

Landing on the Moon with Type-2 Fuzzy Logic

Atakan Sahin

Centre for Process Analytics and Control Technology
University of Strathclyde
G1 1XL, Glasgow, United Kingdom
atakan.sahin@strath.ac.uk

Tufan Kumbasar

Control and Automation Engineering Department
Istanbul Technical University
Istanbul, Turkey
kumbasart@itu.edu.tr

Abstract—In this study, we will present the novel application of Type-2 (T2) fuzzy logic to the popular video game called Lunar Lander. The proposed T2 fuzzy moon landing system structure is composed of the error signal generator and the T2 fuzzy logic control structure which give the opportunity to transform the moon landing problem of the spaceship as a multivariable tracking control problem. The landing problem of the game can be seen as one of the classical multivariable control problems including uncertainties due to the randomization process occurring the game environment. Thus, we will employ T2 fuzzy logic controllers since they are capable of handling a high level of uncertainties. Then, by optimizing the T2 fuzzy moon landing system via the particle swarm optimization, we will show that the resulting T2 fuzzy moon landing system resulted with an adequate control and game performance in the presence of the uncertainties, disturbances and nonlinear system dynamics in comparison with its type-1 and conventional counterparts. We believe that the results of this paper will be an important step for a wider deployment of T2 fuzzy logic in the research area of computer games.

Keywords—Type-2 Fuzzy Logic; Type-2 Fuzzy Sets; Type-2 Fuzzy Logic Controllers, Games; Lunar Lander

I. INTRODUCTION

Computational Intelligence methods have been employed to computer games since they provide dynamic and challenging elements that are similar to real-world problems [1-11]. Consequently, computational intelligence methods have been used to games such as Counter-Strike [2], Flappy Bird [3], Pacman [4], StarCraft [5], Super Mario [6] and Lunar Lander [7-11]. Lunar Lander is one of most cloned game [12] in the game industry. The main goal of the game is to pilot a lunar landing module to a smooth touchdown on the moon surface while trying to keep away from the obstacle on the surface. The game has formed several kinds of researches such as route planning and path following problems as well as learning algorithm comparison on these [7-9], language design for game characteristic learning and planning according to several dynamic Computational Intelligence tasks on them [10, 11].

In the last ten years, one of the most popular research areas of Computational Intelligence is Type-2 (T2) Fuzzy Logic [13]. T2 Fuzzy Logic is an extension and generalization of its Type-1 (T1) counterpart. T2 Fuzzy Logic Controllers (FLCs) have been successfully employed in various real-world engineering problems such [13-20]. That lies because T2-FLCs using T2 Fuzzy Sets (FSs) are more capable of handling the high levels of uncertainties and nonlinear dynamics associated with real world control applications than its T1 and

conventional counterparts [13-14]. The benefits of T2-FLCs are demonstrated in applications such as mobile robots [15,20], flight systems [16], Computing with Words [17], decision making [18] and quite recently in its deployment to the game named Flappy Bird [3].

In this study, we will represent the novel application of the Interval T2 (IT2) FLCs to solve the moon landing problem in the computer game Lunar Lander. To the best of our knowledge, this is the first deployment of the widely used type-2 fuzzy control structure to the lunar lander game. We will present a novel type-2 fuzzy moon landing control system which is composed of two key blocks, namely the error signal generator and the IT2-FLC structure. The error signal generator is the key block which gives the chance to transform the moon landing problem of the spaceship as a multivariable control problem. In this fuzzy control system, we will design two input IT2-FLCs to have adequate control and game performance in the presence of the uncertainties, disturbances and nonlinear system dynamics. In this paper, firstly detailed information about the components of the proposed T2 fuzzy moon landing system is provided. Then, we will present the employed optimization based design approach of the IT2-FLC structure. We will examine the performance of the proposed T2 fuzzy moon landing structure, in comparison with its type-1 and conventional counterparts, by providing comparative results performed in the real game environment as well as simulation studies. The results will show that proposed T2 fuzzy moon landing system, in comparison with its conventional and type-1 counterparts, with a satisfying game and control performance in the presence of nonlinearities and high level of uncertainties.

