Designing Interval Type-2 Fuzzy Controllers by Sarsa Learning

Nooshin Nasri Mohajeri*, Mohammad Bagher Naghibi Sistani**
* Ferdowsi University of Mashhad, Mashhad, Iran, noushinnasri@ieee.org
** Ferdowsi University of Mashhad, Mashhad, Iran, mb-naghibi@um.ac.ir

***Abstract**: In this paper, new Interval type-2 fuzzy controllers, which are designed by Sarsa learning, are proposed. The porposed controllers are A2-C0 Takagi-Sugeno-Kang type. Therefore, the antecedent part of rules is formed by fuzzy type-2 sets and the consequent part is comprised of possible actions. In the output processing section of fuzzy type-2 controllers, in addition to, Karnik-Mendel type reducer accompanied by centroid deffuzification, another output processor called BMM method is applied. Consequently, IT2FSL-KM and IT2FSL-BMM are generated. These new controllers are compared with other fuzzy controllers designed by RL methods in truck backing control problem. Simulation results represent the efficiency and effectiveness of proposed controllers in noiseless and noisy environment.*

Keywords: Fuzzy Q-learning (FQL), fuzzy Sarsa learning (FSL), interval type-2 fuzzy set (IT2F Set), reinforcement learning (RL), type-2 fuzzy logic systems (T2FLS).

1. Introduction

Design of fuzzy controllers is not a simple procedure especially for complicated systems with large input spaces. Therefore, many efforts have been done to design a fuzzy controller automatically by using supervised learning, unsupervised learning, or reinforcement learning methods. Reinforcement learning is an appliance that designs fuzzy controllers via trial and error. In fact, RL uses a weak reinforcement for communication with the environment. An extension of classic Q-learning is known as Fuzzy Q-learning that is an approach to learn a set of fuzzy rules by reinforcement signal [1]. Glorennec and Jouffe in [2] and [3] proposed fuzzy Q-Learning which is the basis of the existing implementations. In [4] Meng Joo Er and Chang Deng presented a new type of FQL called Dynamic fuzzy Q- learning (DFQL), which has self-organizing ability so that fuzzy rules are generated automatically and simultaneously based on Q-learning.

In [5] Derhami, Majd, and Nili Ahmad abadi showed that FQL is incapable of convergence in some problems such as multi-goal grid world, so they introduced Fuzzy Sarsa Learning (FSL) that could handle these problems efficiently. After that, they proposed an Enhanced Fuzzy Sarsa Learning algorithm called EFSL which uses an adaptive learning rate and a fuzzy balancer to manage balance between exploration and exploitation [6].

The FQL and FSL based reinforcement learning methods above focus on the design of type-1 FSs. In [7] Chia Hung Hsu and Chia Feng Juang proposed Self-Organized Interval type-2 fuzzy controller partially designed by both Q-learning and Ant Colony Optimization algorithm. They designed a self-organized fuzzy type-2 controller just by using Q-learning [8]. For more studies, A summation of automatically designed fuzzy type-2 controllers by supervised or unsupervised methods is presented in [9].

In this paper, a new approach to design interval type-2 fuzzy controllers is proposed. These new controllers apply Sarsa algorithm to tune consequent part of fuzzy type-2 rules. Two types of output processor are applied for designing Interval type-2 fuzzy controllers, one of which is Karnik-Mendel type reducer accompanied by centroid deffuzification, and the other one is Begian-Melek_Mendel method. To compare the efficiency of IT2FSL controllers with other fuzzy reinforcement learning algorithmms, they are applied to truck backing control problem. Simulation results show the efficiency of these IT2FSL controllers in comparison of other controllers.

This paper is organized as follows: Section 2 describes the structure of Interval type-2 fuzzy controllers; Section 3 describes the basic concept of fuzzy Sarsa learning followed by introducing IT2FSL; Section 4 applies the IT2FSL to reinforcement truck backing control problem; Finally, Section 5 states the conclusions and ongoing work.

2. Interval Type-2 Fuzzy Sets and Systems

2.1 Interval Type-2 Fuzzy Sets (IT2FSs)

Definition 1: [10] A type-1 fuzzy set A is composed of a district D_A of real numbers with a membership function (MF) $\mu_A \rightarrow D_A \rightarrow [0,1]$, i.e.,

$$A = \int_{D_A} \mu_A(x)/x \qquad (1)$$

Here, $\int$ stands for the set of all point $x \in D_A$ with associated membership degree $\mu_A(x)$.

