



SELF ORGANIZING FUZZY CONTROLLER FOR A NON-LINEAR TIME VARYING SYSTEM

Omar W. Abdul-Wahab
University of Baghdad
College of Engineering
Computer Engineering Department

ABSTRACT

This paper proposes a self organizing fuzzy controller as an enhancement level of the fuzzy controller. The adjustment mechanism provides explicit adaptation to tune and update the position of the output membership functions of the fuzzy controller. Simulation results show that this controller is capable of controlling a non-linear time varying system so that the performance of the system improves so as to reach the desired state in a less number of samples.

INTRODUCTION

While nonadaptive fuzzy control has proven its value in some applications, it is sometimes difficult to specify the rule base for some plants, or the need could arise to tune the rule-base parameters if the plant changes. This provides the motivation for adaptive fuzzy control, where the focus is on the automatic on-line synthesis and tuning of fuzzy controller parameters (i.e., the use of on-line data to continually “learn” the fuzzy controller, which will ensure that the performance objective is met). The most important advantage of adaptive fuzzy control over conventional adaptive control is that adaptive fuzzy controller is capable of incorporating linguistic fuzzy information from human operators (Layne and Passino 1996) . This is especially important for the systems with a high degree of uncertainty, such as chemical process and aircraft, because although these systems are difficult to control from a control theoretical point of view, they are often successfully controlled by human operators (Alci, Karabiyik and Karatepe 2003).

The first adaptive fuzzy controller called the linguistic self-organizing controller (SOC) was introduced in (Procyk and Mamdani 1979). Applications of this method have been studied by (R. Tanscheit and E. Scharf) for robotics applications and by (S. Daley and K.F.Gill) for altitude control of a spacecraft. More recently, the “fuzzy model reference learning controller” (FMRLC) was introduced in (Layne and Passino 1996) , its extensions in Kwong and Passino 1996 , and both simulation (Kwong and Passino 1996) (Passino and Yurkovich 1992) (Kwong and Passino 1995) (Lennon and Passino 1995) and implementation studies (Zumberge and Passino 1996) (Moudgal, Kwong, Passino, and Yurkovich 1995) have shown this method to be quite successful. The self-organizing controller (SOC) has a hierarchical structure in which the lower level is a rule-based controller and the higher level is the adjustment mechanism as shown in **Fig. 1**.

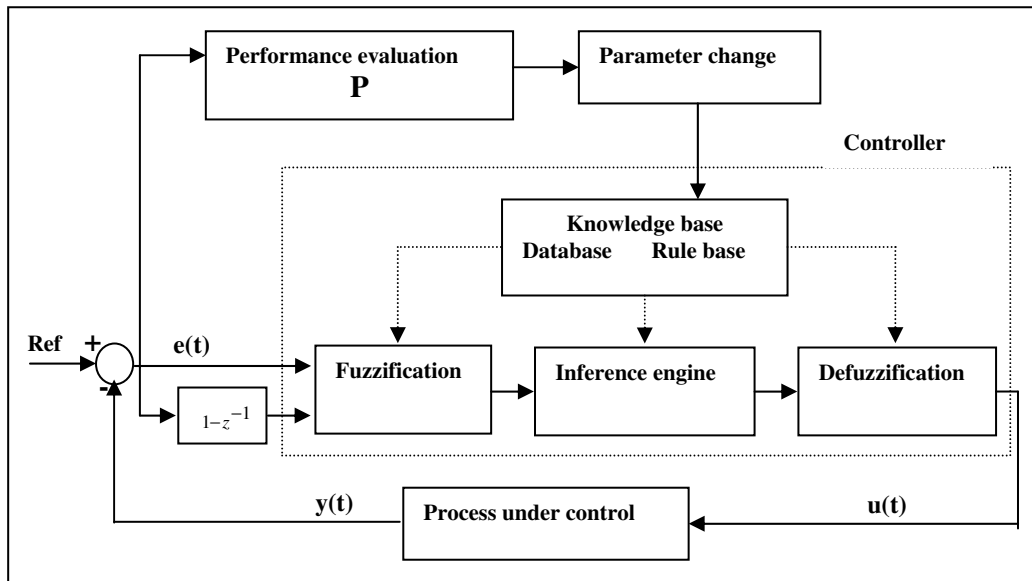


Fig. (1).The hierarchal structure of the self organizing controller.

