# On a Hybrid PI-Neuro-Fuzzy Controller Meant for a Class of Non-minimum Phase Systems

Radu-Emil Precup and Stefan Preitl
“Politehnica” University of Timisoara, Department of Automation
Bd. V. Parvan 2, RO-1900 Timisoara, Romania
Phone: +40-56-204333 ext. 173, 155, 151, Fax: +40-56-192049
email:{rprecup,spreitl}@aut.utt.ro

ABSTRACT: The non-minimum phase systems with transfer functions having “unstable” zeros are difficult to control due to the unusual behaviour with respect to time and to the shape of the open-loop Bode plot. The paper proposes a hybrid PI-neuro-fuzzy controller meant for controlling this class of systems. The controller is based on self-tuning the free parameter of a PI-fuzzy controller by introducing a single neuron with a simplified back-propagation learning algorithm. Digital simulation results prove that the proposed controller provides very good control system performance in comparison with a classical PI controller and a PI-fuzzy controller.

KEYWORDS: standard PI-fuzzy controller, PI-neuro-fuzzy controller, on-line adaptation, non-minimum phase system, reference model, back-propagation learning algorithm

## INTRODUCTION

The controlled plant (briefly, CP) taken into consideration belongs to the class of second order non-minimum phase systems (NFSs) with two negative real poles and one positive zero having the structure presented in Fig.1, where the values of the variable positive parameters {$k_{CP}$, $T_1$, $T_2$, $\acute{a}_3$, $T_3$} depend on the operating point, u represents the control signal, v is the disturbance input, and y stands for the controlled output.

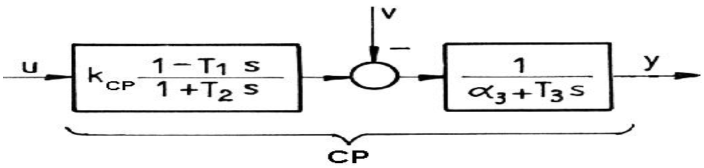


Figure 1: Block diagram of CP.

One of the properties of this class of NFSs is that in the first part of system response a tendency exists of variation in an opposite direction to the direction of control signal variation leading to the existence of downshoot that can produce serious incidents in control system (CS) operation. On the other hand, the rapid decrease of open-loop Bode plot for increasing values of the frequency can be observed for the considered class of NFSs. These peculiar behaviours can be seen as the starting point to reconsider the well accepted methodology of fuzzy controllers development.

The class of CPs from Fig.1 is widely used in its simplest linearised version in several applications as those reported by Klefenz (1973), Hoppe and Tešnjak (1983), Svaricek (1995), Precup (1995). The goal of controller development, meant for controlling the CP from Fig.1, is to ensure as good as possible dynamic as well as steady-state CS performance. Therefore, the paper develops a hybrid PI-neuro-fuzzy controller (PI-NFC) based on the on-line adaptation of a PI-fuzzy controller (PI-FC).

## STANDARD VERSION OF PI-FUZZY CONTROLLER

The standard version of PI-FC considered in this paper is characterised by the introduction of the dynamics in the fuzzy controller by differentiating the control error $e_k = w_k - y_k$ (k – the index of the current sampling interval) and integrating the increment of control signal $\Delta u_k = u_k - u_{k-1}$, Fig.2, Precup and Preitl (1993), Bühler (1994).

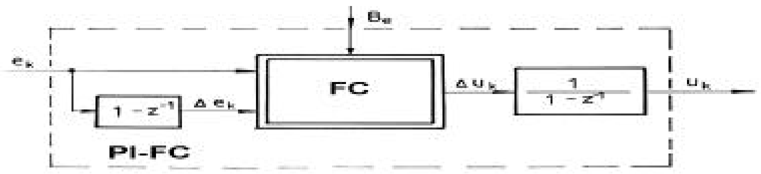


Figure 2: Block diagram of standard PI-FC.

