ME-8930 | LMIs for Robust & Optimal Control Design

Capstone Project [Small Scale Vehicle - Nigel]

Team: Chinmay Samak and Tanmay Samak

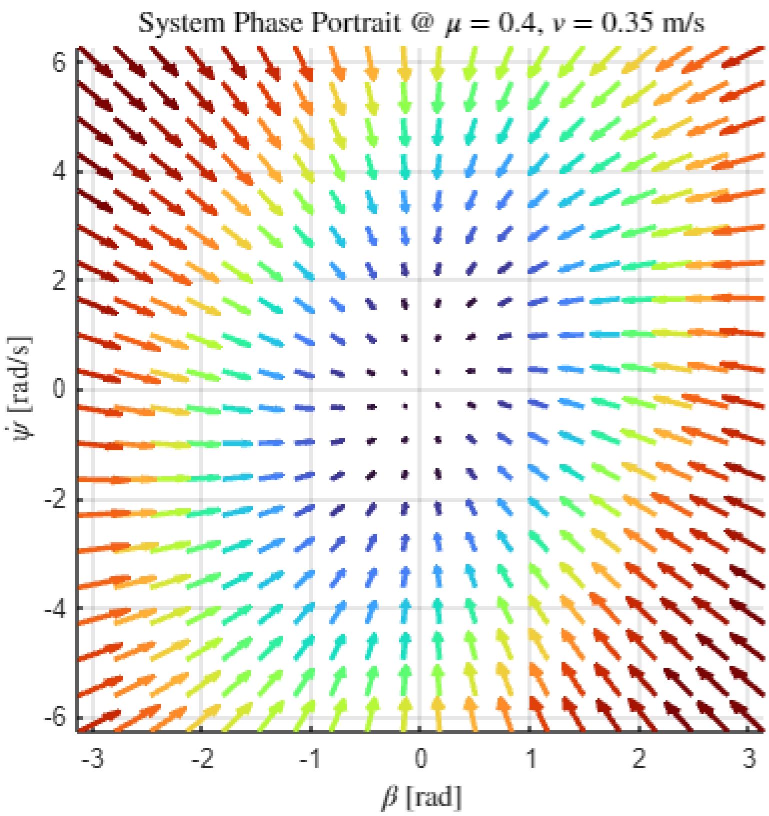
```
clear;
close;
clc;
```

System Modeling & Analysis

```
% System parameters

m = 2.68; % Mass [kg]
J = 0.01944; % Yaw moment of inertia [kg-m^2]
l = 0.1415; % Wheelbase [m]
l_f = 0.44*l; % Distance from CG to front axel [m]
l_r = l-l_f; % Distance from CG to rear axel [m]
C_fl = 22.4768; % Cornering stiffness of FL tire [N/rad]
C_fr = 22.4768; % Cornering stiffness of FR tire [N/rad]
C_rl = 22.4768; % Cornering stiffness of RL tire [N/rad]
C_rr = 22.4768; % Cornering stiffness of RR tire [N/rad]
```

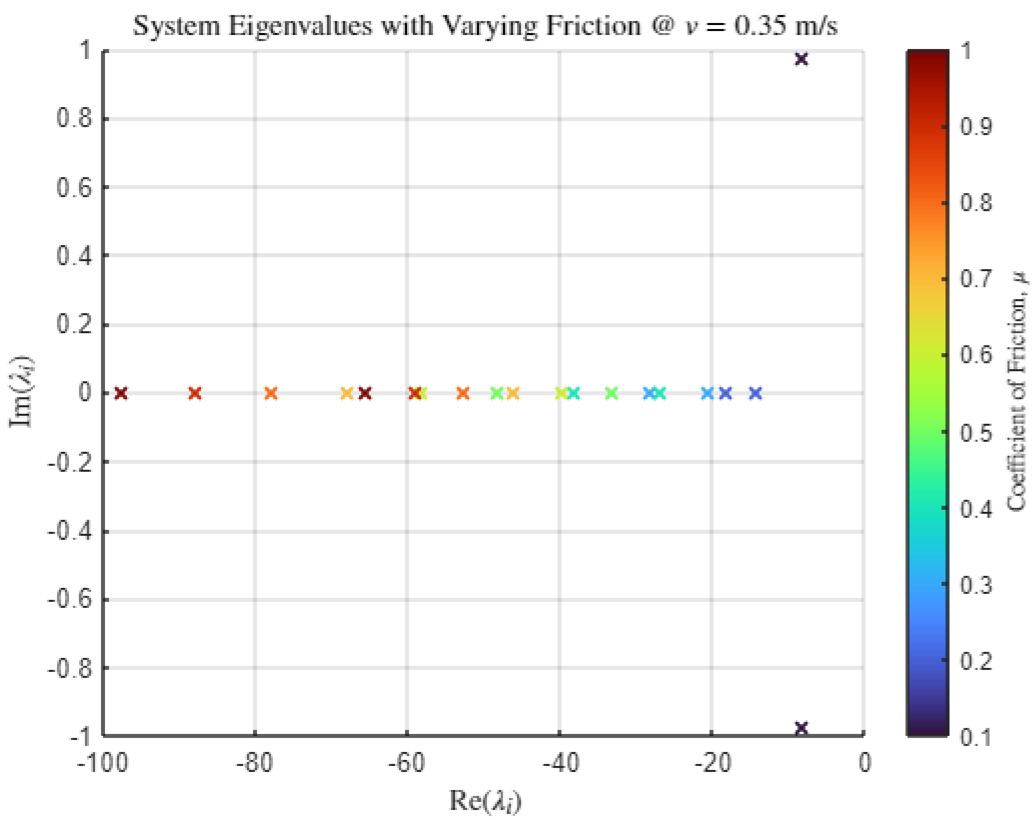
```
% System phase portrait

u_fl = 0.4; % Coefficient of friction for ground and FL tire interconnect
u_fr = 0.4; % Coefficient of friction for ground and FR tire interconnect
u_rl = 0.4; % Coefficient of friction for ground and RL tire interconnect
u_rr = 0.4; % Coefficient of friction for ground and RR tire interconnect
v = 0.35; % 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 @ $\mu = 0.4$, $v = 0.35$ m/s','Interpreter','latex');
grid on;
axis tight;
pbaspect([1 1 1])
```



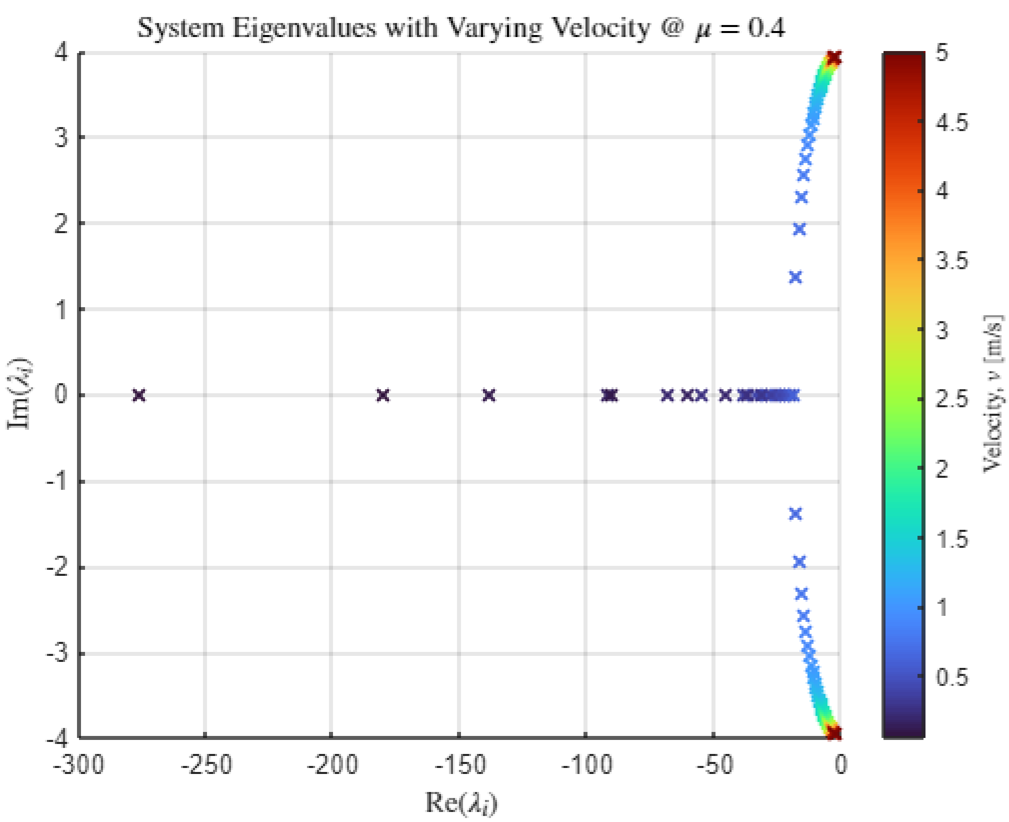

```
% System eignvalues with varying friction

EV = []; % Array to store system eigenvalues
v = 0.35; % 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 @ $v = 0.35$ m/
s','Interpreter','latex');
grid on;
```



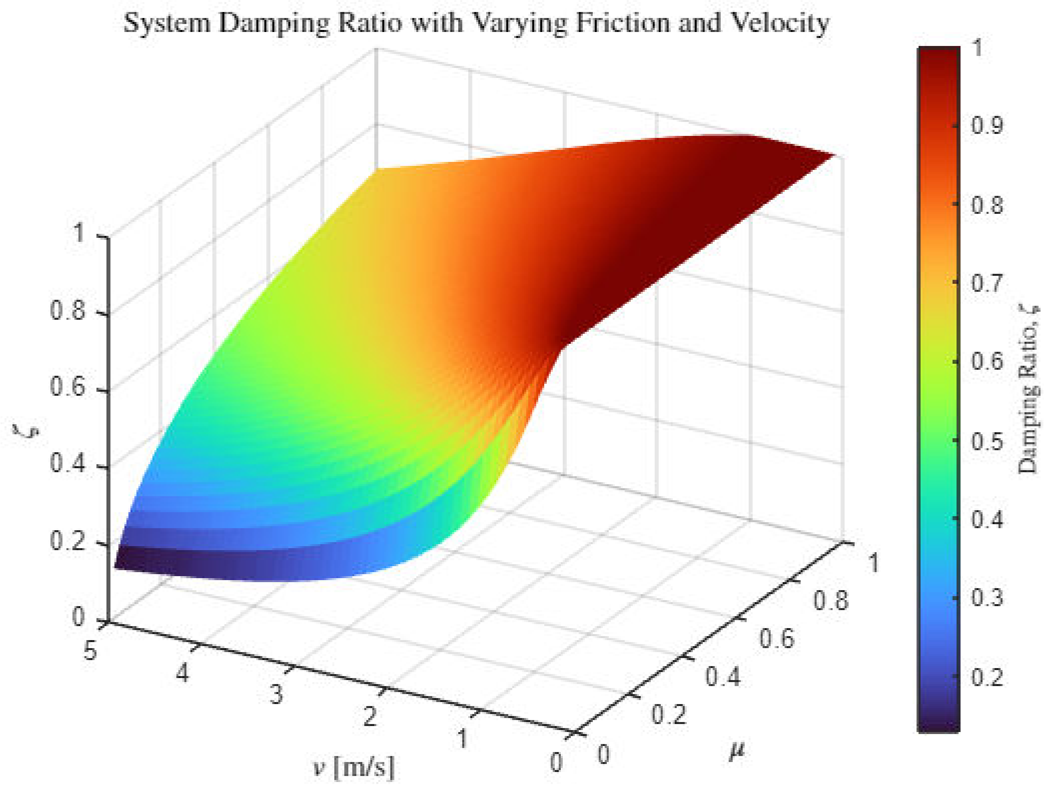

```
% System eignvalues with varying velocity

EV = []; % Array to store system eigenvalues
u = 0.4; % Coefficient of friction for ground and tire interconnect
for v=0.05:0.05:5
    % 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=0.05:0.05:5;
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 @ $\mu =
0.4$','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=0.1:0.1:5
        % 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=0.1:0.1:5;
[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]);
```

