Proceedings of the International Conference on Man-Machine Systems (ICoMMS)
11 – 13 October 2009, Batu Ferringhi, Penang, MALAYSIA

Design of Neural PID Controller For Reduced Order Model

[1]P.Gomathi, [2]T.Manigandan, [3]N.Devarajan
[1]Kongu Engineering College, Erode, India.
[2]School of Electrical Sciences, Kongu Engineering College, Erode, India.
[3]Government College of Technology, Coimbatore, India.

***Abstract*- The aim of this paper is to design a PID controller for higher order systems using the proposed model reduction method with neural networks. In a linear time invariant continuous system (LTICS) the coefficient matrix cannot be stored explicitly in a computer memory. The matrix vector products can be computed relatively in a less expensive manner by using an approximation technique. A novel method is proposed to obtain a reduced model from a higher order linear time invariant continuous system. The proposed scheme is simple, computationally straight forward and does not involve any complex algebra. The reduced order model will always be stable if the higher order system is stable. Using the proposed method, PID controller is designed for the higher order linear time invariant continuous system to meet the performance specifications. The PID controller parameters are tuned by using Back propagation Network.**

***Keyword*- Model Reduction, LTICS, Stability, PID controller, Tuning of controller.**

I. INTRODUCTION

The recent developments and the usage of processor in the analysis, synthesis, simulation, design and implementation of complex systems and several application involving signal processing, estimation filtering and controlling chemical plants, nuclear reactors and process industries increases the importance of reduced order models. The computational and implementation difficulties involved in design of optimal and adaptive controller for higher order linear time invariant continuous system can also be minimized with the help of suitable reduced order models. The reduced order model is designed with a PID controller and the performance is compared with an original higher order system. In the conventional method, the PID controller constants Kp, KI, and KD are tuned to meet the designer specification manually. This needs a lot of experience to adjust these constants. Neural Network is trained for this purpose of tuning the parameters of the PID controller.

A. Statement of the Problem

Consider an nth order stable linear time invariant single input single output (SISO) system described by the transfer function

$$G(S)=\frac{\sum_{i=0}^{n-1} a_i s^i}{\sum_{i=0}^{n} b_i s^i} \quad ; a_i>0;\ b_i>0 \qquad \text{---(1)}$$

Where a_i $(0 \le i \le n-1)$ and b_i $(0 \le i \le n)$ are scalar constants.

The corresponding reduced order model is of the form

$$R_r(S)=\frac{\sum_{i=0}^{r-1} d_i s^i}{\sum_{i=0}^{r} e_i s^i} \quad ; d_i>0;\ e_i>0 \qquad \text{---(2)}$$

Where d_i $(0 \le i \le r-1)$ and e_i $(0 \le i \le r)$ are scalar constants.

The original system described by equation (1), the problem is to find a reduced order model in the form of equation (2) such that the reduced order model retain the important characteristics of the original system and approximates its response as close as possible for the same type of inputs.

B. Algorithm for the Proposed Method of Model Reduction

Step 1: Find the poles (P) and Zeros (Z) for the given transfer function.

Step 2: Arrange poles(P) and zeros(Z) as $Z_1^2<P_1^2<Z_2^2<P_2^2<\ldots\ldots$

Step 3: Choose 'n' as the order of the given system

Step 4: Consider the denominator D(S) of the transfer function

Step 5: Reciprocate the denominator of the original system $s^n.D(1/s)=a_{0,0}s^n+a_{0,1}s^{n-1}+\ldots\ldots\ldots\ldots\ldots+a_{0,n-1}.s+a_{0,n.}$

Step 6: The denominator of the transfer function is divided into even and odd functions using stability equations.

$D_e(s)=a_{0,0}\prod_{i=1}^{k_1}\left[1+\frac{s^2}{z_i^2}\right]$

$D_o(s)=a_{0,1}s\prod_{i=1}^{k_2}\left[1+\frac{s^2}{p_i^2}\right]$

Where $k_1=n/2$ and $k_2=(n-1)/2$

Step 7: Construct the denominator of the reduced transfer function as

$D_r(s) = D_e(s) + D_o(s)$

Step 8: Choose e0 – coefficient in Dr(s)

e_1 - coefficient of s in $D_r(s)$

e_2 - coefficient of s^2 in $D_r(s)$

Step 9: The general second order transfer function in continuous time domain is given by

$R_r(s) = \frac{d_1(s)+d_0}{e_2s^2+e_1s+e_0}$

Step 10: Equate the transfer functions G(s) and Rr(s)

Step 11: Cross multiplying and rearranging the equations obtained in step 10 for the same powers of 's' on both sides we obtain the set of equations in terms of d_1, d_0, e_1 and e_0.

