N93-22...

Application of Genetic Algorithms to Tuning Fuzzy Control Systems

Todd Espy
Endre Vombrack
Jack Aldridge

Togai Infralogic, Inc.
17000 El Camino Real, # 204C
Houston, TX 77058

S3-63
150403
P. 12

Abstract

Real number genetic algorithms (GA) have been applied for tuning fuzzy membership functions of three controller applications. The first application is our "Fuzzy Pong" demonstration, a controller that controls a very responsive system. The performance of the automatically tuned membership functions exceeded that of manually tuned membership functions both when the algorithm started with randomly generated functions and with the best manually-tuned functions. The second GA tunes input membership functions to achieve a specified control surface. The third application is a practical one, a motor controller for a printed circuit manufacturing system. The GA alters the positions and overlaps of the membership functions to accomplish the tuning. This paper discusses the applications, the real number GA approach, the fitness function and population parameters, and the performance improvements achieved. Directions for further research in tuning input and output membership functions and in tuning fuzzy rules are described.

Introduction

A significant task in building fuzzy control systems is tuning the membership functions (MBFs) to improve or optimize the performance of the controller. The tuning task has been accomplished with fuzzy systems[1], neural networks[2], and genetic algorithms[3] (GAs). In this paper, we describe the use of real number genetic algorithms[4] to successfully tune membership functions for several fuzzy control systems. A significant feature of this work is that the input MBFs are tuned whereas many previous efforts have concentrated on tuning the output MBFs. Because both input and output membership functions are required to define the control surface for the fuzzy controller, this offers an added degree of flexibility to the tuning process. Whether such flexibility is, in fact, beneficial to fuzzy controller tuning is yet to be determined.

We first describe some aspects of real number genetic algorithms because that representation of genetic algorithms is less familiar than others. Next, we describe an application of matching a predefined control surface by tuning membership functions for the inputs. Third, we discuss the fuzzy pong application, a controller for an air flow driven by a fan and balancing a ping pong ball at a set position in a plastic tube. Fourth, we briefly discuss results for applying the technique to an AC servomotor control system. We then conclude with remarks about future directions.

Real Number Genetic Algorithms

Many genetic algorithm applications and theorems are based on bit string representations in which the parameters to be optimized are encoded in binary numbers, concatenated, and treated for GA manipulations as one continuous bit string. In tuning fuzzy membership functions, we found it more useful to keep the real number representation for the parameters of the MBFs and to manipulate the numbers using crossover and mutation techniques suitable to the real number representation[4].

Fig. 1 shows the representation of a collection of parameters as a list of real numbers. For the applications discussed below, we used five symmetric triangular membership functions with two parameters each, namely, the upper and lower ends of the support, for each universe of discourse. The fact that we need to represent pairs of ordered numbers favors the real number representation. We used twenty individuals in our populations, for convenience.

Because real number GAs are not extensively used, a standard set of operators is not yet defined. Fig. 2 illustrates our genetic algorithm operators for real number GAs: merge, crossover, mutate, and creep. Merge averages the parameters of two individuals to form the offspring. Crossover exchanges the real numbers between two fit individuals, pairwise. For the problem with two MBFs, the net effect is to replace left or right extents of the MBFs between fit individuals to concentrate the best combinations within a single individual. Presumably, the other individual would lose in the fitness evaluation during the next cycle. Mutate begins by selecting which fuzzy variable is to be selected on a random draw. For our case of two fuzzy input variables, the probability was 50-50 of selecting either one of the MBFs. Having selected the MBF, we perturb its parameters by randomly selected magnitudes. Creep is an operation in which all parameters of an individual are randomly perturbed. Creep is a hybridizing operation well-suited for search in the local area of an individual if the random variations are limited to some maximum. Our process used 5 individuals mated with the most fit of a generation by crossover, 5 most fit individuals mutated, 5 merged individuals from a pairwise competition, and 5 new individuals selected by random draw as the basis for choosing a new generation. A variant of the creep operator was used in later generations.