Definition 2: [10] An IT2FS is described by its MF $\mu_{\tilde{A}}(x,u)$, i.e.,

$$\tilde{A} = \int_{x\in D_{\tilde{A}}} \int_{u\in J_{x\subseteq}[0,1]} \mu_{\tilde{A}}(x,u)/(x,u)$$
$$= \int_{x\in D_{\tilde{A}}} \int_{u\in J_{x\subseteq}[0,1]} 1/(x,u)$$
$$= \int_{x\in D_{\tilde{A}}} \left[\int_{u\in J_{x\subseteq}[0,1]} 1/u\right]/x \quad (2)$$

In this definition x is primary variable in $D_{\tilde{A}}$ and u is secondary variable with domain $J_x \subseteq [0,1]$ for $x \in D_{\tilde{A}}$. For all IT2FSs

$$\forall x \in D_{\tilde{A}} \text{ and } \forall u \in J_{x\subseteq}[0,1] \ , \mu_{\tilde{A}}(x,u) = 1 \quad (3)$$

2.2 Interval Type-2 Fuzzy Logic Systems (IT2FLSs)

If there is at least one fuzzy type-2 set in a rulebase, a fuzzy type-2 output processor should be used to compute crisp output. As it is illustrated in Fig. 1 an IT2FLS is similar to T1FLS, except for output processing section. Type reducer is an extra part to convert fuzzy type-2 output set to fuzzy type-1 output set before defuzzification [10]. In this paper, the A2-C0 Takagi-Sugeno-Kang rules are applied. The antecedent part of which uses interval type-2 fuzzy sets and its consequent part is composed of crisp numbers. The ith rule in the system has the following form:

$$Rule\ R^i: IF\ x_1\ is\ \widetilde{A_1^i}\ and\ \dots and\ x_I\ is\ \widetilde{A_I^i},\ THEN\ y\ is\ a_i$$
$$i = 1,2,\dots,M$$

M is the number of rules. The computation procedure of an IT2FLS includes the fuzzifier, inference engine, type reducer, and defuzzifier as in [11] and [12]. The work of each operator is sketched as follows. The fuzzifier maps crisp input values to fuzzy sets. For the ith fuzzy set $\widetilde{A_j^i}$ in input variable x_j, a triangular primary membership function is used, the boundary of which varies between the intervals $[a-d, a+d]$ and $[c-d, c+d]$. Its membership degree $\mu_{\widetilde{A_j^i}}$ is:

$$\mu_{A_j^i}(x) = \begin{cases} \frac{(x-a)}{(b-a)} & a \le x < b \\ \frac{(x-c)}{(b-c)} & a \le x < b \\ 0 & otherwise \end{cases} \quad (3)$$

Fig. 2 shows the shape of the interval type-2 fuzzy set. This figure shows that the membership degree $\mu_{\widetilde{A_j^i}}(x_j)$ is an interval set and is symbolized by $\mu_{\widetilde{A_j^i}}(x_j) = [\underline{\mu}_{\widetilde{A_j^i}}(x_j), \overline{\mu}_{\widetilde{A_j^i}}(x_j)]$, where $\underline{\mu}_{\widetilde{A_j^i}}(x_j)$ and $\overline{\mu}_{\widetilde{A_j^i}}(x_j)$ stand for lower and upper MFs, respectively, and

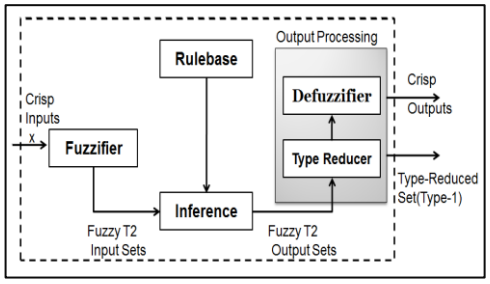

Fig. 1: Schematic diagram of an IT2FLS

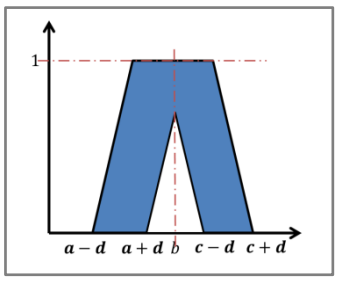

Fig. 2: Triangular interval type-2 fuzzy set

$$\overline{\mu}_{\widetilde{A_j^i}}(x_j) = \begin{cases} \frac{(x-a+d)}{(b-a)} & a-d \le x < b-d \\ 1 & b-d \le x < b+d \\ \frac{(x-c-d)}{(b-c)} & b+d \le x < c+d \\ 0 & \text{elsewhere} \end{cases} \quad (4)$$

