Proceedings of the World Congress on Engineering 2007 Vol I
WCE 2007, July 2 - 4, 2007, London, U.K.

An Adaptive Neural Network Fuzzy Inference Controller Using Predictive Evolutionary Tuning

Gordon K. Lee[1] and Edward Grant[2]

Abstract - **The design of intelligent controllers for nonlinear systems continues to be a challenging problem, particularly when the system is uncertain or the environment noisy. A nonparametric approach which has gained success is to employ a neural network to learn about the unknown plant and fuzzy inference to compensate for the uncertainty (GANFIS control). Inherent in the design of such controllers is the need to tune the weights of the GANFIS controller. Evolutionary learning has been suggested to tune the GANFIS parameters but a difficulty is selecting the parameters for tuning. Further, it is well known that proper selection of the fitness function has an important effect on system performance. In this paper, we integrate two design techniques that we have previously developed into a single generalized ANFIS controller: adaptive tuners to select critical evolutionary parameters and a predictive fitness function for measuring system performance. The adaptive tuners also employ this predictive fitness as part of selection process which is a new approach. Results show that this approach is a feasible method in designing GANFIS controllers using evolutionary tuning and predictive fitness.**

Keywords – **Evolutionary algorithms, fuzzy controllers, neural networks.**

I. INTRODUCTION

Many approaches have been suggested for nonlinear system control; the problem becomes more complex when uncertainties and noise are considered. One approach that has gained success when the system model is complex or uncertain relies on a non-parametric philosophy whereby a fuzzy block is used to handle uncertainties and imperfections while a neural network block addresses the underlying model dynamics. The classical adaptive neural-network based fuzzy inference system (ANFIS) approach [1] is such an architecture and generally provides good overall system performance; however, this approach may require the dynamics of the plant to be known, which may not always be available.

The advantage of using the ANFIS blocks is that the controller can be trained off-line to tune the premise and consequent control parameters and then used on-line for adaptive learning should there be changes in the plant.

One can employ a set of ANFIS blocks to form a generalized ANFIS that can approximate a nonlinear structure [2]. The generalized ANFIS (GANFIS) controller for the three membership case is shown in Figure 1.

In the GANFIS design, the idea is to represent the desired control action by a transfer function approximation as:

$$H(s) = \frac{\sum_{\ell=0}^{m} \beta_\ell s^\ell}{\sum_{\ell=0}^{n} \alpha_\ell s^\ell} = \sum_{j=1}^{m-n} \gamma_j s^j + \gamma_0 + \sum_{j=1}^{n} \frac{\delta_j}{s + a_j} \tag{1}$$

where m is the order of the numerator and n is the order of the denominator of the transfer function approximation to a nonlinear function $f(e)$.

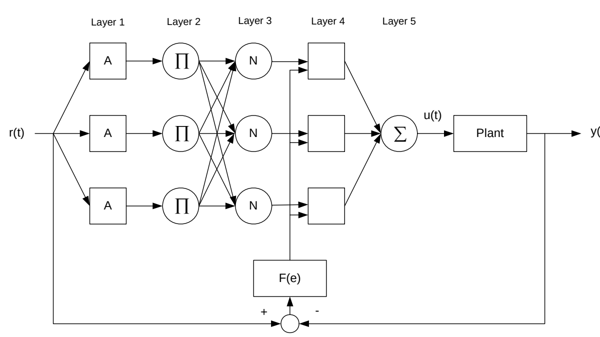

Figure 1: The Generalized ANFIS Controller

One can show that the control law is:

$$u(s) = \sum_{i=1}^{n_r} \overline{\varpi}_i [\sum_{j=1}^{m-n} p_{i,j} s^j E(s) + p_{i,0} E(s) + \sum_{j \neq 1}^{m} \frac{p_{i,j}}{(s + a_{j})} E(s)] \tag{2}$$

where $p_{i,j}$ are the consequent parameters of the ANFIS blocks.

