Genetic-Based Reinforcement Learning For Fuzzy Logic Control Systems

Kuo-Tsai Lee[1], Kuang-Tsang Jean[1,2] and Yung-Yaw Chen[1]

[1]Corresponse Address: Lab. 202, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C..
[2]Telecommunication Laboratories, Ministry of Transportations & Communication, Taiwan, R.O.C..

ABSTRACT

This paper proposes a genetic-based reinforcement learning for fuzzy logic control systems (GR-FLCS) to solve reinforcement learning problems. The proposed GR-FLCS is constructed by integrating a real-coded genetic algorithm with a time accumulator as the fitness evaluator, a success criterion, a fuzzy logic controller (FLC), and a parameter adapter for the FLC. In this simple but powerful architecture, restrictions, usually met in reinforcement learning for FLCs, can be taken off completely. They are, the FLC must be implemented by a neuronlike network, the shapes of the membership functions in the FLC must be in some form, e.g., bell-shaped, the fuzzy operators must be modified, or only the consequent part of the rule base in FLC can be learned. Finally, the applicability and efficiency of GR-FLCS are demonstrated by an simulation example of the cart-pole balancing problem.

1. INTRODUCTION

Starting with the self-organizing control (SOC) techniques of Procyk and Mamdani, many efforts have been done in developing fuzzy logic controllers which can learn from experiences. Among these efforts, some are to solve reinforcement learning problems which, in contrast to supervised learning, is to learn a fuzzy logic controller even when only delayed and weak information, such as binary failure signals, is available. On the basis of Barto, Sutton, and Andersons' neuronlike adaptive elements, C.C. Lee [1] proposed his paradigm in which individual rules engaged in the problem solving process are considered boxes in BOXES system to be blamed or rewarded. Therefore, all the possible rule are listed and only the consequent part of them are to be learned. Berenji [2] and C. T. Lin [3] employed the adaptive heuristic critic (AHC) algorithm, introduced by Sutton, in their fuzzy structures. For doing this, Berenji used one network- based fuzzy logic controller to substitute the action network, and C. T. Lin used two with one as an action network and the other as an evaluation network.

In the last few years, research devoted to search and optimization has significantly grown. Genetic algorithms (GAs), developed by Holland, his colleagues and his students, are more and more valued in this topic. Based on the mechanics of natural selection and natural genetics, GAs have been proved half by theory and half by experiment to be superior to hill-climbing methods in multimodular and non-derivative function cases and to random walk in efficiency and efficacy [4]. Recently, GAs have been successfully used in various areas. Odetayo and McGregor [5] first introduced GAs to solve reinforcement learning problems. In their algorithm, state space is partitioned into non-overlapping regions, as in BOXES system. Each allele value on a string recommends a left-push or a right-push if the system state falls into the corresponding box. Generation of candidate control rules continues to evolve until an eligible control rule is found. Whitely, *etc.* [6] employed their real-coded GA, the GENITOR, as the evaluation network in AHC, whereas the original action network is preserved. An allele value itself in a string is a weight in the action network.

In this paper, a simple but powerful architecture, genetic-based reinforcement learning for fuzzy logic control systems (GR-FLCS), is proposed. With the assistance of GAs, GR-FLCS is able to solve reinforcement learning problems by adapting the FLCS without any restrictions. These restrictions may be: the FLC must be implemented by a neuronlike network, the shapes of the membership functions in the FLC must be in some form, e.g., bell-shaped, the fuzzy operators must be modified, or only the consequent part of the rule base in FLC can be learned.

The organization of the paper is as follows. The architecture of GR-FLCS is briefly introduced in section 2. The example of the cart-pole balancing is demonstrated in section 3. Finally, the conclusion is given in the last section.

2. OUR PARADIGM: GR-FLCS

The architecture of GR-FLCS is shown in fig. 1. The proposed GR-FLCS is constructed by integrating a real-coded genetic algorithm, with a time accumulator as the fitness evaluator, a success criterion, a fuzzy logic controller (FLC), and a parameter adapter for the FLC.