$$\underline{\mu}_{\widetilde{A_j^i}}(x_j) = \begin{cases} \frac{(x-a-d)}{(b-a)} & a+d \le x < d\frac{c+a-2b}{c-a} + b \\ \frac{(x-a+d)}{(b-c)} & d\frac{c+a-2b}{c-a} + b \le x < c-d \\ 0 & elsewhere \end{cases} \quad (5)$$

In this paper, an algebraic product operation is assumed as T-norm for performing operations in inference engine. The result is a firing strength F_i, which is an interval type-1 fuzzy set. The firing strength is computed as follows [13]:

$$F^i = \left[\underline{f^i}, \overline{f^i}\right] \quad (6)$$

Where

$$\overline{f^i} = \prod_{j=1}^{n} \overline{\mu}_{\widetilde{A_j^i}} \ and \ \underline{f^i} = \prod_{j=1}^{n} \underline{\mu}_{\widetilde{A_j^i}} \quad (7)$$

Resolution principle and center-of-sets-type reduction operations are executed by type reducer [13]. Based on these operations and the expression in equation (1), the type reduced set is computed as follows:

$$Y(x) = [y_l, y_r] =$$
$$\int_{a_1} \dots \int_{a_M} \int_{f^1\in[\underline{f^1},\overline{f^1}]} \dots \int_{f^M\in[\underline{f^M},\overline{f^M}]} 1 \Big/ \frac{\sum_{i=1}^{M} f^i a_i}{\sum_{i=1}^{M} f^i} \quad (8)$$

where indices l and r depict the left and right boundaries, respectively. The outcome set of type reducer is an interval type-1 fuzzy set. Equation (8) can not be solved directly. The calculation of the reduced set requires an iterative procedure. In many implementations, equation (8) is computed by using the Karnik-Mendel iterative procedure, which consists of an initialization step followed by four iterative steps. Details of this procedure can be found in [13].

For calculating the system output variable y, the defuzzification operation would be done on the output of type reducer, i.e. interval set $[y_l, y_r]$.

Based on the centroid defuzzification operation, the centroid of the interval set $[y_l, y_r]$ is the average of y_l and y_r. Hence, the defuzzified output is

$$y = \frac{y_l + y_r}{2} \quad (9)$$

To simplify the process of computing iterative procedure, many equivalent formulas have been introduced for Karnik-Mendel type reducer. In [14] there is a summarized description of these simplified KM methods that are a combination of type-reducer and defuzzifier.

In this paper, in addition to KM procedure, the Begian-Melek-Mendel (BMM) method is used to compute crisp output of IT2FLS[15].

The Begian-Melek-Mendel (BMM) Method : Begian, Melek and Mendel proposed following closed-form type-reduction and defuzzification method for TSK-IT2FLSs, i.e.,

$$y = m\frac{\sum_{n=1}^{M}\underline{f^n}y^n}{\sum_{n=1}^{M}\underline{f^n}} + n\frac{\sum_{n=1}^{M}\overline{f^n}y^n}{\sum_{n=1}^{M}\overline{f^n}} \quad (10)$$

Where m and n are adjustable coefficients [15].

3. Interval Type-2 Fuzzy Sarsa Learning

This section describes the main purpose of this paper, designing an interval type-2 fuzzy controller by sarsa learning algorithm. At first, it should be cleared that there have been two points of view about fuzzy reinforcement learning algorithms. One point is that Q-learning or Sarsa learning are used to justify fuzzy controllers, and the other one is that fuzzy systems can act as a function approximator for estimating state spaces to prevent curse of dimensionality. However, the result of one sight is the same as the other one.

