Q(λ)-Learning Fuzzy Controller for the Homicidal Chauffeur Differential Game

Badr M. Al Faiya and Howard M. Schwartz

Abstract— **In this paper, a Q(λ)-learning fuzzy inference system (QLFIS) is applied to a differential game. We use the homicidal chauffeur differential game as an example of the method. The suggested method allows both the evader and the pursuer to learn their optimal strategies. The parameters of the input and the fuzzy rules of a fuzzy controller are tuned autonomously using Q(λ)-learning. Simulation results demonstrate that the players are able to learn their optimal strategies.**

I. INTRODUCTION

In differential games, players need the ability to learn, adapt and interact with an unknown environment. Reinforcement learning (RL) has been used to train the players. A common reinforcement learning technique called Q-learning is introduced in [1]–[3]. In [4] and [5], a reinforcement fuzzy learning is applied to the pursuit-evasion differential game. A fuzzy Q(λ)-learning technique is presented in [4], [6] and a fuzzy actor-critic method is presented in [5]. The authors showed that RL can be used to teach the pursuer to capture the evader and minimize the capture time. However, the evader did not learn to escape from the pursuer. The evader needs to learn from rewards and perform an optimal control action. In this paper, we introduce a technique to make both the evader and the pursuer learn their optimal strategies simultaneously using reinforcement learning.

RL is used train players to learn complex behavior through interactions with the environment without supervision or a teacher [1]. RL also plays an important role in adaptive control. Recently, RL has been applied to train players in differential games [6]–[8]. An interacting learner or player receives feedback as rewards and punishments from the world (environment). The player then learns to perform optimally based on the feedback. Q-learning is one of the common reinforcement learning techniques [1]–[3].

Q-learning estimates the expected rewards received in the future given the current state-action pair. Q-learning is generally used in the case where the state space and the action space are both discrete. In some situations, such as differential games, it is impractical to discretize the state space and the action space [4], [9]. In order to use a RL technique such as Q-learning in a continuous space, one can apply fuzzy reinforcement learning to differential games and use fuzzy systems to represent the continuous state space and action space [10]–[12].

We apply reinforcement fuzzy learning technique to the homicidal chauffeur game. The fuzzy logic controller (FLC) input and output parameters are tuned using Q(λ)-learning. This approach is based on the methods proposed by Desouky et al. [4], [6] and Givigi et al. [5]. We extended the game by adding the distance as an input to the FLC for the evader. Moreover, the capture condition for the game is investigated when training the players. The evader learns to take the appropriate action whenever the pursuer reaches some threshold distance. The trained evader learns to find this distance and to make sharp turns (extreme strategy) to avoid being captured, or maximize the capture time if the capture must occur. At the same time, the pursuer learns to capture the evader. To evaluate and validate our results, the theoretical solution of the game is illustrated.

In this paper, we first introduce the homicidal chauffeur differential game in the next section. The fuzzy controller structure is described in Section III. In Section IV, we describe the fuzzy reinforcement learning technique. Simulation results are presented in Section V. Finally, conclusions are presented in Section VI.

II. HOMICIDAL CHAUFFEUR DIFFERENTIAL GAME

Differential games (DG) [13] are a family of dynamic, continuous time games. The homicidal chauffeur differential game is one type of differential game. It was originally presented by Isaacs in 1954. Isaacs defined the "Homicidal Chauffeur Problem" in a Rand technical report [14]. The game has been extended to include more general pursuit-evasion problems/games [13]. A pursuer or a group of pursuers attempts to capture one evader or a group of evaders in minimal time while the evaders try to avoid being captured.

The game terminates when the evader is within the lethal range of the pursuer (capture or termination time), or when the time exceeds one minute (escape). Players evaluate the current state and then select their next actions. The players' strategies are not shared and therefore each player has no knowledge of the other player's next selected action. We assume that the environment is obstacle-free.

The existence of optimal strategies in the pursuit-evasion differential game is determined by Isaacs condition [13], [15], [16]. The formal results of optimal strategies for pursuit-evasion differential games are given in [13], [17]. The homicidal chauffeur game and Isaacs condition for the game are discussed below.