---

[1]Dept. of Electrical & Computer Engineering; San Diego State University; 5500 Campanile Drive; San Diego, California, 92182-1326. email: glee@kahuna.sdsu.edu

[2]Department of Electrical & Computer Engineering; Center for Robotics and Intelligent Machines; North Carolina State University; Raleigh, North Carolina 27695. email: egrant@ncsu.edu

An important operation in designing the GANFIS controller as well as other versions of an ANFIS-based architecture is selecting the consequent and premise parameters. Gradient techniques and/or estimation are traditionally employed to find these parameters for off-line or on-line implementation but may lead to unstable solutions or slow convergence. Instead we employ evolutionary algorithms in selecting the GANFIS parameters; the claim is that by using the evolutionary process, the GANFIS parameters can be tuned on-line through a more stable structure [3]. The design issue then is to select the evolutionary parameters and in particular the mutation and crossover probabilities which have an impact on the evolutionary process. In this paper, methods previously developed by the authors are modified and enhanced in tuning these key parameters.

One can show that the selection of the fitness function also plays an important role in the convergence properties of an evolutionary controller [4]. While traditional approaches rely only on current state or output information in assessing the performance metric for each chromosome, the approach here is to employ *prediction* so that past and current information may be used in assessing the fitness of each chromosome. The predictive fitness function for each chromosome is then integrated into the tuning process for the mutation and crossover probabilities.

Section II provides a summary of the tuning process while Section III describes the predictive fitness function. By integrating the two methods, we show the attractiveness of the resulting control algorithm through several examples. Section IV presents a nonlinear system example, subjected to noise and parameter variation. Results show the feasibility of employing the GANFIS controller to nonlinear systems with noise or system variation.

II. TUNING THE EVOLUTIONARY PARAMETERS

Consider the generalized ANFIS (GANFIS) controller of Figure 1. In order to design the controller, premise and consequent parameters must be selected for the fuzzy component and the neural network part. In this paper, we use evolutionary methods in selecting these parameters as shown in Figure 2.

In most evolutionary approaches, genetic searching is used which consists of a finite repetition of three steps at each generation: selection of the parent chromosomes for the next generation (usually an elitist selection for a percentage of the next generation), recombination using crossover and mutation operations [5], and a fitness function that describes the *goodness* of individual members of each generation.

In [6], Rajapakse and others employ evolutionary algorithms to tune fuzzy logic controllers, but then use an on-line neural network model of the process as a separate block. We use the evolutionary learning as *part* of the adaptive neural network fuzzy inference controller, rather than separate each operation (evolutionary tuning, fuzzy logic controller, neural network model of the plant) in the design process. Further, the parameters of the evolutionary learning operation (population size, mutation operator, cross-over operator and fitness function) are adaptively changed based upon the overall system performance measure.

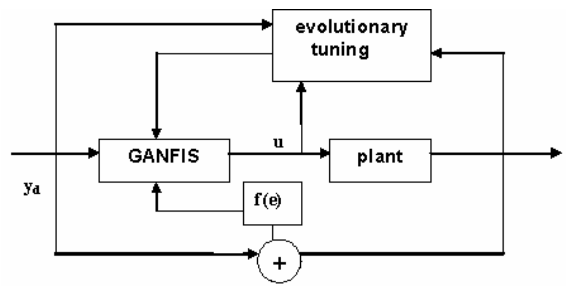

Figure 2: GANFIS with Evolutionary Tuner

The evolutionary module runs several generations of candidate premise and consequence parameter chromosomes and selects the best set, according to a fitness function of the form:

$$F = \sum_{i=1}^{n_r} e_i^2 \tag{3}$$

where the error is the difference between the desired output and the actual output.

The mutation and crossover rates are two important evolutionary parameters and are typically statically set through trial-and-error in classical evolutionary algorithms [7]. However, a parameter that is optimal during the initial stages of a search may not be effective in later stages of the evolutionary process [8]. Hence, adaptively tuning the parameters during a search process would enhance the convergence properties of the evolutionary algorithm and therefore should improve control performance. Pedrycz [9] states that the mutation rate and the crossover rate can be experimentally adjusted from results from a series of observations of past simulation and provides a method using Fuzzy meta-rules.

