Assignment :

Adaptive Control Systems

“ Model Identification of Separately Excited Armature Controlled DC Motor ”

Submitted By :

Raheel Javed

Modelling of Plant:

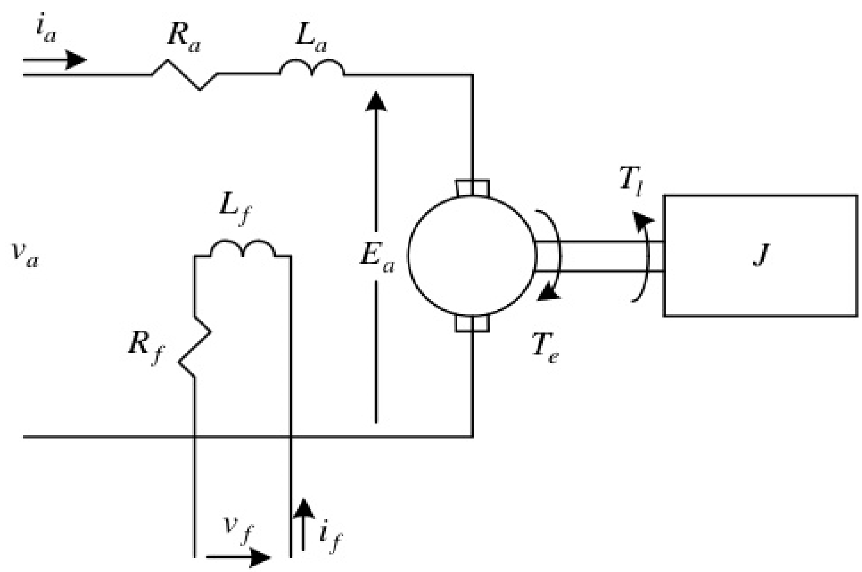


Figure 1: Separately excited DC motor

Fig. 1 shows circuit diagram of a separately excited DC motor. The field circuit consists of a voltage source v_f , field coil resistance R_f and coil inductance L_f. For an armature controlled motor voltage v_f is held constant. The armature circuit consists of armature voltage v_a , coil resistance R_a , armature winding inductance L_a and back EMF E_a generated at rotor terminals. The remaining parameters are:

$T_e =$ Torque generated by motor $\qquad$ $T_l =$ Load torque

$J_o =$ Inertia of motor, load combination referred to the shaft

Writing KVL for armature circuit,

$$L_a \frac{di_a}{dt} + Ri_a + E_a = v_a$$

Where,

$i_a =$ Armature current and E_a can be written as,

$$E_a = K_b \frac{d\theta}{dt}$$

Where,

$K_b =$ Back EMF constant $\qquad$ $\theta =$ Shaft angle

`

So,

(a) $$L_a \frac{di_a}{dt} + Ri_a + K_b \frac{d\theta}{dt} = v_a$$

Writing equation for torque equilibrium,

$$J_o \frac{d^2\theta}{dt^2} + T_l = T_e$$

Or,

(b) $$J_o \frac{d^2\theta}{dt^2} + b_o \frac{d\theta}{dt} = K_t i_a$$

Where,

b_o = Viscous friction coefficient of motor, load combination referred to the shaft

K_t = Torque constant of motor

From equations (a), (b) and using,

$$x_1 = i_a, \quad x_2 = \theta, \quad x_3 = \theta^{\cdot}, \quad u = v_a$$

The state space representation of armature controlled DC motor is given as,

$$x^{\cdot} = Ax + Bu$$

Where,

$$A = \begin{bmatrix} -\frac{R_a}{L_a} & 0 & -\frac{K_b}{L_a} \\ 0 & 0 & 1 \\ \frac{K_t}{J_o} & 0 & -\frac{b_o}{J_o} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix}$$

`

Observer-Like Estimator:

Assuming that all the states of system are measurable and available the structure of estimator can be given as,

$$x^{\wedge\,\cdot} = A^{\wedge}\,x + B^{\wedge}\,u + A_m(x^{\wedge} -\ x)$$

Where,

$x^{\wedge}\ \in R^3$ and u is a scalar. Error dynamics can be given as,

$$e^{\cdot} = \phi\,x +\ \varphi\,u + A_m\,e$$

Where,

$$e = x^{\wedge} - x, \qquad \phi = A^{\wedge} - A, \qquad \varphi = B^{\wedge} - B$$

If lyapunov function is chosen as,

$$V = e^T P e + Trace(\phi^T \phi + \varphi^T \varphi), \qquad P > 0$$

Then, using substitution and,

$$Trace(ab^T)\ =\ b^T a$$

Where, a and b are column vectors.

