ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1993

Seria: MECHANIKA z. 113 Nr kol. 1198

Marek BALAZIŃSKI

Mechanical Engineering
École Polytechnique de Montréal, Canada

Ernest CZOGAŁA, T. SADOWSKI
Technical University of Silesia, Gliwice, Poland

CONTROL OF A MACHINING PROCESS BY MEANS OF NEURAL CONTROLLERS

Summary. This paper recalls the idea of a neural controller with application to the control of an industrial machining process. Structures of such controllers are suggested, and the results of simulations comparing their performance to that of a conventional fuzzy logic controller are shown. The experiments indicate that the performance of the proposed neural controllers is satisfactory.

STEROWANIE PROCESEM OBRÓBKI SKRAWANIEM ZA POMOCĄ REGULATORÓW NEURONOWYCH

Streszczenie. W pracy omówiono ideę regulatora neuronowego i jego zastosowanie do sterowania procesem obróbki skrawaniem. Podano dwie struktury regulatorów neuronowych oraz wyniki symulacji wraz z porównaniem omawianych regulatorów z konwencjonalnym regulatorem rozmytym. Uzyskane wyniki badań pozwalają stwierdzić, że jakość sterowania przy zastosowaniu obu regulatorów neuronowych jest zadowalająca.

УПРАВЛЕНИЕ ПРОЦЕССАМИ РЕЗАНЯ ПРИ ПОМОЩИ НЕВПОРЕГУЛЯТОРОВ

Резюме. В работе оговорена идея невронного регулятора и его применение для процессов обработки резанием. Даны две структуры невронных регуляторов а также результаты имитации ыправления со сравнением оговориваемых регуляторов с ковенциональным разплывчатым регулятором. Полученные результаты исследований показывают, что качество управления с применением обеиь цтруктур невронных регуляторов удовлетворительно.

1. INTRODUCTORY REMARKS

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. However, due to the complexity and vagueness of practical processes, application of these theories is still limited.

In many real processes, control relies heavily upon human experience. Skilled human operators can control such processes quite successfully without any quantitative models in mind. The control strategy of the human operator is mainly based on linguistic qualitative knowledge concerning the behavior of an ill-defined process. Taking into account the fact that most of the machining processes are stochastic, nonlinear and ill-defined, the metal-cutting processes fall into such a category of complex processes which are attractive to be controlled by means of fuzzy logic [5].

Some approaches to the concept of a neural-network-based controllers employed to the control of various ill-defined, complex processes have been reported recently. The aim of this paper is to recall this useful concept by pointing out its potential application to the control of machining processes, such as turning, milling, grinding etc.

2. STRUCTURES OF CONTROLLERS EMPLOYING NEURAL NETS

2.1. The idea of a fuzzy logic controller

The imprecise knowledge delivered by a human operator is usually expressed by a collection of fuzzy control rules having the form

$$\begin{aligned} \text{Rr:}\quad & \text{If Error} = A_i^{(r)} \text{ and Change in Error} = B_j^{(r)} \\ & \text{then Control Action} = U_k^{(r)} \end{aligned} \tag{1}$$

where r stands for the rule index. $A_i^{(r)}$, $B_j^{(r)}$, $U_k^{(r)}$ are linguistic values (fuzzy sets) for the linguistic variables *Error*, *Change in Error* and *Control Action* defined in universes of discourse **X**, **Y**, **U**, respectively.

Such a collection of rules makes up the so-called rule base. We should mention here the explicit connective 'and' between the variables *Error* and *Change in Error* and the implicit rule connective 'also' which links all the rules in the rule base.

A fuzzy control rule is usually implemented by a fuzzy implication (a fuzzy relation in $\mathbf{X} \times \mathbf{Y} \times \mathbf{U}$):

$$R = (A_i^{(r)} \text{ and } B_j^{(r)}) \qquad U_k^{(r)} \tag{2}$$

where $(A_i^{(r)}$ and $B_j^{(r)})$ may be interpreted as a fuzzy set $A_i^{(r)} \times B_j^{(r)}$ in $\mathbf{X} \times \mathbf{Y}$.