In summary, designing an automatic fuzzy controller by RL is a process where RL algorithms such as Q-learning or Sarsa learning make changes only in rulebase by communicating with the environment until they attain the best sate-action values. So, other parts of fuzzy system such as inference engine and output processing sections are fixed and the fuzzifier may change in some implementations such as DFQL. Following, FSL and proposed IT2FSL controllers are described more precisely.

3.1 Fuzzy Sarsa Learning[5]

According to the equations (11) and (12), Sarsa algorithm evaluates the value of action a in state s symbolized by $Q(s, a)$ for the current policy.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \times \Delta Q \quad (11)$$

$$\Delta Q = [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (12)$$

Here, α is the learning rate, γ is the discount factor, and r_{t+1} is the reward received from the environment after executing action a_t in state s_t. In a fuzzy logic system with multi input-single output structure, a zero order TSK fuzzy type-1 rule is as below:

$$Rule\ R^i\text{: } IF\ x_1\ is\ A_1^i\ and\ \dots and\ x_n\ is\ A_n^i, THEN\ y\ is$$

$$(a_{i1}\ with\ value\ q_{i1})\ or\ \dots or\ (a_{im}\ with\ value\ q_{im})$$

To tune fuzzy rules, q values should be adjusted based on the evaluation of corresponding action a. Where $x_i, (i = 1 \dots . n)$ is input state and A_n^i is the normal fuzzy set, m is the number of possible actions for each rule, a_{ij} and q_{ij} are the j-th selected action and the approximate value of the j-th action in the i-th rule, respectively. The algorithm of FSL is summarized as below [6]:

1) Calculate state s_{t+1} and receive reward or punishment r_{t+1}.
2) Choose an appropriate action from each rule using $\varepsilon - greedy$ action selection.
3) Calculate final action a_{t+1} and the approximate action value function $\hat{Q}_t(s_{t+1}, a_{t+1})$ using equations (13) and (14), respectively.

$$a_t(s_t) = \sum_{i=1}^{R}\mu_i(s_t)a_{ii^+} \quad (13)$$

$$\hat{Q}_t(s_t, a_t) = \sum_{i=1}^{R}\mu_i(s_t)q_{ii^+} \quad (14)$$

4) Calculate $\Delta\hat{Q}$ and update q by equations (13) and (14), respectively.

$$\Delta q_{t+1}^{ij} = \begin{cases}\alpha_t \times \Delta\hat{Q}_t(s_t, a_t) \times \mu_i(s_t)\ , j = i^+ \\ 0 \qquad\qquad otherwise\end{cases} \quad (15)$$

$$\Delta\hat{Q}_t(s_t, a_t) =$$

$$r_{t+1} + \gamma\hat{Q}_t(s_{t+1}, a_{t+1}) - \hat{Q}_t(s_t, a_t) \quad (16)$$

5) Calculate new approximate action value function $\hat{Q}_{t+1}(s_{t+1}, a_{t+1})$ using equation (13).
6) Execute the final action.
7) $t \leftarrow t + 1$ and return to first step.

3.2 IT2FSL Controller

Fig. 3 depicts the interval type-2 fuzzy Sarsa learning framework. In this paper, two kinds of IT2FSL controllers are proposed. The first one consists of KM type reducer and centroid deffuzification (called IT2FSL-KM) and the other one is based on BMM method (called IT2FSL-BMM). The controller designing procedure is similar to fuzzy Sarsa learning. Consider a rule base constructed of zero order TSK fuzzy type-2 rules in which fuzzy type-2 sets are applied. In order to calculate interval firing strength, we use equations (6) and (7). After action selection for each activated rule, the output of fuzzy type-2 controller can be computed by using BMM method or KM method.

Using BMM method is less complicated than KM method. Consequently, it is more perceptible. IT2FSL with BMM output processor is as below:

Compute final action a_{t+1} and the approximate action value function $\hat{Q}_t(s_{t+1}, a_{t+1})$ using equation (17) and (18), respectively.

$$a_t(s_t) = m\frac{\sum_{i=1}^{M}\underline{f}_i a_{ii}}{\sum_{i=1}^{M}\underline{f}_i} + n\frac{\sum_{i=1}^{M}\overline{f}_i a_{ii}}{\sum_{i=1}^{M}\overline{f}_i} \quad (17)$$

$$\hat{Q}_t(s_t, a_t) = m\frac{\sum_{i=1}^{M}\underline{f}_i q_{ii}}{\sum_{i=1}^{M}\underline{f}_i} + n\frac{\sum_{i=1}^{M}\overline{f}_i q_{ii}}{\sum_{i=1}^{M}\overline{f}_i} \quad (18)$$

