A Transductive Neuro-Fuzzy Force Control: An Ethernet-Based Application to a Drilling Process

Agustin Gajate[1], Rodolfo Haber[1], and Pastora Vega[2]

[1] Instituto de Automatica Industrial, Spanish Council for Scientific Research
Carretera de Campo Real km. 0,200, 28500 Arganda del Rey, Spain
[2] Departamento de Informatica y Automatica, Universidad de Salamanca
Pza. de los Caidos s/n, 37008 Salamanca, Spain

Abstract. This paper presents the application of a neural fuzzy inference method to the field of control systems using the internal model control paradigm (IMC). Through a transductive reasoning system, a neuro-fuzzy inference system enables local models to be created for each input/output set in the system at issue. These local models are created for modeling the direct and inverse dynamics of the process. The models are then applied according to IMC paradigm. In order to demonstrate the benefits of this technique for control systems, it is applied for networked cutting force control in a high-performance drilling process. A comparative study between a well-established neuro-fuzzy technique and the suggested method is performed.

Keywords: Transductive modeling, Neuro-fuzzy inference, Internal model control, Networked control, High-performance drilling.

1 Introduction

Recent years have been characterized by the development of new paradigms in the field of Artificial Intelligence (AI). One of the research disciplines that has spawned the most developments is soft computing, where computing, reasoning and decision-making make the most out of tolerance for imprecision, uncertainty and approximate reasoning, in order to yield better solutions [1]. The hybridization of fuzzy logic with neural networks is the most well-established and best-known method within soft computing.

By the late nineties, several hybrid neuro-fuzzy systems had already been developed, which may be separated into two major groups: neural networks endowed with the ability to handle fuzzy information [fuzzy-neural networks (FNN)] [2], [3], and fuzzy systems combined with neural networks in order to enhance certain desirable characteristics [neural-fuzzy systems (NFS)] [4], [5]. A combination of evolutionary computation with a neuro-fuzzy system is proposed too in several works.

Most of the above-mentioned neuro-fuzzy strategies apply inductive reasoning systems. In inductive reasoning the key issues is to find a general model drawn from the entire set of input/output data representing the whole system.

C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 573–582, 2009.

In contrast there are transductive reasoning methods which generate a model at a single point of the workspace. For each new datum that has to be processed, the closest examples are selected among the known data, with the goal of creating a new local model that dynamically approximates the process in its new state as close as possible. The main issue is therefore how to assign more weight to the specific information related with the datum to be processed than to the general information provided by the entire training set [6].

Transductive methods have some advantages over inductive methods, because sometimes creating a valid model for the entire space or region of operation is a difficult task, yielding insufficient performance in some cases. In addition, these strategies are capable of functioning correctly with a small training set. Transductive reasoning methods have been applied to text recognition applications, time series prediction and medical diagnosis applications. The use of local models in control tasks first appeared in 1990 for controlling some movements in robotics [7], [8]. However, on the basis of reviewed literature, applications in the field of control manufacturing processes as well as to improve machining processes have not been previously reported.

On the other hand, the design of intelligent controllers necessarily requires control schemes and methods that guarantee the desired features. The internal model control (IMC) paradigm has been applied successfully in intelligent control systems. It is considered a consolidated method for designing fuzzy, neural and neural-fuzzy controllers for process control and in addition a method that guarantees a good dynamic response and reliable behavior in the presence of disturbances [9]. Thus, the characteristics discussed above make the internal model control paradigm and NFI suitable strategies [10], whose synergy can handle processes that display nonlinear and time-variant behavior and where classic techniques have not yielded the desired results.

Several works reported in the literature address the design and implementation of intelligent control systems for machining processes with the objective of optimizing these processes. Of all the machining operations, the drilling process is precisely one of the processes that has received less attention with a view to process improvement through the application of control techniques. This paper is focused on increasing the material removal rate as well as enhancing useful tool life through networked control of the cutting force by means of real-time modification of the tool's own feed rate.

2 Transductive Neuro-Fuzzy System

The transductive neuro-fuzzy inference system (NFI) involves the creation of unique local models for each subspace of the problem, using the Euclidean distance [11]. The system's inputs can be treated in different kinds of physics units but the normalization is recommended. In this paper each input data is normalized subtracting the mean and dividing by the standard deviation of the set of known data or training set. After the normalization, the personalized local model is then created using data from the training set that are the closest to each new input datum. The Euclidean distance is used for selecting this data subset. The size of this subset (N_q) is an input parameter for the algorithm.

