On the Stable Design of Stable Fuzzy Control Systems with Iterative Learning Control

Stefan Preitl, Radu-Emil Precup, Mircea-Bogdan Rădac, Claudia-Adina Dragoş

Department of Automation and Applied Informatics
"Politehnica" University of Timisoara
Bd. V. Parvan 2, RO-300223 Timisoara, Romania
E-mail: stefan.preitl@aut.upt.ro, radu.precup@aut.upt.ro, mircea.radac@aut.upt.ro, claudia.dragos@aut.upt.ro

József K. Tar, János Fodor

Institute of Intelligent Engineering Systems
Budapest Tech Polytechnical Institution
Bécsi út 96/B, H-1034 Budapest, Hungary
E-mail: tar.jozsef@nik.bmf.hu, fodor@bmf.hu

Abstract: The paper proposes the stable design of a new generation of fuzzy control systems using Mamdani PI-fuzzy controllers (PI-FCs) in four control system structures based on the combination beetween feedback control structures and Iterative Learning Control (ILC) algorithms. The design is backed up by a stability analysis method expressed in terms of the specific matrix approach with nonlinearity vectors defined in the matrix space. Part of the theoretical results is validated in controlling servo systems considered as second-order integral type benchmarks.

Keywords: Iterative Learning Control, fuzzy control systems, matrix space, stable design

1 Introduction

Iterative Learning Control (ILC) accepts that control system (CS) performance indices executing repetitively the same tasks can be improved using previous experiments concerning CS operation. The aim of ILC deals with the iterative solving of a parametric optimisation problem, called learning [1], which ensures the minimization of an objective function which specifies CS performance indices. In order to solve this optimisation problem there are implemented ILC algorithms to ensure CS performance enhancement from one experiment / iteration to another. Information acquired from previous experiments is included using adequate memorizing techniques.

The ILC algorithms generate an open-loop signal, which does the approximate inversion of the plant model to guarantee reference tracking and repetitive disturbance rejection. In order to cope with non-anticipative disturbances the ILC algorithms are combined with controllers resulting in several design techniques. They include:

- learning functions of PD-type PD [2-5], which allow controller tuning without requiring the detailed mathematical model of the controlled plant,
- learning functions based on the plant model inversion [6, 7], which guarantee a rapid convergence but are in turn sensitive to modelling errors,
- H_∞ techniques [8, 9], which permit the design of robust and convergent ILC algorithms but having shortcomings in CS dynamic performance,
- quadratic optimisation (Q-ILC) [10-12], based on minimizing integral indices expressed as quadratic objective functions.

Various applications of ILC have been reported in robot control [4, 13-15], machine-tools control [16], electrical and electromechanical drive control [17, 18], autonomous vehicle control [19], ABS control [20], thermal plant control [21, 22], chemical plant control [23], and those specific to servo systems in computing systems [24, 25].

The main advantages of ILC with respect to other control or feedforward approaches are [1, 4, 24]:

- ILC has anticipatory character and can ensure the compensation for repetitive external disturbances by learning (associated with memorization) based on previous iterations,
- ILC does not require knowing the variations of reference and disturbance inputs being necessary just repeating these signals from one iteration to another,
- in some well-stated conditions ILC ensure the CS robustness with respect to process modelling uncertainties.

However, the ILC technique has the following shortcomings [1, 15, 25]:

- the formalization of the connection between robustness and dynamic and steady-state CS performance on the one hand and ensuring the best of these requirements simultaneously on the other hand is not done,
- the situations in which the reference and disturbance inputs do not have repetitive variations are not treated,

- the convergence conditions related generally to any iterative technique are not analyzed in the general framework.

The aim of combining the ILC technique with fuzzy control is to achieve CS performance enhancement in conditions of low-cost [26-28]. The CS performance enhancement results from merging in the same CS structure the benefits of both feedback (due to fuzzy control) and feedforward compensation (due to ILC). This paper presents new fuzzy control system structures based on ILC algorithms in connection with Mamdani PI-fuzzy controllers (PI-FCs). However the systematic design of fuzzy CSs is needed. One way to support these investigations is the stability analysis of the fuzzy control systems with ILC [29]. The stability analysis method is based on the formulation and application of rather general results that employ the specific matrix approach expressed in terms of the nonlinearity vectors applied in the matrix space [30-32]. This approach has been applied in [33] in connection with Iterative Feedback Tuning in the framework of fuzzy control. The combination of that stability analysis method with ILC in the framework of fuzzy control represents an original approach suggested here.

