Fifth IFSA World Congress (1993), 1029-1032

Comparisons of Some Reinforcement Self-Learning Controllers by Cell-to-Cell Mapping

Chi-Fong Pong Yung-Yaw Chen Te-Son Kuo
Electrical Engineering Department
National Taiwan University
Taipei, Taiwan, R. O. C.

Abstract

The construction of the rulebase of a fuzzy controller is usually difficult because experts' knowledge is often hard to derive. To remedy such a problem, a number of self-learning schemes for rulebase formulations were proposed. One of the popular approaches is the reinforcement learning. Many successful examples employing such an idea were proposed and claimed to be with good results in the literature. The purpose of this paper is to discuss and make comparisons between some of the related work in order to provide a better picture regarding their performances. A numerical algorithm for the analysis of nonlinear as well as fuzzy dynamic systems, the Cell-to-Cell Mapping, is used. The analytical results reveals the true behavior of the learning schemes.

1. Introduction

Since the first successful application of fuzzy control in steam engine in 1974, fuzzy logic control has become one of the most active and fruitful areas for research in the applications of fuzzy set theory. Recent researches have indicated that a complex ill-defined process can be controlled by constructing a simple rulebase without the mathematical model of the process. It is shown[14] that the rulebase is the key element of a fuzzy controller. The performance of a fuzzy controller depends entirely on the content of the rulebase.

Generally, there are different methods for the rulebase construction. Transferring the experts' knowledge into the rulebase is the most direct approach. However, human knowledge and experience are often hard to be summarized and converted into linguistic expressions even if qualified experts exist. Several papers have been propounded to construct the rulebases by self-learning schemes. Michie[2] divided the state space into several regions, namely the "boxes". Every box behaved like a rule. The learning machine adjusted their outputs according to the behavior of the system. Barto and Sutton[3] adopted a similar structure as Michie's boxes system with the Associate Search Elements (ASE) and the Associate Critic Elements (ACE) for the evaluation of the effect of the default control effort and the correction of the output of each box. The controller learned how to control a complex plant by reinforcement learning. Anderson[4] used two neural nets to substitute the ASE and the ACE. He called them "evaluation network" and "action network". Lee[5] also used the scheme of reinforcement learning. In his system, fuzzy logic rules replaced boxes to control the plant. Only the membership function of the output of each rule was adjusted to achieve the control purpose. Berenji com- bined Anderson's neural networks with a fuzzy controller [6][11] which is actually a neural network with five layers. Except for the above mentioned, there are some other approaches to construct the controller by self-learning. Lin and Lee[12] proposed a neural-network-based fuzzy logic controller. Jang[13] also built a self- learning fuzzy logic controller through temporal back-propogation. The performance of the first four controllers will be discussed in this paper, while the latest two have not been thoroughly studied and will be left for future discussions.

Cell-to-Cell mapping was proposed by Hsu[7-8] in 1980 and has been proved to be extremely effective for the analysis of complex systems[15-16]. It motivated us to use the method for the performance analysis of the above-mentioned reinforcement learning schemes. The simple cell-to-cell mapping (SCM) method was adopted and modified. This paper compare the testing results of Barto's, Anderson's, Lee's and Berenji's works and discuss their differences. Simulations show that the performances of these self-learning schemes are not very satisfactory. There are still a lot of problems to be solved.

The organization of the paper is as follows: Section 2 briefly summarizes the four different rein-forcement self-learning schemes. Section 3 provides the description of the cell-to-cell mapping and explains what we have improved. Section 4 performs the analysis and discusses the simulation results. The conclusions are given in Section 5. The references are listed at the end of this paper.

2. Some Self-learning Control Schemes

Barto and Sutton proposed a neuron-like self-learning controller in 1983[3]. Its basic structure is shown in Figure 1. They divided the state space into a number of "boxes". Sensors get the information of the plant and transfer it to the decoder. If the plant is a cart-pole system, the information includes pole angle, pole angular velocity, cart position, and cart velocity. The decoder determines the current box where the state lies at the instant. The decoded

signal is then used to fire the box correspondingly. There is also a weighting factor with each box. If the weight is greater than zero, the force is 10 Nt.. Otherwise, the force would be -10 Nt.. Initially, the weight of each box is set randomly. There is also an external reinforcement signal r(t) for ACE. r(t) is zero when the plant is inside the operating region. Otherwise, it will be -1. Based on the values of r(t), the ACE produces he internal reinforcement signal $\hat{r}(t)$ and sends it to ASE:

$$\hat{r}(t) = r(t) + \gamma p(t) - p(t-1) \qquad (1)$$

where γ is called the "discount factor". In the simulations, γ is 0.95. To derive the value of $\hat{r}(t)$, ACE makes a prediction p(t) of the reinforcement. There is a specific value V_i for each box in ACE. We let

$$p(t) = \sum_{i=1}^{n} V_i(t) x_i(t) \qquad (2)$$

where $x_i(t)$ is the output of the decoder. After the internal reinforcement signal $\hat{r}(t)$ transferring into ASE, ASE uses it to correct the values of W_i and V_i.

