Iterative Learning Fuzzy Control

M. Olivares
Dept. of Electronic Eng.
Univ. Tecnica F. Santa Maria
P.O. Box 110-V, Valparaiso, Chile

P. Albertos **A. Sala**
Dept. of Systems Eng. and Control
Univ. Politecnica de Valencia
P.O. Box 22012, E-46071 Valencia, Spain

Abstract

In this paper an iterative learning control design method is depicted, leading to a feedforward controller minimizing tracking error of repetitive trajectories. The approach is extended to the case of a fuzzy controller, where the plant inverse is approximated by a fuzzy system. This provides some extra features, being also suitable to be applied to nonlinear plants. A simple application illustrates the method.

Keywords: Iterative Learning Control, Fuzzy Control, Feedforward Control, Learning Systems

1 Introduction

The universal approximation property of fuzzy systems is being widely used in many areas, in particular nonlinear modeling and complex systems control. In complex systems control, the ones with high dimensionality and/or high nonlinear behavior, fuzzy systems are designed to capture the human operator cumulative knowledge.

Iterative learning control is a well-established methodology, [2], to derive the sequence of control actions to be applied to a process working under repetitive conditions, either in tracking a reference or in rejecting a deterministic disturbance. This approach has been well used in the robotic area, when a manipulator has to repeat a given trajectory many times. One of the disadvantages of the approach is the dependence on the learned trajectory, as it results in an open-loop control with pre-computed actions.

The idea of this paper is to combine fuzzy systems and the iterative learning approach, in such a way that the result of the learning process is no more a sequence of actions but the implicit knowledge in the kernel of a fuzzy system. Thus, under some circumstances, the controller will also work for different inputs or plants. Some preliminary results are presented with a simple example.

2 Fundamentals of Iterative Learning

2.1 Inverse model control

Inverse model control techniques, also called cancellation control, rely on the inversion of the dynamic behavior of the plant to be controlled. Of course, this requires a precise model of the plant as well as some causality constraints.

Consider the pulse response of a sampled linear time invariant system

$$\{h_k\} = \{0, \dots, 0, h_m, h_{m+1}, \dots\} \qquad (1)$$

where m is the relative degree of the system (if $m > 1$ that means that a pure delay is present). The system output is (convolution formula)

$$y(i) = \sum_{j=0}^{i-m} h_{i-j} u(j), \qquad (2)$$

If a finite length response is analysed ($i = m, \cdots, N$), then (2) can be written in matrix form. By defining vectors $u, y \in R^{N-m+1}$

$$\begin{aligned} u &= \begin{bmatrix} u(0) & u(1) & \cdots & u(N-m) \end{bmatrix}^T \\ y &= \begin{bmatrix} y(m) & y(m+1) & \cdots & y(N) \end{bmatrix}^T \end{aligned} \tag{3}$$

$$y = Hu \tag{4}$$

$$H = \begin{bmatrix} h_m & 0 & \cdots & 0 \\ h_{m+1} & h_m & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ h_N & h_{N-1} & \cdots & h_m \end{bmatrix} \tag{5}$$

In that way, the inverse system can be also represented in finite time by a matrix relation $u = My$, where $M = H^{-1}$, and another convolution:

$$u(i-m) = \sum_{j=m}^{i} a_j y(m+i-j) \tag{6}$$

so from any desired finite time output trajectory sequence $y = y_d$, the input sequence u can be derived given the feedforward matrix M. If a model is not available, the feedforward matrix elements a_i $(i = m, \cdots, N)$ can be easily estimated from any previously recorded experimental input-output pairs (u, y) and an estimation of the total delay m, but the approach is very sensitive to model uncertainty, noise and disturbances. Also, it does not apply to nonlinear systems.

Alternatively, there are some techniques from artificial intelligence that can be used for obtaining the inverse of the system in the direction of the desired trajectory y_d. One of them is the fuzzy iterative learning approach to be presented.

2.2 Iterative Learning

The iterative learning methodology can be understood as follows. Given a repetitive system

$$y_k(t) = f_H(u_k(t), t) \quad t \in [t_0, t_f] \tag{7}$$

where $f_H : U \to Y$ is a nonlinear operator, and $u_k(t)$ is the input sequence applied to the system in the repetition (iteration) k at time t, a learning operator f_L will provide the input $u_{k+1}(t)$ to be applied to the plant in the next iteration, taking into account the current $(u_k(t), y_k(t))$ input-output data pair and the desired $y_d(t)$ output

$$u_{k+1}(t) = f_L(u_k(t), f_H(u_k(t), t), y_d(t), t) \tag{8}$$

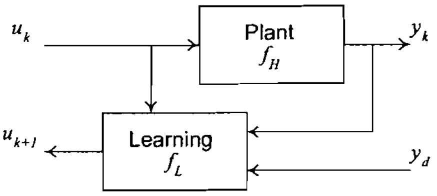

Figure 1: Iterative Learning

During all iterations $y_d(t)$ is kept the same.

