15th Triennial World Congress, Barcelona, Spain

GA-NEURO-FUZZY CONTROL OF FLEXIBLE-LINK MANIPULATORS

M. N. H. Siddique and M. O. Tokhi

Department of Automatic Control and Systems Engineering,
The University of Sheffield, UK

Abstract: A typical method for rule reduction of a PID fuzzy controller is to divide the three-term into two separate PD and PI parts. A further reduction is possible if the controller is switched from PD to PI-type after a certain period of time. In that case only a single set of rules will be executed at a time and thus the controller rule base will be reduced. A further simplification is possible if a single rule-base is used for both the PD and PI-type FLC. This means that the fuzzy sets for change of error and sum of error will be redefined within the same universe of discourse, i.e., the fuzzy sets for both change of error and sum of error will be the same. Such a strategy is adopted in this paper. Accordingly the fuzzy sets are restored by tuning the scaling factors for change of error and sum of error using a single neuron network with non-linear activation function. Genetic algorithms are, on the other hand, used to train the neural network. The proposed method is tested and validated in the control of a single-link flexible manipulator. *Copyright © 2002 IFAC*

Keywords: Fuzzy logic, flexible manipulator, genetic algorithms, neural networks, PD-, PI-, PID-type control, vibration control.

1. INTRODUCTION

Though fuzzy logic controllers (FLCs) show superior applicability and performance over conventional controllers, they posses the same characteristics as traditional PD and PI controllers. The PD-type FLC adds damping to the system and has fast rise time and minimum peak overshoot but does not improve the steady-state error. On the other hand PI-type FLC adds damping and reduces steady-state error but yields penalised rise time and settling time. Methods have been proposed to reduce the steady-state error by fine tuning the rule bases, performing parameter optimisation and increasing the number of rules (Gurocak and Lazaro, 1994). Whereas techniques such as scaling factor adjustment, rule modification and membership function shifting are required to reduce the rise time and settling time (Maeda and Murakami, 1992; Zheng, 1992). A three-term PID-type FLC can obviously show a better performance in respect of rise time, peak overshoot, settling time and steady state error with a large rule base. Theoretically, the number of rules to cover all possible input variations for a three-term fuzzy controller is $n \times n \times n$, where n is the number of linguistic variables for each input (e.g. $7\times7\times7 = 343$). In practice the design and implementation of such a large rule base is a tedious task and takes a lot of memory and reasoning time. Different approaches have been proposed to reduce those difficulties (Brehm, 1994). A typical method for rule reduction of a PID fuzzy controller is to divide the three-term into two separate PD and PI parts. This hybrid PD and PI controller with n linguistic labels in each input variable require $n \times n + n \times n$ (e.g. $7\times7 + 7\times7 = 98$) rules, which is significantly smaller than that required for a PID controller (Chen and Linkens, 1998). A further reduction is possible if the controller is switched from PD to PI-type after a certain period of time. In that case only a set of rules will be executed at a time and thus the controller rule base will be reduced to only $n \times n$ rules (Siddique and Tokhi, 2000). Such a switching type FLC is developed for a flexible-link manipulator where a PD-type FLC is executed first and then switched to a PI-type FLC. A further simplification is possible if a single rule-base is used for both the PD and PI-type FLC. For this purpose the fuzzy sets for change of error and sum of error are redefined within the same universe of discourse. It means that the fuzzy sets for both change of error and sum of error are the same. This redefinition of the fuzzy sets for change of error and sum of error require an adjustment. In many cases, either tuning the scaling factors or adjusting the membership functions can obtain the same result and tuning the scaling factors is a simpler task than adjusting the membership functions. On-line adaptation is also an important issue in this case.

In order to tune the scaling factors, the integral and derivative gains, in an on-line manner a single neuron network with non-linear activation function is used. However, many parameters of the non-linear activation function, such as the optimum shape of the sigmoid function, are determined by trial and error. In this research, genetic algorithm used to optimise the shape, weight and bias of the network whereas the neural network itself is tuning the scaling factors. The developed controller is then applied to a single-link flexible manipulator to verify the performance of the controller and compare with results obtained by other approaches.