The strictly speaking fuzzy controller without dynamics from Fig.2 (the block FC0 is characterised by the followings:

- the fuzzification is performed by means of the membership functions from Fig.3 pointing out the PI-FC tuning parameters $\{B_e, B_{\Delta e}, B_{\Delta u}\}$;

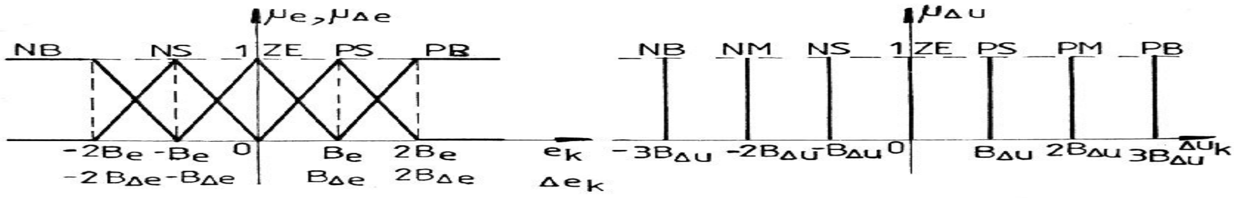


Figure 3: Membership functions of standard PI-FC.

- the inference engine operates on the basis of Mamdani's max-min compositional rule of inference assisted by the MacVicar-Whelan type decision table presented in Table I;

Table I:. Decision table of standard PI-FC.

| $\Delta e_k$ \ $e_k$ | NB | NS | ZE | PS | PB |
|---|---|---|---|---|---|
| PB | ZE | PS | PM | PB | PB |
| PS | NS | ZE | PS | PM | PB |
| ZE | NM | NS | ZE | PS | PM |
| NS | NB | NM | NS | ZE | PS |
| NB | NB | NB | NM | NS | ZE |

- the center of gravity method is employed for defuzzification.

The development of the standard PI-FC involves the following steps to be proceeded in terms of Precup and Preitl (1993):

- determine by a classical design method the parameters $\{k_C, T_i\}$ of the conventional PI controller (to be replaced by the PI-FC) with the transfer function:

$$H_C(s) = k_C\,(1 + 1/(sT_i)); \qquad (1)$$

- choose the sampling period h, digitize and obtain the parameters $\{K_P, K_I, \alpha\}$ of the quasi-continuous digital PI controller:

$$\Delta u_k = K_P \Delta e_k + K_I e_k = K_P(\Delta e_k + \alpha e_k), \qquad (2)$$

with $\Delta e_k = e_k - e_{k-1}$ – the increment of control error;

- apply the relations (3):

$$B_{\Delta e} = \alpha\, B_e, \quad B_{\Delta u} = K_I\, B_e, \qquad (3)$$

where:

$$\alpha = K_I/K_P = 2h/(2T_i - h), \quad K_I = k_C h/T_i, \qquad (4)$$

in the case of Tustin's digitization method.

The free parameter $B_e$ is usually chosen by heuristic rules in accordance with the experience of the control systems specialist; this is the reason why $B_e$ appears as an input to the PI-FC from Fig.2.

The aim of the paper is to obtain CS performance enhancement by the development of a PI-NFC which performs the on-line adaptation of the PI-FC by ensuring the self-tuning of the parameter $B_e$.

## PI-NEURO-FUZZY CONTROLLER

The structure of the PI-NFC is shown in Fig.4, and the additional blocks are described as follows.

