A Fuzzy BOXES Scheme for the Cartpole Control

Sunggyu kwon*

* Faculty of Mechanical and Automotive Engineering, Keimyung University, Daegu, Korea
(Tel : +82-53-580-5231; E-mail: cmack@kmu.ac.kr)

Abstract: Two fuzzy controllers are coordinated to control a cartpole such that the pole is balanced as well as the cart is brought back to the track origin. The coordination is due to the BOXES scheme that is established through the evaluation of the outcomes of the control action by one of the fuzzy controllers. It is found that the control scheme is good at selecting proper fuzzy controller so that the pole is balanced fast while the cart moves back to the track origin steadily.

Keywords: Cartpole, Fuzzy, Reinforcement Learning, BOXES

1. INTRODUCTION

Control of cartpole[1] system has been the object of quite many studies in the literature of control and neural networks. Fig. 1 shows the cartpole system. A rigid pole is hinged to a cart which travels along the track. The pole can rotate in the vertical plane of the cart. An impulsive force is applied to the cart at discrete time intervals to control the system.

Most control problems of the cartpole system are just for balancing the pole without considering the position of the cart. To balance the pole, the cart is pushed back and forth on the track so that the pole never deviates by more than certain, for example 12 degrees from the upright position. However, more demanding control problems require balancing as well as centering which means that the controller is to bring the cart back to the center of the track while keeping the pole upright. Even on a track of limited length, balancing does not imply centering. Some controllers are capable of balancing and centering if a slightly longer track is available [1].

There are many intelligent control techniques to balance the pole of the cartpole system. BOXES scheme is the first attempt at unsupervised learning. Barto et al. [2] proposed a reinforcement learning scheme using two neuron-like adaptive elements and bang-bang control for balancing the pole. Using neural networks, Anderson [3] presented reinforcement and temporal-difference learning methods to balance the pole. A modified structure of CMAC neural networks was proposed to accelerate reinforcement learning control and applied to a cartpole balancing problem [4].

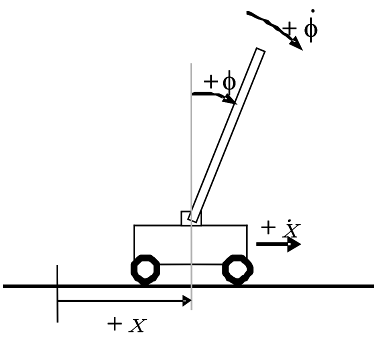

Fig. 1 A cartpole system. 4 variables define the system state: x and $\dot{x}$, the position and the velocity of the cart and ϕ and $\dot{\phi}$, the angular position and the velocity of the pole.

Fuzzy control is used in many techniques for the cartpole control. Lee [5] developed an intelligent control scheme by integrating fuzzy control and reinforcement learning techniques. To balance the pole, 7 linguistic values (labels) for the pole angle and 3 labels for the angular velocity of the pole were employed to generate continuous forces as output. Deng et al. [6] proposed a neural-fuzzy BOXES control system with reinforcement learning where the state space is divided into some overlapping fuzzy boxes by defining input membership functions for each state variable. Although the control scheme is expected to yield more generalization and learning abilities, it is only to balance the pole.

On the other hand, there are few application examples for balancing the pole and centering the cart. A learning architecture was proposed for training a multilayer neural network which provides the proper forces to the cart so that the pole is balanced and the cart is guided to the center of the tack [7]. To determine the force necessary to control the cartpole system, priori knowledge regarding the relationships among state variables for balancing the pole was utilized. Some learning processes were not successful. Berenji and Khedkar [8] proposed a new way of designing and tuning a fuzzy logic controller where learning is achieved by integrating fuzzy inference into a feedforward neural network. The method was applied to a cartpole control system where 4 labels were employed for each of 4 state variables and 9 labels for force. And, 9 fuzzy control rules were involved for balancing and 4 rules for positioning cart at a specific location on the track. Although the control objective is achieved, some linear oscillation of the cart and angular oscillation of the pole may be present. A fuzzy-inference-based reinforcement learning algorithm which combines the fuzzy inference engine with neural networks by reinforcement signal generating was proposed. The algorithm was applied to the cartpole system where 2 state variables and action network's output were used as the inputs to the controller [9]. However, the system performance for hard initial configuration of the cartpole system is questionable.

