Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio, Patras, GREECE 17-19 July, 2000

Applications of Fuzzy Neural Networks with Nonlinear Consequences to System Identification

Jamahl W. Overstreet and Anthony Tzes

Abstract— **The objective of this article is to formulate a generic Fuzzy Logic Identifier (FLI) with a neural network structure for identification purposes of nonlinear systems. This FLI extends the current limited representation of fuzzy models by modifying its consequence part as a ratio of polynomials of the input variable. The weights of the premise and consequence parts are tuned in an adaptive manner based on the backpropagation algorithm. The suggested scheme is applied in identifying the nonlinear aspects of friction in a dc–motor micromaneuvering system.**

I. Introduction

Fuzzy logic (FL) is a tractable scheme for identification of nonlinear systems with unknown dynamics. Several variations of FLI exist in the literature [1–5], where the tuning the fuzzy parameters is based on ad hoc procedure suitable for certain class of nonlinear systems.On the other hand, FLIs can be implemented based on neural networks (NNs) [6–10]. These NN–based FLIs are quite generic and the designer can address the FLI's robustness and convergence rate [6, 11].

In order to alleviate some of the intricacies in the defuzzyfication segment, the use of constant terms in the consequence is adopted. This dictates the use of a large number of rules for proper modeling of nonlinear systems. As a trade-off between increasing the number of rules versus the complexity of the defuzzyfication part, FLIs with primitive nonlinear consequence terms have appeared in the literature [6, 12–14].

The presented work extends the existing work on NN-based FLIs with nonlinear consequence terms by adopting a quite generic representation using ratio of polynomials with respect to the input variable.

For testing purposes, the suggested scheme is applied to the identification problem of the friction inherent in a dc-motor micromaneuvering system [15–17].

This article is organized in the following manner. The next section is devoted to the description of the structure of the NN-based FLI. Section III is concerned with the learning aspects of the FLI, while the appropriate modifications for the aforementioned identification problem of the dc–motor friction is presented in section IV. Concluding remarks are presented in the last section.

J.Overstreet is with Polytechnic University, Mechanical Engineering Department, Brrooklyn, NY 11201, U.S.A.
A.Tzes is with University of Patras, Electrical & Computer Engineering Department, Rio 26500, GREECE, E-mail: tzes@ee.upatras.gr

II. Neural Network-based Fuzzy Logic Identification

A NN-FLI system uses NNs to emulate its fuzzy reasoning [18, 19]. Adaptive NN-FLIs are autonomous units capable of: a) providing the expert rules for decision making and b) self-tuning the membership functions (MFs) of the utilized fuzzy sets via the training data. In the literature, several approaches have appeared to address the tuning process of the parameters used in these adaptive NN-FLIs [20–23], where NNs construct the MFs used in the premise and consequence parts.

In this study, the used NN–FLI has a structure with a typical structure appearing in Figure 1 for a system with two independent input variables and a generic ratio of polynomials used in the consequence part.

The NN-FLI implements the following set of rules, for a system with two (l) inputs (outputs) and an nth (mth) order numerator (denominator) polynomial

$$\text{IF } x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \text{THEN}$$

$$y_l \text{ is } \frac{a_{0l} + a_{1l}x_1 + a_{2l}x_2 + a_{3l}x_1^2 + a_{4l}x_1x_2 + a_{5l}x_2^2 + \cdots + a_{pl}x_2^n}{b_{0l} + b_{1l}x_1 + b_{2l}x_2 + b_{3l}x_1^2 + b_{4l}x_1x_2 + b_{5l}x_2^2 + \cdots + b_{ql}x_2^m},$$

where p (q) and q is the cardinality of the numerator (denominator) polynomial.

A. NN-FLI Consequence Structure

The NN-FLI consequence portion is composed of eight layers. The first layer, Layer N, is constructed so that all inputs can be multiplied by the next layer to the number of powers required; Layer E assembles all the terms for the polynomial in the numerator; Layer F combines the different terms from Layer E and assembles the actual numerator polynomial; Layer I assembles all the terms for the denominator polynomial; Layer J combines the different terms from Layer I and assembles the actual denominator polynomial; Layer G connects the fuzzy rules with numerator and denominator polynomials; while Layer H acts as a defuzzifier for the output values.