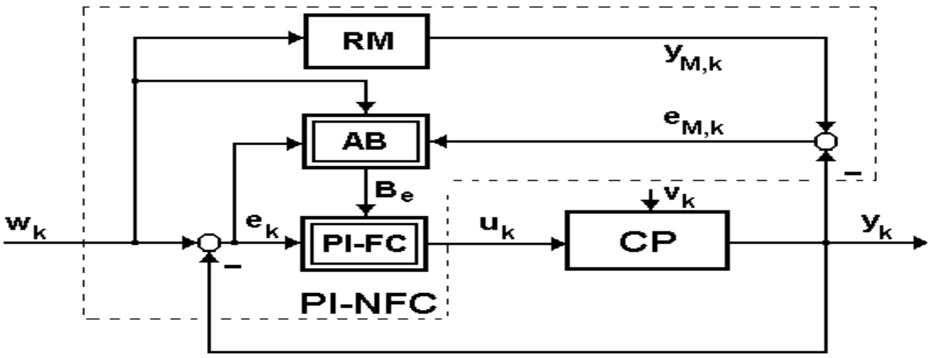


Figure 4: Simplified block diagram of CS with PI-NFC.

REFERENCE MODEL (RM)

Model-based control structures represent attractive solutions which can ensure good dynamic and steady-state control system performance, Morari and Zafiriou (1989), De Neyer and Gorez (1993). For the given NFS the following RM can fulfil the desired CS performance:

$$H_{RM}(s) = (1 - sT_1)/(1 - sT_{imp})^2, \quad (5)$$

where the CS can be specified / imposed by a proper choice of the time constant $T_{imp}$ (the increase of $T_{imp}$ with respect to $T_1$ determines downshoot alleviation).

Digitizing (5) by Tustin's method results in the discrete equation of RM ($y_{M,k}$ – the RM output):

$$y_{M,k} = - d_1 y_{M,k-1} - d_2 y_{M,k-2} - c_0 e_{M,k} + c_1 e_{M,k-1} + c_2 e_{M,k-2}, \quad (6)$$

where:

$$d_1 = 2(T_{imp}-h)/(2T_{imp}+h),\ d_2 = (2T_{imp}-h)^2/(2T_{imp}+h)^2,\ c_0 = h(2T_1-h)/(2T_{imp}+h)^2,$$
$$c_1 = 2h/(2T_{imp}+h)^2,\ c_2 = (2T_1+h)/(2T_{imp}+h)^2. \quad (7)$$

ADAPTATION BLOCK (AB)

The AB is a nonlinear block consisting of a single unbiased neuron with a linear activation function; its structure is not presented here, but it will result from the relations to be presented in the sequel.

The AB performs an on-line adaptation of the parameter $B_e$ by minimising the cost function CF defined in (8):

$$CF = 0.5 \sum_k (y_{M,k}-y_k)^2 = 0.5 \sum_k e_{M,k}^2, \quad (8)$$

where $e_{M,k} = y_{M,k} - y_k$ represents the model following error.

A similar approach is used with very good results in Chen and Linkens (1998), where it performs the on-line adaptation of the shapes (scaling) of the output membership functions. In turn, the present approach deals with the on-line adaptation of the shapes (scaling) of all membership functions because the modification of $B_e$ (see the relation (3) in connection with Fig.2) affects all membership functions.

The well-known Rumelhart-Hinton-Williams back-propagation algorithm with a momentum term for weight updating is used for the sake of self-tuning the parameter $B_e$. It is characterized by, Chen and Linkens (1998):

$$q_k = q_{k-1} + \Delta q_k, \quad (9)$$
$$\Delta q_k = - \eta\, \partial CF/\partial q_k + \lambda\, \Delta q_{k-1}, \quad (10)$$

where $q_k$ is the input weight of the neuron, and $\eta, \lambda \in [0, 1]$ stand for the learning rate and the momentum factors, respectively.