Anderson's controller uses the same idea as Barto's but two neural nets are used to replace ACE and ASE. One is called the Evaluation Network (EN), and the other is Action Network (AN). The AN plays the same role as the ASE with the actions depending on the value of the net-weights summation. Signal p(t) represents the summation of the network output. Then

$$q(t) = \begin{cases} 1, \text{with probabilty } p(t) \\ 0, \text{with probabilty } 1 - p(t) \end{cases} \qquad (3)$$

$$Push(t) = \begin{cases} 10, \text{if } q(t) = 1 \\ -10, \text{if } q(t) = 0 \end{cases} \qquad (4)$$

The EN produces the internal reinforcement.

$$\hat{r}(t+1) = \begin{cases} 0, \text{start state} \\ r(t+1) - v(t,t), \text{failure state} \\ r(t+1) + \gamma v(t, t+1) - v(t,t), \text{other} \end{cases} \qquad (5)$$

where v(t,t) is the output the EN v(t,t) is also a summation of the weights.

Lee's controller is also similar to Barto's with 35 fuzzy linguistic rules in the rulebase. Control forces produced by the rules lie in [-10, 10]. Different from Barto's, the, so called, Associate Critic Neuron (ACN) is used to produce the internal reinforcement signal and the $\hat{r}$ is send into Associate Learning Neuron (ALN) to correct the membership function of the output of each rule:

$$f_i(t) = H(w_i(t) + \text{noise}(t), t), i = 1, \ldots, n \qquad (6)$$

where $f_i(t)$ is the location of the vertex of the triangle membership function of output force. $H(t)$ is a sigmoid function which may be viewed as a dynamic normalization function and provides a continuous output within the range[-10, 10]. ALN only adjusts the membership fun- ction of the action parts of the rules, while the pre-condition parts are fixed with pre-determined fuzzy partitions. The learning process will continue until either the controller fails to balance the pole or the cart is out of bound. To be noted is that, unlike in Barto's and Anderson's work, only the pole position and velocity are discussed in Lee's work.

Figure 2 shows the structure of Berenji's controller. AEN (Action Evaluation Network) plays the same role as EN. AEN has two important factors - one is the external reinforcement r, and the other is the rein- forcement prediction v. AEN uses these two factors to tune the weights and tries to keep the internal reinforcement reaching zero. ASN is a neural-fuzzy controller with five layers. Layer 1 has four nodes as the states in the cart-pole system. Layer 2 works like the fuzzification interface. There are 14 nodes in Layer 2 totally. Every node has three adjustable parameters. One node in Layer 3 represents a rule in the rulebase. The node itself performs the *min* operations. Layer 4 is a de- fuzzification layer. Layer 5 combines the outputs of Layer 3 and Layer 4 and generates the control force. SAM (Sto- chastic Action Modifier) will generate the actual force according to the internal reinforcement and the output of ASN.

All the approaches mentioned above claimed to have good results on their learning behavior in the way that all of them can maintain the balance of the pole for an indefinite period of time. However, in view of controller design, the self-learning scheme will be of much value if the derived controller can drive all the points in the specified stable region of control to the set point, and not just for a few number of specific initial points.

3. Improved Simple Cell-to-Cell Mapping

The method of cell-to-cell mapping was proposed by Hsu in 1980[7] in an attempt to find an efficient and practical way of determining the global behavior of strongly nonlinear systems. The basic idea behind the cell-to-cell mapping is to consider the state space not a continuum but rather as a collection of large number of state cells. Then the original dynamic behavior of the system can be treated as a finite number of cell-to-cell mapping if one or a number of points are chosen to represent a cell. Up to now, there are two methods of cell mapping investigated. One is the simple cell-to-cell mapping (SCM), and the other is the general cell-to-cell mapping (GCM)[9][10]. GCM considers the mapping among cells as a probability distribution and usually attains a finer result with the cost of more computation time and memory. For convenience, SCM is used as the base of the analyzing tool in this paper.