The lower level is a controller whose two inputs are the error e and the change in error Δe . These are multiplied by two gains, g_e and $g_{\Delta e}$ respectively. The output of the controller is the control variable u , which is multiplied by the output gain g_u . (Jantzen 1998)

The idea behind the self-organization is to let an adjustment mechanism update the rule base of the controller, based on the current performance of the controller. Basically, if the performance is poor, the responsible rule should be modified, such that next time that rule is visited, the control signal will be better.

The higher level monitors error (e) and change in error (Δe), then it modifies the rule base through a modifier algorithm when necessary. It uses a performance measure to decide the magnitude of each change. The performance measures are numbers, organized in a table \mathbf{P} , expressing what is desirable, or undesirable rather, in a transient response. The table \mathbf{P} can be built using linguistic rules, but it is often built by trial and error. The same performance table may be used with a different process, without prior knowledge of the process, since it only expresses the desired transient response. The controller can start from scratch with a rule base full of zeros; it will, however, converge faster towards a stable rule base, if the rule base is primed with sensible numbers to begin with.

The SOC learns to control the system in accordance with the desired response. This is called training. At the sample instant n ,

- 1- It records the deviation between the actual state and the desired state, and
- 2- It corrects the rule base of the controller accordingly.

The performance table \mathbf{P} evaluates the current state and returns a performance measure $P(i_n, j_n)$, where i_n is the index corresponding to e_n , and j_n is the index corresponding to Δe_n .

Tables (1) and (2) are examples of early performance tables. Observing these two tables it seems that **table (2)** holds the zeros in a more or less diagonal band, while **table (1)** keeps them in a z-shaped patch. If the performance measure is zero, the state is satisfactory, otherwise unsatisfactory to some degree. In the latter case the parameter change assumes that the control signal must be modified. It cannot be the current control signal that is responsible, however, because it takes some time before a control action shows up in the process output.

The simple strategy is to go back a number of samples in time to correct an earlier control signal. The parameter change must therefore know the time lag in the plant It goes back d samples comparable to the time lag; the integer d is called the delay-in-penalty.

The parameter change assumes that the plant output depends monotonously on the input. It is thus a requirement, for the SOC to function correctly, that an increase in the plant output calls for an adjustment of the control signal always in the same direction, whether it be an increase or a decrease.

The precise adjustment rule is

$$u_n = u_{n-d} + p_n \tag{1}$$

The time subscript n denotes the current sample. In words, it regards the performance measure as an extra contribution to the control signal that should have been, in order to push the plant output to a state with a zero penalty. (Jantzen and Neils 2003)

The proposed self organizing controller uses the adjustment rule

$$u_n = u_{n-d} + \mu_{n-d} p_n \tag{2}$$

so that the amount of shift of the output membership function centers is scaled by the μ_{n-d} certainty of their premises. (Passino and Yurkovich 1998)

Table (1). Example of a performance table (adapted from Procyk and Mamdani, 1979).

