interconnect
for v=1:5:50
    % Set friction value
    u_fl = u;
    u_fr = u;
    u_rl = u;
    u_rr = u;
    % Update system matrix
    a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
    a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr))) -
1;
```

```
    a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
    a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) + l_r^2*(u_rl*C_rl+u_rr*C_rr));
    A = [a11 a12;
         a21 a22];
    % Compute eigenvalues
    EV(:,end+1) = eig(A);
end

% Plot
v=1:5:50;
fig = figure();
fig.Position(3:4) = [560, 420];
scatter(real(EV(1,:)),imag(EV(1,:)),[],v,'LineWidth',1.5,'Marker','x')
hold on;
scatter(real(EV(2,:)),imag(EV(2,:)),[],v,'LineWidth',1.5,'Marker','x')
cb = colorbar();
ylabel(cb,'Velocity, $v$ [m/s]','Interpreter','latex');
colormap turbo;
hold off;
xlabel('Re$(\lambda_i)$','Interpreter','latex');
ylabel('Im$(\lambda_i)$','Interpreter','latex');
title('System Eigenvalues with Varying Velocity @ $\mu =
0.8$','Interpreter','latex');
grid on;
```




```
% System damping ratio with varying friction and velocity
```

```
zeta = []; % Array to store system eigenvalues
for u=0.02:0.02:1
    for v=1:1:50
        % Set friction value
        u_fl = u;
        u_fr = u;
        u_rl = u;
        u_rr = u;
        % Update system matrix
        a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
        a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) -
l_f*(u_fl*C_fl+u_fr*C_fr))) - 1;
        a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
        a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) +
l_r^2*(u_rl*C_rl+u_rr*C_rr));
        A = [a11 a12;
             a21 a22];
        % Compute damping ratio
        EV = eig(A);
        zeta(:,end+1) = -real(EV)./abs(EV);
    end
end

% Plot
u=0.02:0.02:1;
v=1:1:50;
[X,Y] = meshgrid(u,v);
Z = reshape(zeta(1, :), 50, 50);
fig = figure();
fig.Position(3:4) = [560, 420];
s = surf(X,Y,Z,'FaceAlpha',1.0,'EdgeColor','none');
hold on;
cb = colorbar();
ylabel(cb,'Damping Ratio, $\zeta$','Interpreter','latex');
colormap turbo;
hold off;
xlabel('$\mu$','Interpreter','latex');
ylabel('$v$ [m/s]','Interpreter','latex');
zlabel('$\zeta$','Interpreter','latex');
title('System Damping Ratio with Varying Friction and
Velocity','Interpreter','latex');
grid on;
view([-60 30]);
```

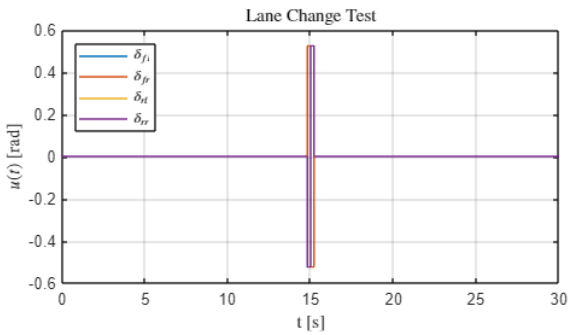


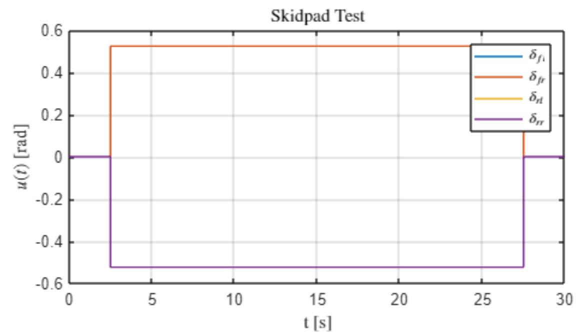
Model Simulation for Benckmark Maneuvers with Varying Friction

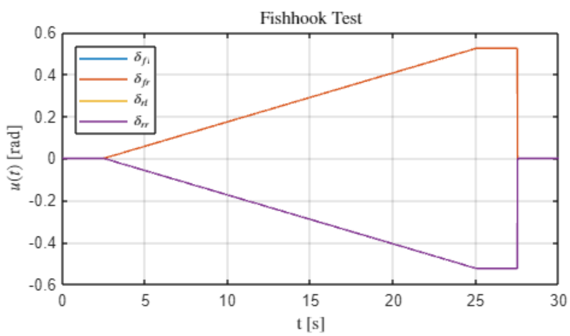
```
% Maneuver parameters

% Lane change test (impulse input)
u1_lane_change = [ones(1,14800)*(deg2rad(0)) ones(1,200)*(deg2rad(30))
ones(1,200)*(deg2rad(-30)) ones(1,14800)*(deg2rad(0))];
u2_lane_change = [ones(1,14800)*(deg2rad(0)) ones(1,200)*(deg2rad(30))
ones(1,200)*(deg2rad(-30)) ones(1,14800)*(deg2rad(0))];
u3_lane_change = [ones(1,14800)*(deg2rad(0)) ones(1,200)*(deg2rad(-30))
ones(1,200)*(deg2rad(30)) ones(1,14800)*(deg2rad(0))];
u4_lane_change = [ones(1,14800)*(deg2rad(0)) ones(1,200)*(deg2rad(-30))
ones(1,200)*(deg2rad(30)) ones(1,14800)*(deg2rad(0))];
u_lane_change = [u1_lane_change; u2_lane_change; u3_lane_change; u4_lane_change];
w_lane_change = zeros(1,30000);

% Skidpad test (step input)
u1_skidpad = [ones(1,2500)*(deg2rad(0)) ones(1,25000)*(deg2rad(30))
ones(1,2500)*(deg2rad(0))];
u2_skidpad = [ones(1,2500)*(deg2rad(0)) ones(1,25000)*(deg2rad(30))
ones(1,2500)*(deg2rad(0))];
u3_skidpad = [ones(1,2500)*(deg2rad(0)) ones(1,25000)*(deg2rad(-30))
ones(1,2500)*(deg2rad(0))];
u4_skidpad = [ones(1,2500)*(deg2rad(0)) ones(1,25000)*(deg2rad(-30))
ones(1,2500)*(deg2rad(0))];
```

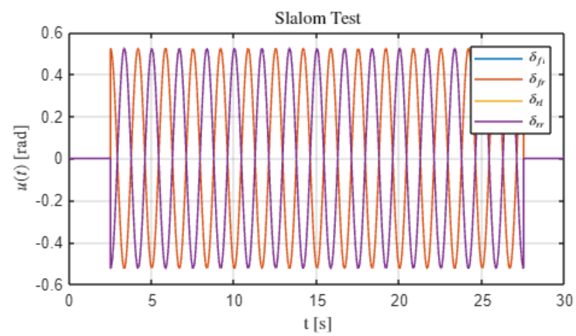
```
u_skidpad = [u1_skidpad; u2_skidpad; u3_skidpad; u4_skidpad];
w_skidpad = zeros(1,30000);

% Fishhook test (ramp input)
angles = deg2rad(linspace(0,30,22500));
u1_fishhook = [ones(1,2500)*(deg2rad(0)) angles ones(1,2500)*deg2rad(30)
ones(1,2500)*(deg2rad(0))];
u2_fishhook = [ones(1,2500)*(deg2rad(0)) angles ones(1,2500)*deg2rad(30)
ones(1,2500)*(deg2rad(0))];
u3_fishhook = [ones(1,2500)*(deg2rad(0)) -angles ones(1,2500)*deg2rad(-30)
ones(1,2500)*(deg2rad(0))];
u4_fishhook = [ones(1,2500)*(deg2rad(0)) -angles ones(1,2500)*deg2rad(-30)
ones(1,2500)*(deg2rad(0))];
u_fishhook = [u1_fishhook; u2_fishhook; u3_fishhook; u4_fishhook];
w_fishhook = zeros(1,30000);

% Slalom test (sine input)
n_cones = 30;
angles = linspace(0,n_cones*pi,25000);
u1_slalom = [ones(1,2500)*(deg2rad(0)) deg2rad(30*cos(angles))
ones(1,2500)*(deg2rad(0))];
u2_slalom = [ones(1,2500)*(deg2rad(0)) deg2rad(30*cos(angles))
ones(1,2500)*(deg2rad(0))];
u3_slalom = [ones(1,2500)*(deg2rad(0)) deg2rad(-30*cos(angles))
ones(1,2500)*(deg2rad(0))];
u4_slalom = [ones(1,2500)*(deg2rad(0)) deg2rad(-30*cos(angles))
ones(1,2500)*(deg2rad(0))];
u_slalom = [u1_slalom; u2_slalom; u3_slalom; u4_slalom];
w_slalom = zeros(1,30000);

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [4000, 2000];
% Lane change (impulse)
subplot(2,2,1)
plot(t/1000,u1_lane_change)
hold on;
plot(t/1000,u2_lane_change)
plot(t/1000,u3_lane_change)
plot(t/1000,u4_lane_change)
legend('$\delta_{fl}$','$\delta_{fr}$','$\delta_{rl}$','$\delta_{rr}
$','Interpreter','latex','Location','northwest')
xlabel('t [s]','Interpreter','latex')
ylabel('$u(t)$ [rad]','Interpreter','latex')
title('Lane Change Test','Interpreter','latex')
grid on;
hold off;
% Skidpad (step)
subplot(2,2,2)
```

```
plot(t/1000,u1_skidpad)
hold on;
plot(t/1000,u2_skidpad)
plot(t/1000,u3_skidpad)
plot(t/1000,u4_skidpad)
legend('$\delta_{fl}$','$\delta_{fr}$','$\delta_{rl}$','$\delta_{rr}
$','Interpreter','latex','Location','northeast')
xlabel('t [s]','Interpreter','latex')
ylabel('$u(t)$ [rad]','Interpreter','latex')
title('Skidpad Test','Interpreter','latex')
grid on;
hold off;
% Fishhook (ramp)
subplot(2,2,3)
plot(t/1000,u1_fishhook)
hold on;
plot(t/1000,u2_fishhook)
plot(t/1000,u3_fishhook)
plot(t/1000,u4_fishhook)
legend('$\delta_{fl}$','$\delta_{fr}$','$\delta_{rl}$','$\delta_{rr}
$','Interpreter','latex','Location','northwest')
xlabel('t [s]','Interpreter','latex')
ylabel('$u(t)$ [rad]','Interpreter','latex')
title('Fishhook Test','Interpreter','latex')
grid on;
hold off;
% Slalom (sine)
subplot(2,2,4)
plot(t/1000,u1_slalom)
hold on;
plot(t/1000,u2_slalom)
plot(t/1000,u3_slalom)
plot(t/1000,u4_slalom)
legend('$\delta_{fl}$','$\delta_{fr}$','$\delta_{rl}$','$\delta_{rr}
$','Interpreter','latex','Location','northeast')
xlabel('t [s]','Interpreter','latex')
ylabel('$u(t)$ [rad]','Interpreter','latex')
title('Slalom Test','Interpreter','latex')
grid on;
hold off;
```







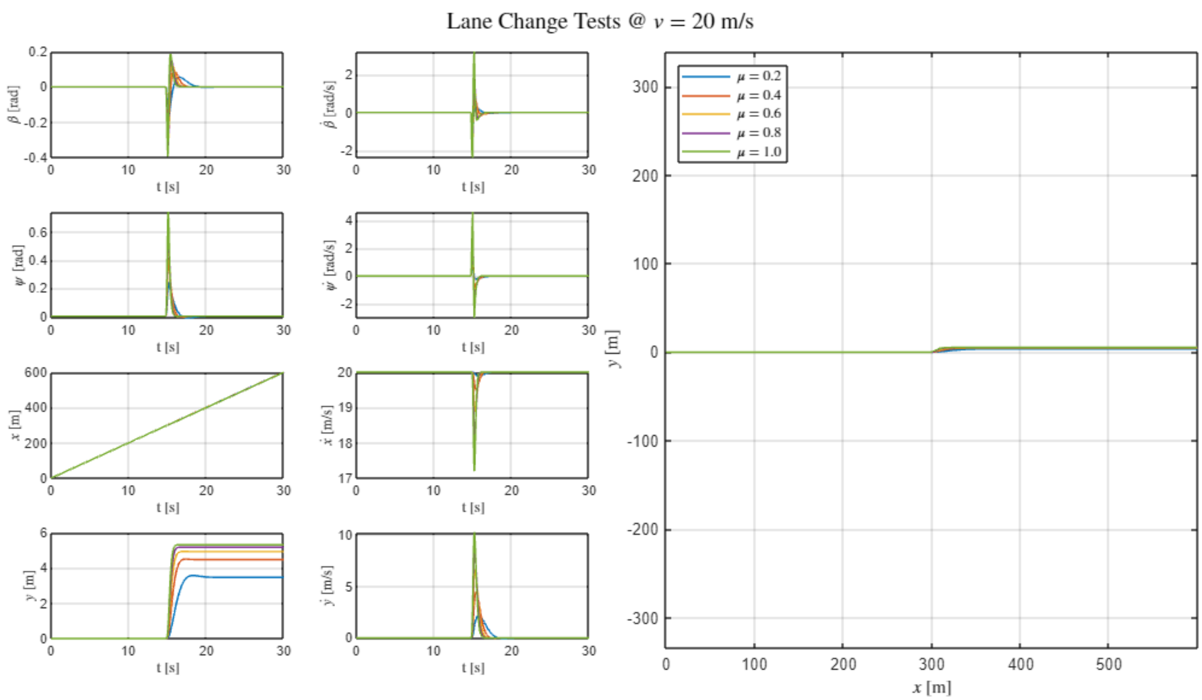



```
% Simulate lane change tests with varying friction

BETA_LC = [];
BETA_DOT_LC = [];
PSI_LC = [];
PSI_DOT_LC = [];
PX_LC = [];
VX_LC = [];
PY_LC = [];
VY_LC = [];

for u=0.2:0.2:1

    % Initialization
    dt = 0.001; % Simulation timestep [s]
    v = 20; % Vehicle velocity [m/s]
    X = [];
    X(:,1) = [0;0];
    X_dot = [];
    Psi = [];
    Psi(1) = 0;
    Beta = [];
    Vx = [];
    Vy = [];
    Px = [];
    Px(1) = 0;
    Py = [];
    Py(1) = 0;
```

```
    % Set friction value
    u_fl = u;
    u_fr = u;
    u_rl = u;
    u_rr = u;

    % Update system matrix
    a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
    a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr))) -
1;
    a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
    a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) + l_r^2*(u_rl*C_rl+u_rr*C_rr));
    A = [a11 a12;
         a21 a22];

    % Update control input matrix
    b11 = (u_fl*C_fl)/(m*v);
    b12 = (u_fr*C_fr)/(m*v);
    b13 = (u_rl*C_rl)/(m*v);
    b14 = (u_rr*C_rr)/(m*v);
    b21 = (l_f*u_fl*C_fl)/J;
    b22 = (l_f*u_fr*C_fr)/J;
    b23 = -(l_r*u_rl*C_rl)/J;
    b24 = -(l_r*u_rr*C_rr)/J;
    B = [b11 b12 b13 b14;
         b21 b22 b23 b24];

    % Update disturbance input matrix
    B_d = [1/(m*v);
           (l_f-l_r)/2];

    % Simulation at each millisecond
    for t=1:30000
        X_dot(:,t) = A*X(:,t) + B*u_lane_change(:,t) +
B_d*(w_lane_change(:,t).*cos(Psi(t)));
        X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
        Beta(t) = X(1,t);
        Psi(t+1) = Psi(t) + X(2,t)*dt;
        Vx(t) = v*cos(Psi(t) + Beta(t));
        Vy(t) = v*sin(Psi(t) + Beta(t));
        Px(t+1) = Px(t) + Vx(t)*dt;
        Py(t+1) = Py(t) + Vy(t)*dt;
    end
    BETA_LC(end+1,:) = Beta;
    BETA_DOT_LC(end+1,:) = X_dot(1,:);
    PSI_LC(end+1,:) = Psi;
    PSI_DOT_LC(end+1,:) = X(2,:);
    PX_LC(end+1,:) = Px;
    VX_LC(end+1,:) = Vx;
```

```
    PY_LC(end+1,:) = Py;
    VY_LC(end+1,:) = Vy;
end

% Plot
t=1:30000;
u=0.2:0.2:1;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_LC(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_LC(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_LC(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
plot(t/1000,PSI_DOT_LC(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_LC(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_LC(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_LC(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_LC(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
```

