IMPLEMENTATION OF A GENETICALLY OPTIMIZED INTERPOLATIVE FUZZY INFERENCE ENGINE IN CONTROLLING A BALL-PLATE LABORATORY SETUP

Caro Lucas  
Center of Excellence for Control and Intelligent Processing, Dept. of Electrical and Computer Eng., University of Tehran, and School of Intelligent Systems, IPM, Tehran, Iran.  
Tel: (+98-21-228-7013), P.O.Box:19395-5746, Email: lucas@ipm.ir

Alireza Fatehi  
Dept. of Electrical and Computer Eng., K. N. Toosi University of Technology, Tehran, Iran.  
Email: fatehi@eetd.kntu.ac.ir

Danial Shahmirzadi  
Center of Excellence for Control and Intelligent Processing, Dept. of Mech. Eng., University of Tehran, and School of Intelligent Systems, IPM, Tehran, Iran.  
Email: Danial_shahmirzadi@yahoo.com

Mohsen M. Takami  
Dept. of Electrical and Computer Eng., K. N. Toosi University of Technology, Tehran, Iran.  
Email: m2takami@yahoo.com

ABSTRACT  This paper goes through giving the results of implementing a genetically optimized interpolative fuzzy engine in controlling a ball-plate laboratory setup. For demonstrating the advantages of our proposed interpolative fuzzy controller over classical fuzzy controllers, we gave some comparison between the fuzzy controllers, one with common CRI inference mechanism and one with our interpolative inference mechanism. As expected, our fuzzy interpolative controller is more efficient than CRI-based controller, with respect to computational space, as well as it is more robust to goal achievement.

KEYWORDS- Fuzzy Inference, Interpolation, Interpolative Reasoning, Ball-Plate System, Robustness,

INTRODUCTION  
Easy interpretability and robust performance are often cited as some of the main advantages of fuzzy control systems [1, 2]. It has usually been argued that these properties, as well as the smoothness of the system response is due to the inherent nature of fuzzy logic based systems where graduated membership replaces abrupt transitions derived from “All or None” attitudes. In general, a fuzzy logic controller has an architecture shown in Fig. 1. The Knowledge is often represented via a set of production rules of the form IF…, THEN…. The implementation of rule based expert systems in real-world applications usually involves a knowledge engineering phase, when the different task of rule induction, has to be carried out. Fuzzy systems can assist in reducing the number of required rules, since there is no need to have a separate rule for every possible environmental condition. Approximate reasoning techniques can infer the appropriate action, even though the observed condition is not identical to any of the rule antecedent. However, a main weakness in fuzzy system applications is the problem of rule base generation. In recent years, several methods for optimum rule base generation, have been proposed using NN, GA and other methodologies [11,12,13,14,15,17,18,20]. But few studies have been conducted to compare the different inference methods with respect to the number of rules required for acceptable functioning, or to design a new inference method that could continue functioning with smaller rule-base. The main concern in this paper is the inference engine. Despite the fact that the initial motivation of fuzzy controllers was logical in nature, the use of mamdani’s interpolation rule or similar inference procedures instead of compositional rule of inference (CRI) and a number of other practices mainly motivated by neuro-fuzzy implementation of the procedures, gradually shifted the main emphasis in inferencing procedure from deduction to analogy. It has been subsequently argued that fuzzy controllers and fuzzy inference procedures in general, carry out
interpolative reasoning [3, 4, 16]. This is not necessarily a disadvantage. In many respects, it even involves important advantages. A fuzzy rule based system need not have too many rules in order to be able to function in various conditions. When the observed state of feedback, based upon which decision is to be made, is different from the antecedents of the rules in the rule base, fuzzy systems can still infer the proper action since they carry out approximate and interpolative, rather than strictly logical and crisp reasoning. Different methods for approximate reasoning have varying capacities for inferencing when the set of rules is sparse [4]. For example, it has been shown that the use of an approximate analytical reasoning method enables a fuzzy load forecasting system to arrive at better predictions using fewer production rules when compared to alternative methods for inferencing [5]. Our motivation is that through using generalized curve fitting techniques that are interpolative in nature, one can construct inference engine capable of conducting approximate reasoning in the presence of sparse rule base. Not only does the rule base that has to be identified thus become smaller so we can simplify the difficult task of rule generation [6], but also the computational burden of the inferencing also decrease significantly [7,8]. To try this assumption, we first discuss the extension of a new interpolation method originally has been given in [16] and fuzzy inferencing based on the mentioned ideas. Next, we examine the implementation of above controller in a ball-plate laboratory setup in order to confirm its convenient operation. Also, we have made an experiment to consider the controllers’ robustness in goal achievement.

