Neural Networks based PID Controller

Dr. Turki Y. Abdalla
College of Engineering
University of Basrah

Abstract

This paper considers the neural network based PID controller. The learning and generalization properties of neural network are utilized in improving the performance of a conventional PID controller. Two different schemes are introduced. Both schemes are studied and their performances are comparatively evaluated on an example for uncertain system.

متحكمة نوع PID تعتمد الشبكات العصبية

الدكتور تركي يونس عبدالله
قسم هندسة الحاسبات
كلية الهندسة/ جامعة البصرة

الملخص

يتعلق هذا البحث بالمتحكمة نوع PID المعتمدة على الشبكات العصبية. استخدمت صفات الشبكات العصبية في التعلم والتعميم في تحسين اداء متحكمة PID الاعتيادية. تم تقديم نوعان من متحكمات PID المعتمدة على الشبكات العصبية. تم دراسة وتقييم اداء النوعين على مثال لنظام غير مؤكد.

Introduction

A compensator that generally used in process control industry is the proportional plus integral plus derivative (PID) controller. It provides proportional, integral and derivative control action for improving a system performance. The transfer function of this controller $G_c(s)$ is given by

$$G_c(s) = K_p + K_d s + (K_i/s)$$

where

K_p, K_d and K_i are the proportional, derivative and integral gains respectively.

PID controller can be designed for linear systems [1,2]. Also methods are available for nonlinear systems [1].

The design process involves finding values of gain K_p, K_d and K_I such that all design specifications are satisfied. Such controller will be effective enough if the accuracy requirement of the control system is not critical. The usual way to optimize the control actions is to tune the PID coefficients [1], but this can not cope with a varying control environment or system uncertainty and nonlinearly. Fuzzy PID controller [3] is an approach for coping with environmental variation and system uncertainty and nonlinearity. The fuzzy algorithm is based on intuition and experience, and can be regarded as a set of heuristic decision rules [3,4]. Also genetic algorithm is used in tuning PID controller [5].

In this paper, a neural network (NN) based PID controller is proposed. The learning and generalization properties of neural network are utilized in improving the performance of conventional PID controller. Two different schemes are introduced. In the first scheme, the structure of PID controller is fixed, and three simple neural networks are used to tune the three coefficients of the PID controller. In the second scheme, one neural network is used for modifying the structure of PID controller, and tuning its coefficients so that the controller can cope with varying control environment or system uncertainties.

Neural Network

A neural network is a collection of processing elements (or neurons), each neuron has two basic parts, the summation function and the threshold function [6] as shown in Fig.(1).

The output y can be expressed by

$$y = f(\sum_{i=1}^{n} w_i x_i)$$

where f is a nonlinear function such as ON-OFF, or sigmoid.

Neurons are interconnected with adjustable weighted links and can be arranged in different ways. Neural networks learn to give correct output by training: Entering the input and the desired output patterns, assigning the initial weight values to the connections within the network, then it will adjust those weights over and over until it gives a correct output. Backpropagation (BP) is a learning algorithm for multilayer feed forward neural network[6]. This algorithm can handle any problem that requires pattern mapping from input pattern to output pattern. The activation function in the BP algorithm is a sigmoid function, which is differentiable. The weight w_{ij} (from node i to node j) is updated using BP algorithm as:-

$$w_{ij}(new) = w_{ij}(old) + \eta\delta_j x_i$$

Where η is the learning rate and δj is the difference. It is defined by the following [6] :-

For output neurons

$$\delta_j = y_j(1 - y_j)(y_d - y_j)$$

And for hidden neurons

$$\delta_j = y_j(1-y_j)\sum_k \delta_j w_{jk}$$

Where y_d is the desired output and y_j is the actual output of the output neuron.

The operation of neural network involves two phases, the forward phase and the backward phase. During the forward phase, the input is presented and propagated toward the output, while during the backward phase, the errors are formed at the output and backpropagated toward the input.

Many well-known results state that any continuous function can be approximated using a two layer neural network with appropriate weights [6,7]. This is known as the neural network universal approximation property. In the proposed controller, the net acts as a universal approximator. To simplify the learning algorithm, the first layer weights are selected randomly and will not be adjusted. The second layer weights are adjustable. The approximation holds [8] for such neural network.

