ANIMESH NEMA - HOMEWORK-6

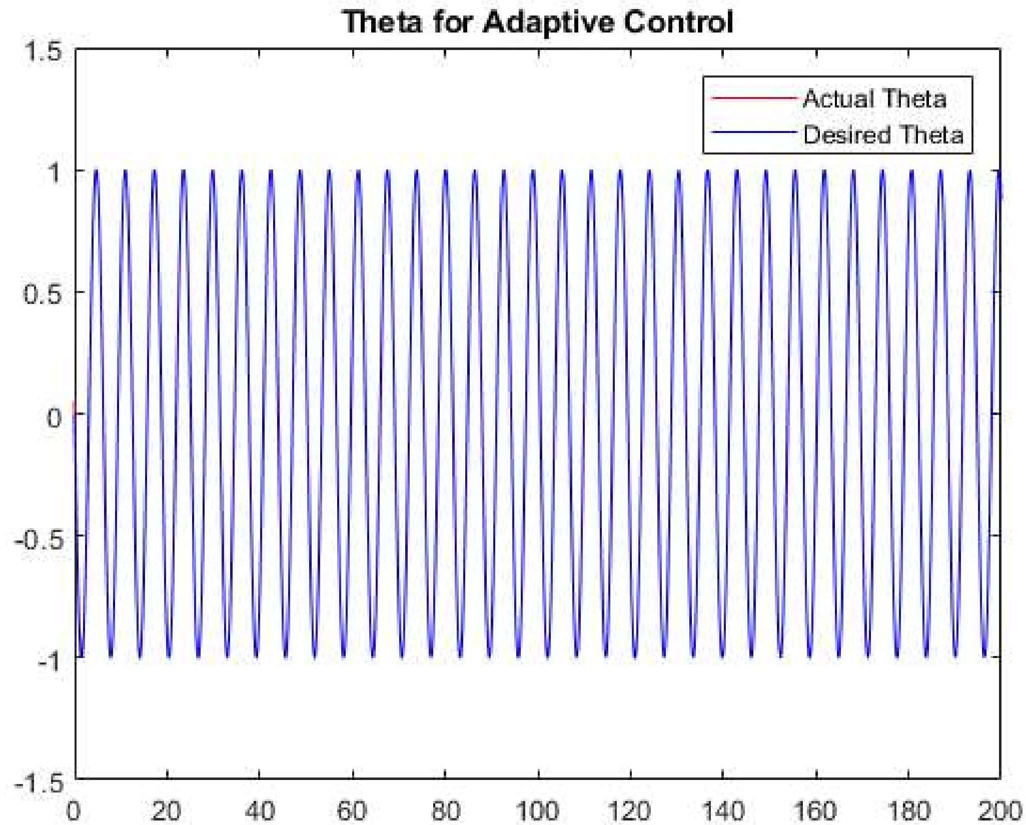
Contents

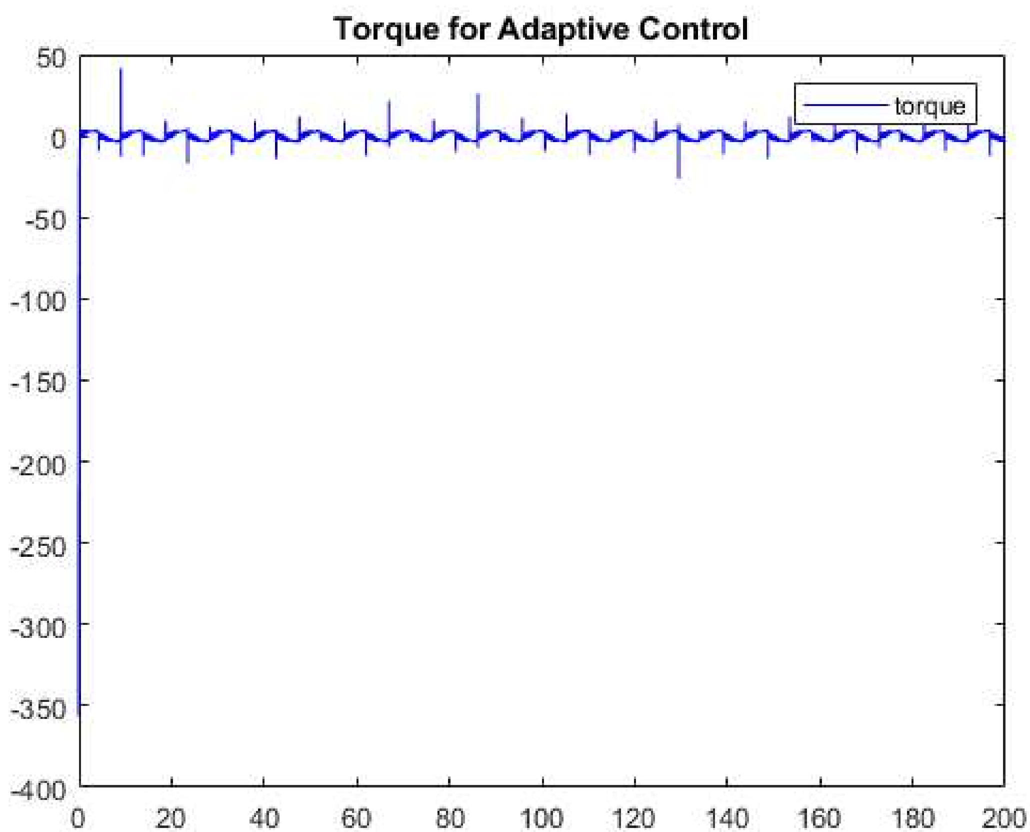
PASSIVITY BASED ADAPTIVE CONTROL

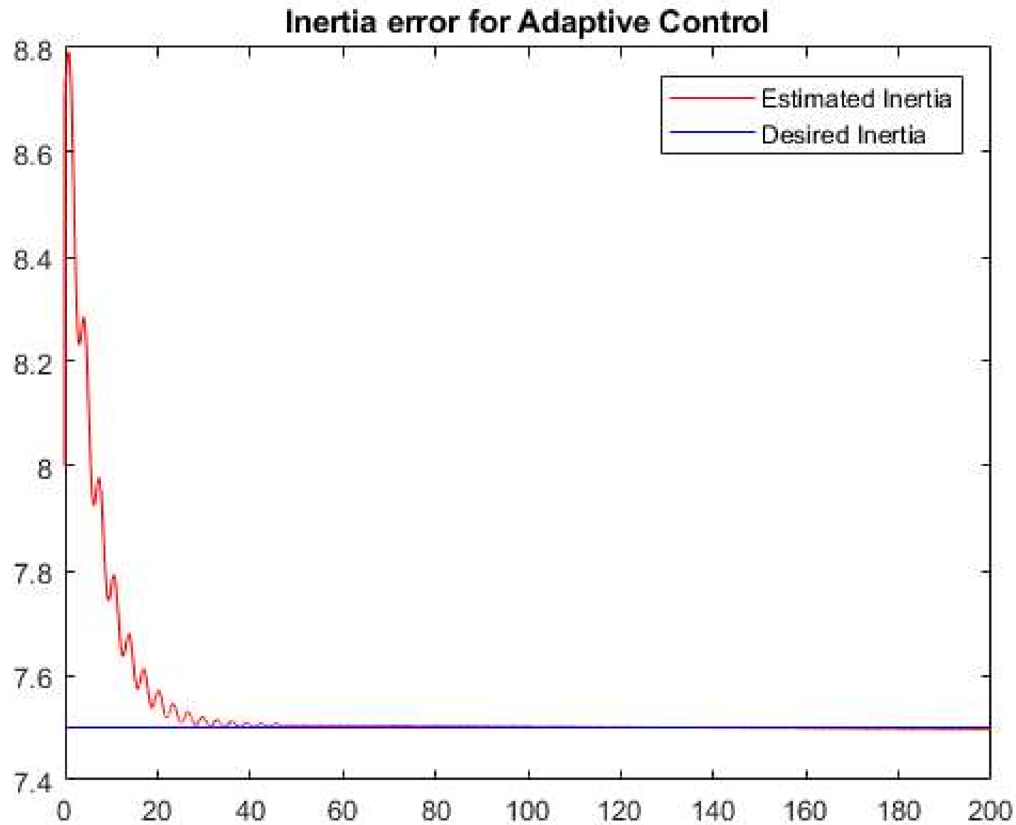
```
clc; clear all; close all
% the following parameters for the arm
I = 7.5;
mgd = 6;
fv =1.5;

%time steps
tf = 200;

global torque
torque =[];

%initial condition
x0 = [0.05,0.1,8,2.5,5];

% Implement the Passivity based Adaptive control.
%options = odeset('RelTol',1e-4,'AbsTol',[1e-4, 1e-4, 1e-4, 1e-4]);
[T,X] = ode45(@(t,x) planarArmODEadaptive(t,x),[0 tf],x0);

figure('Name','Theta for Adaptive control');
plot(T, X(:,1),'r-');
hold on
plot(T, -sin(T), 'b-');
legend('Actual Theta', 'Desired Theta')
title('Theta for Adaptive Control');

figure('Name', 'I/p- Torque for Adaptive control')
plot(T, torque(1,1:size(T,1)), 'b-');
legend('torque')
title('Torque for Adaptive Control');

figure('Name','Inertia error');
plot(T, X(:,3),'r-');
hold on
plot(T, 7.5*ones(size(T,1),1),'b-');
legend('Estimated Inertia', 'Desired Inertia')
title('Inertia error for Adaptive Control');


figure('Name','Force error');
plot(T, X(:,4),'r-');
hold on
plot(T, 1.5*ones(size(T,1),1),'b-');
legend('Estimated Force', 'Desired Force')
title('Force error for Adaptive Control');

figure('Name','Gravity error');
```