Thus, the iterative learning problem is to design an iterative algorithm (8) that converges to an input $u^*(t)$ that minimizes the system output tracking error $y_d(t) - f_H(u^*(t), t)$ in some norm $\forall t \in [t_0, t_f]$ [2].

Instead of directly learn the inputs to the plant, some parameters of a feedforward control policy given by a fuzzy system, as the purpose of the learning process, is proposed. This approach has advantages regarding the number of adjustable parameters in some cases. With respect to adaptation of the feedback block, the reader is referred to other references such as [3]. For brevity, the closed loop controller adaptation will be no further discussed in this paper.

3 The iterative learning controller

The simplest linear iterative learning algorithm modifies the plant input in such a way to correct the current $e_k(t)$ output error, on each sampling time. The iterative learning approach can cope with model uncertainty, as the input sequence is updated at each iteration with actual output system data, and a closed loop controller can manage disturbances and noise, and stabilize the plant if it is unstable.

The relative degree m has to be considered in the learning algorithm (8), so that the input is changed accordingly to the system's causality, proportionally to $p_k(t)$, as follows

$$\begin{aligned} u_{k+1}(t-\bar{m}) &= u_k(t-\bar{m}) + \alpha p_k(t) \\ p_k(t) &= y_d(t) - y_k(t) \end{aligned} \tag{9}$$

where $\bar{m} \geq m$ is a learning delay that assures a significant effect of the input: h_m can be nonzero but small or negative with respect to the final DC gain (nonminimum phase). In that case, a different $\bar{m}$ is chosen. For simplicity, in the following it will be assumed $\bar{m} = m$. For a linear process, let us define as $u^*(t-m)$ the ideal inverting control action sequence obtained from applying (6) to the sequence $y_d(t)$ so zero tracking error is obtained.

Theorem 3.1 (Linear algorithm convergence) *For the system* (4), *with relative degree* m *and sampled state space* $A, B, C, (D = 0)$ *representation, the first order linear iterative learning algorithm* (9), *applied from* $t = m$ *to* $t = N$ *converges to* u^* *if and only if the learning rate satisfies* $0 < \alpha < 2(CA^{m-1}B)^{-1}$ *The maximum convergence speed is obtained for* $\alpha = (CA^{m-1}B)^{-1}$.

Proof: The learning law (9) may be rewritten using (6) for $t = m, \ldots, N$, yielding

$$\sum_{j=m}^{t} a_j y_{k+1}(m+t-j) = \sum_{j=m}^{t} a_j y_k(m+t-j) + \alpha e_k(t)$$

Subtracting from both sides the ideal input $\sum_{j=m}^{t} a_j y_d(m+t-j)$, the tracking error follows:

$$\sum_{j=m}^{t} a_j e_{k+1}(m+t-j) = \sum_{j=m}^{t} a_j e_k(m+t-j) - \alpha e_k(t)$$

Now, if the $\mathcal{Z}$-transform is applied, taking k as the sequence index, the following relationship among the temporal tracking errors is obtained:

$$e_z(t) = \frac{-(z-1)}{(z-1)a_m + \alpha} \sum_{j=m+1}^{t} a_j e_z(m+t-j)$$

As $e_z(t)$ depends only from *past* errors, the preceeding set of equations (t ranges from m to N) has a triangular structure. From the stability condition (pole inside unit circle) and $a_m = h_m^{-1} = (CA^{m-1}B)^{-1}$ the result is derived. With $\alpha = h_m^{-1}$ a minimum time algorithm is obtained.

■

For proofs of convergence of other iterative learning algorithms in nonlinear plants the reader is referred to [1] (without a fuzzy framework).

This result can be extended to iteratively learn the consequents of a **parameterized feedforward fuzzy controller**. The proposed structure is depicted in figure 2.

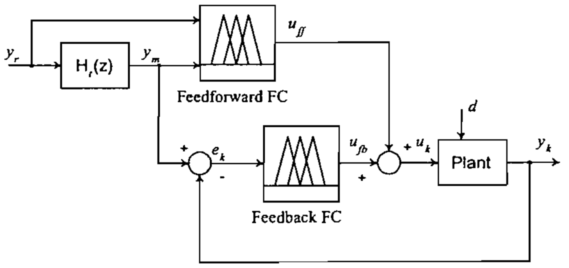

Figure 2: Two degrees of freedom structure

In this paper, an add-1 partition fuzzy controller will be used, with rules such as:

$$r_{\sigma\tau} : \text{if } x \text{ is } G^{\sigma\tau}(x) \text{ then } y \text{ is } h^{\sigma\tau}$$

where $G^{\sigma\tau}(x)$ represents the input fuzzy terms and $h^{\sigma\tau}$ is a scalar consequent. The output will be obtained by an averaging defuzzifier

$$y = \sum_{\sigma=1}^{n_1} \sum_{\tau=1}^{n_2} \mu_\sigma(x_1)\mu_\tau(x_2)h^{\sigma\tau} \qquad (10)$$

Although a general learning structure should have as inputs to the feedforward block a complete reference sequence and time,its dimensionality grows too high. In our case, the inputs are a *constant* reference and the output of a monotonically increasing non oscillatory reference model $H_r(z)$.