Δe

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-6	-6	-6	-6	-6	-6	-6	0	0	0	0	0	0
-5	-6	-6	-6	-6	-6	-6	-6	-3	-2	-2	0	0	0
-4	-6	-6	-6	-6	-6	-6	-6	-5	-4	-2	0	0	0
-3	-6	-5	-5	-4	-4	-4	-4	-3	-2	0	0	0	0
-2	-6	-5	-4	-3	-2	-2	-2	0	0	0	0	0	0
-1	-5	-4	-3	-2	-1	-1	-1	0	0	0	0	0	0
0	-4	-3	-2	-1	0	0	0	0	0	1	2	3	4
1	0	0	0	0	0	0	1	1	1	2	3	4	5
2	0	0	0	0	0	0	2	2	2	3	4	5	6
3	0	0	0	0	2	3	4	4	4	4	5	5	6
4	0	0	0	2	4	5	6	6	6	6	6	6	6
5	0	0	0	2	2	3	6	6	6	6	6	6	6
6	0	0	0	0	0	0	6	6	6	6	6	6	6

e

Table (2). Example of another performance table (adapted from Yamazaki, 1982).

Δe

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-6	-6	-6	-6	-6	-6	-6	-5	-4	-3	-2	-1	0
-5	-6	-6	-6	-6	-5	-4	-4	-4	-3	-2	-1	0	0
-4	-6	-6	-6	-5	-4	-3	-3	-3	-2	-1	0	0	1
-3	-6	-6	-5	-4	-3	-2	-2	-2	-1	0	0	1	2
-2	-6	-5	-4	-3	-2	-1	-1	-1	0	0	1	2	3
-1	-5	-4	-3	-2	-1	-1	0	0	0	1	2	3	4
0	-5	-4	-3	-2	-1	0	0	0	1	2	3	4	5
1	-3	-2	-1	0	0	0	0	1	1	2	3	4	5
2	-2	-1	0	0	0	1	1	1	2	3	4	5	6
3	-1	0	0	0	1	2	2	2	3	4	5	6	6
4	0	0	0	1	2	3	3	3	4	5	6	6	6
5	0	0	1	2	3	4	4	4	5	6	6	6	6
6	0	1	2	3	4	5	6	6	6	6	6	6	6

e

A Linear Approximation To The Performance Measure

The original performance tables were built by hand, based on trial and error. Presumably, if the numbers in the table **P** are small in magnitude, many updates are required before the rule base of the controller converges to a steady rule base. Oppositely, if the numbers in **P** are large in magnitude, the convergence should be faster, but it may also be unstable.

Observing the two original performance tables, Procyk and Mamdani preferred to keep the zeros in a z-shaped region, while Yamazaki keeps them in a more or less diagonal band. As the zeros indicate no penalty, those states are desired. Assuming that the system stays within a zero band along the diagonal, keeping a zero diagonal is equivalent to keeping the sum of the rule base inputs at zero. A simple linear approximation to the table **P** is

$$P(i_n, j_n) = i_n + j_n \tag{3}$$

DESIGN OF THE SELF ORGANIZING CONTROLLER

The self organizing controller has the following three main parts:

FUZZY LOGIC CONTROLLER

In this controller, a singleton fuzzification has been used so that the values of the membership functions are computed at the current input values (Passino and Yurkovich 1998)

For the input variables (error e and change of error Δe), five triangular membership functions have been used. They are Negative Large (NL), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Large (PL). The indices of these membership functions are -2, -1, 0, 1, and 2 respectively. Each membership function is overlapped with the two adjacent membership functions by 0.5. The outermost two membership functions (NL and PL) are saturated. The equation of any one of these membership functions is

$$\mu(x) = \begin{cases} 1 - \frac{|x - c|}{0.5} & \text{if } c - 0.5 \leq x \leq c + 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where c is the center of the membership function.

For the control output variable twenty five singleton membership functions have been used with there centers initialized at u=0. The reason behind using twenty five output membership functions

is that each rule is assigned its own consequent membership function so that changing the consequent membership function of one rule has no effect on the other rules. The input and output membership functions are shown in Fig (2).