```
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_LC(:,1:end-1)',PY_LC(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
sgtitle('Lane Change Tests @ $v = 20$ m/s','Interpreter','latex')
legend('$\mu = 0.2$','$\mu = 0.4$','$\mu = 0.6$','$\mu = 0.8$','$\mu = 1.0$', ...
       'Interpreter','latex','Location','northwest')
```

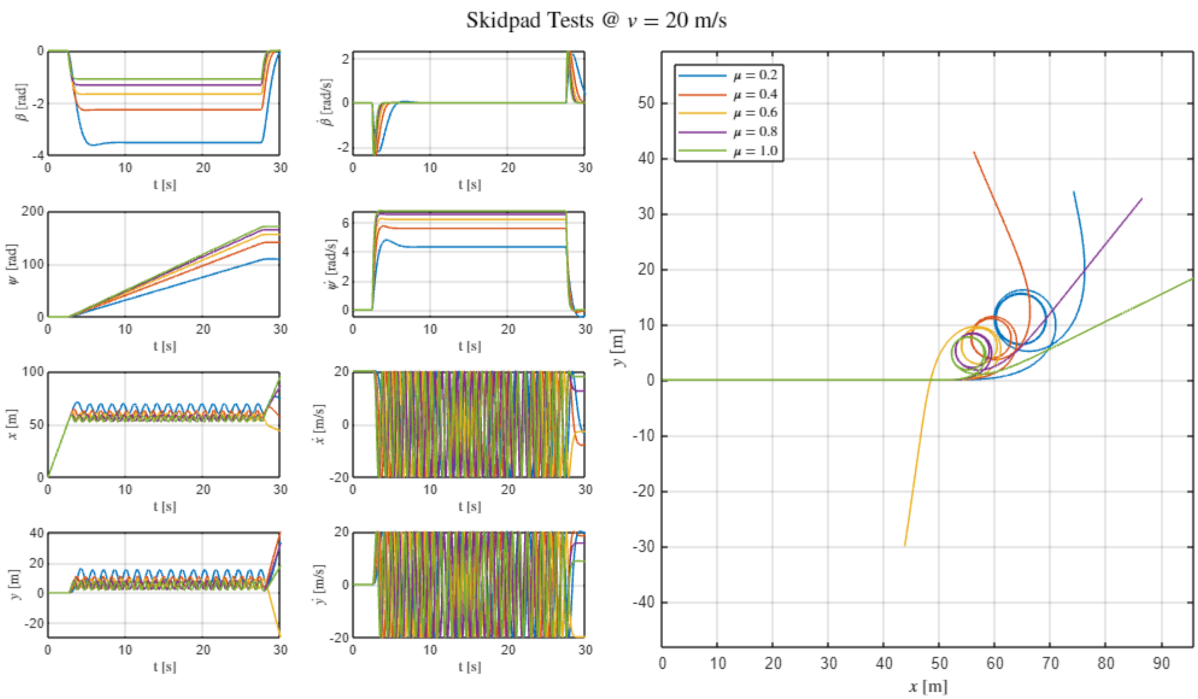



```
% Simulate skidpad tests with varying friction

BETA_SP = [];
BETA_DOT_SP = [];
PSI_SP = [];
PSI_DOT_SP = [];
PX_SP = [];
VX_SP = [];
PY_SP = [];
VY_SP = [];

for u=0.2:0.2:1

    % Initialization
    dt = 0.001; % Simulation timestep [s]
    v = 20; % Vehicle velocity [m/s]
    X = [];
```

```
    X(:,1) = [0;0];
    X_dot = [];
    Psi = [];
    Psi(1) = 0;
    Beta = [];
    Vx = [];
    Vy = [];
    Px = [];
    Px(1) = 0;
    Py = [];
    Py(1) = 0;

    % Set friction value
    u_fl = u;
    u_fr = u;
    u_rl = u;
    u_rr = u;

    % Update system matrix
    a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
    a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr))) -
1;
    a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
    a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) + l_r^2*(u_rl*C_rl+u_rr*C_rr));
    A = [a11 a12;
         a21 a22];

    % Update control input matrix
    b11 = (u_fl*C_fl)/(m*v);
    b12 = (u_fr*C_fr)/(m*v);
    b13 = (u_rl*C_rl)/(m*v);
    b14 = (u_rr*C_rr)/(m*v);
    b21 = (l_f*u_fl*C_fl)/J;
    b22 = (l_f*u_fr*C_fr)/J;
    b23 = -(l_r*u_rl*C_rl)/J;
    b24 = -(l_r*u_rr*C_rr)/J;
    B = [b11 b12 b13 b14;
         b21 b22 b23 b24];

    % Update disturbance input matrix
    B_d = [1/(m*v);
           (l_f-l_r)/2];

    % Simulation at each millisecond
    for t=1:30000
        X_dot(:,t) = A*X(:,t) + B*u_skidpad(:,t) +
B_d*(w_skidpad(:,t).*cos(Psi(t)));
        X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
        Beta(t) = X(1,t);
        Psi(t+1) = Psi(t) + X(2,t)*dt;
```

```
        Vx(t) = v*cos(Psi(t) + Beta(t));
        Vy(t) = v*sin(Psi(t) + Beta(t));
        Px(t+1) = Px(t) + Vx(t)*dt;
        Py(t+1) = Py(t) + Vy(t)*dt;
    end
    BETA_SP(end+1,:) = Beta;
    BETA_DOT_SP(end+1,:) = X_dot(1,:);
    PSI_SP(end+1,:) = Psi;
    PSI_DOT_SP(end+1,:) = X(2,:);
    PX_SP(end+1,:) = Px;
    VX_SP(end+1,:) = Vx;
    PY_SP(end+1,:) = Py;
    VY_SP(end+1,:) = Vy;
end

% Plot
t=1:30000;
u=0.2:0.2:1;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_SP(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_SP(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_SP(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
plot(t/1000,PSI_DOT_SP(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_SP(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_SP(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
```

```
grid('on')
subplot(4,4,13)
plot(t/1000,PY_SP(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_SP(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_SP(:,1:end-1)',PY_SP(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
sgtitle('Skidpad Tests @ $v = 20$ m/s','Interpreter','latex')
legend('$\mu = 0.2$','$\mu = 0.4$','$\mu = 0.6$','$\mu = 0.8$','$\mu = 1.0$', ...
       'Interpreter','latex','Location','northwest')
```

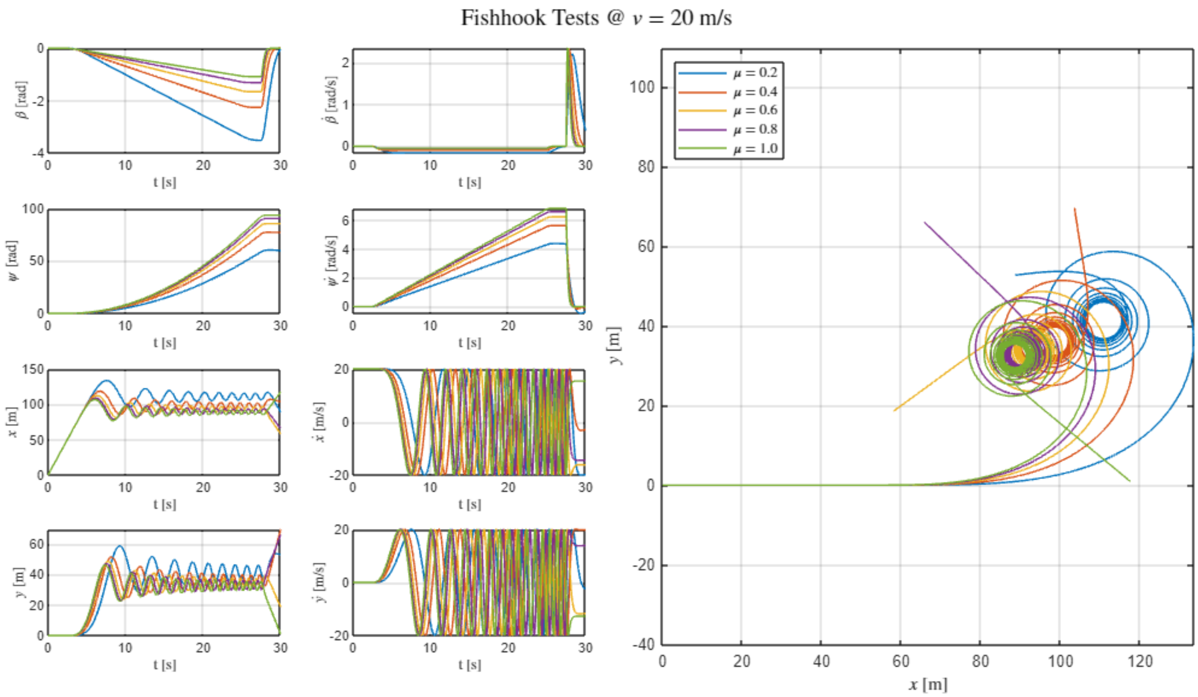



```
% Simulate fishhook tests with varying friction

BETA_FH = [];
BETA_DOT_FH = [];
PSI_FH = [];
PSI_DOT_FH = [];
```

```
PX_FH = [];
VX_FH = [];
PY_FH = [];
VY_FH = [];

for u=0.2:0.2:1

    % Initialization
    dt = 0.001; % Simulation timestep [s]
    v = 20; % Vehicle velocity [m/s]
    X = [];
    X(:,1) = [0;0];
    X_dot = [];
    Psi = [];
    Psi(1) = 0;
    Beta = [];
    Vx = [];
    Vy = [];
    Px = [];
    Px(1) = 0;
    Py = [];
    Py(1) = 0;

    % Set friction value
    u_fl = u;
    u_fr = u;
    u_rl = u;
    u_rr = u;

    % Update system matrix
    a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
    a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr))) -
1;
    a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
    a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) + l_r^2*(u_rl*C_rl+u_rr*C_rr));
    A = [a11 a12;
         a21 a22];

    % Update control input matrix
    b11 = (u_fl*C_fl)/(m*v);
    b12 = (u_fr*C_fr)/(m*v);
    b13 = (u_rl*C_rl)/(m*v);
    b14 = (u_rr*C_rr)/(m*v);
    b21 = (l_f*u_fl*C_fl)/J;
    b22 = (l_f*u_fr*C_fr)/J;
    b23 = -(l_r*u_rl*C_rl)/J;
    b24 = -(l_r*u_rr*C_rr)/J;
    B = [b11 b12 b13 b14;
         b21 b22 b23 b24];
```

```
    % Update disturbance input matrix
    B_d = [1/(m*v);
           (l_f-l_r)/2];

    % Simulation at each millisecond
    for t=1:30000
        X_dot(:,t) = A*X(:,t) + B*u_fishhook(:,t) +
B_d*(w_fishhook(:,t).*cos(Psi(t)));
        X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
        Beta(t) = X(1,t);
        Psi(t+1) = Psi(t) + X(2,t)*dt;
        Vx(t) = v*cos(Psi(t) + Beta(t));
        Vy(t) = v*sin(Psi(t) + Beta(t));
        Px(t+1) = Px(t) + Vx(t)*dt;
        Py(t+1) = Py(t) + Vy(t)*dt;
    end
    BETA_FH(end+1,:) = Beta;
    BETA_DOT_FH(end+1,:) = X_dot(1,:);
    PSI_FH(end+1,:) = Psi;
    PSI_DOT_FH(end+1,:) = X(2,:);
    PX_FH(end+1,:) = Px;
    VX_FH(end+1,:) = Vx;
    PY_FH(end+1,:) = Py;
    VY_FH(end+1,:) = Vy;
end

% Plot
t=1:30000;
u=0.2:0.2:1;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_FH(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_FH(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_FH(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
plot(t/1000,PSI_DOT_FH(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
```

```
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_FH(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_FH(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_FH(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_FH(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_FH(:,1:end-1)',PY_FH(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
sgtitle('Fishhook Tests @ $v = 20$ m/s','Interpreter','latex')
legend('$\mu = 0.2$','$\mu = 0.4$','$\mu = 0.6$','$\mu = 0.8$','$\mu = 1.0$', ...
       'Interpreter','latex','Location','northwest')
```

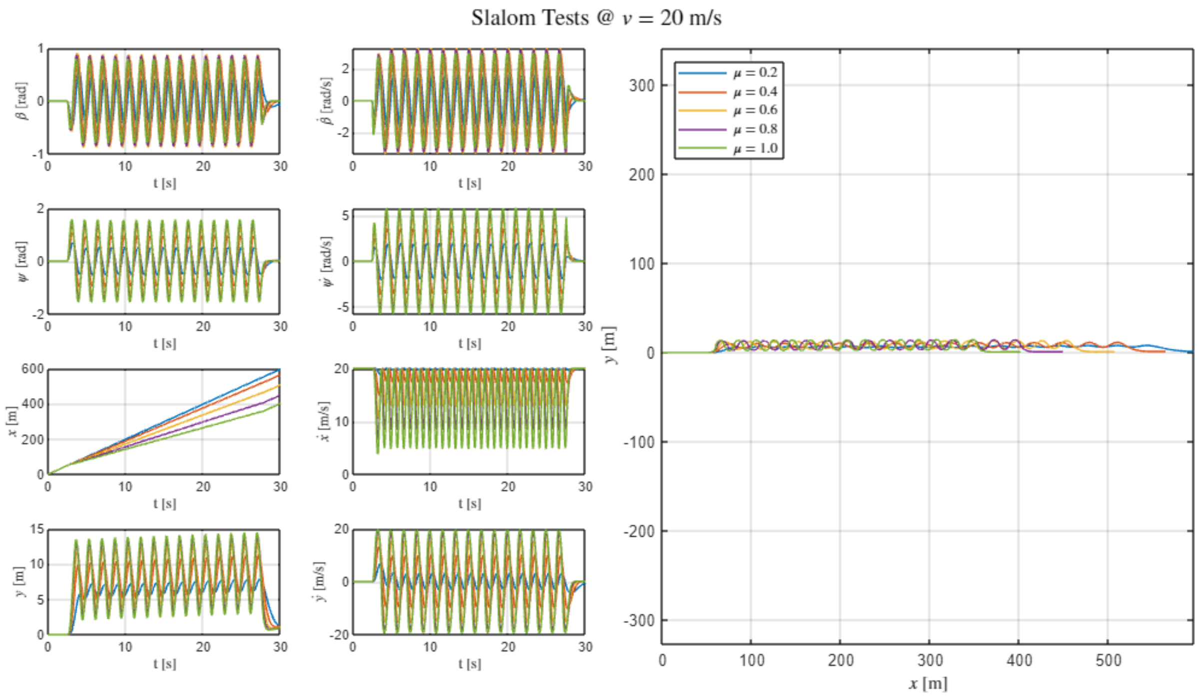



```
% Simulate slalom tests with varying friction

BETA_SL = [];
BETA_DOT_SL = [];
PSI_SL = [];
PSI_DOT_SL = [];
PX_SL = [];
VX_SL = [];
PY_SL = [];
VY_SL = [];

for u=0.2:0.2:1

    % Initialization
    dt = 0.001; % Simulation timestep [s]
    v = 20; % Vehicle velocity [m/s]
    X = [];
    X(:,1) = [0;0];
    X_dot = [];
    Psi = [];
    Psi(1) = 0;
    Beta = [];
    Vx = [];
    Vy = [];
    Px = [];
    Px(1) = 0;
    Py = [];
    Py(1) = 0;
```

```
    % Set friction value
    u_fl = u;
    u_fr = u;
    u_rl = u;
    u_rr = u;

    % Update system matrix
    a11 = -(1/(m*v)) * (u_fl*C_fl+u_fr*C_fr+u_rl*C_rl+u_rr*C_rr);
    a12 = ((1/(m*v^2)) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr))) -
1;
    a21 = (1/J) * (l_r*(u_rl*C_rl+u_rr*C_rr) - l_f*(u_fl*C_fl+u_fr*C_fr));
    a22 = -(1/(J*v)) * (l_f^2*(u_fl*C_fl+u_fr*C_fr) + l_r^2*(u_rl*C_rl+u_rr*C_rr));
    A = [a11 a12;
         a21 a22];

    % Update control input matrix
    b11 = (u_fl*C_fl)/(m*v);
    b12 = (u_fr*C_fr)/(m*v);
    b13 = (u_rl*C_rl)/(m*v);
    b14 = (u_rr*C_rr)/(m*v);
    b21 = (l_f*u_fl*C_fl)/J;
    b22 = (l_f*u_fr*C_fr)/J;
    b23 = -(l_r*u_rl*C_rl)/J;
    b24 = -(l_r*u_rr*C_rr)/J;
    B = [b11 b12 b13 b14;
         b21 b22 b23 b24];

    % Update disturbance input matrix
    B_d = [1/(m*v);
           (l_f-l_r)/2];

    % Simulation at each millisecond
    for t=1:30000
        X_dot(:,t) = A*X(:,t) + B*u_slalom(:,t) + B_d*(w_slalom(:,t).*cos(Psi(t)));
        X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
        Beta(t) = X(1,t);
        Psi(t+1) = Psi(t) + X(2,t)*dt;
        Vx(t) = v*cos(Psi(t) + Beta(t));
        Vy(t) = v*sin(Psi(t) + Beta(t));
        Px(t+1) = Px(t) + Vx(t)*dt;
        Py(t+1) = Py(t) + Vy(t)*dt;
    end
    BETA_SL(end+1,:) = Beta;
    BETA_DOT_SL(end+1,:) = X_dot(1,:);
    PSI_SL(end+1,:) = Psi;
    PSI_DOT_SL(end+1,:) = X(2,:);
    PX_SL(end+1,:) = Px;
    VX_SL(end+1,:) = Vx;
    PY_SL(end+1,:) = Py;
```