Given input information: A' (error) and B' (change in error), the control action U' can be deduced employing the compositional rule of inference, the definitions of fuzzy implication and connectives 'and' and 'also'. Even if we choose a particular compositional rule of inference, fuzzy implication and both connectives 'and' and 'also', the inference process can still be realized in different ways. Namely, if we consider input information (error and change in error) as vectors, we shall write the compositional rule of inference in the form:

$$U' = B' \circ (A' \circ R) \tag{3}$$

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

We can also use another notation and apply the following formula:

$$U' = (B' \times A') \circ R \tag{4}$$

Taking into account, for example, the sup-min (sup-prod) as composition operators, min (prod) for implication, min (prod) for 'and' and max (sum) for 'also' connectives, we get the same inference result from both formulas (3) and (4) respectively.

Using the membership function representation, we can write

$$U'(u) = \max_r \sup_{x \in X, y \in Y} \min\left[\min(B'(y), A'(x)), \min(A_i^{(r)}(x), B_j^{(r)}(y), U_k^{(r)}(u))\right] \tag{5}$$

or, for practical use

$$U'(u) = \sum_r \sup_{x \in X, y \in Y} \left[(B'(y) \cdot A'(x)) \cdot (A_i^{(r)}(x) \cdot B_j^{(r)}(y) \cdot U_k^{(r)}(u))\right] \tag{6}$$

Taking singletons (Kronecker delta) for A'(x), B'(y), when measurements are available, formulas (5) and (6) can be simplified:

$$U'(u) = \max_r \min\left[A_i^{(r)}(x_0), B_j^{(r)}(y_0), U_k^{(r)}(u)\right] \tag{7}$$

or

$$U'(u) = \sum_r \left[A_i^{(r)}(x_0) \cdot B_j^{(r)}(y_0) \cdot U_k^{(r)}(u)\right] \tag{8}$$

As a defuzzification method center of gravity can be used.

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

Two possible structures of a neural controller will be described below.

2.2. Analog (2-h-1) neural controller

In the simplest case, a neural controller may use a two-input and one-output network. Analog values of error and change in error are introduced into the network, which responds with an analog control value. Due to the sigmoidal shape of transfer function, the output value lies within the interval (0, 1); the only transformation required for input and output signals is rescaling.

Such a controller is shown in Fig. 1.

To ensure convergence of the training procedure we must either limit the volume of training data or increase the number of connections in the network. Experimental results show that it is sufficient to quantize the intervals of input and output parameters into ten sub-intervals (i.e. eleven representative values: 0.0, 0.1,..., 1.0). For intermediate values obtained from the process during control, we rely on the interpolation performed by the network. Since multilayer perceptrons exhibit good interpolative properties, such a controller is very efficient in cases when unknown (not learned previously) data is presented at the inputs. It can then interpolate between known patterns to evaluate correct control value. For instance, if the controller had learned input-output triples: (0.1, 0.1, 0.9) and (0.3, 0.3, 0.7), it should respond to the unknown data set (0.2, 0.2) with a control value of approximately 0.8.

Such a controller represents an heuristic approach to the control problem (learning by experience and approximate reasoning); however, the idea of fuzzy sets is not employed in this case.

2.3. Discrete (m-h-n) neural controller (neural fuzzy controller)

Taking into account the input information, two versions of the neural fuzzy controller may be considered [3].

Let Card(X), Card(Y), ..., Card(Z), Card(U), Card(V), ..., Card(W) denote respective cardinal numbers of the aforesaid discretized universes of discourse. The number of input neurons for the 'vector' version of a neural network can be determined as [4]

$$m = \mathrm{Card}(X) + \mathrm{Card}(Y) + \ldots + \mathrm{Card}(Z) \tag{9}$$

while the number of output neurons

$$n = Card(U) + Card(V) + ... + Card(W) \tag{10}$$

For the 'matrix' version of a neural network the number of input neurons can be determined as

$$m = Card(X) * Card(Y) * ... * Card(Z) \tag{11}$$

and the number of output neurons as

$$n = Card(U) * Card(V) * ... * Card(W) \tag{12}$$

The idea of such a discretization was presented in [4].
It is easy to show that the approaches to the inference process in a fuzzy controller mentioned above lead to the previously described construction of two versions of the neural fuzzy controller which may be trained using the same input information.

Considering the discretization of universes of discourse for error - **X**, change in error - **Y**, and control action - **U**, we can now construct two versions of the neural fuzzy controller. The structure of multilayer perceptron seems to be sufficient for the discussed task [1,2,4]. The structure of the input layer is considered to be linear for the 'vector' version and rectangular for the 'matrix' version (see Fig. 2).