The inputs and outputs of each layer appear in the sequel, where the following notation is used: I_x^X and O_x^X represents the xth input and output of a perceptron in Layer X, respectively, W_{yx}^{YX} symbolizes the weight that connects the perceptron x in Layer X to the perceptron y in Layer Y.

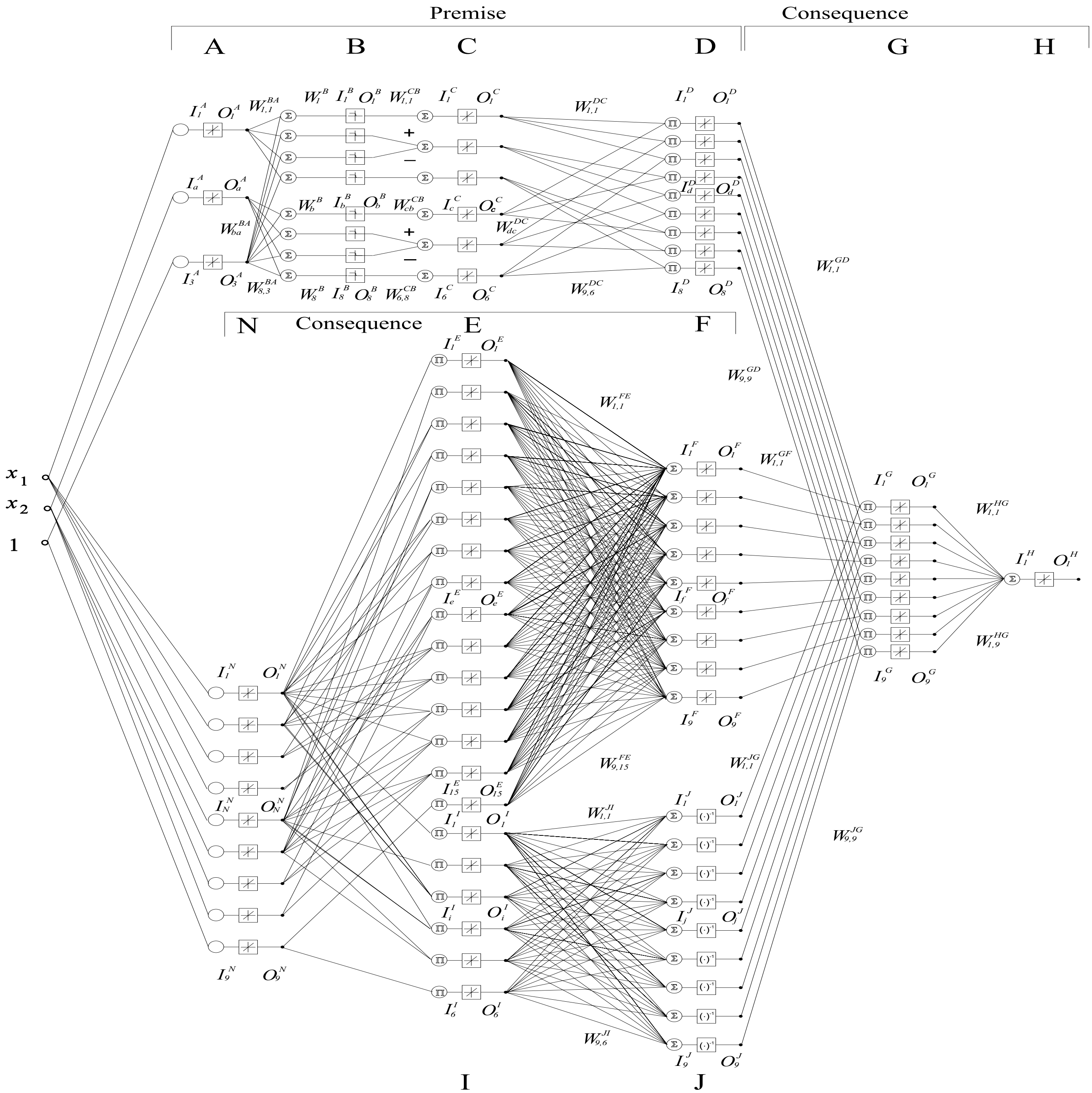

Fig. 1. Generic NN-FLI Structure

$$
\begin{array}{lll}
\mathrm{N} & I_n^N = \sum_{a=1}^{A} W_{na}^{NA} x_a^A & O_n^N = I_n^N \\
\mathrm{E} & I_e^E = \prod_{n=1}^{N} W_{en}^{EN} O_n^N & O_e^E = I_e^E \\
\mathrm{F} & I_f^F = \sum_{e=1}^{E} W_{fe}^{FE} O_e^E & O_f^F = I_f^F \\
\mathrm{I} & I_i^I = \prod_{n=1}^{N} W_{in}^{IN} O_n^N & O_i^I = I_i^I \\
\mathrm{J} & I_j^J = \sum_{i=1}^{I} W_{ji}^{JI} O_i^I & O_j^J = (I_j^J)^{-1} \\
\mathrm{G} & I_g^G = W_{gd}^{GD} O_d^D W_{gf}^{GF} O_f^F W_{gj}^{GJ} O_j^J & O_g^G = I_g^G \\
\mathrm{H} & I_h^H = \sum_{g=1}^{G} W_{hg}^{HG} O_g^G & O_h^H = I_h^H
\end{array}
$$

where n is the highest degree amongst the numerator and denominator polynomials (Layer E and Layer I); a is the number of independent inputs; e (i) is the number of terms in the numerator (denominator) polynomial; f the number of rules in consequence; $j = g$ the number of rules in consequence, and h the number of outputs.