```
    VY_SL(end+1,:) = Vy;
end

% Plot
t=1:30000;
u=0.2:0.2:1;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_SL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_SL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_SL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
plot(t/1000,PSI_DOT_SL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_SL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_SL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_SL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_SL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
```

```
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_SL(:,1:end-1)',PY_SL(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
sgtitle('Slalom Tests @ $v = 20$ m/s','Interpreter','latex')
legend('$\mu = 0.2$','$\mu = 0.4$','$\mu = 0.6$','$\mu = 0.8$','$\mu = 1.0$', ...
       'Interpreter','latex','Location','northwest')
```



Uncertain Parameter Dependent System

```
% Operating conditions

v = 20; % Vehicle velocity [m/s]

% Vehicle model with affine dependence on u_fl, u_fr, u_rl and u_rr

Ap0 = [0 -1;
        0  0]; % Nominal system matrix

Ap1 = [-C_fl/(m*v)    -(l_f*C_fl)/(m*v^2);
      -(l_f*C_fl)/J  -(l_f^2*C_fl)/(J*v)]; % Affine dependence on u_fl

Ap2 = [-C_fr/(m*v)    -(l_f*C_fr)/(m*v^2);
      -(l_f*C_fr)/J  -(l_f^2*C_fr)/(J*v)]; % Affine dependence on u_fr

Ap3 = [-C_rl/(m*v)     (l_r*C_rl)/(m*v^2);
```

```
      (l_r*C_rl)/J   -(l_r^2*C_rl)/(J*v)]; % Affine dependence on u_rl

Ap4 = [-C_rr/(m*v)     (l_r*C_rr)/(m*v^2);
      (l_r*C_rr)/J   -(l_r^2*C_rr)/(J*v)]; % Affine dependence on u_rr

% Ap = Ap0 + u_fl.*Ap1 + u_fr.*Ap2 + u_rl.*Ap3 + u_rr.*Ap4

Bp0 = [0 0 0 0;
       0 0 0 0]; % Nominal input matrix

Bp1 = [C_fl/(m*v)   0 0 0;
      (l_f*C_fl)/J 0 0 0]; % Affine dependence on u_fl

Bp2 = [0 C_fr/(m*v)   0 0;
       0 (l_f*C_fr)/J 0 0]; % Affine dependence on u_fr

Bp3 = [0 0 C_rl/(m*v)    0;
       0 0 -(l_r*C_rl)/J 0]; % Affine dependence on u_rl

Bp4 = [0 0 0 C_rr/(m*v);
       0 0 0 -(l_r*C_rr)/J]; % Affine dependence on u_rr

% Bp = Bp0 + u_fl.*Bp1 + u_fr.*Bp2 + u_rl.*Bp3 + u_rr.*Bp4

Dp = [1/(m*v); (l_f-l_r)/2];

Cp1 = [1 0;
       0 1;
       0 0;
       0 0;
       0 0;
       0 0];

By1 = [0 0 0 0;
       0 0 0 0;
       1 0 0 0;
       0 1 0 0;
       0 0 1 0;
       0 0 0 1];

Dy = [0;
      0;
      0;
      0;
      0;
      0];

Cp2 = [1 0;
       0 1;
       0 0;
```

```
       0 0;
       0 0;
       0 0];

By2 = [0 0 0 0;
       0 0 0 0;
       1 0 0 0;
       0 1 0 0;
       0 0 1 0;
       0 0 0 1];

O = [0;
     0;
     0;
     0;
     0;
     0];

[u_fl_lim, u_fr_lim, u_rl_lim, u_rr_lim] = deal([0.1 1], [0.1 1], [0.1 1], [0.1 1]);

pv = pvec('box', [u_fl_lim; u_fr_lim; u_rl_lim; u_rr_lim]); % Range of parameter
variation
pvinfo(pv); % Print uncertain parameter(s) information
```

```
Vector of 4 parameters ranging in a box
```

Open-Loop System

```
S_ol = psys(pv, [ltisys(Ap0, [Dp Bp0], [Cp1; Cp2], [Dy By1; O By2], 1), ...
                 ltisys(Ap1, [zeros(2,1) Bp1], 0*[Cp1; Cp2], 0*[Dy By1; O By2], 0),
...
                 ltisys(Ap2, [zeros(2,1) Bp2], 0*[Cp1; Cp2], 0*[Dy By1; O By2], 0),
...
                 ltisys(Ap3, [zeros(2,1) Bp3], 0*[Cp1; Cp2], 0*[Dy By1; O By2], 0),
...
                 ltisys(Ap4, [zeros(2,1) Bp4], 0*[Cp1; Cp2], 0*[Dy By1; O By2],
0)]);

psinfo(S_ol); % Print open-loop system information
```

```
Affine parameter-dependent model with 4 parameters (5 systems)
    Each system has 2 state(s), 5 input(s), and 12 output(s)
```

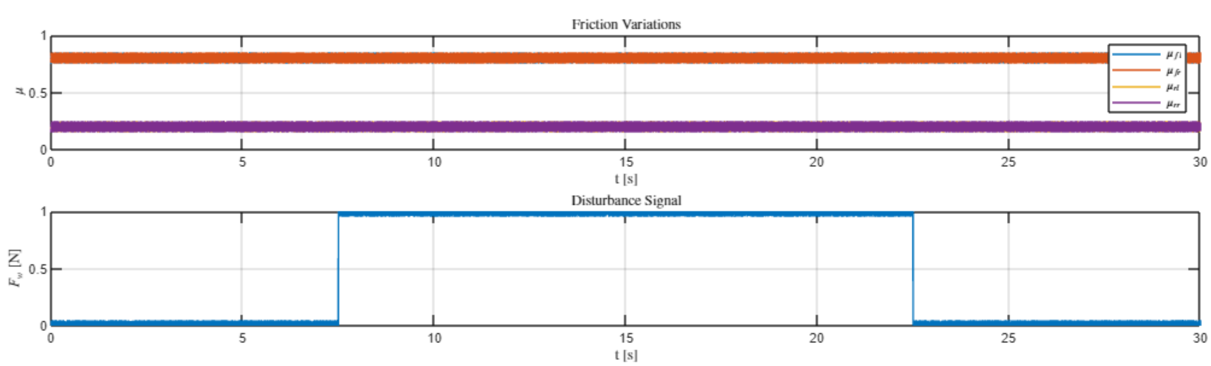
```
% Test robust stability of open-loop system via parametric Lyapunov functions
[tmin_ol, Q0_ol, Q1_ol, Q2_ol, Q3_ol] = pdlstab(S_ol, [0 0])
```

```
 Solver for LMI feasibility problems L(x) < R(x)
    This solver minimizes  t  subject to  L(x) < R(x) + t*I
    The best value of t should be negative for feasibility

 Iteration   :    Best value of t so far
```

```
     1                         0.124843
     2                         0.020427
     3                         0.020427
     4                     8.819273e-03
     5                     8.122708e-03
     6                     6.007728e-03
     7                     5.293805e-03
     8                     5.293805e-03
***                 new lower bound: -1.021966e-04
     9                     5.293805e-03
***                 new lower bound: 3.348398e-04

 Result:  best value of t: 5.293805e-03
          f-radius saturation:  0.000% of R =  1.00e+07

 These LMI constraints were found infeasible

 Trying for the transposed system...

 Solver for LMI feasibility problems L(x) < R(x)
    This solver minimizes  t  subject to  L(x) < R(x) + t*I
    The best value of t should be negative for feasibility

 Iteration   :    Best value of t so far

     1                         0.110427
     2                         0.014980
     3                         0.014980
     4                     9.701941e-03
     5                     8.725562e-03
     6                     7.205384e-03
     7                     6.100566e-03
     8                     6.100566e-03
***                 new lower bound: -2.667113e-05
     9                     6.100566e-03
***                 new lower bound: 4.287754e-04

 Result:  best value of t: 6.100566e-03
          f-radius saturation:  0.000% of R =  1.00e+07

 These LMI constraints were found infeasible

 Robust stability could not be established.
tmin_ol = 0.0061
Q0_ol = 2×2
    0.0202    0.0030
    0.0030    0.0050
Q1_ol = 2×2
   -0.0480    0.0042
    0.0042   -0.0007
Q2_ol = 2×2
   -0.0480    0.0042
    0.0042   -0.0007
Q3_ol = 2×2
    0.1282    0.0113
    0.0113    0.0016
```

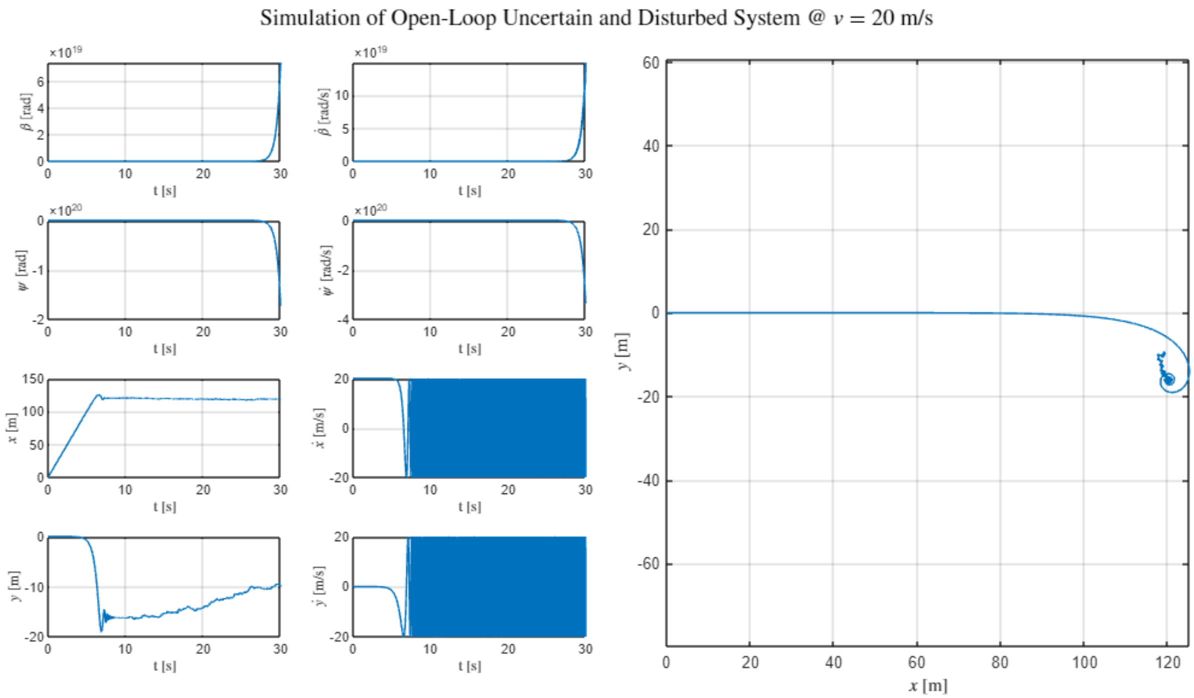
```
% Friction variation and disturbance

rng(1,"twister"); % Fix random seed and algorithm

% Friction
u1_signal = ones(1,30000)*0.8;
```

```
u2_signal = ones(1,30000)*0.8;
u3_signal = ones(1,30000)*0.2;
u4_signal = ones(1,30000)*0.2;
u1_signal = u1_signal - 0.05 + 0.1 * rand(1, length(u1_signal));
u2_signal = u2_signal - 0.05 + 0.1 * rand(1, length(u2_signal));
u3_signal = u3_signal - 0.05 + 0.1 * rand(1, length(u3_signal));
u4_signal = u4_signal - 0.05 + 0.1 * rand(1, length(u4_signal));

% Disturbance
w_signal = [zeros(1,7500) ones(1,15000) zeros(1,7500)];
w_signal = w_signal - 0.05 + 0.1*(max(w_signal)) * rand(1, length(w_signal));

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [4000, 1000];
% Friction
subplot(2,1,1)
plot(t/1000,u1_signal)
hold on;
plot(t/1000,u2_signal)
plot(t/1000,u3_signal)
plot(t/1000,u4_signal)
legend('$\mu_{fl}$','$\mu_{fr}$','$\mu_{rl}$','$\mu_{rr}
$','Interpreter','latex','Location','northeast')
xlabel('t [s]','Interpreter','latex')
ylabel('$\mu$','Interpreter','latex')
ylim([0,1]);
title('Friction Variations','Interpreter','latex')
grid on;
hold off;
% Disturbance
subplot(2,1,2)
plot(t/1000,w_signal)
xlabel('t [s]','Interpreter','latex')
ylabel('$F_w$ [N]','Interpreter','latex')
ylim([0,1]);
title('Disturbance Signal','Interpreter','latex')
grid on;
```



```
% Simulate straight-line motion with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_ol = psinfo(S_ol, 'eval', [u1_signal(t) u2_signal(t) u3_signal(t)
u4_signal(t)]);
    [A_ol, B_ol, C_ol, D_ol] = ltiss(system_ol);
    X_dot(:,t) = A_ol*X(:,t) + B_ol*[(w_signal(t).*cos(Psi(t))); [0;0;0;0]];
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA = Beta;
BETA_DOT = X_dot(1,:);
PSI = Psi;
PSI_DOT = X(2,:);
PX = Px;
VX = Vx;
PY = Py;
VY = Vy;

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
```

```
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
plot(t/1000,PSI_DOT(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX(:,1:end-1)',PY(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
sgtitle('Simulation of Open-Loop Uncertain and Disturbed System @ $v = 20$ m/
s','Interpreter','latex')
```



Controller Design

```
% region = lmireg; % Pole placement constraint (intersection of the left half-plane with x < -0.1 and of the sector centered at the origin and with inner angle 3π/4)
region = [0.2000 + 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   1.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i;
          0.0000 + 0.0000i   0.0000 + 2.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.9239 + 0.0000i  -0.3827 + 0.0000i;
          0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.3827 + 0.0000i   0.9239 + 0.0000i];

g1_opt_global = msfsyn(S_ol, size(By2), [0 0 1 0], region) % Optimal quadratic H∞ performance subject to the pole placement constraint
```

```
Optimization of the H-infinity performance G :
----------------------------------------------

Solver for linear objective minimization under LMI constraints

Iterations   :    Best objective value so far

    1
    2
    3
    4
    5                    87.009423
    6                     8.057242
    7                     8.057242
    8                     3.605377
    9                     3.605377
   10                     1.316814
   11                     1.316814
```

```
    12                  1.316814
    13                  0.306918
    14                  0.252864
    15                  0.252864
    16                  0.147300
    17                  0.147300
    18                  0.147300
    19                  0.060891
    20                  0.060891
    21                  0.060891
    22                  0.060891
    23                  0.041812
    24                  0.041812
    25                  0.036825
    26                  0.036825
***                new lower bound: 4.563905e-03
    27                  0.034557
***                new lower bound: 6.158647e-03
    28                  0.033507
***                new lower bound: 7.110979e-03
    29                  0.033422
***                new lower bound: 7.925945e-03
    30                  0.030823
***                new lower bound: 8.907130e-03
    31                  0.030823
***                new lower bound: 9.629047e-03
    32                  0.030283
***                new lower bound: 9.868065e-03
    33                  0.028841
***                new lower bound:     0.011464
    34                  0.028841
***                new lower bound:     0.011961
    35                  0.028841
***                new lower bound:     0.012031
    36                  0.028409
***                new lower bound:     0.012051
    37                  0.028409
***                new lower bound:     0.013749
    38                  0.028409
***                new lower bound:     0.013801
    39                  0.028409
***                new lower bound:     0.013805
    40                  0.028409
***                new lower bound:     0.013851
    41                  0.028409
***                new lower bound:     0.013884
    42                  0.028409
***                new lower bound:     0.013962
    43                  0.027899
***                new lower bound:     0.014088
    44                  0.027681
***                new lower bound:     0.014614
    45                  0.027681
***                new lower bound:     0.016012
    46                  0.027681
***                new lower bound:     0.016063
    47                  0.027681
***                new lower bound:     0.016089
    48                  0.027063
***                new lower bound:     0.016185
    49                  0.027063
***                new lower bound:     0.016612
    50                  0.027028
***                new lower bound:     0.016716
```

```
    51                0.027028
***              new lower bound:     0.018132
    52                0.026613
***              new lower bound:     0.018247
    53                0.026595
***              new lower bound:     0.018653
    54                0.026595
***              new lower bound:     0.019787
    55                0.026341
***              new lower bound:     0.019893
    56                0.026283
***              new lower bound:     0.020249
    57                0.026283
***              new lower bound:     0.021214
    58                0.026283
***              new lower bound:     0.021250
    59                0.026188
***              new lower bound:     0.021308
    60                0.026188
***              new lower bound:     0.022142
    61                0.026188
***              new lower bound:     0.022162
    62                0.026188
***              new lower bound:     0.022178
    63                0.026188
***              new lower bound:     0.022191
    64                0.026042
***              new lower bound:     0.022240
    65                0.026042
***              new lower bound:     0.022437
    66                0.025996
***              new lower bound:     0.022486
    67                0.025996
***              new lower bound:     0.023154
    68                0.025996
***              new lower bound:     0.023187
    69                0.025996
***              new lower bound:     0.023189

 Result:  feasible solution
          best objective value:     0.025996
          f-radius saturation:  0.000% of R =  1.00e+10
 Termination due to SLOW PROGRESS:
          the objective was decreased by less than
          1.000% during the last 10 iterations.