The paper is organized as follows. The next Section is focused on the problem setting in ILC, the fuzzy CS structures incorporating ILC and their design. Section 3 is dedicated to the new stability method that enables the stable design of the method for the Mamdani PI-FCs. Section 5 deals with the presentation of some real-time experimental results for a case study concerning DC-based servo system speed control. The conclusions are presented at the end.

2 Overview on Iterative Learning Control. Fuzzy Control System Structures and Design

The controlled plant is considered characterized by the following discrete-time linear time-invariant SISO system:

$$y_j(k) = P(q)u_j(k) + d(k)\,, \tag{1}$$

where: y – controlled output, u – control signal, d – exogenous input signal (for example, load-type disturbance input) that repeats each iteration, k –index of current sampling interval, j – index of current iteration / trial, q – forward time-shift operator, $P(q)$ – proper rational function of the plant, with a delay of mT_s (having the relative degree of $m \in N^*$), T_s – sampling period. $P(q)$ is supposed to be asymptotically stable. If not, it can be stabilized firstly in a conventional control system, the ILC being applied afterwards to the closed-loop system.

Considering the following sequences of N samples of plant inputs and output and the reference input sequence is $r(k)$:

$$u_j(k), k \in \{0,1,...,N-1\},\ y_j(k), k \in \{m, m+1,..., N+m-1\},$$
$$d(k), k \in \{m, m+1,..., N+m-1\},\ r(k), k \in \{m, m+1,..., N+m-1\}, \quad (2)$$

the control error signal is

$$e_j(k) = r(k) - y_j(k) \,. \quad (3)$$

A widely used ILC algorithm, the Q-ILC algorithm [1, 3, 8, 14]:

$$u_{j+1}(k) = Q(q)[u_j(k) + L(q)e_j(k+1)] \,, \quad (4)$$

makes use of $Q(q)$ referred to as the Q-filter and $L(q)$ as the learning function.

The lifted forms of mathematical models can be used to analyze the CSs based on ILC with the structure resulted from (1) and (4) in the time-domain. The z-domain mathematical models can be expressed, too [28, 29, 34]. The ILC algorithm (4) can be combined with conventional control systems with feedback controllers in two ways at least generating corresponding control system structures:

- a serial form, where the ILC control signal $u_j(k)$ is added to the reference input before the feedback loop,
- a parallel form, where the ILC control signal $u_j(k)$ is added to the control signal produced by the feedback controller.

Other versions of ILC algorithms are:

- the current-iteration ILC algorithm [1]:

$$u_{j+1}(k) = Q(q)[u_j(k) + L(q)e_j(k+1)] + C(q)e_j(k+1) \,, \quad (5)$$

where $C(q)$ stands for the proper rational function of the feedback controller,

- the PD-type learning function in two forms:

$$u_{j+1}(k) = u_j(k) + k_p e_j(k+1) + k_d[e_j(k+1) - e_j(k)] \,, \quad (6)$$

$$u_{j+1}(k) = u_j(k) + k_p e_j(k) + k_d[e_j(k+1) - e_j(k)] \,, \quad (7)$$

where k_p is the proportional gain and k_i is the derivative gain.

The fuzzy CS structures incorporating ILC result directly by inserting fuzzy logic blocks to the linear CS structures. Their low-cost versions are presented in Figs. 1-4. The following nomenclature has been used in all four fuzzy CS structures: ILCA – Iterative Learning Control algorithm, FILCA – Fuzzy Iterative Learning Control algorithm, F – feedforward filter, r_1 – filtered reference input, d_1, d_2, d_3 – load-type disturbance input types, assumed to be repetitive, M – memory block, FC – fuzzy controller, B-FC – basic two input-single output (TISO) fuzzy controller.