We have,

$$A^{\wedge\cdot} = \phi^{\cdot} =\ -Pex^T$$

$$B^{\wedge\cdot} =\ \varphi^{\cdot} =\ -Peu$$

$$V^{\cdot} = e^T(A_m^T P + PA_m\,)e$$

For $V^{\cdot}$ To be semi negative definite A_m should be chosen stable and $P > 0$ such that,

$$A_m^T P + PA_m\ < 0$$

This guarantees the convergence of $e \rightarrow\ 0$ as $t \rightarrow \infty$ by Barbalet's Lemma which states that $e \rightarrow\ 0$ as $t \rightarrow \infty$ if energy of error is bounded ($e\ \in L_2$) and is uniformly continuous ($e^{\cdot} < \infty$). The convergence of parameters is not guaranteed but the probability of convergence is high for persistent excitation. The convergence of parameters is also dependent upon selection of A_m and P.

`

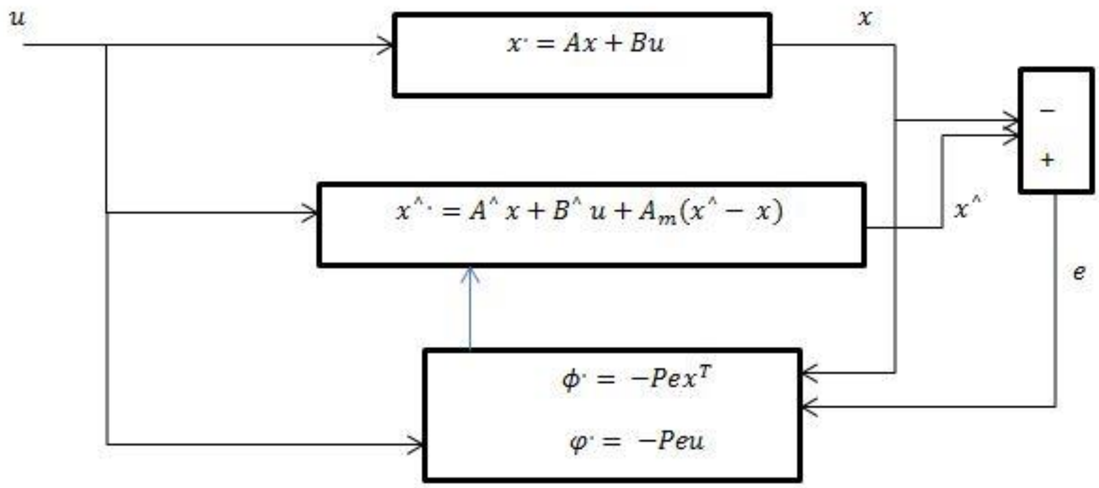


Figure 2: Block diagram implementation of observer-like estimator

Design of Estimator:

- There 5 parameters that are unknown which include 4 entries of A matrix and 1 entry of B matrix.
- Choose a stable A_m matrix of size 3x3.
- Solve the LMI for P. Size of P is 3x3 and P is symmetric positive definite.
- Let $A^{\wedge}(1,1) = a1,\ \ A^{\wedge}(1,3) = a2,\ \ A^{\wedge}(3,1) = a3,\ \ A^{\wedge}(3,3) = a4\ \ and\ \ B^{\wedge}(1) = b$
- Then,

$$a1^{\cdot} = (-Pex^T)(1,1), \qquad a2^{\cdot} = (-Pex^T)(1,3), \qquad a3^{\cdot} = (-Pex^T)(3,1),$$
$$a4^{\cdot} = (-Pex^T)(3,3), \qquad b^{\cdot} = (-Peu)(1)$$

- Fig. 2 shows the implementation of estimator with the help of block diagram.

`

Matlab Implementation:

Code:

Function to Solve Differential Equations:

```
function dy=AC_MI(t, y, A, B, ut, tu, Am, P)
dy=zeros(11, 1);
dx=zeros(3, 1);          %Derivative of system states
dx_=zeros(3, 1);         %Derivative of estimator states
da=zeros(4, 1);          %Derivative unknown A parameters
db=zeros(1, 1);          %Derivative of unknown B parameters
x=zeros(3, 1);
x_=zeros(3, 1);
a=zeros(4, 1);
b=zeros(1, 1);

for i=1:3
    dx(i)=dy(i);
    x(i)=y(i);
end

for i=1:3
    dx_(i)=dy(3+i);
    x_(i)=y(3+i);
end

for i=1:4
    da(i)=dy(6+i);
    a(i)=y(6+i);
end

db=dy(11);
b=y(11);

A_=[a(1) 0 a(2); 0 0 1; a(3) 0 a(4)];
B_=[b; 0; 0];
u=interp1(tu, ut, t);

dx=A*x+B*u;                 %System Model
dx_=A_*x+B_*u+Am*(x_-x);    %Estimator Model

Temp1=-P*(x_-x)*x';         %Adaptation law for A parameters
da(1)=Temp1(1, 1);
da(2)=Temp1(1, 3);
da(3)=Temp1(3, 1);
da(4)=Temp1(3, 3);

Temp2=-P*(x_-x)*u;          %Adaptation law for B parameter
db=Temp2(1);

dy(1:3)=dx;
dy(4:6)=dx_;
```