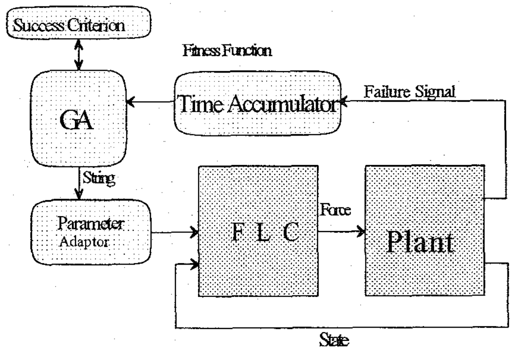


Fig. 1 The architecture of the GR-FLCS.

In our paradigm, the fitness evaluator is in the form of a time accumulator which counts the time steps before the system state falls into the failure region, as in Whiteley's method [6]. The parameters, constituting scaling factors and anchoring points of the membership functions in the FLC, are concatenated into a real-valued string in the environment of a real-coded genetic algorithm, the variable-based genetic algorithm (VGA) [7]. These strings are then forwarded to the parameter adapter to modify the parameters in the fuzzy logic controller. The VGA reinforces the good candidate solutions, the strings, on the basis of their performance measured by the time accumulator. Through crossover and mutation operators, a new generation is yielded. The searching is ongoing till a criterion, which is defined in the success criterion, is met.

The initials conditions are set randomly for each string of a population, which makes sense for the reinforcement learning problems. However, in this situation, a better fuzzy logic controller that receives a poor starting condition will be ranked lower than a worse one that receives a good starting condition, which makes our fitness function very "noisy". Our strategy is to keep the population size small to lessen the influence [6] [8] of the great disturbance. Furthermore, another problem arises. It is that the old criterion in reinforcement learning, which is defined as that the learning is considered to be successful if the controller is able to maintain the system over a given time steps, isn't appropriate. A bad fuzzy logic controller that can not deal with most of the starting conditions may meet the criterion from a fairly good starting condition. Therefore, it is necessary for us to set up a new success criterion.

Our policy is that the learning is considered to be successful and stopped only if some candidate solution can maintain the system more than the threshold time steps for a given number of consequent generations. Thus, whenever a fuzzy logic controller meets the old criterion, it is not considered the one we are searching for in haste, whereas it will be challenged by other initial conditions. This strategy is implemented by the elitism in VGA. The first best candidate solution is preserved and take the first position in the new generation. If it is the eligible solution, it will still take the first position for the next consequent generations, if not, some other candidate solution will replace it. Meanwhile, the whole population is evolving through operators of the VGA. This simple policy allows the fuzzy logic controller visit more of the state space, and find out the solution to the problem over all possible states.

3.THE CART-POLE BALANCING PROBLEM

The cart-pole system, which is illustrated in fig. 2, was simulated using equations as follows by euler method with time step 20ms,

$$\ddot{\theta}=\frac{g\sin\theta+\cos\theta\left[\frac{-f-ml\ddot{\theta}\sin\theta+\mu_c sgn(\dot{x})}{m_c+m}\right]-\frac{\mu_p\dot{\theta}}{ml}}{l\left[\frac{4}{3}-\frac{m\cos^2\theta}{m_c+m}\right]}$$

$$\ddot{x}=\frac{f+ml\left[\theta^2\sin\theta-\ddot{\theta}\cos\theta\right]-\mu_c sgn(x)}{m_c+m},$$

where x is the horizontal position of the cart, $\dot{x}$ is the velocity of the cart, θ is the angle of the pole with respect to the vertical line, $\theta$is the angle of the pole with respect to the vertical line, is the angle velocity of the pole, and f is the force applied to the cart. All the values of the parameters and their meanings in our simulation are: mass of the cart: m = 1 kg, mass of the pole: m_c = 0.1 kg, half length of the pole = 0.5 m, gravitational acceleration: g =

9.8 m / s^2 ,coefficient of friction of cart on track: μ_c = 0.0005, coefficient of friction of pole on cart: μ_p = 0.000002.

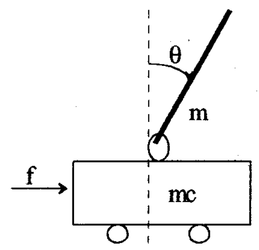


Fig. 2 The cart-pole system.

