**ASSIGNMENT 4: MATLAB FOR ENGINEERS**

**Name: Harshith**

**Branch:Cse**

**Regno:160905008**

**ANN BASED CONTROLLER**

The feedback control has been the basis for the development of simple automatic control systems.

Feedback control systems are inherently simple to comprehend and implement but are not very effective

in compensating plant parameter variations and changes in the environment.

Direct Inverse control is the simplest and most widely used control configuration which makes use

of the inverse plant model.The network is required to be trained offline to learn the inverse dynamics

of the plant.Thus the overall transfer control function of the controller and the plant becomes unity,

that is =1.Therefore ,output is equal to the reference input or in other words,output tracks the reference input.

%% WATER BATH TEMPERATURE CONTROL SYSTEM

The objective of the article is to implement an ANN controller for a temperature control system.The water

bath system consist of water tank in which cold water is entering the tank from one side and hot water is

leaving from the other side.The aim is to obtain the output water at the desired temperature.The specifications

of the water bath system considered for the given problem are as follows:

\* capacity of the water tank

\* Flow rate of inlet/outlet water

\* Wattage of base heater

\* Rotational speed of stirrer

\* Transport delay incorporated

\* Temperature of inlet water

The equation of the Water Bath system are as follows:

dT/dt=F/V(T1-T)+Q/VcpP;

where

T is the tank temperature

F is the flow rate

T is the inlet temperature

V is the volume of the tank

Q is the heat input

cp is the specific gravity

P is density of water

%%implementation of ANN controller

There are 4 steps in controller implementation

They are:

1) Generate input-output training data

2) Create network object.

3) Train the network.

4) Simulate the network controlled system

%% GENERATE THE INPUT OUTPUT TRAINING DATA

% Program to generate input/output data for the water bath heater

n=0;%intitialize the counter

for a = 0:0.06:0.6%increment the heat input at equal intervals

%0.057 Kcal is equivalent to 250 watt

for i=1:4 %index varaible

Tp=45+rands(1,1)\*20;%initialize the Tp between 25-65 C

Ti=25+rands(1,1)\*2;% initialize the Ti between 23-27 C

F=(1/60)+rands(1,1)\*0.008;% initialize tank fluid flow rate

x0=[Tp];% initialize output rate

%solving tank differential equations using runge kutta method

options=[];

%find response at different time instants

tspan=[0 30 120 150 240 270 390 420 540 570];

%heaterwb defines the differential equation of waterbath system

fun='heaterWB';

[t,x] = ode45(fun,tspan,x0,options,Ti,a,F);

%storing the differential input values in arrays to form input

for j=1:2:9

n=n+1;

rTp(n)=x(j,1);%store previous tank temperature

rT(n)=x(j+1,1);%store output tank temperature

rTi(n)=Ti;%store inlet tank temp

rQ(n)=a;%store heat input

rF(n)=F;%store tank fluid flow rate

end

end

end

%store the training input data for the controller

trinC=[rF;rTi;rT;rTp];

%store the output training data for the controller

troutC=[rQ];

%save the training data set in MAT file traindataC1

save traindataC1 trinC troutC;

step2: create network object

net=newff(trinC,troutC,10,{'tansig','tansig'},'traingd');

%trinC ,troutC are training data inputs and outputs

% 10 is the number of neutrons in the hidden layer

%{'tansig','tansig'} is a cell array containing the names of the transfer functions

% traingd is the name of the training function

**program to define weights of artificial neural network controller**

% for the water bath temperature control system , to load the training data saved in program load traindataC1

% define the neural network no of neurons in each layer

% The data preprocessing functions are defined by default while

% creating the network using newff

net= newff(trinC,troutC,10,{'tansig' 'tansig'},'traingd');

**Training of ANN controller**

% to train the neural network as controller for the water Bath system

% to load the training data saved in program load traindataC1;

% to create the neural network

net=newff(trinC,troutC,10,{'tansig' 'tansig'},'traingd');

% define the training algorithm parameters

% only 4 paramters are defined ,rest are taken as default values

net.trainParam.show=50;

net.trainParam.lr=0.05;

net.trainParam.epochs=30000;

net.trainParam.goal=0.0001;

%train network with training algorithm function traingd

net=train(net,trinC,troutC);

**PERFORMANCE ANALYSIS**

%Program for controller simulation for multi-step tracking response for the water bath temperature control system is as follows:

%using trained neural controller obtain the multi step response

%of the water bath system

clear all; % to clear workspace of all variables or data

load traindataC1; % load the training data saved in program 1

net=newff(trinC,troutC,10,{'tansig' 'tansig'},'traingd');%create the neural network

%initialize the variables

n=0;

F=1/60;

Ti=25;

Tp=25;

Q=0;

T=25;

%store the initial values of output tank temperature ,and heat input to water bath in an array

n=n+1;

T(n)=Tp;

q(n)=Q;

for s=1:1:120

if s<40

Tr=30;

end;

if and (s>=40,s<40)

Tr=40;

end;

if s>=80

Tr=50;

end;

Trr(s)=Tr;

%put the inputs as an input vector

pnew=[F;Ti;Tr;Tp];

%scale the input data

[trinCn,trinCs]=mapminmax(trinC);

[troutCn,troutCs]=mapminmax(troutC);

pnewn=mapminma('apply',pnew,trinCs);

%simulate the network controller and obtain the output

tnewn=sim(newt,pnewn);

%to descale the output of the controller

tnew=mapminmax('reverse',tnewn,troutCs);

Q=tnew;

%if 0.6>Q>0,keep Q with in limits as heat cannot be

%negative or more than the rating of the heater element

if Q>0.6

Q=0.6;

end;

if Q<0.6

Q=0;

end;

%apply control input to plant and obtain output of hte plant

%using ODE solver

tspan=[30];

x0=[Tp];

options=[];

[t,x]=ode45('heaterWB',tspan,x0,options,Ti,Q,T);

[r,c]=size(x);

Tp=(r,1);

%store outputs obtained in an array

n=n+1;

T(n)=Tp;

q(n)=Q;

end;

%to plot the response

subplot(2,1,1);%to divide the figure window into 2 plots

plot(T,'r'); % to plot the output temperature to red color

hold on;

plot(Trr,'k:');% plot the setpoint reference tempsignal in black color

legend('MLP','setpoint',0); % write the text for 2 plots

ylabel('tank temp(C)');

subplot(2,1,2);

plot(q,'r');

xlabel('sample number');

ylabel('conrol input Q');