In this paper, in order to control the cartpole system in terms of both balancing and centering, a fuzzy BOXES control scheme is developed. Two independent fuzzy controllers, one is based on the angular position and the angular velocity of the pole and the other based on the position and the velocity of the cart, are employed to produce the control forces. To apply proper control force, one of two fuzzy controllers is selected for every time step due to a BOXES algorithm. The BOXES is established through observation of the outcomes of the control action. The behaviors of the cartpole are analyzed to discuss the fuzzy BOXES control scheme.

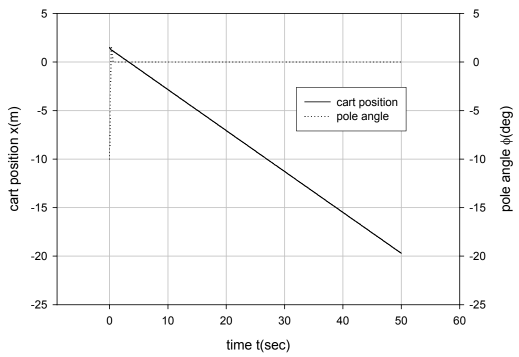

Fig. 2 The cart position and the pole angle versus time of the cartpole due to the ϕ-fuzzy controller.

2. FUZZY CONTROL

Although there may be a few cases of using 4 state variables for fuzzy logic controller for the cartpole system [10], most of fuzzy controllers use 2 state variables, the pole angle (ϕ) and the angular velocity of the pole ($\dot{\phi}$), for the cartpole balancing problem [11]. Unfortunately, with these controllers, there is no way to control the position of the cart while the pole is balanced. In fact, using the other 2 state variables, the position of the cart (x) and the linear velocity of the cart ($\dot{x}$), a fuzzy controller can balance the pole as well as bring the cart back to the center of the track. However, it takes too long time until the fuzzy controller brings the cart back to the track origin.

First of all, it is necessary to discuss about the fuzzy controllers based on 2 state variables, ϕ and $\dot{\phi}$, (called ϕ-fuzzy controller). For a ϕ-fuzzy controller, there are 7 linguistic values for each input state variable and 7 linguistic values for the output variable, force. Triangular membership functions are used for all linguistic values. “Center of gravity” (COG) defuzzification method [11] was used to combine the recommendations represented by the implied fuzzy sets from all the rules. For computer simulations of the cartpole system, a 4^{th} order Runge-Kutta method with a time step of 0.04 seconds was used to approximate numerically the solution of the friction dynamics equations with cartpole parameters in [2].

Fig. 2 shows the cart position and pole angle versus time when the cartpole system operates by the ϕ-fuzzy controller after the system is released at the state of (1.0,-1.0,-10.0,40.0). The figure illustrates that the controller is able to balance the pole very quickly even though the cart is out of control to bring back to the track origin. Therefore, that the ϕ-fuzzy controller based on ϕ and $\dot{\phi}$ can balance the pole means that it is certain that the amplitude of the angular oscillation of the pole decreases fast. Thus, the fuzzy controller should decrease the amplitude of the angular oscillation of the pole in all circumstances whenever the pole does not fall.

On the other hand, it is found that the fuzzy controller based on 2 state variables, x and $\dot{x}$, (called x-fuzzy controller) can move the cart back to the track origin while the amplitude of the angular oscillation decreases very slowly.