2. THE EXPERIMENTAL MANIPULATOR

The experimental rig constituting the flexible manipulator system consists of two main parts: a flexible arm and measuring devices. The flexible arm contains a flexible link driven by a printed armature motor at the hub. The measuring devices are shaft encoder, tachometer, accelerometer and strain gauges along the length of the arm. The shaft encoder, tachometer and accelerometer are essentially utilised in this work. A schematic diagram of the experimental flexible link manipulator is shown in Fig. 1. The flexible arm consists of an aluminium-type beam. The outputs of the sensors as well as a voltage proportional to the current applied to the motor are fed to a computer through a signal conditioning circuit and an anti-aliasing filter for analysis and calculation of the control signal.

3. PD-PI-TYPE FUZZY LOGIC CONTROLLER

The block diagram of the switching PD-PI-type controller is shown in Fig. 2. Only two states of the flexible-link manipulator, namely the hub angle error and torque, are available for controller design. From the hub angle error further two states, change in error and sum of error are derived. The hub angle error, change in error and sum of error are defined as

$$e(k) = \theta_d - \theta(k) \tag{1}$$

$$\Delta e(k) = e(k) - e(k-1) \tag{2}$$

$$\sum e(k) = \sum_{i=1}^{k} e(i) \tag{3}$$

where θ_d is the desired hub angle, Δe is the change in angle error and Σe is the sum of angle error produced from hub angle error. Triangular membership functions are chosen for the inputs and output. To construct a rule base, the hub angle error, change of angle error, sum of error and torque input are partitioned into 5 primary fuzzy sets labelled as {NB, NS, ZO, PS, PB}. PD-type and PI-type controllers are described by

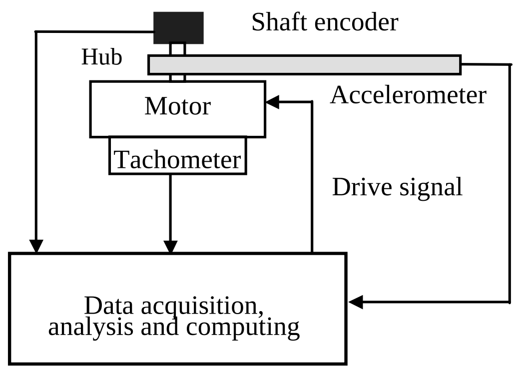

Fig. 1. Schematic diagram of the manipulator.

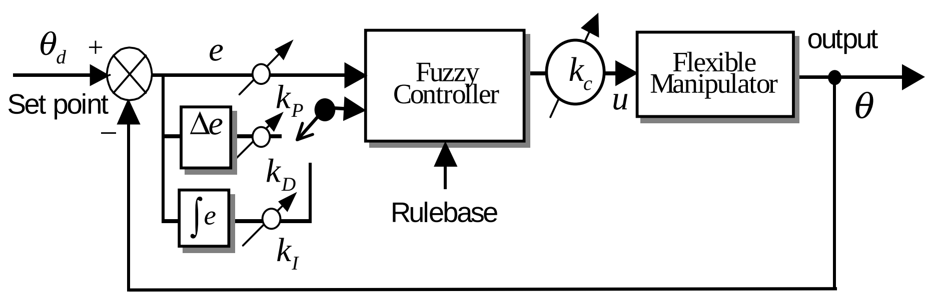

Fig. 2. Block diagram of neuro-fuzzy controller.

$$k_c \cdot u = k_p \cdot e + k_d \cdot \Delta e \quad (4)$$

$$k_c \cdot u = k_P \cdot e + k_I \cdot \sum e \quad (5)$$

where k_p, k_d, k_I and k_c are the proportional, differential, integral and controller gain coefficients and e, Δe and $\sum e$ are the error and change of error and sum of error respectively.

The PD- and PI-type FLCs, accordingly, constitute rules of the form

R_n : IF (e is E_i) and (Δe is C_j) THEN (u is U_k)
R_n : IF (e is E_I) and (Σe is S_j) THEN (u is U_k)

where R_n, n =1,2…25, is the nth fuzzy rule, E_i, C_j, S_j, and U_k, for i, j, k =1,2..5 are the primary fuzzy sets. The rule base is shown in Fig. 3.

Switching time is the time instant at which the controller is switched from PD- to PI-type. It is assumed that if controller is switched at the point of maximum overshoot, it will give better result. But experimental investigations show that a switching point after the maximum overshoot yields best performance. It is then chosen by trial and error method.