Step 12: Using the values of e_2, e_1 and e_0 obtained in step 8 compute the values of d_0 and d_1.

Step 13: Obtain the reduced second order equation $R_2(s)$ using the values of e_2, e_1 and e_0 obtained in step 8 and the values of d_1 and d_0 obtained in step 12.

C. Algorithm for the Design of PID Controller for Ltics

Step 1: Read the open loop transfer function of the given higher order system

Step 2: Form the closed loop transfer function

Step 3: Obtain the step response of the closed loop system

Step 4: Check the response for the required specifications.

Step 5: If the specifications are not met, get the reduced order model (By using proposed method of reduction) and design a controller for the reduced order model..

Step 6: Obtain the initial values of the parameters K_P, K_I and K_D by pole zero cancellation.

Step 7: Cascade the controller with the reduced order model and get the closed loop response with the initial values of the controller parameters.

Step 8: Find the optimum values for the controller parameters which satisfy the required specifications by using Back propagation algorithm.

Step 9: By applying the optimum values , cascade this controller with the original system.

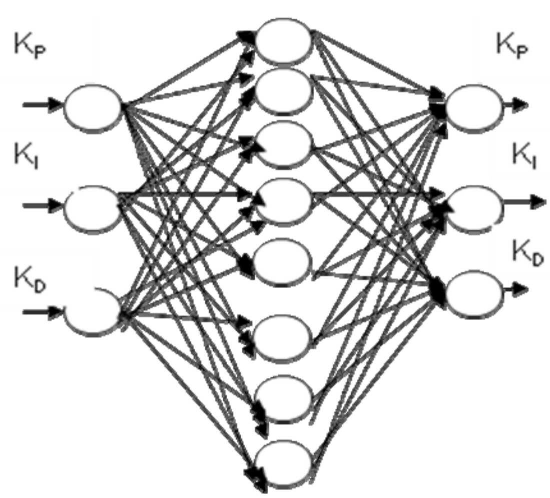
Step 10: Obtain the closed loop response of the original system with the controller.

Step 11: Obtain the closed loop response of the reduced order system with the controller.

D. Need for Neural PID Controller

From the reduced order model the values of K_P, K_I and K_D are calculated using pole placement technique. The step response graph is plotted. In order to meet he designer specification the controller parameters are need to be tuned. Tuning process is a time consuming process and it also needs a lot of experience. To avoid this, Neural Network is used for tuning. By using Back propagation algorithm the controller parameters are tuned.

The network is given with 3 input parameters K_P, K_I and $K_{D.}$. The network is tried with different numbers of hidden layers. Based on the error produced and number of epochs required, 8 hidden nodes are selected for the BPN structure. The network is trained for number of epochs. For tuning it is selected as 1000 epochs based on the error produced.



E. Back propagation Algorithm

Step 1: Initialize the input layer

Step 2: Propagate activity forwarded for all the layers

Step 3: Calculate error in output layer

Step 4: Back propagate the error

Step 5: Update the weights and biases

Step 6: Display the final value of K_P, K_I and K_D

II. ILLUSTRATIONS

Example 1

$$G(s) = \frac{35s^7+1086s^6+13285s^5+82402s^4+278376s^3+511812s^2+482964s+194480}{s^8+33s^7+437s^6+3017s^5+11870s^4+27470s^3+37492s^2+28880s+9600}$$