B. NN-FLI Premise Structure

The premise structure uses NNs with sigmoid activation functions for approximating the applied MFs. A typical premise segment with two inputs ($a = 2$), six ($c = 6$) distinct MFs, and nine combinations of fuzzy sets/rules ($d = 9$) is shown in Figure 2.

Each MF is constructed by the addition/subtraction and appropriate transition of sigmoid functions $f_s(x) = (1 + \exp(-x))^{-1}$. In Figure 2, eight ($b = 8$) sigmoids are used to construct the MFs. According to the notation adopted in this figure, the function of each layer is the following: Layer A is the "input" layer, layer B (C) is used

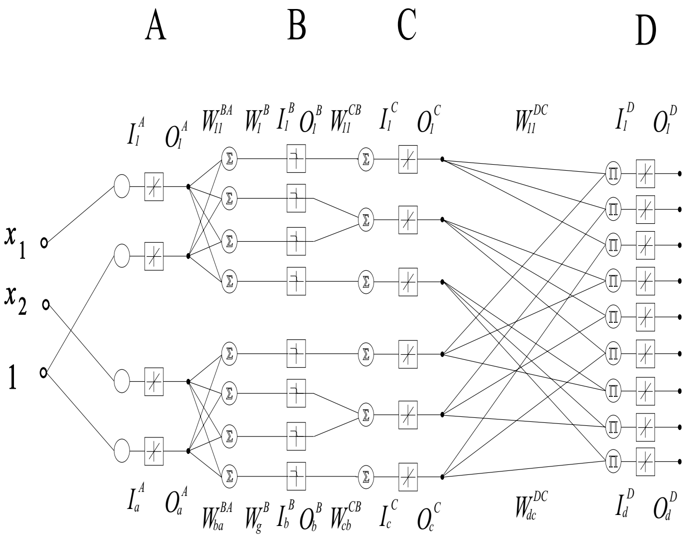

Fig. 2. NN-FLI Premise Segment

to construct the sigmoid functions (MFs), and the rightmost layer D: a) defines the fuzzy rules through all possible combinations of the MFs defined in layer C, and b) defuzzifies the output of each premise through the application of the "center of gravity" defuzzification algorithm. The input/output relationships for each layer are:

A	$I_a^A = x_a$	$O_a^A = I_a^A$
B	$I_b^B = \sum_{a=1}^{A} W_{ba}^{BA} O_a^A$	$O_b^B = f_s(W_b^B I_b^B)$
C	$I_c^C = \sum_{b=1}^{B} W_{cb}^{CB} O_b^B$	$O_c^C = I_c^C$
D	$I_d^D = \prod_{c=1}^{C} W_{dc}^{DC} O_c^C$	$O_d^D = \frac{I_d^D}{\sum_{d=1}^{D} I_d^D}$

where the weights W_{ba}^{BA} shift the MFs in the universe of discourse, the W_b^B weights adjust the MFs' steepness, and the outputs of layer C are the overall defined MFs $A_{ji},\ j = 1,\ldots,J,\ i = 1,\ldots,I$, where I is the number of independent variables, and J the number of fuzzy sets applied to each particular input variable.

III. NN-FLI learning

The parameters of the NN-based FLI that are adjusted are: a) the W_b^B, W_{ba}^{BA} weights which affect the characteristics of the MFs used in the premise part, and b) the W_{fe}^{FE} and W_{ji}^{JI} terms which correspond to the coefficients of the polynomials used in the consequence part.

Throughout the learning process, attention must be paid to the satisfaction of several constraints that affect the behavior of the FLI [24]. As an example, consideration must be given to the output of layer J being the inverse of its input ($O_j^J = \left(I_j^J\right)^{-1}$); precaution should be given to avoid any division by zero. Other "constraints" related to the use of sigmoids to approximate the MFs can be found in [24].

The classical error backpropagation learning scheme was selected for tuning the aforementioned parameters. In this supervised training mode, the weights are updated in the form $W' = W + \Delta W$, where W' (W) is the updated (current) weight, and ΔW its incremental change. The weights' change is aligned with the minimization of an error norm function defined as $E = \frac{1}{2}\sum_{h=1}^{H}\left(d_h^H - O_h^H\right)^2$, where d_h^H is hth training signal corresponding to the output, O_h^H.

The backpropagation algorithm (with the momentum-term enhancement [25]) adjusts the incremental change at time k as $\Delta W(k) = -\eta\sum_{n=0}^{N}\alpha^n \nabla E(k-n) = -\eta\nabla E(k) + \alpha\sum_{n=1}^{N}\Delta W(k-n),\ \eta,\alpha > 0$. The update laws for individual weights appear in the sequel, where for brevity, superscripts representing layer possessions have been removed, (i.e, $I_x \leftrightarrow I_x^X$)

$$W'_{fe} = W_{fe} - \eta_f \left\{\sum_{h=1}^{H}\sum_{g=1}^{G}(d_h - O_h)O_d O_j\right\} O_e \quad (1)$$

$$W'_{ji} = W_{ji} + \eta_j \left\{\sum_{h=1}^{H}\sum_{g=1}^{G}\frac{(d_h - O_h)O_d O_f}{\left(\sum_{i=1}^{I} W_{ji} O_i\right)^2}\right\} O_i \quad (2)$$

$$W'_{ba} = W_{ba} - \eta_b \sum_{h=1}^{H}\sum_{d=1}^{D}\frac{(d_h - O_h)O_f O_j \Gamma_{dc}\Psi_d W_{cb}}{\left(\Xi_{dc}\sum_{b=1}^{B} W_{cb}O_b + \Psi_d\right)^2} \times O_b(1 - O_b)W_b O_a \quad (3)$$

$$W'_{b} = W_{b} - \eta_b \sum_{h=1}^{H}\sum_{d=1}^{D}\sum_{a=1}^{A}\frac{(d_h - O_h)O_f O_j \Gamma_{dc}\Psi_d W_{cb}}{\left(\Xi_{dc}\sum_{b=1}^{B} W_{cb}O_b + \Psi_d\right)^2} \times O_b(1 - O_b)W_{ba} O_a\ , \quad (4)$$

where Γ_{dc}, Ξ_{dc} and Ψ_d are multipliers that are functions of O_c and O_d [24] $\Gamma_{dc} = \Gamma_{dc}(O_c, O_d)$, $\Xi_{dc} = \Xi_{dc}(O_c, O_d)$, $\Psi_d = \Psi_d(O_c, O_d)$.