The input membership functions are symmetrical and described by an upper and lower end of the support. The peak of the triangular shape is midway between these extremes and has membership value of one. The controller we used was a two-input one output generic controller that could be customized to the application. The simplest interpretation is error and error_rate for the two inputs and control for the output. This interpretation varies from application to application as in the control surface generator described in the next section. With five MBFs for each fuzzy variable, the input MBFs are characterized by 20 numbers, the size of an individual in our population. Fig. 1 illustrates the correspondence between the MBF support parameters and the GA individuals.

Matching a Control Surface

The simplest of the tuning applications we performed was the tuning of membership functions to match a prespecified control surface. Although the control surface for a controller is generally not known a priori, in those cases where it is, GA tuning may be useful. One example of such a case might be the operation of a plant by an operator in which the control commanded manually is recorded with the plant sensors. Such relations would define a partial control surface that might be encoded in a fuzzy controller.

To illustrate the capability to tune to a given control surface, we tuned the MBFs of the inputs to a two-input(x,y), one-output(z) controller to match a control surface $x^2 + y^2 = 10z$. The fitness criterion was the sum of squares of differences between the predicted output for the controller and $(x^2 + y^2)/10$. The parameters of the GAs were adjusted to minimize the mean square error between these quantities over the control surface as measured at 121 points chosen in a square pattern across the center of the x-y plane.

Fig. 3 illustrates the performance for several randomly chosen starting populations. The mean square error converges rapidly with generation number. The best fits we have observed converge to approximately 15 on the same fitness scale. This suggests that the effects of local minima are significant and that knowledge of good initial membership functions will greatly assist convergence to optimal controllers.

The Fuzzy Pong Controller

The fuzzy pong is a controlled plant consisting of a ping-pong ball suspended on a column of air provided by a small fan whose voltage is controlled by the fuzzy controller or a proportional-integral-derivative (PID) controller. (The choice is made by which code is loaded into the microcontroller memory.) The ball's location in the plastic tube is determined using an ultrasonic acoustic range sensor located at the bottom of the tube. The servocontroller function is provided by a Hitachi H8/325 microprocessor board that drives a conventional transistor amplifier that serves as the DC voltage control for the motor voltage. The set point for control is provided to the H8 by an external personal computer (PC) that also is used as a monitor and data display device. There are two set points provided by the PC: high and low set points. When the ping pong ball stabilizes its position within user defined limits about either set point for a time preset by the user, the PC commands traversal to the other set point. The fuzzy controller commands the fan voltage based on the error = (set point - ball location) and the rate of change of error = (error(t) - error(t-1)), where t is the current time in units of the sample interval. The ability of the fuzzy controller to provide more precise control than the PID had been previously established through manual tuning to achieve smallest time transitions with minimal overshoot.

The GA tuning used a fitness function that measures the number of successful transitions, up to four, that an individual can accomplish, the rise time achieved in those transitions, and the overshoot that the transitions possess. If an individual cannot achieve success in stabilizing the ball within a predetermined time, the evaluation of the fitness is terminated. The achieving of the set point within a time limit allows the evaluation of other factors and offers a chance to try again up to four attempts. The fitness is evaluated using the hardware and is thus not deterministic because of the sensitivity of the pong to ball spin, initial position, air temperature, etc. The fitness over a sequence of populations thus may not monotonically decrease, even if the best individual from the preceding generation is kept to assure monotonicity.

Fig. 4 illustrates the fitness of the best individual in a generation as a function of generation. There is some improvement within a level established by success in finding the set point. The fitness is clearly dominated by the success in achieving the set point. The loss of a best individual also clearly limits system performance considerably. A strategy for handling this contingency such as requiring a number of generations before a best individual can be omitted might be useful. Development of an improved fitness criterion that places less emphasis on the number of sequential successes - perhaps running a fixed number of trials for each individual - would allow better discrimination of the transition characteristics. Achieving the commanded set point would need to continue to play an important role, however.

Motor Controller Tuning

We conducted experiments on tuning a fuzzy controller for an AC servomotor. The controller has been previously described[5]. It is a fuzzy PD controller capable of either control of the angular rate or the angular position. The controller exhibits "deadbeat" performance[6] - rapid response to unit step input without overshoot - that is faster than critically damped PID control.

This is an application in which tuning the input MBFs is particularly appropriate because the gains on the proportional and velocity controls are determined by MBF placement. The overall control gain achievable by tuning output MBFs alone does not provide the same ability to trade off between error and error rate that the input MBF tuning provides.