BALL-PLATE SYSTEM
The experimental system used in this study is a ball and plate system manufactured by Humusoft Co. which has been shown in Fig. 2. The plate of the system is capable of rotating over its central point that these rotations are supported by two step-motors, one is responsible for rotation around the X axe and the other is for the Y axe. These motors can be fed by signals produced via a digital controller, based on the ball location observations. The location is sensed by a camera positioned above the plate. The digital controller can be implemented by MATLAB® software. Assuming $\alpha$ and $\beta$ as the amount of rotation around the X and Y axes, the dynamical equations governing the system, after simplifications, could be given as:

$$\dot{X} = \frac{5}{7} g \sin \alpha = K_x \alpha$$

(1)

$$\dot{Y} = \frac{5}{7} g \sin \beta = K_y \beta$$

(2)

RULE BASE OPTIMIZATION WITH GENETIC ALGORITHM
The contribution of this paper concerns replacing the common inference procedures, e.g. CRI, with proposed interpolative algorithm. Therefore, the phases preceding the determination of the rule base can be done the same as other methods. The rule base designed for utilization in the fuzzy inference engines are established by a genetic algorithm in order to not only optimize the membership functions, but also the number of rules, so there may not be any rule for some input-output relations (Table 1). However, other methods in which only the specifications of the membership functions are to be optimized, have also been put forward [20]. In the proposed method, obviously, some rules are obtained for specific input-output conditions; however, with interpretation of combinatorial memberships function with MAX fusion operation, this means that, generally, the number of rules has been also minimized. Implementing the GA algorithm can be performed by assuming each rule base as a chromosome, examining it in the fuzzy controller with the linearized ball-plate model and run it for all different initial-final conditions of the system, then calculating the weighted sum of the obtained errors as given in Eq. (3):

$$f(RB) = \frac{1}{\sum_i \alpha_i \times I_i}$$

(3)

in which, $I_j$ is the genetic fitness function of the jth rule base of the fuzzy controller for ith initial-final state value condition. The values of fitness function that must be maximized can be calculated via following relation:

$$I = \int_0^8 [(\dot{\alpha}^2 + \dot{\beta}^2)dt$$

(4)
Where \( e \) and \( u \) are the errors and control signals, respectively. Ultimately, the optimized rule base for the utilized fuzzy controller is given in table 1.

**INTERPOLATION METHOD**

In this paper, we extended our interpolation method developed in [16], by increasing the dimensions of the interpolated function so as to enable it to operate as a fuzzy controller with two inputs (usually error and its variation rate), and by adding another level of data importance evaluation. The method is mainly based on the minimization of the squared error in a way by which we could assume the fuzzy importance of each datum. For clearer illustration of a rule base that is to be minimized, in Fig. 3, we furnish a rule base with 3 rules. The algorithm assumed in this study is given below:

Minimize \[ \lambda : \sum_{j} \left( \sum_{e,\hat{e},y} \left( y - f(e,\hat{e},\lambda) \right)^2 k_j \mu_j \left( e,\hat{e},y \right) \right) \]

where, \( \lambda \) is the set of parameters associated with the interpolation function and \( \mu_j \left( e,\hat{e},y \right) = \mu_{\lambda_j} \left( e,\hat{e},y \right) \)

The coefficient \( k_j \) has been assumed for importance allocation among the rules, so one can give different worth values to any of the rules. We note that the denominator of the Eq. (5) has a constant value. So due to continuity of our membership functions, the Eq. (5) gets the form of:

Minimize \[ \lambda : \sum_{j} \int_{e,\hat{e},y \in V_j} \left( y - f(e,\hat{e},\lambda) \right)^2 k_j \mu_j \left( e,\hat{e},y \right) ded\hat{e}dy \]

