**Submitted By :**

Raheel Javed

**Assignment :**

**Adaptive Control Systems**

**“ Model Identification of Separately Excited Armature Controlled DC Motor ”**

**Modelling of Plant:**

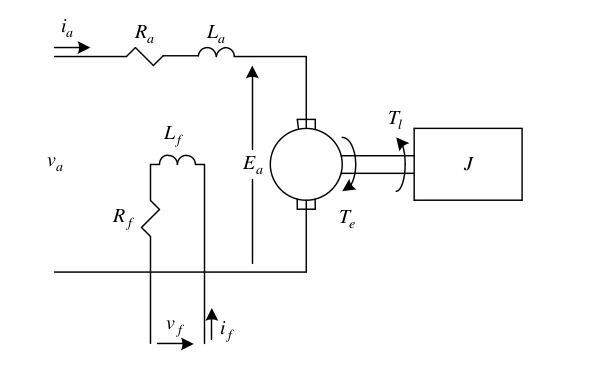


Figure 1: Separately excited DC motor

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

Torque generated by motor Load torque

Inertia of motor, load combination referred to the shaft

Writing KVL for armature circuit,

Where,

Armature current and can be written as,

Where,

Back EMF constant Shaft angle

So,

|  |  |  |
| --- | --- | --- |
|  |  |  |

Writing equation for torque equilibrium,

Or,

|  |  |  |
| --- | --- | --- |
|  |  |  |

Where,

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

Torque constant of motor

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

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

Where,

**Observer-Like Estimator:**

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

Where,

and is a scalar. Error dynamics can be given as,

Where,

If lyapunov function is chosen as,

Then, using substitution and,

Where, and are column vectors.

We have,

For To be semi negative definite should be chosen stable and such that,

This guarantees the convergence of as by Barbalet’s Lemma which states that as if energy of error is bounded () and is uniformly continuous (). 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 and .

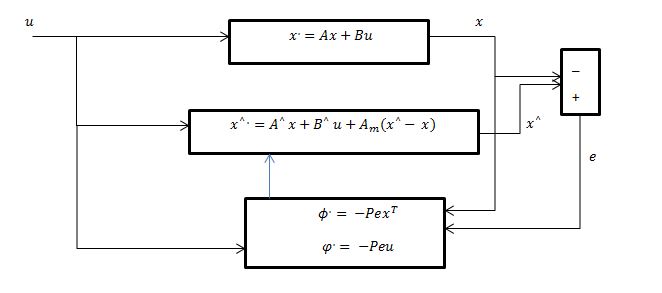


Figure 2: Block diagram implementation of observer-like estimator

**Design of Estimator:**

* There 5 parameters that are unknown which include 4 entries of matrix and 1 entry of matrix.
* Choose a stable matrix of size 3x3.
* Solve the LMI for Size of P is 3x3 and P is symmetric positive definite.
* Let
* Then,
* 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

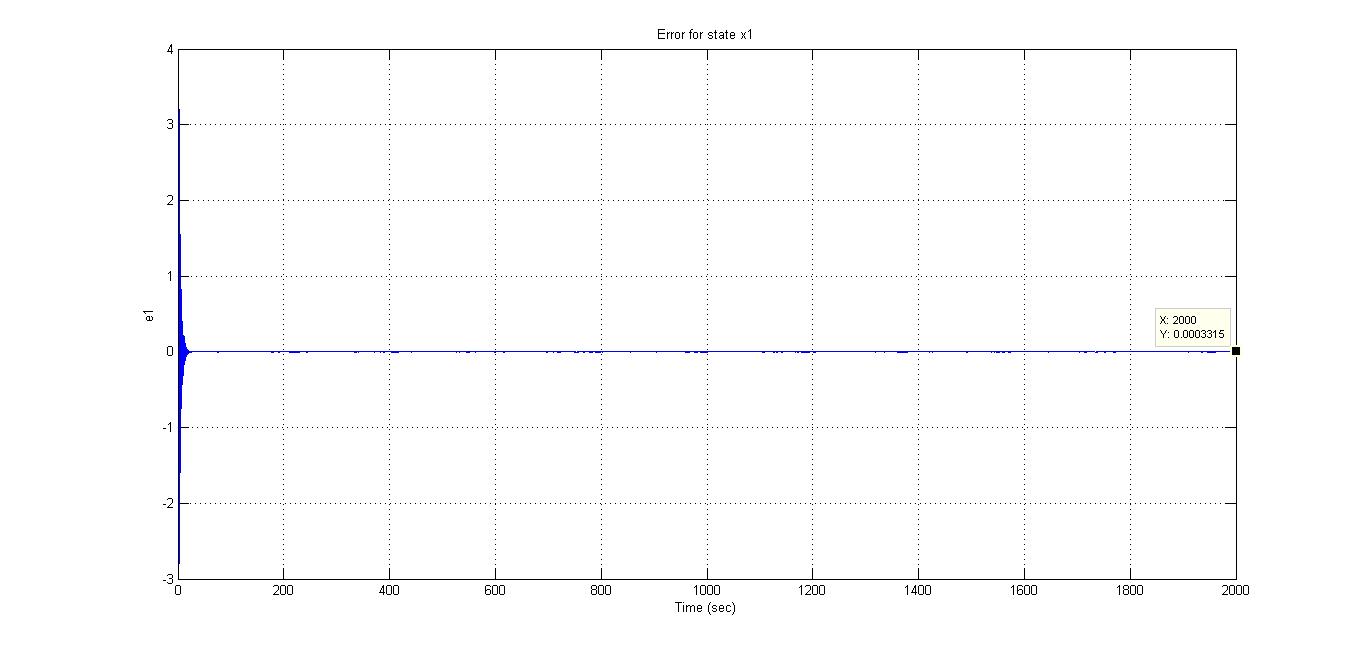
plot(tss, y(:, 11));