Neural network based PID controller

In this approach, a conventional PID controller is designed, based on a nominal model of the process to be controlled, the output of this controller is fed as input for a neural network. The weights are adjusted so that the desired performance specifications are satisfied.

Two schemes of the proposed controller are adopted. The first scheme is shown in Fig.(2). In this scheme, structure of PID controller is fixed and the three neural networks each with one input are employed for tuning the coefficients of the conventional PID controller. The second scheme is shown in Fig.(3). In this scheme one neural network with three input is used in modifying the structure of the controller. The neural network is trained off-line using the input-output pair. The weights are adjusted during training so that the performance index

$$J = \sum_{K=0}^{N} e(kT)^2$$

is minimized

Where $e(kT) = y(kT) - y_d(kT)$

Here $y_d(kT)$ is the desired output, and $Y(KT)$ is the actual output of the system.

Simulation results

In order to study the performance of the two approaches described above, a series of simulation study have been performed, using a numerical example for uncertain system described by, the model,

It is assumed that the desired output is

$$G(s) = \frac{K(s+1)}{(S+a)(s^2+6s+10)}$$

$$with\ K = 1 \pm 0.05$$

$$a = 2 \pm 0.1$$

given by

$$y_d(t) = 1-(1+3t)e^{-3t}$$

The aim is to design a neural network based PID controller that minimize the performance measure J given by

$$J = \sum_{K=0}^{N} e(kT)^2$$

Where $e(kT) = y(kT) - y_d(kT)$

and $y(kT)$ is the actual output

The operation time $T1 = NT = 5$ sec, and sampling time $T = 0.01$ sec.

From conventional PID design method, the approximate values of the parameters of the controller based on nominal system model are evaluated as $k_p=2$, $k_i=2$, $k_d=5$.

First scheme

As shown in Fig.(2), three neural network, each with one input unit, ten hidden units and one output unit are used. Back propagation training algorithm is used. Only the second layer weights (w_i, i=1,2, ..., 10) are adjustable.

The initial weights are chosen randomly between −1 and +1. The first layer weights (V_i, i=1,2, ..., 10) are fixed. Table 1. Show the weights after training for 10000 epochs. The value of the performance index J=0.006.

Table (1):Final weights after training for the first scheme.

a- NN1

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
0.21	0.42	0.63	0.21	0.42	0.63	0.21	0.42	0.63	0.42
w_1	w_2	w_1	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.078	0.156	0.234	0.312	0.390	0.469	0.547	0.626	0.704	0.782

b- NN2

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
0.21	0.42	0.63	0.21	0.42	0.63	0.21	0.42	0.63	0.42
w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.063	0.125	0.188	0.251	0.314	0.376	0.439	0.502	0.565	0.627

c- NN3

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
0.21	0.42	0.63	0.21	0.42	0.63	0.21	0.42	0.63	0.42
w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.005	0.01	0.015	0.02	0.025	0.03	0.035	0.04	0.045	0.05

Second scheme

As shown in Fig.(3), one NN is used. It has three input units, ten hidden units and one output unit. Only the output layer weights (w_i, i=1,2, ..., 10) are adjustable. The initial weights are selected randomly between −1 and +1. Table (2) shows the values of the weights after training for 8000 epochs. The value of J=0.006.

Table (2): Weights values after training for the second scheme.

V11	V12	V13	V14	V15	V16	V17	V18	V19	V110
0.21	0.42	0.21	0.42	0.63	0.63	0.21	0.21	0.42	0.63
V21	V22	V23	V24	V25	V26	V27	V28	V29	V210
0.21	0.42	0.21	0.42	0.63	0.63	0.21	0.21	0.42	0.63
V31	V32	V33	V34	V35	V36	V37	V38	V39	V310
0.21	0.42	0.21	0.47	0.63	0.63	0.21	0.21	0.42	0.63

(a) Weights of first layer v_{ij}.

w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
2.114	2.29	2.45	2.59	2.73	2.87	2.95	3.04	3.12	3.22

(b) Weights of second layer w_i.

A summary of simulation results is shown in table (3). In this table, the first column shows the number of neurons (Nn) in both schemes, the second column gives the number of weights (Nw), while the third column shows the number of epochs (Ne) needed to get the same values of performance index J=0.006.

Results of simulation reveals that the second scheme is better than the first scheme, since it gives a small number of epochs.

Table (3): Summary of simulation results.