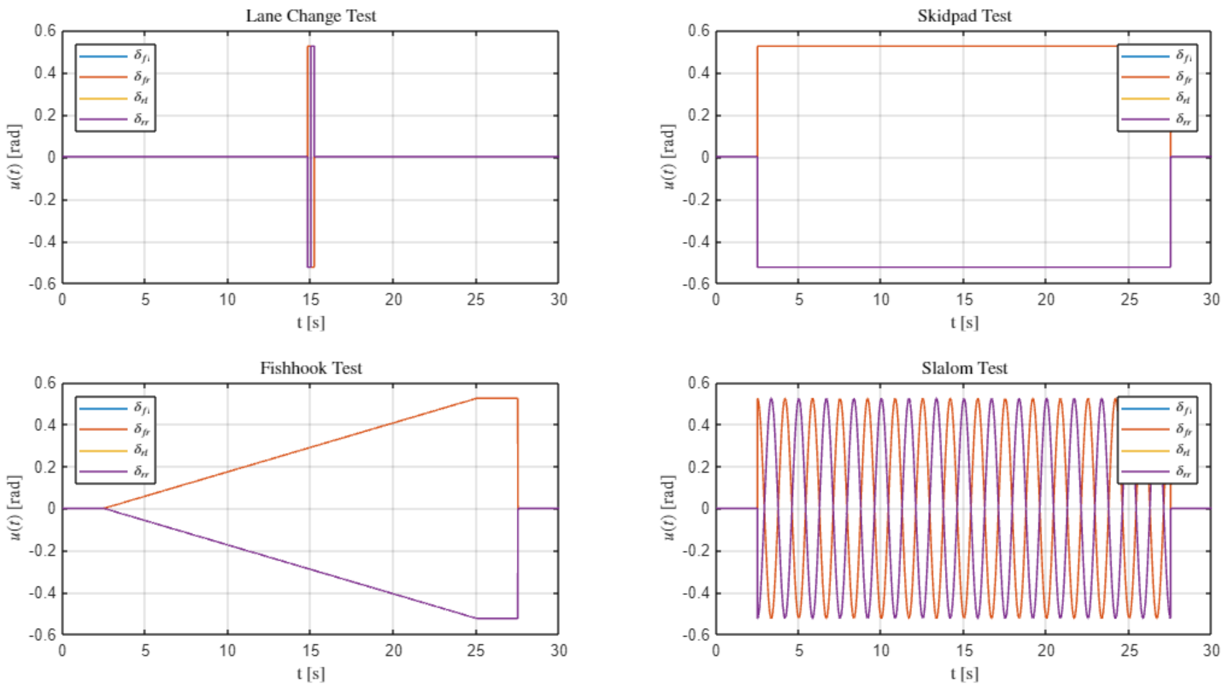


Model Simulation for Benckmark Maneuvers with Varying Friction

```
% 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;
```

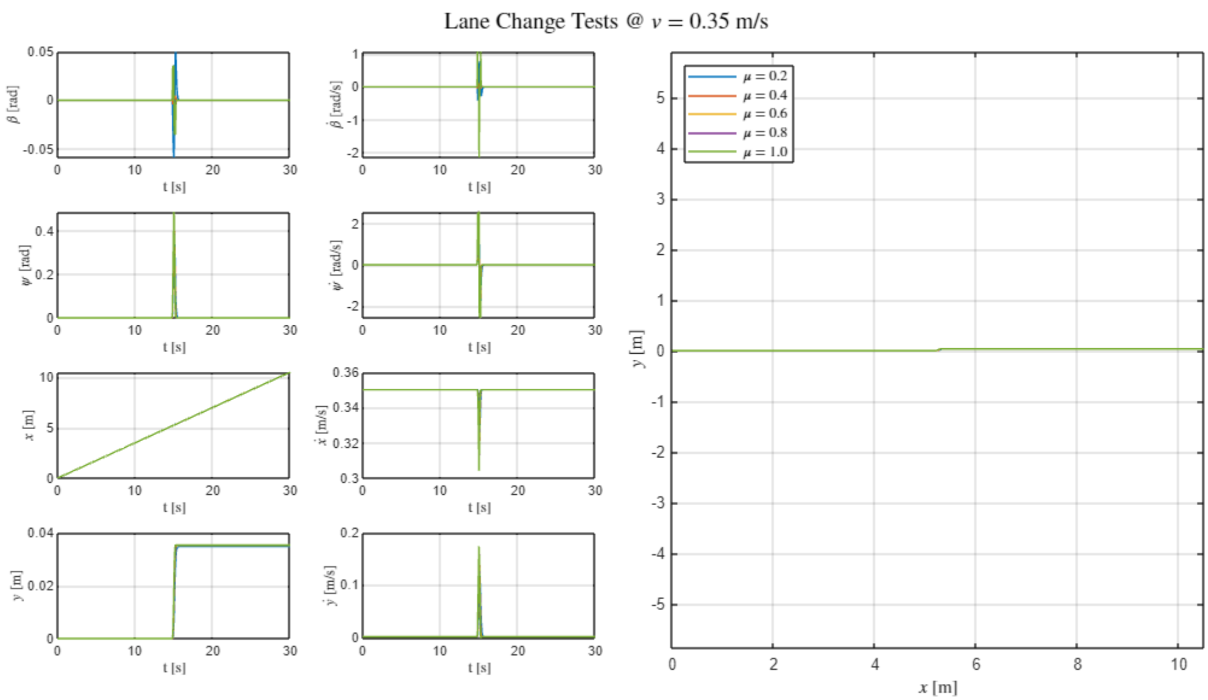



```
% Simulate lane change tests with varying friction

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 = 0.35; % 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 @ $v = 0.35$ m/s','Interpreter','latex')
legend('$\mu = 0.2$','$\mu = 0.4$','$\mu = 0.6$','$\mu = 0.8$','$\mu = 1.0$', ...
       'Interpreter','latex','Location','northwest')
```

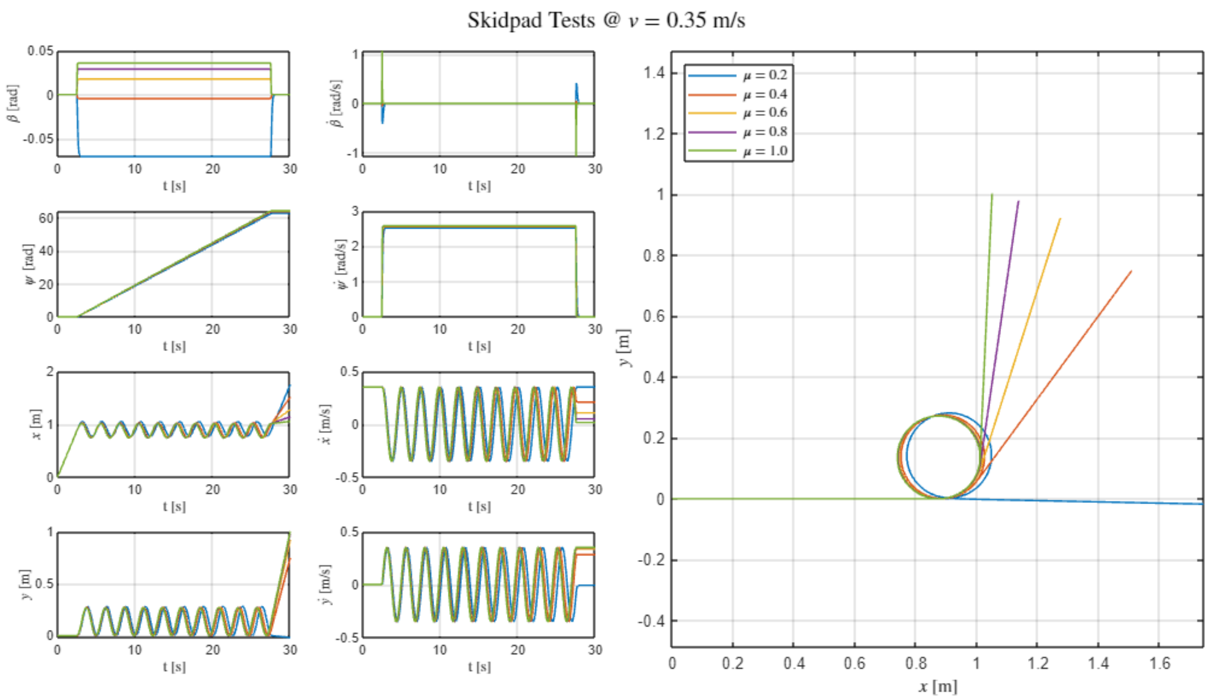



```
% Simulate skidpad tests with varying friction

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 = 0.35; % 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 @ $v = 0.35$ m/s','Interpreter','latex')
legend('$\mu = 0.2$','$\mu = 0.4$','$\mu = 0.6$','$\mu = 0.8$','$\mu = 1.0$', ...
       'Interpreter','latex','Location','northwest')
```