In [10], the effects of the crossover rate P_c and mutation rate P_m to the maximum fitness and average fitness values are discussed. We know that the larger the error is between the fitness values of two individuals, the stronger is the degree of the mutation rate and crossover rate. Hence we have developed a tuning process for adaptively changing the mutation and crossover rates.

In [11] and [12], the mutation and crossover rate are tuned using different functions of the current fitness values. For example, one may select:

$$p_C = \frac{|f_i(n) - \bar{f}(n)|}{\bar{f}(n)} \tag{4}$$

$$p_m = \frac{[\bar{f}(n) - f_i(n)]}{2 * \bar{f}(n)} \tag{5}$$

where $f_i(n)$ is the current fitness function associated with the chromosome that requires the crossover or mutation operation and $\bar{f}(n)$ is the maximum fitness function for generation n. Notice that (4)-(5) do not require any *a priori* knowledge in selecting the probabilistic rates; rather the estimators simply use current fitness values at each generation. In this paper, we extend the idea to include past fitness function values in the tuning mechanism which is discussed in the next section.

III. MODIFYING THE FITNESS FUNCTION

Classical evolutionary algorithms construct a fitness function as in (2) based upon *current* information, in order to assess the performance of each chromosome in a population. This assessment is then used in selecting the next generation of chromosomes, thereby improving the performance of a system over time. Note that this fitness function is *static* in that the value of the present fitness is dependent only upon present information. In order to improve the fitness value, a fitness function based upon current and *past* values is employed here.

Sankar [10] considers modifying (2) as:

$$f(n) = \beta_0 \cdot g(n) + \sum \beta_i \cdot a_i \tag{6}$$

where g(n) is the standard fitness of an individual based upon current information, a_i is the fitness function of its n ancestors, e.g., a_i = f(n-i) and β_i are weighting factors, $\beta_0 > \beta_j$ for j=1…M. In [13] a heuristic fitness function is developed that is not only based upon past and current information but also on *future* knowledge:

$$f(n) = g(n) + h(n) + q(n) \tag{7}$$

where h(n) is the fitness function component based upon some historical information related to a *predicted* target and q(n) is a heuristic function based upon *expected future* knowledge. We differentiate between prediction, based upon deterministic historical data and future heuristics, based upon probabilistic information.

In [13], we show improvements in the performance of the fitness function values over time, when selecting a forward predictor and heuristic function based upon expected future knowledge. That is, let:

$$\begin{aligned} &h(n) = -\delta\{g(n) - \hat{f}(n \mid F_{n-1})\} \\ &\hat{f}(n \mid F_{n-1}) = \theta(f(n-1),\ldots,f(n-M)) = \sum_{k=1}^{M} w_k f(n-k) \\ &q(n) = \lambda / \hat{f}_c(n) \end{aligned} \tag{8}$$

where n is the current generation, $\hat{f}(n \mid F_{n-1})$ is the predicted fitness function value, based upon n-1 past data sets, δ and λ are weights, and $\hat{f}_c(n)$ is the heuristic part, based upon a *future* expectation of the fitness function values.

One can show that a linear forward predictor can be used in generating h(n) while a probabilistic model based upon hypothesis testing and Bayesian approximation can be used to generate q(n) [13]. Thus the heuristic fitness function becomes:

$$f(n) = g(n) + h(n) + q(n) = g(n) - \delta\{g(n) - \hat{f}(n \mid F_{n-1})\} + \lambda/\hat{f}_c(n) \tag{9}$$

Results in [13] using a Khepera robot model show improvements in the fitness function values over time when compared to classical fitness functions.

In this paper, we wish to improve the performance of the evolutionary tuners for the mutation and crossover rates. To do this, we propose to integrate the first two components of the fitness function (9) into (4) and (5). Employing future heuristics into the tuning process is an area of future research. Thus, the fitness function that we employ in (4) and (5) is:

$$f(n) = g(n) + -\delta\{g(n) - \sum_{k=1}^{M} w_k f(n-k)\} \tag{10}$$

In summary then, the parameters for the GANFIS controller (premise and consequent variables) in Figure 2 are tuned using evolutionary methods. The mutation and crossover rates for the evolutionary algorithms are selected adaptively using (4) and (5) but with the fitness function of (10) which captures past and current information in the tuning process. We claim that this modification of the selection of p_c and p_m improves system performance, even under noise and parameter variation. This is illustrated in an example.

