Table of Contents

MAE5803 - HW#4 Part 1 Sliding Controller

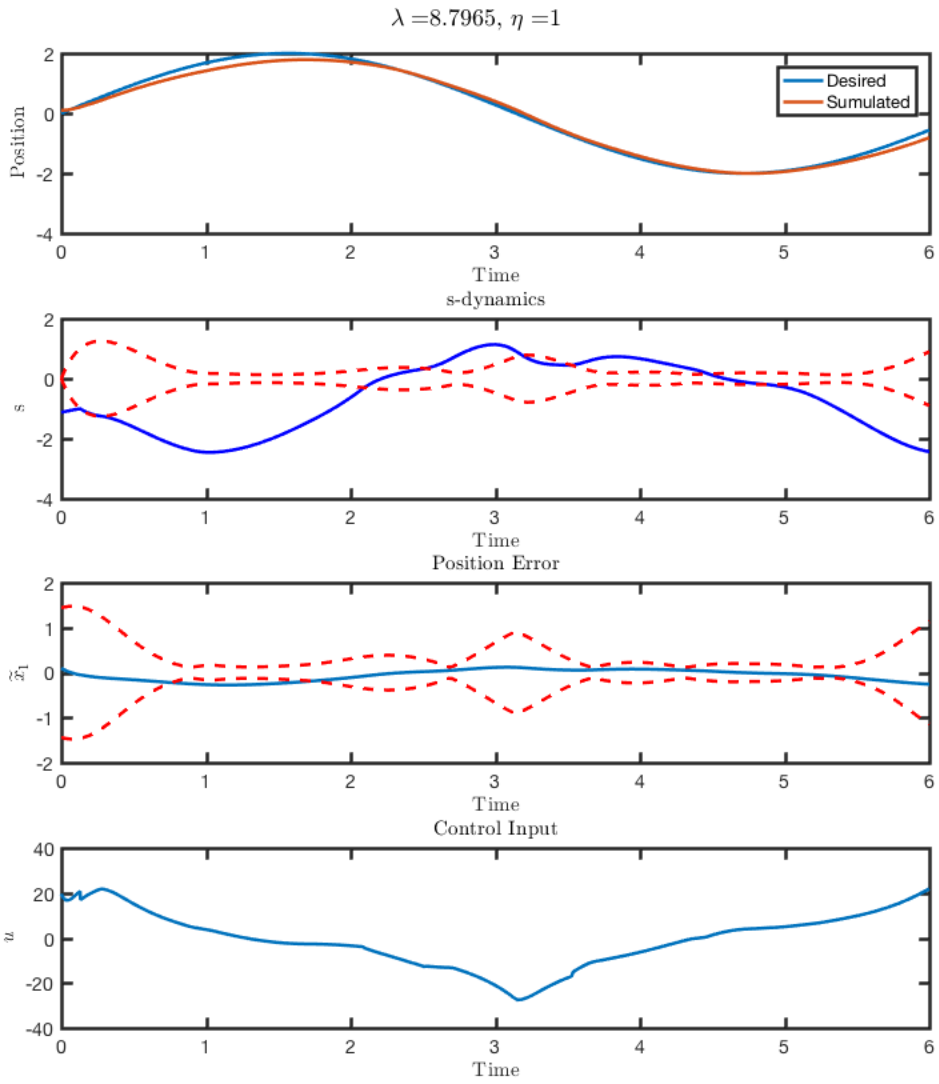
Time-vary mass, Integrate phi

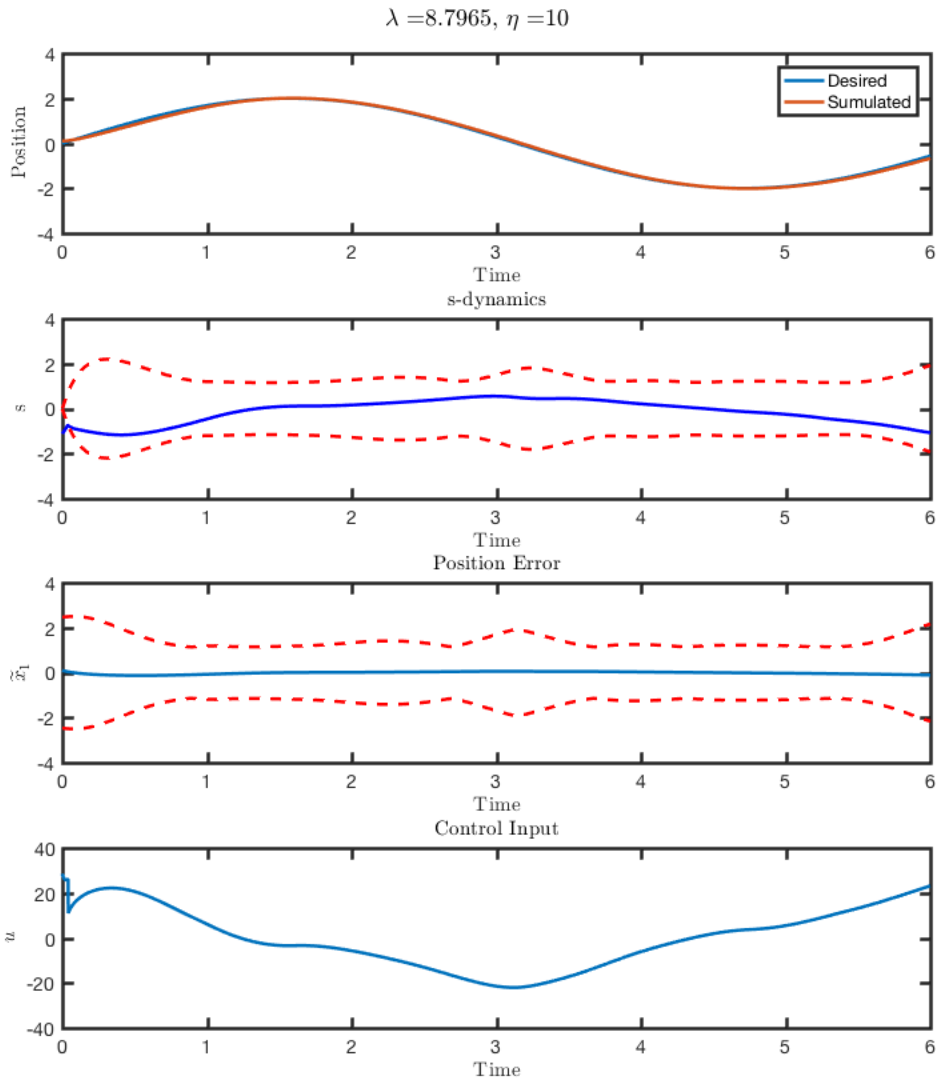
```
function HW4P1_ef()

% Given
n = 2;
m_lims = [1 2];
alpha1_lims = [4 6];
alpha2_lims = [1 2];
dlims = [-1 1];
omega_avoid = 4.2;      % (rad/s)

% Choose parameters
m_hat = mean(m_lims);
alpha1_hat = mean(alpha1_lims);
alpha2_hat = mean(alpha2_lims);
d_hat = mean(dlims);
lambda = [2*pi*omega_avoid/3 2*pi*omega_avoid/5];
eta = [1 10];

tspan = [0 6];
X0 = [0.1; 0; 0];
for i1 = 1:length(lambda)
    for i2 = 1:length(eta)

        casenum = 2*(i1-1) + i2;
        [t,X] = ode45(@slidingEOM_VaryMass,tspan,X0,
[],m_hat,alpha1_hat,alpha2_hat,d_hat,eta(i2),lambda(i1));
        x1 = X(:,1);
        xd = 2*sin(t);
        x1_tilde = x1 - xd;
        % Calc s and phi again (I haven't found a better way to do
 this yet)
        clear s phi u kd
        for i3 = 1:length(t)
            [~,s(i3),phi(i3),u(i3),kd(i3)] =
 slidingEOM_VaryMass(t(i3),X(i3,:),m_hat,alpha1_hat,alpha2_hat,d_hat,eta(i2),lambd
        end
        x_tilde_bound = kd/lambda(i1);
```

