Fifth IFSA World Congress (1993), 1129-1132

ON THE CONTROL OF SELECTED MACHINING PROCESSES BY MEANS OF A NEURAL FUZZY CONTROLLER

M. Balazinski[1], E. Czogala[2] and T. Sadowski[2]

[1] École Polytechnique de Montréal, Mechanical Engineering, C.P. 6079, Succ. «A», Montréal (Québec), Canada H3C 3A7; [2] Technical University of Silesia, Pstrowskiego 16, 44-101 Gliwice, Poland

Abstract – This paper presents the idea of a neural fuzzy controller with application to the control of an industrial machining process. The structure of such a controller, which links the idea of a fuzzy controller and a neural network, is suggested. Results of comparative simulations indicate that the proposed neural fuzzy controller performs equally well as a fuzzy logic controller; moreover, it is more flexible and allows faster data processing.

1. INTRODUCTION

Conventional and modern control theories need a precise knowledge of the model of the process to be controlled and exact measurements of input and output parameters. Due to the complexity and vagueness of many processes, the application of these theories is still limited. On the other hand, skilled human operators can control such processes quite successfully without any quantitative models in mind. The control strategy of a human operator is mainly based on linguistic qualitative knowledge concerning the behaviour of an ill-defined process.

A possible approach to modelling such a strategy is the concept of the fuzzy logic controller. This paper presents another control mechanism, based on the idea of a neural network [4]. While retaining the most valuable feature of fuzzy controllers, i.e. capability of controlling systems that cannot be precisely modelled, neural controllers have the advantages of high processing speed and learning by experience. It should be then possible to build a controller that 'learns' how to handle the process by just observing the human operator's actions.

2. NEURAL FUZZY CONTROLLER

2.1 The idea of a fuzzy logic controller

Fuzzy logic controllers can be viewed as models of a human operator determining the appropriate values of the control signal basing on observation of process variables (eg. *Error*, *Change in Error*, *Sum of Errors* etc.). The imprecise knowledge delivered by a human expert is usually expressed by a finite number (r=1, 2, ..., n) of the heuristic fuzzy rules written in the form:

$$R_r: \quad \text{if } A \text{ is } A_i^{(r)} \text{ and } B \text{ is } B_j^{(r)} \text{ and } \ldots \text{ and } C \text{ is } C_p^{(r)} \text{ then } U \text{ is } U_k^{(r)} \text{ and } V \text{ is } V_l^{(r)} \text{ and } \ldots \text{ and } W \text{ is } W_q^{(r)}. \tag{1}$$

where $A_i^{(r)}, B_j^{(r)}, \ldots, C_p^{(r)}$ denote linguistic values of the condition variables and $U_k^{(r)}, V_l^{(r)}, \ldots, W_q^{(r)}$ stand for linguistic values of the conclusion variables defined in the respective universes of discourse **X**, **Y**, **Z**, **U**, **V**, **W**. Such a rule corresponds to a relation which is usually represented by a fuzzy implication [6,8].

Approximate reasoning is performed by means of the compositional rule of inference [8], which in the case of a two-input, one-output controller may be written in the form

$$U' = B' \circ (A' \circ R). \qquad (2)$$

where R is the global relation obtained by connecting of all the rules.

The formal description of the fuzzy logic controller is known from literature [6,7] and will not be presented here. The final formula describing the fuzzy logic controller, when input data takes the form of fuzzy singletons (assuming that measurements of the process parameters are available), follow:

$$U'(u) = \sum_r \left[A_i^{(r)}(x_0)\cdot B_j^{(r)}(y_0)\cdot U_k^{(r)}(u)\right]. \qquad (3)$$

Defuzzification is performed by the center of gravity method.

It should be noted here that a different selection of operators may produce different control results.

2.2 Neural controllers

Neural controllers employ a neural network for information storage and processing, and input-output interfaces (EI, CEI and UI), which transform the information from the controller's environment into a form acceptable for the network and vice versa. Such a structure is presented in Fig. 1a.

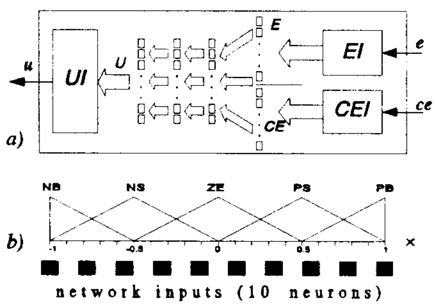

Fig. 1 Structure of the neural fuzzy controller (a) and input discretization (b)

The controller uses a multilayer perceptron [4] in the network part; such a network allows supervised learning and generalization and is relatively well-known.