 Guaranteed Hinf performance: 2.60e-02
g1_opt_global = 0.0260
```

```
g1_opt = []; % Array to store γ1* (Γee)
g2_opt = []; % Array to store γ2* (Γep)
K = []; % Array to store state-feedback controller gain K
S_cl = []; % Array to store closed-loop system representations
X = []; % Array to store corresponding Lyapunov matrices

for g = g1_opt_global:0.01:0.25 % Bound on H∞ performance
    [g1_opt(end+1), g2_opt(end+1), K(:,:,end+1), S_cl(:,:,end+1), X(:,:,end+1)] =
msfsyn(S_ol, size(By2), [g 0 0 1], region); % For a prescribed H∞ performance g>0,
compute the best H2 performance g2opt
end
```

```
 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
    22                  0.207323
    23                  0.051874
    24                  0.051874
***                 new lower bound:    -5.544293
    25                  0.037908
***                 new lower bound:    -1.374275
    26                  0.037908
***                 new lower bound:    -0.565406
    27                  0.027875
***                 new lower bound:    -0.436138
    28                  0.027875
***                 new lower bound:    -0.297301
    29                  0.021787
***                 new lower bound:    -0.204550
    30                  0.021787
***                 new lower bound:    -0.158810
    31                  0.020045
***                 new lower bound:    -0.104828
    32                  0.018457
***                 new lower bound:    -0.058342
    33                  0.018457
***                 new lower bound:    -0.039529
    34                  0.015802
***                 new lower bound:    -0.038287
    35                  0.015802
***                 new lower bound:    -0.034353
    36                  0.015802
***                 new lower bound:    -0.034237
    37                  0.014962
***                 new lower bound:    -0.030763
    38                  0.014962
***                 new lower bound:    -0.016929
    39                  0.014823
***                 new lower bound:    -0.016443
    40                  0.014823
***                 new lower bound: -6.877265e-03
```

```
    41                  0.014823
***               new lower bound: -6.351703e-03
    42                  0.013749
***               new lower bound: -6.149896e-03
    43                  0.013617
***               new lower bound: -4.528192e-03
    44                  0.013617
***               new lower bound: 1.413463e-03
    45                  0.013617
***               new lower bound: 1.662499e-03
    46                  0.013617
***               new lower bound: 1.745628e-03
    47                  0.013078
***               new lower bound: 2.000469e-03
    48                  0.012956
***               new lower bound: 2.982276e-03
    49                  0.012956
***               new lower bound: 6.998061e-03
    50                  0.012956
***               new lower bound: 7.111936e-03
    51                  0.012956
    52                  0.012956
***               new lower bound: 7.219207e-03
    53                  0.012626
***               new lower bound: 7.254106e-03
    54                  0.012626
***               new lower bound: 7.759497e-03
    55                  0.012584
***               new lower bound: 7.843679e-03
    56                  0.012584
***               new lower bound:     0.010143
    57                  0.012450
***               new lower bound:     0.010215
    58                  0.012450
***               new lower bound:     0.010450
    59                  0.012415
    60                  0.012396
***               new lower bound:     0.011154
    61                  0.012396
***               new lower bound:     0.011685
    62                  0.012396
***               new lower bound:     0.011708
    63                  0.012366
***               new lower bound:     0.011718
    64                  0.012359
***               new lower bound:     0.011802
    65                  0.012355
***               new lower bound:     0.012296

 Result:  feasible solution of required accuracy
          best objective value:     0.012355
          guaranteed absolute accuracy:  5.88e-05
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.60e-02
 Guaranteed   H2 performance: 1.11e-01

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far
```

```
      1
      2
      3
      4
      5
      6
      7
      8
      9
     10                   0.196926
     11                   0.124220
     12                   0.124220
     13                   0.124220
     14                   0.082347
     15                   0.063129
     16                   0.063129
     17                   0.055810
     18                   0.055810
***               new lower bound:    -0.367494
     19                   0.055810
***               new lower bound:    -0.358685
     20                   0.055810
     21                   0.055810
***               new lower bound:    -0.357510
     22                   0.037320
     23                   0.034796
***               new lower bound:    -0.314264
     24                   0.034796
***               new lower bound:    -0.203091
     25                   0.029614
***               new lower bound:    -0.198780
     26                   0.029614
***               new lower bound:    -0.149544
     27                   0.029614
***               new lower bound:    -0.146666
     28                   0.025708
***               new lower bound:    -0.141190
     29                   0.025708
***               new lower bound:    -0.106448
     30                   0.022869
***               new lower bound:    -0.104085
     31                   0.022869
***               new lower bound:    -0.076811
     32                   0.020211
***               new lower bound:    -0.074282
     33                   0.018273
***               new lower bound:    -0.062316
     34                   0.018273
***               new lower bound:    -0.047928
     35                   0.016504
***               new lower bound:    -0.046844
     36                   0.016504
***               new lower bound:    -0.039233
     37                   0.014928
***               new lower bound:    -0.038170
     38                   0.014505
***               new lower bound:    -0.032946
     39                   0.014505
***               new lower bound:    -0.023315
     40                   0.012777
***               new lower bound:    -0.022958
     41                   0.012038
***               new lower bound:    -0.021138
     42                   0.012038
```

```
***                 new lower bound:    -0.014698
   43                    0.012038
***                 new lower bound:    -0.014460
   44                9.570879e-03
***                 new lower bound:    -0.014182
   45                9.570879e-03
***                 new lower bound:    -0.012947
   46                8.600132e-03
***                 new lower bound:    -0.011615
   47                8.026808e-03
***                 new lower bound:    -0.010497
   48                8.026808e-03
***                 new lower bound: -9.183973e-03
   49                7.011801e-03
***                 new lower bound: -7.498271e-03
   50                6.564072e-03
***                 new lower bound: -6.747546e-03
   51                6.564072e-03
***                 new lower bound: -6.064402e-03
   52                5.761542e-03
***                 new lower bound: -4.754241e-03
   53                5.761542e-03
***                 new lower bound: -4.227827e-03
   54                5.422460e-03
***                 new lower bound: -3.210050e-03
   55                5.249922e-03
***                 new lower bound: -2.771311e-03
   56                4.681837e-03
***                 new lower bound: -1.102375e-03
   57                4.681837e-03
***                 new lower bound: -8.200983e-04
   58                4.681837e-03
***                 new lower bound: -7.600335e-04
   59                4.483613e-03
***                 new lower bound: -7.388074e-04
   60                4.483613e-03
***                 new lower bound: -4.374236e-04
   61                4.401300e-03
***                 new lower bound: 1.542774e-05
   62                4.401300e-03
***                 new lower bound: 1.232906e-03
   63                4.064281e-03
***                 new lower bound: 1.283315e-03
   64                4.064281e-03
***                 new lower bound: 1.471677e-03
   65                3.952683e-03
***                 new lower bound: 1.816060e-03
   66                3.910209e-03
***                 new lower bound: 1.981110e-03
   67                3.910209e-03
***                 new lower bound: 2.443377e-03
   68                3.865537e-03
***                 new lower bound: 2.636595e-03
   69                3.844874e-03
***                 new lower bound: 2.745564e-03
   70                3.830123e-03
***                 new lower bound: 3.087042e-03
   71                3.819032e-03
***                 new lower bound: 3.299376e-03
   72                3.814315e-03
***                 new lower bound: 3.431071e-03
   73                3.814315e-03
***                 new lower bound: 3.576812e-03
   74                3.814315e-03
```

```
***                 new lower bound: 3.579019e-03
   75                3.805816e-03
***                 new lower bound: 3.585276e-03
   76                3.803411e-03
***                 new lower bound: 3.601847e-03
   77                3.803411e-03
***                 new lower bound: 3.683449e-03
   78                3.803411e-03
***                 new lower bound: 3.684401e-03
   79                3.803411e-03
***                 new lower bound: 3.686692e-03
   80                3.803411e-03
***                 new lower bound: 3.689529e-03

 Result:  feasible solution
          best objective value: 3.803411e-03
          guaranteed absolute accuracy:  1.14e-04
          f-radius saturation:  0.000% of R =  1.00e+10
 Termination due to SLOW PROGRESS:
          the objective was decreased by less than
          1.000% during the last 10 iterations.

 Guaranteed Hinf performance: 3.60e-02
 Guaranteed   H2 performance: 6.17e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10                    0.420877
    11                    0.253576
    12                    0.253576
    13                    0.141201
    14                    0.141201
    15                    0.082875
    16                    0.082875
    17                    0.067310
    18                    0.057692
    19                    0.057692
***                 new lower bound:    -0.320158
    20                    0.057692
***                 new lower bound:    -0.314177
    21                    0.057692
***                 new lower bound:    -0.312646
    22                    0.046889
***                 new lower bound:    -0.306156
    23                    0.046889
***                 new lower bound:    -0.264754
    24                    0.042478
***                 new lower bound:    -0.260714
    25                    0.042478
***                 new lower bound:    -0.163859
```

```
    26                   0.037434
***                 new lower bound:    -0.160250
    27                   0.037434
***                 new lower bound:    -0.129146
    28                   0.033441
***                 new lower bound:    -0.126961
    29                   0.033441
***                 new lower bound:    -0.100465
    30                   0.030041
***                 new lower bound:    -0.098030
    31                   0.030041
***                 new lower bound:    -0.080097
    32                   0.025475
***                 new lower bound:    -0.078022
    33                   0.025475
***                 new lower bound:    -0.071286
    34                   0.024420
***                 new lower bound:    -0.061620
    35                   0.021044
***                 new lower bound:    -0.042227
    36                   0.021044
***                 new lower bound:    -0.039132
    37                   0.021044
***                 new lower bound:    -0.038439
    38                   0.019231
***                 new lower bound:    -0.036223
    39                   0.017761
***                 new lower bound:    -0.032948
    40                   0.017761
***                 new lower bound:    -0.027872
    41                   0.015495
***                 new lower bound:    -0.024207
    42                   0.014440
***                 new lower bound:    -0.021921
    43                   0.013424
***                 new lower bound:    -0.018377
    44                   0.013424
***                 new lower bound:    -0.016584
    45                   0.011483
***                 new lower bound:    -0.013641
    46                   0.010666
***                 new lower bound:    -0.012404
    47               9.900224e-03
***                 new lower bound:    -0.011287
    48               9.183527e-03
***                 new lower bound:    -0.010277
    49               8.513725e-03
***                 new lower bound: -9.362074e-03
    50               7.888855e-03
***                 new lower bound: -8.531885e-03
    51               7.307017e-03
***                 new lower bound: -7.777114e-03
    52               6.766361e-03
***                 new lower bound: -7.089680e-03
    53               6.265080e-03
***                 new lower bound: -6.462442e-03
    54               6.265080e-03
***                 new lower bound: -5.889067e-03
    55               5.486734e-03
***                 new lower bound: -4.479562e-03
    56               5.080548e-03
***                 new lower bound: -4.051703e-03
    57               5.080548e-03
***                 new lower bound: -3.654681e-03
```

```
   58              4.791308e-03
***               new lower bound: -2.740138e-03
   59              4.661325e-03
***               new lower bound: -2.381840e-03
   60              4.661325e-03
***               new lower bound: -4.856003e-04
   61              3.819535e-03
***               new lower bound: -4.170022e-04
   62              3.819535e-03
***               new lower bound: -1.784345e-04
   63              3.549014e-03
***               new lower bound: 2.127962e-05
   64              3.549014e-03
***               new lower bound: 2.485693e-04
   65              3.436615e-03
***               new lower bound: 1.032842e-03
   66              3.381006e-03
***               new lower bound: 1.196327e-03
   67              3.336335e-03
***               new lower bound: 1.570287e-03
   68              3.336335e-03
***               new lower bound: 1.878719e-03
   69              3.293707e-03
***               new lower bound: 2.008732e-03
   70              3.293707e-03
***               new lower bound: 2.111619e-03
   71              3.255801e-03
***               new lower bound: 2.384031e-03
   72              3.255801e-03
***               new lower bound: 2.460041e-03
   73              3.234001e-03
***               new lower bound: 2.664538e-03
   74              3.234001e-03
***               new lower bound: 2.716506e-03
   75              3.220887e-03
***               new lower bound: 2.864705e-03
   76              3.220887e-03
***               new lower bound: 2.899175e-03
   77              3.213757e-03
***               new lower bound: 3.001922e-03
   78              3.213757e-03
***               new lower bound: 3.024112e-03
   79              3.210217e-03
***               new lower bound: 3.091965e-03
   80              3.210217e-03
***               new lower bound: 3.109522e-03
   81              3.208506e-03
***               new lower bound: 3.178025e-03

 Result:  feasible solution of required accuracy
          best objective value: 3.208506e-03
          guaranteed absolute accuracy:  3.05e-05
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.60e-02
 Guaranteed   H2 performance: 5.66e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far
```

```
     1
     2
     3
     4
     5
     6
     7
     8
     9
    10                  0.435997
    11                  0.298155
    12                  0.295785
    13                  0.135966
    14                  0.135966
    15                  0.135966
    16                  0.092340
    17                  0.092340
    18                  0.070788
    19                  0.070788
    20                  0.063694
    21                  0.063694
***                new lower bound:    -0.257952
    22                  0.063694
***                new lower bound:    -0.255730
    23                  0.052341
    24                  0.052341
***                new lower bound:    -0.227649
    25                  0.052341
***                new lower bound:    -0.221793
    26                  0.049320
***                new lower bound:    -0.214114
    27                  0.049320
***                new lower bound:    -0.133975
    28                  0.042557
***                new lower bound:    -0.131326
    29                  0.042557
***                new lower bound:    -0.120971
    30                  0.042142
***                new lower bound:    -0.119004
    31                  0.042142
***                new lower bound:    -0.081020
    32                  0.034544
***                new lower bound:    -0.078791
    33                  0.034227
***                new lower bound:    -0.071733
    34                  0.034227
***                new lower bound:    -0.048525
    35                  0.028033
***                new lower bound:    -0.047560
    36                  0.027504
***                new lower bound:    -0.043385
    37                  0.027504
***                new lower bound:    -0.029187
    38                  0.027504
***                new lower bound:    -0.028855
    39                  0.020881
***                new lower bound:    -0.028388
    40                  0.020881
***                new lower bound:    -0.025908
    41                  0.018213
***                new lower bound:    -0.021935
    42                  0.018213
***                new lower bound:    -0.019959
    43                  0.014360
```

```
***                 new lower bound:    -0.017285
   44                    0.013310
***                 new lower bound:    -0.015769
   45                    0.012327
***                 new lower bound:    -0.014395
   46                    0.011408
***                 new lower bound:    -0.013148
   47                    0.010551
***                 new lower bound:    -0.012014
   48                9.754058e-03
***                 new lower bound:    -0.010981
   49                9.013532e-03
***                 new lower bound:    -0.010039
   50                7.677687e-03
***                 new lower bound: -9.178999e-03
   51                7.096505e-03
***                 new lower bound: -8.400782e-03
   52                6.048417e-03
***                 new lower bound: -7.687358e-03
   53                5.592686e-03
***                 new lower bound: -7.039279e-03
   54                5.173462e-03
***                 new lower bound: -6.442839e-03
   55                5.173462e-03
***                 new lower bound: -5.892880e-03
   56                4.761050e-03
***                 new lower bound: -3.383793e-03
   57                4.761050e-03
***                 new lower bound: -3.033259e-03
   58                4.169085e-03
***                 new lower bound: -2.505758e-03
   59                4.169085e-03
***                 new lower bound: -2.162511e-03
   60                3.774319e-03
***                 new lower bound: -1.392194e-03
   61                3.625906e-03
***                 new lower bound: -1.095617e-03
   62                3.498146e-03
***                 new lower bound: -7.625279e-04
   63                3.498146e-03
***                 new lower bound: -3.673982e-04
   64                3.319997e-03
***                 new lower bound: 3.558065e-04
   65                3.247601e-03
***                 new lower bound: 5.473603e-04
   66                3.247601e-03
***                 new lower bound: 9.406645e-04
   67                3.143225e-03
***                 new lower bound: 1.317391e-03
   68                3.104461e-03
***                 new lower bound: 1.449911e-03
   69                3.074576e-03
***                 new lower bound: 1.798178e-03
   70                3.074576e-03
***                 new lower bound: 2.067943e-03
   71                3.044068e-03
***                 new lower bound: 2.162688e-03
   72                3.044068e-03
***                 new lower bound: 2.238071e-03
   73                3.020191e-03
***                 new lower bound: 2.433147e-03
   74                3.020191e-03
***                 new lower bound: 2.486354e-03
   75                3.006328e-03
```

```
***                 new lower bound: 2.629134e-03
    76               3.006328e-03
***                 new lower bound: 2.664841e-03
    77               2.998319e-03
***                 new lower bound: 2.769552e-03
    78               2.998319e-03
***                 new lower bound: 2.792870e-03
    79               2.994222e-03
***                 new lower bound: 2.864442e-03
    80               2.994222e-03
***                 new lower bound: 2.880190e-03
    81               2.992268e-03
***                 new lower bound: 2.925044e-03
    82               2.992268e-03
***                 new lower bound: 2.936502e-03
    83               2.991383e-03
***                 new lower bound: 2.962965e-03