B. Al Faiya is with the Department of Systems and Computer Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada abadr@sce.carleton.ca

H. M. Schwartz is with Faculty of Systems and Computer Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada schwartz@sce.carleton.ca

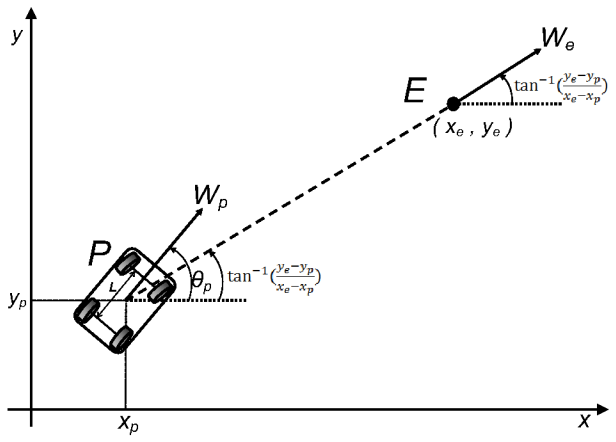


Fig. 1: Homicidal chauffeur problem model

In our game, a homicidal chauffeur game is played by autonomous robots. The chauffeur (the pursuer P) is a car-like mobile robot and the pedestrian (the evader E) is a point that can move in any direction instantaneously. In Isaacs' homicidal chauffeur differential game, a pursuer aims to minimize the capture time of an evader. The evader's objective is to maximize the capture time and avoid capture.

We assume that the players move at a constant forward speed w_i. The pursuer's speed is greater than the evader's speed, but the evader can move in any direction instantaneously. The steering angle of the pursuer is given as $-u_{p_{max}} \leq u_p \leq u_{p_{max}}$, where $u_{p_{max}}$ is the maximum steering angle. The maximum steering angle results in a minimum turning radius R_p defined by

$$R_p = \frac{L_p}{\tan(u_{p_{max}})} \tag{1}$$

where L_p is the pursuer's wheelbase.

The dynamic equations for the pursuer P and the evader E are [13]

$$\begin{aligned} \dot{x}_p &= w_p \cos(\theta_p) \\ \dot{y}_p &= w_p \sin(\theta_p) \\ \dot{\theta}_p &= \frac{w_p}{R_p} u_p \\ \dot{x}_e &= w_e \cos(u_e) \\ \dot{y}_e &= w_e \sin(u_e) \end{aligned} \tag{2}$$

where (x,y), w, and θ denote the position, the velocity, and the orientation respectively as shown in Fig.1.

The angle difference ϕ between the pursuer and the evader is given as

$$\phi = \tan^{-1}(\frac{y_e - y_p}{x_e - x_p}) - \theta_p \tag{3}$$

The relative distance between pursuer and evader is found as

$$d = \sqrt{(x_e - x_p)^2 + (y_e - y_p)^2} \tag{4}$$

The capture occurs when the distance $d \leq \ell$ where ℓ is the capture radius.

In ([13], p. 232-237), Isaacs presented a condition such that the pursuer can succeed in capturing the evader. Assuming that the pursuer's speed is greater than the evader's speed, the capture condition is given as

$$l/R_p > \sqrt{1-\gamma^2} + \sin^{-1}\gamma - 1 \tag{5}$$

where ℓ/R_p is the ratio of the radius of capture to the minimum turning radius of the pursuer, and $\gamma = w_e/w_p < 1$ is the ratio of the evader's speed to the pursuer's speed. If inequality (5) is reversed, E escapes from P indefinitely.