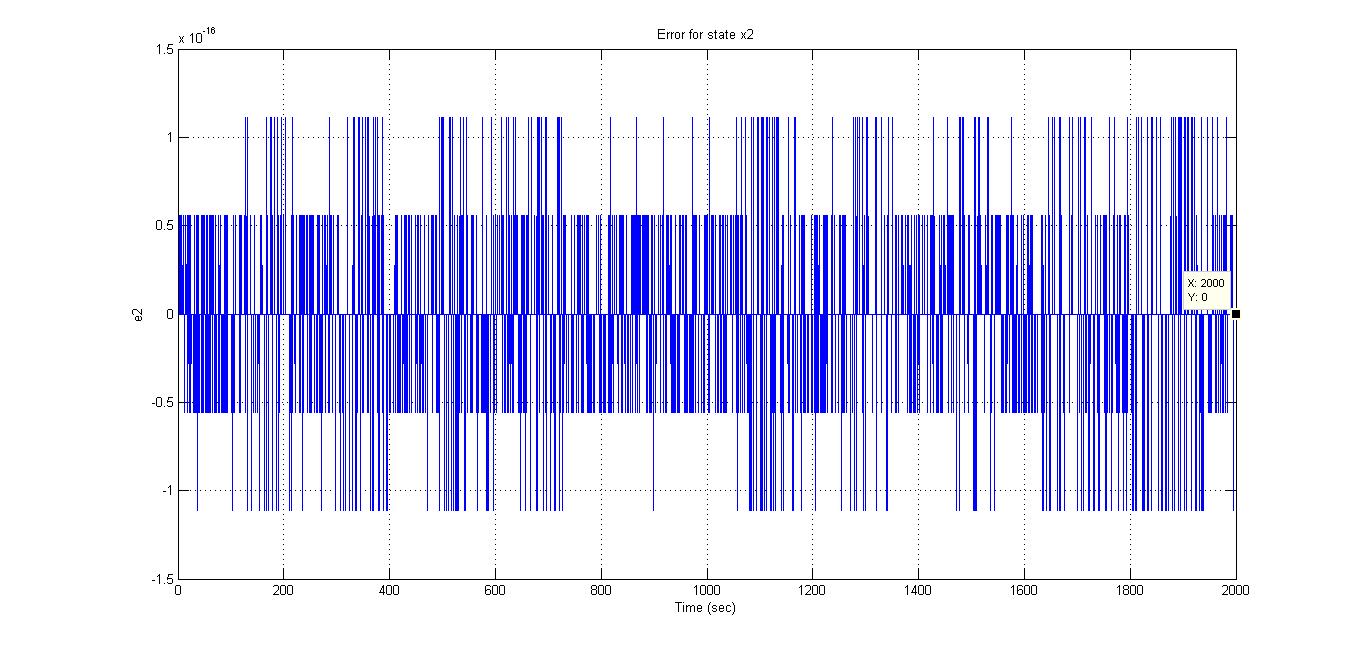
title('Parameter b')