NFI uses a Mamdani-type inference method and fuzzy membership functions are typically Gaussian-type. This type of membership function that is derivable enables the use of supervised learning algorithms such as back-propagation algorithms. The evolving clustering method (ECM) is used to create membership functions and fuzzy rules [11]. It consists in a single-iteration algorithm for the dynamic on-line clustering of a data set. For all following data, on the basis of Euclidean distances and the clustering threshold value (D_{thr}), the algorithm either adds each datum to an existing set (updating the center and the radius of the set) or creates a new set. For that purpose, the center of the set is taken as the center of the Gaussian function, and the radius is taken as the width.

Considering P inputs, one output and M fuzzy rules initially defined by the clustering algorithm, the lth rule has the form:

R_l: If x_l is ϕ_{l1} and x_2 is ϕ_{l2} and... x_P is ϕ_{lP}, then y is γ_l. (Cluster l)

$$\phi_{lj} = \alpha_{lj} \exp\left[-\frac{(x_{ij} - m_{lj})^2}{2a_{lj}^2}\right] \tag{1}$$

$$\gamma_l = \exp\left[-\frac{(y - n_l)^2}{2{\delta_l}^2}\right] \tag{2}$$

where m and n are the centers of the Gaussian functions for the inputs and outputs, a and δ are the widths, $i = 1,2,\ldots,N_q$ is the index representing the number of closest neighbors, $j = 1,2,\ldots,P$ represents the number of input variables, and $l = 1,2,\ldots,M$ represents the number of fuzzy rules.

The centers m and n and the widths a and δ are obtained as the result of the ECM algorithm, while the parameter α_{lj} is chosen by design (α_{lj}=1) and represents the weight of each of the input membership functions. These parameters are adjusted with the back-propagation algorithm.

Using the center of area defuzzification method, the output of the NFI for an input vector $\overline{x}_i = [x_1, x_2, \ldots, x_p]$ is calculated as follows:

$$O(\overline{x}_i) = \frac{\sum_{l=1}^{M} \frac{n_l}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp\left[-\frac{(x_{ij}-m_{lj})^2}{2a_{lj}^2}\right]}{\sum_{l=1}^{M} \frac{1}{\delta_l^2} \prod_{j=1}^{P} \alpha_{lj} \exp\left[-\frac{(x_{ij}-m_{lj})^2}{2a_{lj}^2}\right]} \tag{3}$$

The system uses input/output data of the closest training data $[\overline{x}_i, t_i]$ and the goal is to minimize the following target function:

$$E = \frac{1}{2} v_i \left[f(\overline{x}_i) - t_i\right]^2 \tag{4}$$

where v_i=1-$(d_i - \min(\overline{(}d)))$ with $i = 1,2,\ldots,N_q$ and $\overline{(}d)$=$\left[d_1, d_2, \ldots, d_{N_q}\right]$ as the vector of distances calculated in the first step. Fig.1 shows the block diagram of the implemented algorithm.

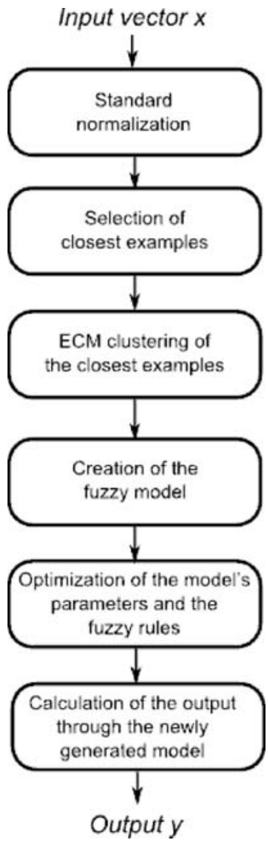

Fig. 1. Block diagram of the NFI algorithm [11]

3 The High-Performance Drilling Process

Drilling is one of the most intensely used processes in the manufacture of aeronautic and automobile components in addition to the manufacture of moulds and dies. One of the main targets of manufacturing companies is to reduce production times and increase the quality of their products. Lower cycle times in a drilling process are directly related to increasing the material removal rate without damaging the cutting tool and without affecting the finish quality. In the high-performance drilling process the work is performed at very high cutting speeds at or near the limitations of the material, tool and machine tool.

Drilling force is the most important variable in the drilling process. Higher feed rates increase the material removal rate along with an increase in drilling force. However, higher cutting forces and drilling torques yield negative effects as well, such as rapid drill wear, tool vibration and the risk of catastrophic tool failure. Thus, it is important not only to maintain a constant drilling force, but also to obtain good closed-loop dynamic behavior without oscillations and overshoots, thus increasing useful tool life.