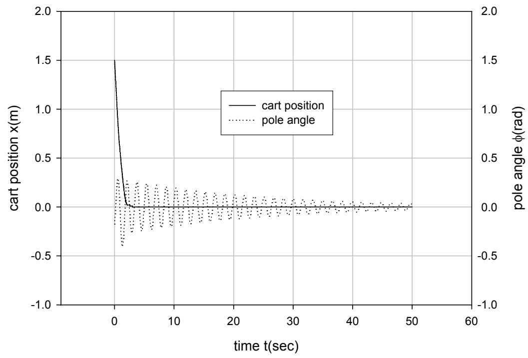

Fig. 3 The cart position and the pole angle versus time of the cartpole due to the x-fuzzy controller.

The x-fuzzy controller is designed using the same parameters as the ϕ-fuzzy controller. Fig. 3 shows that the cart moves quickly to the center of the track as the pole oscillates continuously while the magnitude of its angular oscillation decreases very slowly after the system is released at the state of (1.5,-1.0,.-10.0,40.0). Thus, it is noted that the x-fuzzy controller is definitely able to bring the cart back to the track origin. However, it seems that because of the very lazy convergence of the x-fuzzy controller, most fuzzy controllers use the other 2 state variables for the cartpole control.

The comparison of Fig. 2 and Fig. 3 shows that the system behaviors due to these fuzzy controllers are very contrasting. The ϕ-fuzzy controller is very quick to control the angular position of the pole whereas the x-fuzzy controller is too slow to balance the pole. On the other hand, the ϕ-fuzzy has no control over the distance of the cart from the track origin while the x-fuzzy controller has the capability to move the cart to the track origin very quickly.

Hence, it is supposed that proper coordination of these two controllers should accomplish a good control in terms of balancing the pole as well as centering the cart of the cartpole system. Here is a simple scheme to do this: Given a state of the system, at start, the ϕ-fuzzy controller is preferentially to produce the control force. If the distance of the cart from the track origin decreases because of the control force, the ϕ-fuzzy controller keeps the control of the system for next time step. Otherwise, the x-fuzzy controller gets the job. Then, the control force due to the x-fuzzy controller is applied. Then, the angular displacement of the pole is examined without regarding the position of the cart since the cart will converge to the track origin soon as long as the x-fuzzy controller keeps the control. If the angular displacement is less than the previous one, the controller continues producing the control force. Otherwise, the control is to be transferred back to the ϕ-fuzzy controller. This procedure continues.

Fig. 4 shows the system behavior in terms of the changes of x and ϕ due to this control scheme. It takes about 50 seconds for the two cooperating fuzzy controllers to move the cart back to the track origin while the angular amplitude of the pole decreases regularly with some discontinuous changes of the pole angle. Although both the cart and the pole oscillate for a while, magnitudes of the oscillations become negligible soon. It is very interesting to see that the cooperation of two

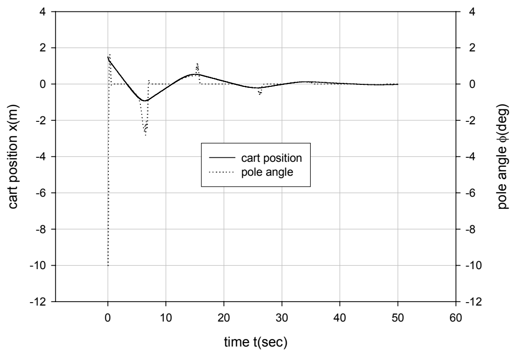

Fig. 4 The cart position and the pole angle versus time of the cartpole due to the combination of the ϕ-fuzzy controller and the x-fuzzy controller.

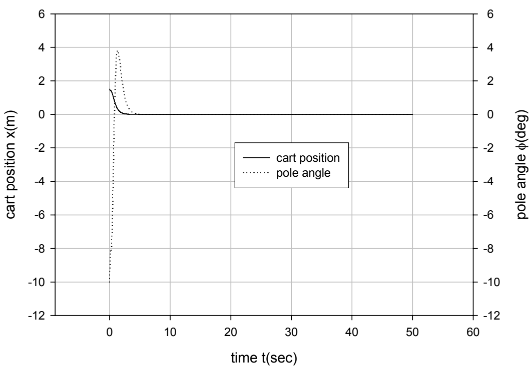

Fig. 5 The cart position and the pole angle versus time of the cartpole due to a linear control law.

fuzzy controllers whose behaviors are quite distinctive comes up with balancing and centering of the cartpole system seen as in Fig. 4. This figure can be compared with Fig. 5 that is due to a linear control law [12]. It seems that the system behavior of the Fig. 4 is more natural in terms of slow convergence than that of the Fig. 5.