IV. NN–based FLI of a DC–Motor Experimental Setup

The aforementioned NN–based FLI was applied for the identification problem of the friction inherent in a dc–motor experimental setup. The friction characteristic [26] relates the applied voltage command V_u to the motor with the dc–motor's angular velocity ω. The system was excited with a sinusoidal–sweep signal

$$V_u(t) = \sum_{i=0}^{L} k_s \left\{\left(t - \frac{2\pi k}{\omega_i}\right)\sin(\omega_i t) - \left(t - \frac{2\pi k}{\omega_{i-1}}\right)\sin(\omega_{i-1} t)\right\} u\left(t - \frac{1}{2\pi k\omega_i}\right);$$

the typical relationship given 2,500 observed data points $(V_u(0.1k), \omega(0.1k),\ k = 0,\ldots,2499)$ from the experimental setup [27] appears in Figure 3.

NN–FLIs were trained with different MFs based on the assumption of using the following fuzzy rules

$$R^d : \text{If } V_u \text{ is } A_{d,c1} \text{ then } \omega = \frac{\sum_{i=0}^{n}\alpha_{d,i}\left(V_u\right)^i}{\sum_{j=0}^{m}\alpha_{d,n+j+1}\left(V_u\right)^j}\ , \quad (5)$$

where an nth (mth) order numerator (denominator) polynomial is used in the consequence part.

In this study, FLIs with different MFs ($d = 2, 3, 4$) and polynomial configurations ($n = 0,\ldots,3,\ m = 0$) were used to identify the friction characteristics. The fuzzy neural networks were trained in an off–line manner over a set of 100,000 repeated training cycles.