According to formula (9) and taking into account the discretization of universes of discourse, the vector version will have m1 = Card(**X**) + Card(**Y**) input neurons. The number of output neurons is given by n = Card(**U**). Denoting the number of units in the hidden layers as h1, h2,... , we can annotate the structure of the 'vector' network as (m1 - h1 - h2 - ... - n).

For the 'matrix' version according to formula (11) and using the same discretization as in the vector version, we will have m2 = Card(**X**) * Card(Y) input neurons. Assuming the same number of output neurons i.e. n = Card(U) and denoting the number of units in the hidden layers as h1, h2, ... , we can also annotate the structure of the network matrix version as (m2 - h1 - h2 - ... - n)

For instance, let us consider a 'vector' neural fuzzy controller using 10 input neurons for error and change in error, and 10 output neurons for control value. Assuming that all the values lie within intervals (0, 1), for the triple (0.1, 0.4, 0.6) we may write the following input and output vectors:

input: 0100000000 0001000000
output: 0000010000

Note that discretization enforces rounding input and output data to several values, corresponding to input or output neurons. Such a network does not exhibit interpolative properties; however, its advantage is that it may accept 'sampled' fuzzy information, not only crisp (singleton) data.

2.4 Training and process control

The above presented neural controllers may be trained off-line by means of quantitative measurements expressed by triples (*Error, Change in Error, Control Action*) obtained during the observation of the process (sampling its parameters). Depending on the structure of the controller, input-output data requires preprocessing (rescaling and/or discretization). It should be mentioned here that the neural nets may be also initially trained using information obtained from the control rules (qualitative knowledge) [4]. As a learning scheme, the widely used backpropagation algorithm can be applied.

After the training, the network can be used to control the process. This is accomplished by feeding process data (error and change in error) to the input layer of the network, which then recalls an appropriate action. Note that the analog neural controller operates as an interpolative network, while the neural fuzzy controller - as an associative memory.

3. NUMERICAL RESULTS

We will present here some numerical results obtained by simulating the control of a machining process.

In order to obtain comparable results we have used a slightly modified knowledge base originating from Zhu et al., described in [1]. The fuzzy controller used in our experiments employed sup-prod for the compositional operation, prod for the 'and' connective between rule premises, sum for the sentence connective 'also'.

As an example let us mention a turning process in which a constant cutting force (static case) should be assumed to assure the proper wear of the cutting tool. Changeable depth of cutting is compensated by the change of the feed rate. The relation between the cutting depth, feed rate and the cutting force in the y-direction can be approximated by the following formula [1]:

$$F_y = C_y \cdot d^{e_y} \cdot f^{u_y} \tag{13}$$

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 [1], formula (13) takes the form

$$F_y = 876 d^{0.9} f^{0.75} \tag{14}$$

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

In the first stage of our experiment we used a fuzzy controller represented by equations (6) and (8) to simulate a human operator. The set point was preprogrammed to change within the interval [3..5] (see Fig. 3). The results of control, shown in Fig. 3a, were then used to train the neural controllers. For modelling the controllers we employed three-layer feedforward networks with sigmoidal elements. In the case of an analog neural controller the structure used was 2-10-1, while for the neural fuzzy controller we applied a 20-20-10 structure. A backpropagation algorithm was used for training, with the learning rate of 0.6 and momentum of 0.3. Random pattern presentation scheme was used for training; 200 training rounds were performed. The connectivity matrices were initially randomized with values from the interval [-0.5..0.5]. The ranges of error, change in error and drive were rescaled to 'fill' the whole range covered by the input and output neurons; rescaling was performed on the basis of operator's experience. The error and change in error values were clamped to the interval (-0.5, 0.5), while drive values lied within the interval of (0.75, 1.4).

In the second stage the same control program was performed making use of the previously trained neural controllers. The results are shown in Figs. 3b and 3c.

For the purpose of comparative study a quality index was defined as below:

$$QI = \sum_{i=0}^{N} \frac{(z_i - SP_i)^2}{N+1} \tag{15}$$

where z_i denotes the controlled value (cutting depth), SP is the set point and N is the total number of observation points.

Comparing both stages we can note that both the fuzzy controller and neural controllers behave similarly. In the case of the analog neural controller we obtained a slightly better quality index (10.974 versus 11.797 of the fuzzy logic controller), resulting from its quicker response; in the case of neural fuzzy controller the quality index (12.546) was minimally worse, since oscillations were observed in steady states. It should be noted that the speed of a neural controller is greater than that of a classical fuzzy controller, even though the parallel structure of the neural network is simulated.