The GA tuning was able to tune a controller from a random starting population to a controller with performance equal to a laboriously tuned manual case within 5 generations. In only one case did a manually tuned controller exceed the performance of the GA tuned controllers.

Further Research

There are extensions to the techniques described here that are needed to fully evaluate the utility of this technique to tuning in general. First, the restriction of the population to twenty individuals needs to be relaxed. Second, the operators need to be chosen randomly with parameters to determine how often the operators should occur in the random choice, similar to the practice in bit string based GAs. Third, in cases where the best individual from a previous generation may not evaluate to the same fitness value, the "fencing" of the individual to prevent loss of his data from the pool may be useful[7]. Fourth, the usefulness of using three (or more) parameters to describe a MBF should be explored. This would allow asymmetric MBFs. Such flexibility would be useful in permitting variable gain systems in which the placement of the center of adjacent MBFs determines the gain and the extent of the MBF is determined by the location of the center of the closest MBF to one of these. The effect of limiting the extent to half a support is to make the gain zero over that interval. Fifth, addition of search techniques that would allow local optimization of fitness before comparison could be useful. In a real number space, such techniques, subject to restrictions that will be applied to the resulting individuals (e.g., that the membership function's center must lie between the two ends of the support), should permit more rapid convergence of the GA search.

Summary

We have shown the applicability of real number genetic algorithms to the problem of automated tuning of membership functions for fuzzy controllers. The application tunes input membership functions which is a matching of control regions to the controller rather than the adjustment of gain of the controller. In a practical system, retention of the best individual may not assure monotonic convergence due to noise in the fitness function evaluation.

The GA search is most effective for tuning the controller in circumstances such as simulation when the failure of a system is inconsequential. For applications in which the stability of control must be maintained, such as automatic optimization of performance of an autonomous system, the applicability of a global search mechanism is questionable if the evaluation of fitness depends on controlling the device.

Acknowledgement

This work was sponsored in part by NASA Johnson Space Center under contract number NAS9-18527. We wish to thank Dr. Robert Lea for his support and encouragement of these efforts.

References

1. "Performance Evaluation of a Self-Tuning Fuzzy Controller," Walter C. Daugherity, Balaji Rathakrishnan, and John Yen, p. 389, Proceedings of the IEEE International Conference on Fuzzy Systems 1992 (FUZZ-IEEE 1992), San Diego, CA, March 8-12, 1992

2. "Neural Networks and Fuzzy Logic in Intelligent Control," Hamid Berenji, p.916, Proceedings of the 5th International Symposium on Intelligent Control 1990, Philadelphia, PA, September 5-7, 1990

3. "Fuzzy Logic and Genetic Algorithms in Time Varying Control Problems," C.L. Karr and D.A. Stanley, p. 285, 1991 Workshop Proceedings of the North American Fuzzy Information Processing Society, University of Missouri-Columbia, May 14-17, 1991

4. Handbook of Genetic Algorithms, ed. by Lawrence Davis, Ch. 5, (Van Nostrand Reinhold, New York, 1991)

5. Ron Maor and Yashvant Jani, p. 155, Working Notes of First International Workshop on Industrial Applications of Fuzzy Control and Intelligent Systems,Texas A&M University, College Station, TX, November 21-22, 1991

6. Julius Tou, Modern Control Theory, p.118 (McGraw-Hill Book Company, New York, 1964)

7. "Robot Path Planning Using a Genetic Algorithm," Timothy F. Cleghorn, Paul T. Baffes, and Lui Wang, p. 383, Proceedings of the Second Annual Workshop on Space Operations Automation and Robotics (SOAR '88) Dayton, OH, July 20-23, 1988 (NASA Conference Publication 3019)

