

Imitation of State Feedback Controllers by Fuzzy Linguistic Control Rules

Yung-Yaw Chen
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.

Jyh-Shing Jang
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720

Abstract

There has been much dispute on the role of fuzzy logic in the field of automatic feedback control. Modern control researchers have not yet been fully convinced of the functionality of fuzzy control because its structure and stability has not been rigidly defined and proved. In this paper, we imitate the performance of a state feedback controller by using linguistic control rules, which is commonly used in fuzzy control. We argue that a fuzzy logic controller should be able to achieve a performance at least as good as a state feedback controller's when properly designed. A simulation-based comparison between the state feedback controller and the proposed fuzzy logic controller is also performed.

Introduction

It is well known that conventional control theories, established on rigorous mathematical modeling and analysis, have failed to cope with the complexity, nonlinearity and imprecision of many realistic control systems and/or industrial processes. On the other hand, human expert can often perform superbly well under conditions of imprecise sensor information and system uncertainty. To take the advantage of this phenomenon, Mamdani and Assilian [1] pioneered the implementation of a fuzzy logic controller, which was based on fuzzy set theory [2,3], to model the decision-making processes of human experts.

Many applications of fuzzy logic control (fuzzy control) have appeared in a wide range of fields since Mamdani's first steam engine controller. Some recent applications include aircraft flight [4], vehicle navigation [5], robot arm [6], nuclear plant [7], automobile transmission [8], Sendai automatic train [9], automobile transmission control [10], etc. In spite of these successful applications, there is still much controversy on the role of fuzzy logic control in the field of automatic control. Modern control theorist are not fully convinced of the validity of fuzzy logic control because its structure is not rigidly defined and its stability is not completely investigated. In this paper, we exhibit the flexibility and versatility of fuzzy control by exploring the possibility of using linguistic control rules to imitate the most elementary control action: a linear function of state variables. We show that the behavior of a state feedback controller (SFC) can be achieved by a fuzzy logic controller (FLC) with properly designed linguistic control rules.

The organization of the paper is as follows. In the next section, we propose an algorithm to encode a set of linguistic control rules based on the given state feedback control actions. In section 3, we discuss two properties of the proposed algorithm. Then, in section 4, we apply our algorithm to an inverted pendulum system to compare its performance with the proposed fuzzy logic controller. Finally, we conclude that, given a set of appropriate rules, a fuzzy logic controller can achieve a performance at least as good as a state feedback controller's. Some potential future work is also explored in the last section.

Imitate State Feedback Control by Fuzzy Implication Rules

In this section, we are going to introduce an algorithm that can imitate a given state feedback controller, which is characterized by its control actions, by some linguistic control rules with a certain bounded average error. Let us assume that the control actions of a state feedback controller can be denoted by

$$u = f_{SF}(\vec{x}) \quad (2.1)$$

where u is the controller output and \vec{x} is the vector of state variables. The state feedback plane (SF plane) used in this paper represents the surface of u in (2.1) with \vec{x} as independent variables. Similarly, the fuzzy control surface (FC surface) is defined to be the surface of u in (2.2) with \vec{x} as independent variables.

$$u = f_{FC}(\vec{x}) \quad (2.2)$$

where FC denotes Fuzzy Control.

The purpose of our algorithm is to imitate the SF plane of (2.1) by some implication rules which in term will be represented by an FC plane. In order to display the control actions plane in a 3-D diagram, we only consider the control systems with two state variables, i.e. we assume

$$f_{SF}(x,y) = p*x + q*y : \text{the state feedback plane,}$$

$$f_{FC}(x,y) : \text{the fuzzy control surface, where } x \text{ and } y \text{ are the state variables.}$$

For a general control system having $m > 2$ state variables, this algorithm is still applicable with some decomposition techniques to be discussed

later. The algorithm is then described in the following steps:

Step 1: We first define the domain of interest of our imitation algorithm. It is reasonable and practical to consider only a limited region of interest instead of the whole space since all the variables are bounded in finite time in the real world. The domain of interest can be characterized by the following parameters:

$$\begin{aligned} x_{\max} &= \text{maximum of the absolute values of } x \text{ considered,} \\ y_{\max} &= \text{maximum of the absolute values of } y \text{ considered,} \\ x_u &= \max\{x_{\max}, y_{\max} * q/p\}, \\ y_u &= \max\{y_{\max}, x_{\max} * p/q\}, \\ x_l &= -x_u, \\ y_l &= -y_u. \end{aligned}$$

In other words, the region formed by the coordinate points (x_l, y_l) , (x_l, y_u) , (x_u, y_l) and (x_u, y_u) is the smallest rectangle covering the possible operating range of the state variables, which has the zero crossing line L ($p*x + q*y = 0$) as its diagonal. From the definitions, we have

$$p*x_u + q*y_l = 0 \quad (2.3)$$

$$p*x_l + q*y_u = 0 \quad (2.4)$$

Without loss of generality, it is assumed that the slope of line L is negative (i.e. p and q are of the same sign). In case of a positive slope, the format can be maintained simply by change of variables.