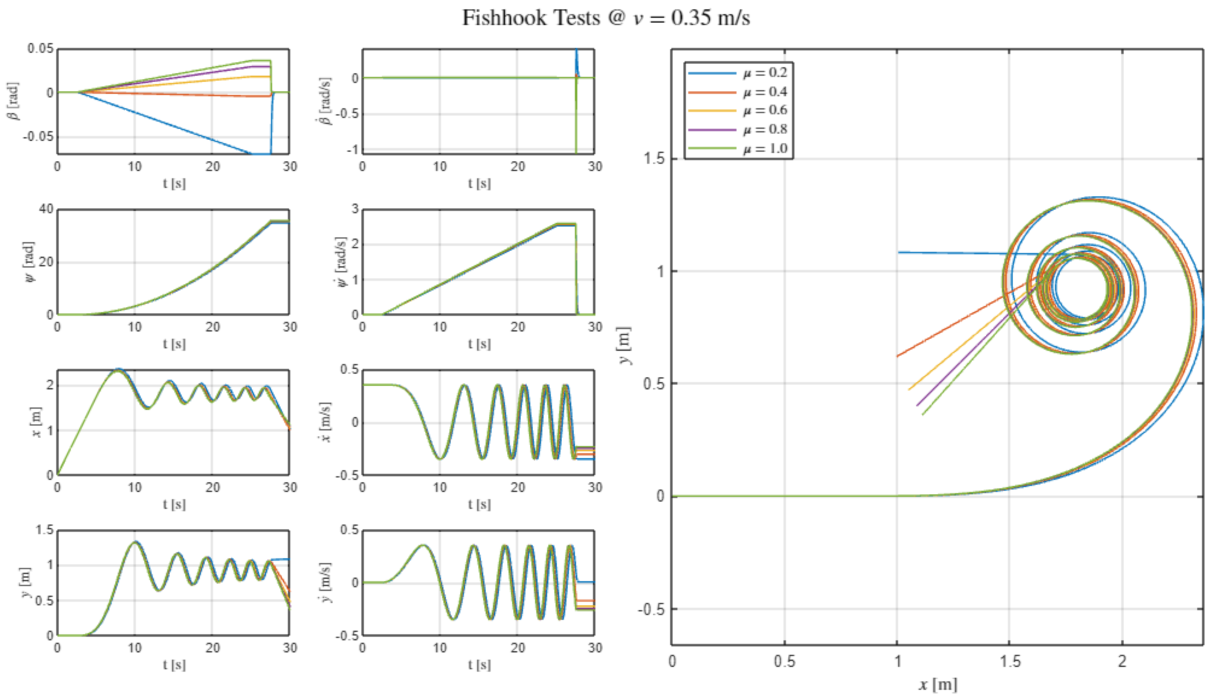



```
% Simulate fishhook tests with varying friction

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 = 0.35; % 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 @ $v = 0.35$ m/s','Interpreter','latex')
legend('$\mu = 0.2$','$\mu = 0.4$','$\mu = 0.6$','$\mu = 0.8$','$\mu = 1.0$', ...
       'Interpreter','latex','Location','northwest')
```

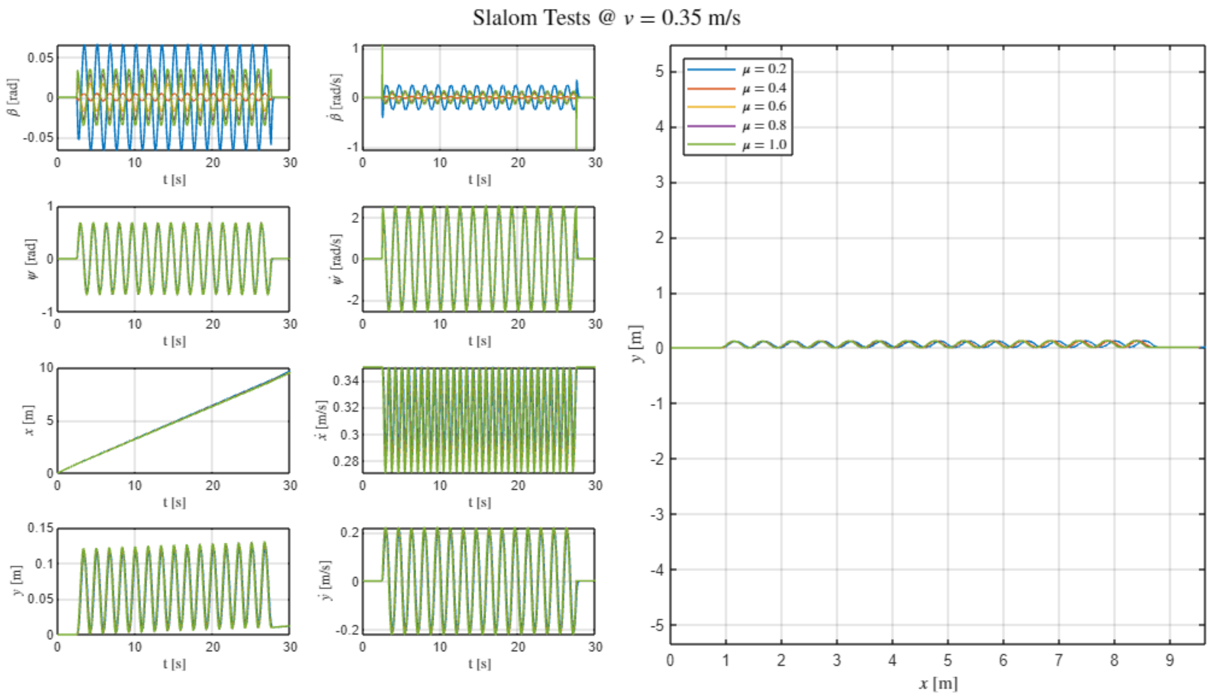



```
% Simulate slalom tests with varying friction

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 = 0.35; % 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 @ $v = 0.35$ 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 = 0.35; % 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)
```

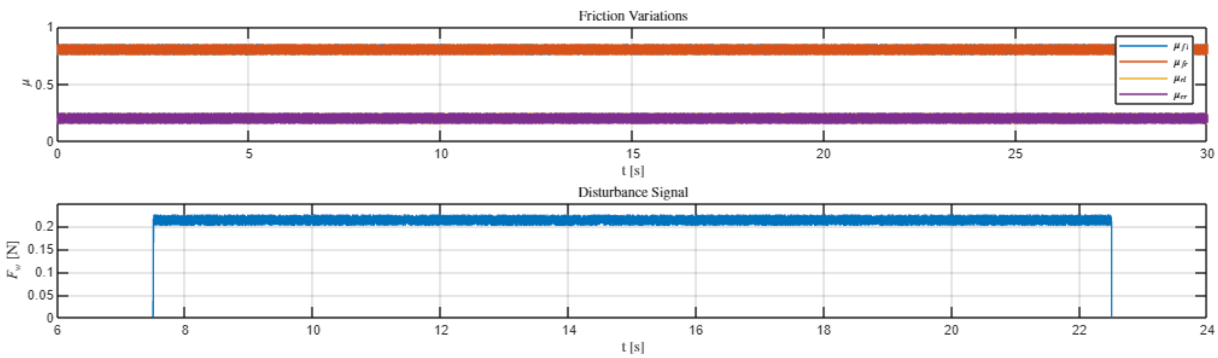
```
% Test robust stability of open-loop system via parametric Lyapunov functions
[tmin_ol, Q0_ol, Q1_ol, Q2_ol, Q3_ol] = pdlstab(S_ol, [0 0])
```

```
 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                        0.109182
     2                        0.011921
     3                        0.011921
     4                    7.406228e-03
     5                   -2.473266e-03

 Result:  best value of t: -2.473266e-03
          f-radius saturation:  0.000% of R =  1.00e+07

 This system is stable for the specified parameter trajectories
tmin_ol = -0.0025
Q0_ol = 2×2
    0.0130    0.0069
    0.0069    0.1507
Q1_ol = 2×2
   -0.0019   -0.0093
   -0.0093   -0.0419
Q2_ol = 2×2
   -0.0019   -0.0093
   -0.0093   -0.0419
Q3_ol = 2×2
   -0.0015    0.0064
    0.0064   -0.0335
```

```
% 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) 0.25*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')
```

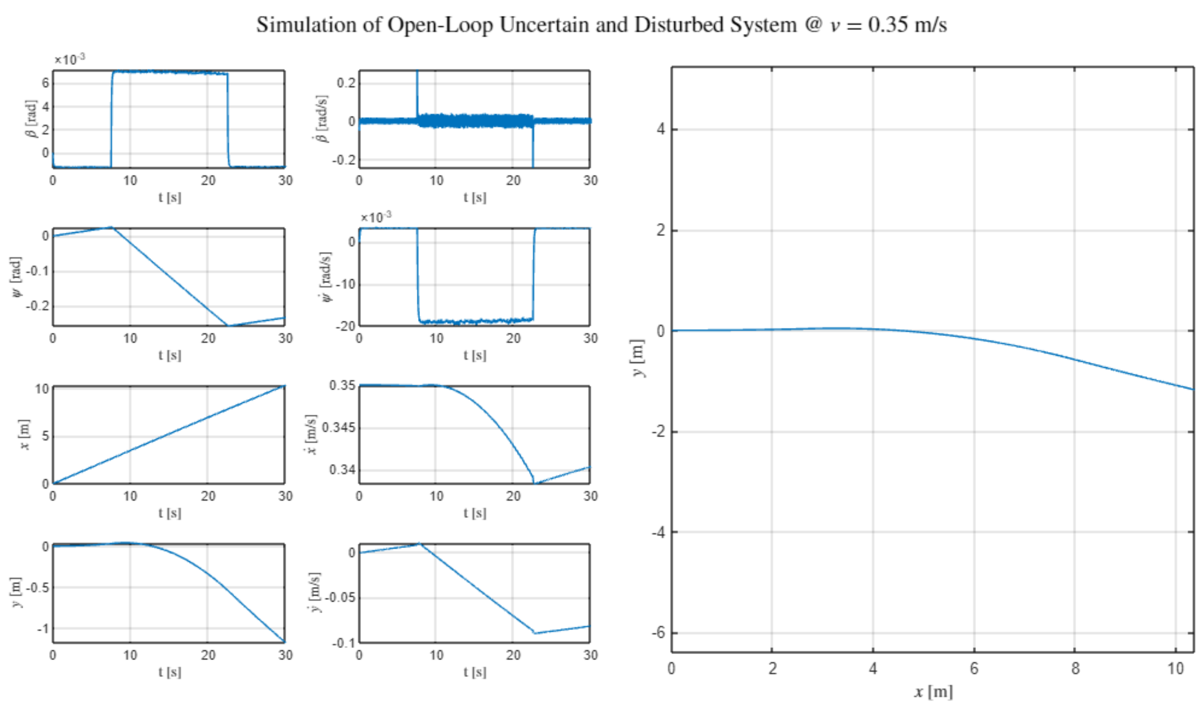
```
xlabel('t [s]','Interpreter','latex')
ylabel('$\mu$','Interpreter','latex')
ylim([0,1]);
title('Friction Variations','Interpreter','latex')
grid on;
hold off;
% Disturbance
subplot(2,1,2)
plot(t/1000,w_signal)
xlabel('t [s]','Interpreter','latex')
ylabel('$F_w$ [N]','Interpreter','latex')
ylim([0,0.25]);
title('Disturbance Signal','Interpreter','latex')
grid on;
```




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

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 0.35; % 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 @ $v = 0.35$ 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                  64.616755
     6                   5.458629
     7                   5.458629
     8                   2.721752
     9                   2.142491
    10                   1.533455
    11                   1.052437
    12                   0.809563
    13                   0.809563
    14                   0.289264
    15                   0.246906
    16                   0.246906
    17                   0.200706
    18                   0.200706
    19                   0.200706
    20                   0.171254
    21                   0.171254
    22                   0.152607
    23                   0.152607
    24                   0.152607
    25                   0.133582
    26                   0.133582
    27                   0.131517
    28                   0.131517
***                 new lower bound:     0.049831
    29                   0.122336
***                 new lower bound:     0.050383
    30                   0.122336
***                 new lower bound:     0.055419
    31                   0.122336
    32                   0.122052
    33                   0.122052
***                 new lower bound:     0.075447
    34                   0.122052
***                 new lower bound:     0.075735
    35                   0.122052
    36                   0.118958
    37                   0.118958
***                 new lower bound:     0.079319
    38                   0.118596
***                 new lower bound:     0.079973
    39                   0.118596
***                 new lower bound:     0.092650
```