Through many experiments with this control scheme, it is found that the ϕ-fuzzy controller should play a major role in the cooperative control in this scheme for balancing as well as centering. However, it is the x-fuzzy controller that makes the cooperation ended up with moving the cart back to the track origin.

According to this scheme, for some time steps, the selection of a fuzzy controller is altered from one fuzzy controller to the other if the pole angle is increased due to the x-fuzzy controller or the cart distance from the track origin is increased due to the ϕ-fuzzy controller. This means that the scheme cannot help suffering from some wrong selections although the cooperation of the fuzzy controllers is due to the changes of selection of the fuzzy controllers for some time steps. So, it will take longer time until the cartpole system gets stable and require more control actions. For example, for the Figure 4, there are 514 times of changing the selection of the fuzzy controllers out of 1251 time steps. This amounts to about 40 % of total time steps. This means that about 40 % of control actions were improper. It is estimated that those

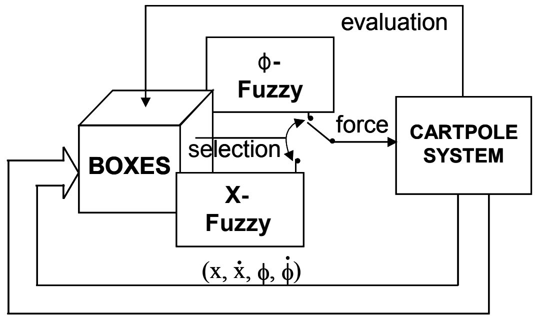

Fig. 6 A scheme of the fuzzy BOXES controller where a 4-D BOXES is integrated with two fuzzy controllers for the cartpole control.

improper control actions result in more discontinuous changes of angular position of the pole and cause a longer time for control the system.

However, this process of selecting proper fuzzy controller can be regarded as a learning process which will implement a mechanism to select a proper fuzzy controller for every time step based on the system state by observing the outcome of the control.

3. BOXES Scheme

BOXES scheme was proposed by Michie and Chambers [13] and was applied to control the cartpole system. The scheme is to partition the continuous state space into a small number of sub-spaces, or called 'boxes.' Each box is supposed to contain a local demon. The local demon has a switch to select a control action so that determining which system state is entered into its box.

Now, for a fuzzy BOXES controller, a BOXES learning scheme is devised to implement the mechanism that is to select a proper fuzzy controller for every time step in the cartpole control in terms of balancing and centering. Fig. 6 shows a fuzzy BOXES controller where the (4-D) BOXES is composed by all the boxes that correspond to all of the combinations of the intervals that divide each state variable into a small numbers of disjoint regions by quantizing the variable. Thus, they divide the continuous 4-dimensional cartpole state space into disjoint boxes by quantizing 4 state variables. Two fuzzy controllers, the ϕ-fuzzy controller and the x-fuzzy controller, are integrated with the BOXES such that for every time step one of the fuzzy controllers is selected by the BOXES in order to produce proper control force. Here, it is noted that the number of linguistic values for each input variables for the fuzzy controllers may be different from the number of quantizing intervals for composing the BOXES.

All the boxes of the BOXES are initialized before establishing the fuzzy BOXES controller. This means that the BOXES has never been trained. Training process for the controller is as following: (A) For starting with a given system state, the BOXES is called to locate the box that represents the system state. (B) If the content of the box is meaningful, for example +1 indicating that the ϕ-fuzzy controller is good for the corresponding system state and –1 is for the x-fuzzy controller, one of the fuzzy controllers is selected to produce the proper control force. With the force, the cartpole system operates.