Traditional control methods still have much room for improvement and thus far have not yielded the expected improvement of the drilling process on the basis of linear models. If a linear model is available, designing a PID controller is a straightforward affair. However, sometimes technical constraints make it impossible to carry out experimental identification and modeling, or the validity of the linear model is very limited. The performance of the linear controller deteriorates as the linear model's behavior diverges from the actual process's behavior.

4 Internal Model Control

Internal model control (IMC) is an intensely used, well-established technique for designing intelligent controllers. This closed-loop control scheme explicitly uses a model of the dynamics of the plant to be controlled situated in parallel with the plant. Furthermore, it also contains another model of the inverse of the plant's dynamics situated in series with the process and acting as a controller.

One of the advantages of this control scheme is that dynamic analysis and robustness properties can be easily checked. However, inverting nonlinear models is not an easy task, and there may be no analytical solutions. Another associated problem is that inverting the process model can lead to unstable controllers when the system is a non-minimum phase system.

In this work, the inverse model G'_M (5) and the direct model G_M (6) are output error (OE) models. The NFI algorithm is used to create the models (both direct and inverse) on line. With each new input into the control scheme, both models are calculated. Using this neuro-fuzzy inference technique, the creation of the inverse model proves to be simpler and always offers a solution.

$$f(k) = G'_M(F(k), F(k-1), f(k-1)) \tag{5}$$

$$F(k) = G_M(f(k), f(k-1), F(k-1)) \tag{6}$$

where $F(k)$ is the cutting force estimated by the direct model and $f(k)$ is the feed rate calculated by the inverse model.

Once the models have been obtained, a low-pass filter is included in the control scheme. The filter is incorporated in the control system to reduce the high-frequency gain and to enhance the systems' robustness. It also works to soften fast, brusque signal changes, thus improving the controllers' response.

The direct model must be trained to learn the process's dynamics. A NFI system is used with a training set made up of input/output data, where the input is the feed rate, while the cutting force is used as the output variable (Fig.2a). In order to calculate the inverse model, instead of inverting the direct model found analytically, another NFI system is used, whose training set contains data with cutting-force values as the input and feed rate values as the output. This way,

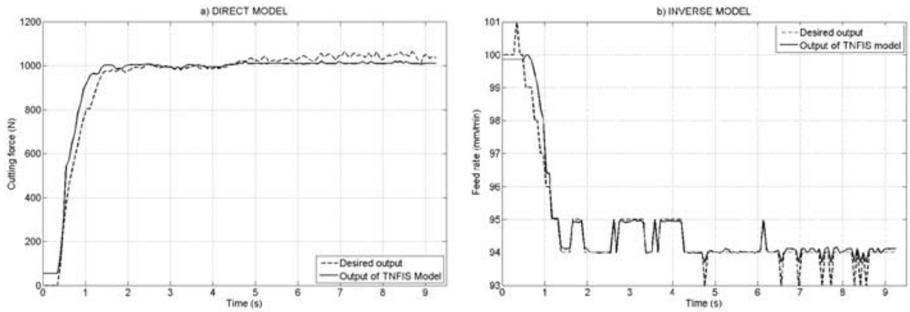

Fig. 2. a) Direct model and b) inverse model of the high-performance drilling process yielded by NFI

the system succeeds at learning the inverse dynamics of the high-performance drilling process (Fig.2b).

The training data for both the direct model and the inverse model were obtained from real drilling operations with test pieces made of A395 material (ASTM) under actual cutting conditions. Nevertheless, the training data set does not have to be very extensive, because representative values of each operating region are enough. The accuracy of the models depends on the choice of certain parameters of the NFI algorithm, such as the number of closest neighbors, the maximum number of iterations and the learning rate of the back-propagation algorithm, and the set-clustering threshold value (parameter of the clustering algorithm used). When choosing these parameters, the goal is to find a tradeoff between the accuracy and the quality of the dynamic response of the local models.

5 Experimental Results

In order to validate the networked intelligent control system by means of the suggested approach, it was decided to introduce the NFI-IMC in a real high-performance drilling process. Drilling processes are conducted in a machine tool Kondia HS1000 equipped with an open computer numerical control (CNC) Sinumerik 840D. A personal computer (PC1) is connected with the open CNC via a Profibus network. PC1 has a Windows 2000 operating system and performs three tasks. The first task is to measure force. The force is directly measured from PC1 with a Kistler 9257B dynamometer at a sampling frequency of 5kHz. The second task is communication with a second personal computer, PC2, via Ethernet with standard CORBA middleware. Filtered force F and other parameters and variables (e.g., spindle speed and depth of drill) are passed to PC2 in this way. The third task is to receive the control signal computed by PC2 with the data interface and synchronization tasks performed by commercial software (Labview, NC-DDE application) over Ethernet.

