```matlab
%% Adaptive Passive Controller
%% RBE 502 Fall 2018
%% Homework 6
%% December 6, 2018

function dxdt = adaptivePassive(t,x,cont_input,original_param,estimate_param,alfa0)
global ddtheta

%persistent k
%persistent ii
%% Parameters of the controller

lambda=cont_input{1,1};
kv=cont_input{1,2};
L=cont_input{1,3};
theta_dd=ddtheta
%% Parameters of the original system
I=original_param(1,1);
mgd=original_param(1,2);
fv=original_param(1,3);

%% Parameters of the estimated model
I_e=estimate_param(1,1);
mgd_e=estimate_param(1,2);
fv_e=estimate_param(1,3);

%% Desired Trajectory setting up
theta_d=-sin(t);
dtheta_d=-cos(t);
ddtheta_d=sin(t);
%% defining the controller

 e=x(1)-theta_d;                            %position error vector
 e_dot=x(2)-dtheta_d;                       %Velocity error vector
 e_ddot=theta_dd-ddtheta_d                  %Acceleration error

 r=e_dot+(lambda*e);
 r_dot=e_ddot+(lambda*e_dot);

 a=theta_dd-r_dot;                         %Acceleration input
 v=x(2)-r;                                 %Velcotiy input

 Y=[sin(x(1)) v a];                        %The regressor

 alfa_gradient=-inv(L)*Y'*r;

 alfa=[x(3); x(4); x(5)];

 % The next portion is for if we want to augment the integration manually
 % inside the ODE
 %if t==0

     %alfa=alfa0;
     %k=0;

 %else
     %k=t-k;
     %alfa=alfa+alfa_gradient*k;

 %end
```

```
 u=Y*alfa-kv*r;                                  %Torque Input



theta_dd=(u-mgd*sin(x(1))-fv*x(2))/I;           %The oro

% Defining the first order state vector.
dxdt=zeros(5,1);
dxdt(1)=x(2);
dxdt(2)= theta_dd;
dxdt(3)=alfa_gradient(1)
dxdt(4)=alfa_gradient(2)
dxdt(5)=alfa_gradient(3)

%k=t;
ddtheta=theta_dd
end
```