STUDY OF SOFTWARE INTERFACE FOR ADAPTIVE CONTROL SYSTEM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Electronics & Instrumentation Engineering

By

ABHISHEK KUMAR SRIVASTAVA
ROLL NO -10307003
&
SANDEEP CHHABRA
ROLL NO -10307020

Department of Electronics and Communication Engineering
(Electronics and Instrumentation Engineering)
National Institute of Technology
Rourkela
2007

STUDY OF SOFTWARE INTERFACE FOR ADAPTIVE CONTROL SYSTEM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Electronics & Instrumentation Engineering

By

ABHISHEK KUMAR SRIVASTAVA
ROLL NO -10307003
&
SANDEEP CHHABRA
ROLL NO -10307020

Under the Guidance of
Prof.S.K. PRAMANIK

Department of Electronics and Communication Engineering
(Electronics and Instrumentation Engineering)
National Institute of Technology
Rourkela

2007

CERTIFICATE

This is to certify that the thesis entitled, "Study of Software Interface For Adaptive Control System" submitted by Sri Abhisek Kumar Srivastava & Sri Sandeep Chhabra in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electronics & Instrumentation Engineering at the National Institute of Technology ,Rourkela (Deemed University) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof. S.K. Pramanik
Dept. of ECE (E&IE)
National Institute of Technology
Rourkela - 769008

ACKNOWLEDGEMENT

We owe a deep sense of gratitude to one and many for the successful completion of this project. Special thanks to my project guide Prof. S.K. Pramanik without whose able guidance this wouldn’t have been possible in the first place.

We extend our thanks to Dr. G.Panda, HOD, and ECE Dept. for accepting our project.

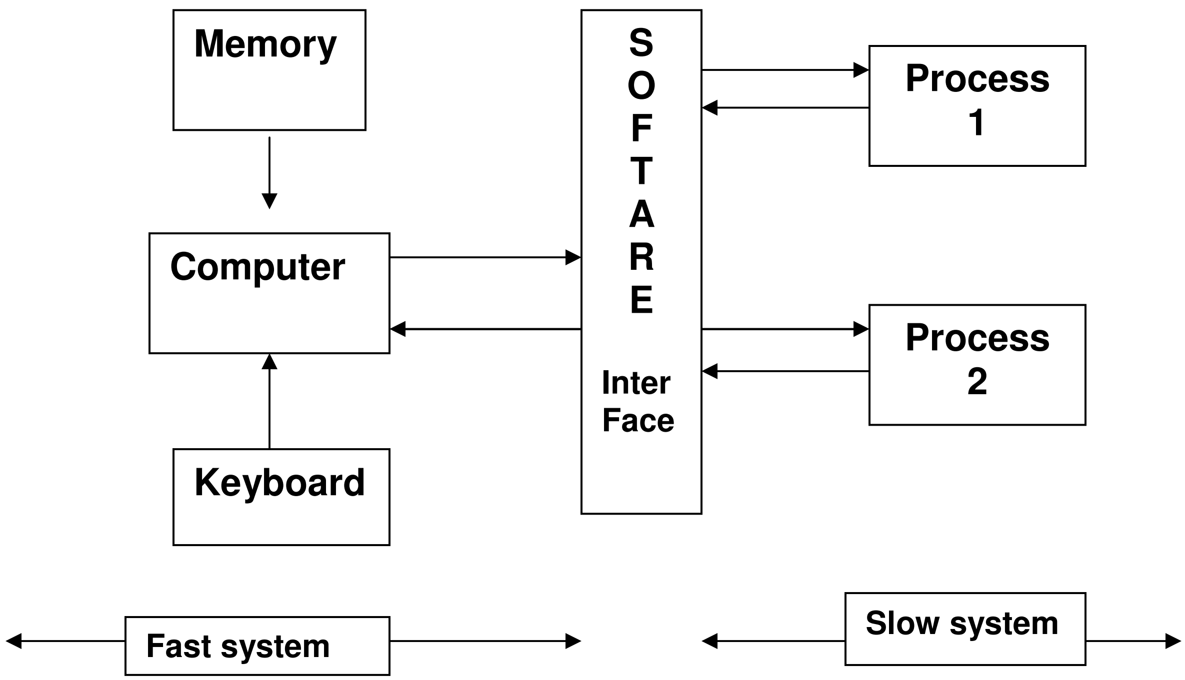
CONTENTS

Abstract

Title:

Study of Software interface for Adaptive control system

1) **Introduction:**



Software interface will synchronize the two processors

This is the basic software interface of the personal computer in which two processes that is fast system consisting of memory, computer, keyboard and process which is slow system is synchronized with software interface.

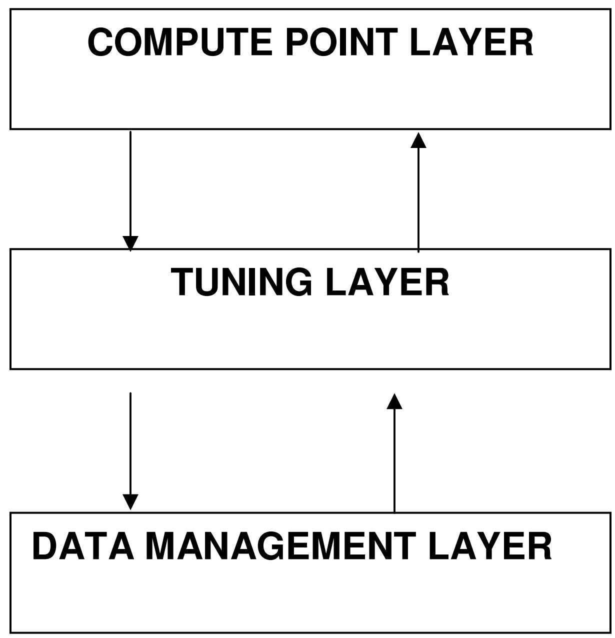
2) **Simulation:**

The software interface for adaptive control system has been analyzed.

The results for first order system have been analyzed, plotted and interpreted for various initial conditions. The graph has been available in matlab.

3) **Layered architecture:**

This stage takes care of the total integrated software system. The layer architecture has been conceived which is shown below



In computer point layer we do mathematical modeling which has been described in simulation section. The results are stored into various routines which has been designated as private and public depending on the methods of data sharing .Tuning layer tunes the process control by performing various tasks such as checking for divergence etc. It takes data from compute point layer. Next comes the data management layer which is responsible to various functions namely Input/Output graphics display etc. So, various routines have been tried to get this layer into operation.

Conclusion:

The integrated software which has been developed here is expected to be class room model of a commercial system available in industry. Hence this has been submitted as Undergraduate project.