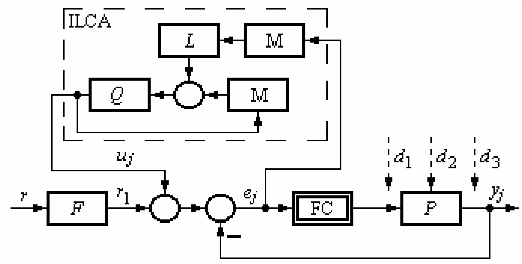


Figure 1
Fuzzy control system structure with serial ILC

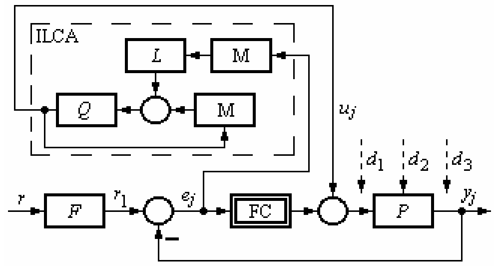


Figure 2
Fuzzy control system structure with parallel ILC

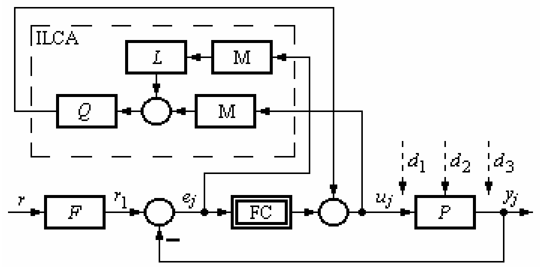


Figure 3
Fuzzy control system structure with current-iteration ILC

The two-level fuzzy CS structure presented in Fig. 4 is based on the fuzzification of the PD block in (6) and (7). The block with the transfer function q^{-1} is necessary only in the fuzzified version corresponding to (7). That is necessary because in conventional ILC algorithms it is difficult to ensure the compromise to both converged error performance and robustness. Ensuring these requirements simultaneously can be achieved by means of the correct tuning of B-FC placed on the higher level in Fig. 4. In fact B-FC is a variable structure controller that ensures the bumpless interpolation between separately designed linear controllers.

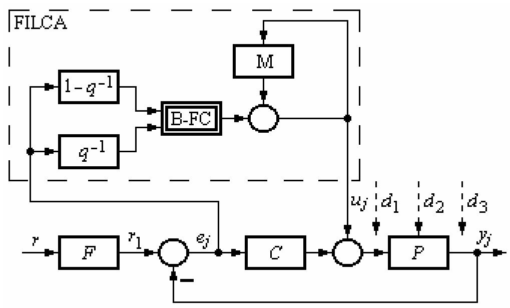


Figure 4

Fuzzy control system structure with PD-type learning function

Other fuzzy control system structures are possible also by the proper combination of the first four ones. The general design method for the fuzzy CS structures in Figs. 1-3 will be presented as follows under a unified expression concentrated on Mamdani PI-FCs with the structure presented in Fig. 5 and membership function shapes shown in Fig. 6. The key element in Fig. 5 is the basic fuzzy controller, B-FC, a TISO nonlinear system that employs Mamdani's MAX-MIN compositional rule of inference and the centre of gravity method for defuzzification.

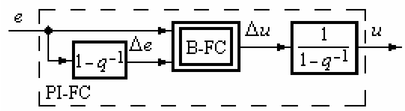


Figure 5

PI-fuzzy control system structure without scaling factors

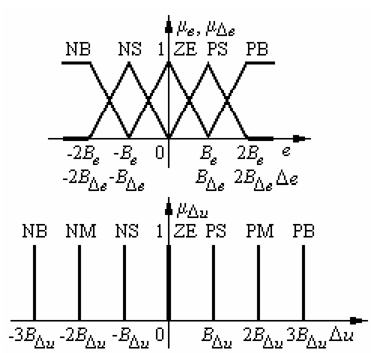


Figure 6

Membership function shapes

The design method consists of the following design steps:

I Steps of the ILCA tuning which are different from one structure to another one.

II Steps of the linear controller design, the initial controller replacing the block FC in the fuzzy control systems structures and representing in fact a two-degree-of-freedom (2-DOF) PI controller:

- tune the feedforward filter $F(s)$ (in continuous-time) and the continuous-time linear PI controller with the transfer function $C(s)$:

$$C(s) = k_c(1+sT_i)/s = k_C[1+1/(sT_i)],\ k_C = T_i k_c, \tag{8}$$

  with k_C – controller gain and T_i – integral time constant, using a continuous-time design method depending on the controlled plant and on the desired / imposed CS performance indices,

- choose the sampling period, T_s, according to the requirements of quasi-continuous digital control,

- express the discrete-time equation of the incremental digital PI controller:

$$\Delta u(k) = K_P \Delta e(k) + K_I e(k) = K_P[(\Delta e(k) + \alpha \cdot e(k)], \tag{9}$$

  with Δx standing generally for the increment of a certain variable, x, and calculate the parameters $\{K_P, K_I, \alpha\}$. For example, the expressions of these parameters are presented in (10) in case of Tustin's method:

$$K_P = k_C[1-T_s/(2T_i)],\ K_I = k_C T_s/T_i,\ \ \alpha = K_I/K_P = 2T_s/(2T_i - T_s). \tag{10}$$

III Steps of the PI-fuzzy controller design based on the transfer of results from the linear case to the fuzzy one in terms of the modal equivalence principle:

- set the value of the controller parameter B_e according to the experience of the control systems designer,

- apply the modal equivalence principle:

$$B_{\Delta e} = \alpha B_e, B_{\Delta u} = K_I B_e. \tag{11}$$

The stability analysis to be presented in Section 4 will offer useful information to setting the value of the free parameter B_e. Thus it is justified to consider the design method presented here as stable design method if it is combined with a stability analysis algorithm expressed from the stability analysis method.