Based on the capture condition in (5) and Isaacs' solution of the game, the evader's optimal strategy can be obtained by solving the following two problems [13], [18], [19]:

1- When the evader is far enough from the pursuer, the evader's control strategy is to maximize the distance between the evader and the pursuer as follows

$$u_e = \tan^{-1}\frac{y_e - y_p}{x_e - x_p} \tag{6}$$

2- When the pursuer approaches the evader such that $d \leq R_p$, the evader adopts a second control strategy to avoid capture. The pursuer cannot turn more than a minimum turning radius R_p. The evader will make a sharp turn, normal to its direction, and enter the pursuer's non-holonomic constraint region. As shown in Fig.2, a non-holonomic player is constrained to move along a path with a bounded curvature such as the pursuer's minimum turning radius R_p given in Eq.(1). The evader's second control strategy is given as

$$u_{e_{extreme}} = \theta_e \pm \pi/2 \tag{7}$$

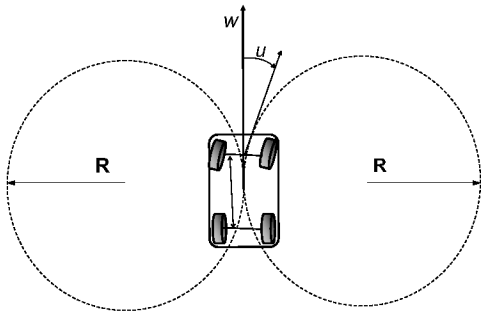


Fig. 2: The vehicle cannot turn into the circular region defined by its minimum turning radius R.

The pursuer's optimal control strategy is to minimize the distance and capture the evader in minimum time. The

pursuer controls its steering angle as follows [4], [16], [17]

$$u_p = \tan^{-1}\left(\frac{y_e - y_p}{x_e - x_p}\right) - \theta_p \tag{8}$$

III. FUZZY CONTROLLER STRUCTURE

We use two inputs (fuzzy variables) for the fuzzy controller and generate one output from the fuzzy controller. The inputs for the pursuer are the angle difference ϕ and its rate of change $\dot{\phi}$. The inputs for the evader are the angle difference ϕ and the distance d. In this paper, we add the distance as an input to the fuzzy controller for the evader. The reason is that the evader has higher maneuverability than the pursuer and the distance between the evader and the pursuer is critical for the evader to decide if it needs to make a sharp turn.

For simplicity and to avoid the "curse of dimensionality" problem, we use two inputs and three fuzzy sets for each input to construct the controller. The pursuer's fuzzy sets are positive (P), zero (Z) and negative (N) for the angle difference ϕ and its derivative $\dot{\phi}$. The evader's fuzzy sets are positive (P), zero (Z) and negative (N) for the angle, and far (F), close (C) and very close (V) for the distance.

We apply a zero-order Takagi-Sugeno (TS) fuzzy inference system (FIS) [20]. TS FIS consists of fuzzy IF-THEN rules and a fuzzy inference engine. Given the fuzzy variables x_i and the corresponding fuzzy sets A_i and B_i, the fuzzy IF-THEN rules are

$$\Re_l : IF\ x_1\ is\ A_l\ AND\ x_2\ is\ B_l\ THEN\ f_l = K^l \tag{9}$$

where x_i represents ϕ and $\dot{\phi}$ for the pursuer, ϕ and d for the evader. The term f_l is the output function of rule l and K^l is the parameter for the consequence part of the fuzzy rules.

Three membership functions (MF) are used for each input which results in constructing $3^2 = 9$ rules. The Gaussian MFs are given as

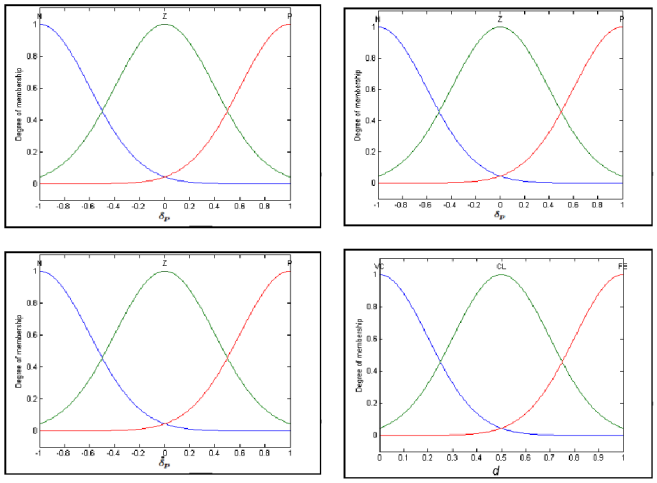
$$\mu_{A_l}(x_i) = \exp\left(-\left(\frac{x_i - c_i^l}{\sigma_i^l}\right)^2\right) \tag{10}$$

The Gaussian MFs parameters are the mean c and the standard deviation σ, which are the input parameters to be tuned by RL signals. Figures 3a and 3b show the initial MFs before tuning.