By expressing the derivative of the cost function:

$$\partial CF/\partial q_k = (\partial CF/\partial y_k)(\partial y_k/\partial B_e)(\partial B_e/\partial y_k), \quad (11)$$

and computing the partial derivatives (the linear activation function with its derivative equal to 1 reduces with one the number of partial derivatives):

$\partial CF/\partial y_k = -(y_{M,k} - y_k) = -e_{M,k}$, (12)

$\partial B_e/\partial q_k = -e_k$, (13)

only the derivative ($\partial y_k/\partial B_e$) remains to be computed. But, for slow plants (and this is the case of the considered plant), for a simpler implementation it is fully justified to make the replacement:

$\partial y_k/\partial B_e = \Delta y_k/\Delta B_e$. (14)

Then, the substitution of (12) … (14) into (11) and, then, into (10), results in the following relatively simple relation:

$\Delta q_k = -\eta\, e_{M,k}(\Delta y_k/\Delta B_e)e_k + \lambda\, \Delta q_{k-1}$. (15)

Therefore, the AB is characterised by the relations (8) and (15) together with the relation (16) for obtaining the output $B_e$ from the input $e_k$ (the activation function is absent as it is accepted to be a linear one):

$B_e = q_k\, e_k$. (16)

The parameters $\eta$ and $\lambda$ are obtained by heuristic rules in connection with the desired CS performance.

## DIGITAL SIMULATION RESULTS

For testing the proposed solution of hybrid PI-NFC the plant from Fig.1 was considered with the following values of its parameters, Precup (1995): $k_{CP} = 1$, $T_1 = 2.2$ sec, $T_2 = 1.1$ sec, $T_3 = 6.8$ sec.

The assessment of the performance ensured by the PI-NFC is done by performing a comparison with the cases of controlling the same plant with two controllers: a PI controller (PI-C) designed in the frequency domain, and a PI-fuzzy controller (PI-FC) with a constant $B_e$: $B_e = 0.5$. The comparison is done by the analysis of CS performance defined in the unit step responses with respect to the reference input w (of the type illustrated in Fig.5) and with respect to the disturbance input v (of the type illustrated in Fig.6). In Fig.5 and Fig. 6 y stands for the controlled output and t for time (in seconds).

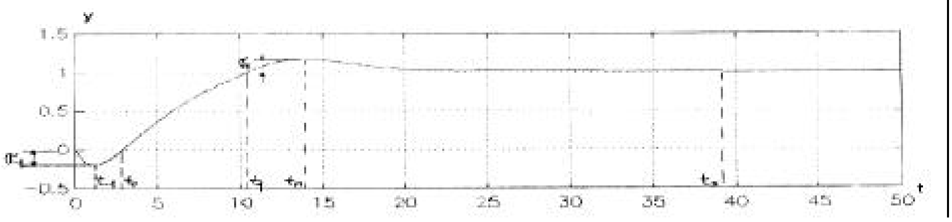

Figure 5: Definition of CS performance in the unit step response with respect to w.

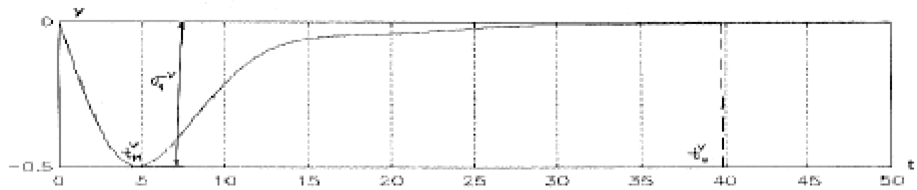

Figure 6: Definition of CS performance in the unit step response with respect to v.

For the sake of a brief presentation only the following values of parameters are listed here: $k_C = 1$, $T_i = 6.8$ sec, $T_{imp} = 5$ sec, $h = 0.05$ sec.

Table II and Table III present CS performance with respect to w and v, respectively.