4 Stability Analysis Method

The nonlinear function characteristic to symmetric nonlinearities (specific to FCs) is considered in (12) to fulfil Dirichlet's condition:

$$u = N(e) = \sum_{\lambda=0}^{\infty} a_{\lambda+1} \sin[(\lambda+1)\frac{\pi}{\theta} e] \cdot \tag{12}$$

The matrix form of (12) results if it has a finite number of terms written for $(\lambda+1)$ values of the input e:

$$\begin{aligned} &\mathbf{z} = \boldsymbol{\sigma}\,\mathbf{b}, \mathbf{b} = [b_1 \quad b_2 \quad \ldots \quad b_{\lambda+1}]^T, \\ &\mathbf{z} = [z_1 \quad z_2 \quad \ldots \quad z_{\lambda+1}]^T, \boldsymbol{\sigma} = [\sin[(2i+1)\frac{\pi}{\theta} e_j]_{i,j=\overline{1,\lambda+1}}, \\ &e_j = j \cdot h_e, j = \overline{1, \lambda+1}, \end{aligned} \tag{13}$$

where $h_e > 0$ is a step, **b** is the specific vector of the mean nonlinearity and **z** is referred to as the nonlinearity vector [32]. But **σ** is a regular matrix and the following relationship holds in case of the FCs accepted in Section 3:

$$\begin{aligned} &\mathbf{I}_R \mathbf{b} = \mathbf{n}, \mathbf{I}_R = [2J_1\{(2j-1)\frac{\pi}{\theta} A_i\} / A_i]_{i,j=\overline{1,\lambda+1}}, \\ &\mathbf{n} = [n(A_1) \quad n(A_2) \quad \ldots \quad n(A_{\lambda+1})]^T, \\ &e = A_i \sin(\omega t), i = \overline{1, \lambda+1}, \end{aligned} \tag{14}$$

where **n** is the equivalent gain vector, J_1 is the Bessel function of the first kind and A_j stand for the input (e) magnitudes. The following relationships results from the last two ones:

$$\mathbf{H}\,\mathbf{z} = \mathbf{n}, \mathbf{H} = \mathbf{I}_R \boldsymbol{\sigma}^{-1} \cdot \tag{15}$$

(15) highlights a linear transform with the regular matrix **H** independent of the nonlinearity involved. Therefore it can be viewed as a convenient linearization.

The matrix plane approach starts with the expression (16) of the characteristic equation of the closed-loop FCS supposed to be of n-th order, where the nonlinear part (here the FC is involved) is expressed by the equivalent gains as part of the vector **n**:

$$a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0 = 0 \cdot \tag{16}$$

The quasi-continuous digital control is accepted in relation with (16) and it justifies the continuous-time approach. The coefficients in (16) depend on the parameters of the linear part and the equivalent gains of the nonlinear one. Next the following matrix depending on the frequencies is defined in (17):

$$\boldsymbol{\Omega} = [(\sin(k\pi/2) + \cos(k\pi/2))\omega_r^k]_{r=\overline{1,n}, k=\overline{0,p}}, \tag{17}$$

and the following matrices can be expressed:

$$\mathbf{P}_0 = \begin{bmatrix} a_0(n(A_1)) & 0 & a_2(n(A_1)) & 0 & a_4(n(A_1)) & ... \\ a_0(n(A_2)) & 0 & a_2(n(A_2)) & 0 & a_4(n(A_2)) & ... \\ ... & ... & ... & ... & ... & 0 \end{bmatrix}^T,$$

$$\mathbf{Q}_0 = \begin{bmatrix} 0 & a_1(n(A_1)) & 0 & a_3(n(A_1)) & 0 & ... \\ 0 & a_1(n(A_1)) & 0 & a_3(n(A_2)) & 0 & ... \\ ... & ... & ... & ... & ... & 0 \end{bmatrix}^T, \tag{18}$$

$$\mathbf{P}_\sigma = \begin{bmatrix} \pi_0 & -\pi_1' & \pi_2 & -\pi_3' & \pi_4 & ... & 0 \\ ... & ... & ... & ... & ... & ... & 0 \end{bmatrix}^T,$$