Calculate firing strength of ith rule by equation (19). Compute $\Delta\hat{Q}$ and update q by equations (20) and (21), respectively.

$$\mu_i(s_t) = \frac{m \times \underline{f}_i(s_t) + n \times \overline{f}_i(s_t)}{m+n} \quad (19)$$

$$\Delta q_{t+1}^{ij} = \begin{cases} \alpha_t \times \Delta \hat{Q}_t(s_t, a_t) \times \mu_i(s_t), j = i^+ \\ 0 \qquad otherwise \end{cases} \quad (20)$$

$$\Delta \hat{Q}_t(s_t, a_t) = r_{t+1} + \gamma \hat{Q}_t(s_{t+1}, a_{t+1}) - \hat{Q}_t(s_t, a_t) \quad (21)$$

For final steps calculate new approximate action value function $\hat{Q}_{t+1}(s_{t+1}, a_{t+1})$ using equation (17) and execute the final action. Replace t with $t+1$ and return to first step.

In IT2FSL-KM actions and action values are calculated by using KM procedure instead of BMM formula. For more details about KM refer to [7] and [8].

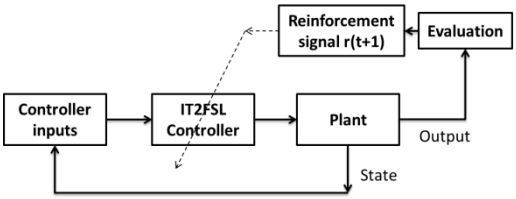

Fig. 3: The interval type-2 fuzzy Sarsa learning framewor

4. SIMULATION

Truck backing control problem is an appropriate problem to check the validity of the proposed controller. In this problem [8], there are three variables changing in following ranges:

1. $x \in [-15,35]$; The horizontal position of the truck
2. $y \in [\alpha, \beta]$; The vertical position of the truck, α, β are arbitrary. For example, in this paper they are assumed [0,160].
3. $\varphi \in [0°, 180°]$; Angle of the truck with the horizontal axis

The agent (driver) is able to control the truck by changing the steering angle θ on the interval [-40, 40]. Fig. 4 shows input variables (x, y, φ) and output variable θ. It should be noticed that in this problem, only backing up is considered, and the truck moves backward by a fixed unit distance in every time step.

For simplifying the problem, y is ignored as an input for fuzzy controller [8]. The truck system dynamics are

$$x(t+1) = x(t) + \cos[\varphi(t) + \theta(t)] + \sin[\theta(t)] \sin[\varphi(t)]$$

$$y(t+1) = y(t) + \sin[\varphi(t) + \theta(t)] - \sin[\theta(t)] \sin[\varphi(t)]$$

$$\varphi(t+1) = \varphi(t) - sin^{-1}[2\sin(\theta(t))/b] \quad (22)$$

The main objective of the controller is to back up the truck to an appropriate position with an appropriate truck angle. For this purpose $x \in [9,11]$ and $\varphi \in [80^{\circ}, 100^{\circ}]$ are defined as desired situation.

The agent has to learn to back up the truck in less than 80 time steps and it must keep driving in the desired situation for 150 time steps after entering the desired situation region. For reaching the best policy, it is assumed that this process should be repeated 10 times without any interruption. If the truck goes out of the desired situation, it will receive a punishment $r(t+1) = -1$, if not; a reward $r(t+1) = 0$ is received at each time step. In IT2FSL, the set of actions is $\{-40, -35, \ldots, 0, \ldots, 35, 40\}$.

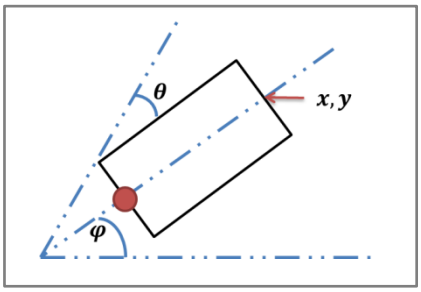

Fig. 4: x, y and φ are input variables of truck system dynamics and θ is the output variable.