If the outcome of the operation due to the selection of the

ϕ-fuzzy controller is good such that the new distance of the cart is smaller than the previous one, then the content of the box, +1 is kept as it is. Otherwise, the content is to be reinitialized. Similarly, if the outcome of the operation due to the selection of the x-fuzzy controller is good such that the new angular displacement of the pole is smaller than the previous one, then the content of the box, -1 is kept as it is. Otherwise, the content is to be reinitialized. For the next time step, the BOXES is called by getting the state vector of the cartpole system to identify the box that represents the system state for the BOXES.

(C) However, when the BOXES is called at start, if the content of the box is not meaningful value, for example 0 indicating the BOXES has nothing to do with selecting a proper fuzzy controller, preferentially the ϕ-fuzzy controller has to go ahead and produce the control force. Then, the BOXES gets a learning opportunity for the box that represents the current system state based on the outcome of applying the control force. +1 is stored in the box if the outcome is evaluated as good. Otherwise, no learning occurs for the BOXES and the content of the box is initialized again.

(D) For the continuing time step, now, without calling the BOXES, the x-fuzzy controller is to produce the control force for the system state. Then, the BOXES has another learning opportunity based on the outcome of applying the force. –1 is stored in the box if the control is good. Otherwise, no learning occurs for the BOXES and the content of the box is initialized again.

For the next time step that corresponds to the third time step after the BOXES failed in yielding a meaningful value, the process follows the procedure (A). Hence, the procedures (A) and (B) can be considered as the regular operation procedures, while the procedures (C) and (D) are for training the BOXES. That the preference is given to the ϕ-fuzzy controller when the BOXES fails in yielding a meaningful value for the selection of one of the fuzzy controllers is due to the experience with the cooperative control with two fuzzy controllers.

The system operation continues and the BOXES gets some training opportunities from time to time. The content of some boxes may be updated several times while the cartpole system operates. The reason is that the neighboring system states are represented by the same box and the box may be called several times. Moreover, the content of a box may be updated differently from time to time since the box can be called for some neighboring states for which the selection of the fuzzy controller by BOXES may be different. While the system operates, the content of some of the boxes are +1, some other boxes contain –1, and the others contain 0.

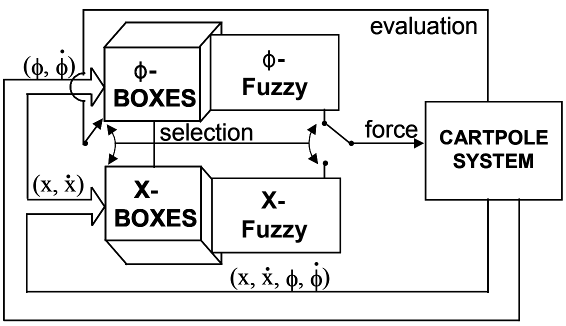

Fig. 7 A scheme of the fuzzy BOXES controller where 2 sets of 2-D BOXES are integrated with two fuzzy controllers for the cartpole control.

Now, let's discuss the problem of constructing the BOXES. The cartpole system state is defined by 4 variables, x, $\dot{x}$, ϕ and $\dot{\phi}$. Then, for the selection of a proper fuzzy controller, the BOXES usually uses all of these 4 state variables to compose its boxes [2]. On the other hand, two fuzzy controllers use only 2 variables respectively. Thus, for each control action, only a half of the number of the state variables gets involved with producing the control force although the system state vector is defined by the whole variables.

Here, let's consider the size of the BOXES in terms of the number of boxes. Based on the number of partitions for each variable, the number of boxes is determined. If there are 10 partitions for each variable, the number of boxes amounts to $10^4 = 10,000$ for the 4-D BOXES. However, it is noted that the more number of partitions does not necessarily improve the performance of the BOXES learning scheme. Also, more boxes require more learning opportunities and more computation cost. Although as small as 1,296 ($= 6^4$) number of boxes could be a BOXES for the fuzzy BOXES controller, reducing the dimension of the BOXES is preferred to have less number of the boxes.