List of figures

CHAPTER 1

Introduction

Theory of Adaptive Control System

Theory of Layered Architecture

Water fall diagram of software development

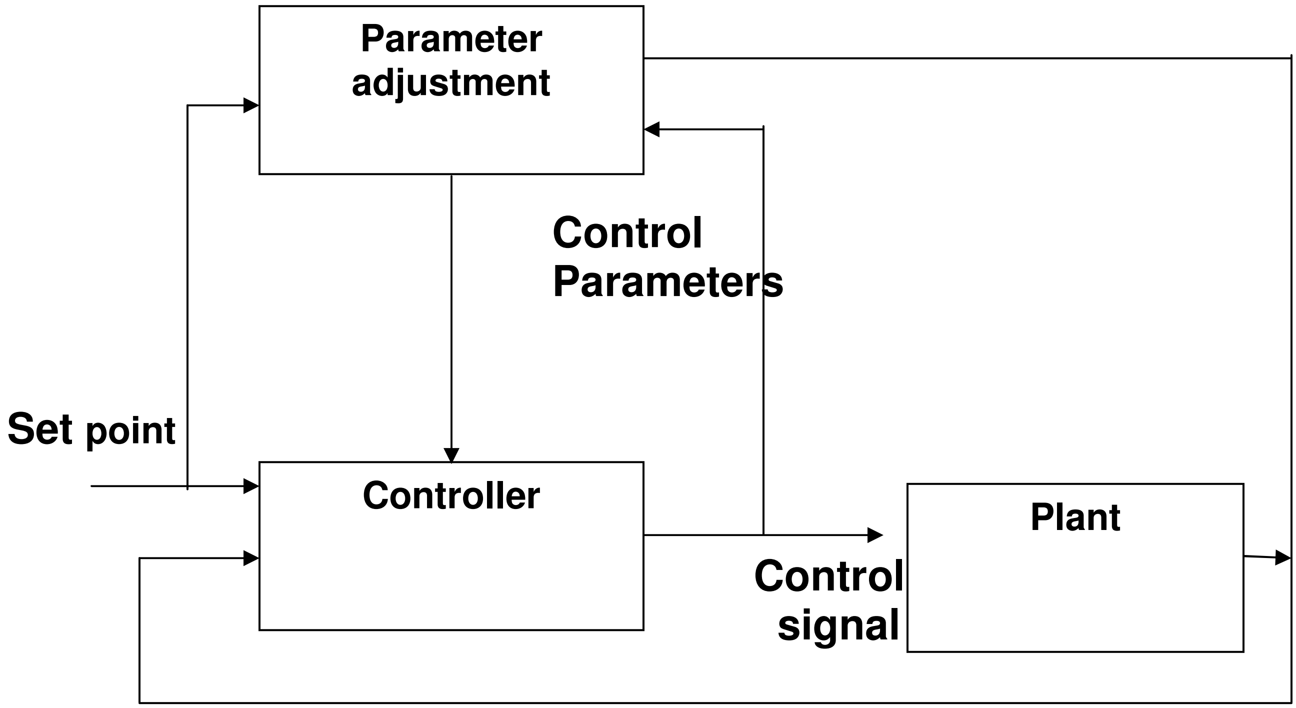
CHAPTER I

1.1 Introduction

Adaptive control is a special type of non linear feedback control in which the stages of the process can be separated into two categories which can change at different rates. The slowly changing states are viewed as parameter with fast time scale for the ordinary feedback and slower one for updating regulator parameters. One of the goals of adaptive control is to compensate for parameter variations which may occur due to nonlinear actuators ,changes in the operating conditions of the process and non stationary disturbances acting on the process.

1.2 Theory of adaptive control system :

An adaptive controller is a controller that can modify its behavior in response to changes in the dynamics of the process and the disturbances. Adaptive control can be considered as a special type of nonlinear feedback control in which the stages of the process can be separated into two categories, which can change at different rates. The slowly changing states are viewed as parameters with a fast time scale for the ordinary feedback and a slower one for updating regular parameters. One of the goals of adaptive control is to compensate for parameter variations, which may occur due to nonlinear actuators, changes in the operating conditions of the process, and non stationary disturbances acting on the process.



An adaptive Controller (fig. 1.1)

An adaptive controller is a controller with adjustable parameter and a mechanism for adjusting the parameters. An adaptive control system can be thought of as having two loops. One loop is normal feedback with the process (plant) and controller. The other loop is a parameter adjustment loop. A block diagram of an adaptive control system is shown above.

1.3 Theory of layered architecture:

A layered architecture is designed as a hierarchy of client –server process that minimizes interaction between layers. Each layer act as a client for the module above it and acts as server from the model below it in a layered architecture .Layered architecture has been used in data base systems, operating systems, computer to computer communication systems and layered control for robots. The digital VAX, virtual memory system (VMS) architecture is an example.

The user layer of VMS is the only layer visible to the system user. It provides the tools, editors, compilers and application packages needed by users. The supervisor layers provide the command language interpreter (CLI), which provides an interface between users and inner layers of the operating system. Command lines are typed after the system prompt ‘$’ as in

```
$ Show time                    {display current date and time}
     3-5-2004                           12.00.00
```

The supervisor layer provides system services and a record management system. The kernel layer takes care of memory management, input/output, process and time management .This form of layering is very restrictive to prevent users from tempering with kernel and other system functions. It is security minded form of layering.

Kernel layer:

Handle input/output (devices), schedules and control process, manages memory.

Executive layer:

Provides record managements.

Supervisor layer:

Provides command language, interpreter (interface between user and inner layer of **VMS**).

User layer:

Provides utilities, application libraries, tools, programming languages.

Others, less restrictive form of layering are possible and have been introduced .A recent and successful alternative to VMS form of layered architecture has been developed by Brooks in organizing control systems for intelligent mobile robots.(Brooks ,1986).

BLOCK DIAGRAM OF LAYERED ARCHITECTURE

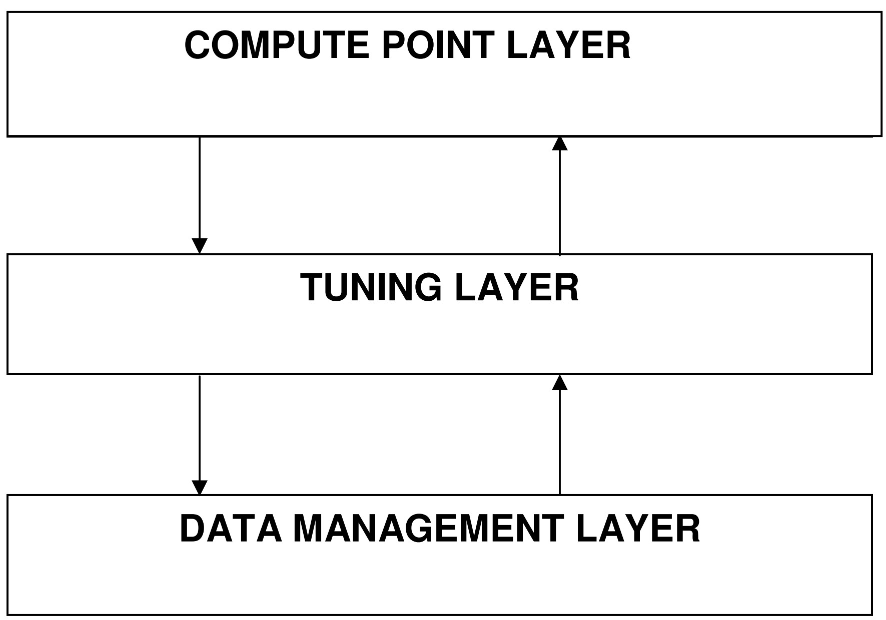


fig.(1.2)

Waterfall diagram of system Development

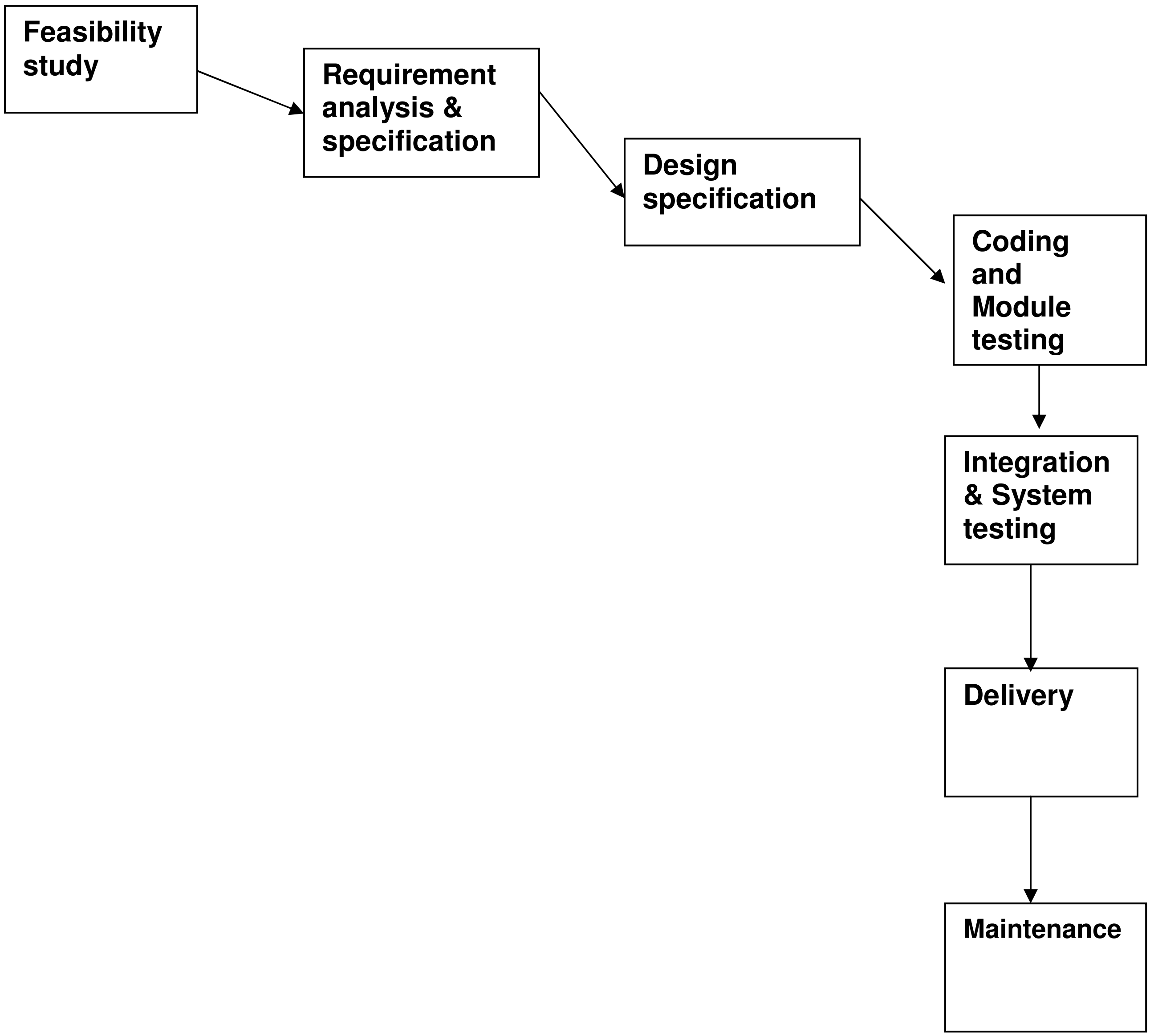


fig.(1.3)

CHAPTER 2

Mathematical model of Adaptive Control

Self tuning control

CHAPTER 2

2.1 MATHEMATICAL MODEL OF ADAPTIVE CONTROL:

Consider a first order system.