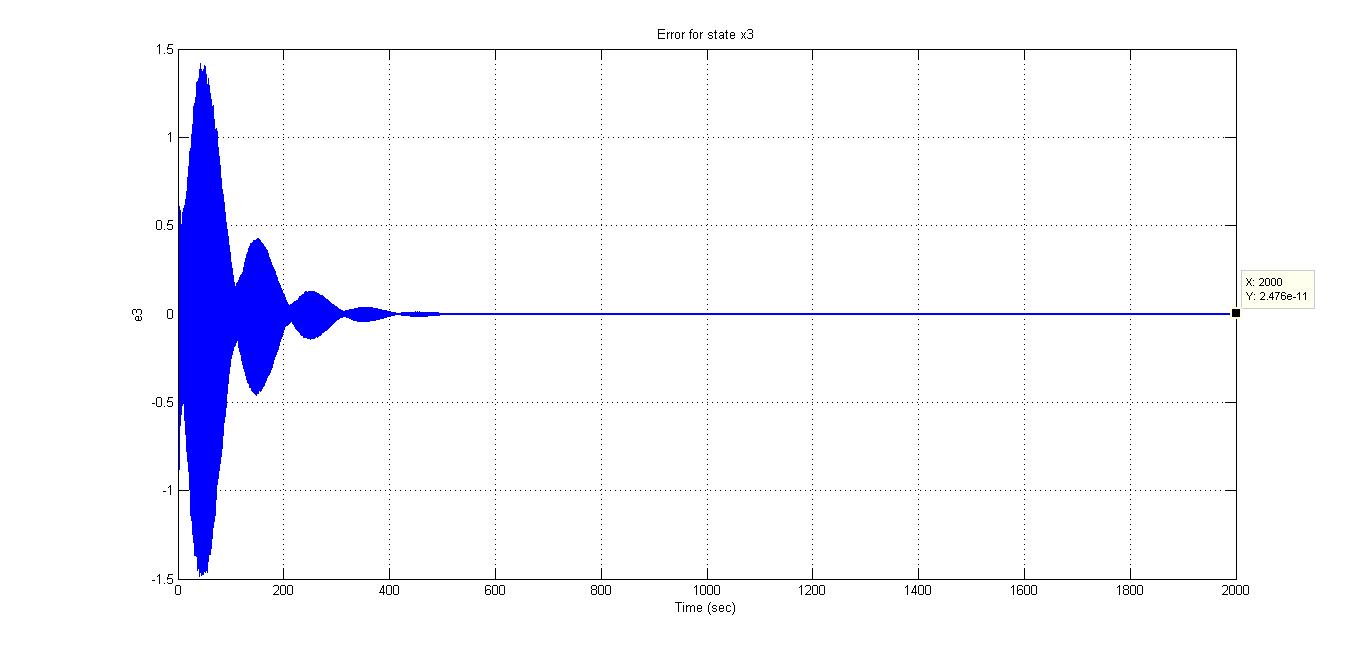
xlabel('Time (sec)')