 Result:  feasible solution of required accuracy
          best objective value: 2.991383e-03
          guaranteed absolute accuracy:  2.84e-05
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 5.60e-02
 Guaranteed   H2 performance: 5.47e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10                   0.425924
    11                   0.425924
    12                   0.129233
    13                   0.129233
    14                   0.129233
    15                   0.078009
    16                   0.078009
    17                   0.078009
    18                   0.055039
    19                   0.055039
    20                   0.055039
    21                   0.050865
    22                   0.050865
    23                   0.050865
    24                   0.044944
    25                   0.044944
    26                   0.044944
    27                   0.043047
    28                   0.043047
***                 new lower bound:    -0.055109
    29                   0.043047
***                 new lower bound:    -0.054208
```

```
    30                   0.043047
    31                   0.036324
***                 new lower bound:    -0.053230
    32                   0.034788
***                 new lower bound:    -0.047510
    33                   0.034788
***                 new lower bound:    -0.030828
    34                   0.034788
***                 new lower bound:    -0.030526
    35                   0.026631
***                 new lower bound:    -0.030203
    36                   0.024500
***                 new lower bound:    -0.027550
    37                   0.024500
***                 new lower bound:    -0.025168
    38                   0.020460
***                 new lower bound:    -0.023170
    39                   0.018897
***                 new lower bound:    -0.021154
    40                   0.017445
***                 new lower bound:    -0.019325
    41                   0.016097
***                 new lower bound:    -0.017664
    42                   0.014846
***                 new lower bound:    -0.016153
    43                   0.013686
***                 new lower bound:    -0.014776
    44                   0.012611
***                 new lower bound:    -0.013521
    45                   0.011617
***                 new lower bound:    -0.012375
    46                   0.010698
***                 new lower bound:    -0.011329
    47               9.849363e-03
***                 new lower bound:    -0.010372
    48               9.066499e-03
***                 new lower bound: -9.497716e-03
    49               7.667097e-03
***                 new lower bound: -8.697021e-03
    50               7.063085e-03
***                 new lower bound: -7.972860e-03
    51               6.506892e-03
***                 new lower bound: -7.308518e-03
    52               5.995154e-03
***                 new lower bound: -6.698705e-03
    53               5.079274e-03
***                 new lower bound: -6.138629e-03
    54               5.079274e-03
***                 new lower bound: -5.629992e-03
    55               4.651518e-03
***                 new lower bound: -3.359426e-03
    56               4.651518e-03
***                 new lower bound: -3.045086e-03
    57               4.651518e-03
***                 new lower bound: -2.915501e-03
    58               4.084092e-03
***                 new lower bound: -2.586956e-03
    59               4.084092e-03
***                 new lower bound: -2.250108e-03
    60               3.682339e-03
***                 new lower bound: -1.467722e-03
    61               3.533729e-03
***                 new lower bound: -1.172184e-03
    62               3.405947e-03
```

```
***                 new lower bound: -8.495453e-04
   63                3.405947e-03
***                 new lower bound: -4.571174e-04
   64                3.225644e-03
***                 new lower bound: 2.500587e-04
   65                3.152375e-03
***                 new lower bound: 4.409282e-04
   66                3.091828e-03
***                 new lower bound: 8.253119e-04
   67                3.091828e-03
***                 new lower bound: 1.160013e-03
   68                3.008767e-03
***                 new lower bound: 1.438181e-03
   69                2.976377e-03
***                 new lower bound: 1.555047e-03
   70                2.951724e-03
***                 new lower bound: 1.873010e-03
   71                2.951724e-03
***                 new lower bound: 2.113252e-03
   72                2.926101e-03
***                 new lower bound: 2.177186e-03
   73                2.913298e-03
***                 new lower bound: 2.242375e-03
   74                2.903980e-03
***                 new lower bound: 2.445418e-03
   75                2.903763e-03
***                 new lower bound: 2.571267e-03
   76                2.893179e-03
***                 new lower bound: 2.638266e-03
   77                2.889301e-03
***                 new lower bound: 2.662895e-03
   78                2.886664e-03
***                 new lower bound: 2.743139e-03
   79                2.886278e-03
***                 new lower bound: 2.789957e-03
   80                2.886278e-03
***                 new lower bound: 2.832686e-03
   81                2.886278e-03
***                 new lower bound: 2.834678e-03
   82                2.884691e-03
***                 new lower bound: 2.835451e-03
   83                2.884374e-03
***                 new lower bound: 2.840468e-03
   84                2.884374e-03
***                 new lower bound: 2.862348e-03

 Result:  feasible solution of required accuracy
          best objective value: 2.884374e-03
          guaranteed absolute accuracy:  2.20e-05
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 6.60e-02
 Guaranteed   H2 performance: 5.37e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
```

```
      4
      5
      6
      7
      8
      9
     10
     11                     0.621726
     12                     0.343634
     13                     0.343634
     14                     0.125820
     15                     0.125820
     16                     0.125820
     17                     0.089040
     18                     0.089040
     19                     0.089040
     20                     0.084613
     21                     0.084613
     22                     0.067410
     23                     0.067410
     24                     0.067410
     25                     0.066601
     26                     0.066601
     27                     0.052545
     28                     0.052545
***                 new lower bound:    -0.106691
     29                     0.049866
***                 new lower bound:    -0.089370
     30                     0.049866
***                 new lower bound:    -0.051414
     31                     0.041802
***                 new lower bound:    -0.050577
     32                     0.041802
***                 new lower bound:    -0.045840
     33                     0.036171
***                 new lower bound:    -0.045365
     34                     0.036171
***                 new lower bound:    -0.041281
     35                     0.030861
***                 new lower bound:    -0.035477
     36                     0.028579
***                 new lower bound:    -0.032311
     37                     0.026439
***                 new lower bound:    -0.029470
     38                     0.024435
***                 new lower bound:    -0.026905
     39                     0.022563
***                 new lower bound:    -0.024580
     40                     0.020819
***                 new lower bound:    -0.022468
     41                     0.019197
***                 new lower bound:    -0.020547
     42                     0.017691
***                 new lower bound:    -0.018797
     43                     0.016295
***                 new lower bound:    -0.017202
     44                     0.015004
***                 new lower bound:    -0.015746
     45                     0.012690
***                 new lower bound:    -0.014416
     46                     0.011689
***                 new lower bound:    -0.013215
     47                     0.010766
***                 new lower bound:    -0.012115
```

```
   48                9.913923e-03
***                 new lower bound:    -0.011107
   49                9.129532e-03
***                 new lower bound:    -0.010182
   50                7.725487e-03
***                 new lower bound: -9.333980e-03
   51                7.118812e-03
***                 new lower bound: -8.564079e-03
   52                6.030829e-03
***                 new lower bound: -7.855829e-03
   53                5.108901e-03
***                 new lower bound: -7.210968e-03
   54                4.709199e-03
***                 new lower bound: -6.622348e-03
   55                4.709199e-03
***                 new lower bound: -6.078470e-03
   56                4.346743e-03
***                 new lower bound: -2.962310e-03
   57                4.346743e-03
***                 new lower bound: -2.655797e-03
   58                3.902406e-03
***                 new lower bound: -2.256216e-03
   59                3.902406e-03
***                 new lower bound: -1.937990e-03
   60                3.542943e-03
***                 new lower bound: -1.227872e-03
   61                3.408927e-03
***                 new lower bound: -9.489290e-04
   62                3.294573e-03
***                 new lower bound: -5.952141e-04
   63                3.198186e-03
***                 new lower bound: -1.977758e-04
   64                3.198186e-03
***                 new lower bound: 2.047200e-04
   65                3.060971e-03
***                 new lower bound: 7.147709e-04
   66                3.006398e-03
***                 new lower bound: 8.723528e-04
   67                2.962448e-03
***                 new lower bound: 1.231777e-03
   68                2.927156e-03
***                 new lower bound: 1.530593e-03
   69                2.927156e-03
***                 new lower bound: 1.776067e-03
   70                2.890033e-03
***                 new lower bound: 1.848585e-03
   71                2.870661e-03
***                 new lower bound: 1.933141e-03
   72                2.856370e-03
***                 new lower bound: 2.184670e-03
   73                2.856370e-03
***                 new lower bound: 2.355444e-03
   74                2.842547e-03
***                 new lower bound: 2.378210e-03
   75                2.835362e-03
***                 new lower bound: 2.437142e-03
   76                2.830250e-03
***                 new lower bound: 2.567968e-03
   77                2.829278e-03
***                 new lower bound: 2.645657e-03
   78                2.829278e-03
***                 new lower bound: 2.719894e-03
   79                2.823486e-03
***                 new lower bound: 2.722920e-03
```

```
   80               2.823486e-03
***                new lower bound: 2.730928e-03
   81               2.823486e-03
***                new lower bound: 2.733015e-03
   82               2.820899e-03
***                new lower bound: 2.749636e-03
   83               2.820899e-03
***                new lower bound: 2.758309e-03
   84               2.820417e-03
***                new lower bound: 2.793623e-03

 Result:  feasible solution of required accuracy
          best objective value: 2.820417e-03
          guaranteed absolute accuracy:  2.68e-05
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 7.60e-02
 Guaranteed   H2 performance: 5.31e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

    1
    2
    3
    4
    5
    6
    7
    8
    9
   10                   0.656394
   11                   0.542771
   12                   0.542771
   13                   0.304645
   14                   0.304645
   15                   0.304645
   16                   0.304645
   17                   0.214358
   18                   0.214358
   19                   0.214358
   20                   0.214358
   21                   0.195625
   22                   0.195625
   23                   0.142480
   24                   0.142480
   25                   0.142480
   26                   0.142480
   27                   0.126722
   28                   0.109043
   29                   0.109043
***                new lower bound:    -0.426690
   30                   0.109043
***                new lower bound:    -0.411138
   31                   0.109043
   32                   0.103205
***                new lower bound:    -0.375038
   33                   0.103205
***                new lower bound:    -0.209158
   34                   0.103205
```

```
    35                   0.086189
***                 new lower bound:    -0.204937
    36                   0.081924
***                 new lower bound:    -0.186082
    37                   0.081924
***                 new lower bound:    -0.119657
    38                   0.081924
***                 new lower bound:    -0.118570
    39                   0.081924
***                 new lower bound:    -0.117179
    40                   0.069071
***                 new lower bound:    -0.115823
    41                   0.069071
***                 new lower bound:    -0.106414
    42                   0.062944
***                 new lower bound:    -0.104554
    43                   0.062944
***                 new lower bound:    -0.095623
    44                   0.060737
***                 new lower bound:    -0.086423
    45                   0.060737
***                 new lower bound:    -0.055738
    46                   0.047871
***                 new lower bound:    -0.054770
    47                   0.044123
***                 new lower bound:    -0.049988
    48                   0.044123
***                 new lower bound:    -0.045651
    49                   0.037485
***                 new lower bound:    -0.043198
    50                   0.034610
***                 new lower bound:    -0.039449
    51                   0.031942
***                 new lower bound:    -0.036045
    52                   0.029467
***                 new lower bound:    -0.032951
    53                   0.027171
***                 new lower bound:    -0.030136
    54                   0.025042
***                 new lower bound:    -0.027574
    55                   0.023070
***                 new lower bound:    -0.025239
    56                   0.021244
***                 new lower bound:    -0.023109
    57                   0.019555
***                 new lower bound:    -0.021166
    58                   0.017994
***                 new lower bound:    -0.019392
    59                   0.016551
***                 new lower bound:    -0.017771
    60                   0.015220
***                 new lower bound:    -0.016288
    61                   0.013992
***                 new lower bound:    -0.014932
    62                   0.012861
***                 new lower bound:    -0.013691
    63                   0.010841
***                 new lower bound:    -0.012554
    64               9.969293e-03
***                 new lower bound:    -0.011525
    65               9.167359e-03
***                 new lower bound:    -0.010582
    66               8.429649e-03
***                 new lower bound: -9.715712e-03
```

```
    67              7.751383e-03
***                new lower bound: -8.920343e-03
    68              7.128084e-03
***                new lower bound: -8.189718e-03
    69              6.555569e-03
***                new lower bound: -7.518339e-03
    70              6.555569e-03
***                new lower bound: -6.901196e-03
    71              4.473935e-03
***                new lower bound: -4.457125e-03
    72              4.473935e-03
***                new lower bound: -4.076954e-03
    73              4.417970e-03
***                new lower bound: -2.773130e-03
    74              4.417970e-03
***                new lower bound: -1.256880e-03
    75              3.435559e-03
***                new lower bound: -1.072715e-03
    76              3.435559e-03
***                new lower bound: -8.121751e-04
    77              3.286541e-03
***                new lower bound: -4.040146e-04
    78              3.188989e-03
***                new lower bound: -1.791137e-04
    79              3.106307e-03
***                new lower bound: 1.879811e-04
    80              3.106307e-03
***                new lower bound: 5.367118e-04
    81              2.984903e-03
***                new lower bound: 8.945770e-04
    82              2.937581e-03
***                new lower bound: 1.038419e-03
    83              2.900060e-03
***                new lower bound: 1.382839e-03
    84              2.870346e-03
***                new lower bound: 1.664899e-03
    85              2.870346e-03
***                new lower bound: 1.891632e-03
    86              2.838897e-03
***                new lower bound: 1.923627e-03
    87              2.838897e-03
***                new lower bound: 1.999849e-03
    88              2.812503e-03
***                new lower bound: 2.194454e-03
    89              2.812503e-03
***                new lower bound: 2.249310e-03
    90              2.797178e-03
***                new lower bound: 2.395288e-03
    91              2.797178e-03
***                new lower bound: 2.432518e-03
    92              2.788195e-03
***                new lower bound: 2.540425e-03
    93              2.788195e-03
***                new lower bound: 2.564970e-03
    94              2.783458e-03
***                new lower bound: 2.639610e-03
    95              2.783458e-03
***                new lower bound: 2.655132e-03
    96              2.781172e-03
***                new lower bound: 2.703624e-03
    97              2.781172e-03
***                new lower bound: 2.715983e-03
    98              2.780095e-03
***                new lower bound: 2.753684e-03
```

```
 Result:  feasible solution of required accuracy
          best objective value: 2.780095e-03
          guaranteed absolute accuracy:  2.64e-05
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 8.60e-02
 Guaranteed   H2 performance: 5.27e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10                    0.625828
    11                    0.407831
    12                    0.407831
    13                    0.218778
    14                    0.218778
    15                    0.173002
    16                    0.163202
    17                    0.163202
    18                    0.163202
    19                    0.163202
    20                    0.163202
    21                    0.163202
    22                    0.127587
    23                    0.127587
    24                    0.127587
    25                    0.119975
    26                    0.119975
    27                    0.119975
    28                    0.101218
    29                    0.101218
    30                    0.101218
    31                    0.096473
    32                    0.096473
    33                    0.096473
    34                    0.096473
    35                    0.079967
    36                    0.079967
***                 new lower bound:    -0.124206
    37                    0.078013
***                 new lower bound:    -0.121519
    38                    0.078013
***                 new lower bound:    -0.075512
    39                    0.060541
***                 new lower bound:    -0.074222
    40                    0.060541
***                 new lower bound:    -0.067375
    41                    0.054814
***                 new lower bound:    -0.063464
    42                    0.050625
```

```
***                   new lower bound:    -0.057754
    43                     0.046742
***                   new lower bound:    -0.052667
    44                     0.046742
***                   new lower bound:    -0.048079
    45                     0.030268
***                   new lower bound:    -0.041202
    46                     0.027967
***                   new lower bound:    -0.037766
    47                     0.023768
***                   new lower bound:    -0.034637
    48                     0.021920
***                   new lower bound:    -0.031811
    49                     0.020203
***                   new lower bound:    -0.029228
    50                     0.017079
***                   new lower bound:    -0.026865
    51                     0.015709
***                   new lower bound:    -0.024723
    52                     0.013217
***                   new lower bound:    -0.022758
    53                     0.012124
***                   new lower bound:    -0.020972
    54                     0.011116
***                   new lower bound:    -0.019330
    55                 9.286270e-03
***                   new lower bound:    -0.017822
    56                 8.484635e-03
***                   new lower bound:    -0.016446
    57                 7.030751e-03
***                   new lower bound:    -0.015147
    58                 6.393806e-03
***                   new lower bound:    -0.013992
    59                 5.247769e-03
***                   new lower bound:    -0.012666
    60                 4.745829e-03
***                   new lower bound:    -0.011710
    61                 4.745829e-03
***                   new lower bound:    -0.010460
    62                 4.366961e-03
***                   new lower bound: -3.871928e-03
    63                 4.366961e-03
***                   new lower bound: -3.531981e-03
    64                 4.366961e-03
***                   new lower bound: -2.578716e-03
    65                 3.912317e-03
***                   new lower bound: -2.491978e-03
    66                 3.912317e-03
***                   new lower bound: -2.167735e-03
    67                 3.531732e-03
***                   new lower bound: -1.439275e-03
    68                 3.388556e-03
***                   new lower bound: -1.152314e-03
    69                 3.266046e-03
***                   new lower bound: -8.359526e-04
    70                 3.162851e-03
***                   new lower bound: -4.462300e-04
    71                 3.162851e-03
***                   new lower bound: -3.088990e-05
    72                 3.018714e-03
***                   new lower bound: 5.577140e-04
    73                 2.960227e-03
***                   new lower bound: 7.200174e-04
    74                 2.912562e-03
```

```
***                 new lower bound: 1.074554e-03
    75               2.873848e-03
***                 new lower bound: 1.372865e-03
    76               2.873848e-03
***                 new lower bound: 1.621609e-03
    77               2.831540e-03
***                 new lower bound: 1.715239e-03
    78               2.810222e-03
***                 new lower bound: 1.803880e-03
    79               2.794399e-03
***                 new lower bound: 2.062770e-03
    80               2.782234e-03
***                 new lower bound: 2.243656e-03
    81               2.777867e-03
***                 new lower bound: 2.365895e-03
    82               2.777867e-03
***                 new lower bound: 2.510799e-03
    83               2.777867e-03
***                 new lower bound: 2.512894e-03
    84               2.777867e-03
***                 new lower bound: 2.516900e-03
    85               2.767564e-03
***                 new lower bound: 2.523437e-03
    86               2.765109e-03
***                 new lower bound: 2.542795e-03
    87               2.765109e-03
***                 new lower bound: 2.624369e-03
    88               2.765109e-03
***                 new lower bound: 2.625797e-03
    89               2.759311e-03
***                 new lower bound: 2.629310e-03
    90               2.758060e-03
***                 new lower bound: 2.639483e-03

 Result:  feasible solution
          best objective value: 2.758060e-03
          guaranteed absolute accuracy:  1.19e-04
          f-radius saturation:  0.000% of R =  1.00e+10
 Termination due to SLOW PROGRESS:
          the objective was decreased by less than
          1.000% during the last 10 iterations.