$$\mathbf{Q}_\sigma = \begin{bmatrix} \pi_0' & \pi_1 & \pi_2' & \pi_3 & \pi_4' & ... & 0 \\ ... & ... & ... & ... & ... & ... & 0 \end{bmatrix}^T,$$

where the matrices $\mathbf{P}_0$ and $\mathbf{Q}_0$ correspond to the steady-state regime, $\mathbf{P}_\sigma$ and $\mathbf{Q}_\sigma$ to the transients, π_k and π_k' are polynomials in the variable $\sigma, \sigma < 0,$ from $s = \sigma + i\omega \ (i^2 = -1)$, with the expressions according to [32]. The matrix plane is defined as the matrix **M** is defined having double elements:

$$\mathbf{M} = \begin{bmatrix} c_{11} & c_{12} & ... & c_{1q} \\ d_{11} & d_{12} & ... & d_{1q} \\ c_{21} & c_{22} & ... & c_{2q} \\ d_{21} & d_{22} & ... & d_{2q} \\ ... & ... & ... & ... \\ c_{m1} & c_{m2} & ... & c_{mq} \\ d_{m1} & d_{m2} & ... & d_{mq} \end{bmatrix}, \tag{19}$$

taken from the elements of the matrices $\mathbf{\Omega P}$ and $\mathbf{\Omega Q}$ valid in both steady-state regimes and transients:

$$\mathbf{\Omega P} = [c_{ij}]_{i=\overline{1,m},\, j=\overline{1,q}}, \mathbf{\Omega Q} = [d_{ij}]_{i=\overline{1,m},\, j=\overline{1,q}}. \tag{20}$$

Two step-type curves are also defined:

$$c_{\rho\eta} < c < c_{(\rho+1)\varepsilon},\ d_{\lambda\beta} < d < d_{(\gamma+1)\beta},\ \rho, \gamma = \overline{1, m-1},\ \varepsilon, \beta = \overline{1, q-1}. \tag{21}$$

The intersection of the curves for $c = d = 0$ gives the so-called coincidence points (in the matrix plane) corresponding to the limit cycles with solutions expressed as the two coordinates in the matrix plane, the magnitude A_i and pulsation (frequency) ω of the input signal fed to the nonlinearity.

Concluding, the stability analysis method can be expressed as follows and it can be applied in case of fuzzy IFT-based fuzzy CS structures [33]:

- Step 1. Express the linearized characteristic equation of the fuzzy CS.
- Step 2. Calculate the matrix **M** in (19) and the step-type curves in (21).
- Step 3. Search for the limit cycles by calculating the intersection of the step-type curves. If no limit cycles exist, then the fuzzy CS will be stable. If yes, continue with the step 4.
- Step 4. A limit cycle exists and the fuzzy CS admits a periodic solution sufficiently close to $e = A_0 \sin(\omega_0 t)$. The limit cycle is stable if for a sufficiently small value of σ the coincidence point is placed in the matrix plane at a transient magnitude that is larger than the magnitude A_0 of the limit cycle. Hence the system will be stable. Otherwise the system will be unstable.

5 Real-time Experimental Results

A case study focused on a PI-fuzzy controller design for the class of plants with the transfer function $P(s)$ characterizing simplified mathematical models used in servo systems in the framework of mechatronics and embedded systems:

$$P(s) = k_P /[s(1 + T_\Sigma s)], \tag{22}$$

where k_P is the controlled plant gain and T_Σ is the small time constant or an equivalent time constant as sum of parasitic time constants. One solution to cope with the accepted class of plants is represented by PI control. A simple and efficient way to tune the parameters of the PI controller dedicated to this plant is represented by the Extended Symmetrical Optimum (ESO) method [35], characterized by only one design parameter, β. The choice of β within the domain $1 < \beta < 20$, leads to the modification of the CS performance indices (σ_1 – overshoot, $\hat{t}_r = t_r / T_\Sigma$ – normalized rise time, $\hat{t}_s = t_s / T_\Sigma$ – normalized settling time defined in the unit step modification of r, φ_m – phase margin) according to designer's option and to a compromise to these performance indices using the diagrams presented in Fig. 7 in the situation without feedforward filter. The presence of the feedforward filter with the transfer function $F(s)$ improves the CS performance indices. The PI tuning conditions, specific to the ESO method, are:

$$k_c = 1/(\beta\sqrt{\beta}T_\Sigma^2 k_P),\ T_i = \beta T_\Sigma , \tag{23}$$

and they highlight the presence of just β as design parameter.