Step 2: Partition the x -axis bounded by x_l and x_u into n regions with $n+1$ partition points (including x_u and x_l). For the i -th ($i = 1$ to $n+1$) partition point x_i , assign a linguistic term A_i such that the membership function of this term is a equal-sided triangle centered at x_i and sided at x_{i-1} and x_{i+1} . Repeat the same process on y -axis to get partition points y_i ($i = 1$ to $n+1$) and the linguistic terms B_i .

Step 3: Construct $(n+1)^2$ linguistic control rules with a general form as follows:

$$\text{If } x \text{ is } A_i \text{ and } y \text{ is } B_j, \text{ then the control action is } F_{i,j}$$

where $F_{i,j}$ is a linguistic term with a membership function:

$$\mu_{F_{i,j}}(z) = \begin{cases} (z-H_{i+j-1}) * S(z-H_{i+j-1}) - (z-H_{i+j}) * S(z-H_{i+j}) & \text{if } i+j > n \\ 0 & \text{if } i+j = n \\ (-z-H_{i+j-1}) * S(-z-H_{i+j-1}) - (-z-H_{i+j}) * S(-z-H_{i+j}) & \text{if } i+j < n \end{cases} \quad (2.5)$$

In (2.5), $S(\cdot)$ is unit step function, and

$$H_{i+j-1} = (i+j-n-1) * dH \quad (2.6)$$

$$H_{i+j} = H_{i+j-1} + dH, \quad (2.7)$$

where

$$dH = (p*x_u + q*y_u)/n \quad (2.8)$$

$$= -(p*x_l + q*y_l)/n \quad (2.9)$$

Therefore the weight of the rule in (2.5) when x and y are given is

$$W_{i,j} = \min\{\mu_{A_i}(x), \mu_{B_j}(y)\} \quad (2.10)$$

The control action corresponding to each rule is given by

$$F_{i,j} = (i+j-n-1) * dH + W_{i,j} * dH \quad (2.11)$$

The output of the fuzzy controller is calculated by taking the weighted average of the control actions of all the rules according to their weights respectively. The overall control action can be derived as:

$$f_{FC}(x,y) = \frac{\sum_{i=0}^{i=n} \sum_{j=0}^{j=n} F_{i,j} * W_{i,j}}{\sum_{i=0}^{i=n} \sum_{j=0}^{j=n} W_{i,j}} \quad (2.12)$$

Properties of the Algorithm

We have proposed an algorithm to imitate the control action surface (i.e. SF plane) of a state feedback controller by using some linguistic control rules. The complete rule base is actually a working fuzzy controller except that its rules are not based on the experiences of the human experts but a

state feedback control scheme.

In the section, we wish to discuss some of the properties of the algorithm in the followings.

Property 1: At each grid point (where x and y coordinates belong to one of the partition points), the imitated control action is exactly the same as the value given by the state feedback controller.

It is easy to see that only one rule will be fired with weight 1 at $x = x_i$ and $y = y_j$ (i.e. at the grid points), for $i = 1, \dots, n+1, j = 1, \dots, n+1$. The control actions created by the proposed fuzzy logic controller can be shown to be equal to the control actions of the state feedback controller at each grid points.

Property 2: At each non-grid point, at most 4 rules can be fired.

This property comes from the fact that given a specific value of any state variable, it has at most 2 non-zero membership value with respect to all the linguistic terms related to this variable. And because of this property, the computational complexity can be made independent of the number of rules.

Property 3: As the number of linguistic control rules increase, the average error between SF plane and FC surface decreases, where the average error is defined as

$$\text{Average Error} = \int_{x_1}^{x_u} \int_{y_1}^{y_u} |f_{FC}(x,y) - f_{SF}(x,y)| dx dy \quad (3.1)$$

Property 3 can be inferred intuitively from property 1. Since SF plane and FC plane coincide at each grid point, the average error is expected to decrease as the number of grid points (which is proportional to the number of fuzzy implication rules) increases. Rigorous mathematical proof of this property is complicated and will not be discussed in this paper. However, computer simulations were employed and the results are shown and discussed through a special control problem in the next section.

We have successfully dealt with the case of 2 state variables in section 2 and 3. Furthermore, for the cases of more than 2 state variables, a decomposition scheme is proposed to divide the problem into the format of the solvable 2-state case. Simply stated, for an m-variable ($m > 2$) system, we can decompose the system into many 2-state sub-systems with possibly one variable alone as one system. Then we apply the algorithm to each sub-systems and sum all the outputs of the $\lfloor m/2 \rfloor$ systems to get the final output.

Simulation Results

In this section, we will use a famous control problem, the inverted pendulum, to test our algorithm and compare the performance between state feedback controller and the imitated one, i.e., fuzzy logic controller.

The inverted pendulum system consists of a pole and a motor-driven cart. The cart moves on the rail tracks to its right, or left, depending on the controller's output. The pole is hinged to the cart through a frictionless free joint (i.e. it has only one degree of freedom). The goal of control is to balance the pole by supplying appropriate voltage to the cart motor.