4. REDUCING NUMBER OF SCALING FACTORS

The parameters k_p, k_d, k_I and k_c are the proportional, differential, integral and controller gain coefficients of the PD-PI-type controllers. On-line adaptation of the four parameters will be time consuming. Therefore, a simplification is done in order to reduce the number of parameters, and this can be achieved with a single neuron network. Dividing both sides of equations (4) and (5) by k_p yields

Angle error	Change/sum of error				
	NB	NS	ZO	PS	PB
NB	PB	PB	PB	PS	ZO
NS	PB	PS	PS	ZO	NS
ZO	PS	ZO	ZO	ZO	NS
PS	PS	ZO	NS	NS	NB
PB	ZO	NS	NB	NB	NB

Fig. 3. Single rule-base for PD-PI-type FLC.

$$k_c' \cdot u = e + k_d' \cdot \Delta e \quad (6)$$

$$k_c' \cdot u = e + k_I' \cdot \sum e \quad (7)$$

where $k_d' = \dfrac{k_d}{k_p}$, $k_I' = \dfrac{k_I}{k_p}$ and $k_c' = \dfrac{k_c}{k_p}$.

The controller parameters are thus reduced to three and the parameters k_d' and k_I' are adjusted by the neural network and k_c' is chosen by a heurictic rule.

5. NEURAL NETWORK FOR RE-ADJUSTING THE MEMBERSHIP FUNCTIONS

Since the fuzzy sets for change of error and sum of error were redefined within the same universe of discourse for a single common rule-base, the membership functions of the fuzzy sets for both change of error and sum of error are readjusted at this stage. In many cases the same result can be obtained by tuning the scaling factors or adjusting the membership functions. Adjustment of membership functions requires learning of several parameters and hence scaling factor tuning is a much simpler task than adjustment of parameters (Chen and Linkens, 1998). The fuzzy sets, which were shifted from their original universe of discourse, are now readjusted by tuning the scaling factor k_d' and k_I' while scaling factor k_c' selected by heuristic rules. The self-learning task of multilayer perceptron could simply be replaced by a single neuron with a non-linear activation function (Yamada, T. and Yabuta, T., 1992). Thus, the architecture of the neural network becomes very simple as shown in Fig. 4 along with the structure of the neuro-fuzzy controller.

The sigmoid function used is defined as

$$f(x) = \frac{1 - e^{-ax}}{1 + e^{-ax}} \quad (8)$$

where x is the network output and a is the parameter that defines the shape of the sigmoid function.

6. GA FOR TRAINING THE NEURAL NETWORK

Interest in training neural networks (NNs) using genetic algorithms (GAs) has been growing rapidly in recent years (Caudell and Dolan, 1989; Montana and Davis, 1989; Whiteley *et al.*, 1990). One of the most popular training algorithms for feed-forward NNs is backpropagation (BP). Backpropagation is a gradient descent search algorithm, which is based on minimization of the total mean square error between actual output and a desired output. However, the BP algorithm suffers from a number of problems. It is very often trapped in local minima and is very inefficient in searching for global minimum of the search space. BP's speed and robustness are sensitive to several parameters of the algorithm and the best

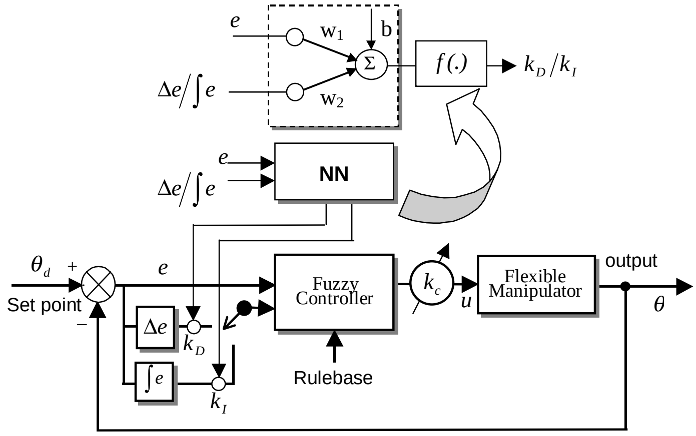

Fig. 4. Structure of neural network with PD-PI FLC

parameters to use appear to vary from problem to problem (Caudell and Dolan, 1989). On the other hand, there are several basic arguments suggesting that applying GAs to NN weight optimization is advantageous. GAs have the potential to produce a global minimum of the weight space and thereby avoid local minima. It is also an advantage to apply GAs to problems where gradient information is either not available or costly to obtain (Whiteley *et al.*, 1990).