IV. EXAMPLE

Consider the nonlinear system [14]:

$$\begin{aligned} &\dot{x}_1(t) = x_2^3(t) + u(t) \\ &\dot{x}_2(t) = u(t) \\ &y(t) = x_1(t) \end{aligned} \tag{11}$$

where y(t) is the output and u(t) is the control input. It is desired that the output track the function:

$$y_d(t) = \sin 2t * e^{-1.5t} \tag{12}$$

Results using the GANFIS controller with a gradient method for selecting the premise parameters and the Kalman Filter method for choosing the consequent parameters are detailed in [2]. Further, employing evolutionary learning, i.e., (4) and (5), in the tuning of the GANFIS parameters is investigated in [12] where only current fitness function values are employed.

The parameters were tuned at each generation using an elitism selection, recombination and fitness evaluation. The control parameters for our tests were selected as follows: the population size of 20, six membership functions in the ANFIS block, four bits for each chromosome and an elitist selection saving half the population.

The crossover and mutation probability rates have a large influence on the performance of the evolutionary algorithms. Mutation is used to create diversity in the population and thus avoid trapping in local minima. Too high a value can cause the evolutionary algorithm into a random search while too low a value can cause it to be trapped in local minima. Crossover on the other hand is used to create two individuals (children) from two existing individuals (parents) picked from the current population based on selection criteria. This becomes more evident when we have noise in the system or parameter variation [15].

Hence we wish to evaluate the proposed algorithm (EVGANFIS) for several cases of the nonlinear system. First we tested the case when the system is subjected to 20db noise (but no parameter variation). Using the proposed algorithm results in the trajectory shown in Figure 3. We note that the controller compensates for the noise and follows the desired trajectory.

In order to investigate the effects of noise on the proposed approach, we increased the noise to 40db. The results are shown in Figure 4. While there are relatively large initial deviations as expected, the response begins to track the desired output after 1.5 seconds and reaches the desired steady state value.

Next we wish to investigate the effects of plant parameter variation using EVGANFIS. Eight case have been studied as summarized in Table I.

Table I: Case Studies in Parameter Variation

Case #	*Coefficient*	*Time of Change*
1	0.1	2sec
2	0.01	2 sec
3	10	2 sec
4	100	2 sec
5	0.1	0.5 sec
6	0.01	0.5 sec
7	10	0.5 sec
8	100	0.5 sec

The coefficient represents the new value of the coefficient associated with the nonlinear state in the dynamics (11), i.e., from unity while the third column defines when the variation occurred. The input noise remains at 20 db. Figures 5-12 provide the results which illustrate that even with plant parameter variation, the EVGANFIS can compensate for this effect and track the desired output trajectory.

V. CONCLUSIONS

The crossover probability determines the frequency of the crossover operation which in turn helps to find candidate solutions. A low value can slow down the rate of convergence while a high value can make the search rotate around one solution. The mutation probability controls the diversity in the population and hence a high number brings diversity but at the same time causes instability while a low value causes the search to be trapped in local minima.

Further the selection of the fitness function determines what chromosomes are kept for the next generation, how the new population is produced based upon given parent pairs and thus impacts convergence of the overall algorithm.

In this paper, evolutionary tuning using adaptive mutation and crossover rates, based upon a fitness function that employs past and present values, show promising results, even when the system is subjected to noise and plant parameter variation. In the future, we plan to extend the approach to tuners which contain past and current and future fitness function information. These extensions may enhance the adaptivity feature of the evolutionary tuner for the GANFIS controller.