For testing the efficiency of the obtained policy, this problem has two phases, training phase and testing phase. In training phase there are three initial states $[3, 135]$, $[12, 45]$ and $[18, 30]$. For testing phase there are two new initial states $[6, 120]$ and $[9, 60]$. The root mean square error (RMSE) between the truck position and the desired position $x = 10$ after 100 time steps for each of the two initial states in test phase is calculated by following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{2} \sum_{t=101}^{150} (x_e^i(t) - 10)^2}{100}} \quad (23)$$

Current study simulates 100 runs. Each run consists of 7500 trials. A trial means an episode in which controller starts backing up the truck from an initial state and stops when reaches the desired situation. At the end of each run, there is a successful or an unsuccessful controller. If the agent does not find a successful controller by 7500 trials, the resultant controller will be an unsuccessful controller.

In this study, for each IT2FSL controller, its equivalent off-policy controller by Q-learning is simulated to compare the effectiveness of proposed controllers with that of IT2FQL controllers.

To evaluate the robustness of the IT2FSL controllers in presence of noise, they are examined in noiseless and noisy environments. They are compared with four other controllers FQL, FSL, IT2FQL-BMM and IT2FQL-KM. The considerable point is that the following uniform noise is added to inputs in test phase, which means they are not noisy in training phase.

$$x \in [-1,1] \text{ and } \varphi \in [-4.5, 4.5]$$

For simulating IT2FSL controllers in learning phase, γ=0.9,$\alpha_t = 0.1$ and ε=0.001 are considered. For BMM output processor $m = 1$ and $n = 2$ is achieved by trial and error. The number of membership functions for each variable is seven, so there are 49 rules in rulebase. Triangular interval type-2 fuzzy sets are illustrated in

Fig. 5. To avoid gap or jump discontinuities in output, the input domain is completely covered by lower membership functions and upper membership functions [16]. Fig. 6 shows x and φ values in $x-\varphi$ plane. This figure relates to noiseless environment. The trajectory of the truck using an IT2FSL-KM in noiseless environment and in noisy environment is illustrated in Fig. 7 and Fig. 8, respectively.

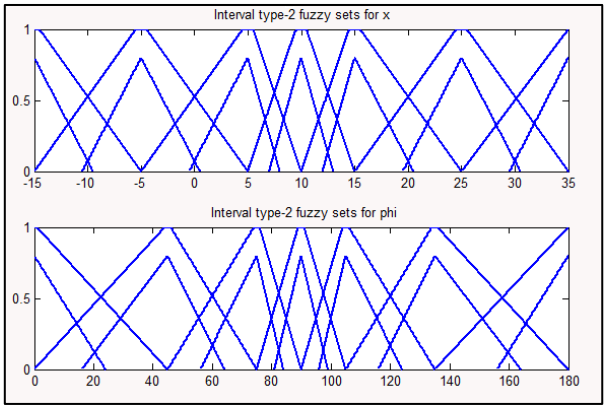

Fig. 5: Lower membership functions and upper membership functions for input variables

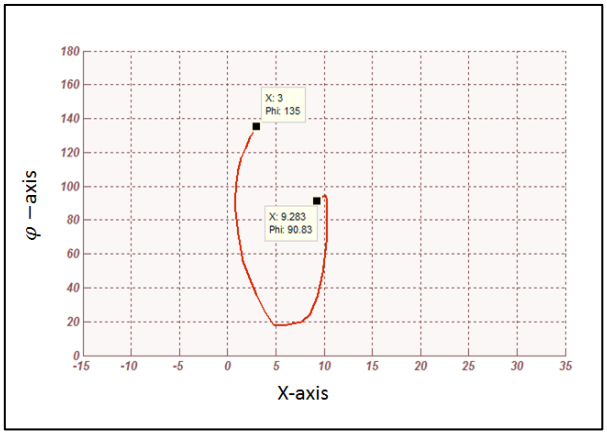

Fig. 6: Truck position in $x-\varphi$ plane, the initial state is $x=3$ and $\varphi=135$, truck stops in $x=9.28$ and $\varphi=90.83$