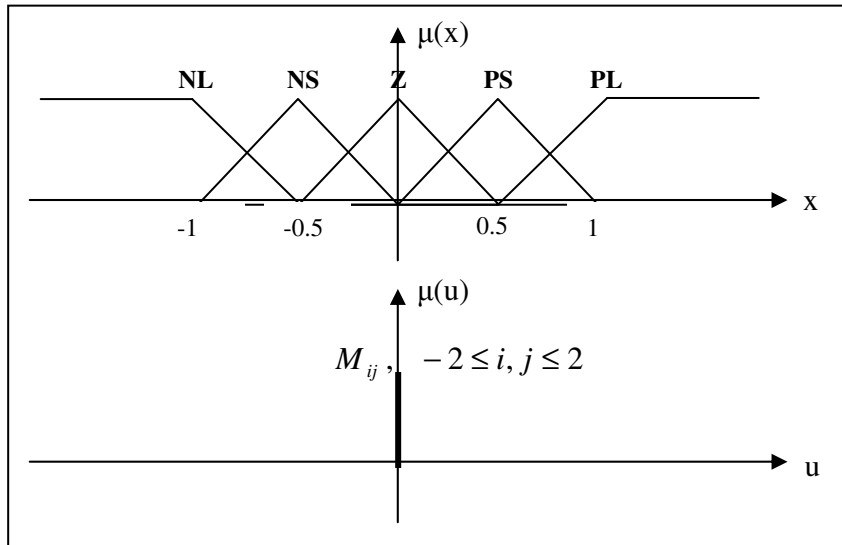


Fig (2). The input and output membership functions.

The rule base of this controller is given in table (3) below. This rule base was generated according to the following equation:

$$Rule_{ij} = k, \quad 0 \leq k \leq 24 \tag{5}$$

where i and j are the indices of the input membership functions and $Rule_{ij}$ is the index of the output membership function.

Table (3). Rule base of the fuzzy controller

		Δe				
		NL(-2)	NS(-1)	Z(0)	PS(1)	PL(2)
e	NL(2)	0	1	2	3	4
	NS(1)	5	6	7	8	9
	Z(0)	10	11	12	13	14
	PS(-1)	15	16	17	18	19
	PL(-2)	20	21	22	23	24

The inference step is taken by computing, for each rule, the implied fuzzy set. Defuzzification was achieved using center-average defuzzification technique, according to the following equation

$$u = \frac{\sum_{i=0}^{R-1} b_i \times \mu_i(e, \Delta e)}{\sum_{i=0}^{R-1} \mu_i(e, \Delta e)} \tag{6}$$

where b_i is the center (position) of the i th output membership function associated with rule i , $\mu_i(e, \Delta e)$ is the membership value for the premise of rule i which represents the certainty that rule i holds for the given inputs e and Δe , and R is the number of rules.

PERFORMANCE EVALUATION

The original performance tables were built by trial and error. Presumably, if the numbers in the table \mathbf{P} are small in magnitude, many updates are required before the rule base of the controller converges to a steady rule base. Oppositely, if the numbers in \mathbf{P} are large in magnitude, the convergence should be faster, but it may also be unstable.

Observing the two original performance tables, Procyk and Mamdani preferred to keep the zeros in a z-shaped region, while Yamazaki keeps them in a more or less diagonal band. As the zeros indicate no penalty, those states are desired. Assuming that the system stays within a zero band along the diagonal, keeping a zero diagonal is equivalent to keeping the sum of the rule base inputs at zero (Jantzen and Neils 2003). A simple linear approximation to the table \mathbf{P} is,

$$P(i_n, j_n) = i_n + j_n \tag{7}$$

The result is the following table:

Table (4). Performance evaluation table of the self organizing controller.