	Nn	Nw	Ne
1st ver.	33	33	10000
2nd ver.	11	31	8000

To test the performance of the proposed controller in the presence of model uncertainty or disturbance, assume that the parameters k and a in the process model may vary by ±5%. Fig.(4) and Fig.(5) show the step responses of the system in the presence of ±5% uncertainty in both k and a for the first scheme and second scheme respectively.

Conclusions

A neural network based PID controller is presented. Two scheme are proposed. In the first scheme, three NN are used in modifying the gains of conventional PID controller. In the second scheme, one NN is used in modifying the structure of PID controller. The proposed controller is robust and can be used for controlling uncertain or nonlinear systems. Results of simulation show good performance for the second scheme, since it gives the smaller number of neurons, weights and number of epochs.

References

[1] R. C. Dorfand, R. H. Bishop, "Modern control systems", Addison-wesley, 1998.

[2] J. A. Astrom and B. W. Henmark, "Computer controlled system", Prentice-Hall, 1997.

[3] M. P. Kevin and Y. Stephen, "Fuzzy control", Addison-Wesley,1998.

[4] Y. F. Li and C. C. Lau, "Development of fuzzy algorithms for servo system', IEEE control system magazine, Vol.9, No.3, 1983.

[5] M. Z. othman and M. J. Mohamed, "Different PID structures tuned by genetic algorithm', IJCCCE, No.1, Vol.2, 2001.

[6] J. M. Zurada, "Introduction to artificial neural systems', Jaico publishing house,1996.

[7] R. R. Selmic and F. L. Lewis, "Neural network approximation of piece wise continuous function', In soft computing and intelligent systems N. K. Sinh, Ed, 1998.

[8] L. Fausett. "Fundamental of neural networks, Prentice-Hall Int. Inc., 1999.

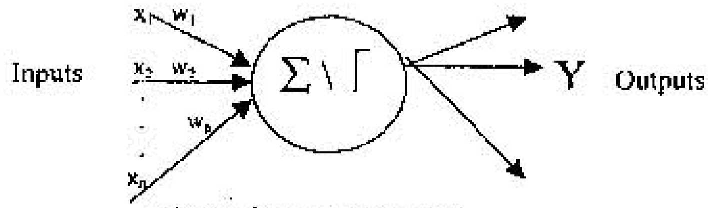


Fig.(1): Neuron Structure.

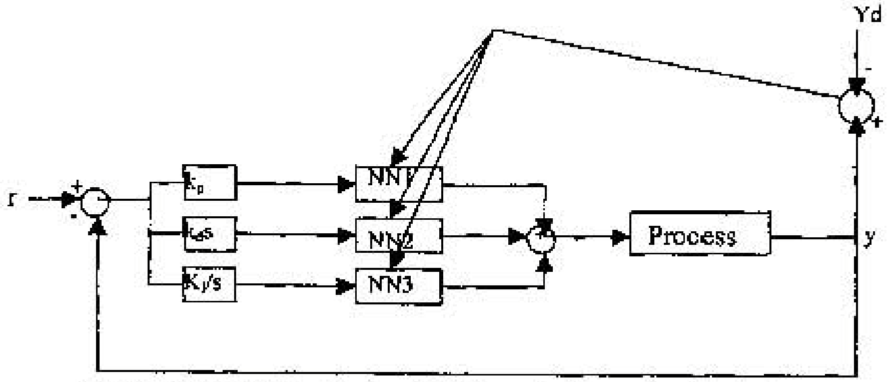


Fig.(2): Neural network based PID controller using first scheme.

