

Nonlinear Input Mapping in Fuzzy Control Systems

Chiy-Ferng, Perng¹, *Student Member IEEE*, Sinn-Cheng, Lin¹, *Student Member IEEE*,

and Yung-Yaw, Chen¹, *Member IEEE*

Electrical Engineering Department, National Taiwan University

Taiwan, R.O. C.

1. Rm. 202, Electrical Engineering Dept., National Taiwan University, e-mail: pcf@ipmc.ee.ntu.edu.tw

Abstract

The input scaling factors in a fuzzy control system are often used as a transformation from the real input data to the desired space. Sometimes, they are tuned for better performance just as the coefficients in a PID controller. Theoretically, they are constant and should be adjusted for better performance while the operating point is changed. In this paper, we suggest nonlinear mapping functions to substitute the role of input scaling factors. The results will show how the nonlinear mapping function work and the performance could be better by proper adjustment. It is also noticed that such mapping will change the shapes of the membership functions. That means we need not to tune every membership function of linguistic variables, and just to choose a proper nonlinear mapping function could reach the same effects. In this paper, we used a inverted pendulum system to verify the results.

I. Introduction

Since the introduction of fuzzy set theory by Zadeh [7], fuzzy set theory has been applied in many fields successfully. Fuzzy control is one of these fields. After Mamdani use a controller based on fuzzy linguistic rules to control a steam engine [5], fuzzy control theory did not only be developed in the laboratory but also get into the real world. Especially in Japan, such as washing machines, vacuum cleaners, auto-focusing cameras, etc., have adopted fuzzy control to promote their performance.[6] "Fuzzy" becomes a magic word and could increase the value of the merchants.

A fuzzy controller often composes of four parts - a input-output interface, a rulebase, a database, and an inference engine. The I/O interface is also described as a fuzzification and defuzzification procedure. It will interpret the linguistic meanings of

input and output of a fuzzy controller. The rulebase is the soul of the controller. It is in charge of control motions. The database stores the necessary coefficients or data in the controller. The inference engine collects the input data to calculate the appropriate actions. The performance of a fuzzy controller depends on the design of the rulebase. The basic idea is to make use of the knowledge of field experts and transform the knowledge into linguistic if-then rules. Fuzzy control does not involve complex mathematical model. More often, it depends on the instinct of designers.

To design a fuzzy control rulebase, the definition of fuzzy linguistic term such as small, large, should be given on the universe that the state really happens. For convenience, the strategy to normalize a state space within the domain [-1,1] is common. Through input scaling factors, normalizing strategy can easily transform the actual variables into normalized variables. There are several papers [2][3][4] which discussed the effects of input scaling factors. There are also several learning fuzzy controller that achieve the optimal performance by tuning the scaling factors.[4][5] In fuzzy controllers, tunable parameters include membership function location, vertex, etc. Input scaling factors are just one of them. In these cases, input scaling factors are constant. Changing them did not change the shape of a membership function. And compared with the actual rulebase, the relative location would not vary with the input scaling factors. Scaling factors just magnify the scale proportionally.

Based on the work of Chen and Perng [8], change of input scaling factors is the same as change of the universe of discourses. On the other hand, the rulebase is also changed. Learning procedure also changes the rulebase. Constant input scaling factors do not adjust the shapes or positions of the membership functions of linguistic variables but nonlinear mappings could because the gains are

different everywhere. If we choose a proper nonlinear mapping function, the rulebase could be adjusted as desired.

II. Nonlinear Mapping Strategy

The basic structure of a fuzzy controller with input scaling factors GE and GD is shown in figure 1.

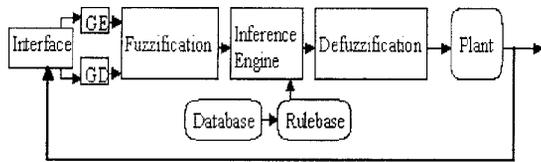


fig.1 the basic structure of a fuzzy controller

GE and GD play important roles in the controller. They are responsible for adjusting the scale of input errors and differential errors suitable for the normalized space. After multiplying the scaling factors, the scale of the rulebase is changed:

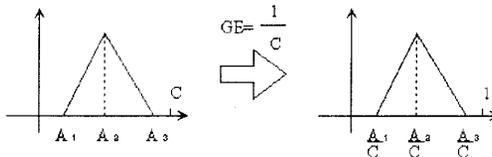


fig.2 The transformation of rulebase under linear input scaling factor

Although the position of the membership function shift into the normalized domain, the relative position to the other linguistic terms remain unchanged and the shapes are the same. That means input scaling factors did not affect structure of the rulebase. If the input scaling factors are different in different domain, that is to say the gains of inputs are different everywhere and the structure will change.