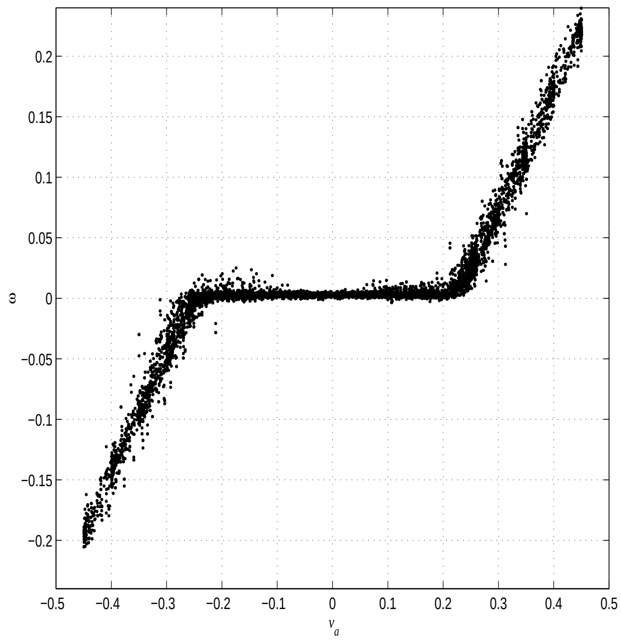

Fig. 3. DC-Motor Friction Characteristic

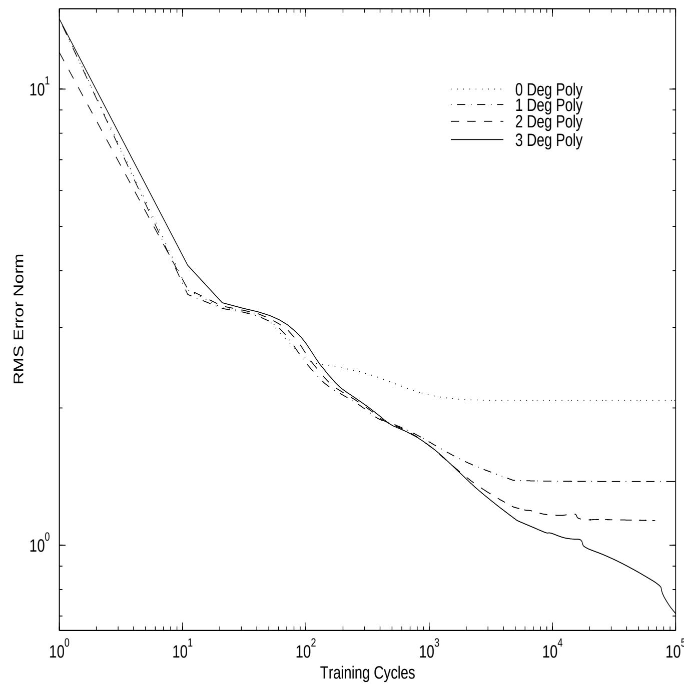

Fig. 4. Error Convergence Curve for NN–FLI with 3 MFs

The graph of the error convergence vs. the training cycle during the learning process of the FLI appears in Figure 4. In each case, the plotted error value corresponds to $\sum_{k=0}^{2499}\left[\omega\left(kT_s\right)-\hat{\omega}_j\left(kT_s\right)\right]^2$, where $\hat{\omega}_j\left(kT_s\right)$ is the kth–sample estimated system output during the jth–training cycle. The results indicate that when the number of parameters that needs to be updated is increased (i.e., 3rd order polynomial), the convergence rate is slower and the converged value is smaller.

The graph of the resulting MFs over their universe of discourse after the 100,000th training cycle is shwon in Figure 5. The input's universe of discourse for this experiment was $V_u \in [-0.45, 0.45]$. In each sub-case the "converged" graphs of all MFs are displayed, where it is shown that while there is significant shift in the center of MFs its stepness did not change considerably.

For each sub-case, based on the resulting MFs and consequence polynomials (after the 100,000th cycle), the plot of the estimated friction relationship from the FNN–based model is provided in Figure 6.

Typical time–history plots (one for each sub–case) indicating the resulting $\hat{\omega}(t), t \in [35, 80]$ sec (solid line) from the NN–FLI model overlaid with the actual angular velocity $\omega(t)$ (dashed line) from the dc–motor are presented in Figure 7.

V. Conclusion

A generic NN–based FLI was designed in this article. This identifier extends the current results in the literature by using an enhanced consequence part with ratio of polynomials defined over the input variable. The identifier has a NN–structure and the classical error backpropagation routine is used to adjust the parameters affecting the shape of