```
    40                    0.118596
***                 new lower bound:     0.093388
    41                    0.117982
    42                    0.117982
***                 new lower bound:     0.098633
    43                    0.117695
***                 new lower bound:     0.099132
    44                    0.117695
***                 new lower bound:     0.107503
    45                    0.117695
***                 new lower bound:     0.107642
    46                    0.117492
    47                    0.117492
***                 new lower bound:     0.111006
    48                    0.117492
***                 new lower bound:     0.111161

 Result:  feasible solution
          best objective value:     0.117492
          guaranteed absolute accuracy:  6.33e-03
          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: 1.17e-01
g1_opt_global = 0.1175
```

```
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:0.01:0.5 % 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
    23
    24
    25
    26
    27
    28
    29
    30
    31
    32
    33
    34
    35
    36
    37
    38
    39
    40
    41
    42
    43
    44
    45
    46
    47
    48
    49
    50
    51
    52
    53
    54                   8.813650
    55                   8.813650
    56                   4.303078
    57                   2.487326
    58                   1.020421
    59                   1.020421
***                 new lower bound:    -7.646160
    60                   0.911127
    61                   0.911127
    62                   0.837689
    63                   0.806614
***                 new lower bound:    -1.542024
    64                   0.774720
***                 new lower bound:    -0.391795
    65                   0.774720
***                 new lower bound:    -0.153963
    66                   0.756838
***                 new lower bound:     0.658672
    67                   0.752805
***                 new lower bound:     0.743075
    68                   0.751527
***                 new lower bound:     0.748385

 Result:  feasible solution of required accuracy
```

```
         best objective value:     0.751527
         guaranteed absolute accuracy:  3.14e-03
         f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.17e-01
 Guaranteed   H2 performance: 8.67e-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
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21                   6.690681
    22                   6.690681
    23                   0.663545
    24                   0.663545
    25                   0.663545
    26                   0.663545
    27                   0.586590
    28                   0.586590
    29                   0.542272
    30                   0.505783
    31                   0.505783
    32                   0.479199
    33                   0.479199
***                 new lower bound:    -0.285505
    34                   0.474509
***                 new lower bound:    -0.168081
    35                   0.448355
***                 new lower bound:     0.397597
    36                   0.443542
***                 new lower bound:     0.429991
    37                   0.443542
    38                   0.443542
    39                   0.441352
***                 new lower bound:     0.438015

 Result:  feasible solution of required accuracy
         best objective value:     0.441352
         guaranteed absolute accuracy:  3.34e-03
         f-radius saturation:  0.000% of R =  1.00e+10
```

```
 Guaranteed Hinf performance: 1.27e-01
 Guaranteed   H2 performance: 6.64e-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
    11
    12
    13
    14
    15
    16
    17                   21.878313
    18                    4.756127
    19                    4.756127
    20                    1.940960
    21                    0.972253
    22                    0.972253
    23                    0.830431
    24                    0.830431
    25                    0.830431
    26                    0.637802
    27                    0.637802
    28                    0.589957
    29                    0.589957
***               new lower bound:    -0.725094
    30                    0.484738
***               new lower bound:    -0.523960
    31                    0.484738
***               new lower bound:    -0.463680
    32                    0.436130
***               new lower bound:    -0.315052
    33                    0.412175
***               new lower bound:    -0.274591
    34                    0.412175
***               new lower bound:    -0.235625
    35                    0.412175
***               new lower bound:    -0.075366
    36                    0.405913
***               new lower bound:     0.296524
    37                    0.392489
***               new lower bound:     0.366702
    38                    0.389130
***               new lower bound:     0.383351
    39                    0.388974
***               new lower bound:     0.387294

 Result:  feasible solution of required accuracy
          best objective value:     0.388974
          guaranteed absolute accuracy:  1.68e-03
```

```
         f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.37e-01
 Guaranteed   H2 performance: 6.24e-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
    11
    12
    13
    14
    15                  19.992215
    16                   4.355720
    17                   4.355720
    18                   0.743749
    19                   0.743749
    20                   0.743749
    21                   0.622811
    22                   0.622811
    23                   0.513826
    24                   0.513826
    25                   0.436682
    26                   0.436682
    27                   0.406329
***                 new lower bound:    -0.314782
    28                   0.406329
***                 new lower bound:    -0.270665
    29                   0.403780
***                 new lower bound:    -0.192041
    30                   0.403780
***                 new lower bound:    -0.061452
    31                   0.377833
***                 new lower bound:     0.254016
    32                   0.377833
    33                   0.364902
***                 new lower bound:     0.335221
    34                   0.364902
    35                   0.361781
***                 new lower bound:     0.358344

 Result:  feasible solution of required accuracy
          best objective value:     0.361781
          guaranteed absolute accuracy:  3.44e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.47e-01
 Guaranteed   H2 performance: 6.01e-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
    11
    12
    13                    5.766973
    14                    5.766973
    15                    3.130339
    16                    1.293096
    17                    0.678330
    18                    0.678330
    19                    0.678330
    20                    0.678330
    21                    0.508063
    22                    0.508063
    23                    0.432591
    24                    0.432591
***                 new lower bound:    -0.423078
    25                    0.371832
***                 new lower bound:    -0.276020
    26                    0.371832
***                 new lower bound:    -0.239429
    27                    0.371832
***                 new lower bound:    -0.122507
    28                    0.364677
***                 new lower bound:     0.224073
    29                    0.347980
***                 new lower bound:     0.313631
    30                    0.344385
***                 new lower bound:     0.335994
    31                    0.343661
***                 new lower bound:     0.341365

 Result:  feasible solution of required accuracy
          best objective value:     0.343661
          guaranteed absolute accuracy:  2.30e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.57e-01
 Guaranteed   H2 performance: 5.86e-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
    11
    12                    5.746616
    13                    5.746616
    14                    3.110982
    15                    1.242386
    16                    0.625016
    17                    0.625016
    18                    0.625016
    19                    0.625016
    20                    0.473473
    21                    0.473473
    22                    0.372400
    23                    0.372400
***               new lower bound:    -0.353637
    24                    0.372400
***               new lower bound:    -0.196505
    25                    0.362622
***               new lower bound:    -0.148435
    26                    0.350889
***               new lower bound:     0.247333
    27                    0.333768
    28                    0.332637
    29                    0.331982
***               new lower bound:     0.288946
    30                    0.331546
***               new lower bound:     0.308115
    31                    0.331546
***               new lower bound:     0.316782
    32                    0.331325
***               new lower bound:     0.328341

 Result:  feasible solution of required accuracy
          best objective value:     0.331325
          guaranteed absolute accuracy:  2.98e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.67e-01
 Guaranteed   H2 performance: 5.76e-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
    11
```

```
   12                   1.206922
   13                   1.206922
   14                   1.206922
   15                   0.795842
   16                   0.633401
   17                   0.633401
   18                   0.633401
   19                   0.558219
   20                   0.558219
   21                   0.546788
   22                   0.546788
   23                   0.356469
   24                   0.356469
   25                   0.356469
***                 new lower bound:    -0.170020
   26                   0.356469
***                 new lower bound:    -0.164250
   27                   0.349891
***                 new lower bound:    -0.149199
   28                   0.334256
***                 new lower bound:     0.261927
   29                   0.334256
   30                   0.323710
***                 new lower bound:     0.317256
   31                   0.323710
   32                   0.321918
***                 new lower bound:     0.320102

 Result:  feasible solution of required accuracy
          best objective value:     0.321918
          guaranteed absolute accuracy:  1.82e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.77e-01
 Guaranteed   H2 performance: 5.67e-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
   11                   5.488095
   12                   5.488095
   13                   0.699220
   14                   0.699220
   15                   0.699220
   16                   0.526167
   17                   0.526167
   18                   0.424470
   19                   0.424470
   20                   0.358418
   21                   0.358418
```

```
    22                   0.358418
    23                   0.336449
***              new lower bound:    -0.190743
    24                   0.336449
***              new lower bound:    -0.145551
    25                   0.333227
***              new lower bound:     0.208500
    26                   0.317583
    27                   0.316229
***              new lower bound:     0.294306
    28                   0.314977
***              new lower bound:     0.309564
    29                   0.314528
***              new lower bound:     0.313048

 Result:  feasible solution of required accuracy
          best objective value:     0.314528
          guaranteed absolute accuracy:  1.48e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.87e-01
 Guaranteed   H2 performance: 5.61e-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
    11                   5.397997
    12                   0.644754
    13                   0.644754
    14                   0.644754
    15                   0.458463
    16                   0.458463
    17                   0.378444
    18                   0.378444
    19                   0.378444
    20                   0.378444
    21                   0.378444
    22                   0.342511
    23                   0.342511
    24                   0.336419
***              new lower bound:     0.156162
    25                   0.317405
***              new lower bound:     0.266517
    26                   0.310262
    27                   0.309692
    28                   0.309142
***              new lower bound:     0.287444
    29                   0.308940
***              new lower bound:     0.292713
    30                   0.308940
```

```
***                 new lower bound:     0.301096
    31                   0.308854
***                 new lower bound:     0.307268

 Result:  feasible solution of required accuracy
          best objective value:     0.308854
          guaranteed absolute accuracy:  1.59e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 1.97e-01
 Guaranteed   H2 performance: 5.56e-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                   5.441992
    11                   5.441992
    12                   0.741771
    13                   0.741771
    14                   0.741771
    15                   0.566657
    16                   0.566657
    17                   0.392630
    18                   0.392630
    19                   0.360051
    20                   0.360051
    21                   0.330460
    22                   0.330460
***                 new lower bound:    -0.190587
    23                   0.330460
***                 new lower bound:    -0.134391
    24                   0.330460
***                 new lower bound:    -0.128891
    25                   0.329807
***                 new lower bound:    -0.096852
    26                   0.314601
***                 new lower bound:     0.246491
    27                   0.307011
***                 new lower bound:     0.290045
    28                   0.307011
    29                   0.304880
***                 new lower bound:     0.302816

 Result:  feasible solution of required accuracy
          best objective value:     0.304880
          guaranteed absolute accuracy:  2.06e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.07e-01
 Guaranteed   H2 performance: 5.52e-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                   5.357088
    11                   0.571191
    12                   0.571191
    13                   0.571191
    14                   0.571191
    15                   0.379356
    16                   0.379356
    17                   0.379356
    18                   0.376774
    19                   0.376774
    20                   0.321983
***                 new lower bound:     0.186733
    21                   0.307929
***                 new lower bound:     0.271060
    22                   0.307929
    23                   0.301707
***                 new lower bound:     0.293048
    24                   0.301707
    25                   0.300607
***                 new lower bound:     0.298374

 Result:  feasible solution of required accuracy
          best objective value:     0.300607
          guaranteed absolute accuracy:  2.23e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.17e-01
 Guaranteed   H2 performance: 5.48e-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                   5.303173
    11                   0.891700
    12                   0.891700
```

```
   13                    0.891700
   14                    0.429517
   15                    0.429517
   16                    0.384593
   17                    0.384593
   18                    0.326217
   19                    0.326217
   20                    0.326217
   21                    0.325782
***                 new lower bound:     0.147383
   22                    0.310355
***                 new lower bound:     0.242505
   23                    0.298353
   24                    0.298353
***                 new lower bound:     0.285637
   25                    0.298353
***                 new lower bound:     0.295718