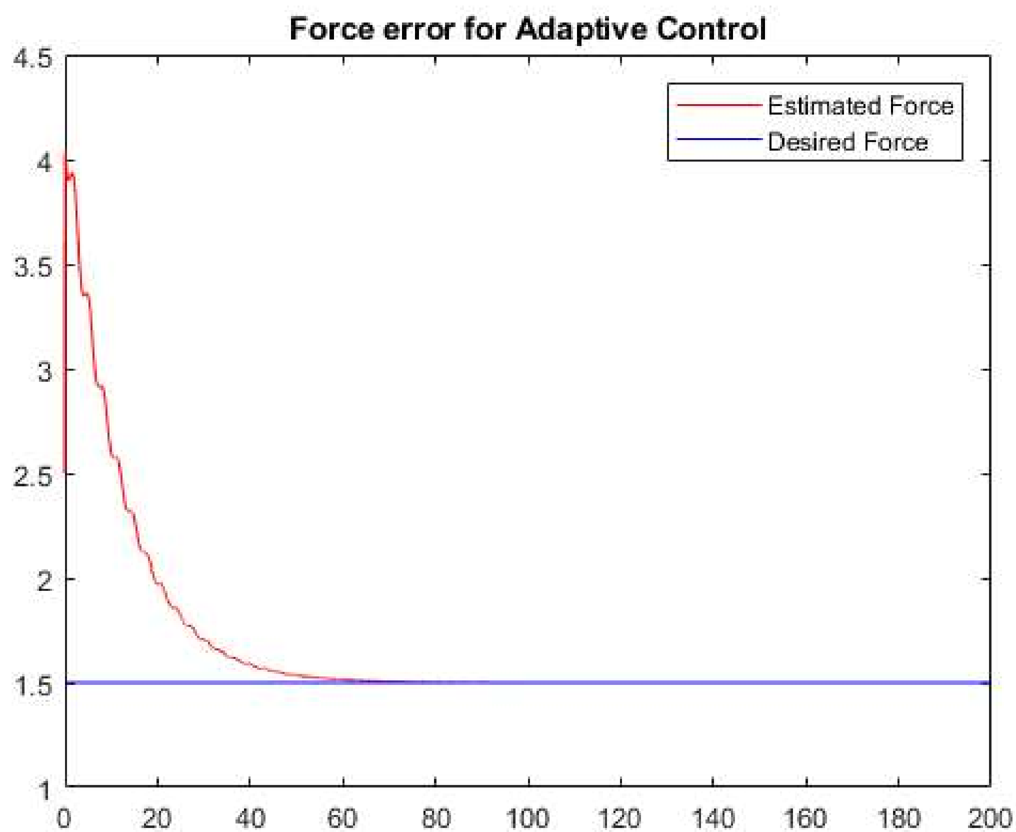
```
plot(T, X(:,5),'r-');
hold on
plot(T, 6*ones(size(T,1),1) ,'b-');
legend('Estimated Gravity', 'Desired Gravity')
title('Gravity error for Adaptive Control');

function [dx] = planarArmODEadaptive(t,x)
% desired trajectories
theta_d = [-sin(t)];
dtheta_d = [-cos(t)];
ddtheta_d = [sin(t)];

% given trajectories
theta = x(1);
dtheta= x(2);
% changing parameters
I_bar = x(3);
fv_bar = x(4);
mgd_bar = x(5);

% errors
global lambda e de a v r
lambda = 0.999;
e = theta - theta_d;
de = dtheta - dtheta_d;
a = ddtheta_d - (lambda*de);
v = dtheta_d - (lambda*e);
r = de + (lambda*e);
% a positive definite matrix (to be used later for theta_tilda)
P = 0.019*eye(3);

% True model
global M C G
M = [7.5];
C = [1.5];
G = [6*sin(x(1))];
invM = inv(M);
invMC= inv(M)*C;

% Estimated model
global M_bar C_bar G_bar
M_bar = [I_bar];
C_bar = [fv_bar];
G_bar = [mgd_bar*sin(x(1))];

tau = adaptive_ctrl(theta_d, dtheta_d, ddtheta_d, theta, dtheta);
global torque
torque = [torque, tau];
%update the system state, compute dx
dx=zeros(5,1);
dx(1) = x(2);
dx(2) = -invMC* x(2) - invM*G + invM*tau; % because ddot theta = -M^{-1}(C \dot Theta) + M^{-
1} tau
% WE HAVE
% M(q)q_ddot + C(q,q_dot)q_dot +G(q) = Tau

% Tau = M_bar(q)*a+ C_bar(q,q_dot)*v + G_bar(q) - Kv*r
```