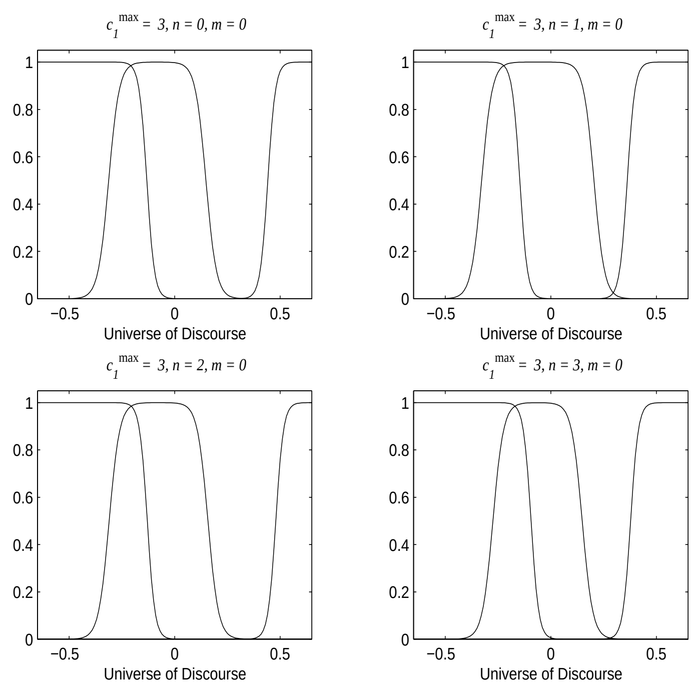

Fig. 5. $A_{i,c1}(V_u),\ i = 1, 2, 3$ MF from NN–FLI Model

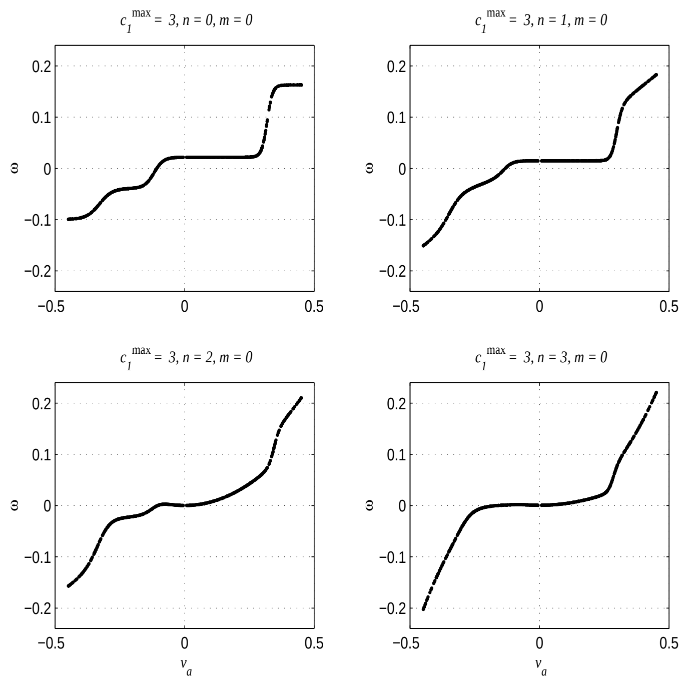

Fig. 6. $\hat{\omega}(V_u)$ curve from NN–FLI model with 3 MFs

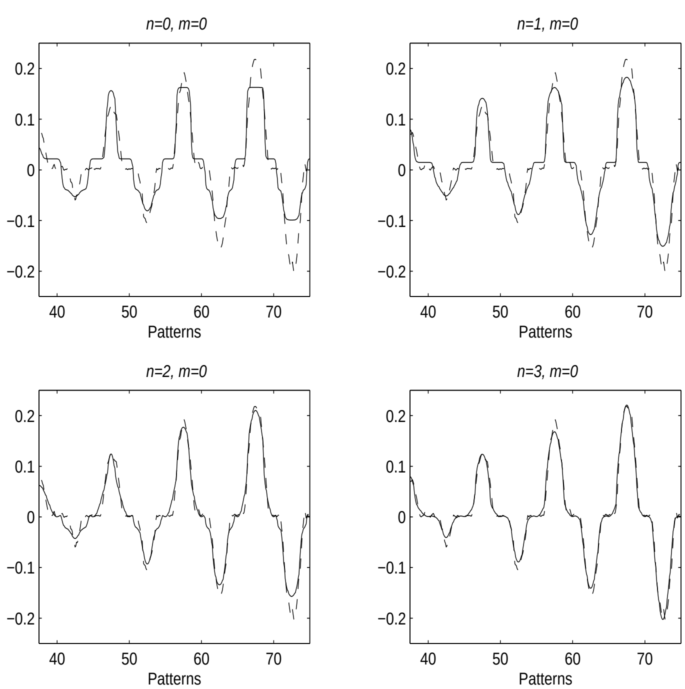

Fig. 7. Actual ($\omega(t)$) and 3MF-NNFLI output ($\hat{\omega}$)

the MFs and the coefficients of the polynomials. The suggested scheme is applied in experimental studies for identifying the friction characteristics of a dc–motor setup.

References

[1] Y. Joo, H. Hwang, K. Kim, and K. Woo, "Linguistic model identification for fuzzy systems," *Electronics Letters*, vol. 31, no. 4, pp. 330–331, 1995.

[2] C. Bor-Sen, L. Ching-Hsiang, and C. Yeong-Chan, "H^∞ tracking design of uncertain nonlinear SISO systems: Adaptive fuzzy approach," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 1, pp. 32–43, 1996.

[3] A. Lotfi, H. Andersen, and A. Tsoi, "Matrix formulation of fuzzy rule-based systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, no. 2, pp. 332–340, 1996.

[4] C. Lin and C. Lee, "Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 1, pp. 46–63, 1994.

[5] R.-N. Singh and W. Bailey, "Fuzzy logic applications to multisensor-multitarget correlation," *IEEE Transactions on Aerospace and Electronics Systems*, vol. 33, no. 3, pp. 752–769, 1997.