The paper is organized as follows. Section II provides information about the game space of lunar lander. Section III presents the proposed T2 fuzzy moon landing system. The comparative experimental results with designed controllers with their design methodology are given in Section IV. Finally, the conclusions and future works are summarized in Chapter V.

II. THE GAME SPACE OF LUNAR LANDER

The game logic of the Lunar Lander game is to control the engine of the spaceship in the x-y coordinate system such that to land on the dock softly. The player can arrange the spaceship's angular rotation by pressing the right or left arrow keys on a cumulative basis. The player can also produce thrust against the gravity by pushing the spacebar key. The direction of the force obviously depends on the spaceship's angular position. In this study, we will use the Matlab clone of the Lunar Lander that can be found at [21].

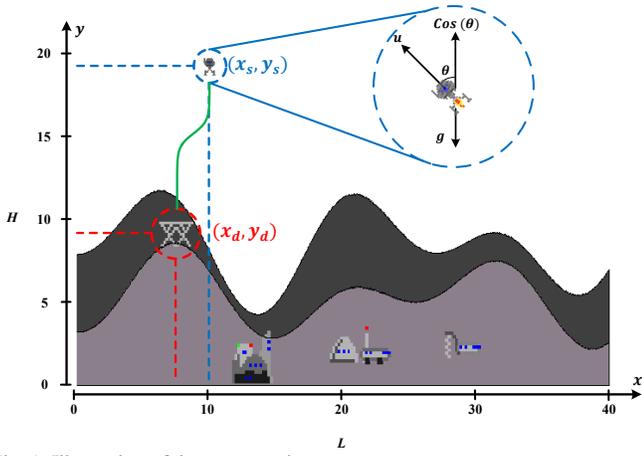


Fig. 1. Illustration of the game environment.

According to this clone, the game parameters shown in Fig.1 are defined as follows:

- **World height (H)** is the distance between the ceiling and the ground of the screen is fixed to a value of 20 units.
- **World length (L)** is the distance between the left and the right bounds of the screen is fixed value of 40 units.
- **The terrain** is defined and generated with randomly generated sinus functions. In Fig.1, light gray basements define the basic terrain, which is also obstacles to be avoided by the player. On the other hand, the dark gray ones only provide a realistic game environment, thus are not obstacles.
- **Position of the Dock (x_d, y_d)** is the target position landing defined in x-y coordinates. The x-axis position of the dock is randomly generated while its y-axis position is assigned with respect to the terrain level which is randomly generated.

- **Position of the Spaceship (x_s, y_s)** is defined in x-y coordinates. In the beginning of the game, the x-axis position of the spaceship is randomly assigned in the range of the world length while its y-axis position is set to a 20 units.
- **Speed of the Spaceship (v_x, v_y)** defines the linear speed of spaceship in x-y the coordinate system.
- **Gravitational constant (g)** is the acceleration constant of the spaceship in the y axis direction and has a default value assigned as 0.4 units per frame.
- **The angular position of the Spaceship (θ)** is the angle between the spaceship's direction and y-axis. The angle can be controlled with right and left arrow keys by player.
- **Thrust power (T)** defines the thrust force to be employed to the spaceship engine and is controlled by the player.
- **Fuel ($\int T$)** defines the maximum total thrust power that can be consumed by player.

In this Matlab clone [21], the game ends when

- The spaceships touches/ hits the terrain with failure
- The spaceship consumes more than 200-unit fuel before a successful landing with failure
- The Euclidean distance between the spaceship and the dock is less than 2 units with successful landing

Moreover, we have added the following conditions for a successful landing to make the game more realistic:

- Vertical velocity (v_y) must be less than -0.5 when the spaceship has landed
- The angular position of the spaceship (θ) must be between $[-\pi/16, \pi/16]$.