 Result:  feasible solution of required accuracy
          best objective value:     0.298353
          guaranteed absolute accuracy:  2.63e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.27e-01
 Guaranteed   H2 performance: 5.46e-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                    5.277338
   11                    1.117180
   12                    1.117180
   13                    0.616078
   14                    0.616078
   15                    0.515604
   16                    0.515604
   17                    0.354624
   18                    0.354624
   19                    0.327342
   20                    0.327342
***                 new lower bound:    -0.235911
   21                    0.327342
***                 new lower bound:    -0.152722
   22                    0.327342
***                 new lower bound:    -0.146573
   23                    0.323778
***                 new lower bound:    -0.119251
   24                    0.323778
***                 new lower bound:     0.091726
   25                    0.301355
***                 new lower bound:     0.253594
```

```
    26                   0.301355
    27                   0.301355
    28                   0.297122
***                 new lower bound:     0.284563
    29                   0.297122
    30                   0.297122
    31                   0.294835
***                 new lower bound:     0.293410

 Result:  feasible solution of required accuracy
          best objective value:     0.294835
          guaranteed absolute accuracy:  1.42e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.37e-01
 Guaranteed   H2 performance: 5.43e-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                   5.269241
    11                   1.278793
    12                   1.278793
    13                   0.746219
    14                   0.746219
    15                   0.516149
    16                   0.516149
    17                   0.386284
    18                   0.354094
    19                   0.354094
    20                   0.326314
***                 new lower bound:    -0.267034
    21                   0.326314
***                 new lower bound:    -0.233233
    22                   0.326314
***                 new lower bound:    -0.152528
    23                   0.326314
***                 new lower bound:    -0.141258
    24                   0.317470
***                 new lower bound:    -0.108482
    25                   0.317470
***                 new lower bound: 6.299568e-03
    26                   0.304212
***                 new lower bound:     0.224269
    27                   0.293425
***                 new lower bound:     0.275955
    28                   0.292145
***                 new lower bound:     0.289369

 Result:  feasible solution of required accuracy
          best objective value:     0.292145
```

```
         guaranteed absolute accuracy:  2.78e-03
         f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.47e-01
 Guaranteed   H2 performance: 5.41e-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                  56.439812
    10                  16.765966
    11                   6.732069
    12                   2.172628
    13                   2.172628
    14                   0.933548
    15                   0.933548
    16                   0.933548
    17                   0.768574
    18                   0.768574
    19                   0.415204
    20                   0.415204
***                 new lower bound:    -0.605689
    21                   0.381927
***                 new lower bound:    -0.278211
    22                   0.381927
***                 new lower bound:    -0.249706
    23                   0.314494
***                 new lower bound:    -0.207621
    24                   0.314494
***                 new lower bound:    -0.181640
    25                   0.314494
***                 new lower bound:     0.145817
    26                   0.302218
    27                   0.294935
***                 new lower bound:     0.255879
    28                   0.290617
***                 new lower bound:     0.281735
    29                   0.290438
***                 new lower bound:     0.287841

 Result:  feasible solution of required accuracy
          best objective value:     0.290438
          guaranteed absolute accuracy:  2.60e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.57e-01
 Guaranteed   H2 performance: 5.39e-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                   5.350707
    10                   0.597270
    11                   0.597270
    12                   0.597270
    13                   0.597270
    14                   0.465943
    15                   0.404960
    16                   0.404960
    17                   0.360092
    18                   0.360092
    19                   0.332074
    20                   0.332074
***                 new lower bound:    -0.230244
    21                   0.310461
***                 new lower bound:    -0.146776
    22                   0.310461
***                 new lower bound:    -0.122614
    23                   0.310461
***                 new lower bound:     0.159333
    24                   0.303403
***                 new lower bound:     0.236337
    25                   0.290751
    26                   0.289239
    27                   0.289239
***                 new lower bound:     0.279682
    28                   0.288242
***                 new lower bound:     0.285504

 Result:  feasible solution of required accuracy
          best objective value:     0.288242
          guaranteed absolute accuracy:  2.74e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.67e-01
 Guaranteed   H2 performance: 5.37e-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                   5.325718
    10                   5.325718
```

```
   11                   1.149855
   12                   1.149855
   13                   0.802671
   14                   0.623550
   15                   0.623550
   16                   0.623550
   17                   0.565800
   18                   0.565800
   19                   0.389679
   20                   0.358979
   21                   0.358979
***                 new lower bound:    -0.395988
   22                   0.308738
***                 new lower bound:    -0.209787
   23                   0.308738
***                 new lower bound:    -0.185907
   24                   0.308738
***                 new lower bound:    -0.089342
   25                   0.304680
***                 new lower bound:     0.182499
   26                   0.290482
***                 new lower bound:     0.260396
   27                   0.288019
***                 new lower bound:     0.280113
   28                   0.288019
   29                   0.286992
***                 new lower bound:     0.284962

 Result:  feasible solution of required accuracy
          best objective value:     0.286992
          guaranteed absolute accuracy:  2.03e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.77e-01
 Guaranteed   H2 performance: 5.36e-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                   5.309449
   10                   5.309449
   11                   1.133781
   12                   1.133781
   13                   0.788412
   14                   0.609408
   15                   0.609408
   16                   0.609408
   17                   0.552951
   18                   0.552951
   19                   0.380687
   20                   0.350780
   21                   0.350780
```

```
***                 new lower bound:    -0.384089
    22                   0.325580
***                 new lower bound:    -0.202953
    23                   0.303591
***                 new lower bound:    -0.179142
    24                   0.303591
***                 new lower bound:    -0.157075
    25                   0.303591
***                 new lower bound:    -0.074399
    26                   0.300469
***                 new lower bound:     0.250377
    27                   0.287403
    28                   0.286120
    29                   0.286120
***                 new lower bound:     0.252638
    30                   0.286120
***                 new lower bound:     0.283581

 Result:  feasible solution of required accuracy
          best objective value:     0.286120
          guaranteed absolute accuracy:  2.54e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.87e-01
 Guaranteed   H2 performance: 5.35e-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                   5.300476
    10                   5.300476
    11                   1.090293
    12                   1.090293
    13                   0.747382
    14                   0.569362
    15                   0.569362
    16                   0.569362
    17                   0.569362
    18                   0.428780
    19                   0.428780
    20                   0.332256
    21                   0.332256
***                 new lower bound:    -0.292593
    22                   0.311150
***                 new lower bound:    -0.167451
    23                   0.311150
***                 new lower bound:    -0.141712
    24                   0.306979
***                 new lower bound:     0.155258
    25                   0.292744
    26                   0.288664
***                 new lower bound:     0.157820
```

```
   27                   0.287067
***               new lower bound:     0.175981
   28                   0.286661
***               new lower bound:     0.227249
   29                   0.286661
***               new lower bound:     0.267209
   30                   0.284750
***               new lower bound:     0.283076

 Result:  feasible solution of required accuracy
          best objective value:     0.284750
          guaranteed absolute accuracy:  1.67e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 2.97e-01
 Guaranteed   H2 performance: 5.34e-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                   5.296315
   10                   5.296315
   11                   1.039457
   12                   1.039457
   13                   0.698160
   14                   0.698160
   15                   0.485699
   16                   0.485699
   17                   0.331887
   18                   0.331887
   19                   0.306158
   20                   0.306158
***               new lower bound:    -0.232250
   21                   0.306158
***               new lower bound:    -0.138127
   22                   0.306158
***               new lower bound:    -0.125078
   23                   0.306158
***               new lower bound:    -0.085354
   24                   0.301493
***               new lower bound:     0.181066
   25                   0.287063
   26                   0.285629
***               new lower bound:     0.262735
   27                   0.283661
***               new lower bound:     0.278168
   28                   0.283202
***               new lower bound:     0.281703

 Result:  feasible solution of required accuracy
          best objective value:     0.283202
          guaranteed absolute accuracy:  1.50e-03
```

```
         f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.07e-01
 Guaranteed   H2 performance: 5.32e-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                  5.293456
    10                  5.293456
    11                  0.998243
    12                  0.998243
    13                  0.657561
    14                  0.657561
    15                  0.457603
    16                  0.457603
    17                  0.315448
    18                  0.315448
    19                  0.315448
    20                  0.315448
    21                  0.315448
    22                  0.308669
    23                  0.294014
***                 new lower bound:     0.219890
    24                  0.285819
    25                  0.285819
    26                  0.283433
***                 new lower bound:     0.270945
    27                  0.282690
***                 new lower bound:     0.281079

 Result:  feasible solution of required accuracy
          best objective value:     0.282690
          guaranteed absolute accuracy:  1.61e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.17e-01
 Guaranteed   H2 performance: 5.32e-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                   5.287749
    10                   5.287749
    11                   0.971918
    12                   0.971918
    13                   0.631805
    14                   0.631805
    15                   0.440237
    16                   0.440237
    17                   0.304838
    18                   0.304838
    19                   0.304838
    20                   0.304838
    21                   0.304838
    22                   0.304838
    23                   0.299131
***                 new lower bound:     0.241526
    24                   0.282832
    25                   0.282240
    26                   0.281793
***                 new lower bound:     0.277207
    27                   0.281793
    28                   0.281296
***                 new lower bound:     0.280124

 Result:  feasible solution of required accuracy
          best objective value:     0.281296
          guaranteed absolute accuracy:  1.17e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.27e-01
 Guaranteed   H2 performance: 5.30e-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                   5.275133
    10                   5.275133
    11                   0.956183
    12                   0.956183
    13                   0.618099
    14                   0.618099
    15                   0.431693
    16                   0.431693
    17                   0.299274
    18                   0.299274
    19                   0.299274
    20                   0.299274
    21                   0.299274
    22                   0.299274
    23                   0.296552
```

```
***                 new lower bound:     0.247087
   24                    0.281780
   25                    0.280800
   26                    0.280800
***                 new lower bound:     0.274572
   27                    0.280800
***                 new lower bound:     0.278296

 Result:  feasible solution of required accuracy
          best objective value:     0.280800
          guaranteed absolute accuracy:  2.50e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.37e-01
 Guaranteed   H2 performance: 5.30e-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                    5.252649
   10                    5.252649
   11                    0.942960
   12                    0.942960
   13                    0.610021
   14                    0.610021
   15                    0.427203
   16                    0.427203
   17                    0.296175
   18                    0.296175
   19                    0.296175
   20                    0.296175
   21                    0.296175
   22                    0.296175
   23                    0.295110
***                 new lower bound:     0.249027
   24                    0.280638
   25                    0.280216
   26                    0.279811
***                 new lower bound:     0.276728
   27                    0.279736
***                 new lower bound:     0.278866

 Result:  feasible solution of required accuracy
          best objective value:     0.279736
          guaranteed absolute accuracy:  8.70e-04
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.47e-01
 Guaranteed   H2 performance: 5.29e-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                   5.219047
    10                   5.219047
    11                   0.925107
    12                   0.925107
    13                   0.601228
    14                   0.601228
    15                   0.421917
    16                   0.421917
    17                   0.322250
    18                   0.322250
    19                   0.322250
    20                   0.322250
    21                   0.317687
***                 new lower bound:     0.104542
    22                   0.317687
    23                   0.287770
***                 new lower bound:     0.235486
    24                   0.287770
    25                   0.281751
***                 new lower bound:     0.268314
    26                   0.281751
    27                   0.281751
    28                   0.279742
***                 new lower bound:     0.278161