For a development of constructing the BOXES with less number of dimensions, it is noted that the fuzzy controllers use only 2 state variables to produce the control forces even though the system state is described by 4 state variables. To the fuzzy controllers, two state variables, ϕ and $\dot{\phi}$ or x and $\dot{x}$, are enough to represent the system state. Accordingly, it is natural to consider a set of 2 state variables for the BOXES (2-D BOXES) for each fuzzy controller, ϕ-BOXES for ϕ and $\dot{\phi}$ and X-BOXES for x and $\dot{x}$.

Fig. 7 shows the diagram of the fuzzy BOXES controller with two 2-D BOXES. For this scheme, each BOXES gets a set of 2 state variables to identify the box which represents the system state. Thus, 2 boxes are identified for every time step. If the values of those two boxes are meaningful, then the selection of the fuzzy controllers is decided preferentially according to the value of the box identified by the ϕ-BOXES. When only a box of the X-BOXES has a meaningful value, the x-fuzzy controller is selected to produce the control force. If both BOXES fail in yielding a meaningful value, then the learning process gets to follow the procedure (C) described above.

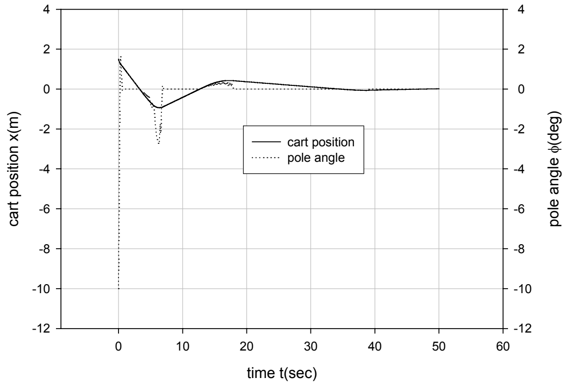

Fig. 8 The cart position and the pole angle versus time of the cartpole due to a fuzzy 2-D BOXES controller.

Table 1 Some system states and the active controllers for some time steps after the cartpole system releases from the state (1.5,-1.0,-10.0,40.0)

Time	x	$\dot{x}$	ϕ	$\dot{\phi}$	C'l	box
6.20	-.929	-.102	-2.708	5.229	X	50
6.24	-.930	.051	-2.731	-6.373	ϕ	104
6.28	-.931	-.070	-2.747	5.619	X	50
6.32	-.931	.084	-2.753	-5.952	ϕ	104
6.36	-.930	-.035	-2.754	5.920	ϕ	107
6.40	-.932	-.065	-2.436	9.960	ϕ	107
6.44	-.934	-.076	-1.991	12.209	X	49
6.48	-.934	.079	-1.746	0.004	X	50
6.52	-.928	.232	-1.989	-12.091	ϕ	118
6.56	-.921	.100	-2.221	0.501	ϕ	106
6.60	-.919	.039	-2.070	7.032	ϕ	107

Fig. 8 shows the system behavior due to the fuzzy BOXES controller of Fig. 6 for 50 seconds after the cartpole system is released at (1.5,-1.0,-10.0,40.0). The system behavior is compared with that of Fig. 4 which is for the two cooperating fuzzy controllers. For 1,251 time steps, 942 control actions were directly due to the value of boxes either of the ϕ-BOXES or the X-BOXES. For other time steps, 9 arbitrary actions were good for learning. However, the other 300 time steps were evaluated as bad. This amounts to about 24 % of all control actions. So, about 24 % of control actions were improper. Comparing this 24 % with 40 % for the case of the cooperative fuzzy controllers, it is possible to appreciate that more effective coordination of selection of fuzzy controllers could cause to improve the system behavior. The pole and the cart oscillate less frequently and the control converges soon.

The number of total boxes of two 2-D BOXES is much less than those of the 4-D BOXES. If 10 partitions are employed for each state variable, $10^2 = 100$ boxes are required for each 2-D BOXES. For two 2-D BOXES, $2 \times 100 = 200$ boxes are necessary. This number is quite comparable to the number 1,296 with 6 partitions for each state variable by the 4-D BOXES.