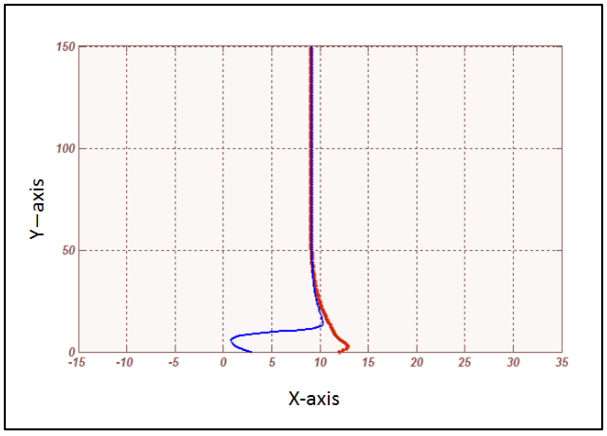

Fig. 7: The trajectory of the truck achieved by IT2FSL-KM controller in noiseless environment

The performance of the controllers is compared with four other controllers in two tables. Table-1 demonstrates the performance in noiseless environment. As we can see, the minimum value of Average of Average Time Steps is 241.41 that relates to IT2FSL-BMM. It means the proposed controller has highest speed in backing up the truck among other controllers. Considering table-2, it also has an acceptable speed in noisy environment.

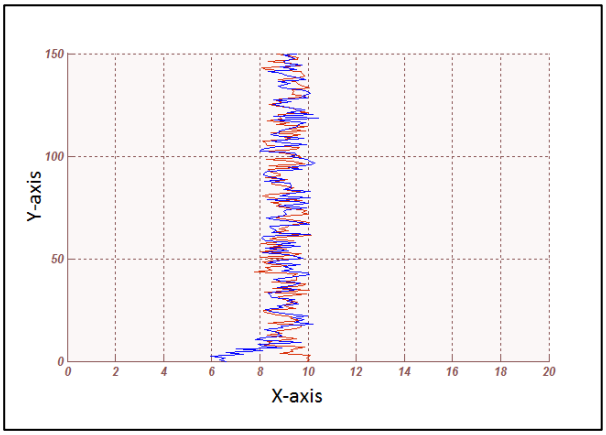

Fig. 8: The trajectory of the truck achieved by IT2FSL-KM controller in noisy environment

By Comparing RMSE values in each environment, it is considered that Average RMSE in noisy environment is 0.51 for IT2FSL-BMM, which is the lowest one, but by using t-test for $\alpha=0.01$ and $number\ of\ samples=100$ the hypothesis that IT2FSL has the most robustness in presence of noise is failed. It can be seen that average RMSE in noisy environment is not much lower than other controllers, but we cannot deduce that it is not robust in presence of noise. Because among minimum RMSE values, the lowest one stands for proposed controller, which means Sarsa learning can design a fuzzy type-2 controller that outperforms other controllers. Nevertheless, there is no guarantee to achieve this successful controller in each run. Because in designing procedure, there is no adjustable parameter associated with robustness of controller; all literatures express that fuzzy type-2 systems are more robust than fuzzy type-1 controllers; although, when they are designed by reinforcement learning methods the noise resistance ability of fuzzy type-2 controllers with proposed structure is not much higher than noise robustness of fuzzy type-1 controllers.

5. CONCLUSION

In this study, IT2FSL-KM and IT2FSL-BMM are proposed. These new controllers are independent of collecting training input-output data set like other fuzzy controllers designed by RL methods. Since it is a hard task and even impossible, these new controllers are designed on-line by a weak reinforcement signal.

They are compared with other controllers designed automatically by RL methods. The speed of convergence in proposed controllers is higher than FQL and is as well as FSL. The proposed algorithm can back up the truck in minimum time steps. Designed fuzzy type-2 controllers

are as robust as FSL in presence of noise. This study shows RL methods are capable of designing fuzzy type-2 controllers, but there is no guarantee for high robustness in comparison of fuzzy type-1 controllers. Investigating effective factors in designing robust fuzzy type-2 controllers by RL methods will be studied in the future.