 Result:  feasible solution of required accuracy
          best objective value:     0.279742
          guaranteed absolute accuracy:  1.58e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.57e-01
 Guaranteed   H2 performance: 5.29e-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                   5.174277
    10                   5.174277
    11                   0.899070
```

```
   12                   0.899070
   13                   0.899070
   14                   0.503422
   15                   0.503422
   16                   0.346047
   17                   0.346047
   18                   0.317744
   19                   0.317744
   20                   0.317744
   21                   0.317744
   22                   0.314525
   23                   0.314525
***                 new lower bound:     0.068910
   24                   0.314525
***                 new lower bound:     0.080598
   25                   0.283114
***                 new lower bound:     0.238772
   26                   0.283114
   27                   0.279512
***                 new lower bound:     0.276856

 Result:  feasible solution of required accuracy
          best objective value:     0.279512
          guaranteed absolute accuracy:  2.66e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.67e-01
 Guaranteed   H2 performance: 5.29e-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                  5.118491
    10                  5.118491
    11                  2.987645
    12                  2.131077
    13                  1.645651
    14                  1.645651
    15                  1.645651
    16                  1.645651
    17                  1.503729
    18                  1.503729
    19                  1.503729
    20                  1.087154
    21                  1.087154
    22                  1.087154
    23                  0.988715
    24                  0.988715
    25                  0.568632
    26                  0.522442
***                 new lower bound:    -0.741054
    27                  0.440505
```

```
***                 new lower bound:    -0.668575
    28                   0.370759
***                 new lower bound:    -0.612918
    29                   0.340453
***                 new lower bound:    -0.563760
    30                   0.340453
***                 new lower bound:    -0.518707
    31                   0.314828
***                 new lower bound:    -0.205915
    32                   0.314828
***                 new lower bound:    -0.184169
    33                   0.299496
***                 new lower bound:    -0.120238
    34                   0.299496
***                 new lower bound:    -0.095521
    35                   0.294546
***                 new lower bound:     0.179888
    36                   0.281504
***                 new lower bound:     0.253441
    37                   0.278738
***                 new lower bound:     0.271841
    38                   0.278738
    39                   0.278160
***                 new lower bound:     0.276295

 Result:  feasible solution of required accuracy
          best objective value:     0.278160
          guaranteed absolute accuracy:  1.87e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.77e-01
 Guaranteed   H2 performance: 5.27e-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                   5.051525
    10                   5.051525
    11                   2.934265
    12                   2.079541
    13                   2.079541
    14                   0.864748
    15                   0.864748
    16                   0.596081
    17                   0.596081
    18                   0.366637
    19                   0.366637
    20                   0.366637
    21                   0.361749
***                 new lower bound:    -0.231332
    22                   0.361749
***                 new lower bound:    -0.052681
```

```
   23                  0.292280
***              new lower bound:     0.186009
   24                  0.282695
   25                  0.279558
   26                  0.279558
***              new lower bound:     0.197217
   27                  0.278705
***              new lower bound:     0.264727
   28                  0.277737
***              new lower bound:     0.275099

 Result:  feasible solution of required accuracy
          best objective value:     0.277737
          guaranteed absolute accuracy:  2.64e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.87e-01
 Guaranteed   H2 performance: 5.27e-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                  4.972957
   10                  4.972957
   11                  2.872541
   12                  2.019762
   13                  2.019762
   14                  0.831971
   15                  0.831971
   16                  0.697380
   17                  0.697380
   18                  0.426441
   19                  0.390358
   20                  0.390358
***              new lower bound:    -0.562760
   21                  0.390358
***              new lower bound:    -0.266292
   22                  0.348011
***              new lower bound:    -0.233584
   23                  0.324324
***              new lower bound:    -0.206216
   24                  0.324324
***              new lower bound:    -0.181535
   25                  0.295031
***              new lower bound:     0.117466
   26                  0.289105
***              new lower bound:     0.223285
   27                  0.280017
   28                  0.278817
   29                  0.278056
***              new lower bound:     0.239685
   30                  0.277853
```

```
***                 new lower bound:     0.252874
    31                   0.277853
***                 new lower bound:     0.268838
    32                   0.277165
***                 new lower bound:     0.275460

 Result:  feasible solution of required accuracy
          best objective value:     0.277165
          guaranteed absolute accuracy:  1.70e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 3.97e-01
 Guaranteed   H2 performance: 5.26e-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                   4.882299
    10                   4.882299
    11                   2.802051
    12                   1.951124
    13                   1.951124
    14                   1.164806
    15                   1.164806
    16                   0.824345
    17                   0.824345
    18                   0.552701
    19                   0.416372
    20                   0.416372
    21                   0.382491
***                 new lower bound:    -0.344127
    22                   0.382491
***                 new lower bound:    -0.310265
    23                   0.346908
***                 new lower bound:    -0.223337
    24                   0.346908
***                 new lower bound:    -0.199923
    25                   0.297758
***                 new lower bound:    -0.181722
    26                   0.297758
***                 new lower bound:    -0.156669
    27                   0.297758
***                 new lower bound:     0.153035
    28                   0.287284
***                 new lower bound:     0.227378
    29                   0.279061
    30                   0.277611
***                 new lower bound:     0.266832
    31                   0.276594
***                 new lower bound:     0.273967

 Result:  feasible solution of required accuracy
```

```
         best objective value:     0.276594
         guaranteed absolute accuracy:  2.63e-03
         f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.07e-01
 Guaranteed   H2 performance: 5.26e-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                  20.522156
    10                   4.983219
    11                   4.983219
    12                   1.812666
    13                   0.791272
    14                   0.791272
    15                   0.791272
    16                   0.593552
    17                   0.593552
    18                   0.367185
    19                   0.367185
    20                   0.337018
***                 new lower bound:    -0.261360
    21                   0.337018
***                 new lower bound:    -0.225821
    22                   0.292146
***                 new lower bound:    -0.163545
    23                   0.292146
***                 new lower bound:    -0.139643
    24                   0.292146
***                 new lower bound:     0.164380
    25                   0.285093
***                 new lower bound:     0.234657
    26                   0.277287
***                 new lower bound:     0.265305
    27                   0.276278
***                 new lower bound:     0.273654

 Result:  feasible solution of required accuracy
         best objective value:     0.276278
         guaranteed absolute accuracy:  2.62e-03
         f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.17e-01
 Guaranteed   H2 performance: 5.26e-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                 20.365134
    10                  5.184928
    11                  5.184928
    12                  1.970300
    13                  0.933385
    14                  0.933385
    15                  0.759952
    16                  0.759952
    17                  0.514488
    18                  0.429346
    19                  0.357559
***                new lower bound:    -0.673224
    20                  0.357559
***                new lower bound:    -0.597414
    21                  0.329073
***                new lower bound:    -0.239833
    22                  0.329073
***                new lower bound:    -0.216490
    23                  0.307964
***                new lower bound:    -0.163539
    24                  0.307964
***                new lower bound:    -0.141342
    25                  0.302017
***                new lower bound:     0.126889
    26                  0.302017
    27                  0.284377
***                new lower bound:     0.237228
    28                  0.277663
***                new lower bound:     0.266119
    29                  0.277663
    30                  0.276374
***                new lower bound:     0.274936

 Result:  feasible solution of required accuracy
          best objective value:     0.276374
          guaranteed absolute accuracy:  1.44e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.27e-01
 Guaranteed   H2 performance: 5.26e-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                  20.200108
    10                   5.422973
    11                   5.422973
    12                   2.149450
    13                   1.091906
    14                   1.091906
    15                   0.708330
    16                   0.708330
    17                   0.530126
    18                   0.441301
    19                   0.364660
    20                   0.364660
***                 new lower bound:    -0.694196
    21                   0.336178
***                 new lower bound:    -0.283347
    22                   0.336178
***                 new lower bound:    -0.256915
    23                   0.312174
***                 new lower bound:    -0.176648
    24                   0.312174
***                 new lower bound:    -0.155268
    25                   0.312174
***                 new lower bound:    -0.136357
    26                   0.307588
***                 new lower bound:     0.115658
    27                   0.307588
    28                   0.281454
***                 new lower bound:     0.237654
    29                   0.281454
    30                   0.277338
***                 new lower bound:     0.272475
    31                   0.277338
    32                   0.275652
***                 new lower bound:     0.274435

 Result:  feasible solution of required accuracy
          best objective value:     0.275652
          guaranteed absolute accuracy:  1.22e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.37e-01
 Guaranteed   H2 performance: 5.25e-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                  20.027703
    10                   5.699485
    11                   5.699485
    12                   2.351402
    13                   1.267784
```

```
    14                  1.267784
    15                  0.837332
    16                  0.837332
    17                  0.620285
    18                  0.515161
    19                  0.381842
    20                  0.381842
***                 new lower bound:    -0.824391
    21                  0.323714
***                 new lower bound:    -0.319381
    22                  0.323714
***                 new lower bound:    -0.290806
    23                  0.299913
***                 new lower bound:    -0.177688
    24                  0.299913
***                 new lower bound:    -0.157328
    25                  0.299913
***                 new lower bound:    -0.124416
    26                  0.299913
***                 new lower bound:    -0.096068
    27                  0.297821
***                 new lower bound:     0.145505
    28                  0.281986
    29                  0.281010
***                 new lower bound:     0.157622
    30                  0.281010
***                 new lower bound:     0.234179
    31                  0.275955
***                 new lower bound:     0.272832
    32                  0.275955
    33                  0.275235
***                 new lower bound:     0.274318