Poles are: -1, -1±j, -4, -3, -5, -8, -10

Zeros are: -1.0346±j0.6310, -2.6349, -3.8345, -4.9021, -7.8014, -9.7845

Arrange poles (P) and zeros (Z) as

$$Z_1^2 < P_1^2 < Z_2^2 < P_2^2 < \ldots\ldots$$

1.4685 < 1.9994 < 6.9427 < 9 < 14.7034 < 16 < 24.0306

$a_{0,0} = 9600$; $a_{0,1} = 28880$

$n = 8$; $k_1 = 8/2 = 4$; $k_2 = (n-1) / 2 = 7/2 = 3.5 = 3$

$$D_e(s) = a_{0,0} \prod_{i=1}^{k_1} \left[1 + \frac{s^2}{z_i^2}\right]$$

$$= 9600[1+0.6810s^2][1+0.144s^2]\ [1+0.068s^2]\ \ [1+0.0416s^2]$$

Neglecting higher order terms,

$$D_e(s) = 9600+8672.16s^2$$

$$D_o(s) = 28880s\ (1+0) = 28880s$$

$$D_r(s) = D_e(s) + D_o(s) = 8672.16s2+28880s+9600$$

Choose the values of e_2, e_1 and e_0 as,

$e_2 = 8972.16$; $e_1 = 28880$ and $e_0 = 9600$

$$\frac{8s^2+6s+2}{s^3+4s^2+5s+2} = \frac{d_1(s)+d_0}{e_2s^2+e_1s+e_0}$$

Cross multiplying and rearranging we get,

$$35e_2 = d_1$$

$$35e_1+1086e_2 = 33d_1+d_0$$

$$35e_0+1086e_1+13285e_2 = 437d_1+33d_0$$

$$1086e_0+13285e_1+82402e_2 = 3017d_1+437d_0$$

$$82402e_0+278376e_1+511812e_2 = 27470d_1+11870d_0$$

$$13285e_0+82402e_1+278376e_2 = 11870d_1+3017d_0$$

$$78376e_0+511812e_1+482964e_2 = 37492d_1+27470d_0$$

$$194480e_0 = 9600d_0$$

By substituting the values of e_2, e_1 and e_0 we get d_0 = 194480 and d_1 = 12667.911.

Therefore the reduced order model will be,

$$R_r(s) = \frac{35s+587.326}{s^2+17.32s+26.5237}$$

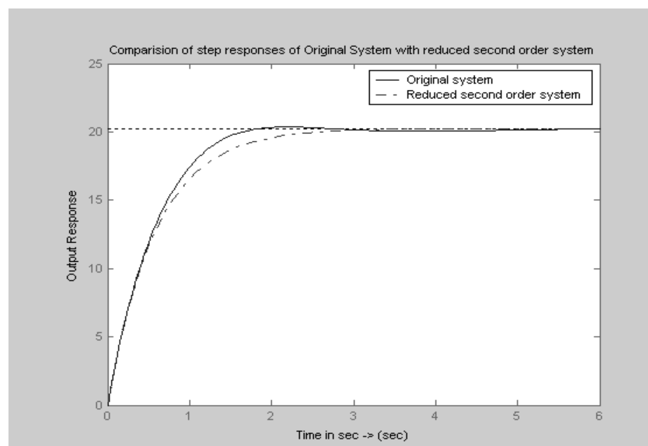


Fig 1. Comparison of step responses of original system and reduced order system

To meet the designer specification controller is designed. By using pole placement technique initial values of the PID controller are taken as

$K_P = 17.32$; $K_I = 26.5237$; $K_D = 1$;

By using the BPN algorithm of Neural Network the controller parameters are tuned and the final values obtained are

$K_P = 21.88$; $K_I = 34.97$; $K_D = 1$;

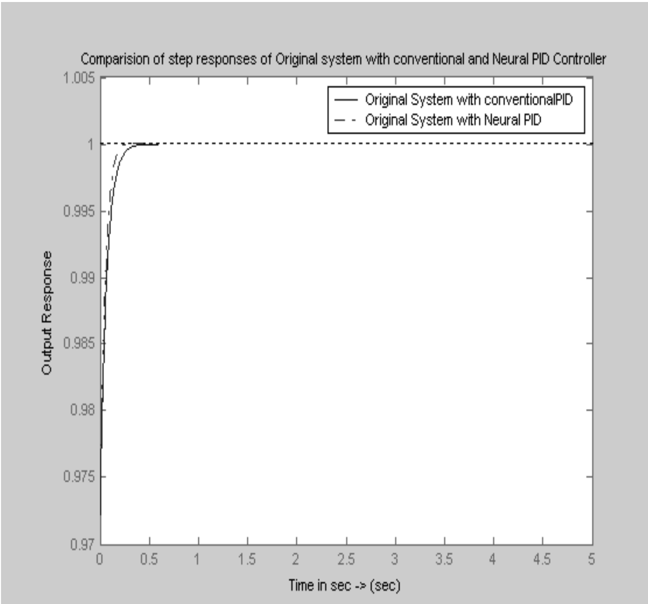


Fig. 2. Response of reduced order system with conventional and Neural PID controller