The steering angle u is the output formed by the weighted average defuzzifier expressed as

$$u = \frac{\sum_{l=1}^{9}\left(\left(\prod_{i=1}^{2}\mu_{A_l}(x_i)\right)K^l\right)}{\sum_{l=1}^{9}\left(\prod_{i=1}^{2}\mu_{A_l}(x_i)\right)} \tag{11}$$

The fuzzy rules are illustrated using the tabular format. Tables I and II show the fuzzy decision table and the output constant K^l for the pursuer and the evader respectively before learning.



(a) The membership functions of the pursuer before training.

(b) The membership functions of the evader before training.

Fig. 3: Membership functions before training

TABLE I: The pursuer's fuzzy decision table and the output constant K^l before learning

ϕ \ $\dot{\phi}$	N	Z	P
N	-0.5	-0.25	0.0
Z	-0.25	0.0	0.25
P	0.0	0.25	0.5

IV. REINFORCEMENT LEARNING

A learning agent in a reinforcement learning problem interacts with the environment and receives a reward r_t at each time step t. The agent's goal is to maximize the long run discounted return R_t [1]

$$R_t = \sum_{k=0}^{T}\gamma^k r_{t+k+1} \tag{12}$$

where $(0 \le \gamma \le 1)$ is the discount-factor, t is the current time step, and T is the episode terminal time.

One common type of reinforcement learning is Q-learning. Q-learning estimates the action-value function $Q(s,a)$ to achieve the best expected return. The action-value function is given as

$$Q(s,a) = E\left\{\sum_{k=0}^{\infty}\gamma^k r_{k+t+1} | s_t = s, a_t = a\right\} \tag{13}$$

where s is the state and a is the action.

A. Q(λ)-learning Fuzzy Inference System

Desouky et al. [4] proposed a Q(λ)-learning fuzzy inference system QLFIS technique. In the QLFIS technique, the controller and the function approximator are represented by fuzzy systems. Q(λ)-learning is used to tune the input and the output parameters of the fuzzy logic controller (FLC) and the function approximator implemented by FIS. The advantage of this QLFIS technique is that one can use Q-learning in

TABLE II: The evader's fuzzy decision table and the output constant K^l before learning

ϕ \ d	VC	CS	FA
N	$-\pi/2$	$-\pi/2$	$-\pi/4$
Z	$-\pi/2$	$\pi/2$	0.0
P	$\pi/2$	$\pi/2$	$\pi/4$

a continuous domain by using a fuzzy inference system to represent the continuous state space and action space.

In [4], QLFIS was successfully applied to train the pursuer to capture the evader in minimum time, but the evader did not learn. Moreover, the capture condition for the game has not been investigated when training the players. In this paper, we apply QLFIS algorithms for the homicidal chauffeur game to train both the evader and the pursuer.

The construction of the learning system is shown in Fig. 4. Desouky et al. [4] derived and presented the update rules of the learning process.

As shown in Fig.4, the TD error δ_t is given as

$$\delta_t = r_{t+1} + \gamma \max_{\acute{a}} Q_t(s_{t+1}, \acute{a}) - Q_t(s_t, a_t) \tag{14}$$

The Q(λ)-learning action-value function in is updated by

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \delta_t \, e_t \tag{15}$$

where $(0 < \alpha \leq 1)$ is the learning rate, and the replacing eligibility traces e_t with the trace decay parameter $(0 \leq \lambda \leq 1)$ is represented as

$$e_t = \gamma \lambda e_{t-1} + \frac{\partial Q_t(s_t, a_t)}{\partial \xi} \tag{16}$$

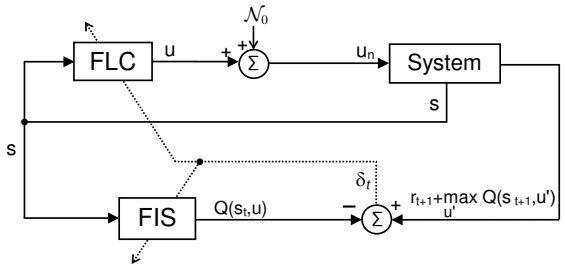


Fig. 4: Construction of the learning system where the white Gaussian noise $\mathcal{N}(0, \sigma_n^2)$ is added as an exploration mechanism.