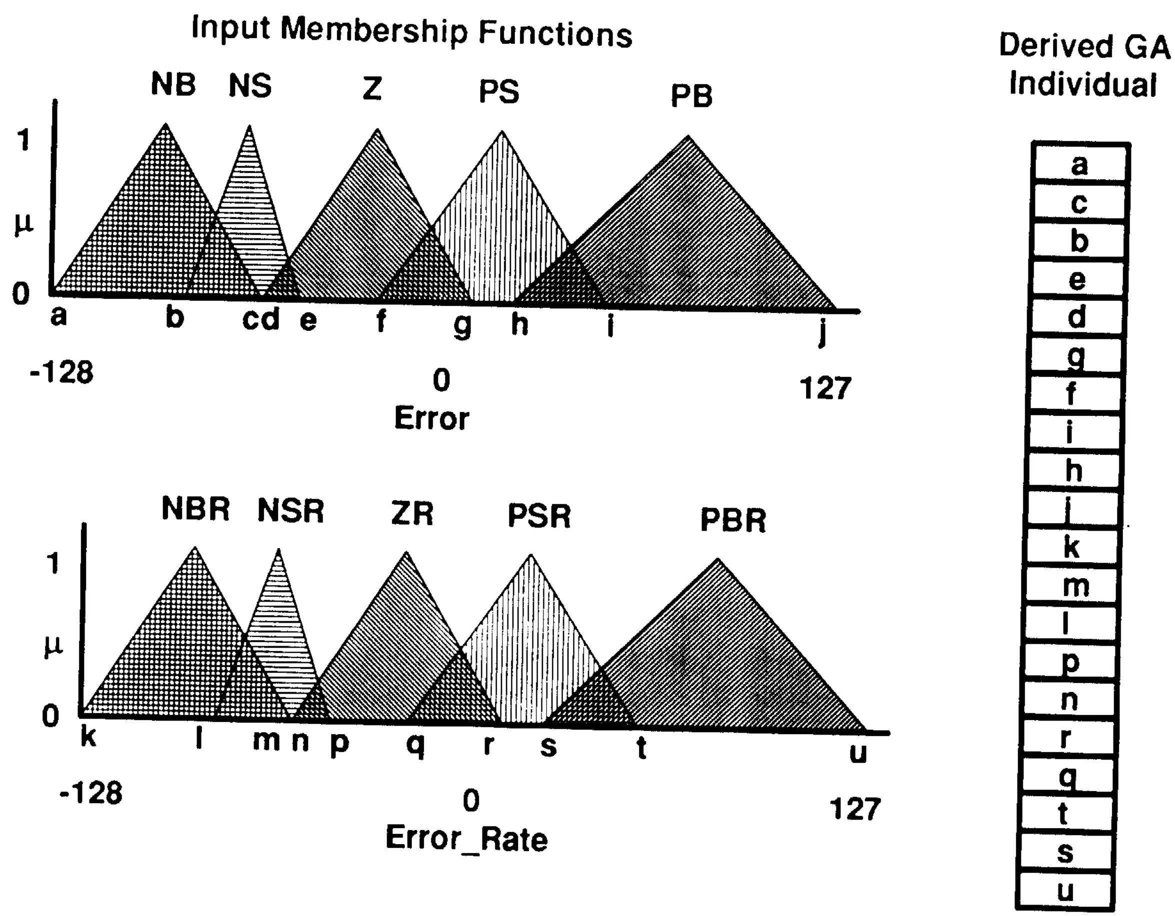
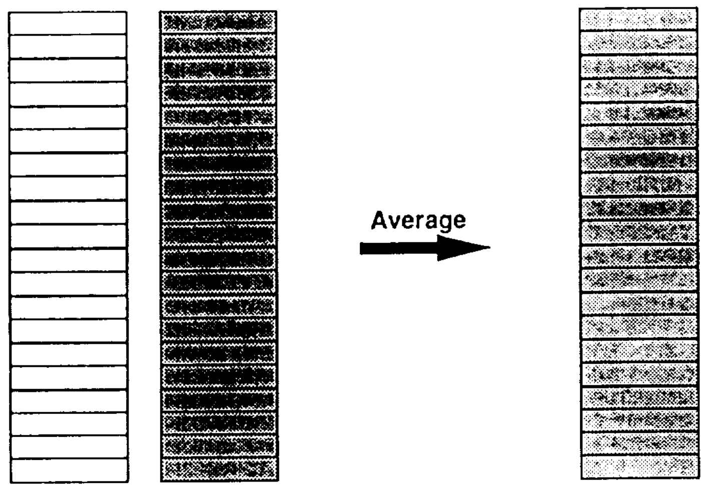


Fig. 1 Correspondence between input membership functions and the fields of an individual for the genetic algorithm tuner. The small letters represent integer values from the universe of discourse giving the ends of the support for each symmetrical membership function.