Reference model:

$dy_m/dt=-a_m y_m+b_m r$ ……………………………………(1)

Where r is the reference signal and am >0

Process plant:

$dy/dt=-ay+bu$ ……………………………………(2)

Controller:

$u=\theta_1 r-\theta_2 y$ ……………………………………(3)

where θ_1 and θ_2 are parameters

error:

$e=y-y_m$ …………………………………… (4)

Substitution of control law in the plant dynamics yields

$dy/dt=-ay+b(\theta_1 r-\theta_2 y)$ …………………………………… (5)

$\Rightarrow dy/dt=-(a+b\theta_2)\, y +b(\theta_1 r)$ ……………………………………..(6)

Subtracting reference model equation from process plant equation.

$dy/dt - dy_m/dt =de/dt$ …………………………………… (7)

$\Rightarrow de/dt=-a_m e-(b\theta_2+a-a_m)+(b\theta_1-b\theta m)r$ ……………………… (8)

From the error dynamics given in eqn above ,we can observe that the tracking error will go to zero if $b\theta_2=a_m-a$ and $b\theta_1 = b_m$. Thus the parameter should reach this goal.

Usually the Lyapunov function is quadratic in tracking error and controller parameter estimation error since it is expected that the adaptation mechanism will drive both the tracking error and error in estimation of controller parameters to zero. Thus a valid Lyapunav function for the error dynamics equation can be taken as

$$V(e,\theta_1,\theta_2) = \tfrac{1}{2} (e^2 +(1/b\gamma)(b\theta_2 + a - a_m)^2 + (1/b\gamma)(b\theta_1 - b_m)^2) \quad \ldots\ldots\ldots.(9)$$

Where $b\gamma>0$.

This function is zero when e is zero and the controller parameters are equal to correct values. For a valid Lyapunov function, the time derivative of the Lyapunov function must be negative. The derivative is given as

$$dV/dt = e\,de/dt + (1/\gamma)(b\theta_2 + a - a_m)d\theta_2/dt + (1/\gamma)(b\theta_1 - b_m)d\theta_1/dt \quad \ldots\ldots(10)$$

Substituting eqn. (8) in the above equation and rearranging we get,

$$dV/dt=-a_m e^2+(1/\gamma)(b\theta_2+a-a_m)(d\theta_2/dt-\gamma ye)+(1/\gamma)(b\theta_1-b_m)(d\theta_1/dt+\gamma re) \quad \ldots.(11)$$

If the parameter are updated as

$$d\theta_1/dt=-\gamma re$$

$$d\theta_2/dt =\gamma ye$$

Then $dV/dt=-a_m e^2$

Thus dV/dt is negative semidefinite which implies that $V(t)\geq V(0)$. This ensures that e must be bounded. To ensure tracking error goes to zero, We compute second time derivative of Lyapunov function.

$$d^2V/dt^2=-2a_m e\,de/dt$$

Substituting eqn. (8) in the above eqquationand rearranging we get,

$$d^2V/dt^2=-2ame(-ame-(b\theta_2+a-am)y+(b\theta_1-bm)r)= f(e,\theta_1,\theta_2,y,r)$$

Since all the parameters are bounded and $y=e+y_m$ is bounded , $d^2v/dt^2$is also bounded which in turn implies that dV/dt is uniformly continuous . Now using Barbalat's lemma, we conclude that tracking error converges to zero.

2.2 Self tuning Control:

Self tuning control has two essential components

1. Parameter estimation
2. Control law

Parameter estimation: Parameter estimation is a key element in self –tuner and is performed on-line. The model parameters are based on the measurable process input, process output, and state signals. A number of recursive parameter estimation schemes are employed for self –tuning control. The most population scheme is the recursive least squares estimation method.

Control law: The control law is derived based on control performance criterion optimization. Since the parameters are estimated on-line, the calculation of control law is based on procedure called certainty equivalent in which the current parameter estimates are accepted while ignoring their uncertainties. This approach of designing controller using estimated parameters of the transfer function process is known as indirect self-tuning method.

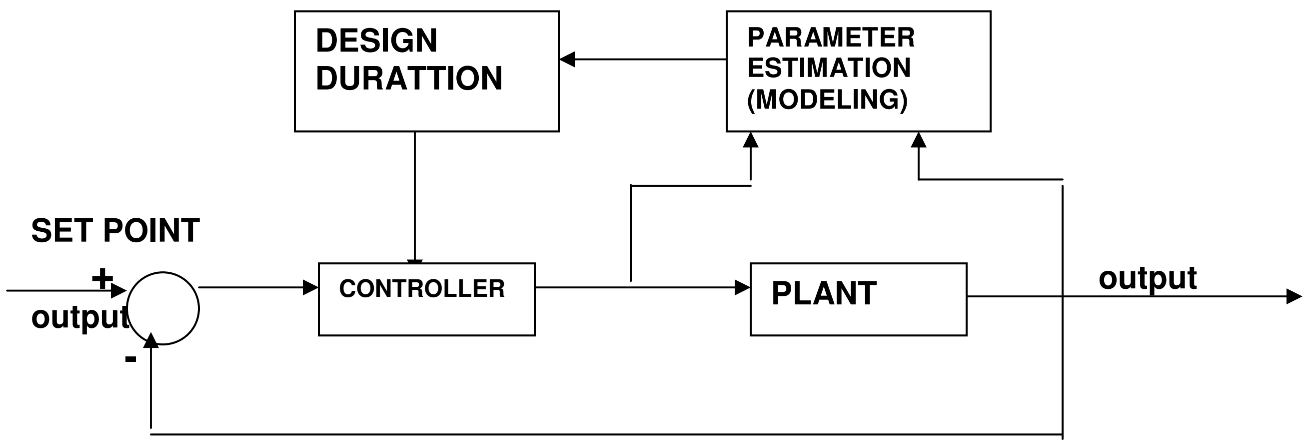


Fig. 2.1

A GENERAL CONFIGURATION OF SELF TUNING CONTROLLER

CHAPTER 3

Numerical analysis of Adaptive Control algorithm
Software modules developed
Module 1
Module 2

Chapter-3

3.1 Numerical analysis of adaptive control algorithm:

Reference model:

$dy_m/dt = -a_m y_m + b_m r$

where r is the reference signal and am>0.