4. CONCLUDING REMARKS

The results of numerical experiments show that the both neural controllers perform equally well as a conventional fuzzy logic controller. They are, moreover, much more flexible (adaptive) and faster than the latter. The accuracy of control is sufficient, as it results from the performed experiments.

As the objective for future research, the input and output discretization problem should be considered: the larger the number of input (output) neurons, the better the accuracy of the controller; however, the larger the network itself, the longer the training time. Also the number of hidden neurons and learning parameters should be examined deeper.

It should be also noted that the structure of the neural fuzzy controller should allow introduction of data expressed as fuzzy sets (which are 'sampled' at its inputs and outputs), while the "2-h-1" analog neural controller allows only crisp values at its inputs and output).

REFERENCES

[1] Balazinski M., Czogała E., Sadowski T., Neural controllers and their application to the control of a machining process (accepted in Archiwum Informatyki Teoretycznej i Stosowanej PAN).

[2] Brunsmann U., Czogała E., von Koch H., On modelling of a fuzzy controller by means of two versions of multilayer feedforward neural networks. BUSEFAL 50, March 1992.

[3] Buckley J.J., Czogała E., Fuzzy models, fuzzy controllers, and neural nets (accepted in Archiwum Informatyki Teoretycznej i Stosowanej PAN).

[4] Czogała E., Sadowski T., A Method of Conversion of Fuzzy Decision Tables into Neural Networks. Biocybernetics and Biomedical Engineering, Inst. of Biocybernetics and Biomedical Engineering, PAN, Warsaw (to appear).

[5] Ralston P. A. S., Ward T. L., Lambert L. M., Fuzzy logic control of machining. Manufacturing Review vol. 3, no. 3, 1990, 147-154.

Recenzent: Dr hab. Adam Mrózek

Wpłynęło do Redakcji dnia 20. 11. 1992

Streszczenie

W pracy omówiono ideę regulatorów rozmytych (fuzzy logic controllers), w których wykorzystuje się wiedzę eksperta (operatora) o złożonym (słabo zdefiniowanym) procesie oraz ideę regulatorów neuronowych (neurall controllers), zastosowanych do sterowania procesem obróbki skrawaniem (toczenie). Artykuł składa się z dwu części. Pierwsza z nich, poza opisem koncepcji regulatora rozmytego, zawiera opis dwu struktur regulatorów neuronowych, tj. analogowego regulatora neuronowego i tzw. neuronowego regulatora rozmytego. Krótko omówiono tryby uczenia i uczenia (sterowania) regulatorów neuronowych. W drugiej części artykułu zamieszczono wyniki symulacji cyfrowej sterowania procesem toczenia za pomocą wyżej wymienionych regulatorów. Dokonano porównania działania wszystkich regulatorów za pomocą odpowiedniego wskaźnika jakości. Uzyskane wyniki badań pozwalają stwierdzić, że jakość sterowania przy zastosowaniu obu regulatorów neuronowych jest zadowalająca.

FIGURES

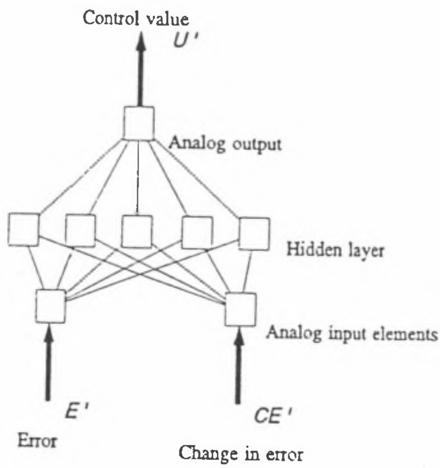


Fig. 1. Structure of an analog neural controller
Rys. 1. Struktura analogowego regulatora neuronowego