The primary control tasks are to keep the pole vertically balanced and to keep the cart within the rail track boundaries. A failure happens whenever $|x| \geq 2.4m$ or $|\theta| \geq 15^\circ$. The force applied to the cart-pole system can be any value between -20 newtons to +20 newtons. The initial state for each candidate solution is random in the region of $|x| \leq 2.0m$ and $|\theta| \leq 12^\circ$. Whenever some string is able to maintain the system over 10000 time steps and is preserved over 8 consequent generations, it is considered a eligible solution and the learning is stopped

Thirteen rules [2] are assumed to be available as in Table 1, where x_1 is θ, x_2 is $\dot{\theta}$, x_3 is x, and x_4 is $\dot{x}$. We also assumed that the anchoring points of a membership function are the peak points of its two neighboring ones, which makes the inference output curve smooth [9], and the labels are symmetric to the y-axis. The inference method we used is the so-called weighted average method. Therefore, a string in the environment of the VGA composes of a_1~a_7 , and N1~N4 as in the fig. 3. We set the base mutation rate M_p ,variance, and population size n, to be 0.5, 0.8, and 10, respectively. The small population helps to improve the performance of genetic algorithms in noisy environment , which has be mentioned. An eligible solution came out in the 67th generation. Fig. 4 shows that this solution can handle initial conditions randomly generated.

Here we made some assumptions to make the whole demonstration easier. Actually, we can learn the FLC well without these assumptions at the price of longer simulation time.

4. CONCLUSIONS

Integrating FLC and GAs into it, the genetic-based reinforcement learning for fuzzy logic control systems (GR-FLCS) gives more freedom for the FLC when prior knowledge of the experts or experienced operators is adopted in a system. Furthermore, it has the merits both of the FLC and GAs, which makes the learning easy but powerful.

Rule 1: If x_1 is PO1 and x_2 is PO2 then u is PL.
Rule 2: If x_1 is PO1 and x_2 is ZE2 then u is PM
Rule 3: If x_1 is PO1 and x_2 is NE2 then u is ZE
Rule 4: If x_1 is ZE1 and x_2 is PO2 then u is PS
Rule 5: If x_1 is ZE1 and x_2 is ZE2 then u is ZE
Rule 6: If x_1 is ZE1 and x_2 is NE2 then u is NS
Rule 7: If x_1 is NE1 and x_2 is PO2 then u is ZE
Rule 8: If x_1 is NE1 and x_2 is ZE2 then u is NM
Rule 9: If x_1 is NE1 and x_2 is NE2 then u is NL
Rule 10: If x_1 is VS1 and x_2 is VS2 and x_3 is PO3 and x_4 is PO4 then u is PS
Rule 11: If x_1 is VS1 and x_2 is VS2 and x_3 is PO3 and x_4 is PS4 then u is PVS
Rule 12: If x_1 is VS1 and x_2 is VS2 and x_3 is NE3 and x_4 is NE4 then u is NS
Rule 13: If x_1 is VS1 and x_2 is VS2 and x_3 is NE3 and x_4 is NS4 then u is NVS

Table 1 The thirteen rules for balancing cart-pole system.

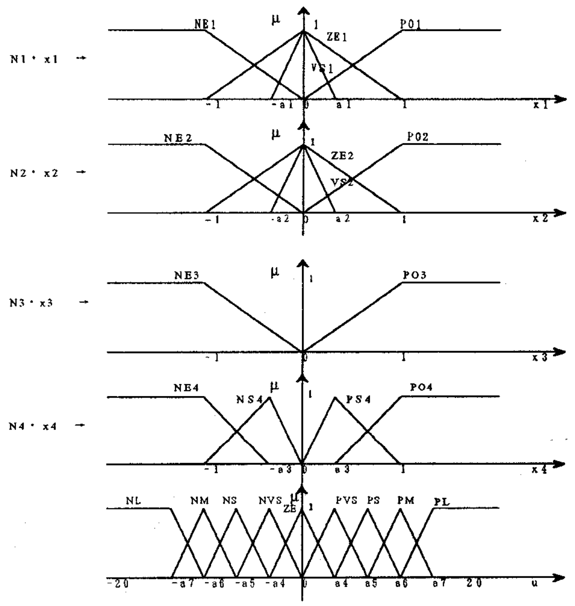
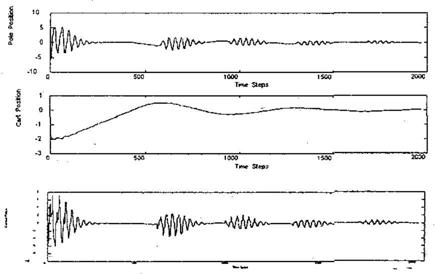
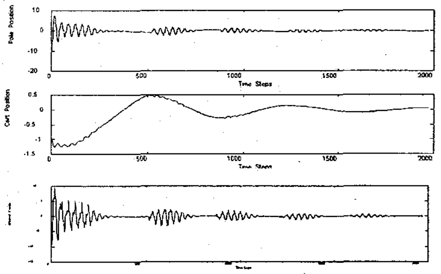