Process Plant:

$dy/dt = -ay + bu$

Controller:

$u = \theta_1 r - \theta_2 y$

from the above two equations we get

$dy/dt = -ay + b(\theta_1 r - \theta_2 y)$

$\Rightarrow (y_{n+1} - y_n)/dt = (-a - b\theta_2) y_n + b\theta_1 r$

$\Rightarrow (y_{n+1} - y_n) = h[(-a - b\theta_2) y_n + b\theta_1 r]$

$\Rightarrow y_{n+1} = y_n + h[(-a - b\theta_2) y_n + b\theta_1 r]$

$\Rightarrow y_{n+1} = y_n (1 - h(a + b\theta_2)) + b\theta_1 r$ …………(1)

Similarly for reference model

$dy_m/dt = -a_m y_m + b_m r$

$\Rightarrow y_{m,n+1} - y_{m,n} = h(-a_m y_{m,n} + b_m r)$

$\Rightarrow y_{m,n+1} = y_{m,n} + h(-a_m y_{m,n} + b_m r)$

$\Rightarrow y_{m,n+1} = (1 - h a_m) y_{m,n} + h b_m r$ ……………(2)

Error:

$e = y - y_m$

$\Rightarrow e_{,n+1} = y_{,n+1} - y_{m,n+1}$

Parameters are updated as

$d\theta_1/dt=-\gamma re$

$=>(\theta_{1,n+1}-\theta_{1,n})/dt =-\gamma re_n$

$=>(\theta_{1,n+1}-\theta_{1,n}) =-\gamma re_n h$

$=>\theta_{1,n+1} = \theta_{1,n} - \gamma re_n h$ …………...(4)

$d\theta_2/dt=\gamma ye$

$=>(\theta_{2,n+1}-\theta_{2,n})/dt =\gamma y_n e_n$

$=>\theta_{2,n+1} = \theta_{2,n} + \gamma y_n e_n h$ …………...(5)

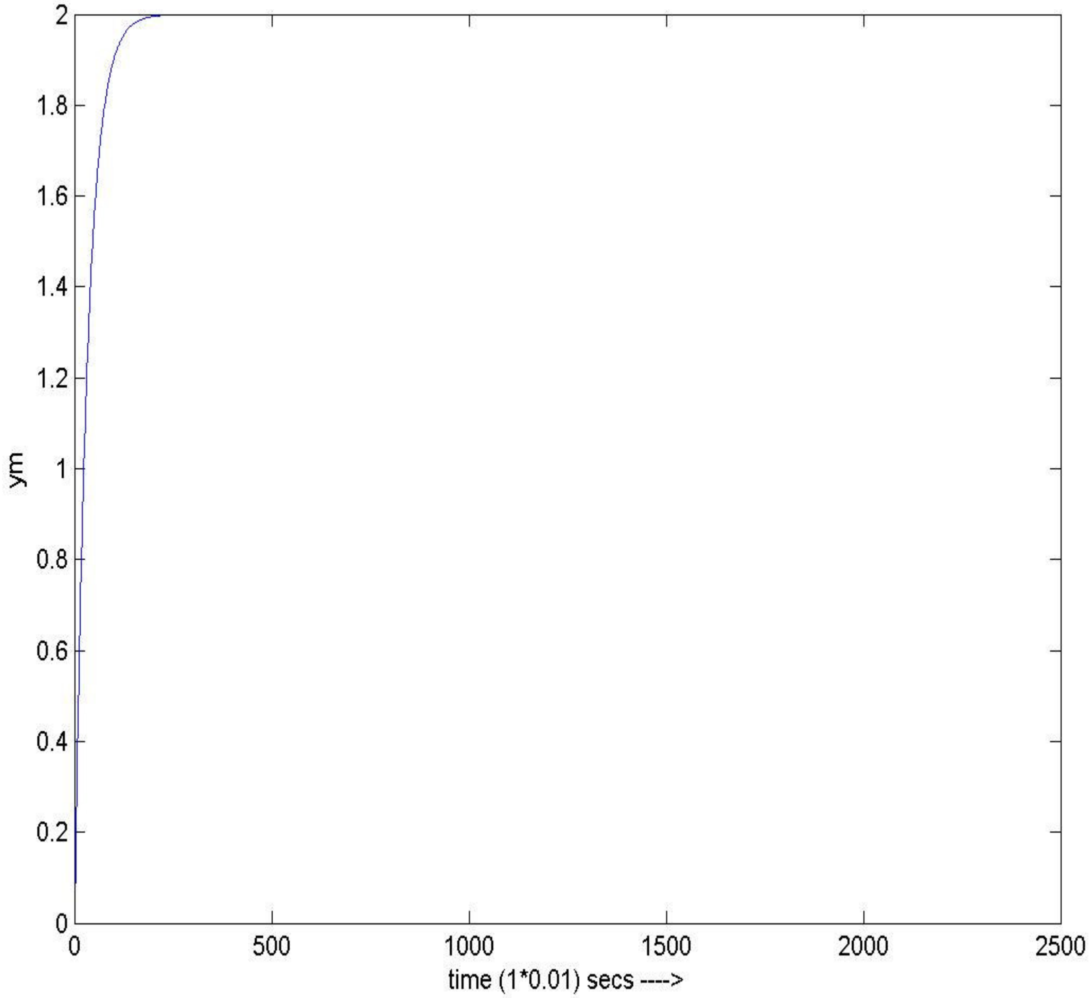
From the above equations software modules are developed. The parameters and variables are initialized appropriately.

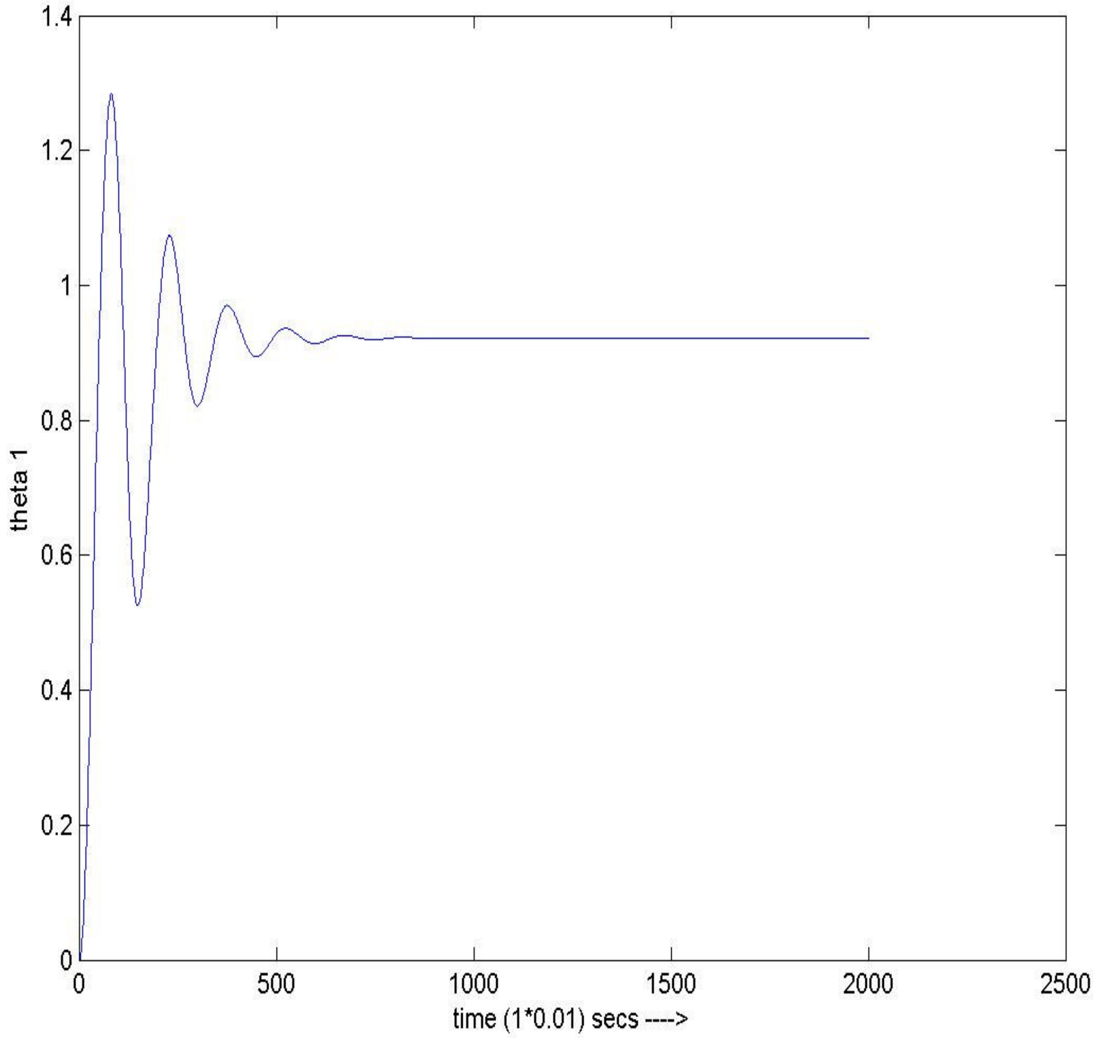
3.2 Software modules developed with explanation