If the fuzzy BOXES controller with two 2-D BOXES is able to control the cartpole system, the controller is much more effective than that with 4-D BOXES. Since the latter uses less number of boxes, it will take less training time and be more active in updating its content of boxes.

4. Discussion and Results

The fuzzy 4-D BOXES controller with 1,296 boxes due to 6 partitions for each state variable could take care of coordinating the selection of the fuzzy controllers. And, the system behavior was very similar to that of Figure 8.

For the fuzzy 2-D BOXES controller, 2 state variables, x and $\dot{x}$, were partitioned into 14 intervals for the domains of [-2.5,2.5] (m) and [-150,150] (m/sec) respectively and 196 boxes were employed for X-BOXES. ϕ and $\dot{\phi}$ were partitioned into 18 and 14 intervals for the domains of [-40,40] (degrees) and [-150,150] (degrees/sec) respectively and 252 boxes were necessary for ϕ-BOXES. Total number of boxes 448 is only 35 % of 1,296 which is the number of boxes for the 4-D BOXES. Even the fuzzy 2-D BOXES controller with 100 boxes for the X-BOXES and 140 boxes for the ϕ-BOXES is able to coordinate the selection of the fuzzy controllers.

Adopting two 2-D BOXES brings an interesting effect. Some different system states defined by 4 state variables are represented to the x-fuzzy or the ϕ-fuzzy controller as the same state based on 2 state variables. Due to this, for some different system states, the content of the same box may be updated. This characteristic seems to make the BOXES scheme adapted for most recent system states.

It is important for this scheme to update the content of the box which represents the system state at the corresponding time step according to the outcome of applying a fuzzy controller. When the content of a box is updated, the previous content is simply ignored.

Table 1 lists the system states from t=6.20 to t=6.60 for 0.40 seconds when the system releases from the state (1.5,-1.0,-10.0,40.0) of Fig. 8. In this table, C'l stands for controller and box denotes the box number. For example, at time t=6.20, the 50th box of the X-BOXES represents the system state and the ϕ-BOXES is not able to represent the system state since any box of the ϕ-BOXES has no meaningful value at the moment. Then, the x-fuzzy controller produces the force 3.968 (Newtons). Due to the control action, the angle of the pole increases from -2.708 to 2.731 (degrees). Then, it is determined that it is not good to let the x-fuzzy controller keep the control of the cartpole system at next time step since the pole reclines more although there is no doubt about centering the cart by the x-fuzzy controller in the end. So, the x-fuzzy controller should lose the control of the cartpole system and the content of the box of the X-BOXES is reinitialized. At time t=6.24, the ϕ-fuzzy controller takes the control.

On the other hand, at time t=6.32, the ϕ-fuzzy controller was good since the cart moves toward the track origin from x=-0.931 to x=-0.930. Furthermore, at time t=6.44, by the 49th box of the X-BOXES system, x-fuzzy controller was able to control successfully.

However, at most of time steps in the table except at t=6.32, 6.44, 6.52 and 6.56, and 6.60, the fuzzy BOXES controller is not able to coordinate the selection of the fuzzy controllers since it has not been established with meaningful values at those boxes which represent the system states. In these situations, the control is due to the ϕ-fuzzy controller and x-fuzzy controller consecutively. In the mean time, through these mandatory control actions, the BOXES systems could get opportunities to establish their content.

Not like other intelligent control techniques, this control scheme is able to produce the control forces which are odd symmetry with respect to system states. However, it seems that this symmetric force producing is mainly due to the characteristic of the fuzzy controllers.

Comparing Fig. 8 with Fig. 5 due to the a linear control law, it is obvious that the fuzzy BOXES controller is very slow to converge while the system behavior looks stable. However, it is noted that the control scheme is simple to achieve both balancing and centering.