Given the parameters $\xi = [K \; c \; \sigma]^\top$ to be tuned in the fuzzy systems, the update rules for the FIS are defined by [4]

$$\xi_{FIS}(t+1) = \xi_{FIS}(t) + \eta \delta_t \left[\gamma \lambda e_{t-1} + \frac{\partial Q_t(s_t, u_t)}{\partial \xi_{FIS}} \right] \tag{17}$$

and the update rules for the FLC are defined by

$$\xi_{FLC}(t+1) = \xi_{FLC}(t) + \zeta \delta_t \left[\frac{\partial u}{\partial \xi_{FLC}} \left(\frac{u_n - u}{\sigma_n} \right) \right] \tag{18}$$

where

$$\frac{\partial Q_t(s_t, u_t)}{\partial \xi_{FIS}} = \begin{bmatrix} \frac{\partial Q_t(s_t, u_t)}{\partial K^l} \\ \frac{\partial Q_t(s_t, u_t)}{\partial \sigma_i^l} \\ \frac{\partial Q_t(s_t, u_t)}{\partial c_i^l} \end{bmatrix} = \begin{bmatrix} \sum_l \bar{\omega}_l \\ \frac{(K^l - Q_t(s_t, u_t))}{\sum_l \omega_l} \omega_l \frac{2(x_i - c_i^l)}{(\sigma_i^l)^2} \\ \frac{(K^l - Q_t(s_t, u_t))}{\sum_l \omega_l} \omega_l \frac{2(x_i - c_i^l)^2}{(\sigma_i^l)^3} \end{bmatrix} \tag{19}$$

$$\frac{\partial u}{\partial \xi_{FLC}} = \begin{bmatrix} \frac{\partial u}{\partial K^l} \\ \frac{\partial u}{\partial \sigma_i^l} \\ \frac{\partial u}{\partial c_i^l} \end{bmatrix} = \begin{bmatrix} \sum_l \bar{\omega}_l \\ \frac{(K^l - u)}{\sum_l \omega_l} \omega_l \frac{2(x_i - c_i^l)^2}{(\sigma_i^l)^3} \\ \frac{(K^l - u)}{\sum_l \omega_l} \omega_l \frac{2(x_i - c_i^l)}{(\sigma_i^l)^2} \end{bmatrix} \tag{20}$$

with the learning rate η for the FIS and ζ for the FLC. The firing strength ω_l and the normalized firing strength $\bar{\omega}_l$ of rule l are defined as follows [4]

$$\omega_l = \prod_{i=1}^{2} \exp\left(-\left(\frac{x_i - c_i^l}{\sigma_i^l} \right)^2 \right) \tag{21}$$

$$\bar{\omega}_l = \frac{\omega_l}{\sum_{l=1}^{9} \omega_l} \tag{22}$$

The learning algorithm used in our simulation is shown in Algorithm 1, where M is the number of episodes (games) and N is the number of steps (plays) in each episode.

Algorithm 1 QLFIS Algorithm.