In which, \( V_j \) is the space corresponds to jth rule. The burden of the controller designer is to cope with appropriately selecting the function \( f \) and the coefficients \( k_j \) for each rule. The computation of \( \lambda \) can be carried out via some iterative optimization routine e.g.:

\[ \lambda_{i+1} \leftarrow \lambda_i + \kappa \nabla_{\lambda} f(e,\hat{e},\lambda) \sum_{j} \int_{V_j} \left( y - f(e,\hat{e},\lambda) \right) k_j \mu_j \left( e,\hat{e},y \right) ded\hat{e}dy \]

or whenever possible, in closed form by solving the equation obtained through differentiation of Eq. (7). In this paper, we assumed the interpolation function as given in the formula (9):

\[ f(e,\hat{e},\lambda) = G(e + \hat{e}) \]

where the vector of interpolation parameters, \( \lambda \), in the above function, becomes only a scalar parameter, \( G \). However, many other polynomials and nonpolynomials functions, with different orders, have also been examined but didn’t yield results as good as the one obtained through Eq. (8). It is important to note that the computation of the interpolating function can be carried out off line, so that the computational load of the controller itself is very light. Therefore, our proposed inference engine can be implemented through various hardwares or firmwares (e.g. microcontrollers or DSPs) in real time very easily. The coefficients \( k_j \)'s for the above 25 rules are given in the following array:

\[ K_j = [0.0005, 0.003, 0.0005, 0.001, 0.1, 0.0005, 0.0005, 0.0005, 0.0005, 0.0005, 0.001, 0.0012, 0.12, 0.0005, 0.0005, 0.004, 1, 0.0008, 0.0018, 0.11, 0.0006, 0.0018, 0.0022, 0.0045, 0.0045] \]

Through the above factors for the 25 rules, we may recognize that the importances of 4 rules are in a level much higher than the rest 21 rules. In the next section, we also examine a fuzzy-CRI controller with these only 4 rules in hope that it would demonstrate a satisfactory response. After passing through the mentioned procedures, the gain of the equation 9 resulted in the value of \( G = 0.5039 \).
SIMULATION RESULTS
In this section, we describe the specifications of a linearized model of the ball-plate system. Generally, the above system is decoupled in X and Y directions. Thus, we set our control strategy, by the way that two identical controllers are to control the position of the ball in X and Y directions. Therefore, our simulation’s control circuit consists of one DOF plant with one controller. Again, we mention that the control system and the results obtained from it, can be interpreted exactly for behavior of the system in either X or Y direction. The transfer function of the ball-plate system we utilized in the simulations is given in Eq. (10):

\[
G(s) = \frac{25.68}{s^2(s+5)}
\]  
(10)

The responses of the system’s one-directional position, with the Fuzzy-CRI controller developed in [19], and the Fuzzy-interpolative controller, are given in Fig.s 4 and 5, respectively. As observed, both responses are good enough, however, the response of the fuzzy-interpolative controller is faster and smoother. The responses of the system via fuzzy-CRI and fuzzy-interpolative inference methods with a rule base of only 4 much more important rules (rules No. 5,13,17,20), are shown in Fig. 5 that confirms the fact that a very big burden of the inferencing task is in the shoulders of these 4 specified rules and the other 21 rules play a small rule to achieve the favorable response (reducing the overshoot and steady-state error), although, the interpolative mechanism still can infer a satisfactory response. In the next section, we are going to implement these controllers for the concrete laboratory system.

IMPLEMENTATION RESULTS
Eventually, we must evaluate the functionality of the above controllers in their experimental implementation in laboratory setup. The specifications of the utilized ball-plate laboratory setup have already been given. It should be mentioned that the ball we used in our experiments was a Ping-Pong (Table-Tennis) ball in white color. Following are the responses of the experimental ball-plate setup with the Fuzzy-CRI and Fuzzy-Interpolative controllers. The left and right columns of the Fig.s 6 correspond to the results of Fuzzy-Interpolative and Fuzzy-CRI controllers, respectively. Plots (a) of the Fig. 6 are the X position versus Y position of the ball. In plots (b), each X and Y positions are drawn in time duration. Plots (c) and (d), represent the error of X (and Y), the error of its derivative and the control effort in X (and Y) direction. The smoother behaviors of the Fuzzy-Interpolative controller in comparison with Fuzzy-CRI controller, as well as the more direct path to the desired objective point in the former controller, are very striking in Fig. 6. For considering the robustness of the two utilized controllers, in another experiment, we tried to make the system’s dynamics, deviate from its origin. To do that, we put a fan near the system to provide a relatively uniform wind, preventing the ball to reach its goal, in the normal conditions. The same results of the Fig.s 6, for these disturbed conditions, are given in the Fig.s 7. Finally, the X-Y plot and time histories of X and Y of the system with a PID controller, tuned by the manufacturer of the above system, are presented in the Fig.s 8(a) and 8(b), respectively, where in the figures, we can see the high overshoots in reaching the desired position of the ball.