User interface, data visualization and record, and the IMC neuro-fuzzy system are implemented in PC2 (Networked control was done through PC2). Free distribution software (RTAI-Linux and Qt) is used in this level of the developed platform. PC2 cannot be connected directly to the CNC due to proprietary CNC software constraints. NFI-IMC was developed in C++ and the control scheme was incorporated to PC2. The NFI-IMC system processed the force measurement, calculated the control signal (f) and sent it to PC1, which later modified the corresponding variable in the open-architecture CNC.

Network-induced delays are usually not known a priori and are difficult to measure and estimate on line. The maximum delay, including network-induced delays and process dead time, can be estimated thus:

$$[\tau_{SC} + \tau_{CA} + \tau_{P}]_{MAX} = 0.4s \tag{7}$$

where τ_{SC} is the delay from the dynamometric sensor to PC1, τ_{CA} is the delay from PC1 to the CNC and τ_{P} is the intrinsic dead time of the drilling process.

It is important to note that the delay due to switched Ethernet is not remarkable for many real-time applications, thus indicating a very promising alternative for networked control systems.

$$\tau_{S2} + \tau_{A2} = 0.005s \tag{8}$$

where τ_{S2} is the delay from PC1 to PC2, τ_{A2} is the delay from PC2 to PC1.

Different drilling tests were conducted on A564 precipitation-hardening stainless steel test pieces (martensitic steel). This is a material used heavily in the naval and aerospace industry. The optimum conditions recommended by tool manufacturers for drilling are: 780 rpm (spindle speed), 2000 N (reference value of force) and 93 mm/min (feed rate[1]). The experiments were carried out using a Sandvik R840-1000-30-A0A 10-mm-diameter tool of solid hard metal with a TiN/TiAlN coating.

The NFI algorithm parameters chosen are the same for the direct model and the inverse model (with the exception of the training sets, which represent different dynamics although they contain the same data). The chosen number of closest neighbors (N_q) is 5, and the number of iterations of the back-propagation algorithm is 20, with a learning rate of 0.001. The whole number of input and output membership functions is directly related with the number of closest neighbors (N_q), and therefore the maximum number of membership functions is 5 but changes dynamically according each new datum and the ECM algorithm. The threshold value selected for set generation in the ECM algorithm is D_thr=1. According with these parameters and models (5) and (6), the mean time elapsed between data input and control signal computation by NFI-IMC was 2.3ms on the basis of an Intel Core 2 CPU (CPU 6400, 2.13 GHz, 0.98 GB RAM) and Windows XP operating system.

A comparative study was performed with another very well-known neuro-fuzzy technique (ANFIS) [4] in an IMC scheme reported in [12]. The summary of

[1] Feed rate is set initially in the CNC operating program in order to begin part program execution, and later it is manipulated by the NFI-IMC control system.

Table 1. Neuro-fuzzy algorithms in internal model control paradigm

Algorithm	ANFIS	NFI
System	Single-input/Single output	Single-input/Single output
Membership functions type	Gaussian	Gaussian
Number of membership functions	2	Variable each run (maximum 5)
Inference system	Takagi-Sugeno	Mamdani
Number of rules	2	Variable each run (maximum 5)
Iterations	5	20
Learning rate	0.01	0.0001
Training algorithms	Back propagation	Back propagation
Training data set	139 samples	139 samples
Validation data set	134 samples	5 samples
Training time	134.4 ms	2.06 ms
Mean computing time	0.01 ms	2.3 ms