In SCM, the sampling time is constant which will sometimes cause serious problems when dealing with complex dynamic systems. If the dynamics of the system is fast enough, the cell mapping algorithm could choose a distant cell as the next cell in the mapping after one period of sampling time. As a result, the trajectory of the cell mapping is discontinuous and some unreasonable phenomena such as cell trap and trajectory crossing may be deduced. On the other hand, upon slow dynamics, SCM may produce a fake equilibrium point because the trajectory will still stay in the same cell in the short, constant period of time.

To solve these problems, we proposed an im- proved cell-to-cell mapping algorithm called ISCM[17]. It adopted the same structure as the SCM except for the concept of variable integration time. Starting from an initial cell, the algorithm will set up a boundary passing through the centers of all the neighboring cells. Then based on the dynamics of the system, the trajectory will be calculated until it reaches the boundary in a certain neighboring cell which is the goal cell of the mapping. In this way, the cell mapping can only generate goal cells among the neighboring cells and thus avoid the problems of trajectory crossing, etc. The ISCM can reveal more of the system dynamics and avoid suffering of the same demerits as the SCM. Comparing with GCM, it can get the same results without further refinements. It is definitely better than SCM and save more time and memory than GCM.

Barto's, Anderson's, and Lee's works all concentrate

on the well-known cart-pole system. Due to the fact that Lee's controller can only control the angle of the pole, this paper will just compare the effects of pole angles controlling of each controller. The position and the velocity of the cart will be neglected. The operating range of the angle is from -12 deg. to +12 deg., and the angular velocity from -50 deg./sec to +50 deg./sec. The size of the cell is 0.1 deg.* 1deg./sec. Total number of cells is 24000.

4. Comparisons and Simulation Results

To be analyzed by the ISCM, the coefficients of each controller must be fixed, i.e. the learning mechanism of all the four controllers are deactivated when the system performances are considered to be satisfactory. Barto's and Lee's schemes are deterministic and can be easily used for the application of ISCM. However, both Anderson's and Berenji's controllers involve stochastic processes and can only be modified to serve our purpose. There is a random number generator for determining the output of the controller in Anderson's work. The scheme compares the random number in [0,1] with the prediction p(t). The output force is +10 Nt. if the random number is smaller than p(t) and -10 Nt. vice versa. To comply with the structure of the ISCM, the generated random number in Anderson's work is set to be 0.5. If p(t) is greater than 0.5, the output force would be +10 Nt. Otherwise, the output force would be -10 Nt. The modified scheme can be viewed as an averaging approximation of the original one and, in the limiting case, Anderson's scheme should be converging to the modified one if the learning is successful and convergent. Similar situation applies in Berenji's case. The stochastic action modifier in Berenji's controller is eliminated after the training is completed.

The above four self-learning controllers are designed to keep on learning until the failure signal appears and are not aimed for the constructions of stand-alone controllers. However, in view of controller design, it is not acceptable to have a controller which is, in a sense, unpredictable. Therefore, we managed to evaluate the four controllers by eliminating the learning mechanisms after successfully completing 500,000 time steps. The controllers under evaluations are somehow different from what they really were.

Simulation results are shown in the following figures. Figures 3 to 6 show the regions of convergence for Barto and Sutton's, Anderson's, Lee's, and Berenji and Khedkar's controllers respectively. In the simulation, the viewing area reveal the adimttable control space ranging from 12 deg. to -12 deg. and 50 deg./sec and -50 deg./sec. There are only two groups of cells in the cell maps of the first three control systems where group 1 is for the "sink cell" and group 2 is for the convergent cells. (The sink cell means the cell which belongs to some trajectory going toward the outside region.)

In figure 3(Barto's case), there is a limit cycle for the cells in group 2, which is around the origin of the angle-velocity plane which means the controller does not keep the pole stably balanced and indulges in oscillations. Actually, the system behaves like a Bang-Bang control system which can balance the pole stay in the limiting region but excludes the reasonable equilibrium point because limit cycle locates outside the central region.