5. Conclusion

A control scheme was devised to coordinate two fuzzy controllers to balance the pole as well as to bring the cart back to the track origin of the cartpole system. One fuzzy controller based on the pole angle and the angular velocity of the pole is able to balance the pole while the other fuzzy controller based on the cart position and the linear velocity of the cart is able to bring the cart back to the track origin. The coordination is due to the BOXES system which is established through some learning experience. It is found that the fuzzy BOXES controller is able to control the cartpole system and the training strategy works good to establish the BOXES system while the cartpole system never fails during the training.

Although there are quite many control techniques which are able to balance the pole, they cannot be utilized to bring the cart back to the track origin. Also, some of control schemes which are capable of balancing and centering do not perform well such that the controllers require longer training time or suffer from some residual oscillations. In contrast, the control scheme presented in this paper is simple to implement. In addition, it is expected that similar fuzzy controllers for other applications can be coordinated by adopting the learning scheme in a fuzzy BOXES controller. For example, a ball-beam control problem [14] could be performed by the control scheme presented in this paper.

REFERENCES

[1] S. Geva and J. Sitte, “A Cartpole Experiment Benchmark for Trainable Controllers,” *IEEE Control Systems Magazine*, Vol. 13, No. 5, pp. 40-51, 1993.

[2] A. G. Barto, R. S. Sutton and C. W. Anderson, “Neuron-like Adaptive Elements that can Solve Difficult Learning Control Problems,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 13, No. 5, pp. 834-846, 1983.

[3] C. W. Anderson, “Learning to Control an Inverted Pendulum using Neural Networks,” *IEEE Control Systems Magazine*, Vol. 9, No. 4, pp. 31-37, April 1989.

[4] X. Xu, D. Hu and H. He, “Accelerated Reinforcement Learning Control using Modified CMAC Neural Networks,” *Proceedings of the 9th International Conference on Neural Information Processing* (ICONIP’02), Vol. 5, pp. 2575-2578, 2002.

[5] 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.

[6] Z. Deng, Z. Zhang, and P. Jia, “A Neural-Fuzzy BOXES Control System with Reinforcement Learning and its Application to Inverted Pendulum,” *IEEE International Conference on Intelligent Systems for the 21st Century*, Vol. 2, pp. 1250-1254, Oct. 22-25, 1995.

[7] V. Williams and K. Matsuoka, “Learning to Balance the Inverted Pendulum using Neural Networks,” *IEEE International Joint Conference on Neural Networks*, Vol. 1, pp. 214-219, Nov. 18-21, 1991.

[8] H. R. Berenji and P. Khedkar, “Learning and Tuning Fuzzy Logic Controllers Through Reinforcements,” *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp. 724-740, September 1992.

[9] H. B. Jun, D. W. Lee, D. J. Kim and K. B. Sim, “Fuzzy Inference-based Reinforcement Learning of Dynamic Recurrent Neural Networks,” *Proceedings of the 36th SICE Annual Conference, International Session Papers*, pp. 1083-1088, July 29-31, 1997.

[10] F. Song, S. M. Smith and C. G. Rizk, “Reducing Memory Requirement of Cell State Space Based Fuzzy Logic Controller Design Approaches Using K-d Trees,” *IEEE International Conference on System, Man, and Cybernetics*, Vol. VI, pp. 144-149, 1999.

[11] K. M. Passino, *Biomimicry for Optimization, Control, and Automation,* Springer-Verlag London Limited, pp. 222-223, 2005.

[12] A. Guez and J. Selinsky, “A Trainable Neuromorphic Controller,” *Journal of Robotic Systems*, Vol. 5, No. 4, pp. 363-388, 1988.

[13] D. Michie and R. A. Chambers, “’BOXES’ as a model of Pattern-formation,” in *Towards a Theoretical Biology*, Vol. 1, Prolegomena, C. H. Waddington, Ed., Edinburgh: Edinburgh Univ. Press, pp. 206-215, 1968.

[14] Y. Jiang, C. McCorkell and R. B. Zmood, “Application of Neural networks for real time control of a Ball-Beam System,” *Proceedings of International Conference on Neural Networks*, pp. 2397-2402, 1995.