MFs $\leftarrow$ Fig.3
$K^l \leftarrow$ Tables I and II
$Q(s,u) \leftarrow 0$ {FIS Q-values}
$e \leftarrow 0$ {eligibility traces of the FIS}
$\gamma \leftarrow 0.95$; $\lambda \leftarrow 0.9$; $\sigma_n \leftarrow 0.08$
for $i \leftarrow 1$ **to** M **do**
  $\eta \leftarrow (0.1 - 0.09\,(\frac{i}{\mathrm{M}}))$
  $\zeta \leftarrow (0.01 - 0.009\,(\frac{i}{\mathrm{M}}))$
  $(x_p, y_p) \leftarrow (0,0)$ {pursuer initial position}
  initialize (x_e, y_e) randomly {evader initial position}
  update $s_p = (\phi, \dot{\phi})$
  update $s_e = (\phi, d)$
  $u \leftarrow$ Eq. (11) {for the pursuer and the evader}
  for $j \leftarrow 1$ **to** N **do**
    $u_n \leftarrow u + \mathcal{N}_0$ {for the pursuer and the evader}
    $Q(s_t, u) \leftarrow$ Eq. (11)
    play the game, observe the next states s'_p and s'_e and the reward r
    $Q_{(s_{t+1}, u')} \leftarrow$ Eq. (11)
    $\delta_t \leftarrow$ Eq. (14)
    $e_t \leftarrow$ Eq. (16) {for the FIS}
    $\xi(t+1)_{FIS} \leftarrow$ Eq. (17) {update FIS input and output parameters}
    $\xi(t+1)_{FLC} \leftarrow$ Eq. (18) {update FLC input and output parameters}
    $s_t \leftarrow s_{t+1}$; $u \leftarrow u'$
  end for
end for

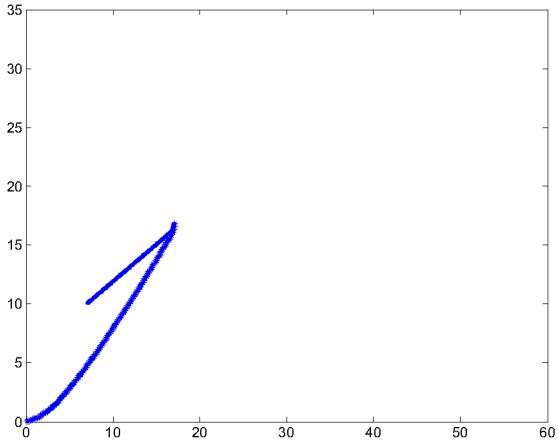


Fig. 5: The pursuer captures the evader with 100 learning episodes

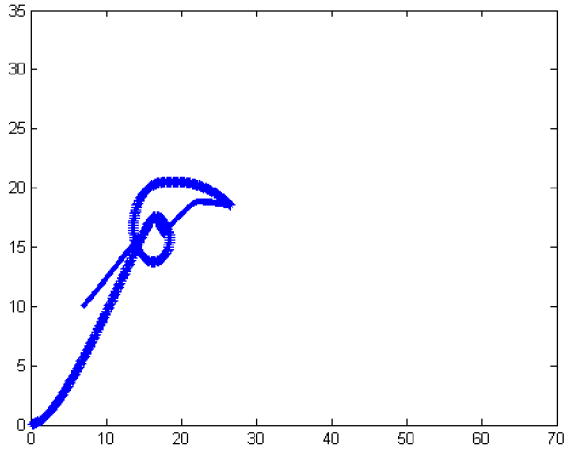


Fig. 6: The evader increases the capture time after 500 learning episodes

V. Simulation Results

The pursuer is twice as fast as the evader such that $w_p = 2m/s$ and $w_e = 1m/s$. The wheelbase of the pursuer is $L = 0.3m$. In each episode, the position of the evader is initialized randomly. The initial position of the pursuer is at the origin $(x_p, y_p) = (0,0)$ and the initial orientation is $\theta = 0rad$. We simulate the kinematic equations of the pursuer and the evader given in Eq. (2).

The game terminates when the pursuer captures the evader, or when the time exceeds 60 sec (escape). The capture radius is $\ell \leq 0.15m$, which is half the wheelbase of the pursuer $\ell \leq \frac{L_p}{2}$. The maximum steering angle of the pursuer is $-0.5rad \leq u_{p_{max}} \leq 0.5rad$ with $R_p = 0.5491m$. Given the stated parameters of the system and using Isaacs' capture condition (5), there exists a strategy for the evader to avoid capture.

We simulate the game using the learning algorithm given in Algorithm 1. The game is initialized such that the pursuer can capture the evader. However, the capture conditions (5) are set such that the evader can theoretically escape. We then run the simulation allowing both the evader and the pursuer to learn simultaneously. Both players use the same learning algorithm. The goal of initializing the parameters so that the pursuer captures the evader, is to test whether the evader will eventually learn to escape.