The weights (w_1 ,w_2) bias b and parameter a of the network are updated by GA with the objective function

$$J = \sum_{k=1}^{N} |e(k)| \tag{9}$$

where $e(k)$ is angle error and N is some reasonable number of time units by which the system can be assumed to have settled quite close to a steady state. The evaluation of the population is a tedious process and that is why the population size is limited to 10 and the generation to 5.

Real valued chromosome representation is used for (w_1, w_2, b, a). Two parents are chosen randomly, a single point crossover and mutation is applied. The objective function is evaluated by applying control on the experimental manipulator. The worst two individuals are replaced by two new offsprings.

7. EXPERIMENTAL RESULTS

The developed control strategies have been tested on a single-link flexible manipulator. Due to tedious work of evaluation of the objective function, small population is tested up to 5th generation. Fig. 5 shows the system response with the PD-PI-type controller with optimised parameters for different switching points and k'_c. This switching type FLC has the advantage that it works well with a small rule base of 25 rules. In comparison to a PID-type controller the rule base is very small. Fig. 6 shows the average fitness i.e. absolute sum of hub angle error of best 3 individuals from generation 1 to 5 of GA optimisation and it shows that the GA converges well. Fig. 7 shows the system response of the best 4 individuals in generation 1 and Fig. 8 shows the system response of the best 3 individuals in generation 5. Tuning of scaling factors k'_d and k'_I is also playing an important role in adjusting the membership function for change of error and sum of error and yields an improved performance of the controller. Fuzzy, neural and genetic algorithms are meta-heuristics and applying these meta-heuristics to the flexible manipulators can lead to instability of the controller. Figure 9 shows the so-called linguistic trajectory, which represents the stability of the developed controller.

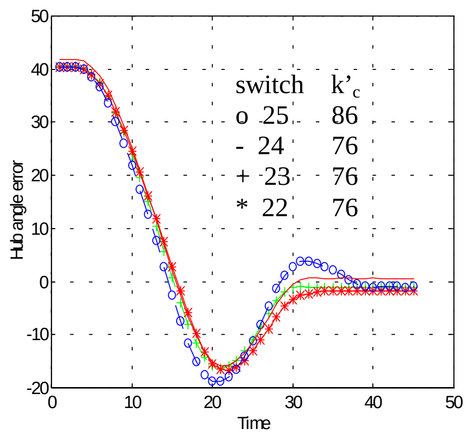

Fig. 5. Hub angle error for different switching points.

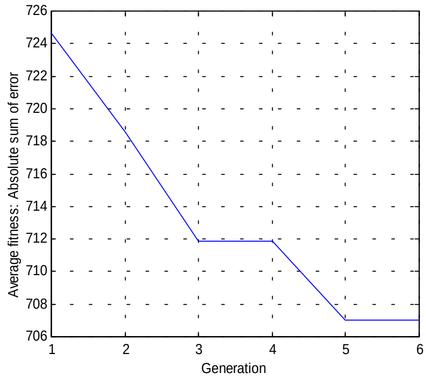

Fig. 6. Convergence of the fitness function.

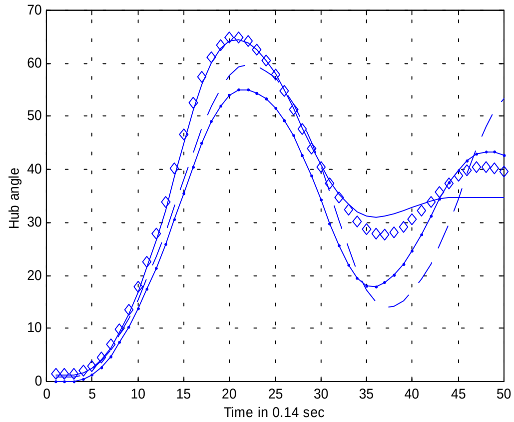

Fig. 7. Hub angle at generation 1- best 4 individuals.