 Guaranteed Hinf performance: 9.60e-02
 Guaranteed   H2 performance: 5.25e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10                    0.627417
    11                    0.627417
    12                    0.217355
```

```
    13                   0.217355
    14                   0.164429
    15                   0.164429
    16                   0.154791
    17                   0.154791
    18                   0.154791
    19                   0.154791
    20                   0.154791
    21                   0.119075
    22                   0.119075
    23                   0.119075
    24                   0.119075
    25                   0.115647
    26                   0.115647
    27                   0.091547
    28                   0.091547
    29                   0.091547
    30                   0.091547
    31                   0.077479
    32                   0.077479
    33                   0.076554
    34                   0.056223
    35                   0.056223
***                 new lower bound:    -0.057727
    36                   0.045724
***                 new lower bound:    -0.055961
    37                   0.042186
***                 new lower bound:    -0.050077
    38                   0.042186
***                 new lower bound:    -0.045619
    39                   0.024886
***                 new lower bound:    -0.035046
    40                   0.022916
***                 new lower bound:    -0.032186
    41                   0.021094
***                 new lower bound:    -0.029587
    42                   0.017795
***                 new lower bound:    -0.027209
    43                   0.016353
***                 new lower bound:    -0.025052
    44                   0.015023
***                 new lower bound:    -0.023070
    45                   0.013797
***                 new lower bound:    -0.021250
    46                   0.011579
***                 new lower bound:    -0.019576
    47                   0.010610
***                 new lower bound:    -0.018052
    48               9.719403e-03
***                 new lower bound:    -0.016648
    49               8.107926e-03
***                 new lower bound:    -0.015356
    50               7.403907e-03
***                 new lower bound:    -0.014176
    51               6.758219e-03
***                 new lower bound:    -0.013087
    52               5.589460e-03
***                 new lower bound:    -0.012084
    53               5.078643e-03
***                 new lower bound:    -0.011165
    54               4.614552e-03
***                 new lower bound:    -0.010151
    55               4.614552e-03
***                 new lower bound: -9.224218e-03
```

```
   56              4.246134e-03
***               new lower bound: -3.424042e-03
   57              4.246134e-03
***               new lower bound: -3.111431e-03
   58              4.195388e-03
***               new lower bound: -2.352918e-03
   59              4.195388e-03
***               new lower bound: -3.376353e-04
   60              3.232815e-03
***               new lower bound: -2.866821e-04
   61              3.135986e-03
***               new lower bound: -7.755264e-05
   62              3.135986e-03
***               new lower bound: 2.127548e-04
   63              2.997691e-03
***               new lower bound: 5.776970e-04
   64              2.940199e-03
***               new lower bound: 7.377017e-04
   65              2.893400e-03
***               new lower bound: 1.087176e-03
   66              2.893400e-03
***               new lower bound: 1.381854e-03
   67              2.833163e-03
***               new lower bound: 1.535018e-03
   68              2.807521e-03
***               new lower bound: 1.635386e-03
   69              2.788184e-03
***               new lower bound: 1.920655e-03
   70              2.773552e-03
***               new lower bound: 2.124845e-03
   71              2.768685e-03
***               new lower bound: 2.273243e-03
   72              2.768685e-03
***               new lower bound: 2.445701e-03
   73              2.768685e-03
***               new lower bound: 2.448812e-03
   74              2.751021e-03
***               new lower bound: 2.456199e-03
   75              2.751021e-03
***               new lower bound: 2.478206e-03
   76              2.750013e-03
***               new lower bound: 2.485488e-03
   77              2.750013e-03
***               new lower bound: 2.590611e-03
   78              2.741445e-03
***               new lower bound: 2.595213e-03
   79              2.740158e-03
***               new lower bound: 2.608142e-03
   80              2.740158e-03
***               new lower bound: 2.661356e-03
   81              2.740158e-03
***               new lower bound: 2.664057e-03
   82              2.740158e-03
***               new lower bound: 2.666086e-03
   83              2.735632e-03
***               new lower bound: 2.668517e-03
   84              2.735632e-03
***               new lower bound: 2.676476e-03

 Result:  feasible solution
          best objective value: 2.735632e-03
          guaranteed absolute accuracy:  5.92e-05
          f-radius saturation:  0.000% of R =  1.00e+10
 Termination due to SLOW PROGRESS:
```

```
         the objective was decreased by less than
         1.000% during the last 10 iterations.

 Guaranteed Hinf performance: 1.06e-01
 Guaranteed   H2 performance: 5.23e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ---------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10                    0.618037
    11                    0.618037
    12                    0.195467
    13                    0.195467
    14                    0.147351
    15                    0.147351
    16                    0.145750
    17                    0.145750
    18                    0.126733
    19                    0.126733
    20                    0.126733
    21                    0.108939
    22                    0.108939
    23                    0.108939
    24                    0.108939
    25                    0.108939
    26                    0.108939
    27                    0.108253
    28                    0.091334
    29                    0.091334
    30                    0.091334
    31                    0.090367
    32                    0.090367
    33                    0.060470
    34                    0.060470
    35                    0.052405
    36                    0.052405
    37                    0.033495
***                 new lower bound:    -0.047767
    38                    0.030861
***                 new lower bound:    -0.042499
    39                    0.026099
***                 new lower bound:    -0.038832
    40                    0.024015
***                 new lower bound:    -0.035674
    41                    0.022087
***                 new lower bound:    -0.032816
    42                    0.018592
***                 new lower bound:    -0.030203
    43                    0.017063
***                 new lower bound:    -0.027830
    44                    0.015654
```

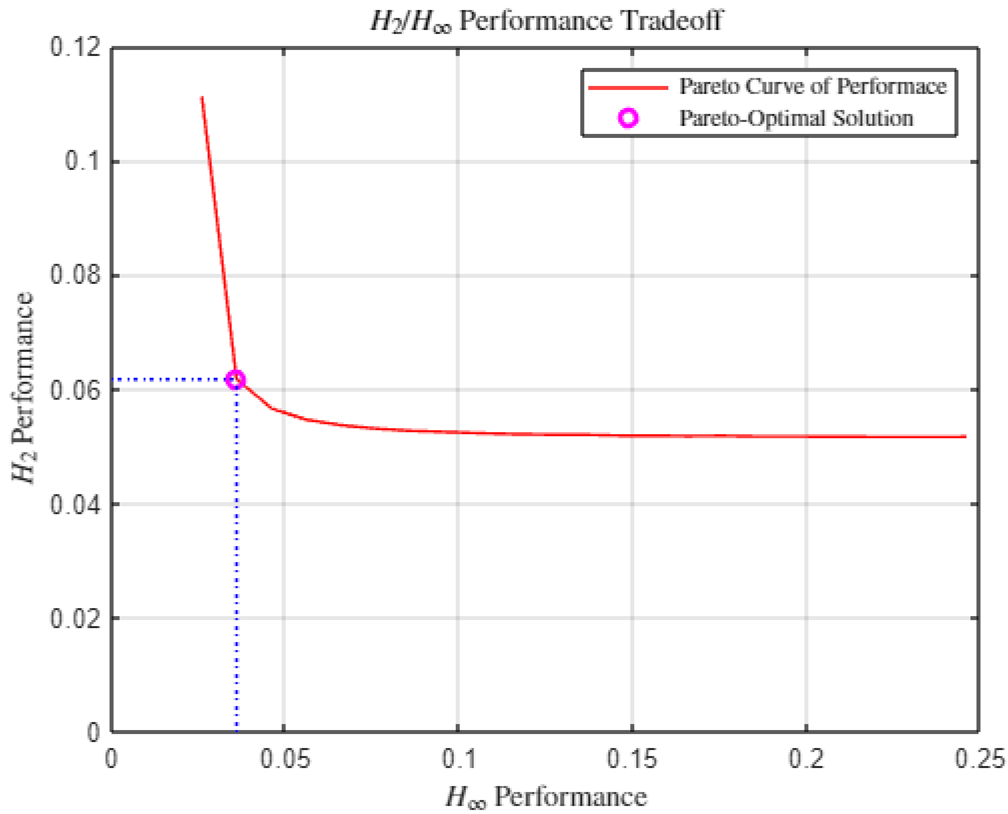
```
***                 new lower bound:    -0.025649
   45                    0.013100
***                 new lower bound:    -0.023644
   46                    0.011983
***                 new lower bound:    -0.021817
   47                    0.010957
***                 new lower bound:    -0.020134
   48                    0.010014
***                 new lower bound:    -0.018584
   49                8.306333e-03
***                 new lower bound:    -0.017156
   50                7.559860e-03
***                 new lower bound:    -0.015851
   51                6.877881e-03
***                 new lower bound:    -0.014537
   52                5.647029e-03
***                 new lower bound:    -0.013322
   53                5.108829e-03
***                 new lower bound:    -0.012319
   54                4.623187e-03
***                 new lower bound:    -0.011087
   55                4.623187e-03
***                 new lower bound: -9.983138e-03
   56                4.257310e-03
***                 new lower bound: -3.654694e-03
   57                4.257310e-03
***                 new lower bound: -3.327131e-03
   58                4.185559e-03
***                 new lower bound: -2.402578e-03
   59                4.185559e-03
***                 new lower bound: -1.633367e-04
   60                3.212575e-03
***                 new lower bound: -1.141737e-04
   61                3.212575e-03
***                 new lower bound: 7.735940e-05
   62                3.062079e-03
***                 new lower bound: 2.198671e-04
   63                3.062079e-03
***                 new lower bound: 4.007448e-04
   64                2.933702e-03
***                 new lower bound: 8.034915e-04
   65                2.884771e-03
***                 new lower bound: 9.482215e-04
   66                2.845683e-03
***                 new lower bound: 1.286230e-03
   67                2.814549e-03
***                 new lower bound: 1.565916e-03
   68                2.814549e-03
***                 new lower bound: 1.793881e-03
   69                2.780914e-03
***                 new lower bound: 1.837773e-03
   70                2.763592e-03
***                 new lower bound: 1.915204e-03
   71                2.750879e-03
***                 new lower bound: 2.147521e-03
   72                2.750879e-03
***                 new lower bound: 2.303539e-03
   73                2.738337e-03
***                 new lower bound: 2.320836e-03
   74                2.731926e-03
***                 new lower bound: 2.372578e-03
   75                2.727402e-03
***                 new lower bound: 2.492520e-03
   76                2.726681e-03
```

```
***                 new lower bound: 2.563404e-03
    77               2.726681e-03
***                 new lower bound: 2.629480e-03
    78               2.722393e-03
***                 new lower bound: 2.632556e-03
    79               2.721631e-03
***                 new lower bound: 2.640069e-03
    80               2.721631e-03
***                 new lower bound: 2.672508e-03
    81               2.721631e-03
***                 new lower bound: 2.673262e-03
    82               2.719588e-03
***                 new lower bound: 2.674081e-03
    83               2.719588e-03
***                 new lower bound: 2.677584e-03

 Result:  feasible solution
          best objective value: 2.719588e-03
          guaranteed absolute accuracy:  4.20e-05
          f-radius saturation:  0.000% of R =  1.00e+10
 Termination due to SLOW PROGRESS:
          the objective was decreased by less than
          1.000% during the last 10 iterations.

 Guaranteed Hinf performance: 1.16e-01
 Guaranteed   H2 performance: 5.21e-02

 Optimization of   0.000 * G^2 +  1.000 * H^2 :
 ----------------------------------------------

 Solver for linear objective minimization under LMI constraints

 Iterations   :    Best objective value so far

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10                   0.867492
    11                   0.558759
    12                   0.558759
    13                   0.194236
    14                   0.194236
    15                   0.194236
    16                   0.131864
    17                   0.131864
    18                   0.131864
    19                   0.107412
    20                   0.107412
    21                   0.107412
    22                   0.105093
    23                   0.105093
    24                   0.105093
    25                   0.072593
    26                   0.072593
    27                   0.071298
    28                   0.071298
    29                   0.028983
```

```
    30                  0.026613
***                 new lower...
```

Closed-Loop System

```
% Find Pareto-optimal solution
dist = sqrt(g1_opt.^2 + g2_opt.^2);
idx_popt = find(dist == min(dist));
g1_popt = g1_opt(idx_popt); % γ1* (Γee)
g2_popt = g2_opt(idx_popt); % γ2* (Γep)
K = K(:,:,idx_popt); % State-feedback controller gain K
S_cl = S_cl(:,:,idx_popt); % Closed-loop system representation
X = X(:,:,idx_popt); % Lyapunov matrix

% Plot
figure()
plot(g1_opt, g2_opt, 'r-')
hold on
plot(g1_popt, g2_popt, 'mo', LineWidth=2)
plot([0, g1_popt], [g2_popt, g2_popt], 'b:');
plot([g1_popt, g1_popt], [0, g2_popt], 'b:');
xlabel('$H_{\infty}$ Performance', 'Interpreter','latex')
ylabel('$H_2$ Performance', 'Interpreter','latex')
legend('Pareto Curve of Performace', 'Pareto-Optimal Solution',
'Interpreter','latex')
hold off
title('$H_2/H_{\infty}$ Performance Tradeoff','Interpreter','latex');
grid('on')
```




```
psinfo(S_cl); % Print closed-loop system information
```

```
Polytopic model with 16 vertex systems
    Each system has 2 state(s), 1 input(s), and 12 output(s)
```

```
% Test robust stability of closed-loop system via parametric Lyapunov functions
[tmin_cl, Q0_cl, Q1_cl, Q2_cl, Q3_cl] = pdlstab(S_cl, [0 0])
```

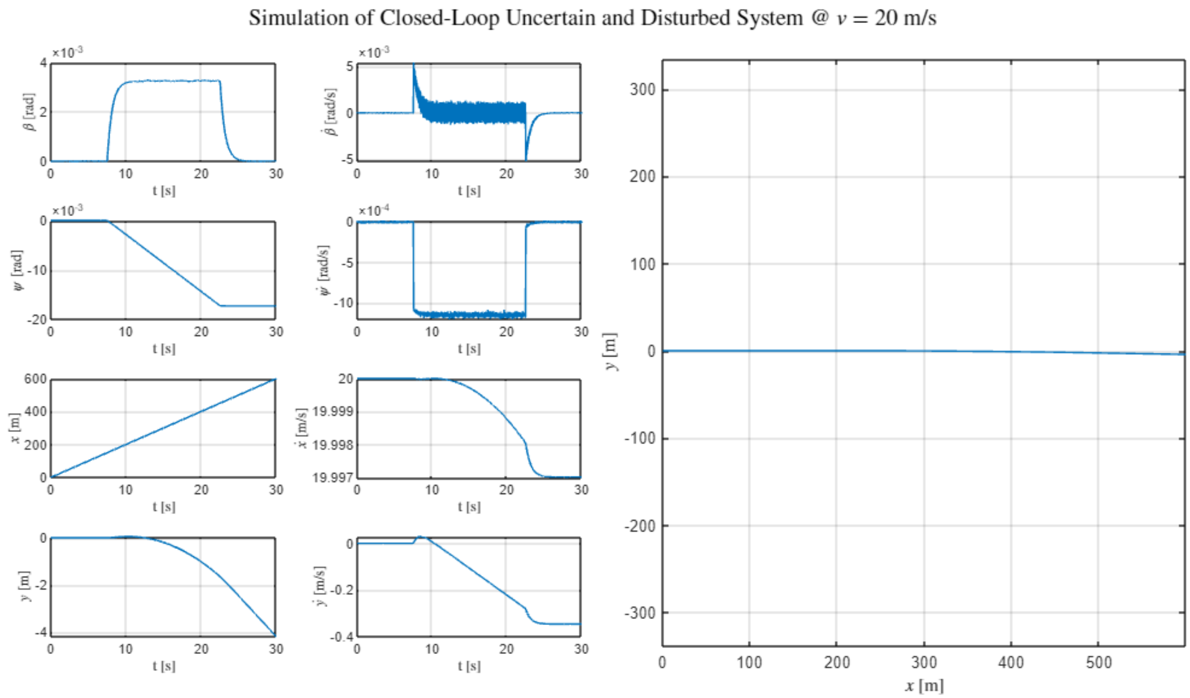
```
 Solver for LMI feasibility problems L(x) < R(x)
    This solver minimizes  t  subject to  L(x) < R(x) + t*I
    The best value of t should be negative for feasibility