Table II:. CS performance with respect to the unit step modification of w.

| Controller type | CS performance | | | | | |
|---|---|---|---|---|---|---|
| | $t_{-1}$ [sec] | $\sigma_{-1}$ [%] | $t_0$ [sec] | $t_M$ [sec] | $\sigma_1$ [%] | $t_s$ [sec] |
| PI-C | 2.7 | 20.78 | 3.5 | 14.45 | 9.26 | 24.3 |
| PI-FC | 2.7 | 14.31 | 4.1 | - | - | 27.1 |
| PI-NFC | 2.75 | 5.58 | 4.15 | - | - | 19.3 |

Table II and Table III show that:

- with respect to w: the PI-NFC ensures much more better CS performance (excepting the values of $t_{-1}$ and $t_0$);
- with respect to v: the PI-NFC ensures better CS performance (excepting the values of $t_s^v$).

Table III:. CS performance with respect to the unit step modification of v.

| Controller type | CS performance | | |
|---|---|---|---|
| | $t_M^v$[sec] | $\sigma_1^v$[%] | $t_s^v$[sec] |
| PI-C | 6.5 | 51.04 | 28.5 |
| PI-FC | 5.75 | 46.26 | 28.35 |
| PI-NFC | 5.6 | 45.72 | 28.7 |

## CONCLUSIONS

The paper proposes a hybrid PI-NFC based on self-tuning the free parameter of a PI-FC by means of a single unbiased neuron with a simplified back-propagation learning algorithm with a momentum term for weight updating. This simplified structure was chosen for ensuring the goal of an as simple as possible implementation.

Part of the digital simulation results illustrated in the paper prove that the PI-NFC is successful when it copes with control of non-minimum phase systems by ensuring control system performance enhancement in comparison with the case of a classical PI controller and of a PI-fuzzy controller.

The symbols used in the paper (especially in its “neural” part) are not those widely used in the field of neural networks. But, the symbols from the field of automation were used because it is considered that they better point out the information flow as part of the PI-NFC.

Although the proposed controller is relatively simple (but, nevertheless, efficient), the paper emphasises the potential of the merge between fuzzy sets and neural networks in the case of process control.

## ACKNOWLEDGEMENTS

Thanks from behalf of the authors go to Dr. Minyou Chen from University of Sheffield, UK, for offering very good references in this field, and to National Council for Higher Education Research (CNCSIS) from Bucharest, Romania, for the financial support through two grants.

## REFERENCES

Bühler, H., 1994, “Réglage par logique floue”, Presses Polytechniques et Universitaires Romandes, Lausanne.
Chen, M.; Linkens, D.A., 1998, “A Hybrid Neuro-fuzzy PID Controller”, Fuzzy Sets and Systems, vol. 99, pp. 27 – 36.
De Neyer, M.; Gorez, R., 1993, “Fuzzy and Quantitative Model-based Control Systems for Robotic Manipulators”, International Journal of Systems Science, vol. 24, no. 10, pp. 1863 – 1883.
Hoppe, M.; Tešnjak, S., 1983, “Modellbildung und Simulations des dynamisches Verhaltens eines Wasserkraftanlage”, Schriftenreihe des Lehrstuhls fur Mess- und Regelungstechnik, Ruhr-Universität Bochum, no. 20.
Klefenz, G., 1973, “Die Regelung von Dampfkraftwerken”, Bibliographisches Institut, Mannheim.
Morari, M.; Zafiriou, E., 1989, “Robust Process Control”, Prentice Hall, Englewood Cliffs/NJ.
Precup, R.-E., 1995, “On the Parameter Adaptation of a Fuzzy Controller Meant for the Speed Control of Hydrogenerators”, Third EUFIT’95 European Congress, Aachen, vol. 2, pp. 1105 - 1109.
Precup, R.-E.; Preitl, S., 1993, “Fuzzy Control of an Electrohydraulic Servosystem under Nonlinearity Constraints”, First EUFIT’93 European Congress, vol. 3, pp. 1524 - 1530.
Svaricek, F., 1995, “A Nonlinear Controller for the Optimum Running of a Nonminimum-phased Continuous Stirred Tank Reactor”, ECC’95 European Control Conference, Rome, pp. 1888 - 1893.