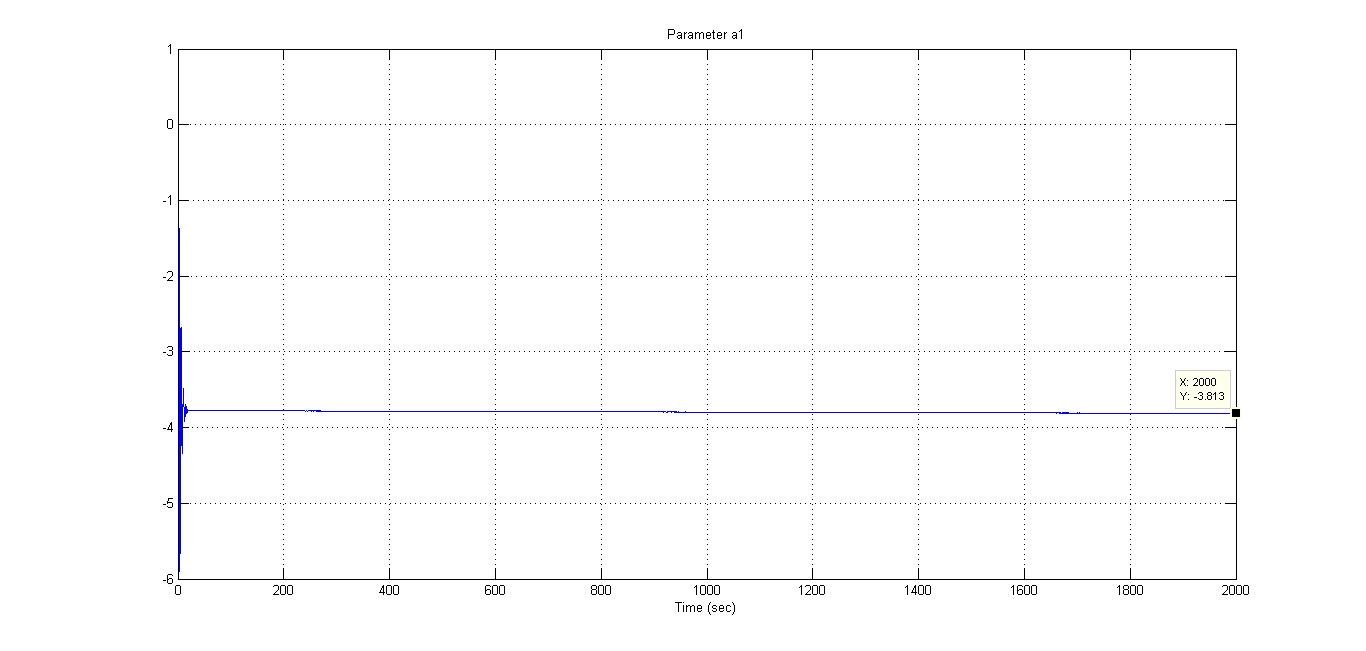
grid

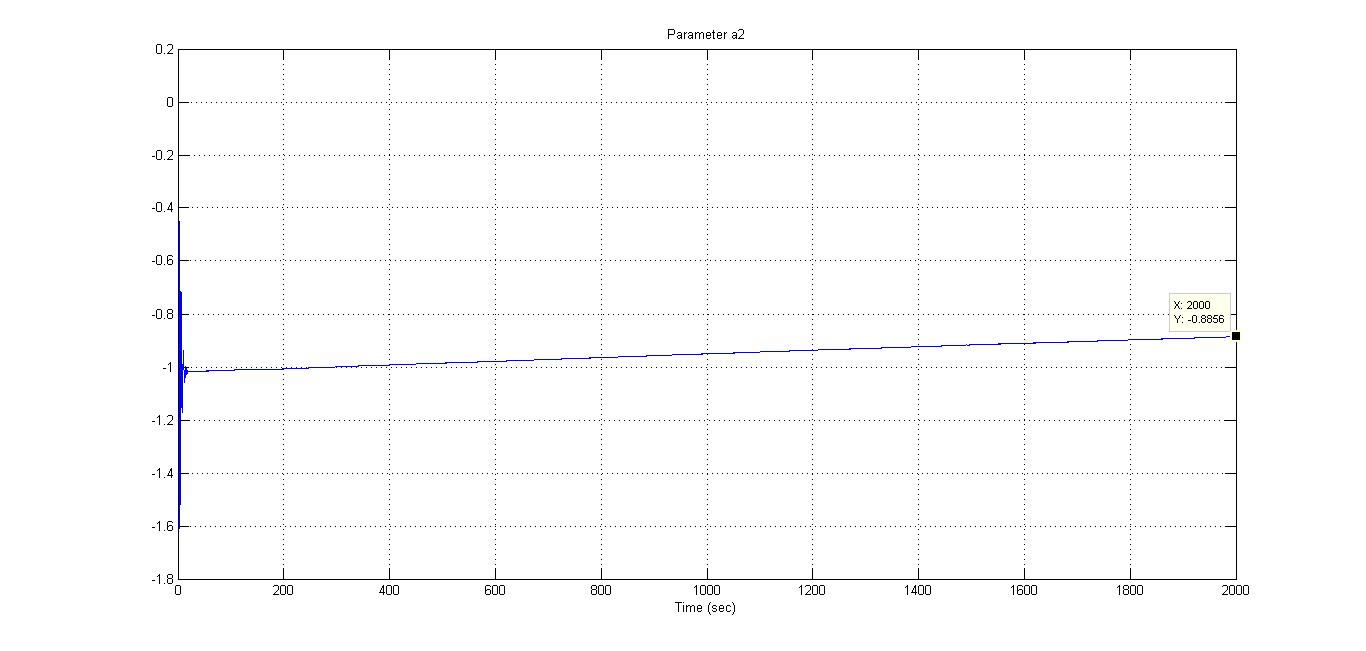
**Simulation Results:**

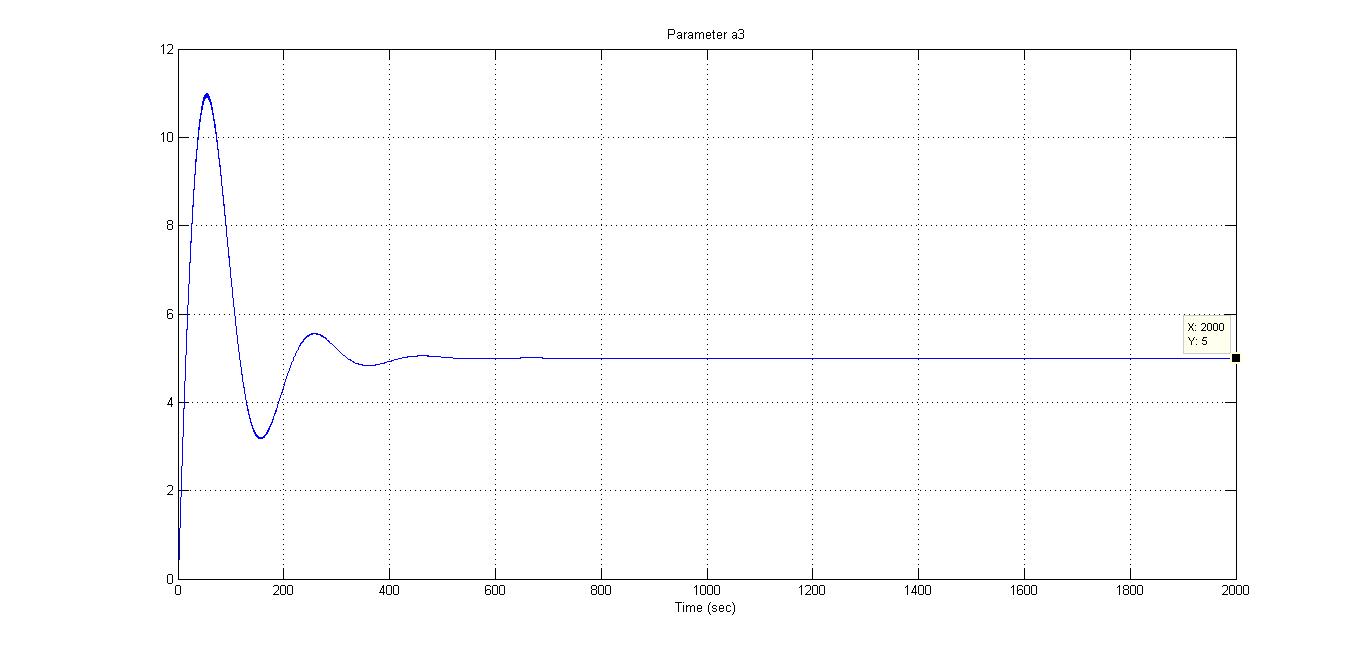