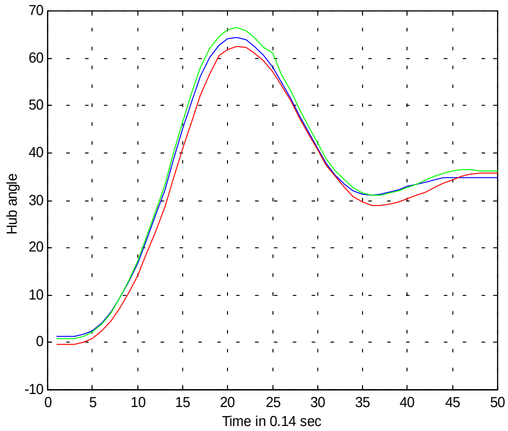

Fig. 8. Hub angle at generation 5- best 3 individuals.

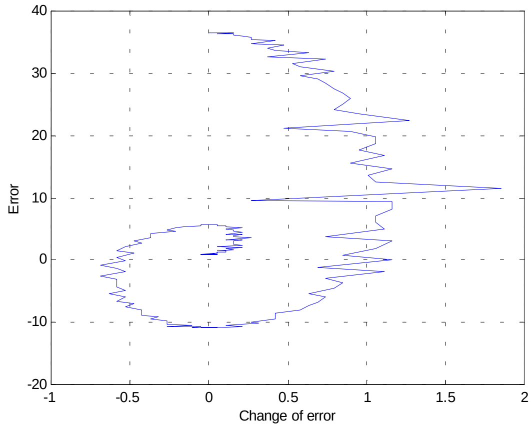

Fig. 9. Stability of GA-neuro-fuzzy controller.

8. CONCLUSION

Scaling factor tuning is a much simpler task than membership functions adjustment. On-line adaptation of the scaling factors is also an important task. After training the neural network by genetic algorithms, the scaling factors k'_d and k'_I are updated at each iteration during the control execution and, as demonstrated, this resulted an improved performance.

REFERENCES

Brehm, T. (1994). Hybrid fuzzy logic PID controller, *Proceedings of 3rd IEEE Conference on Fuzzy Systems* **3,** pp. 1682-1687.

Chen, M. and Linkens, D.A. (1998). A hybrid neuro-fuzzy controller, *Fuzzy Sets and Systems*, Vol. 99, pp. 27-36.

Caudell, Thomas P. and Dolan, Charles P. (1989). Parametric Connectivity: Training of Constrained Networks using Genetic Algorithms, *Proceedings of the 3rd International Conference on Genetic Algorithms (ICGA '89)*, George Mason University, June 4-7, pp. 370-374.

Chao, C.-T and Teng, C.-C. (1997). A PD-like Self-tuning Fuzzy Controller without Steady-state Error, *Fuzzy Sets and Systems*, **87**, pp. 141-154.

Chung, H.-Y, Chen, Bor-C and Lin, J.-J, (1998). A PI-type Fuzzy Controller with Self-tuning Scaling Factors, *Fuzzy Sets and Systems*, **93**, pp. 23-28.

Guerocak, H.B. and Lazaro, A. de S., (1994). A fine tuning method for fuzzy logic rule bases, *Fuzzy Sets and Systems*, **67**, pp. 147-161.

Maeda, M and Murakami, S., (1992). A self tuning fuzzy controller, *Fuzzy Sets and Systems*, **51**, pp. 29-40.

Montana, D. J. and Davis, L.(1989). Training Feedforward Neural Network using Genetic Algorithms, *Proceedings of 11th International*

Joint Conference on Artificial Intelligence, San Mateo, CA, Morgan Kaufmann, pp. 762-767.

Siddique, M.N.H. and Tokhi, M.O.(2000). PD-PI Fuzzy Logic Control of Flexible-link Manipulators, *Proceedings of 7th UK Workshop on Fuzzy Systems*, Sheffield, 26-27 October, pp. 36-41.

Tokhi, M.O. and Siddique, M.N.H. (2001). Soft Computing Control of Flexible-link Manipulators, *5th School on Active Noise and Vibration Control Methods*, Krynica, 22-25 May, Poland. (to appear)

Whiteley, D., Starkweather, T. and Bogart, C. (1990). Genetic Algorithms and neural Networks: Optimizing Connections and Connectivity, *Parallel Computing*, **14**, pp. 347-361.

Yamada, T. and Yabuta, T (1992). Neural Network Controller using Autotuning Method for Nonlinear Functins, *IEEE Trans. on Neural Networks;* **3**(4), pp. 595-601.

Zheng, L., (1992). A practical guide to tune of proportional and integral (PI) like fuzzy controller, *Proceedings of 1st IEEE International Conference on Fuzzy Systems*, San Diego, pp. 633-640.