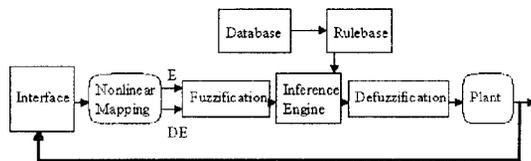


fig.3 Nonlinear mapping fuzzy controller

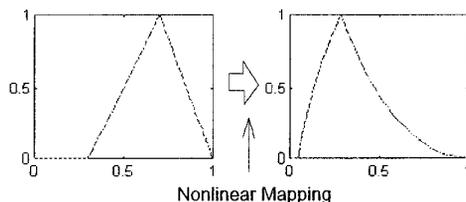


fig.4 Membership function changed under nonlinear mapping

In figure 4, the nonlinear mapping function is:

input : x

output: y

nonlinear mapping function:

$$y = 1 - \sqrt{1 - x^2} \quad (1)$$

We use this nonlinear mapping function to test how it change the shape of a membership function. It is obvious that the position and the shape of the membership function has changed. If different membership functions could deduce different performance, using nonlinear mapping properly, we can get suitable membership function to achieve better performance. The following block diagram is flow chart of the control motions under nonlinear mapping.

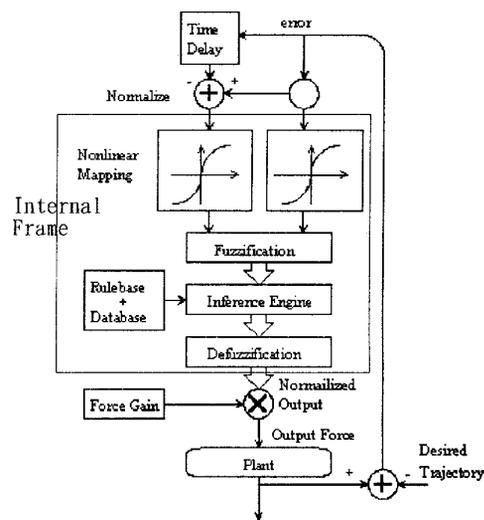


fig.5 The flow chart of nonlinear mapping fuzzy controller

At first, input signals should be normalized. After normalized, signals enter the nonlinear mapping function and they are mapped into a new domain. Then the new signals flow into the internal frame. The mapping procedure is the major part of the internal frame. This procedure is critical for the rulebase. In fact, the internal frame is similar to a standard fuzzy controller. Because of the nonlinear mapping function, the rulebase changed while observed from outside. The computation inside the internal frame is the same with other fuzzy controllers. Because the inputs are normalized, we inclined to normalize the output signals. It should be multiplied by a constant gain. In the following section, we will show the simulation results.

III. Simulation Results

The plant is an inverted pendulum showed in figure 6. In this simulation, we ignored the cart position and the cart velocity. The pole angle and