 Result:  feasible solution of required accuracy
          best objective value:     0.275235
          guaranteed absolute accuracy:  9.17e-04
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.47e-01
 Guaranteed   H2 performance: 5.25e-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                 19.848822
    10                  5.578587
    11                  5.578587
    12                  2.276763
    13                  1.208282
    14                  1.208282
    15                  0.787871
    16                  0.787871
```

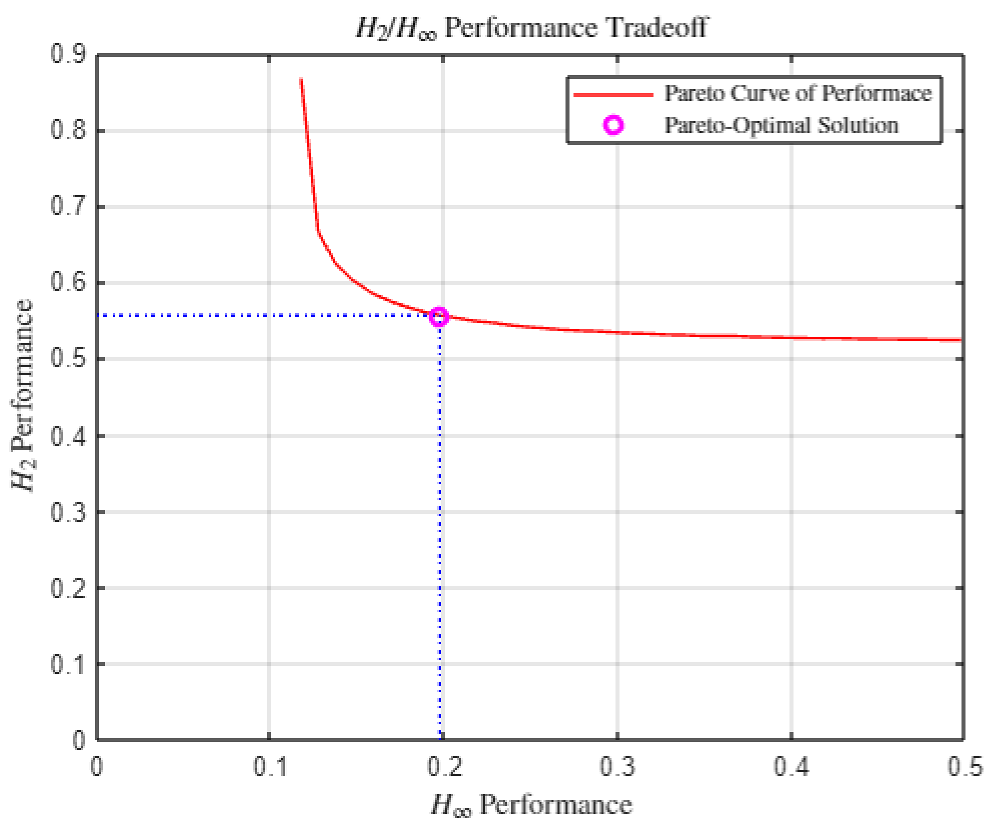
```
   17                  0.585324
   18                  0.486272
   19                  0.360308
   20                  0.360308
***                 new lower bound:    -0.780525
   21                  0.332258
***                 new lower bound:    -0.302471
   22                  0.332258
***                 new lower bound:    -0.274914
   23                  0.307974
***                 new lower bound:    -0.179294
   24                  0.307974
***                 new lower bound:    -0.158390
   25                  0.307974
***                 new lower bound:    -0.133933
   26                  0.306759
***                 new lower bound:    -0.111289
   27                  0.306759
***                 new lower bound: -3.684644e-04
   28                  0.286998
***                 new lower bound:     0.209284
   29                  0.277244
***                 new lower bound:     0.259144
   30                  0.275536
***                 new lower bound:     0.270868
   31                  0.275536
   32                  0.274986
***                 new lower bound:     0.273772

 Result:  feasible solution of required accuracy
          best objective value:     0.274986
          guaranteed absolute accuracy:  1.21e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.57e-01
 Guaranteed   H2 performance: 5.24e-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                19.664713
    10                 5.438906
    11                 5.438906
    12                 2.185182
    13                 1.132283
    14                 1.132283
    15                 0.925695
    16                 0.925695
    17                 0.684693
    18                 0.625741
    19                 0.470951
    20                 0.389203
```

```
***                 new lower bound:    -0.832996
    21                   0.353727
***                 new lower bound:    -0.130600
    22                   0.353727
    23                   0.328989
    24                   0.307694
    25                   0.307694
***                 new lower bound:    -0.124115
    26                   0.307694
***                 new lower bound:    -0.114729
    27                   0.298954
***                 new lower bound:     0.132954
    28                   0.298954
    29                   0.280551
***                 new lower bound:     0.237680
    30                   0.275151
***                 new lower bound:     0.265591
    31                   0.274767
***                 new lower bound:     0.272157

 Result:  feasible solution of required accuracy
          best objective value:     0.274767
          guaranteed absolute accuracy:  2.61e-03
          f-radius saturation:  0.000% of R =  1.00e+10

 Guaranteed Hinf performance: 4.67e-01
 Guaranteed   H2 performance: 5.24e-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                  19.477077
    10                   5.278553
    11                   5.278553
    12                   2.073804
    13                   2.073804
    14                   0.700033
    15                   0.700033
    16                   0.700033
    17                   0.531676
    18                   0.531676
    19                   0.359692
    20                   0.359692
    21                   0.329234
    22                   0.329234
***                 new lower bound...
```

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)
```

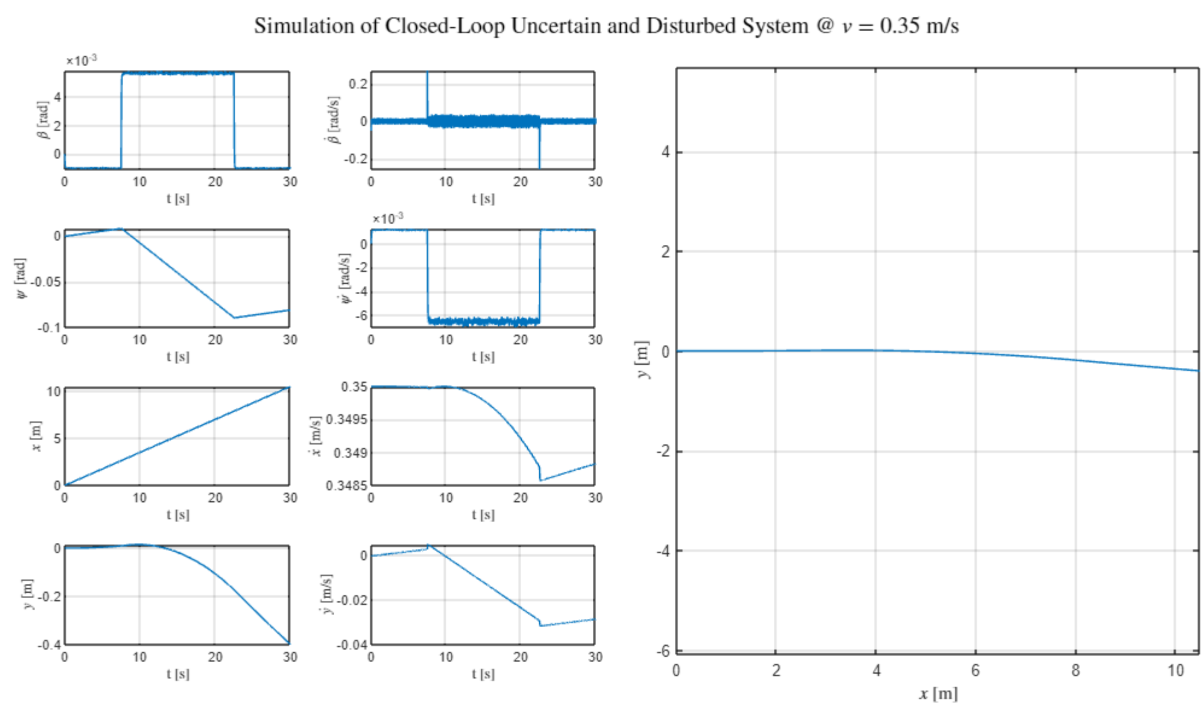
```
% Test robust stability of closed-loop system via parametric Lyapunov functions
[tmin_cl, Q0_cl, Q1_cl, Q2_cl, Q3_cl] = pdlstab(S_cl, [0 0])
```

```
 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                        0.225681
     2                        0.045101
     3                        0.040600
     4                    9.256315e-03
     5                    6.698021e-03
     6                    -8.031560e-03

 Result:  best value of t: -8.031560e-03
          f-radius saturation:  0.005% of R =  1.00e+07

 This system is stable in the specified parameter range
tmin_cl = -0.0080
Q0_cl = 2×2
    2.1291    0.3112
    0.3112    1.0034
Q1_cl = 2×2
    2.1291    0.3112
    0.3112    1.0034
Q2_cl = 2×2
    2.1291    0.3112
    0.3112    1.0034
Q3_cl = 2×2
    2.1291    0.3112
    0.3112    1.0034
```

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

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 0.35; % 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)')
```

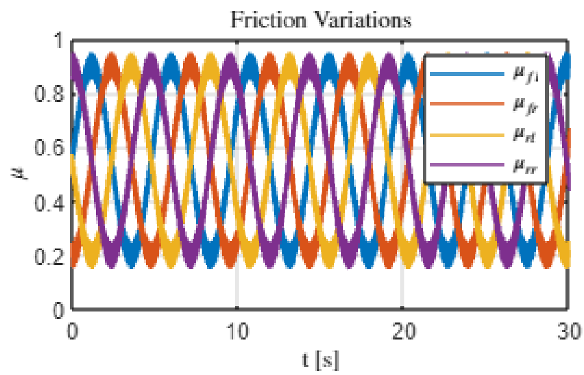
```
xlabel('t [s]','Interpreter','latex')
ylabel('$\dot{x}$ [m/s]','Interpreter','latex')
grid('on')
subplot(4,4,13)
plot(t/1000,PY(:,1:end-1)')
xlabel('t [s]','Interpreter','latex')
ylabel('$y$ [m]','Interpreter','latex')
grid('on')
subplot(4,4,14)
plot(t/1000,VY(:,1:end)')
xlabel('t [s]','Interpreter','latex')
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 @ $v = 0.35$ m/s','Interpreter','latex')
```

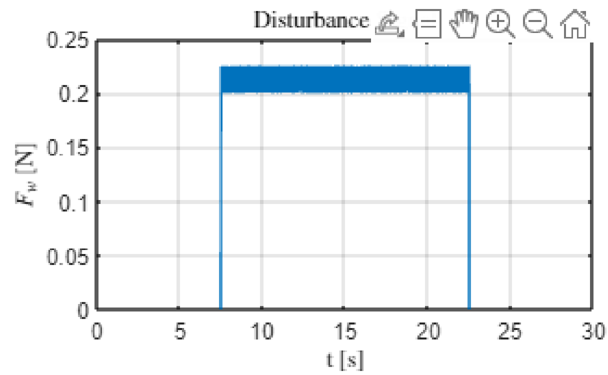


Open-Loop vs. Closed Loop System Performance

```
% 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) 0.25*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([0,0.25]);
title('Disturbance Signal','Interpreter','latex')
grid on;
```





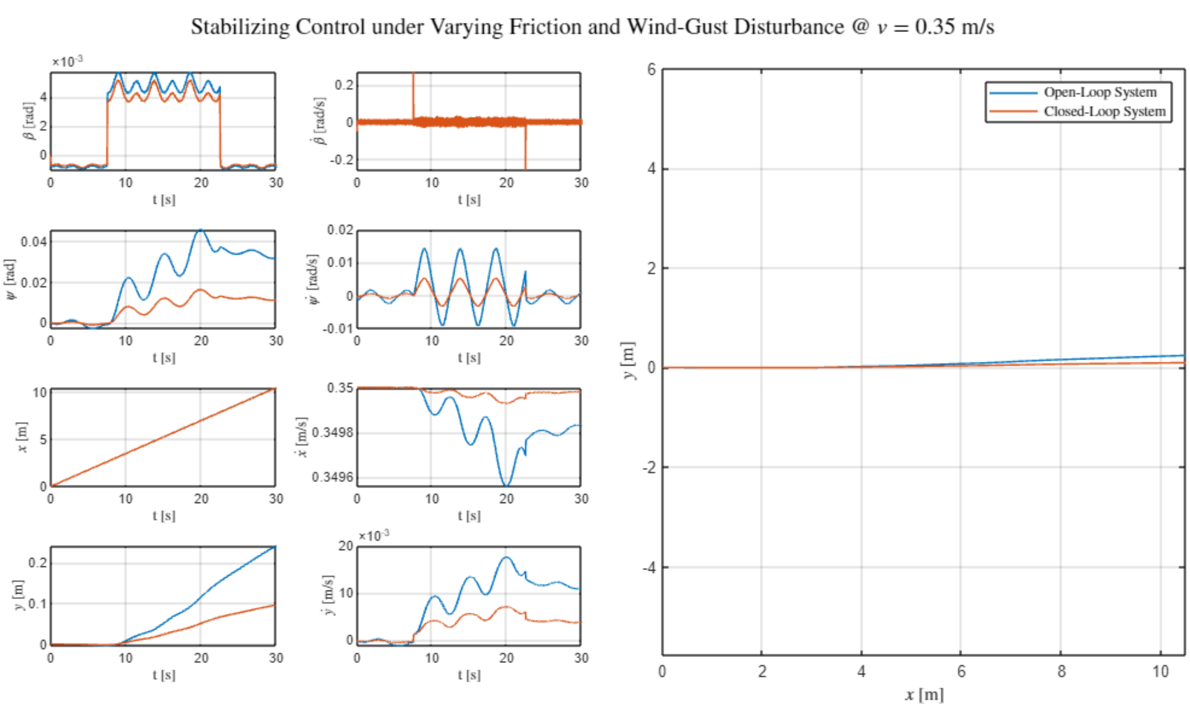
Stabilizing Controller

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

% Initialization
dt = 0.001; % Simulation timestep [s]
v = 0.35; % 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 = 0.35; % 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 @ $v
= 0.35$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', ...
       'Interpreter','latex','Location','northeast')
```