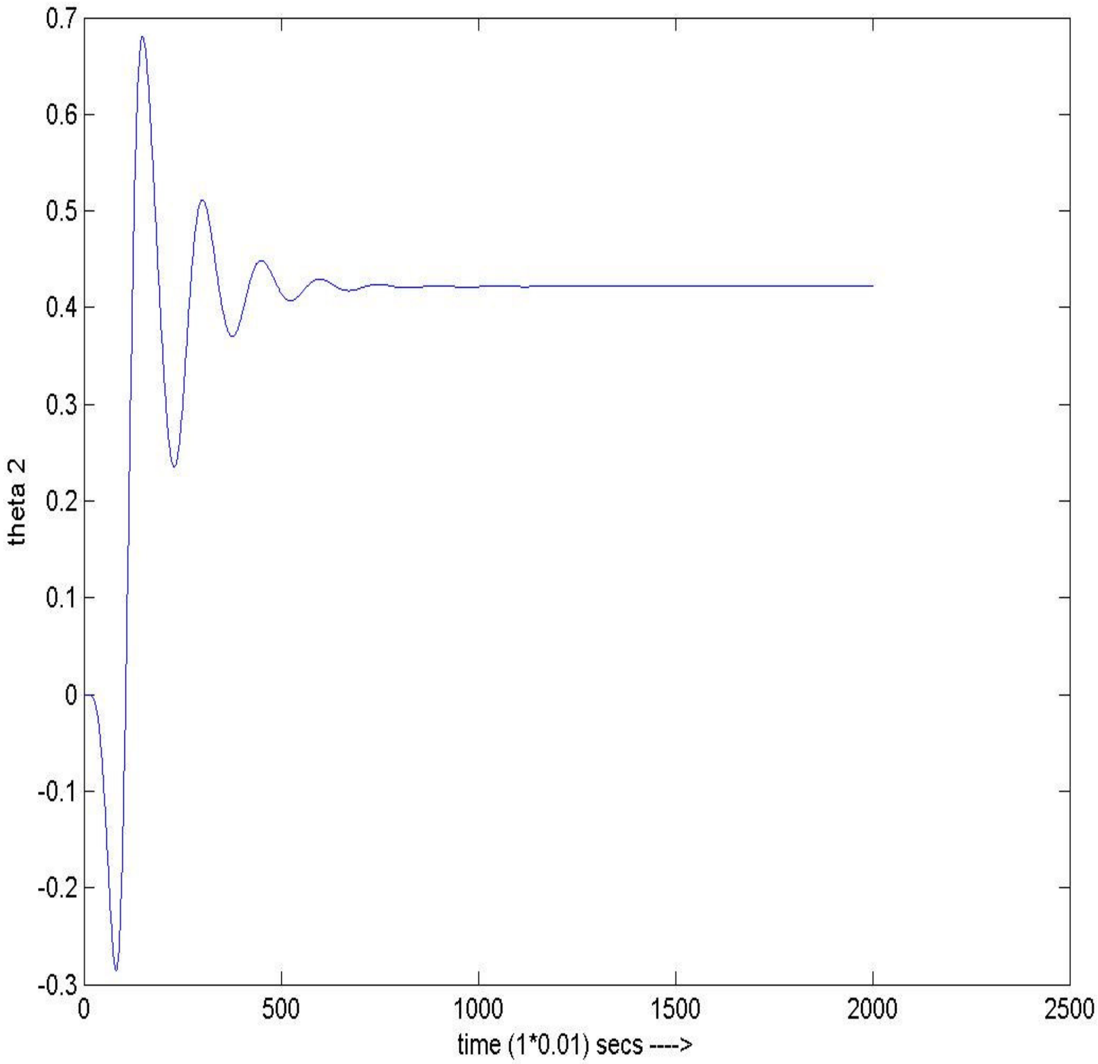
3.2.1 Module 1: The plant dynamics is given asdy/dt=-ay+bu, where a=1 and b=2.The plant response is required to follow the model given by $dy_m/dt = -3y_m + 3r$.The adaptation gain is chosen to be equal to 1.5. The initial values of both parameters of the controller are chosen to be zero. The initial conditions of the plant and the model are both zero. The reference signal r(t) =2 is used.

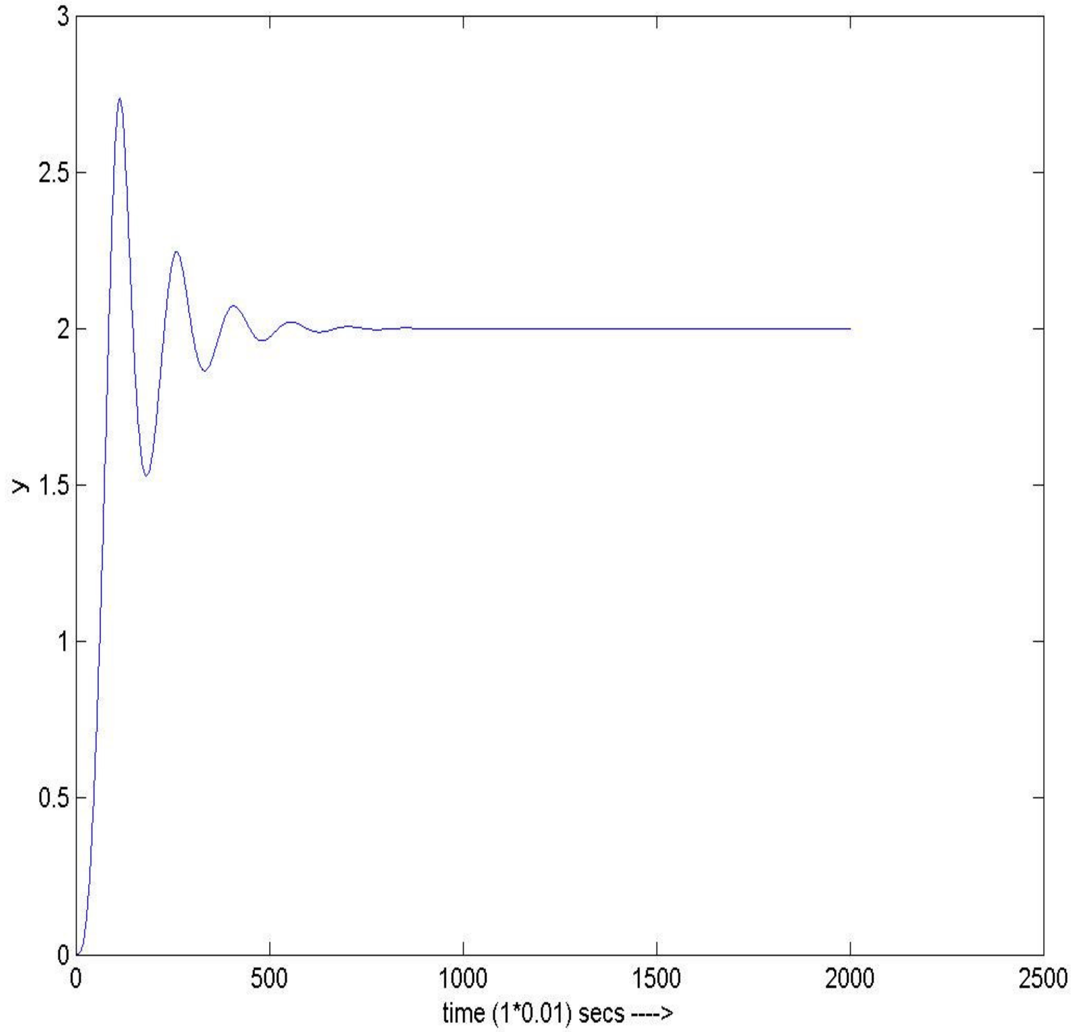
```
clear all;
clc;
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k=input('Enter the number of iterations: ');
ym= zeros(1,k+1);
y= zeros(1,k+1);
e= zeros(1,k+1);
theta= zeros(k+1,2);
h= 0.01;
am= 3;
bm= 3;
r= 2;
a=1;
b=2;
gama=1.5;

for i=1:k
    y(i+1)= y(i)*(1-h*(a+b*theta(i,2)))+h*b*theta(i,1)*r;
```

```
    ym(i+1)=((1-h*am)*ym(i))+h*bm*r;
    e(i+1)=y(i+1)-ym(i+1);
    theta(i+1,1)= theta(i,1)-gama*r*e(i)*h;
    theta(i+1,2)= theta(i,2)+gama*y(i)*e(i)*h;
end;

figure(1);
plot(ym);
figure(2);
plot(theta(:,1));
figure(3);
plot(theta(:,2));
figure(4);
plot(y);
figure(5);
plot(e);
```