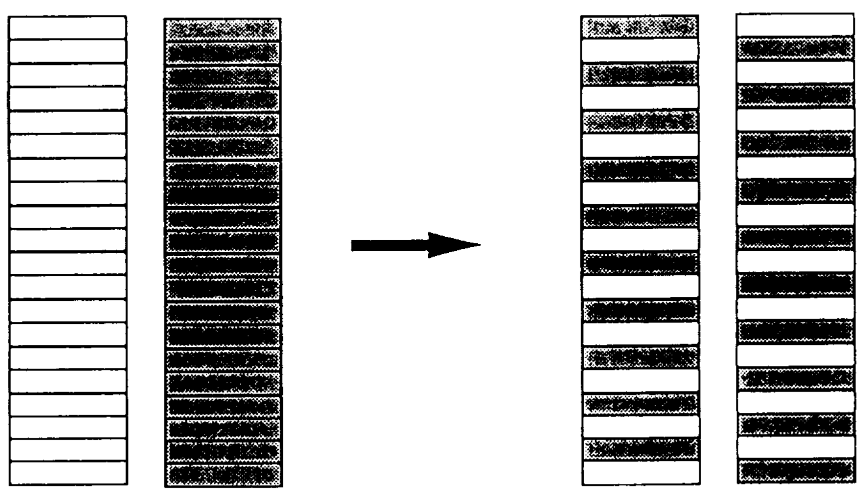
FUZZY GENETIC ALGORITHM MERGE



USE MOST FIT TO BREED BY PAIRWISE COMPETITION

(a)

REAL NUMBER GENETIC ALGORITHM CROSSOVER

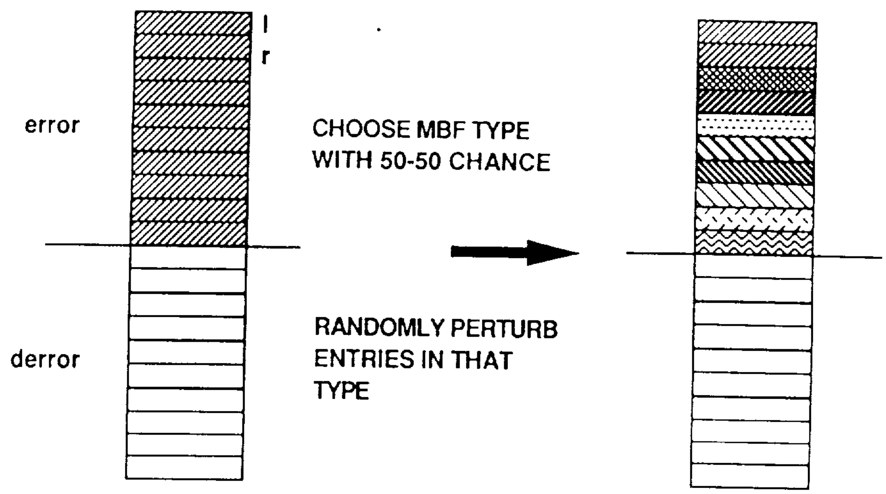


SMALL CHANGES ON MINIMUM SURFACE BRINGS CLOSER TO MAXIMUM

(b)

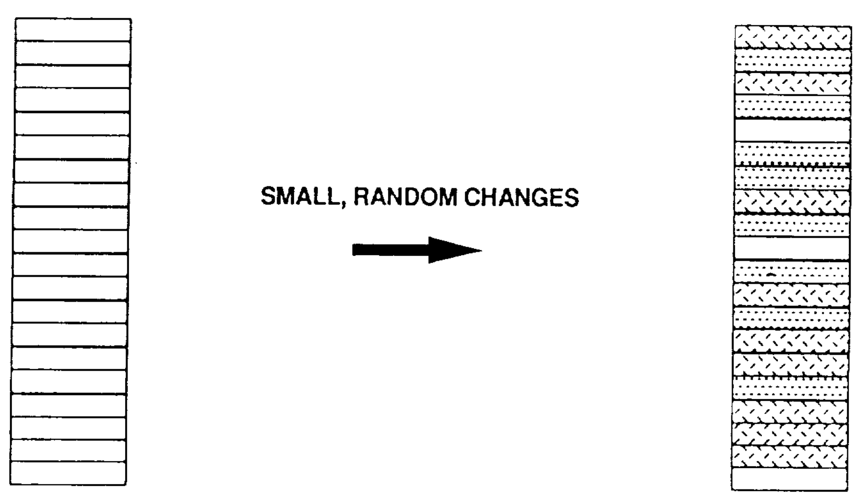
Fig. 2 Real number genetic operators defined for this tuning process