The number of steps in each episode is 600, and the sampling time is $0.1sec$. The system id simulated for different number of learning episodes. At the beginning of learning, the pursuer always captured the evader, as shown in Fig.5. After 500 episodes, as shown in Fig.6, the evader increased the capture time and made successful maneuvers. Figure 7 and table III show that the evader learned to escape from the pursuer after approximately 900 episodes. The evader makes sharp turns when the distance $d \leq R_p$. The evader avoids capture by changing its direction and entering into the pursuer's turning radius constraint. The tuned fuzzy consequence parameters K^l of the players after 900 episodes are shown in tables IV and V.

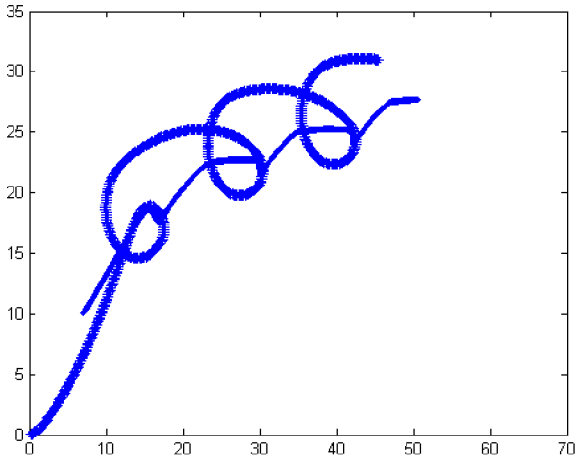


Fig. 7: The evader learns to escape after 900 learning episodes

TABLE IV: The evader's fuzzy decision table and the output constant K^l after 900 learning episodes

ϕ \ d	VC	CL	FE
N	-1.5848	-1.5782	-0.4074
Z	-1.5758	1.5526,	0.0331
P	1.5930,	1.5794	0.2626

TABLE III: This table summarizes the capture time for different number of learning episodes compared to the optimal solution for the homicidal chauffeur game

Game	no. of episodes	Capture time (sec)
Theoretical	–	escape (> 60)
After learning using QLFIS	100	12.90
	500	25.10
	900	escape (> 60)

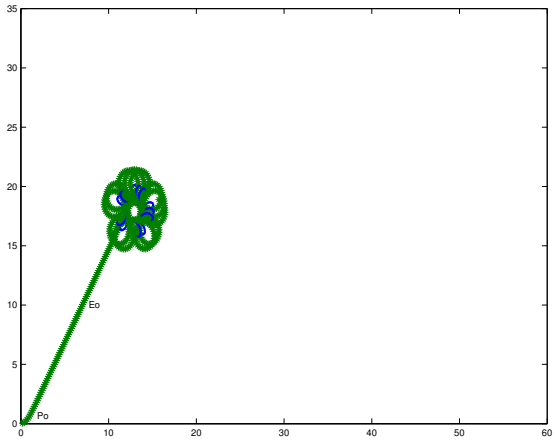


Fig. 8: The evader avoids capture when $u_{p_{max}} = 0.5rad$.

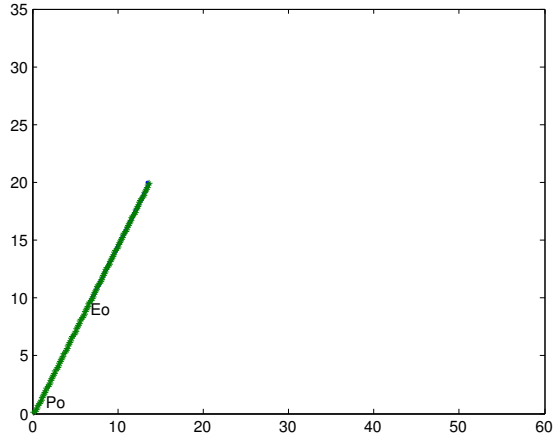


Fig. 9: The pursuer can capture the evader when $u_{p_{max}} = 0.7rad$.