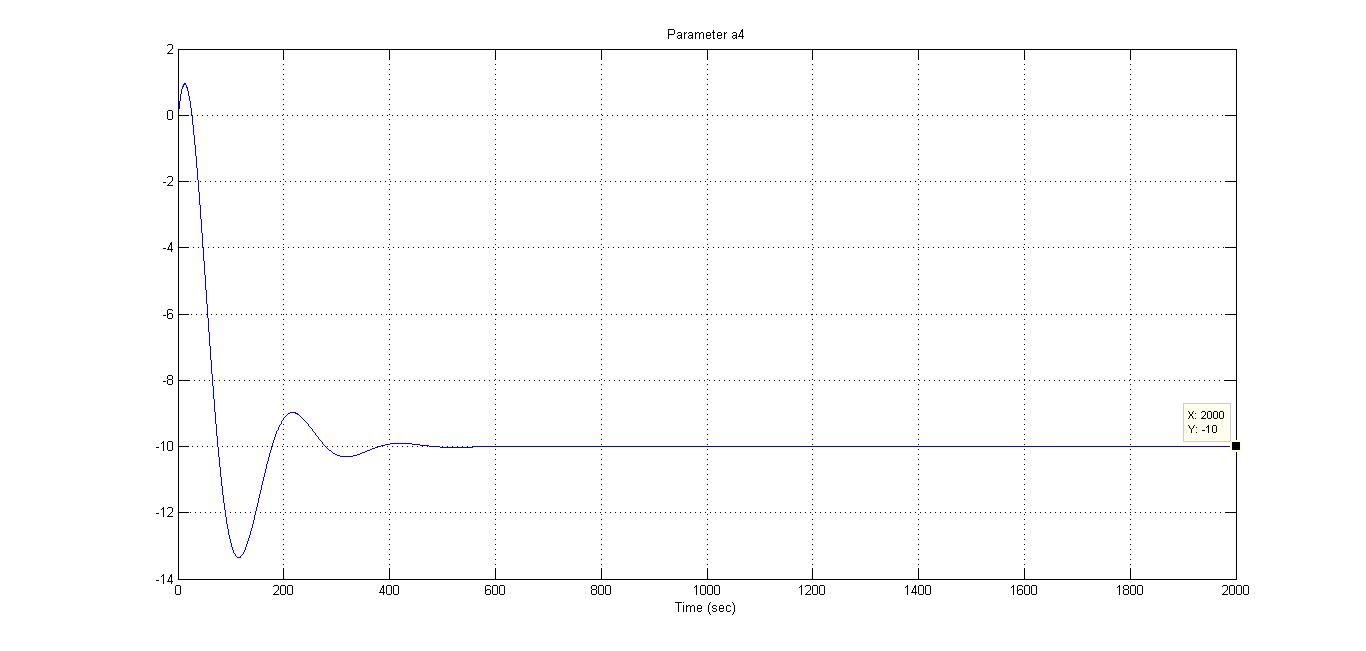
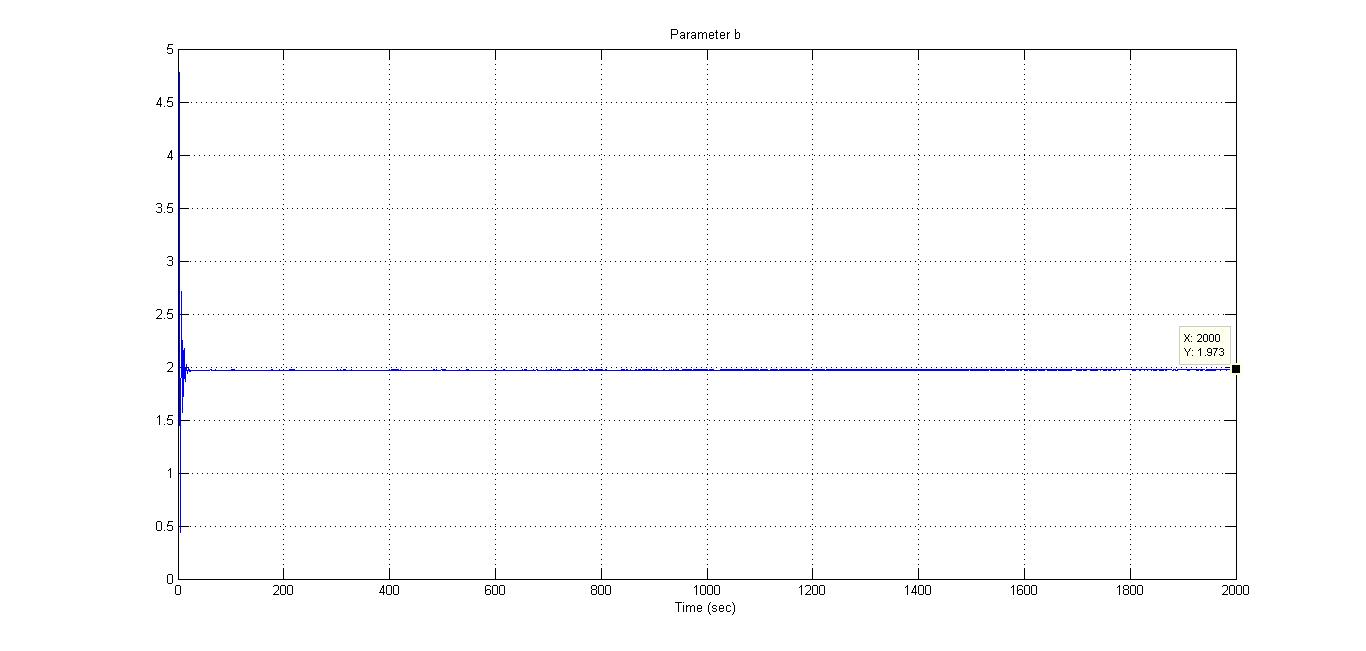












**Comments:**

* All states errors converge to zero.
* Two parameters and converge to their true value.
* Parameter b converges but there is a small error between true and converged value.
* Parameters and 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 entries.