FUZZY GENETIC ALGORITHM MUTATE



(c)

REAL NUMBER GENETIC ALGORITHM CREEP



SMALL CHANGES ON MINIMUM SURFACE BRINGS CLOSER TO MAXIMUM

(d)

Fig. 2 (cont'd) Real number genetic operators

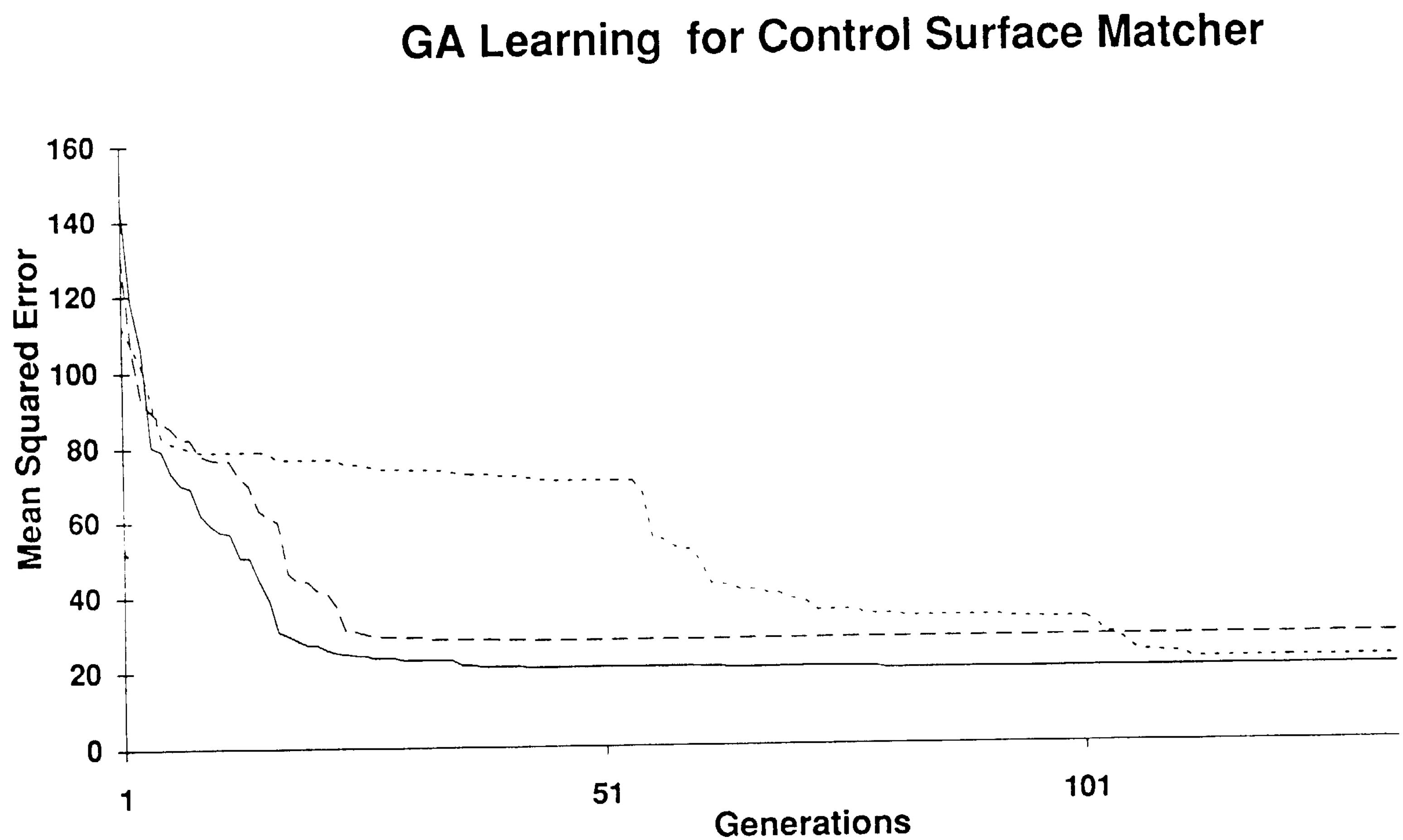


Fig.3 Results for control law learning problem

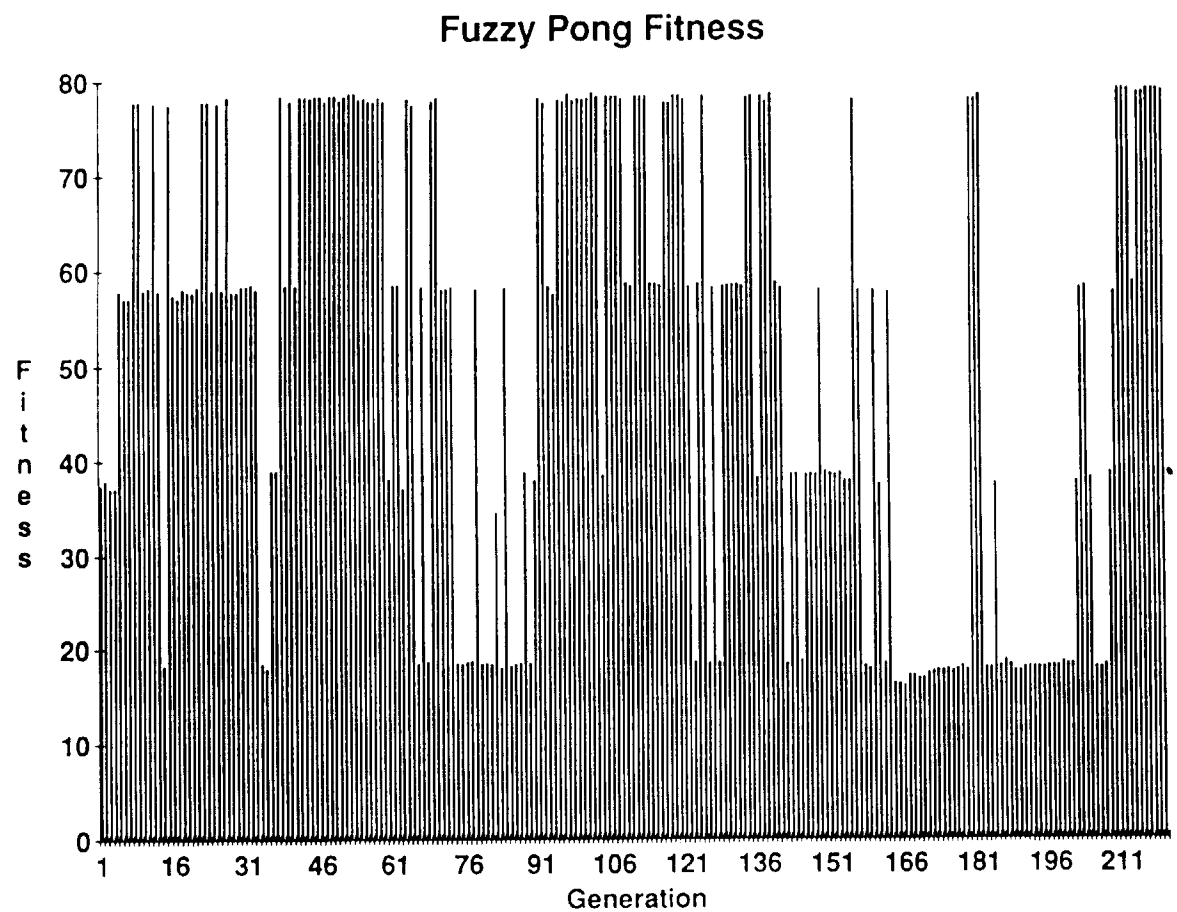


Fig.4 Results for air flow controller tuning