`

```
dy(7:10)=da;
dy(11)=db;

end
```

Main Routine:

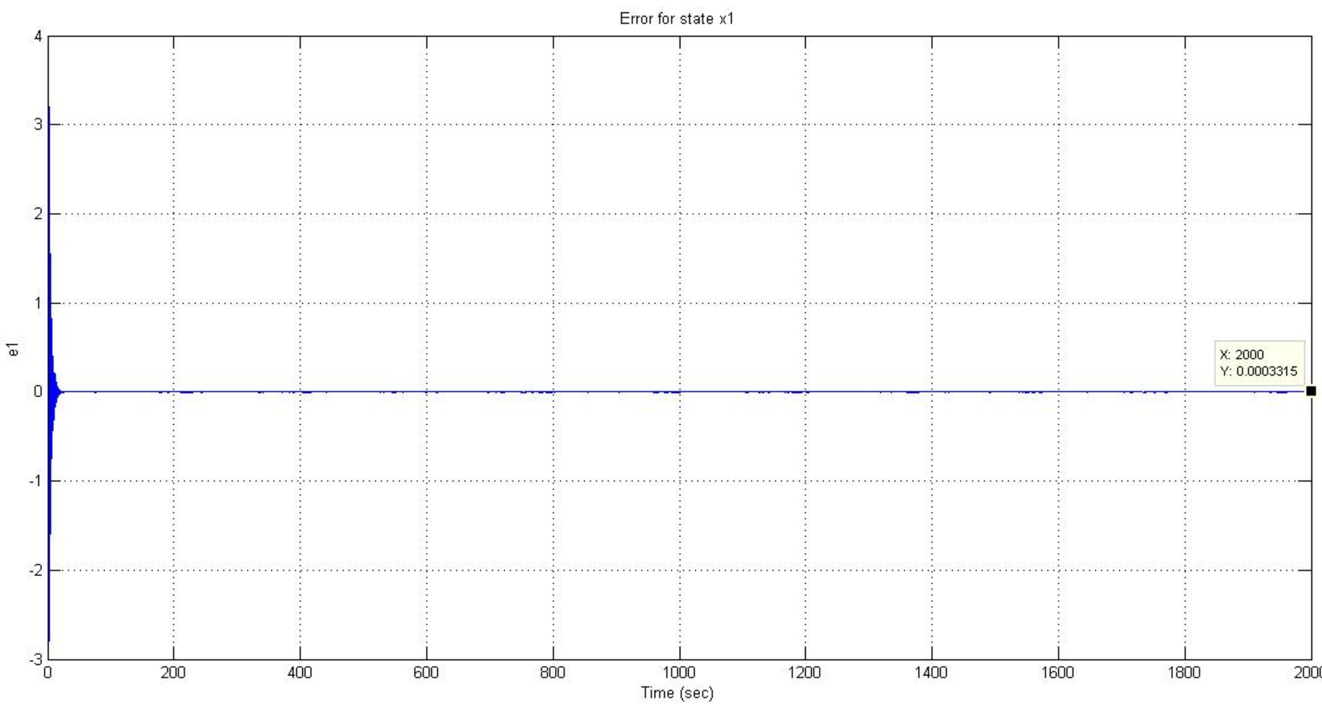
```
clear
clc
close all

ts=1:0.1:2000;              %Time for solution
tu=ts;
ut=20*sin(2*3.142*8*tu);         %Input signal

Am=blkdiag(-1, -3, -2);     %Am matrix calculation

setlmis([]);                %Solution of LMI for P
P=lmivar(1, [3, 1]);
lmiterm([1 1 1 P], Am', 1, 's');
lmiterm([-2 1 1 P], 1, 1);
lmisys=getlmis;
[T, X]=feasp(lmisys);
P=dec2mat(lmisys, X, P);

%System Parameters
R=2;         %Ohms
L=0.5;       %Henrys
Kt=0.1;
Kb=0.1;
b=0.2;       %Nms
J=0.02;      %kg.m^2/s^2
A=[-R/L 0 -Kb/L; 0 0 1; Kt/J 0 -b/J]
B=[1/L; 0; 0]

%Solution of differential equations
yo(1:11)=zeros(1, 11);
[tss, y] = ode45(@(t, y) AC_MI(t, y, A, B, ut, tu, Am, P), ts, yo);

%Final value of found parameters
N=size(tss);
n=max(N);
a1=y(n, 7)
a2=y(n, 8)
a3=y(n, 9)
a4=y(n, 10)
b1=y(n, 11)

%Plots of parameters and errors vs time
figure
plot(tss, y(:, 4)-y(:, 1));
title('Error for state x1')
xlabel('Time (sec)')
```