		Δe				
		NL(-2)	NS(-1)	Z(0)	PS(1)	PL(2)
e	NL(2)	0	1	2	2	2
	NS(1)	-1	0	1	2	2
	Z(0)	-2	-1	0	1	2
	PS(-1)	-2	-2	-1	0	1
	PL(-2)	-2	-2	-2	-1	0

PARAMETER CHANGE

According to equation (1), the output of the controller at $n-d$, must be changed by an amount p_n . This may be done by changing the centers of the output membership functions that were ‘‘ON’’ at time $n-d$ by p_n .

$$\begin{aligned}
 u_{n-d} &= \frac{\sum_{i=0}^{R-1} (b_i + p_n) \times \mu_i(e, \Delta e)}{\sum_{i=0}^{R-1} \mu_i(e, \Delta e)} = \frac{\sum_{i=0}^{R-1} b_i \times \mu_i(e, \Delta e) + \sum_{i=0}^{R-1} p_n \times \mu_i(e, \Delta e)}{\sum_{i=0}^{R-1} \mu_i(e, \Delta e)} \\
 &= \frac{\sum_{i=0}^{R-1} b_i \times \mu_i(e, \Delta e)}{\sum_{i=0}^{R-1} \mu_i(e, \Delta e)} + \frac{\sum_{i=0}^{R-1} p_n \times \mu_i(e, \Delta e)}{\sum_{i=0}^{R-1} \mu_i(e, \Delta e)} \\
 &= u_{n-d} + p_n
 \end{aligned}$$

SIMULATION RESULTS

A simulation of a single-stage rocket system was used to show the effect of the proposed self organizing controller in controlling the velocity of the rocket (Passino and Yurkovich 1998). The model of this system is given by

$$\frac{dv(t)}{dt} = c(t)\left(\frac{m}{M - mt}\right) - g\left(\frac{R}{R + y(t)}\right) - 0.5v^2(t)\left(\frac{\rho_a AC_d}{M - mt}\right) \tag{8}$$

where $v(t)$ is the rocket velocity at time t (the plant output), $y(t)$ is the altitude of the rocket (above sea level), and $c(t)$ (the plant input) is the velocity of the exhaust gases. The constant values $M, m, A, g, R, \rho_a, C_d$ are given in (Passino and Yurkovich 1998).

Due to the loss of fuel resulting from combustion and exhaust the rocket has a time-varying mass. Furthermore, it can be determined by inspection of equation (8) that the system is a nonlinear process. The primary purpose for considering this control application is to investigate the capability of the self organizing controller for controlling non-linear time-varying processes.

Fig. (3). shows the simulation results for a step response transient process. First the rule base of the controller was initialized so that the centers of the output membership function initialized at zero. Therefore the response of the first run has a large overshoot and a large settling time. After this run the rule base is changed so that for the same values of e and Δe the values of the control variable u will produce a better response. This is obvious in run 2. The problem that arises here is that while the rule base forces the control variable to enhance the transient response in terms of settling time, the overshoot get worse (become larger). The same thing is true for run 3, so that $run2=run3$. Also $run\ i = run\ 2$ for $i > 2$.