The experimental setup is the AMIRA DR300 laboratory DC drive used as benchmark in speed control applications. The DC motor is loaded using a current controlled DC generator, mounted on the same shaft, and the drive has built-in analog current controllers for both DC machines having rated speed equal to 3000 rpm, rated power equal to 30 W, and rated current equal to 2 A. The speed control of the DC motor is digitally implemented using an A/D-D/A converter card. The speed sensors are a tacho generator and an additional incremental rotary encoder mounted at the free drive-shaft. A picture of the experimental setup (without the computer connected to the controlled plant), shot from the Intelligent Control Systems Laboratory of "Politehnica" University of Timisoara, is shown in Fig. 8.

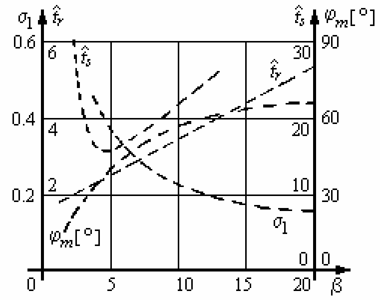
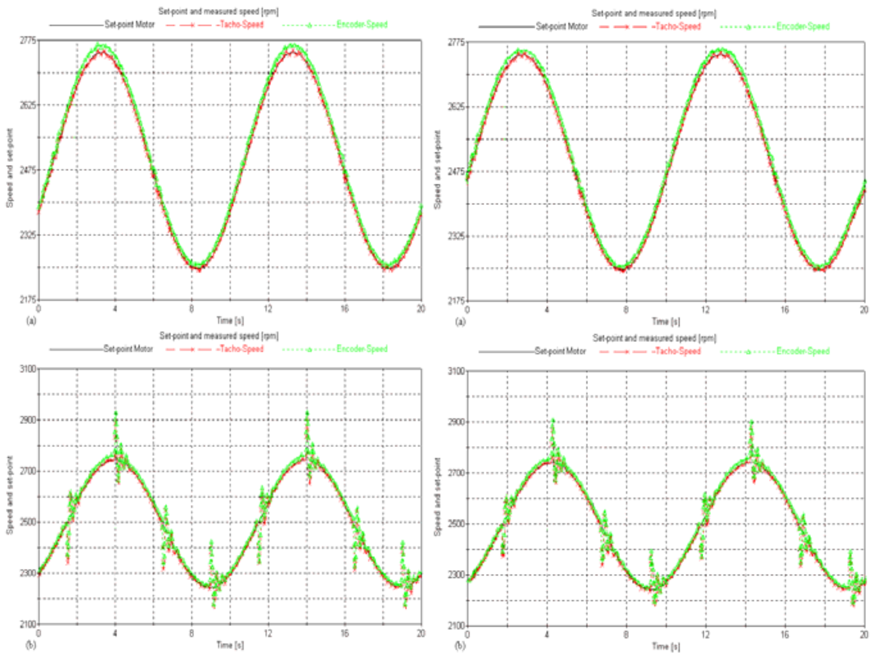


Figure 7
Control system performance indices versus β in the situation without feedforward filter

Figure 8
Experimental setup without computer connected

The mathematical model of the plant can be well approximated by the transfer function $P(s)$ in (20), with k_P = 4900 and T_Σ = 0.035 s. The design method proposed in Section 3 is applied, and for the sake of simplicity only the main parameter values are presented. The method starts with the choice of the design parameter, β = 6. The following values of the PI-fuzzy controller tuning parameters have been obtained: B_e = 0.3 (obtained from the stability analysis method applied as illustrated in Section 4), $B_{\Delta e}$ = 0.03, $B_{\Delta u}$ = 0.0021, and the ILCA employs a Q-filter of 20 Hz bandwidth and a PD-type learning function.

Part of the real-time experimental results, consisting of the variations of r and y versus time, are presented in Fig. 9. The results concern the linear CS (with linear PI controller) in Fig. 9 (a) and the fuzzy CS in Fig. 9 (b), without load in the upper pictures and with a 5 s period of 10% d_2-type rated load and r = 2500 rpm in the lower ones.



(a) (b)

Figure 9

Control system behaviour with PI controller (a) and PI-fuzzy controller (b)

Conclusions

The paper deals with the stable design of fuzzy control system structures combined with Iterative Learning Control. The aim was to achieve the control system performance enhancement for low-cost automation solutions. However although the design and controller structures are simple and transparent, the stability analysis method is not simple. It is computationally demanding.