CONCLUDING REMARKS
In this paper, we showed that an effective fuzzy interpolative engine developed previously by authors [16], could be extended to be utilized in the control strategies with more dimensions. However, we enriched the extended interpolative controller with a second layer of importance assessment, in its interpolation configuration. Thus, our proposed controller performs an interpolation through the fuzzy data, with regard to the both fuzziness of each datum and the rule to which the datum belongs. The proposed algorithm has been utilized to improve the performance of a fuzzy-genetic ball-plate control system. The responses are very satisfactory relative to the corresponding fuzzy-genetic ball-plate control system. The responses are very satisfactory relative to the corresponding fuzzy-CRI controller [19]. So we were motivated to implement it on the laboratory setup of a ball-plate system. Results of this implementation were in favor of our efficient and excellent fuzzy-interpolative controller. Some performances of the experimental setup were also considered in the case of deviated system dynamics, where the fuzzy-interpolative controller had a less-oscillated path relative to that of the correspondent fuzzy-CRI controller.

REFERENCES
FIGURES

Figure 1- Schematic structure of a fuzzy logic controller (FLC)

Figure 2- Photograph of the utilized Ball-Plate setup

Figure 3- Rule base representation in 3-D space
Figure 4 - Position of the Ball via Fuzzy-Interpolative and Fuzzy-CRI Controllers

Figure 5 - Position of the Ball via INCOMPLETE Fuzzy-Interpolative and Fuzzy-CRI Controllers
Figure 6: L(a), L(b), L(c), L(d), R(a), R(b), R(c), R(d)  (L: Left, R: Right)
Results of the experimental ball-plate system (left and right columns correspond to Fuzzy-interpolative and Fuzzy-CRI controllers).
Figure 7: L(a), L(b), L(c), L(d), R(a), R(b), R(c), R(d) (L: Left, R: Right)
Results of the experimental ball-plate system in the disturbed conditions (left and right columns correspond to Fuzzy-interpolative and Fuzzy-CRI controllers).
Figure 8—Results of the system with PID controller

### TABLES

Table 1—Genetically optimized fuzzy rule base (Each rule number that have been assigned in the rule base, is given in a parenthesis in front of it)

<table>
<thead>
<tr>
<th>Error</th>
<th>Nlarge</th>
<th>Nsmall</th>
<th>Ezero</th>
<th>Psmall</th>
<th>Plarge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nlarge</td>
<td>Phuge(7)</td>
<td></td>
<td></td>
<td>Psmall(10)</td>
<td>Phuge(9)</td>
</tr>
<tr>
<td>Nsmall</td>
<td>Plarge(3)</td>
<td>Nhuge(2)</td>
<td>Nhuge(5)</td>
<td>Nlarge(24)</td>
<td>Nlarge(25)</td>
</tr>
<tr>
<td>Ezero</td>
<td>Nhuge(20)</td>
<td>Nsmall(4)</td>
<td>Zhuge(6)</td>
<td>Phuge(17)</td>
<td>Psmall(21)</td>
</tr>
<tr>
<td></td>
<td>Nlarge(14)</td>
<td>Nsmall(22)</td>
<td></td>
<td></td>
<td>Plarge(19)</td>
</tr>
<tr>
<td></td>
<td>Nlarge(16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Psmall</td>
<td>Zhuge(12)</td>
<td>Zhuge(18)</td>
<td>Plarge(23)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plarge</td>
<td></td>
<td>Nsmall(1)</td>
<td></td>
<td>Nhuge(15)</td>
<td></td>
</tr>
</tbody>
</table>