The first input specializes the learning to constant references and reduces the number of parameters. The replacement of time by the reference model output has the equivalent effect of fuzzy granularity reduction for high t, where $u(t) = y_r/D$, being D the system's DC gain, so precision requirements are less stringent once the system is near steady state (errors there can be easily compensated with integral feedback). This is a way to tackle iterative learning with long experiment durations. In many cases, the fuzzy system will have a minimum approximation error so that the sequence $u^*(t)$ will not be able to be exactly replicated. In this case, the error will not converge to zero, but it can be made as small as desired (and even smaller due to the additional feedback).

Regarding the fuzzy controller, a consequent update of the rule $r_{\sigma\tau}$ that takes into account distribution of p_k among the fired rules, is considered

$$\Delta h_k^{\sigma\tau} = \alpha \sum_{t=m}^{N} \mu_\sigma(y_r)\mu_\tau(y_m(t-m))p_k(t) \quad (11)$$

Depending on the configuration, the learning rate α may need to be reduced, even in the linear case.

The scheme works equally for nonlinear plants. In that case, the training with different (constant) reference values y_r would produce a non-linear system inversor. In linear systems, straightforward scaling would solve the problem.

The number of parameters in the scheme is not exponentially dependent on the plant or reference model orders. The loss of generality of this approach allows for less number of parameters than a generic option such as training a fuzzy system to mimic a complex plant's inverse with experimental data inverting the role of inputs and outputs. The reduced number of parameters allows for easier convergence with smaller amounts of data.

4 Example

Consider a continuous linear system with delay, and a reference model

$$H(s) = \frac{3e^{-s}}{(s+3)(s+4)} \quad H_r(s) = \frac{1}{(s+1)}$$

discretized with 1 s sampling time. A feedforward FC is designed by iterative learning, using $n_1 = n_2 = 7$ triangular fuzzy sets per input equally distributed over the universes $U_1 = U_2 = [-1.5, 1.5]$.

The learning rate is chosen as $\alpha = 0.75$, 6 times lower than the minimum-time gain from the theorem. After $k = 50$ iterations of duration $t_f - t_0 = 20\,s$ with a step reference, a rule base is obtained achieving a maximum tracking error $\|y_d(t) - y_k(t)\| = 0.0254$.

To test the results in a 2DF framework, a feedback adaptive sliding-mode fuzzy controller is also put into operation [4]. The response with reference changes and step disturbances is shown in fig. 3.

Conclusions

In this paper a fuzzy feedforward iterative learning control for repetitive situations has been presented, into a 2 degree of freedom framework.

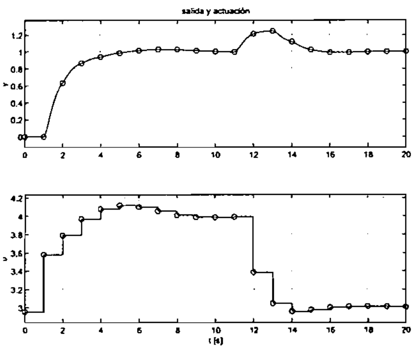

Figure 3: Iterative learning results

The convergence has been proved for the linear case, but it works well for some nonlinear plants. Its convergence depends on adjusting the learning rate based on the assumption that a bound on the instantaneous gain $|y(t+m)|/|u(t)|$ exists.

The use of a fuzzy approximator allows the system to calculate the inverse for several references in nonlinear systems, and also allows to reduce the number of parameters on the time axis given that precision requirements change with time. The proposed approach may be better in practice than other brute-force inversion techniques because parameterizations are independent of system and model orders.

References

[1] Y. Chen and C. Wen. *Iterative Learning Control: Convergence, Robustness and Applications*, volume 248 of *Lecture Notes in Control and Information Sciences.* Springer-Verlag, London, 1999.

[2] K.L. Moore. *Iterative Learning Control for Deterministic Systems.* Advances in Industrial Control. Springer-Verlag, Berlin, 1993.

[3] R. Ordoñez, J. Zumberge, J.T. Spooner, and K.M. Passino. Adaptive fuzzy control: Experiments and comparative analyses. *IEEE Trans. Fuzzy Systems*, 5(2):167–188, May 1997.

[4] A. Sala, P. Albertos, and M. Olivares. Variable structure learning controllers. In *Workshop on Computer Aided Systems and Technology ECAST'99*, pages 543–549, Vienna, Austria, 1999.