Fig. 3 The definitions of the labels in Table 1. N1~N4, and a_1~a_7 are concatenated to be a string in VGA.

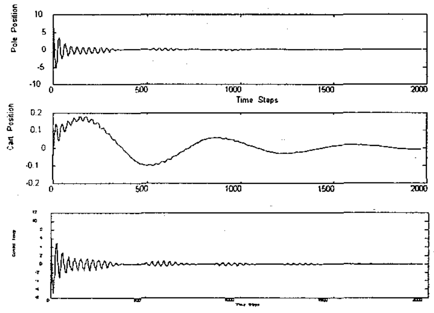
a.



b.



c.



d.

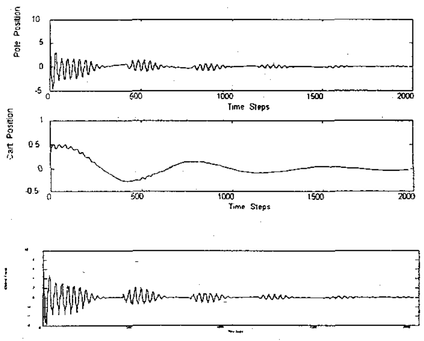


Fig. 4 Four initial conditions were generated randomly to test the output solution.

a. Initial condition = [-5.9085° 0 -1.8756m 0]

b. Initial condition = [-10.7813 0 -0.9790m 0]

c. Initial condition = [-7.3726 0 -0.0495m 0]

d. Initial condition = [-6.2683 0 0.3570m 0]

The figures of force look like the figures of pole position in shape, which is because 9 of the 13 rules were used for balancing the pole, only 4 were for balancing the cart.

REFERENCES

[1] C .C. Lee, "A Self-Learning Rule-Based Controller Employing Approximate Reasoning and Neural Net Concepts", *International Journal of Intelligent Systems*, Vol. 6, pp. 71-93, 1991.

[2] Hamid R. Berenji, "Learning and Tuning Fuzzy Logic Controllers Through Reinforcements", IEEE Trans. on Neural Networks, vol. 3, No. 5,September, 1992.

[3] Chin-Teng Ling, C. S. George Lee, "Reinforcement Structure/Parameter Learning for Neural-Network- Based Fuzzy Logic Control Systems", *IEEE Trans. on Fuzzy Systems*, vol. 2, no. 1, Feb. , 1994.

[4] Goldberg, David E. *Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley*, 1989.

[5] M. O. Odetayo, D. R. McGregor, "Genetic Algorithm for Inducing control rules for a dynamic systems," in *Pro. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 177-182.

[6] Darrell Whitley, Stephan Dominic, Rajarshi Das, Charles W. Anderson, "Genetic Reinforcement Learning for Neurocontrol Problems", *Machine Learning*, 13, 1993, pp. 259-284.

[7] K. T. Jean, Y. Y. Chen, "A Variable-Based Genetic Algorithm", *IEEE Int.* Conf. *Syst., Man, Cybern.*, 1994.

[8] Fitzatrick, J., Grefenstette,. "Genetic algorithms in noisy environment", *Machine Learning*, 3(2-3), pp. 101-120, 1988.

[9] Dimiter Driankov, Hans Hellendoorn, Michael Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, 1993.