TABLE V: The pursuer's fuzzy decision table and the output constant K^l after 900 learning episodes

ϕ \ $\dot{\phi}$	N	Z	P
N	-0.4763	-0.2503	-0.0075
Z	-0.2413	0.0023,	0.1522
P	-0.0046,	0.2650	0.4777

For comparison, we show the results of the theoretical solution described in Sect. II. Given the stated parameters of the system and using Isaacs' capture condition, there exists a strategy for the evader to avoid capture when $-0.5rad \leq u_{p_{max}} \leq 0.5rad$. Fig. 8 shows that the evader can escape from the pursuer by making sharp turns. We then increase the maximum steering angle of the pursuer to $-0.7rad \leq u_{p_{max}} \leq 0.7rad$. In this case, the capture condition is satisfied. The pursuer can capture the evader as shown in Fig. 9 with capture time $= 11.90sec$.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

The simulation of our model shows that the evader learns to turn and escape from the pursuer by using the Q(λ)-learning algorithm to tune the input and the output parameters of a fuzzy logic controller. Isaacs showed that if the capture conditions are not satisfied, there exists a strategy for the evader to avoid capture. We show that the evader can learn the optimal strategy to avoid capture. The technique shows that the learning algorithm converges to equilibrium after approximately 900 episodes. Furthermore, the evader's strategy can be improved by adding more membership functions.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[2] C. Watkins and P. Dayan, "Technical note: Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[3] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge University, England, 1989.

[4] S. Desouky and H. Schwartz, "Q(λ)-learning fuzzy logic controller for a multi-robot system," in *Systems, Man and Cybernetics, 2010. SMC 2010. IEEE International Conference on*, Istanbul, Turkey, Oct. 2010.

[5] S. Givigi, H. Schwartz, and X. Lu, "A reinforcement learning adaptive fuzzy controller for differential games," *J. Intelligent and Robotic Systems*, vol. 59, no. 1, pp. 3–30, July 2010.

[6] S. Desouky and H. Schwartz, "Learning in n-pursuer n-evader differential games," in *Systems, Man and Cybernetics, 2010. SMC 2010. IEEE International Conference on*, Istanbul, Turkey, Oct. 2010.

[7] A. H. K. M.E. Harmon, L.C. Baird, "Reinforcement learning applied to a differential game," *Adaptive Behavior*, vol. 4, no. 1, pp. 3–28, 1996.

[8] S. Givigi, H. Schwartz, and X. Lu, "An experimental adaptive fuzzy controller for differential games," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, oct. 2009, pp. 3017 –3023.

[9] X. Dai, C. Li, and A. Rad, "An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 3, pp. 285 – 293, Sept. 2005.

[10] P. Glorennec and L. Jouffe, "Fuzzy q-learning," in *Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on*, vol. 2, Jul 1997.

[11] 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, no. 3, pp. 1478 –1489, June 2004.

[12] L.-X. Wang, *A course in fuzzy systems and control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.

[13] R. Isaacs, *Differential Games*. John Wiley and Sons, 1965.

[14] ——, *Rand Reports RM-1391 (30 november 1954), RM-1399 (30 november 1954), RM-1411 (21 december 1954), and RM-1486 (25 march 1955), all entitled in part* Differential Games. Rand Reports, 1954,1955.

[15] A. MERZ, "The homicidal chauffeur," *AIAA Journal*, vol. 12, no. 3, pp. 259–260, March 1974.

[16] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Academic Press, New York: SIAM, 1999.

[17] S. H. Lim, T. Furukawa, G. Dissanayake, and H. F. D. Whyte, "A time-optimal control strategy for pursuit-evasion games problems," in *Proceeding. of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 2004.

[18] M. Pachter, "Simple-motion pursuit-evasion differential games," in *Proceedings of the 10th Mediterranean Conference on Control and Automation*, Lisbon, Portugal, July 2002.

[19] S. Desouky and H. Schwartz, "Hybrid intelligent systems applied to the pursuit-evasion game," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, San Antonio, TX, Oct. 2009, pp. 2603 –2608.

[20] J.-S. R. Jang and C.-T. Sun, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.