To perform the control tasks successfully, a neural controller must be first trained. The training process is performed off-line and uses quantitative measurements expressed by triples (*Error, Change in Error, Control Action*) obtained during the observation of the process controlled by a human operator. Large volume of analog process data can be cut down by simple compression methods (training data are rounded to several representative values).

Trained network is used to control the process. Process data (error and change in error) are introduced via the input interface into the network which then recalls an appropriate action. The response of the network is translated to actual control value by the output interface.

2.3 The idea of the neural fuzzy controller

Let Card(X), Card(Y), Card(U) denote the respective cardinal numbers of the discretized universes of discourse mentioned in Section 1. The number of input neurons can be then determined as [4]

$$m = \mathrm{Card}(X) + \mathrm{Card}(Y), \qquad (4)$$

while the number of output neurons

$$m = \mathrm{Card}(U). \qquad (5)$$

The idea of such a discretization is illustrated in Fig. 1b, where squares represent input neurons of the network. The whole network structure can be annotated as (m_1-h_1-h_2-...-n).

The input vector for the network consists of binary groups corresponding to subsequent process variables. Each group has a "1" at position k, where

$$k = \mathrm{ROUND}((N-1)x)+1, \qquad (6)$$

and "0"s at remaining positions. N is the number of neurons in the group ($1 \leq k \leq N$), and x is the value of process variable. Since discretization enforces rounding input and output data to several values (points) corresponding to input or output neurons, no intermediate values are allowed. Discretization performed in the control phase causes loss of information that may lead to oscillations or unacceptably large static error in steady states. To avoid this, a kind of interpolation can be introduced for input and output values during the control phase. Now, two elements of the input vector take the following nonzero values:

$$\begin{cases} i_k = 1 - \mathrm{FRAC}((N-1)x) \\ i_{k+1} = \mathrm{FRAC}((N-1)x) \end{cases} \qquad (7)$$

where k determines the position of the first input element:

$$k = \mathrm{TRUNC}(N-1)x+1. \tag{8}$$

In the last formula, N is the number of input neurons in the group (numbered from 1 to N) and x is the input value normalized to the interval [0, 1). The remaining input neurons take the input value of 0. It should be noted, however, that the training phase is performed without interpolation. This allows us to reduce the volume of training information.

To calculate the output (control) value, a center-of-gravity-like formula can be applied:

$$o = \frac{1}{M-1}\left(\frac{\sum_{k=1}^{M} k \cdot o_k}{\sum_{k=1}^{M} o_k} - 1\right), \tag{9}$$

where M is the number of output neurons.

The above described interpolating neural fuzzy controller is more accurate than the 'classical' one described in [1,2]. By "fuzzifying" input information it allows to implement the idea of fuzzy sets. However, it relies strongly on the interpolative properties of the network.

3. CONTROLLED SYSTEM

We have tested the neural fuzzy controller on a static turning process, in which a constant cutting force should be assumed to assure the proper wear of the cutting tool. The changeable depth of cutting is compensated by the change of the feed rate. Basing on [5], the relation between the cutting depth, feed rate and the cutting force in the y-direction can be approximated by the following formula:

$$F_y = C_y \cdot d^{e_y} \cdot f^{u_y}, \tag{10}$$

where d denotes the cutting depth, f stands for the respective feed rate and C_y, e_y, u_y are constant coefficients.

Under chosen cutting conditions [5], formula (10) takes the form

$$F_y = 876 d^{0.9} f^{0.75}. \tag{11}$$

Assuming a constant force F_{yo} = 3050.4 [N], the range of the cutting depth $d \in$ [3..5] [mm] corresponds to the feed rate $f \in$ [0.75..1.4] [mm/s]. In this case the cutting depth was the value controlled, the feed rate being the control value.

4. NUMERICAL RESULTS

In the first stage of the experiment, control was performed by a simple fuzzy logic controller using a rule base consisting of 9 rules. Next, the same control program was performed by the neural fuzzy controller trained on the basis of fuzzy logic control results. The results are shown in Figures 2 and 3.

The neural fuzzy controller employed a 20-20-10 backpropagation network with sigmoidal elements. The learning rate used was 0.6, the momentum term 0.3, number of training cycles: 200 (random pattern presentation), randomization interval for connectivity matrices: [-0.5..0.5]. Values of error, change in error and drive were rescaled on the basis of experimental data.