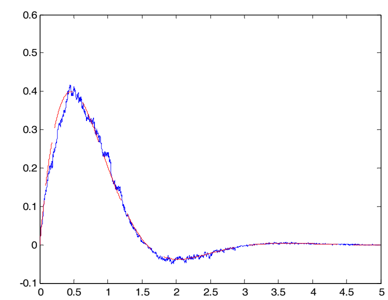

Figure 3: Desired Output versus Actual Output using EVGANFIS for 20 db Input Noise

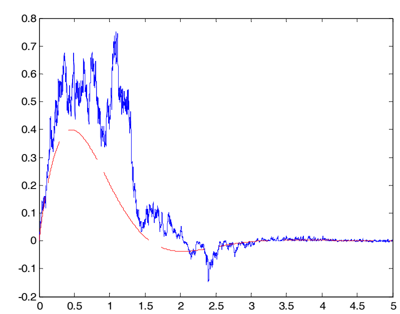

Figure 4: Desired Output versus Actual Output using EVGANFIS for 40 db Input Noise

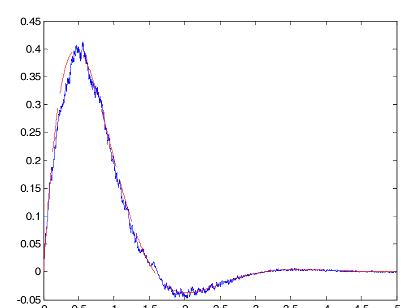

Figure 5: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 1

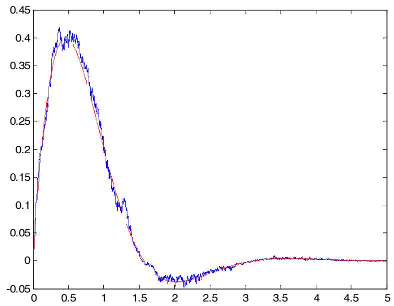

Figure 6: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 2

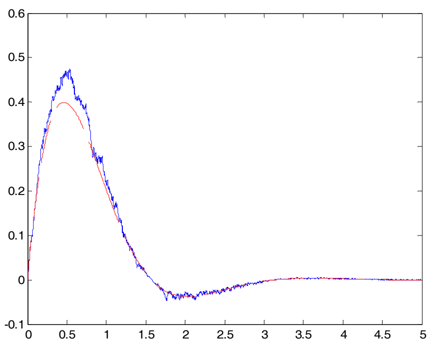

Figure 7: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 3

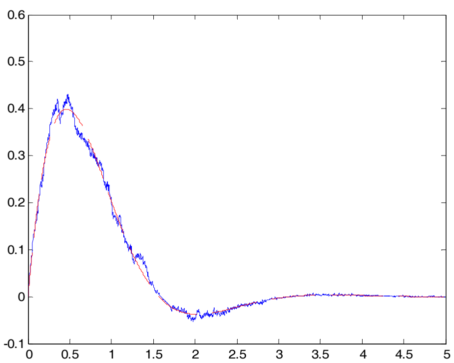

Figure 8: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 4

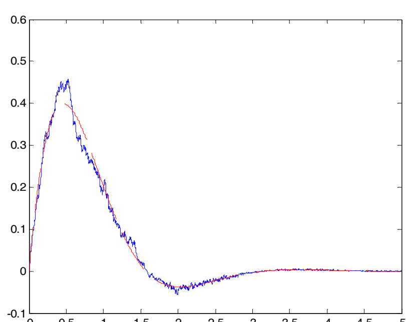

Figure 9: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 5

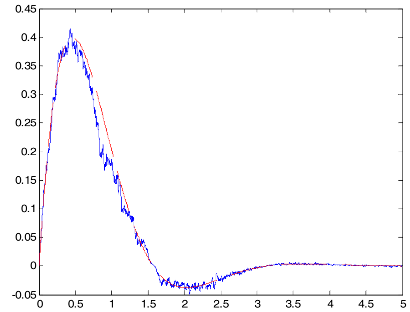

Figure 10: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 6

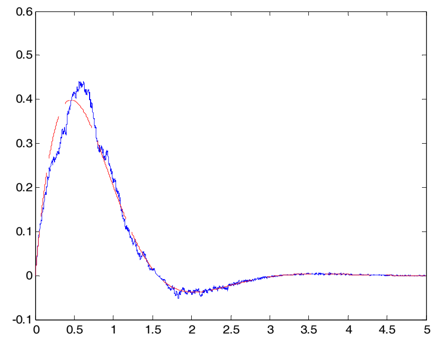

Figure 11: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 7

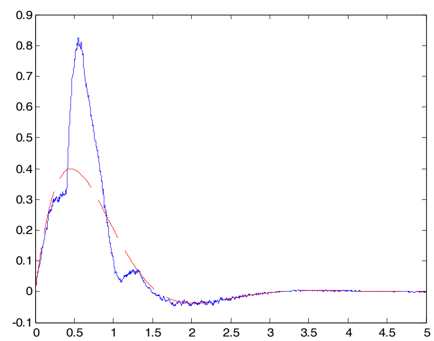

Figure 12: Desired Output versus Actual Output using EVGANFIS for Plant Variation Case 8

REFERENCES

[1] Jang, J.-S.R.,"ANFIS: Adaptive-Network based fuzzy inference system", *IEEE Transactions on Systems, Man and Cybernetics*", Vol. 23, No. 3, pp. 665-685, 1993.

[2] Lee, G. and Grant, E., "A generalized adaptive neural network fuzzy inference structure for nonlinear control", Proc. of the 14th ISCA International Conference on Computers in Industry and Engineering, Las Vegas, 2001.

[3] Lee, H.K. and Lee, G., "Convergence properties of genetic algorithms", Proc. of the ISCA Int'l Conference on Computers and Their Applications, Seattle, 2004.

[4] Bhat, S. and Lee, G., "An adaptive fuzzy inference fitness function for evolutionary robot learning", Proc. of the ISCA Int'l Conference on Computer Applications in Industry and Engineering, Honolulu, 2005.

[5] Nolfi, S. and Floreano, D., *Evolutionary Robotics, The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press/Branford Books, Cambridge, 2000.