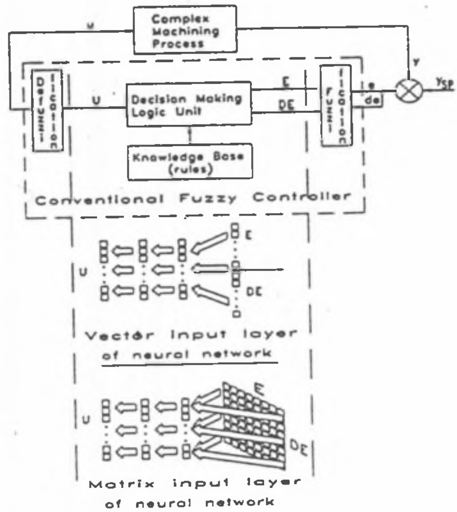
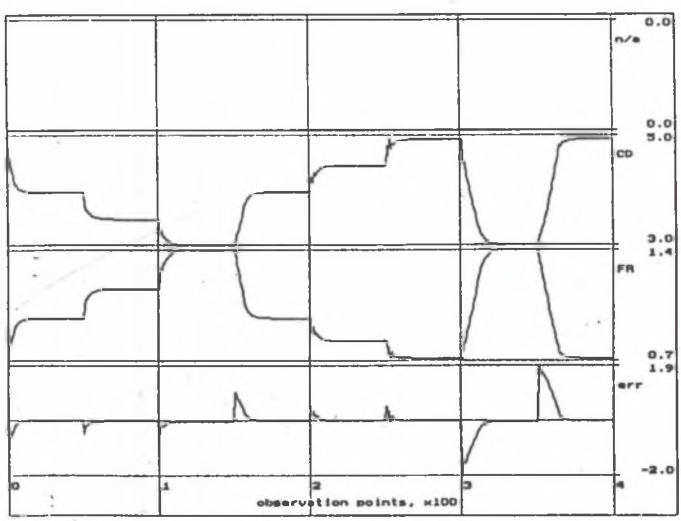


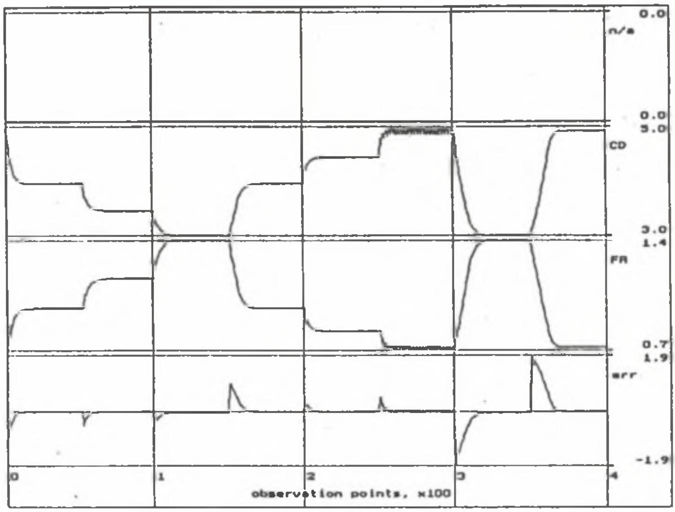
Fig.2 Control loop employing a fuzzy logic controller and two possible versions of a neural fuzzy controller

Rys.2 Układ regulacji wykorzystujący regulator rozmyty i dwie proponowane struktury neuronowego regulatora rozmytego.

3a



3b



3c

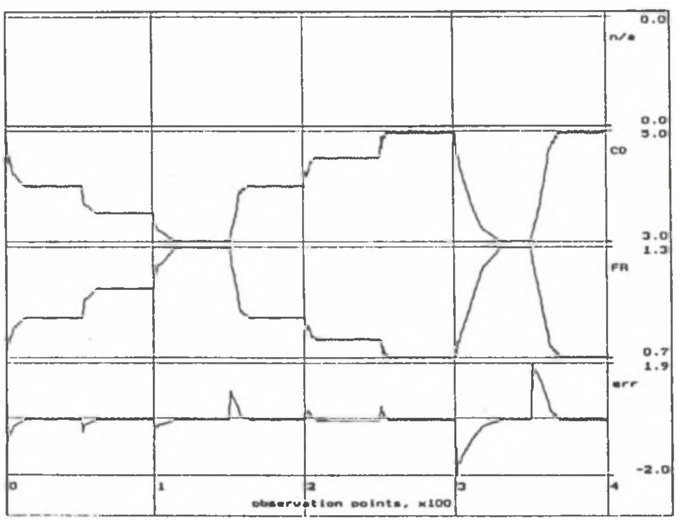


Fig. 3. Results of control using:
a) fuzzy logic controller
b) analog neural controller
c) fuzzy neural controller

Rys. 3. Wyniki sterowania z wykorzystaniem
a) regulatora rozmytego
b) analogowego regulatora neuronowego
c) rozmytego regulatora neuronowego