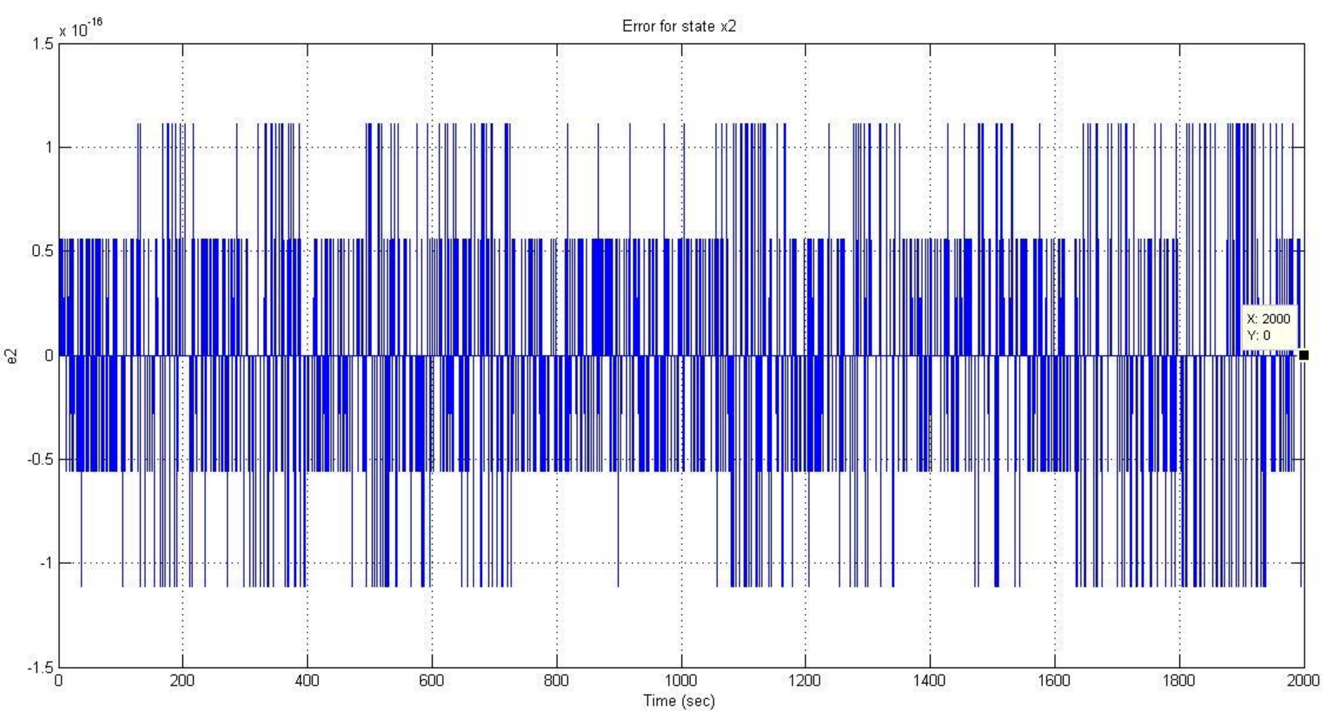
`

```
ylabel('e1')

figure
plot(tss, y(:, 5)-y(:, 2));
title('Error for state x2')
xlabel('Time (sec)')
ylabel('e2')
grid

figure
plot(tss, y(:, 6)-y(:, 3));
title('Error for state x3')
xlabel('Time (sec)')
ylabel('e3')
grid

figure
plot(tss, y(:, 7));
title('Parameter a1')
xlabel('Time (sec)')
grid

figure
plot(tss, y(:, 8));
title('Parameter a2')
xlabel('Time (sec)')
grid

figure
plot(tss, y(:, 9));
title('Parameter a3')
xlabel('Time (sec)')
grid

figure
plot(tss, y(:, 10));
title('Parameter a4')
xlabel('Time (sec)')
grid

figure
plot(tss, y(:, 11));
title('Parameter b')
xlabel('Time (sec)')
grid
```