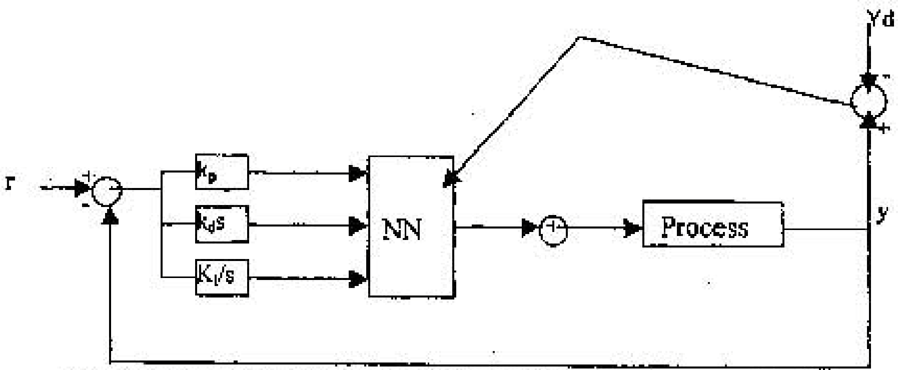
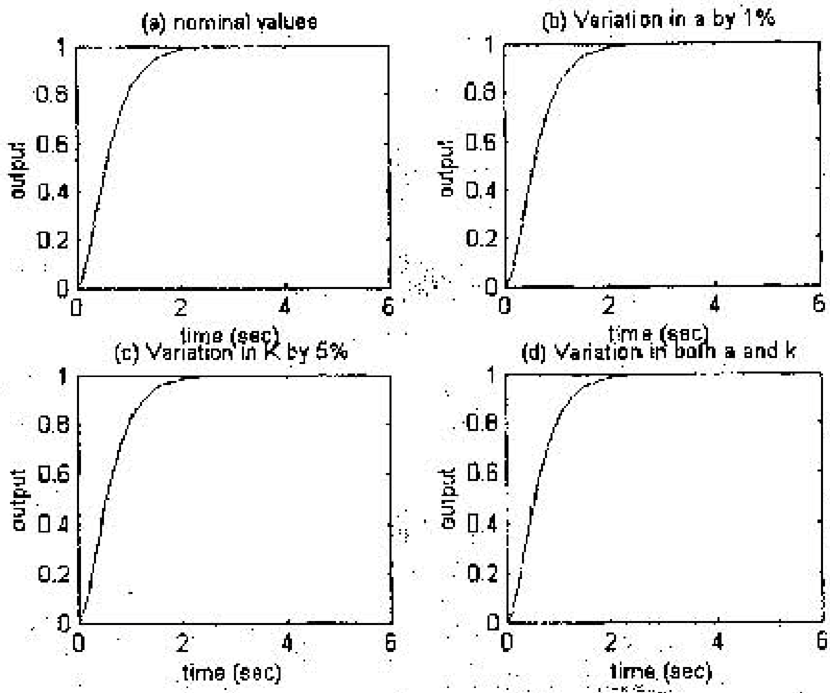
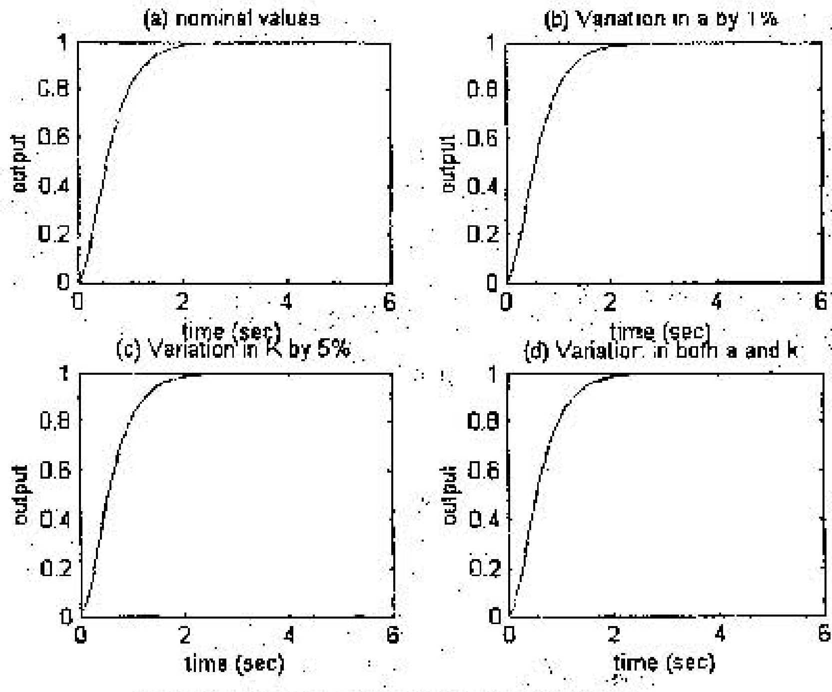


Fig.(3): Neural network based PID controller using second scheme.



Fig(4) Response of the system using first scheme



Fig(5) Response of the system using second scheme