Real-time experimental results validate one of the fuzzy control system structures and the design method. The validation corresponds to Mamdani PI-fuzzy controllers. The application to other fuzzy controllers is not straightforward.

Future research will be concentrated on deriving simple and transparent design methods for all fuzzy control system structures suggested in this paper accompanied by systematic analyses in all situations. A stability analysis algorithm will be designed and implemented based on the four-step stability analysis method suggested in Section 4.

Acknowledgement

The support stemming from the cooperation between Budapest Tech Polytechnical Institution and "Politehnica" University of Timisoara in the framework of the Hungarian-Romanian Intergovernmental Science & Technology Cooperation Program is acknowledged. The support from the CNCSIS and CNMP of Romania is also acknowledged.

References

[1] D. A. Bristow, M. Tharayil, A. G. Alleyne: A Survey of Iterative Learning Control, in IEEE Control Systems Magazine, 2006, Vol. 26, No. 3, pp. 96-114

[2] S. S. Saab: Stochastic P-type/D-type Iterative Learning Control Algorithms, in International Journal of Control, 2003, Vol. 76, No. 2, pp. 139-148

[3] J.-X. Xu, Y. Tan: Robust Optimal Design and Convergence Properties Analysis of Iterative Learning Control Approaches, in Automatica, 2002, Vol. 38, No. 11, pp. 1867-1880

[4] D. A. Bristow, A. G. Alleyne: A Manufacturing System for Microscale Robotic Deposition, in Proceedings of the American Control Conference, 2003, pp. 2620-2625

[5] H. Elci, R. W. Longman, M. Q. Phan, J.-N. Juang, R. Ugoletti: Simple Learning Control Made Practical by Zero-Phase Filtering: Applications to Robotics, in IEEE Transactions on Circuits and Systems Part I: Fundamental Theory and Applications, 2002, Vol. 49, No. 6, pp. 753-767

[6] K. Kinosita, T. Sogo, N. Adachi: Iterative Learning Control Using Adjoint Systems and Stable Inversion, in Asian Journal of Control, 2002, Vol. 4, No. 1, pp. 60-67

[7] T. Sogo: Stable Inversion for Nonminimum Phase Sampled-Data Systems and its Relation with the Continuous-time Counterpart, in Proceedings of 41st IEEE Conference on Decision and Control, 2002, pp. 3730-3735

[8] K. L. Moore, Y. Chen, H.-S. Ahn: Algebraic H_∞ Design of Higher-order Iterative Learning Controllers, in Proceedings of 2005 IEEE International Symposium on Intelligent Control, 2005, pp. 1213-1218

[9] S. S. Saab: Optimality of First-Order ILC among Higher Order ILC, in IEEE Transactions on Automatic Control, 2006, Vol. 51, No. 8, pp. 1332-1336

[10] D. Gorinevsky: Loop Shaping for Iterative Control of Batch Processes, in IEEE Control Systems Magazine, 2002, Vol. 22, No. 6, pp. 55-65

[11] B. G. Dijkstra, O. H. Bosgra: Convergence Design Considerations of Low Order Q-ILC for Closed Loop Systems, Implemented on a High Precision Wafer Stage, in Proceedings of 41st IEEE Conference on Decision and Control, 2002, pp. 2494-2499

[12] K. Fujimoto, T. Sugie: Iterative Learning Control of Hamiltonian Systems: I/O-based Optimal Control Approach, in IEEE Transactions on Automatic Control, 2003, Vol. 48, No. 10, pp. 1756-1761

[13] M. Norrlof: An Adaptive Iterative Learning Control Algorithm with Experiments on an Industrial Robot, in IEEE Transactions on Robotics and Automation, 2002, Vol. 18, No. 2, pp. 245-251

[14] Y. Q. Chen, K. L. Moore: An Optimal Design of PD-type Iterative Learning Control with Monotonic Convergence, in Proceedings of IEEE International Symposium on Intelligent Control, 2002, pp. 55-60

[15] Y. Ye, D. Wang: Learning more Frequency Components Using P-type ILC with Negative Learning Gain, in IEEE Transactions on Industrial Electronics, 2006, Vol. 53, No. 2, pp. 712-716

[16] M.-S. Tsai, M. T. Lin, H.-T. Yau: Development of Command-based ILC Algorithms with Consideration of Friction, Disturbance, and Noise Effects, in IEEE Trans. on Control Systems Technology, 2006, Vol. 14, No. 3, pp. 511-518