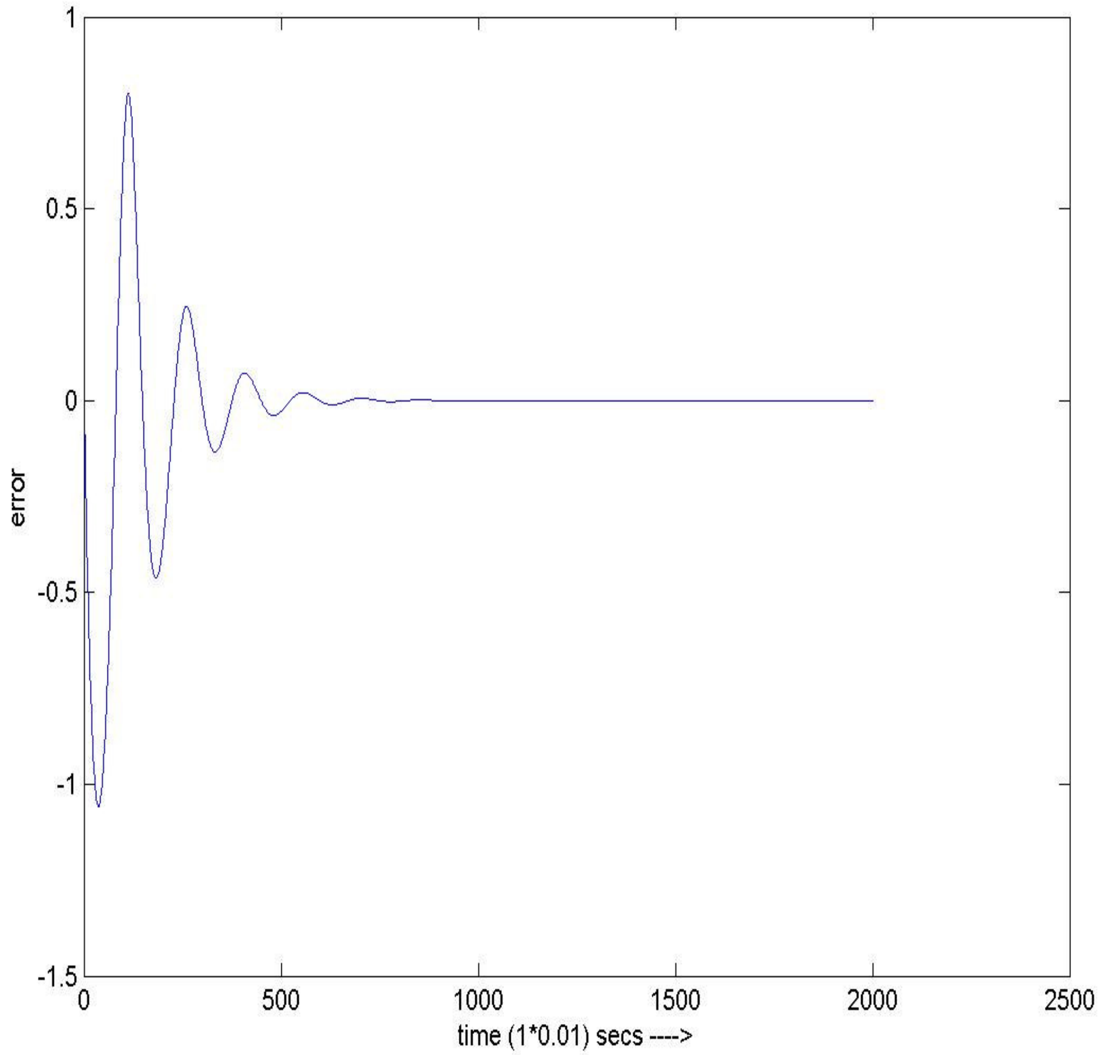

2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
ym
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->


1.4
1.2
1
0.8
0.6
0.4
0.2
0
theta 1
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
theta 2
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->



3
2.5
2
y 1.5
1
0.5
0
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->


1
0.5
0
-0.5
-1
-1.5
error
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->

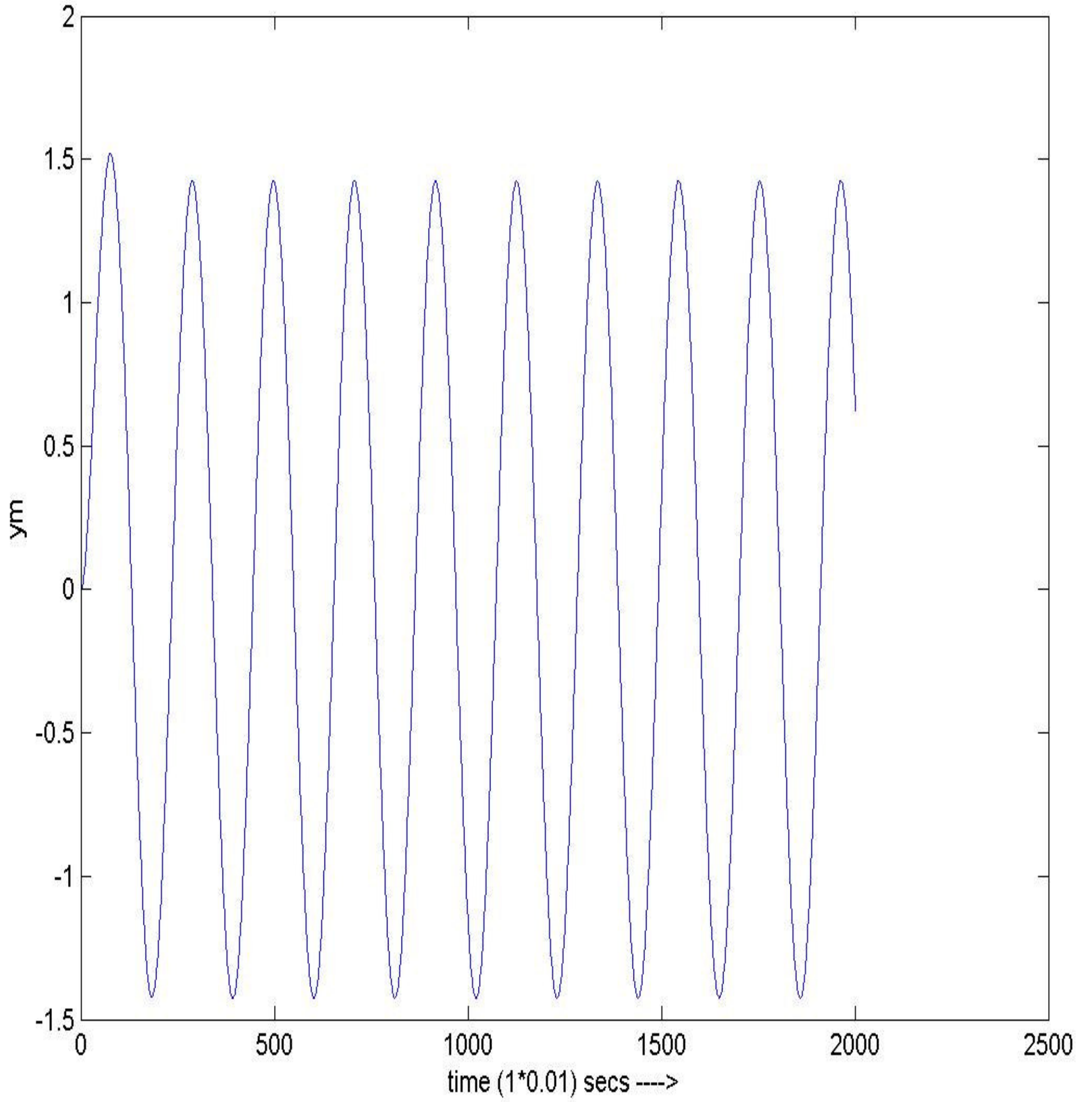
3.2.2 Module 2: In this module the problem statement is same as in the previous module. The reference signal used in this module is r(t)= 2*sin(3t).

```
clear all;
clc;
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k=input ('Enter the number of iterations: ');
ym= zeros(1,k+1);
y= zeros(1,k+1);
e= zeros(1,k+1);
theta= zeros(k+1,2);
h= 0.01;
am= 3;
bm= 3;
a=1;
b=2;
gama=1.5;
for i=1:k
   r=2*sin(3*h*i);
   y(i+1)= y(i)*(1-h*(a+b*theta(i,2)))+h*b*theta(i,1)*r;
   ym(i+1)=((1-h*am)*ym(i))+h*bm*r;
   e(i+1)=y(i+1)-ym(i+1);
   theta(i+1,1)= theta(i,1)-gama*r*e(i)*h;
   theta(i+1,2)= theta(i,2)+gama*y(i)*e(i)*h;
end;
```