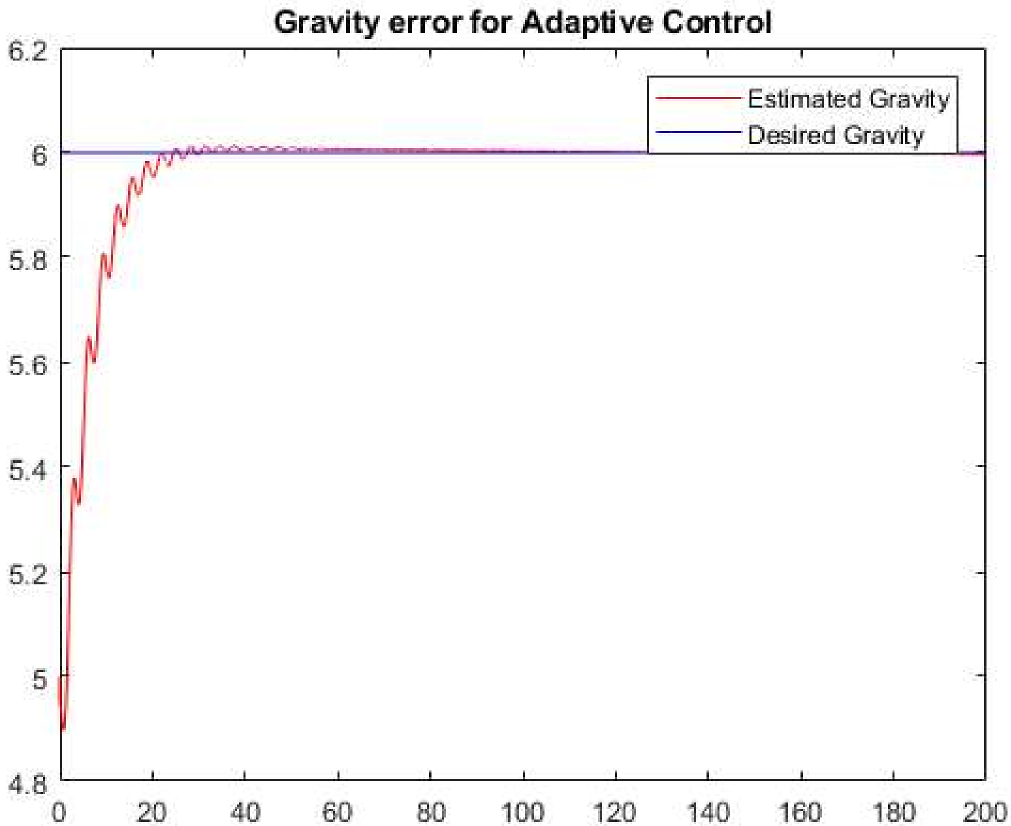
```
% On equating the above equations

% M(q)q_ddot + C(q,q_dot)q_dot +G(q) = M_bar(q)*a+ C_bar(q,q_dot)*v + G_bar(q) - Kv*r

% q_ddot = a + r_dot

% q_dot = v + r

% On substituting the values we get,

% M(q)(a + r_dot) + C(q,q_dot)(v + r) +G(q) = M_bar(q)*a+ C_bar(q,q_dot)*v + G_bar(q) - Kv*r

% M(q)(r_dot) + C(q,q_dot)(r) + Kv*r = (M_bar- M)*a+ (C_bar- C)*v + (G_bar- G)(q)

% On parameterizing the Right hand side of equation
% We obtain

% % M(q)(r_dot) + C(q,q_dot)(r) + Kv*r = Y(a,v,q)*Theta_tilda

%where Y = [a,v,sin(theta)] and

%Theta_tilda= [(I_bar - I);(fv_bar - fv);(mgd_bar - mgd)]

Y = [a,v,sin(x(1))];
dx(3:5) = -(inv(P)*transpose(Y)*r);
end

% function to calculate torque
function tau = adaptive_ctrl(theta_d, dtheta_d, ddtheta_d, theta, dtheta)
global M C M_bar C_bar G_bar lambda e de a v r
%Kp = 100*eye(1);
Kv = 300*eye(1);
tau = (M_bar*a)+ (C_bar*v) + (G_bar) - Kv*r;
end
```


Theta for Adaptive Control
Actual Theta
Desired Theta
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100
120
140
160
180
200


Torque for Adaptive Control
torque
50
0
-50
-100
-150
-200
-250
-300
-350
-400
0
20
40
60
80
100
120
140
160
180
200


Inertia error for Adaptive Control
Estimated Inertia
Desired Inertia
8.8
8.6
8.4
8.2
8
7.8
7.6
7.4
0
20
40
60
80
100
120
140
160
180
200


Force error for Adaptive Control
Estimated Force
Desired Force
4.5
4
3.5
3
2.5
2
1.5
1
0
20
40
60
80
100
120
140
160
180
200


Gravity error for Adaptive Control
Estimated Gravity
Desired Gravity
6.2
6
5.8
5.6
5.4
5.2
5
4.8
0
20
40
60
80
100
120
140
160
180
200