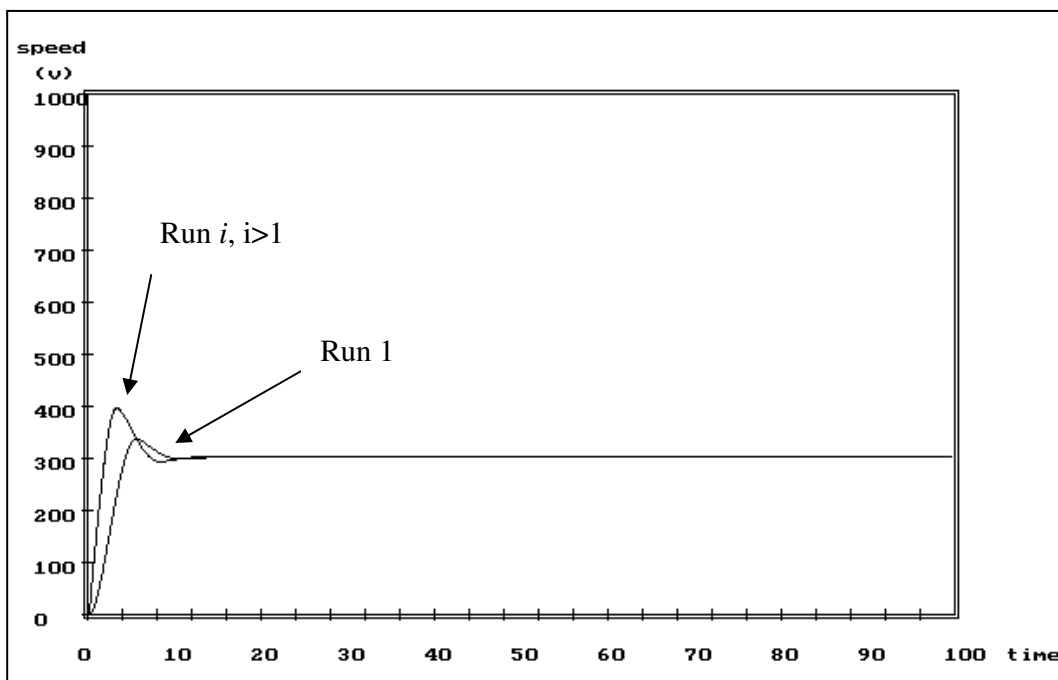


Fig. (3).Several step response runs of the simulated rocket system.

To get rid of this problem, the performance evaluation **table (5)** was used. This performance evaluation table is the same as that in **table (4)**, the only difference is the cell that corresponds to $(e, \Delta e) = (0, -1)$, which is here equal to -2 instead of -1 . The result of applying the latter performance evaluation table to the rocket system is shown in **Fig. (4)**.

Table (5). A new performance evaluation table of the self organizing controller.

		Δe				
		NL(-2)	NS(-1)	Z(0)	PS(1)	PL(2)
e	NL(2)	0	1	2	2	2
	NS(1)	-1	0	1	2	2
	Z(0)	-2	-2	0	1	2
	PS(-1)	-2	-2	-1	0	1
	PL(-2)	-2	-2	-2	-1	0

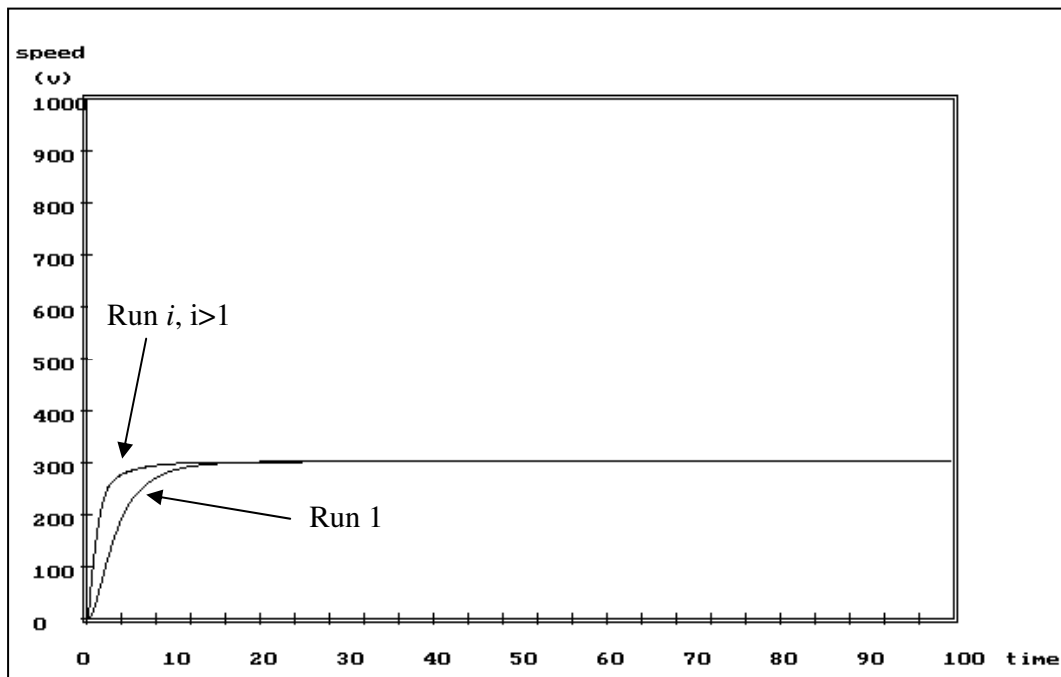


Fig. (4). Step response runs of the simulated rocket system with the new performance table.