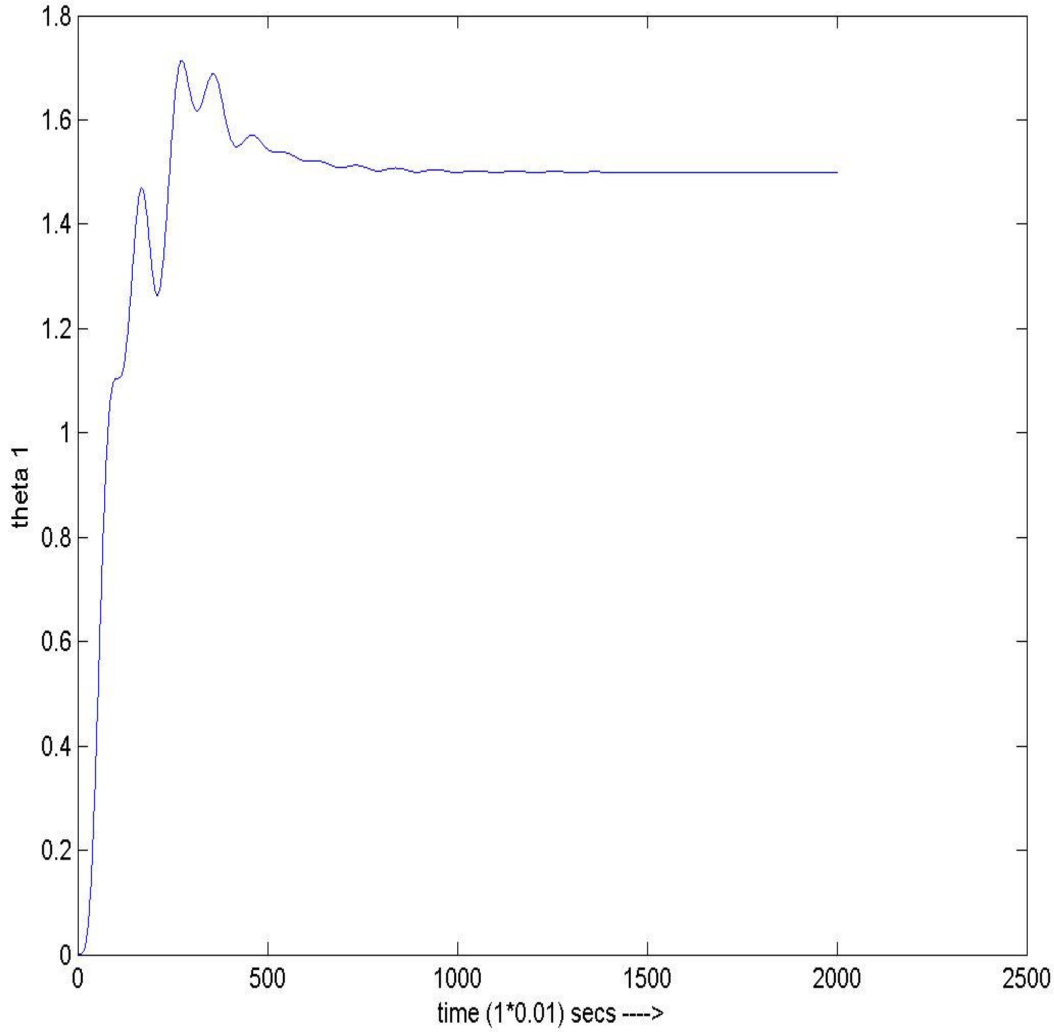


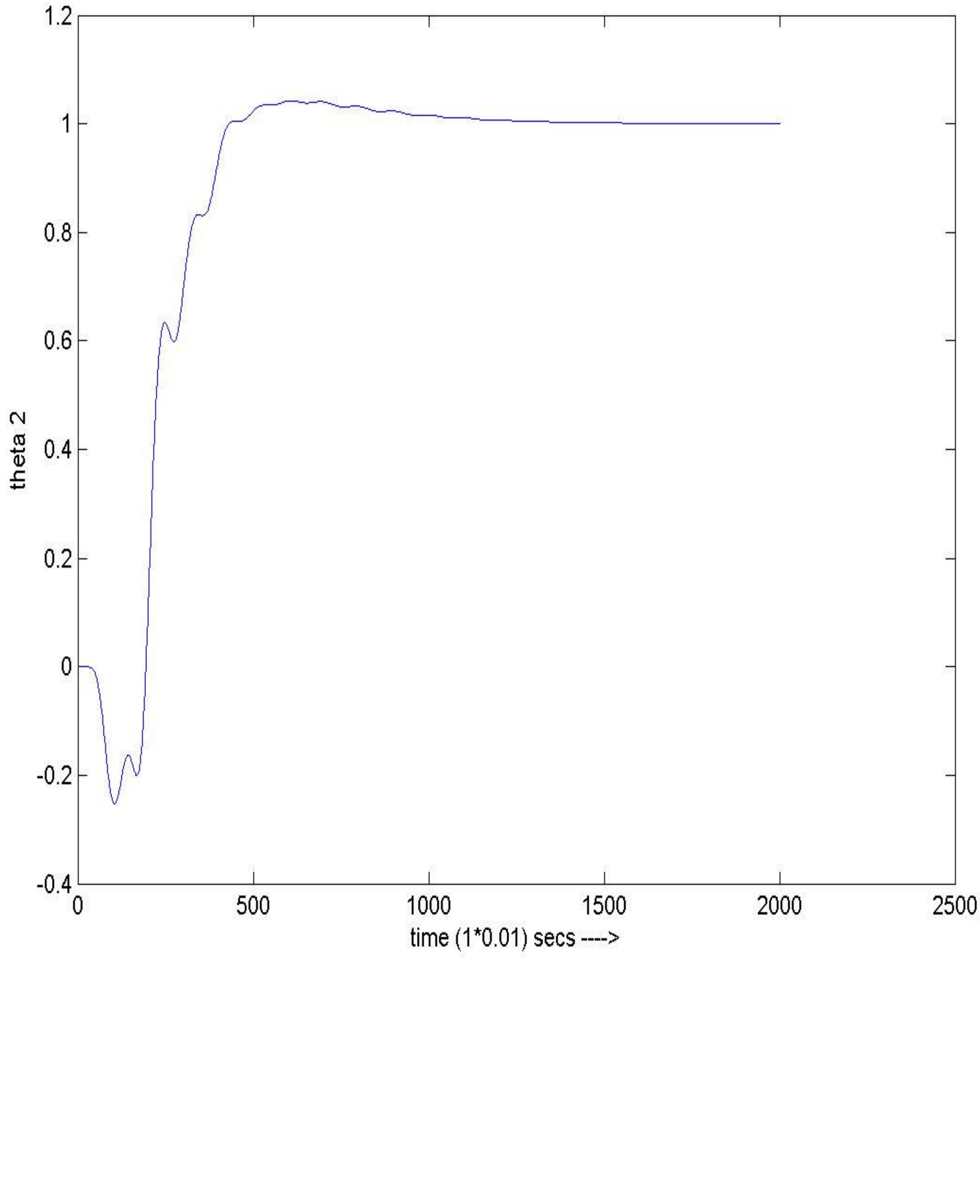
```
figure(1);
```

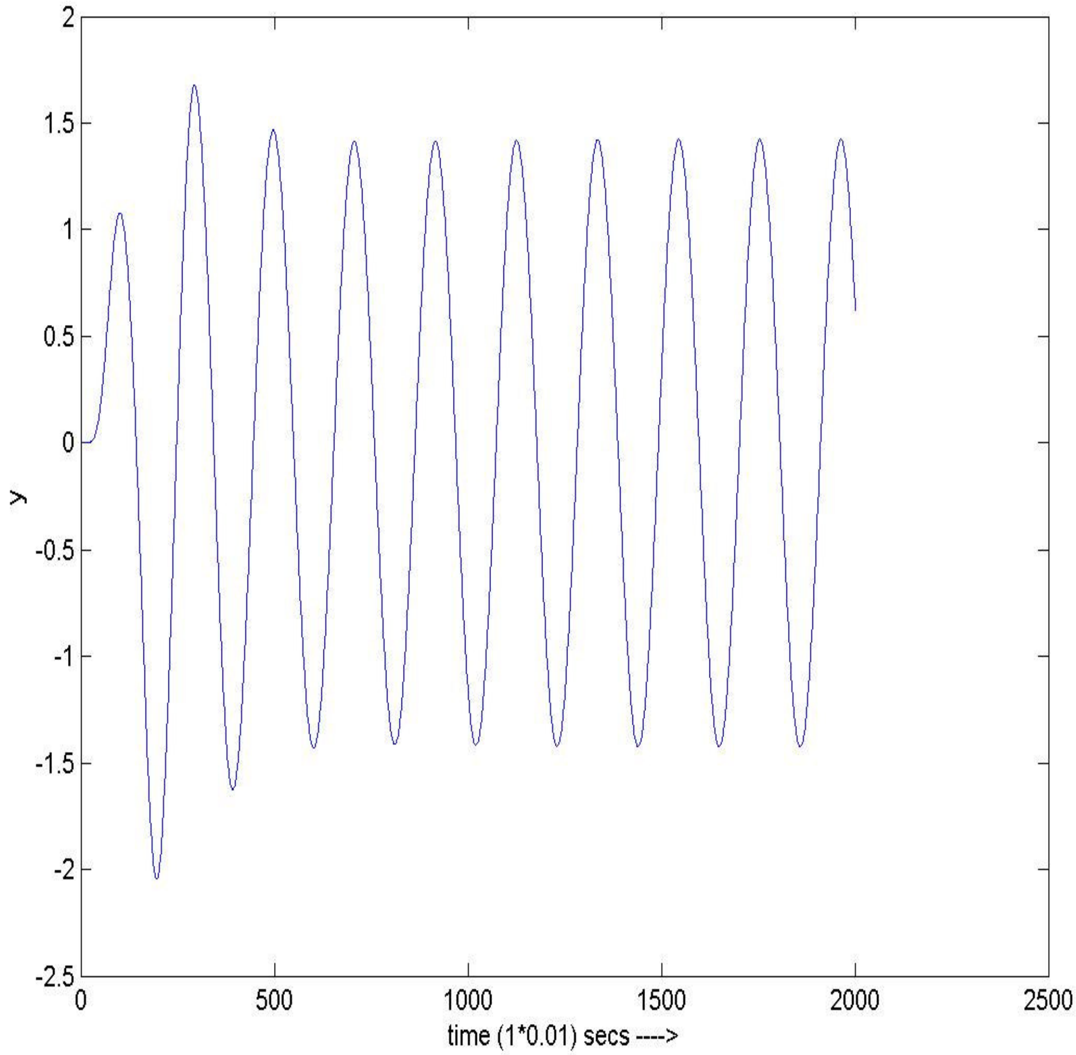
```
plot(ym);
figure(2);
plot(theta(:,1));
figure(3);
plot(theta(:,2));
figure(4);
plot(y);
figure(5);
plot(e);
```



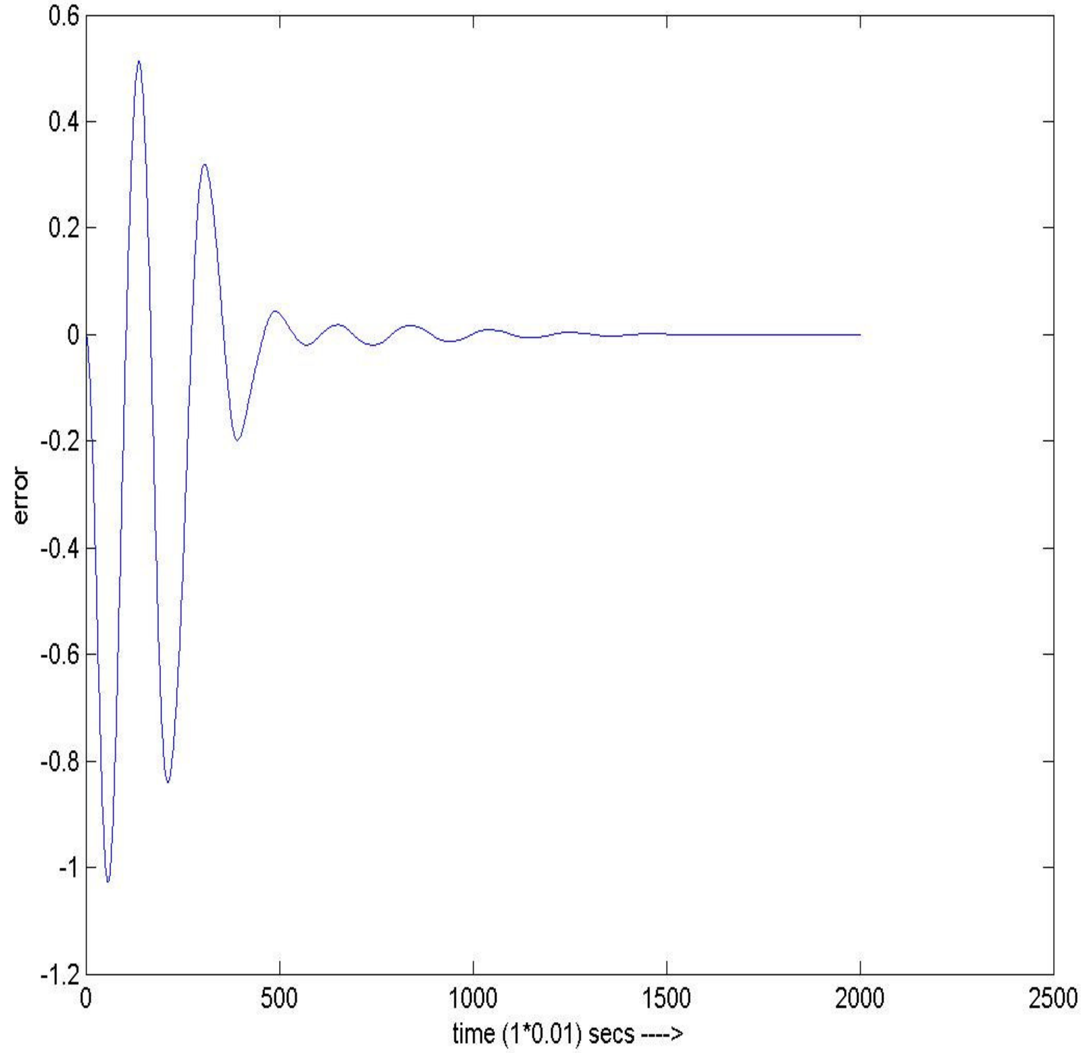
2
1.5
1
0.5
0
-0.5
-1
-1.5
ym
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->


1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
theta 1
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
theta 2
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->



2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
y
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->


0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
error
0
500
1000
1500
2000
2500
time (1*0.01) secs ---->

CHAPTER 4

Conclusion

Industrial Applications

Limitations of Existing Technology

Advantages of Adaptive Control

4.1 Concluding comment about the success of the project

The Model Reference Adaptive Controller (MRAC) for a first order system is illustrated using Lyapunov synthesis approach. We see from the simulation that the process plant follows the reference model and the tracking error converges to zero. The controller parameters also converge to their optimum values. Two modules are simulated using two different reference models.

4.2 Probable industrial applications of the project

1) Application in an autonomous underwater vehicle

Autonomous underwater vehicles (AUVs) are no longer engineering curiosities. They have been under development for over three decades and in last few years there have been significant advances towards their use in operational motions. All though ROVs play an important role in the offshore industries, their operational effectiveness is limited by the tethered cable and the reliance and the cost of some kind of support platforms. This paper investigates the applications of process control architecture to control the yaw angle of an AUV.

2) Application in batch chemical reactor