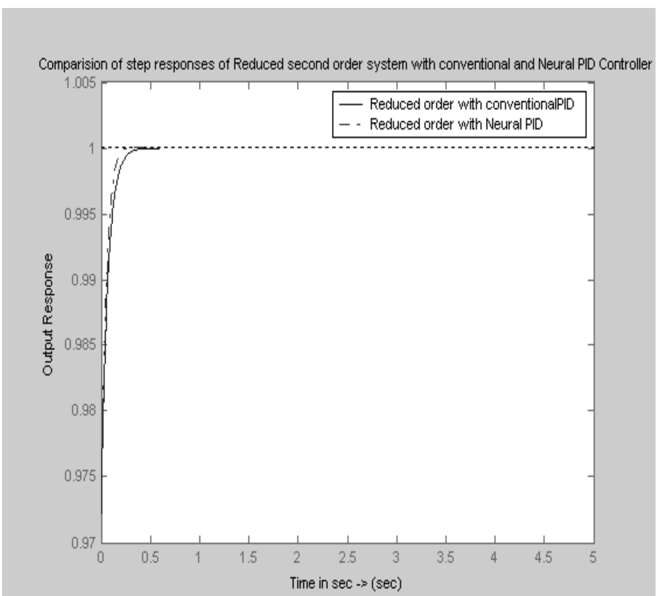


Fig. 3. Response of reduced order system with conventional and Neural PID controller

Details of Back Propagation Network

Number of input and output nodes	: 3
Number Hidden Nodes Selected for BPN Structure	: 8
Number of Epochs	: 1000

III. CONCLUSION

In this paper a simple method for obtaining the reduced order models of a large scale linear time invariant stable system is presented. The proposed method of model reduction is computationally simple and does not involve much procedure. It gives a stable reduced system if the original system is stable. The PID controller is designed and tuned for the original system and Reduced order system using pole zero cancellation method. This method needs a lot of experience and also takes a lot of time. In order to avoid this, a Back Propagation Network is designed and the controller parameters are tuned using BPN algorithm of Neural Network. The tuned values of these controller parameters are attached with the original system and its closed loop response for a unit step input is found to be in good agreement with the response of reduced order model. The results obtained are with 8 hidden nodes and 1000 epochs.

REFERENCES

[1] Nagaraj B., Subha S., Rampriya B., (2008) 'Tuning Algorithms for PID controller using Soft Computing Technique' International Journal of Computer Science and Network Security vol. 8 No.4.
[2] Gomathi P., Manigandan T., Devarajan N., (2005), 'An Algebraic Method of Linear System Model Reduction', Proceedings of NACPED-05,conducted by Government College of Engineering, Thirunelveli. pp. 19-22
[3] Kiam Heong Ang. Gregory Chong, (2005), 'PID Control System Analysis, Design and Technology', IEEE transactions on Control Systems Technology Vol. 13, No.4, pp. 559-576.
[4] Frank Blomilling and Heinrich Voss, (2004) 'Exploiting model order reduction technique for solving rational Eigen problems'
[5] Mittal A.K., Prasad. R., Sharma S.P.,(2004), 'Reduction of LINEAR Dynamic Systems using an Error minimization Technique' IE(I) Journal-EL Vol. 84, pp. 161-165

TABLE 1
COMPARISON TABLE

	Original system	Reduced order system	Original System with Conventional PID controller	Original System with Neural PID controller	Reduced System with Conventional PID controller	Reduced System with Neural PID controller
Rise time (Sec)	1.06	1.2	0.141	0.114	0.14	0.11
Settling time(Sec)	1.56	1.6	0.29	0.229	0.26	0.18
%Peak overshoot	0.519	-	-		-	-
Peak Time(Sec)	2.21	-	-		-	-