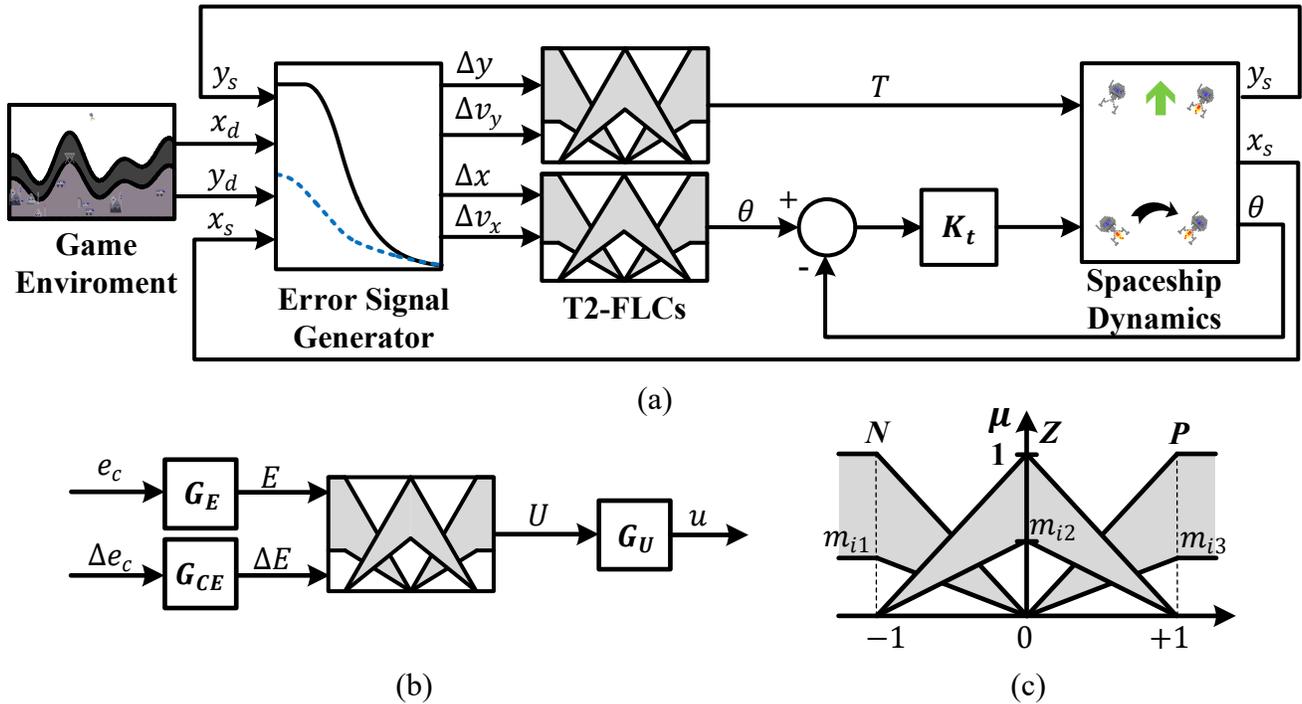


Fig. 2. Illustration of the (a) Type-2 Fuzzy Moon Landing System (b) IT2-FLC structure, (b) IT2-FSs.

III. THE TYPE-2 FUZZY MOON LANDING SYSTEM

In this section, we will convert the defined game space of the lunar lander game into a T2 fuzzy control problem. In this problem, we will handle the spaceship as the system to be controlled, the position of the dock as the desired reference signal. The gravity randomly generated terrain, and initial spaceship position will have defined as the disturbances and uncertainties in the control loop. The proposed T2 fuzzy moon landing scheme, shown in Fig.2a, will generate the required angle (θ) and thrust (T) signals such that the position of the space ship (x_s, y_s) converges successfully to the position of dock (x_d, y_d) in the presence of nonlinear system dynamics, noise and disturbances. Note that, we have also designed an inner loop proportional controller to speed up the response time of angle of the spaceship (θ). In all experiments, we have set and fixed this controller gain as $K_t = 1.648$. The proposed T2 fuzzy control system is composed of three main parts that are the error signal generator, the IT2-FLCs, and the system dynamics of the spaceship.