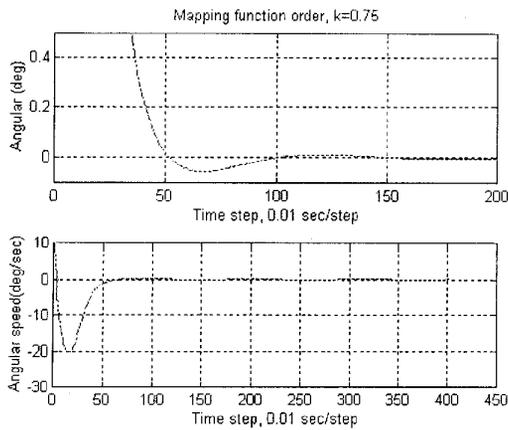


fig. 12 The performance of $k=0.75$

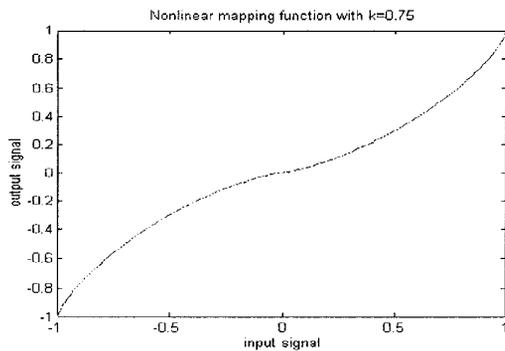


fig. 13 Nonlinear mapping function with $k=0.75$

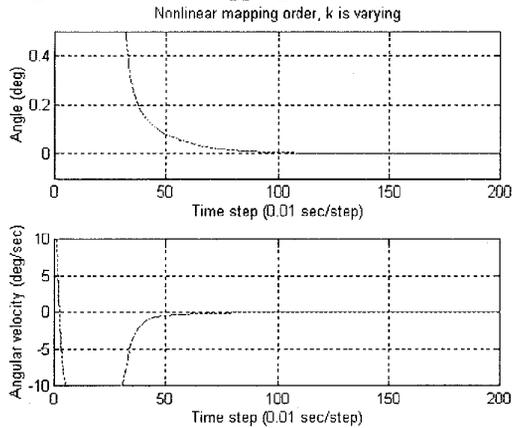


fig. 14 The performance of varying k

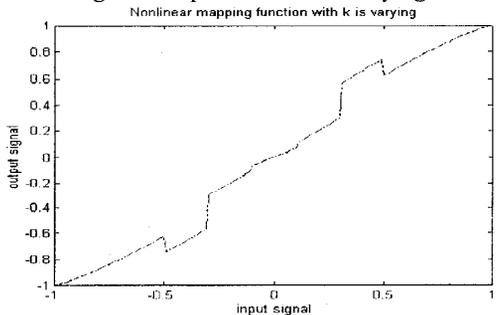


fig. 15 Nonlinear mapping function with varying k

IV Discussions

From the performance plots, we can conclude that higher k can speed up the convergence but it would induce the oscillation around the equivalent point. The shape of the nonlinear mapping function while k is 2 shows that it will map the domain surrounding the zero into a greater domain in the new space. If the states are getting close to the central area, through nonlinear mapping, the new states in the new domain would be a little far away from the equivalent point. That means the fired rules would not be the original central rules. Even the states are near the equivalent point and the force should close to zero. Because of the nonlinear mapping, the force would be larger than what we desired. This action would induce oscillation around the equivalent point. On the other hand, if the states are a little far away from the origin, the fired rules would be those locating at the edge of the state space. Generally speaking, these rules often use full strength such that the plant could be push back to the central area. The speed of convergence would be faster. Lower k could reduce the oscillation of the system, but the speed of convergence also reduced. Because the nonlinear mapping function will condense the domain surrounding the equivalent point, the number of fired rules will be smaller. The relative output will also be smaller. This action will slow down the convergence.

For accelerating the convergent speed and avoiding the system from oscillating, varying k is a good strategy. When the states are still far away from the equivalent point, higher k could make the system behave forward the equivalent point. When the states are behaving forward zero readily, k needs to be reduced such that the oscillation of the system would not happen. In our simulation, when the normalized inputs are greater than 0.5, k is 1.2. Inputs are greater than 0.3 but smaller than 0.5 and k is 1.5. Inputs are greater than 0.1 but smaller than 0.3 and k is 1. Inputs are smaller than 0.1 and k is 0.9. This is a heuristic scheme. From the figure 14, the performance is indeed better.

V. Conclusions

The nonlinear mapping function could change the shape and position of a linguistic term in the rulebase. From the simulation, choosing the function properly, the performance of a fuzzy controller could be improved. The problem is how to decide the nonlinear function. As generally known, learning algorithm could increase the performance of the fuzzy controller. For the same reason, we can also use learning algorithms to learn the nonlinear function. Self-learning could be the solution to solve the problem of how to choose nonlinear function.

References

- [1] Y.-Y. Chen, "Global analysis of fuzzy dynamical systems". Ph. D. thesis, University of California, Berkeley, 1989
- [2] W. Bare, R. Mulholland, and S. Sofer, "Design of a self-tuning rule based controller for a gasoline refinery catalytic reformer", *IEEE Trans. of Automatic Control*, Vol. 35, pp. 156-164, 1990
- [3] M. Braae and D. A. Rutherford, "Selection of parameters for a fuzzy logic controller", *Fuzzy Sets and Systems*, Vol. 2, pp. 185-199, 1979
- [4] M. Maeda and S. Murakami, "A self-tuning fuzzy controller", *Fuzzy Sets and Systems*, Vol. 51, pp. 29-40, 1992
- [5] T. Procyk and E. Mamdani, "A linguistic self-organizing process controller", *Automatica*, Vol. 15, pp. 15-30, 1979
- [6] M. Sugeno ed., *Industrial applications of fuzzy control*, 1986
- [7] Zadeh, L. A., "Fuzzy Sets", *Information and Control*, pp. 338-353, 1965
- [8] Y.-Y. Chen and C.-F. Perng, "Input scaling factors in fuzzy control systems", *IEEE-FUZZ* 1994