 Iteration   :    Best value of t so far

     1                        0.232109
     2                        0.047558
     3                        0.047558
     4                        0.012415
     5                        0.012415
     6                    9.993207e-03
     7                    9.993207e-03
     8                    9.084775e-03
     9                    7.786999e-03
    10                       -0.102098

 Result:  best value of t:    -0.102098
          f-radius saturation:  0.037% of R =  1.00e+07

 This system is stable in the specified parameter range
tmin_cl = -0.1021
Q0_cl = 2×2
  233.9261   35.5462
```

```
   35.5462   10.2701
Q1_cl = 2×2
  233.9261   35.5462
   35.5462   10.2701
Q2_cl = 2×2
  233.9261   35.5462
   35.5462   10.2701
Q3_cl = 2×2
  233.9261   35.5462
   35.5462   10.2701
```

```
% Simulate straight-line motion with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_cl = psinfo(S_cl, 'eval', polydec(pv, [u1_signal(t) u2_signal(t)
u3_signal(t) u4_signal(t)]));
    [A_cl, B_cl, C_cl, D_cl] = ltiss(system_cl);
    X_dot(:,t) = A_cl*X(:,t) + B_cl*(w_signal(:,t).*cos(Psi(t)));
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA = Beta;
BETA_DOT = X_dot(1,:);
PSI = Psi;
PSI_DOT = X(2,:);
PX = Px;
VX = Vx;
PY = Py;
VY = Vy;
```

```
% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
plot(t/1000,PSI_DOT(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX(:,1:end-1)',PY(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
```

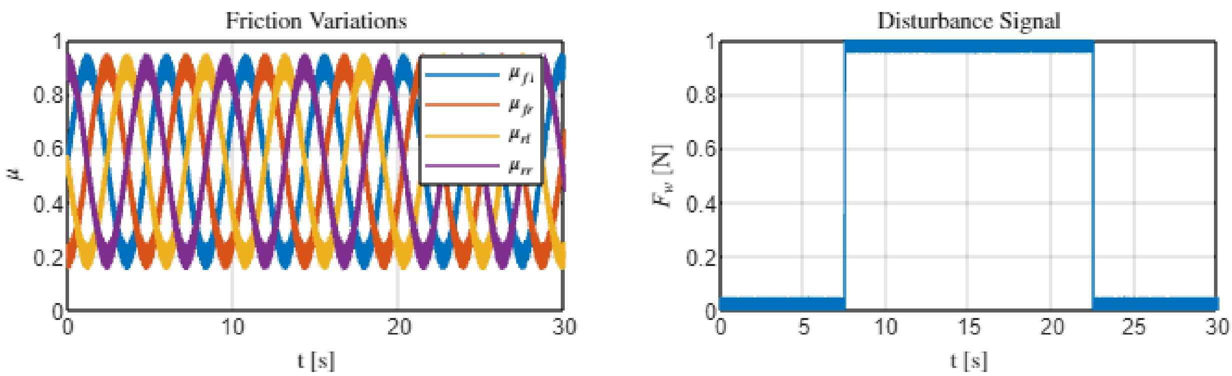
```
axis equal
grid('on')
% Common title and legend
sgtitle('Simulation of Closed-Loop Uncertain and Disturbed System @ $v = 20$ m/
s','Interpreter','latex')
```



Open-Loop vs. Closed Loop System Performance

```
% Friction variation and disturbance

rng(1,"twister"); % Fix random seed and algorithm

% Friction
theta = linspace(0, 2*pi, 30000);
frequency = 1;
u1_signal = 0.35 * sin(2*pi*frequency*theta) + 0.55;
u2_signal = 0.35 * sin(2*pi*frequency*theta - pi/2) + 0.55;
u3_signal = 0.35 * sin(2*pi*frequency*theta - pi) + 0.55;
u4_signal = 0.35 * sin(2*pi*frequency*theta - 3*pi/2) + 0.55;
u1_signal = u1_signal - 0.05 + 0.1 * rand(1, length(u1_signal));
u2_signal = u2_signal - 0.05 + 0.1 * rand(1, length(u2_signal));
u3_signal = u3_signal - 0.05 + 0.1 * rand(1, length(u3_signal));
u4_signal = u4_signal - 0.05 + 0.1 * rand(1, length(u4_signal));

% Disturbance
w_signal = [zeros(1,7500) ones(1,15000) zeros(1,7500)];
w_signal = w_signal - 0.05 + 0.1*(max(w_signal)) * rand(1, length(w_signal));

% Plot
```

```
t=1:30000;
fig = figure();
fig.Position(3:4) = [800, 200];
% Friction
subplot(1,2,1)
plot(t/1000,u1_signal)
hold on;
plot(t/1000,u2_signal)
plot(t/1000,u3_signal)
plot(t/1000,u4_signal)
legend('$\mu_{fl}$','$\mu_{fr}$','$\mu_{rl}$','$\mu_{rr}
$','Interpreter','latex','Location','northeast')
xlabel('t [s]','Interpreter','latex')
ylabel('$\mu$','Interpreter','latex')
xlim([0,30]);
ylim([0,1]);
title('Friction Variations','Interpreter','latex')
grid on;
hold off;
% Disturbance
subplot(1,2,2)
plot(t/1000,w_signal)
xlabel('t [s]','Interpreter','latex')
ylabel('$F_w$ [N]','Interpreter','latex')
xlim([0,30]);
ylim([0,1]);
title('Disturbance Signal','Interpreter','latex')
grid on;
```



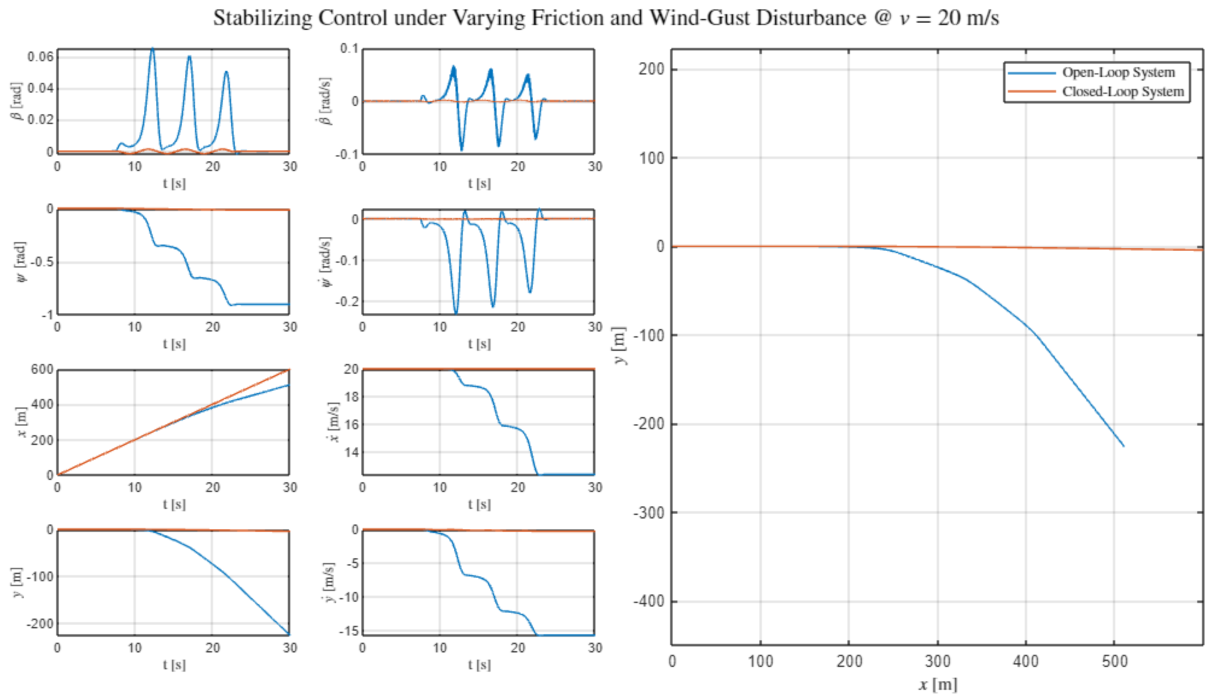
Stabilizing Controller

```
% Simulate open-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
```

```
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_ol = psinfo(S_ol, 'eval', [u1_signal(t) u2_signal(t) u3_signal(t)
u4_signal(t)]);
    [A_ol, B_ol, C_ol, D_ol] = ltiss(system_ol);
    X_dot(:,t) = A_ol*X(:,t) + B_ol*[(w_signal(t).*cos(Psi(t))); [0;0;0;0]];
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_OL = Beta;
BETA_DOT_OL = X_dot(1,:);
PSI_OL = Psi;
PSI_DOT_OL = X(2,:);
PX_OL = Px;
VX_OL = Vx;
PY_OL = Py;
VY_OL = Vy;

% Simulate closed-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;
```

```
% Simulation at each millisecond
for t=1:30000
    system_cl = psinfo(S_cl, 'eval', polydec(pv, [u1_signal(t) u2_signal(t)
u3_signal(t) u4_signal(t)]));
    [A_cl, B_cl, C_cl, D_cl] = ltiss(system_cl);
    X_dot(:,t) = A_cl*X(:,t) + B_cl*(w_signal(:,t).*cos(Psi(t)));
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_CL = Beta;
BETA_DOT_CL = X_dot(1,:);
PSI_CL = Psi;
PSI_DOT_CL = X(2,:);
PX_CL = Px;
VX_CL = Vx;
PY_CL = Py;
VY_CL = Vy;

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_OL(:,1:end)')
hold on
plot(t/1000,BETA_CL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_OL(:,1:end)')
hold on
plot(t/1000,BETA_DOT_CL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_CL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
```

```
subplot(4,4,6)
plot(t/1000,PSI_DOT_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_DOT_CL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_OL(:,1:end-1)')
hold on
plot(t/1000,PX_CL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_OL(:,1:end)')
hold on
plot(t/1000,VX_CL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_OL(:,1:end-1)')
hold on
plot(t/1000,PY_CL(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_OL(:,1:end)')
hold on
plot(t/1000,VY_CL(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_OL(:,1:end-1)',PY_OL(:,1:end-1)')
hold on
plot(PX_CL(:,1:end-1)',PY_CL(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
sgtitle('Stabilizing Control under Varying Friction and Wind-Gust Disturbance @ $v
= 20$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', ...
       'Interpreter','latex','Location','northeast')
```



Tracking Controller

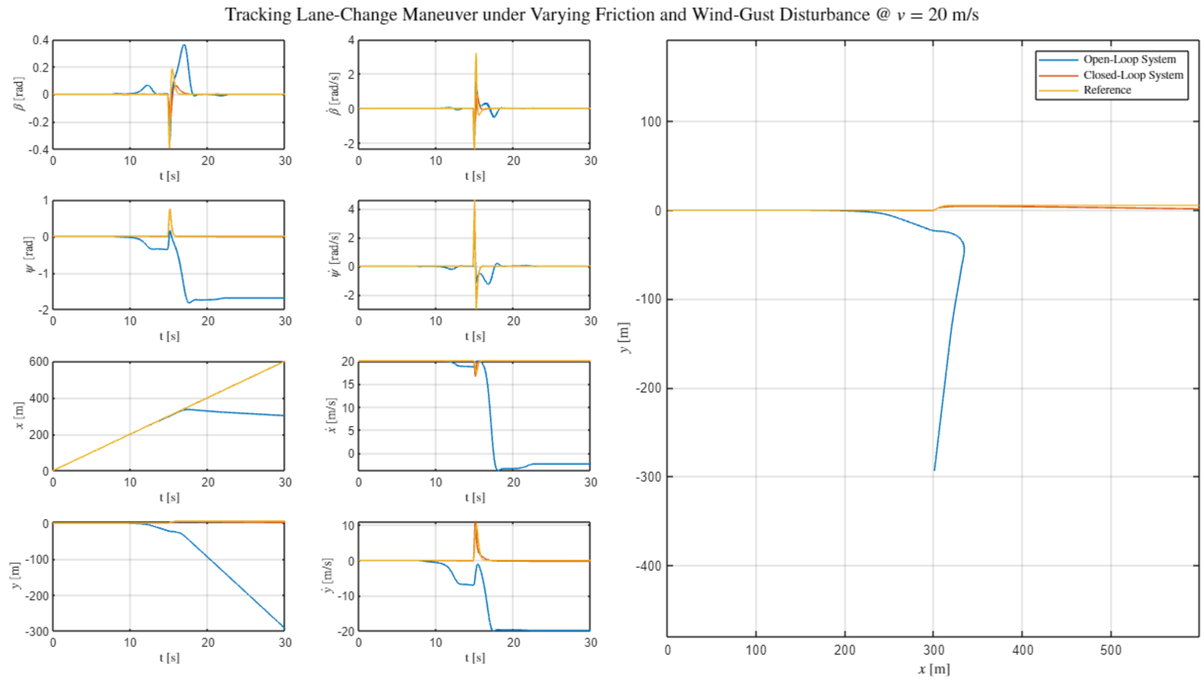
```
% Reference states (LC = lane-change, SP = skidpad, FH = fishhook, SL = slalom)
BETA_REF = BETA_LC(5,:);
BETA_DOT_REF = BETA_DOT_LC(5,:);
PSI_REF = PSI_LC(5,:);
PSI_DOT_REF = PSI_DOT_LC(5,:);
PX_REF = PX_LC(5,:);
VX_REF = VX_LC(5,:);
PY_REF = PY_LC(5,:);
VY_REF = VY_LC(5,:);
DELTA_REF = u_lane_change;

% Simulate open-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
```

```
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_ol = psinfo(S_ol, 'eval', [u1_signal(t) u2_signal(t) u3_signal(t)
u4_signal(t)]);
    [A_ol, B_ol, C_ol, D_ol] = ltiss(system_ol);
    X_dot(:,t) = A_ol*X(:,t) + B_ol*[(w_signal(:,t).*cos(Psi(t))); DELTA_REF(:,t)];
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_OL = Beta;
BETA_DOT_OL = X_dot(1,:);
PSI_OL = Psi;
PSI_DOT_OL = X(2,:);
PX_OL = Px;
VX_OL = Vx;
PY_OL = Py;
VY_OL = Vy;

% Simulate closed-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_cl = psinfo(S_cl, 'eval', polydec(pv, [u1_signal(t) u2_signal(t)
u3_signal(t) u4_signal(t)]));
    [A_cl, B_cl, C_cl, D_cl] = ltiss(system_cl);
```

```
    X_dot(:,t) = A_cl*(X(:,t) - [BETA_REF(t);PSI_DOT_REF(t)]) +
B_cl*(w_signal(:,t).*cos(Psi(t)));
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_CL = Beta;
BETA_DOT_CL = X_dot(1,:);
PSI_CL = Psi;
PSI_DOT_CL = X(2,:);
PX_CL = Px;
VX_CL = Vx;
PY_CL = Py;
VY_CL = Vy;

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_OL(:,1:end)')
hold on
plot(t/1000,BETA_CL(:,1:end)')
plot(t/1000,BETA_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_OL(:,1:end)')
hold on
plot(t/1000,BETA_DOT_CL(:,1:end)')
plot(t/1000,BETA_DOT_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_CL(:,1:end-1)')
plot(t/1000,PSI_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
plot(t/1000,PSI_DOT_OL(:,1:end-1)')
```

```
hold on
plot(t/1000,PSI_DOT_CL(:,1:end-1)')
plot(t/1000,PSI_DOT_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_OL(:,1:end-1)')
hold on
plot(t/1000,PX_CL(:,1:end-1)')
plot(t/1000,PX_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_OL(:,1:end)')
hold on
plot(t/1000,VX_CL(:,1:end)')
plot(t/1000,VX_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_OL(:,1:end-1)')
hold on
plot(t/1000,PY_CL(:,1:end-1)')
plot(t/1000,PY_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_OL(:,1:end)')
hold on
plot(t/1000,VY_CL(:,1:end)')
plot(t/1000,VY_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_OL(:,1:end-1)',PY_OL(:,1:end-1)')
hold on
plot(PX_CL(:,1:end-1)',PY_CL(:,1:end-1)')
plot(PX_REF(:,1:end-1)',PY_REF(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
```