At first, we derive a state feedback controller which can keep the pole balanced:

$$f_{SF}(x,y) = 880*x + 80*y$$

where x is the angle between pole and the vertical position, and y is the time derivative of x, which is the angular velocity.

The imitating fuzzy controllers with different number of partitions are constructed according to the procedures described in section 2. Then the differences between the SF plane and the FC surfaces with different n's can be shown in Figure 1.

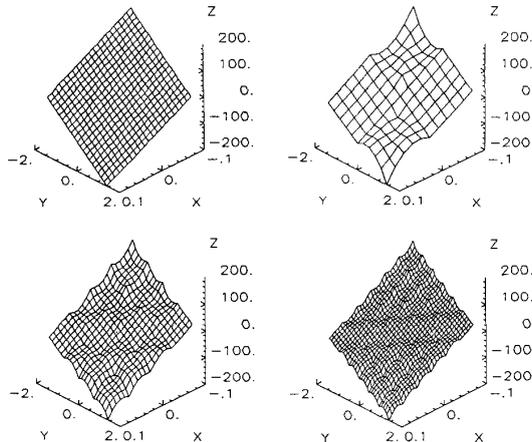


Figure 1: (a) SF plane and various FC surfaces with (b)n=2 (c)n=4 (d)n=6

From the figure, it can be seen that the imitating FC surfaces are closely matched to the desired SF plane. It can also be seen in Figure 2 that the matchings become better as the number of partitions increases, which is stated in Property 3.

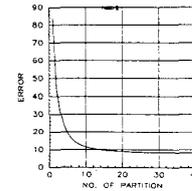


Figure 2: The average error between the SF and FC planes

What we have done so far is only related to the matchings of the control actions planes, not to the control systems directly. In order to see how the FC surfaces work as compared to the original SF plane, we applied both controllers to the inverted pendulum system. Both systems start from the same nonzero initial condition with sampling time equal to 5m second. The performances are shown in Figure 3.

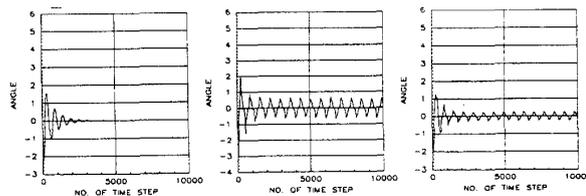


Figure 3: Pole angle using (a) SF and various FC with (b)n=8, (c)n=12.

From the above two diagrams, it's obvious that the performance of fuzzy controller resembles state feedback controller's as n increases. However, due to the idiosyncrasy of fuzzy controller, it seems fuzzy controllers always have a larger steady state error.

Conclusions and Future Work

In this paper, we have proposed an algorithm that can encode a set of linguistic control rules to imitate a given state feedback plane. We have shown how the average error between these control actions decreases with the increase of the number of rules. An inverted pendulum system was used to compare the performances of these two schemes. For a system with more than 2 state variables, a simple decomposing method was suggested such that our algorithm can still be applied.

Although we have successful using linguistic control rules to imitate a PD controller (state feedback controller), this is only the initial exhibition of the flexibility of fuzzy controller. It is expected that with some appropriate rules and various defuzzification strategy, fuzzy controller can imitate any given surface (presumably a optimal control action surface) with limited average error. Furthermore, with the auto-tuning process proposed by [11], the average error is expected to go down as a result of the tuning of the membership function of each linguistic term.

Reference

- [1] Mamdani, E. H., and Assilian, S., "An Experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.
- [2] Zadeh, L. A., "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- [3] Zadeh, L. A., "Fuzzy Logic," *IEEE Computer*, pp. 83-93, 1988.
- [4] Larkin, L. I., "A Fuzzy Logic Controller for Aircraft Flight Control," *Proc. of the 23rd Conference of Decision and Control*, Las Vegas, 1984.
- [5] Hogle, P. A. and Bonissone, P., "A Fuzzy Algorithm for Path Selection in Autonomous Vehicle Navigation," *Proc. 23rd IEEE Conference on Decision and Control*, Las Vegas, 1984.
- [6] Scharf, E. M., "The Application of a Fuzzy Controller to the Control of a Multi-Degree-of-Freedom Robot Arm," *Industrial Application of Fuzzy Control*, Sugeno, M. ed., North-Holland, 1985.
- [7] Bernard, J. A., "The Construction and Use of a Knowledge Base in the Real-Time Control of Research Reactor Power," *proc. 6th power Plant Dynamics, Control, and Testing Symp.*, Knoxville, TN, 1986.
- [8] Kasai, Y. and Morimoto, Y., "Electrically Controlled Continuously Variable Transmission (ECVT-II)," *Proc. of the Int. Congress on Transportation Electronics*, Dearborn, 1988.
- [9] S. Yasunobu and S. Myamoto, "Automatic train operation by predictive fuzzy control," in *Industrial Applications of Fuzzy Control*, M. Sugeno, ed., Amsterdam: North Holland, 1985.