```
%Controller Gains
Kpc = 0.3; %wn^2/kr;
Kdc = 0.1; %(2*zeta*wn-kd)/kr;

theta_ref = deg2rad(-20); %In radians to use ode45
tspan = [0, 120]; %Set to random time

%Set initial state space conditions
% Recall x = [phi(angle error); omega(yaw rate)], rearrange angle error for theta so x = [theta; omega]
theta_ic = deg2rad(90); %Set to desired angle
omega_ic = 1; %Set to desired yaw
x0 = [theta_ic; omega_ic];

%Call ODE
[t, x] = ode45(@(t, x) sailboatode(t, x, Kpc, Kdc, theta_ref), tspan, x0);

%Get all values for theta and omega from the two x matrix columns
theta = x(:,1);
omega = x(:,2);

%Convert from rad to deg to graph
theta_deg = rad2deg(theta);
omega_deg = rad2deg(omega);
theta_ref_deg = rad2deg(theta_ref)*ones(size(t)); %Create ref angle vector

%Make graph
figure;
plot(t, theta_deg); hold on;
plot(t, theta_ref_deg, 'r--');
legend('\theta(t)', '\theta_{ref}', 'Location','best');
xlabel('Time (s)');
ylabel('Sailboat Angle \theta (degrees)');
title('Sailboat Angle Over Time');
grid on;
```

```
function dx = sailboatode(t, x, Kpc, Kdc, theta_ref )

%Assign x matrix columns
theta = x(1);
omega = x(2);

%PD controller input (rudder angle)
e = theta_ref - theta;
delta_r = Kpc*e - Kdc*omega;

%Set derivatives
d_theta = omega;
d_omega = delta_r;
dx = [d_theta; d_omega];
end
```