[6] Rajapakse, K., Furuta, K., and Kondo, S., "Evolutionary learning of fuzzy logic controllers and their adaptation through perpetual evolution, *IEEE Trans. On Fuzzy Systems*, Vol. 10, No. 3, June 2002.

[7] Kothari, J., Grant, E. and Lee, G., "Tuning an adaptive neural network fuzzy inference controller using evolutionary learning", Proc. Of the ISCA Int'l Conference on Computer Applications in Industry and Engineering, Orlando, 2004.

[8] Uchibe, E., Yanase, M. and Asada, M., "Behavior generation for a mobile robot based on an adaptive fitness function", *Journal of Robotics and Autonomous Systems,* Vol. 40, Nos. 2-3, pp. 69-77, Aug. 2002.

[9] Pedrycz, W., *Computational Intelligence: An Introduction*, CRC Press, 1997.

[10] Pal, S.K. and Wang, P.P., *Genetic Algorithms for Pattern Recognition*, CRC Press, 1996.

[11] Tang, P., Bhat, S. Kothari, J. and Lee, G., "A modified evolutionary algorithm using a heuristic evaluation function with mutation rate and crossover rate tuning", Proc. Of the ISCA Int'l Conference on Computers and Their Applications, New Orleans, 2005.

[12] Lee, G. and Grant, E., "On parameter selection for an adaptive fuzzy neural network inference controller using evolutionary tuning", Proc. of the ISCA Int'l Conference on Computers and Their Applications, Seattle, 2006.

[13] Tang, P and Lee, G., "An adaptive fitness function for evolutionary algorithms using heuristics and prediction", Proc. of the World Automation Congress, ISSCI, Budapest, 2006.

[14] Slotine, J.-J.E., Tracking control of nonlinear systems using sliding surfaces", Ph.D. Dissertation, Massachusetts Institute of Technology, 1983.

[15] Kothari, J. and Lee, G., "Investigation of the effects of noise and plant parameter variations for an adaptive neuro fuzzy inference controller with evolutionary tuning", Proc. of the ISCA Int'l Conference on Computer Applications in Industry and Engineering, Honolulu, 2005.