[17] S. S. Saab: A Stochastic ILC Algorithm with Application to an Induction Motor, in International Journal of Control, 2004, Vol. 77, No. 2, pp. 144-163

[18] W. Hoffmann, K. Peterson, A. G. Stefanopoulou: Iterative Learning Control for Soft Landing of Electromechanical Valve Actuator in Camless Engines, in IEEE Transactions on Control Systems Technology, 2003, Vol. 11, No. 2, pp. 174-184

[19] Y. Q. Chen, K. L. Moore: A Practical Iterative Learning Path-following Control of an Omni-directional Vehicle, in Asian Journal of Control, 2002, Vol. 4, No. 1, pp. 90-98

[20] C. Mi, H. Lin, Y. Zhang: ILC of Antilock Braking of Electric and Hybrid Vehicles, in IEEE Transactions on Vehicular Technology, 2005, Vol. 54, No. 2, pp. 486-494

[21] D. R. Yang, K. S. Lee, H. J. Ahn, J. H. Lee: Experimental Application of a Quadratic Optimal Iterative Learning Control Method for Control of Wafer Temperature Uniformity in Rapid Thermal Processing, in IEEE Transactions on Semiconductors Manufacturing, 2003, Vol. 16, No. 1, pp. 36-44

[22] D. Gorinevsky: Loop Shaping for Iterative Control of Batch Processes, in IEEE Control Systems Magazine, 2002, Vol. 22, No. 6, pp. 55-65

[23] M. Mezghani, G. Roux, M. Cabassud, M. V. Le Lann, B. Dahhou, G. Casamatta: Application of ILC to an Exothermic Semibatch Chemical Reactor, in IEEE Transactions on Control Systems Technology, 2002, Vol. 10, No. 6, pp. 822-834

[24] H. Melkote, Z. Wang, R. J. McNab: An Iterative Learning Controller for Reduction of Repetitive Runout in Disk Drives, in IEEE Transactions on Control Systems Technology, 2006, Vol. 14, No. 3, pp. 467-473

[25] M. R. Graham, R. A. De Callafon: An Iterative Learning Design for Repeatable Runout Cancellation in Disk Drives, in IEEE Transactions on Control Systems Technology, 2006, Vol. 14, No. 3, pp. 474-482

[26] C. J. Chien: A Sampled-Data Iterative Learning Control Using Fuzzy Network Design, in International Journal of Control, 2000, Vol. 73, pp. 902-913

[27] J.-X. Xu, J. Xu: A New Fuzzy Logic Learning Control Scheme for Repetitive Trajectory Tracking Problems, in Fuzzy Sets and Systems, 2003, Vol. 133, No. 1, pp. 57-75

[28] R.-E. Precup, S. Preitl, J. K. Tar, M. Takács: Optimization Aspects in a Class of Fuzzy Controlled Servosystems, in Proceedings of 11th International Conference on Intelligent Engineering Systems INES 2007, pp. 235-240

[29] R.-E. Precup, S. Preitl, J. K. Tar, M. L. Tomescu, M. Takács, P. Korondi, P. Baranyi: Fuzzy Control System Performance Enhancement by Iterative Learning Control, in IEEE Transactions on Industrial Electronics, 2008, Vol. 55, No. 9, pp. 3461-3475

[30] D. Teodorescu: Analysis and Synthesis of Nonlinear Control Systems by Means of a Sampled-data Nonlinearity Matrix, in IEE Proceedings, 1971, Vol. 118, No. 11, pp. 1655-1660

[31] D. Teodorescu: Die umgekehrten Beschreibungsreihen - ein neues Mittel zur Synthese nichtelinearer Regelungssysteme, in Regelungstechnik, 1972, Vol. 20, No. 6, pp. 253-261

[32] D. Teodorescu: Entwurf nichtlinearer Regelsysteme mittels Abtastmatrizen, Hüthig Verlag, Heidelberg, 1973

[33] R.-E. Precup, S. Preitl, P. A. Clep, I.-B. Ursache, J. K. Tar, J. Fodor: Stable Fuzzy Control Systems with Iterative Feedback Tuning, in Proceedings of 12th International Conference on Intelligent Engineering Systems INES 2008, 2008, pp. 287-292

[34] M. Norrlof, S. Gunnarsson: Time and Frequency Domain Convergence Properties in Iterative Learning Control, in International Journal of Control, 2002, Vol. 75, No. 14, pp. 1114-1126

[35] S. Preitl, R.-E. Precup: An Extension of Tuning Relations after Symmetrical Optimum Method for PI and PID Controllers, in Automatica, 1999, Vol. 35, pp. 1731-1736