For the purpose of comparative study an integral quality index (QI) was defined as below:

$$QI = \sum_{i=0}^{N} \frac{(CD_i - SP_i)^2}{N+1} \tag{12}$$

where CD_i denotes the controlled cutting depth, SP_i is the current set point and N is the total number of observation points. The other quality measures were:

- rise time, ie. time needed to change the cutting depth from 3 mm to within 5% limit of set point of 5 mm, ie. 4.75 mm),
- maximum static error, ie. difference between cutting depth and set point in steady state, relative to set point (maximum value through whole simulation time).

Figures 2, 3 and Table 1 contain the summary of simulation results.

Table 1 Comparison of controllers

Controller	QI	Rise time [sec]	Static error [%]
Fuzzy logic	12.39	0.15	-0.73
Fuzzy neural	11.47	0.15	-2.03

The quality index for the neural fuzzy controller is slightly better than that of the fuzzy logic controller, which results from its faster response for intermediate set point changes. The rise times (in the case of maximum set point change) are identical. The neural fuzzy controller leaves a nonzero static error, which, however, does not exceed acceptable limits.

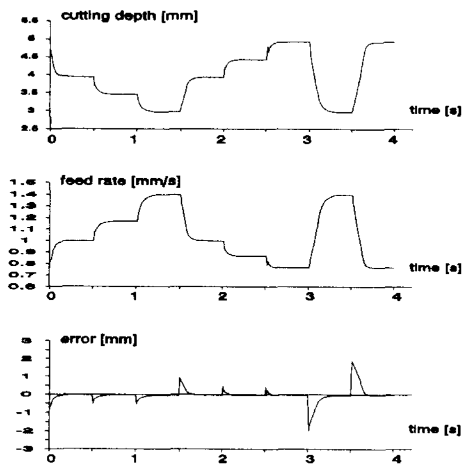

Fig. 2 Simulation results – fuzzy logic controller

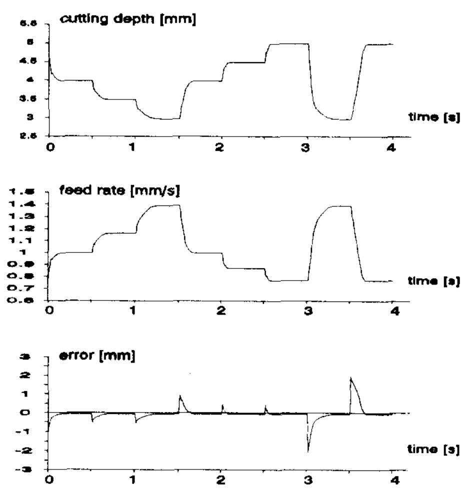

Fig. 3 Simulation results – neural fuzzy controller

5. CONCLUSIONS

The results of control performed by the neural fuzzy controller are comparable with those of fuzzy logic control (used as a 'teacher'). The accuracy and speed of control are satisfactory. Moreover, the neural fuzzy controller has the capability of learning by experience (adaptivity) and can take advantage from fast, parallel data processing.

The most important objectives for future should cover the problem of input and output discretization, and the number of hidden neurons. The question of controller response for data expressed as fuzzy sets should be also examined deeper, since research in this field can lead to application of the presented idea in fuzzy expert systems [3].

REFERENCES

1. Balazinski M., Czogala E., Sadowski T., Control of metal-cutting process using neural fuzzy controller, Proc. of the Second IEEE International Conference on Fuzzy Systems FUZZ-IEEE 1993, March 28-April 1, 1993, San Francisco, CA, USA.

2. Balazinski M., Czogala E., Sadowski T., Modelling of neural controllers with application to the control of a machining process, Fuzzy Sets and Systems 56 (1993), 8.

3. Buckley J.J., Hayashi Y., Czogala E., On the equivalence of neural nets and fuzzy expert systems, Fuzzy Sets and Systems 53, 1993, 129-134.

4. Czogala E., Sadowski T., A Method of Conversion of Fuzzy Decision Tables into Neural Networks, Biocybernetics and Biomedical Engineering, Inst. of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, Vol. 12, No. 1-4, 1992, 77-85.

5. Kaczmarek J., Principles of Machining by Cutting, Abrasion and Erosion, Peter Peregrinus Ltd., 1976.

6. Lee C.C., Fuzzy logic in control systems: fuzzy logic controller - Part I/Part II, IEEE Trans. on Systems, Man and Cybernetics, Vol. 20, No. 2, March/April 1990, 404-435.

7. Mamdani E.H., Applications of fuzzy algorithms for simple dynamic plant, Proc. IEE, Vol. 121, No. 12, 1974, 1585-1588.

8. Zadeh L.A., Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. on Systems, Man and Cybernetics, Vol. 3, 1973, 28-44.