```
sgtitle('Tracking Lane-Change Maneuver under Varying Friction and Wind-Gust
Disturbance @ $v = 20$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', 'Reference', ...
       'Interpreter','latex','Location','northeast')
```



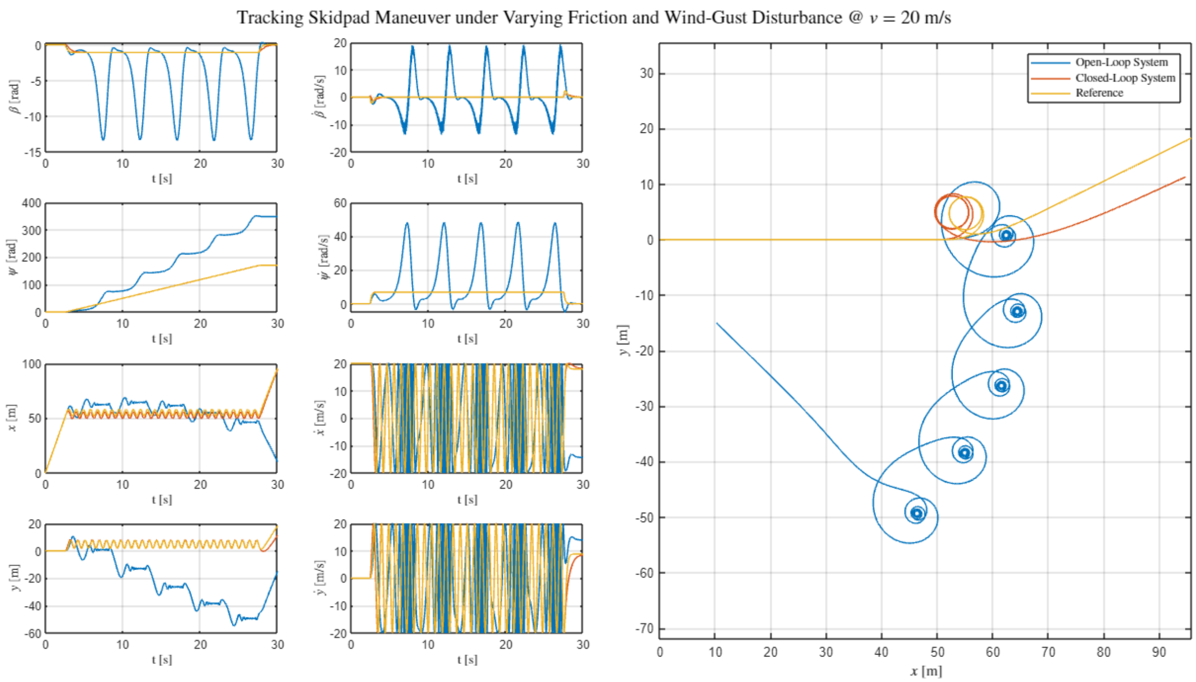

```
% Reference states (LC = lane-change, SP = skidpad, FH = fishhook, SL = slalom)
BETA_REF = BETA_SP(5,:);
BETA_DOT_REF = BETA_DOT_SP(5,:);
PSI_REF = PSI_SP(5,:);
PSI_DOT_REF = PSI_DOT_SP(5,:);
PX_REF = PX_SP(5,:);
VX_REF = VX_SP(5,:);
PY_REF = PY_SP(5,:);
VY_REF = VY_SP(5,:);
DELTA_REF = u_skidpad;

% Simulate open-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
```

```
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_ol = psinfo(S_ol, 'eval', [u1_signal(t) u2_signal(t) u3_signal(t)
u4_signal(t)]);
    [A_ol, B_ol, C_ol, D_ol] = ltiss(system_ol);
    X_dot(:,t) = A_ol*X(:,t) + B_ol*[(w_signal(:,t).*cos(Psi(t))); DELTA_REF(:,t)];
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_OL = Beta;
BETA_DOT_OL = X_dot(1,:);
PSI_OL = Psi;
PSI_DOT_OL = X(2,:);
PX_OL = Px;
VX_OL = Vx;
PY_OL = Py;
VY_OL = Vy;

% Simulate closed-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_cl = psinfo(S_cl, 'eval', polydec(pv, [u1_signal(t) u2_signal(t)
u3_signal(t) u4_signal(t)]));
```

```
    [A_cl, B_cl, C_cl, D_cl] = ltiss(system_cl);
    X_dot(:,t) = A_cl*(X(:,t) - [BETA_REF(t);PSI_DOT_REF(t)]) +
B_cl*(w_signal(:,t).*cos(Psi(t)));
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_CL = Beta;
BETA_DOT_CL = X_dot(1,:);
PSI_CL = Psi;
PSI_DOT_CL = X(2,:);
PX_CL = Px;
VX_CL = Vx;
PY_CL = Py;
VY_CL = Vy;

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_OL(:,1:end)')
hold on
plot(t/1000,BETA_CL(:,1:end)')
plot(t/1000,BETA_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_OL(:,1:end)')
hold on
plot(t/1000,BETA_DOT_CL(:,1:end)')
plot(t/1000,BETA_DOT_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_CL(:,1:end-1)')
plot(t/1000,PSI_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
```

```
plot(t/1000,PSI_DOT_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_DOT_CL(:,1:end-1)')
plot(t/1000,PSI_DOT_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_OL(:,1:end-1)')
hold on
plot(t/1000,PX_CL(:,1:end-1)')
plot(t/1000,PX_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_OL(:,1:end)')
hold on
plot(t/1000,VX_CL(:,1:end)')
plot(t/1000,VX_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_OL(:,1:end-1)')
hold on
plot(t/1000,PY_CL(:,1:end-1)')
plot(t/1000,PY_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_OL(:,1:end)')
hold on
plot(t/1000,VY_CL(:,1:end)')
plot(t/1000,VY_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_OL(:,1:end-1)',PY_OL(:,1:end-1)')
hold on
plot(PX_CL(:,1:end-1)',PY_CL(:,1:end-1)')
plot(PX_REF(:,1:end-1)',PY_REF(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
```

```
sgtitle('Tracking Skidpad Maneuver under Varying Friction and Wind-Gust Disturbance
@ $v = 20$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', 'Reference', ...
       'Interpreter','latex','Location','northeast')
```



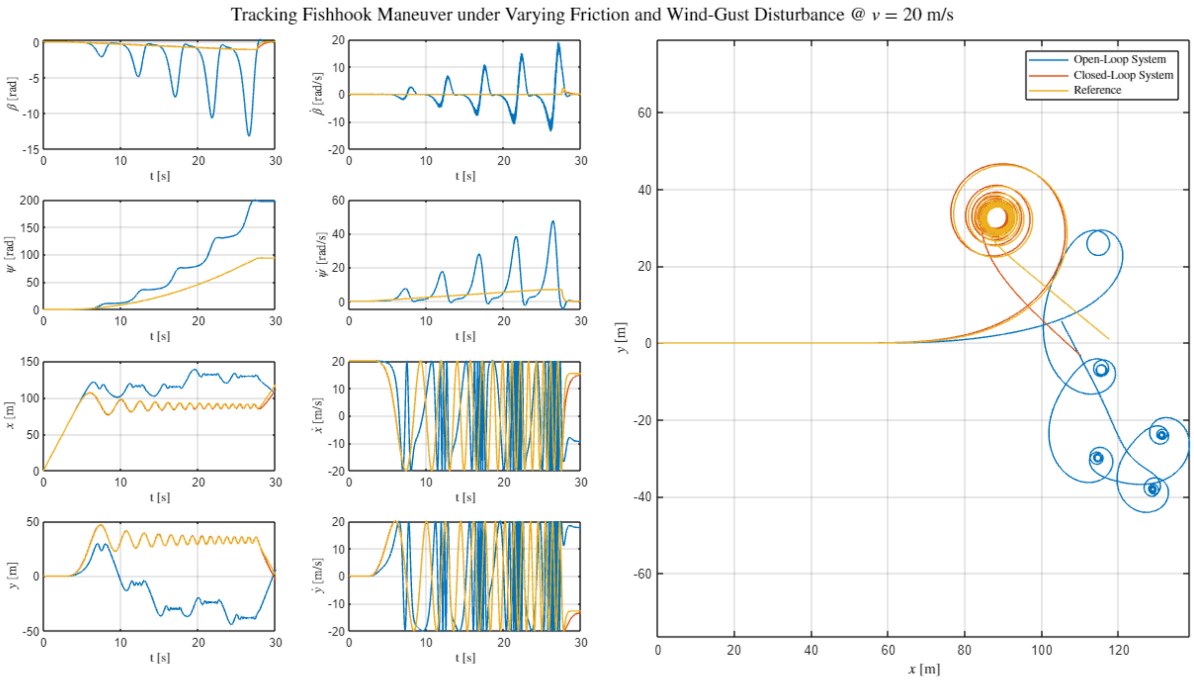

```
% Reference states (LC = lane-change, SP = skidpad, FH = fishhook, SL = slalom)
BETA_REF = BETA_FH(5,:);
BETA_DOT_REF = BETA_DOT_FH(5,:);
PSI_REF = PSI_FH(5,:);
PSI_DOT_REF = PSI_DOT_FH(5,:);
PX_REF = PX_FH(5,:);
VX_REF = VX_FH(5,:);
PY_REF = PY_FH(5,:);
VY_REF = VY_FH(5,:);
DELTA_REF = u_fishhook;

% Simulate open-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
```

```
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_ol = psinfo(S_ol, 'eval', [u1_signal(t) u2_signal(t) u3_signal(t)
u4_signal(t)]);
    [A_ol, B_ol, C_ol, D_ol] = ltiss(system_ol);
    X_dot(:,t) = A_ol*X(:,t) + B_ol*[(w_signal(:,t).*cos(Psi(t))); DELTA_REF(:,t)];
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_OL = Beta;
BETA_DOT_OL = X_dot(1,:);
PSI_OL = Psi;
PSI_DOT_OL = X(2,:);
PX_OL = Px;
VX_OL = Vx;
PY_OL = Py;
VY_OL = Vy;

% Simulate closed-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_cl = psinfo(S_cl, 'eval', polydec(pv, [u1_signal(t) u2_signal(t)
u3_signal(t) u4_signal(t)]));
```

```
    [A_cl, B_cl, C_cl, D_cl] = ltiss(system_cl);
    X_dot(:,t) = A_cl*(X(:,t) - [BETA_REF(t);PSI_DOT_REF(t)]) +
B_cl*(w_signal(:,t).*cos(Psi(t)));
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_CL = Beta;
BETA_DOT_CL = X_dot(1,:);
PSI_CL = Psi;
PSI_DOT_CL = X(2,:);
PX_CL = Px;
VX_CL = Vx;
PY_CL = Py;
VY_CL = Vy;

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_OL(:,1:end)')
hold on
plot(t/1000,BETA_CL(:,1:end)')
plot(t/1000,BETA_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_OL(:,1:end)')
hold on
plot(t/1000,BETA_DOT_CL(:,1:end)')
plot(t/1000,BETA_DOT_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_CL(:,1:end-1)')
plot(t/1000,PSI_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
```

```
plot(t/1000,PSI_DOT_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_DOT_CL(:,1:end-1)')
plot(t/1000,PSI_DOT_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_OL(:,1:end-1)')
hold on
plot(t/1000,PX_CL(:,1:end-1)')
plot(t/1000,PX_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_OL(:,1:end)')
hold on
plot(t/1000,VX_CL(:,1:end)')
plot(t/1000,VX_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_OL(:,1:end-1)')
hold on
plot(t/1000,PY_CL(:,1:end-1)')
plot(t/1000,PY_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_OL(:,1:end)')
hold on
plot(t/1000,VY_CL(:,1:end)')
plot(t/1000,VY_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_OL(:,1:end-1)',PY_OL(:,1:end-1)')
hold on
plot(PX_CL(:,1:end-1)',PY_CL(:,1:end-1)')
plot(PX_REF(:,1:end-1)',PY_REF(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
```

```
sgtitle('Tracking Fishhook Maneuver under Varying Friction and Wind-Gust
Disturbance @ $v = 20$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', 'Reference', ...
       'Interpreter','latex','Location','northeast')
```




```
% Reference states (LC = lane-change, SP = skidpad, FH = fishhook, SL = slalom)
BETA_REF = BETA_SL(5,:);
BETA_DOT_REF = BETA_DOT_SL(5,:);
PSI_REF = PSI_SL(5,:);
PSI_DOT_REF = PSI_DOT_SL(5,:);
PX_REF = PX_SL(5,:);
VX_REF = VX_SL(5,:);
PY_REF = PY_SL(5,:);
VY_REF = VY_SL(5,:);
DELTA_REF = u_slalom;

% Simulate open-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
```

```
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_ol = psinfo(S_ol, 'eval', [u1_signal(t) u2_signal(t) u3_signal(t)
u4_signal(t)]);
    [A_ol, B_ol, C_ol, D_ol] = ltiss(system_ol);
    X_dot(:,t) = A_ol*X(:,t) + B_ol*[(w_signal(:,t).*cos(Psi(t))); DELTA_REF(:,t)];
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_OL = Beta;
BETA_DOT_OL = X_dot(1,:);
PSI_OL = Psi;
PSI_DOT_OL = X(2,:);
PX_OL = Px;
VX_OL = Vx;
PY_OL = Py;
VY_OL = Vy;

% Simulate closed-loop system with varying friction and disturbance

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 20; % Vehicle velocity [m/s]
X = [];
X(:,1) = [0;0];
X_dot = [];
Psi = [];
Psi(1) = 0;
Beta = [];
Vx = [];
Vy = [];
Px = [];
Px(1) = 0;
Py = [];
Py(1) = 0;

% Simulation at each millisecond
for t=1:30000
    system_cl = psinfo(S_cl, 'eval', polydec(pv, [u1_signal(t) u2_signal(t)
u3_signal(t) u4_signal(t)]));
```

```
    [A_cl, B_cl, C_cl, D_cl] = ltiss(system_cl);
    X_dot(:,t) = A_cl*(X(:,t) - [BETA_REF(t);PSI_DOT_REF(t)]) +
B_cl*(w_signal(:,t).*cos(Psi(t)));
    X(:,t+1) = X(:,t) + X_dot(:,t)*dt;
    Beta(t) = X(1,t);
    Psi(t+1) = Psi(t) + X(2,t)*dt;
    Vx(t) = v*cos(Psi(t) + Beta(t));
    Vy(t) = v*sin(Psi(t) + Beta(t));
    Px(t+1) = Px(t) + Vx(t)*dt;
    Py(t+1) = Py(t) + Vy(t)*dt;
end
BETA_CL = Beta;
BETA_DOT_CL = X_dot(1,:);
PSI_CL = Psi;
PSI_DOT_CL = X(2,:);
PX_CL = Px;
VX_CL = Vx;
PY_CL = Py;
VY_CL = Vy;

% Plot
t=1:30000;
fig = figure();
fig.Position(3:4) = [1500, 750];
% Plot states
subplot(4,4,1)
plot(t/1000,BETA_OL(:,1:end)')
hold on
plot(t/1000,BETA_CL(:,1:end)')
plot(t/1000,BETA_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\beta$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,2)
plot(t/1000,BETA_DOT_OL(:,1:end)')
hold on
plot(t/1000,BETA_DOT_CL(:,1:end)')
plot(t/1000,BETA_DOT_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\beta}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,5)
plot(t/1000,PSI_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_CL(:,1:end-1)')
plot(t/1000,PSI_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\psi$ [rad]','Interpreter','latex')
grid('on')
subplot(4,4,6)
```

```
plot(t/1000,PSI_DOT_OL(:,1:end-1)')
hold on
plot(t/1000,PSI_DOT_CL(:,1:end-1)')
plot(t/1000,PSI_DOT_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{\psi}$ [rad/s]','Interpreter','latex')
grid('on')
subplot(4,4,9)
plot(t/1000,PX_OL(:,1:end-1)')
hold on
plot(t/1000,PX_CL(:,1:end-1)')
plot(t/1000,PX_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$x$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,10)
plot(t/1000,VX_OL(:,1:end)')
hold on
plot(t/1000,VX_CL(:,1:end)')
plot(t/1000,VX_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY_OL(:,1:end-1)')
hold on
plot(t/1000,PY_CL(:,1:end-1)')
plot(t/1000,PY_REF(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY_OL(:,1:end)')
hold on
plot(t/1000,VY_CL(:,1:end)')
plot(t/1000,VY_REF(:,1:end)')
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{y}$ [m/s]','Interpreter','latex')
grid('on')
% Plot trajectory
subplot(4,4,[3,4,7,8,11,12,15,16])
plot(PX_OL(:,1:end-1)',PY_OL(:,1:end-1)')
hold on
plot(PX_CL(:,1:end-1)',PY_CL(:,1:end-1)')
plot(PX_REF(:,1:end-1)',PY_REF(:,1:end-1)')
xlabel('$x$ [m]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
axis equal
grid('on')
% Common title and legend
```

```
sgtitle('Tracking Slalom Maneuver under Varying Friction and Wind-Gust Disturbance
@ $v = 20$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', 'Reference', ...
       'Interpreter','latex','Location','northeast')
```