CONCLUSIONS

A self-organizing controller of a fuzzy logic controller has been proposed as an enhancement level of the fuzzy controller. At any sampling instant each active rule is being incrementally changed according to the activation degree of this rule and the output of the performance evaluation unit so that the control surface is changing gradually in appropriate way to fit the process dynamics. This controller was used to control the velocity of a simulated rocket system, which is a non-linear time varying plant. Simulation results shows that the step response of the system was improved due to adjusting the rule base so as to reach the desired state $(e, \Delta e) = (0, 0)$ in a less number of samples.

REFERENCES

- 1- Alci, M., Karabiyik, A., and Karatepe, E., "Adaptive fuzzy control of nonlinear systems and comparative analyses," IJCI Proc. of international Conf. Signal Processing, vol. 1, no. 2, Sep., 2003.



- 2- Daley, S. and Gill, K.F., "Altitude control of a spacecraft using an extended self-organizing fuzzy logic controller," Proc. I. Mech. E., vol. 201, no. 2, pp. 97-106, 1987.
- 3- Jantzen, J., "The self-organizing fuzzy controller", Tech. Report no. 98-H 869 (soc), 19 Aug 1998.
- 4- Jantzen, J. and Neils K. Poulsen, "Adaptation in the fuzzy self-organizing controller", <http://www.eunite.org/eunite/events/eunite2003/eunite2003-ProgrammeOverview-04-07.pdf>.
- 5- Kwong, W. A. and Passino, K. M. "Dynamically focused fuzzy learning control," IEEE Trans. Syst., Man, Cybern., vol. 26, pp.53-74, Feb. 1996.
- 6- Kwong, W. A., Passino, K.M., Lauknonen, E. G. and Yurkovich, S., "Expert supervision of fuzzy learning systems for fault tolerant aircraft control," Proc. IEEE, Spec.. Issue Fuzzy Logic Eng. Applicat., vol. 83, pp. 466-483, Mar. 1995.
- 7- Layne, J. R. and Passino, K. M., "Fuzzy model reference learning control," J. Intell. Fuzzy Syst., vol. 4, no. 1, pp. 33-47, 1996.
- 8- Layne, J., Passino, K and Yurkovich, S. "Fuzzy learning control for antiskid braking systems," in Proc. IEEE Conf. Decision Contr., Tucson, AZ, Dec. 1992, pp. 2523-2528.
- 9- Lennon, W. and Passino, K., "Intelligent control for brake systems," in Proc. IEEE Int. Symp. Intell. Contr., Monterey, CA, Aug. 1995, pp. 499-504.
- 10- Passino, K. M. and Yurkovich, S. "Fuzzy control", Addison Wesley, 1998.
- 11- Procyk, T. and Mamdani, E., "A linguistic self-organizing process controller," Automatica, vol. 15, no. 1, pp. 15-30, 1979.
- 12- Moudgal, V. G., Kwong, W. A., Passino, K. M. and Yurkovich, S. "Fuzzy learning control for flexible-link robot," IEEE Trans. Fuzzy Syst., vol. 3, pp. 199-210, May 1995.
- 13- Tanscheit, R. and scharf, E., "Experiments with the use of a rule-based self-organizing controller for robotics applications," Fuzzy Sets and Systems, vol. 26, pp.195-214, 1988.
- 14- Zumberge, J. and Passino, K. M., "A case study in intelligent control for a process control experiment," in Proc. IEEE Int. Symp. Intell. Contr., Dearborn, MI, Sept. 1996, to be published.