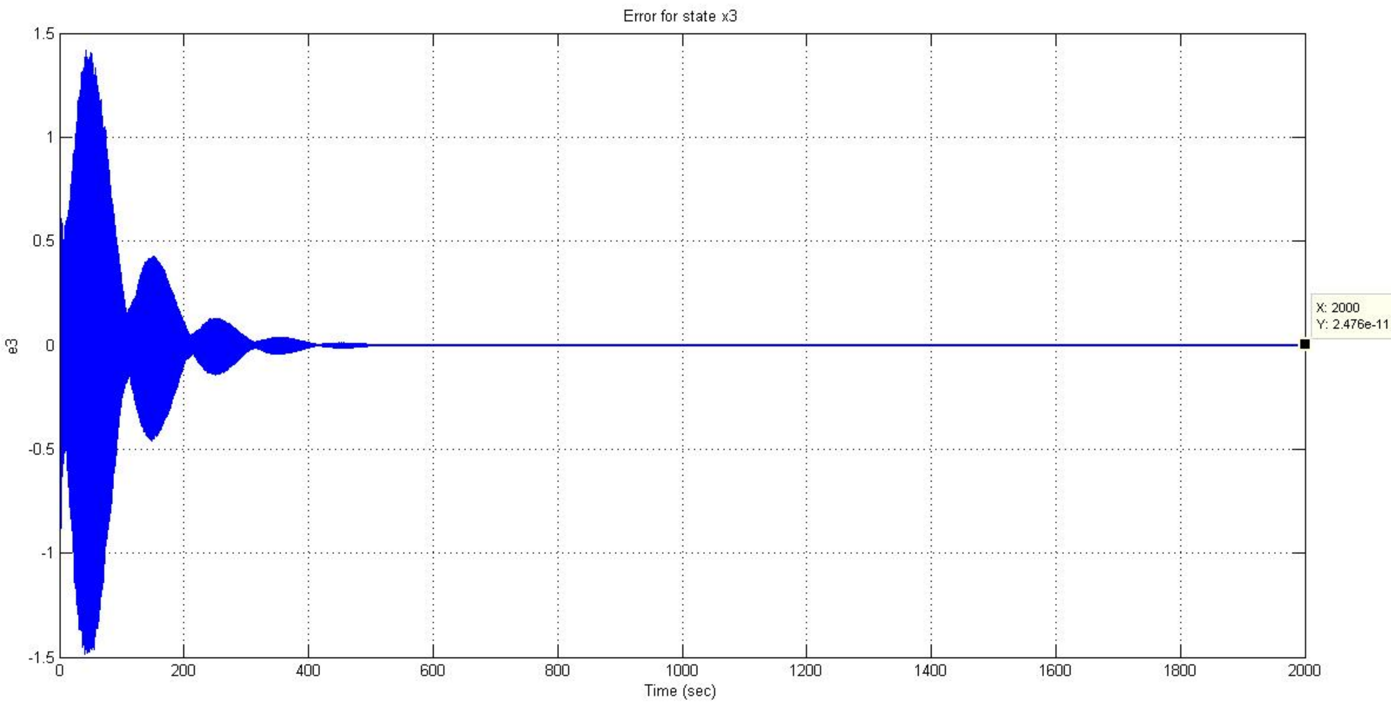
`

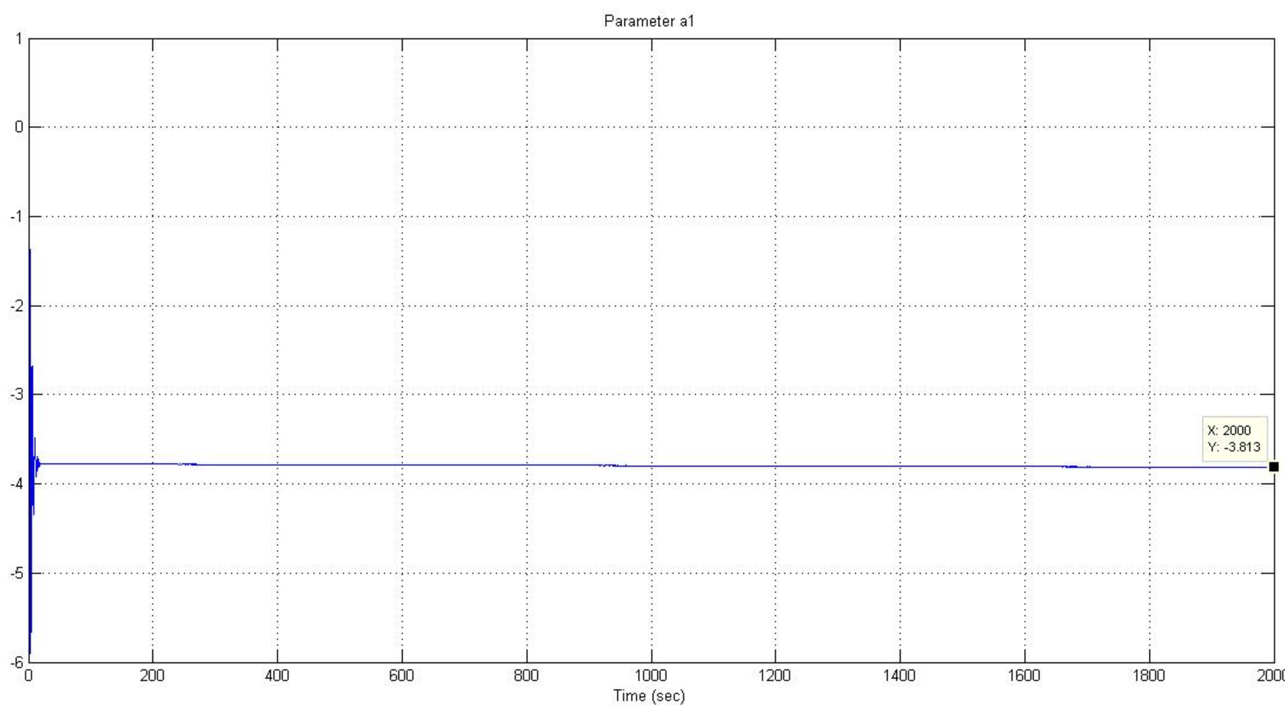
Simulation Results:



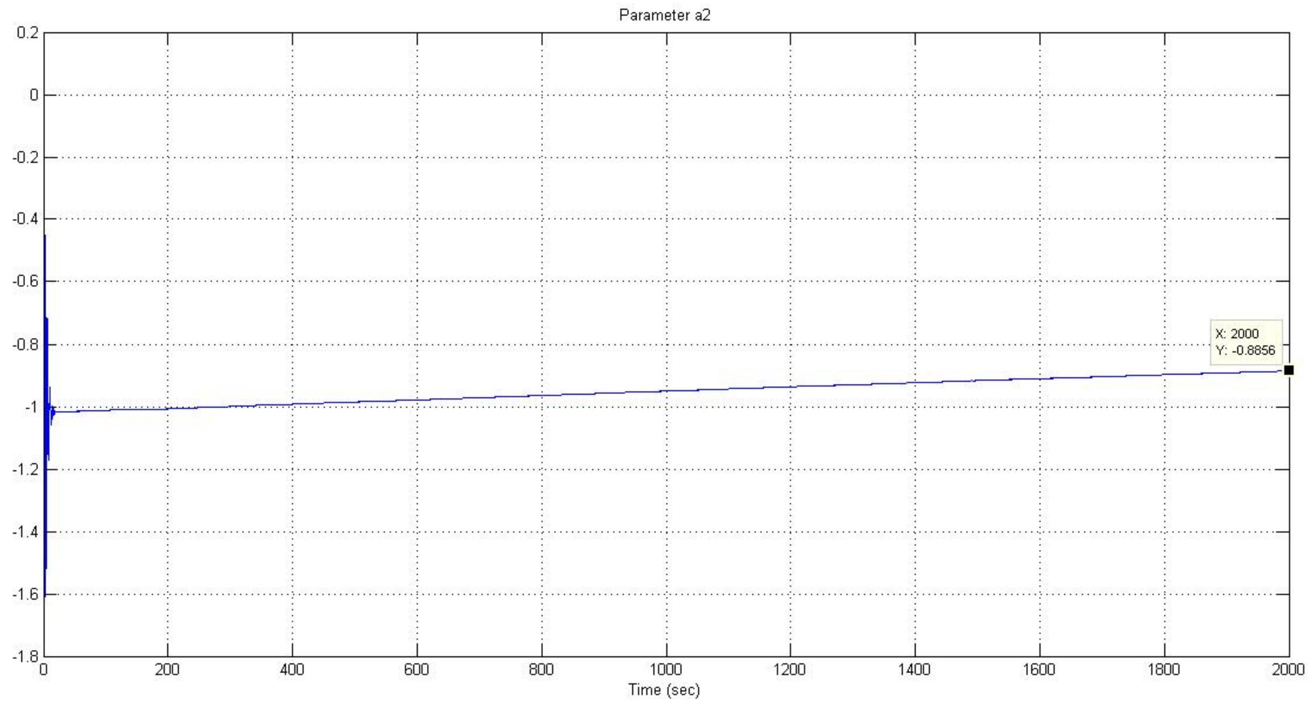


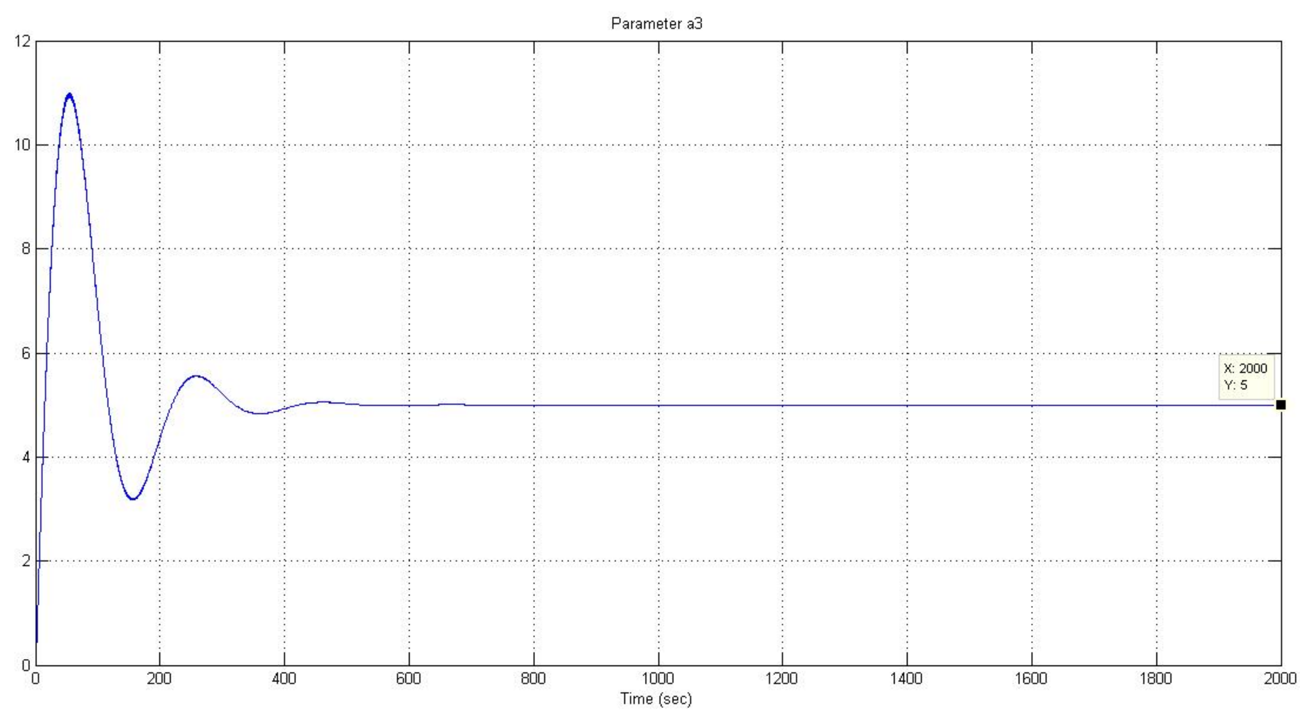
`

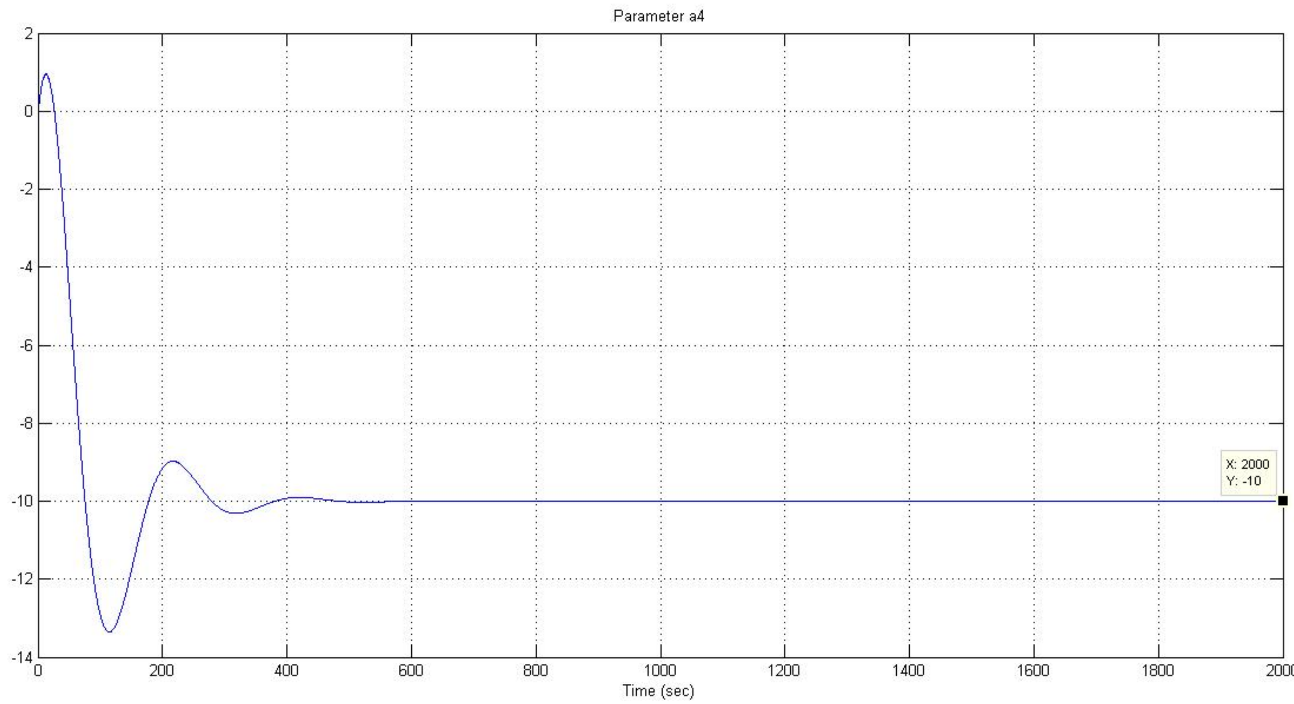

Error for state x3
e3
Time (sec)
X: 2000
Y: 2.476e-11

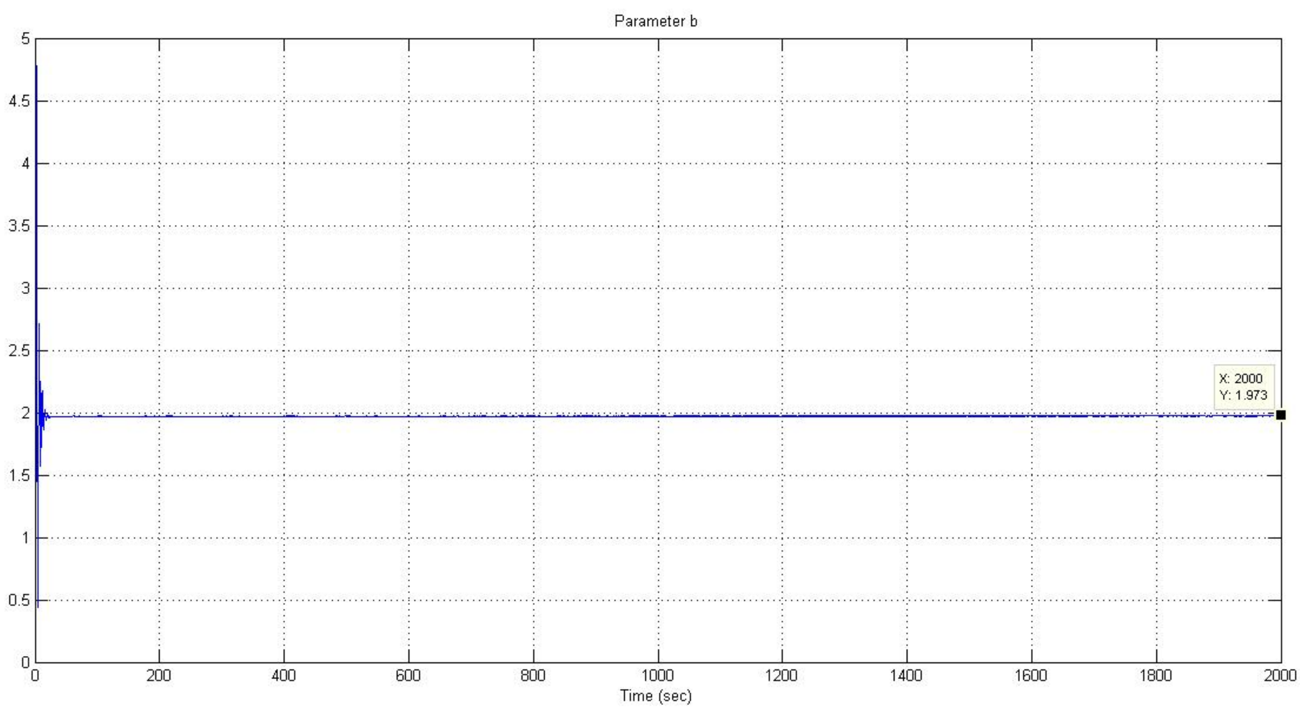

Parameter a1
Time (sec)
X: 2000
Y: -3.813

`


Parameter a2
X: 2000
Y: -0.8856
Time (sec)


Parameter a3
X: 2000
Y: 5
Time (sec)


Parameter a4
X: 2000
Y: -10
Time (sec)


Parameter b
X: 2000
Y: 1.973
Time (sec)

`

Comments:

- All states errors converge to zero.
- Two parameters $a3$ and $a4$ converge to their true value.
- Parameter b converges but there is a small error between true and converged value.
- Parameters $a1$ and $a2$ are converging but need more time.
- Convergence of parameters increases for more time, high input amplitude, a specific range of input frequencies and smaller magnitude of A_m entries.

`