The cell map from Anderson's scheme shows some strange phenomena. There are 39 groups totally in its cell map. In the learning procedure, we can see that Anderson's controller does not have a consistent behavior. The same p(t) associated with one state could have two different output forces at different time instants. The random process in Anderson's approach helps the searching in the state space for proper control forces. However, the approach should be modified in the line of the simulated anealing so that the learning can converge in the long run. Otherwise, the stochastic behavior will continue and the scheme will not be able to derive a controller with consistent performances. We speculate that the 39 groups found in the cell map indicating the transient and random behavior of Anderson's control scheme. We also find all groups except group 1 have no equilibrium point and have only limit cycles. Group 2 has more cells than the other groups and is the dominant one relatively.

Lee's controller seems the most normal one in figure 5. The cell map appears to have only one equilibrium point. There is no limit cycle in group 2. The equilibrium point is the origin. However, the demerit of the controller is that the controllable area is small. The state outside the controllable region would be out of control. In simulations, we also observed that there is a "path" leading outside of the controllable area to the initial state. This shows that the initial state will have influence on the cell map, i.e. Lee's controller is "initial state dependent".

Generally speaking, Berenji's controller has the best performance. It seems to be able to control the majority of the state space (Figure 8). However, Berenji's controller has a serious problem that it can not maintain the pole and the cart in steady states at the same time. From the simulations, we find that it will oscillate seriously. Oscillations also happen in the learning process and sometimes the controller would diverge. We can find the stochastic elements affect the controller. Barto's and Lee's works exclude such random modifier, so their behaviors are more reasonable. If we need a stable controller, we should eliminate this stochastic behavior at first.

5. Conclusions

By comparison of these four controllers, it is obvious that although all of them claimed to be able to control a cart-pole system. None of them can be considered to be a good controller design methodology because none can guarantee a reasonably large, compact region of stability. Actually, the region of controllable states is far smaller than we speculated.

Although the self-learning schemes discussed above have their problems. They do suggest a new way to establish the control rulebase under insufficient informations. In fact, there are many new methods proposed after them. It has also been shown that the ISCM can be used to provide some ideas about the dynamic behavior of a complex system.

References

[1] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plants", *Proc. IEE* 121(12), pp.1585-1588, 1974

[2] D. Michie and R. A. Chambers, "'Boxes' as a model of pattern-formation", in *Toward a Theoretical Biology*, vol. 1, *Prolegomena*, C. H. Waddington, ed. Edinburgh Univ. Press, 1968, pp. 206-215

[3] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems", *IEEE Tran. SMC*, vol. 13, no. 5, 1983, pp. 834-846

[4] C. W. Anderson, "Strategy learning with multilayer connectionist

representation", Tech. Rep. TR87-509.3, GTE Laboratories Inc., May 1988
[5] C. C. Lee, "Self-learning rule-based controller employing approximate reasoning and neural-net concepts", *Int. J. Intelligent Systems*, vol. 6, pp. 71-93, 1991
[6] H. R. Berenji, "A reinforcement learning-based architecture for fuzzy logic control", *Int. J. Approximate Reasoning*, no. 6, 1992, pp.267-299
[7] C. S. Hsu, "A theory of cell-to-cell mapping dynamical systems", *J. Applied Mechanics*, 47, pp. 931-939
[8] C. S. Hsu, "Cell-to-cell mapping", 1987, ed. Springer-Verlag
[9] C. S. Hsu, "A generalized theory of cell-to-cell mapping for nonlinear dynamical systems", *J. Applied Mechanics*, 48, pp. 634-842
[10] C. S. Hsu, "A probabilistic theory of nonlinear dynamical systems based on the cell state space concept", *J. Applied Mechanics*, 49, pp.895-902, 1982
[11] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements", *IEEE Trans. Neural Networks*, Sep. 1992, pp. 724-740
[12] C.-T. Lin and C.-S. G. Lee, "Neural -network-based fuzzy control and decision system", *IEEE Trans. Computer*, Dec. 1991, pp.1320-1336
[13] J.-S. R. Jan, "Self-learning fuzzy controller based on temporal backprogation", *IEEE Trans. Neural Networks*, Sep. 1992, pp.714-723
[14] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, Part I", *IEEE Trans. SMC*, Vol. 20, 2, 1990
[15] C. S. Hsu and H. M. Chiu, "A cell mapping method for nonlinear deterministic and stochastic systems-part I: the method of analysis", *J. Applied Mechanics*, 53, pp. 695-701, 1986
[16] J. Q. Sun and C. S. Hsu, "A statistical study of generalized cell mapping", *J. Applied Mechanics*, 55. pp. 694-701, 1988
[17] C. F. Pong and Y. Y. Chen, "Cell-to-cell mapping with variable integration time",