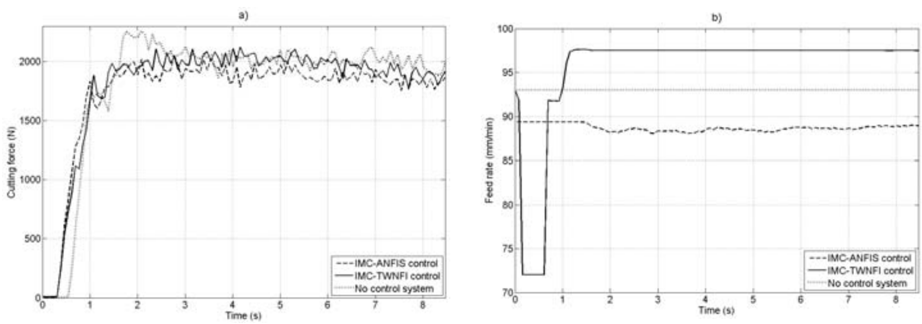

Fig. 3. a) Real system response and b) Control signal in operation with A564

the main characteristics of both neuro-fuzzy systems is shown in table 1. In order to compare both systems, the integral of time weighted absolute error ($ITAE$), the integral of the time square error ($ITSE$) and the integral of the square time multiplied by square error (IT^2SE) were used as performance indices. The overshoot (Ovt) was also included in the comparative study due to the influence of the transient dynamic on the useful tool life. The results of the experimental tests are shown in Fig.3 and in table 2. The dynamic behavior of ANFIS-IMC and NFI-IMC seems to be similar at first glance, but the NFI-IMC yields less overshoot than ANFIS-IMC. Furthermore, the NFI-IMC has better performance in $ITAE$, $ITSE$ and IT^2SE criteria than the ANFIS-IMC strategy. In general, NFI-IMC strategy outperforms the ANFIS-IMC approach.

In addition, there is a significant difference at the beginning and at the end of the operation. This means that the entry of the drill on the workpiece is much smoother and that the evacuation of the chip at the end is more correct (important factors for the tool wear and surface quality). Fig. 3 shows this situation.

Table 2. Performance indices for the experiments

Controller	$ITAE$	$ITSE$	IT^2SE	$Ovt(\%)$
Without Control	3.29	0.45	1.92	17.34
ANFIS-IMC	2.04	0.20	1.00	7.23
NFI-IMC	1.50	0.18	0.55	6.22

Therefore, for all the reason mentioned above, this experimental test serves to conclude that NFI-IMC strategy outperforms the ANFIS-IMC approach.

6 Conclusions

From the best of authors' knowledge, the NFI paradigm is applied for the first time to process control. The transductive NFI system is capable of creating local models of the high-performance drilling process for each new input into the control system. With the NFI-IMC system, the material removal rate is successfully increased. In addition, the good quality of the transient response and the overshoot free response contribute from the industrial viewpoint to get more out of the tool's useful life while increasing the material removal rate.

Thus it has been demonstrated that NFI is a simple, fast, precise, computationally viable tool for modeling manufacturing processes. The use of transductive modeling techniques in processes such as drilling proves to be beneficial. Another advantage of NFI is that it makes acceptable predictions with a very small number of data (e.g., the ANFIS-IMC system uses twice as many data in its training set).

The method developed herein is completely valid for control in other conditions. All that needs to be done is to enter in the training set experimental data under the required working conditions and adjust the parameters of the algorithms to match the required working conditions. In this paper a viable application of Ethernet in real-time networked process control is shown and is profiled as a low-cost solution.

References

1. Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks 11, 748–768 (2000)
2. Keller, J.M., Hunt, D.J.: Incorporating fuzzy membership functions into the perceptron algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI 7, 693–699 (1985)
3. Hunt, K.J., Sbarbaro, D.: Neural networks for nonlinear internal model control. In: IEE Proceedings D: Control Theory and Applications, vol. 138, pp. 431–438 (1991)
4. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23, 665–685 (1993)

5. Kim, J., Kasabov, N.: HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. Neural Networks 12, 1301–1319 (1999)
6. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Inc., New York (1998)
7. Atkeson, C.G.: Using local models to control movement. In: Advances in Neural Information Processing Systems, vol. 2, pp. 316–323 (1990)
8. Schaal, S., Atkeson, C.G.: Robot juggling: implementation of memory-based learning. IEEE Control Systems Magazine 14, 57–71 (1994)
9. Li, H.X., Deng, H.: An approximate internal model-based neural control for unknown nonlinear discrete processes. IEEE Transactions on Neural Networks 17, 659–670 (2006)
10. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally Weighted Learning for Control. Artificial Intelligence Review 11, 75–113 (1997)
11. Song, Q., Kasabov, N.K.: NFI: A neuro-fuzzy inference method for transductive reasoning. IEEE Transactions on Fuzzy Systems 13, 799–808 (2005)
12. Gajate, A., Haber, R.E.: Internal Model Control Based on a Neurofuzzy System for Network Applications. A Case Study on the High-Performance Drilling Process. IEEE Transactions on Automation Science and Engineering 6, 367–372 (2009)