[6] A. Tzes and P.-Y. Peng, "Fuzzy neural network control for DC-Motor micromaneuvering," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 119, no. 2, pp. 312–315, 1997.

[7] S.-J. Huang and R.-J. Lian, "A hybrid fuzzy logic and neural network algorithm for robot motion control," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 3, pp. 408+, 1997.

[8] C. Kambhampati, R. Craddock, M. Tham, and K. Warwick, "Inverting recurrent neural networks for internal model control of nonlinear systems," vol. 2, pp. 975+, 1998.

[9] R. King, "Artificial neural networks and computational intelligence," *IEEE Computer Applications in Power*, vol. 11, no. 4, pp. 14–25, 1998.

[10] S. Weerasooriya and M. A. El-Sharkawi, "Laboratory implementation of a neural network trajectory controller for a dc motor," *IEEE Transactions on Energy Conversion*, vol. 8, no. 1, 1993.

[11] J. Zhang and A. Morris, "Fuzzy Neural Networks for Nonlinear Systems Modelling," *IEE Proceedings-D: Control Theory and Applications*, vol. 142, no. 6, p. 551, 1995.

[12] J. T. Teeter, M. y. Chow, and J. J. B. Jr., "A novel fuzzy friction compensation approach to improve the performance of a dc motor control system," *IEEE Transactions on Industrial Electronics*, vol. 42, no. 1, pp. 516–521, 1995.

[13] B. S. Butkiewicz, "Steady-state error of a system with fuzzy controller," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 6, pp. 855–860, 1998.

[14] S. Bolognani and M. Zigliotto, "Hardware and software effective configurations for multi-Input fuzzy logic controllers," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 173+, 1998.

[15] S. Weerasooriya and M. A. El-Sharkawi, "Identification and control of a DC motor using back-Propagation neural networks," *IEEE Transactions on Energy Conversion*, vol. 6, no. 4, 1991.

[16] J. T. Teeter, M. y. Chow, and J. J. B. Jr., "A novel fuzzy friction compensation approach to improve the performance of a DC motor control system," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 1, pp. 113+, 1996.

[17] S. Saneifard, N. R. Prasad, H. A. Smolleck, and J. J. Wakileh, "Fuzzy-Logic-Based speed control of a shunt DC motor," *IEEE Transactions on Education*, vol. 41, no. 2, pp. 159+, 1998.

[18] T. Arnould and S. Tano, "Interval-valued fuzzy backward reasoning," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 4, pp. 425–437, 1995.

[19] F. G. Jota and P. R. S. Jota, "High-impedance fault identification using a fuzzy reasoning system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 6, pp. 990+, 1995.

[20] S. Chiu, "Using fuzzy logic in control applications: beyond fuzzy pid control," *IEEE Control Systems Magazine*, vol. 18, no. 5, pp. 100–104, 1998.

[21] J. Chong, D. Lee, K. Jung, and W. Kim, "Fuzzy neural network with rule generation," *Electronic Letters*, vol. 32, no. 9, pp. 825–827, 1996.

[22] D. A. Linkens and H. O. Nyongesa, "Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications," *IEE Proceedings-D: Control Theory and Applications*, vol. 143, no. 4, pp. 367–386, 1996.

[23] S. Wang and N. P. Archer, "A neural network based fuzzy set model for organization decision making," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 28, no. 2, 1998.

[24] J. Overstreet, "Remote monitoring and advanced control designs for internet–based experimentation," Master's thesis, Polytechnic University, Mechanical Engineering Department, 2000.

[25] J. M. Zurada, *Introduction to Artifical Neural Systems.* St. Paul, MN: West Publishing Co., 1992.

[26] J. Overstreet and A. Tzes, "Fuzzy Neural Network–based Internal Model Contol for DC–Motor Micromaneuvering," in *8th IEEE Mediterranean 2000 Conference on Control and Automation*, (Patras, Greece), July 2000.

[27] A. Tzes, P.-Y. Peng, and J. Guthy, "Genetic–based Fuzzy Clustering for DC–motor Friction Identification and Compensation," *IEEE Trans. on Control Systems Technology*, 1997.