In a batch chemical reactor, the reactants are initially charged into the reaction vessel of the batch reactor and are then agitated for a certain period of time to allow the reaction to take place. Upon completion of the reaction, The products are discharged. For a specific reaction there is an optimum temperature profile according to which the temperature of the reactor mass should be varying to obtain best results. Automatic temperature control is achieved by adaptive control systems.

4.3 Limitations of the existing technology

There are several problems with the existing technology. These include:

1) Impulse and step response models over parameterized and limit application of the Algorithm to strictly stable process
2) Tuning is required to achieve normal stability
3) Hardware limitations

Impulse and step response models are known to be over parameterized .The dynamics of the first order process ,for example ,can be described by three numbers using a parametric response will typically require from 30 to 120 coefficients to describe the same dynamics. These difficulties can be over come at the identification step by first identifying a low order model and then calculation of the impulse coefficients however the impulse response still requires more storage space than is necessary. A potentially more significant problem with the impulse and step response model is that they are limited to strictly stable process while it is certainly possible to modify the algorithms to accommodate a pure integrator, these modification may lead to other problems, such as adding the derivative of a noisy output signal in the feedback bath.

4.4 Advantages of the adaptive control

High performance control systems can be implemented when the plant dynamic characteristics are poorly known or when large and unpredictable variations occur. An adaptive controller that does not require any knowledge about the signs or bounds of the high frequency gain is proposed for a scalar plant. Adaptive control compensates for parameter variations, which may occur due to nonlinear actuators, changes in the operating conditions of the process, and non stationary disturbances acting on the process.

REFERENCES

1. Gopal Nagrath, “Control system engg”

2. Peters J.F. & Pedryez W, “Software Engg. -An engineering approach”

3. Prakap Rudra, “Matlab”