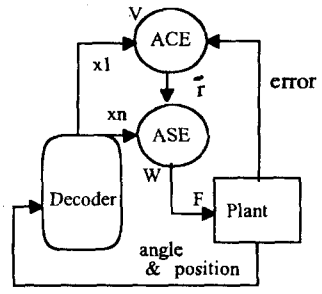

Fig. 1 Barto and Suttton's structure of their controller

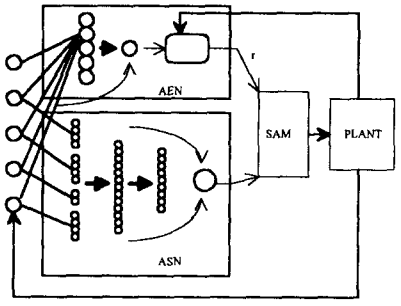

Fig. 2 The structure of Berenji's controller

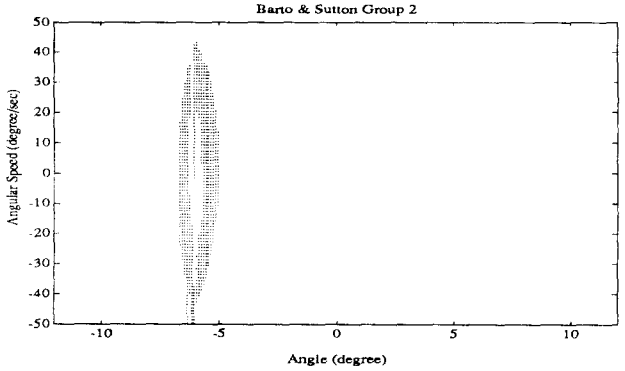

Fig. 3 Barto and Sutton's Cell Map of Group 2

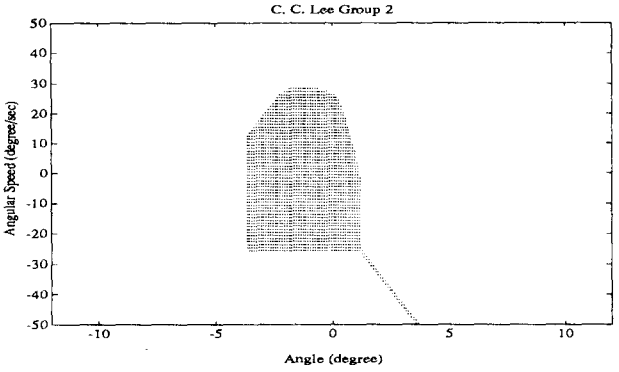

Fig. 5 Lee's Cell Map of Group 2

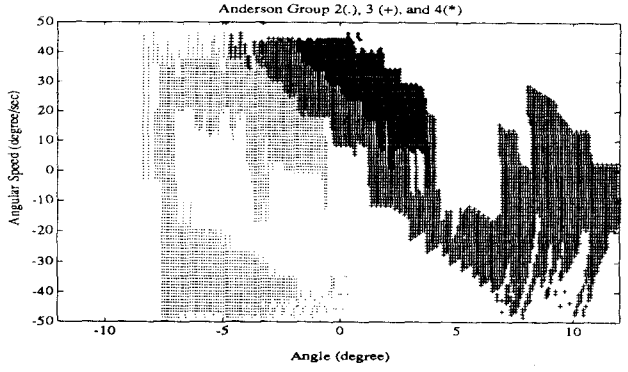

Fig. 4 Anderson's Cell Map of Group 2, 3, and 4

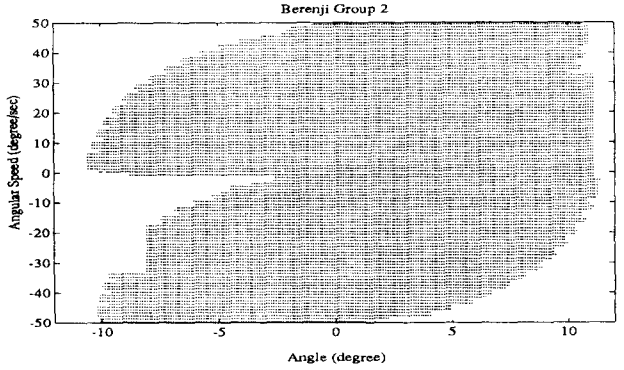

Fig. 6 Berenji's Cell Map of Group 2