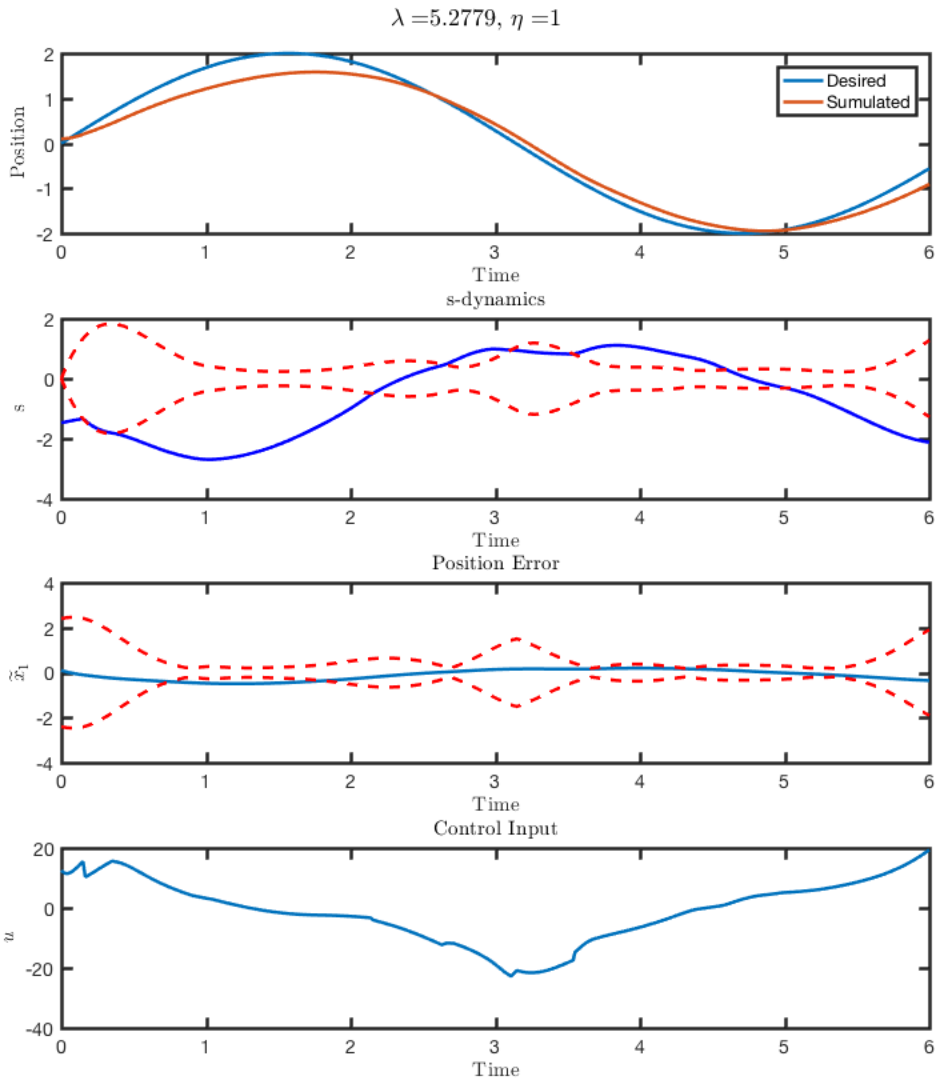
Plots

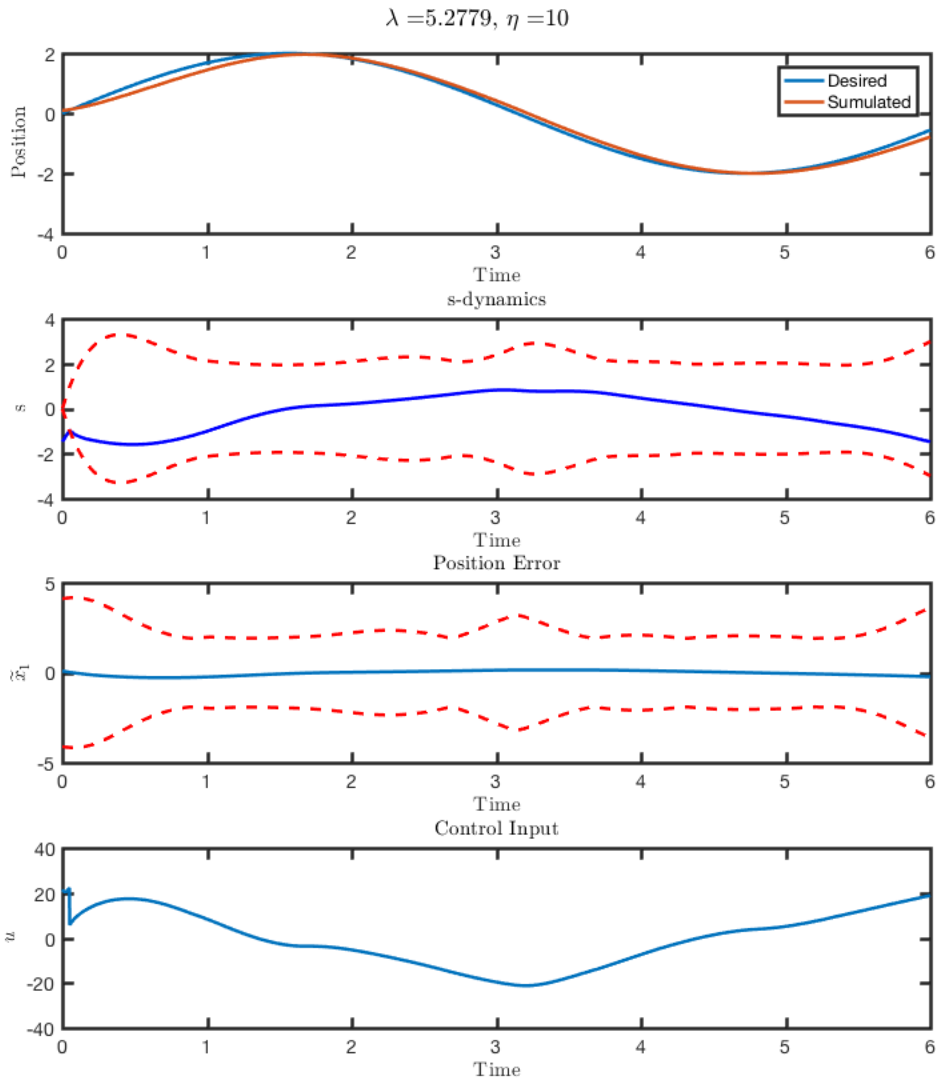
```
        fh = figure(casenum);
```

```
        set(fh,'Position',[0 0 840 1050])
        suptitle(['$\lambda = $' num2str(lambda(i1)) ', $\eta = $'
num2str(eta(i2))]);
        % Dynamics
        subplot(411)
        plot(t,xd,t,x1)
        legend('Desired','Sumulated','location','northeast')
        xlabel('Time'); ylabel('Position');

        % s-Dynamics
        subplot(412)
        plot(t,s,'b',t,phi,'r--',t,-phi,'r--')
        title('s-dynamics')
        xlabel('Time'); ylabel('s');

        % Error Plot
        subplot(413)
        plot(t,x1_tilde,t,x_tilde_bound,'r--',t,-x_tilde_bound,'r--');
        title('Position Error')
        xlabel('Time'); ylabel('$\widetilde{x}_1$');

        % Control Input
        subplot(414)
        plot(t,u);
        title('Control Input')
        xlabel('Time'); ylabel('$u$');
```


λ =8.7965, η =1
Desired
Sumulated
Position
Time
s-dynamics
s
Time
Position Error
$\tilde{x}_1$
Time
Control Input
u
Time


λ =8.7965, η =10
Desired
Sumulated
Position
Time
s-dynamics
s
Time
Position Error
$\tilde{x}_1$
Time