Tracking Controller

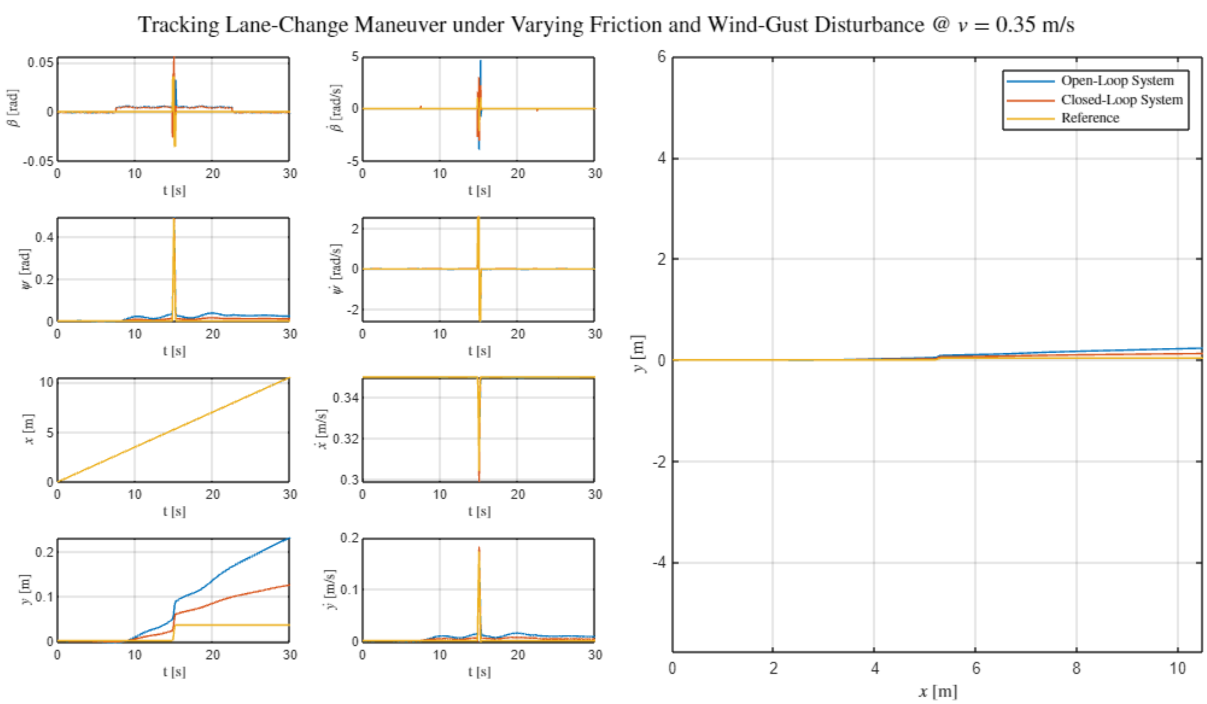
```
% Reference states (LC = lane-change, SP = skidpad, FH = fishhook, SL = slalom)
BETA_REF = BETA_LC(5,:);
BETA_DOT_REF = BETA_DOT_LC(5,:);
PSI_REF = PSI_LC(5,:);
PSI_DOT_REF = PSI_DOT_LC(5,:);
PX_REF = PX_LC(5,:);
```

```
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 = 0.35; % 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 = 0.35; % 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 @ $v = 0.35$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', 'Reference', ...
       'Interpreter','latex','Location','northeast')
```



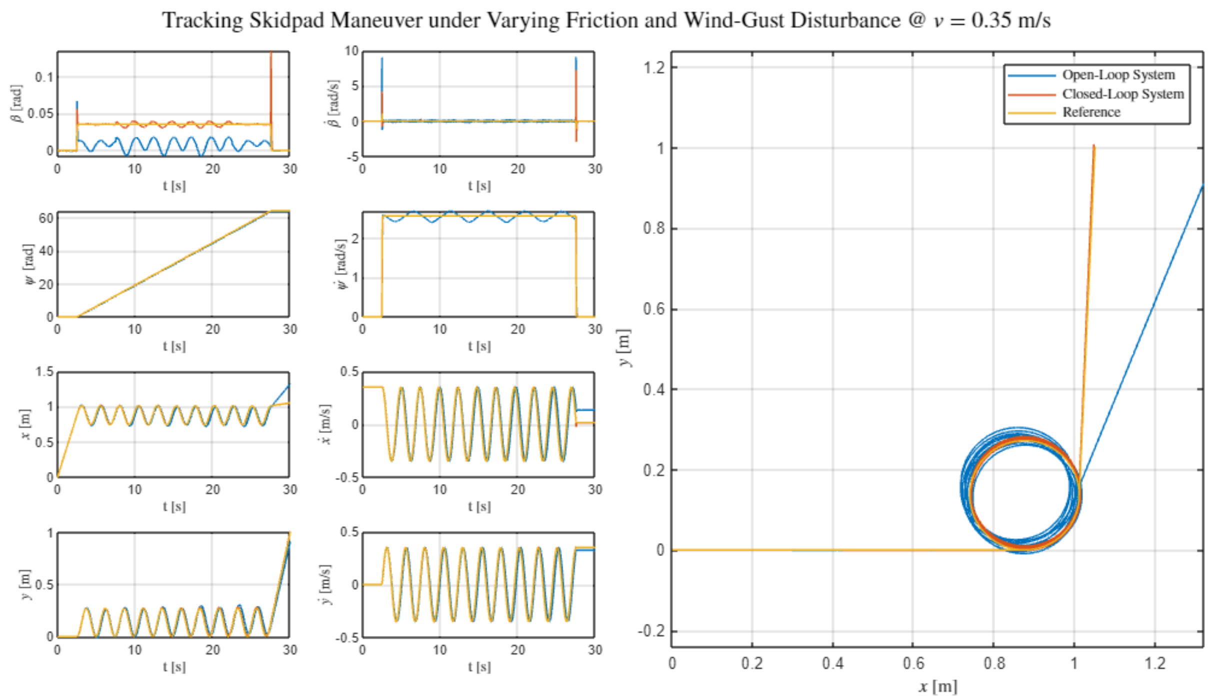

```
% Reference states (LC = lane-change, SP = skidpad, FH = fishhook, SL = slalom)
BETA_REF = BETA_SP(5,:);
BETA_DOT_REF = BETA_DOT_SP(5,:);
PSI_REF = PSI_SP(5,:);
PSI_DOT_REF = PSI_DOT_SP(5,:);
PX_REF = PX_SP(5,:);
```

```
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 = 0.35; % 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 = 0.35; % 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
@ $v = 0.35$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', 'Reference', ...
       'Interpreter','latex','Location','northeast')
```



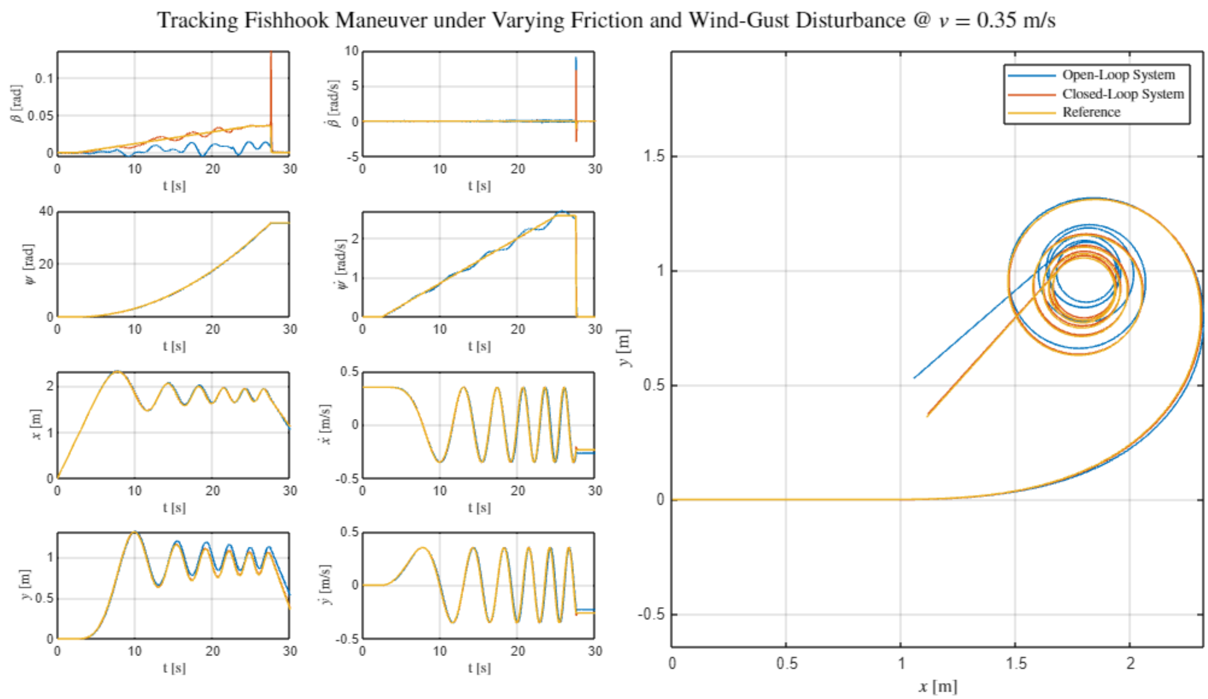

```
% Reference states (LC = lane-change, SP = skidpad, FH = fishhook, SL = slalom)
BETA_REF = BETA_FH(5,:);
BETA_DOT_REF = BETA_DOT_FH(5,:);
PSI_REF = PSI_FH(5,:);
PSI_DOT_REF = PSI_DOT_FH(5,:);
PX_REF = PX_FH(5,:);
```

```
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 = 0.35; % 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 = 0.35; % 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 @ $v = 0.35$ 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 = 0.35; % 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 = 0.35; % 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
@ $v = 0.35$ m/s','Interpreter','latex')
legend('Open-Loop System', 'Closed-Loop System', 'Reference', ...
       'Interpreter','latex','Location','northeast')
```