Table 1: Comparisons of IT2FSL controllers with other reinforcement controllers in noiseless environment

Methods		FQL	FSL	IT2FQL-KM	IT2FQL-BMM	IT2FSL-KM	IT2FSL-BMM
Trial	Mean	1159	539	663.6	655.92	605.11	685.4
	Std	270.6	94.4	171.81	173.88	102.59	77.6
Average of Time Step	Mean	264.69	306	267.61	259.06	259.67	251.3
	Std	45.76	31.4	37.59	35.49	27.04	45.29
RMSE	Mean	0.57	0.54	0.629	0.6811	0.6739	0.515
	Std	0.09	0.29	0.162	0.3472	0.3722	0.442
	Minimum	0.514	1.00E-09	0.0004	1.6835E-07	8.44	2.445E-07
Success Percentage		73.68	84.24	77.51	83.27	84.28	84.09

Table 2: Comparisons of IT2FSL controllers with other reinforcement controllers in noisy environment

Methods		FQL	FSL	IT2FQL-KM	IT2FQL-BMM	IT2FSL-KM	IT2FSL-BMM
Trial	Mean	1884.87	545.61	677.83	697.44	751.25	664.26
	Std	647.37	141.21	219.56	154.71	91.38	39.26
Average of Time Step	Mean	270.25	305.52	260.11	246.39	239.34	241.41
	Std	72.24	40.46	35.41	30.18	51.33	44.73
RMSE	Mean	0.791	0.763	0.734	0.839	0.822	0.787
	Std	0.009	0.127	0.131	0.113	0.144	0.148
	Minimum	0.584	0.535	0.552	0.529	0.561	0.514
SuccessPercentage		71.53	79.43	83.11	80.45	60.43	81.34

REFRENCES

[1] A. Bonarini, A. Lazaric, F. Montrone, and M. Restelli, "Reinforcement distribution in fuzzy Q-learning," *Fuzzy sets and systems,* vol. 160, pp. 1420-1443, 2009.

[2] P. Y. Glorennec and L. Jouffe, "Fuzzy Q-learning," in *Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on*, 1997, pp. 659-662.

[3] L. Jouffe, "Fuzzy inference system learning by reinforcement methods," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 28, pp. 338-355, 1998.

[4] M. J. Er and C. Deng, "Online tuning of fuzzy inference systems using dynamic fuzzy Q-learning," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 34, pp. 1478-1489, 2004.

[5] V. Derhami, J. V. Majd, and M. N. Ahmadabadi, "Fuzzy Sarsa learning and the proof of existence of its stationary points," *Asian Journal of Control,* vol. 10, pp. 535-549, 2008.

[6] V. Derhami, J. V. Majd, and M. N. Ahmadabadi, "Exploration and exploitation balance management in fuzzy reinforcement learning," *Fuzzy sets and systems,* vol. 161, pp. 578-595, 2010.

[7] C. H. Hsu and C. F. Juang, "Self-Organizing Interval Type-2 Fuzzy Q-learning for reinforcement fuzzy control," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2011, pp. 2033-2038.

[8] C. F. Juang and C. H. Hsu, "Reinforcement interval type-2 fuzzy controller design by online rule generation and Q-value-aided ant colony optimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 39, pp. 1528-1542, 2009.

[9] O. Castillo and P. Melin, "A review on the design and optimization of interval type-2 fuzzy controllers," *Applied Soft Computing,* 2011.

[10] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *Fuzzy Systems, IEEE Transactions on,* vol. 10, pp. 117-127, 2002.

[11] D. Wu, "A brief Tutorial on Interval type-2 fuzzy sets and systems," *Fuzzy sets and systems,* 2010.

[12] D. Wu and J. M. Mendel, "Enhanced Karnik-Mendel Algorithms for interval type-2 fuzzy sets and systems," in *Fuzzy Information Processing Society, 2007. NAFIPS'07. Annual Meeting of the North American*, 2007, pp. 184-189.

[13] Q. Liang, N.N. Karnik, and J.M. Mendel, "Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 30, pp. 329-339, 2000.

[14] M. B. Begian, W. W. Melek, *and* J.M. Mendel,"Stability analysis of type-2 fuzzy systems," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, 2008, pp. 947-953.

[15] M. Biglarbegian, W. W. Melek, *and* J.M. Mendel,"On the Stability of Interval Type-2 TSK Fuzzy Logic Control Systems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 40, pp. 798-818, 2010.

16] D. Wu and J. M. Mendel, "Examining the continuity of type-1 and interval type-2 fuzzy logic systems,"" in *Proc. IEEE World Congress on Computational Intelligence*, 2010.