Control Input
u
Time


λ =5.2779, η =1
Position
Desired
Sumulated
Time
s-dynamics
s
Time
Position Error
$\tilde{x}_1$
Time
Control Input
u
Time




```
    end
end
end
```

Equation of Motion

```
function [dx, s, phi, u, kd] =
 slidingEOM_VaryMass(t,x,m_hat,a1_hat,a2_hat,d_hat,eta,lambda)
dx = zeros(size(x));
x1 = x(1);
x2 = x(2);
phi = x(3);
xd = 2*sin(t);
xd_dot = 2*cos(t);
```

```
xd_dd = -2*sin(t);

m = 2 - abs(cos(1.5*t));
a1 = 5 + cos(t);
a2 = 1 + abs(sin(2*t));
d = cos(1.3*t);
f = (-1/m)*((a1 + a2*cos(x1)^2)*abs(x2)*x2 + d);
fd = (-1/m)*((a1 + a2*cos(xd)^2)*abs(xd_dot)*xd_dot + d);
b = 1/m;
f_hat = (-1/m_hat)*((a1_hat + a2_hat*cos(x1)^2)*abs(x2)*x2 + d_hat);
fd_hat = (-1/m_hat)*((a1_hat + a2_hat*cos(xd)^2)*abs(xd_dot)*xd_dot +
 d_hat);
F = abs(f-f_hat);
Fd = abs(fd-fd_hat);
k = F + eta;
kd = Fd + eta;
% phi = 0.1;
% phi = kd/lambda;
% phi_dot = 0;
phi_dot = kd - lambda*phi;
s = x2 - xd_dot + lambda*(x1-xd);
u_hat = -f_hat + xd_dd - lambda*(x2-xd_dot);

if abs(s) >= phi
    u = u_hat - k*sign(s);
else
    u = u_hat - (k-phi_dot)*(s/phi);
end

dx(1) = x2;
dx(2) = f + b*u;
dx(3) = phi_dot;

end
```

Published with MATLAB® R2016a