A. The System Dynamics of the Game

The dynamics of the spaceship are based on the classical motion equations. Thus, the acceleration (a) of the spaceship at k^{th} frame in x-y coordinate system can be defined as:

$$\begin{aligned} a_x(k) &= -\sin(\theta(k)) * T(k) * d_t \\ a_y(k) &= \cos(\theta(k)) * T(k) * d_t \end{aligned} \quad (1)$$

where d_t is sampling time of the game which has fixed value of 0.1. Correspondingly, the velocity (v) equations of the both axis can be derived as:

$$\begin{aligned} v_x(k+1) &= v_x(k) + a_x(k) * d_t \\ v_y(k+1) &= v_y(k) + a_y(k) * d_t - g * d_t \end{aligned} \quad (2)$$

accordingly the position of spaceship (x_s, y_s) equations can also derive:

$$\begin{aligned} x_s(k+1) &= x_s(k) + v_x(k) * d_t \\ y_s(k+1) &= y_s(k) + v_y(k) * d_t \end{aligned} \quad (3)$$

B. The Error Signal Generator for the T2 Fuzzy System

Here, the error signal generator is described which transforms the landing of the spaceship into a control problem. As it has been asserted in Section II, the spaceship is controlled in defined in the x-y coordinate system. Thus, for landing on the dock, the error signals to be minimized by T2 fuzzy moon landing scheme are generated as follows:

$$\begin{aligned} \Delta x(k) &= x_d - x_s(k) \\ \Delta y(k) &= y_d - y_s(k) \end{aligned} \quad (4)$$

Note that, the values x_d and y_d are fixed to the randomly generated values at the beginning of the game as defined at Section II until the game ends.

C. The Structure of the IT2-FLCs

As it has been asserted in Section II, the player has to control the thrust and angle of the spaceship for a successful landing. In this context, we will design two IT2-FLCs to

control the thrust power and angular position of the spaceship to provide a successful landing as illustrated at Fig.2a.

In this study, we will employ the two input IT2-FLC structure presented in Fig.2b. The IT2-FLCs are constructed by choosing the inputs to be e_c as error and Δe_k as change of error and the output as the control signal (u_k) where $e_c = \Delta x(k)$, $\Delta e_c = v_x(k)$ and $u_c = \theta(k)$ for $c = 1$; moreover $e_c = \Delta y(k)$ and $\Delta e_c = v_y(k)$ and $u_c = T(k)$ for $c = 2$. Each input of the IT2-FLC is normalized with two input Scaling Factors (SFs) (G_E, G_{CE}) and the output is denormalized the SF G_U . In this study, the SF G_E is defined as $G_e = 1/e_{kmax}$ where e_{kmax} is the maximum error value at the beginning of the game. The IT2-FLCs use and employs the 3×3 rule base presented in Table I. The rule (R) structure is defined as follow:

$$R_q: \text{ IF } E \text{ is } \tilde{A}_{1j} \text{ and } E \text{ is } \tilde{A}_{2j} \text{ THEN } U \text{ is } C_q \quad (5)$$

where C_q is the crisp consequent Membership Functions (MFs) ($q = 1, \dots, Q = 9$) is defined five linguistic terms Negative Big (NB), Negative (N), Zero (Z), Positive (P) and Positive Big (PB) that represent $-1, -0.5, 0, 0.5, 1$, respectively. The antecedent part of the rule is defined with IT2-FSSs ($\tilde{A}_{1j}, \tilde{A}_{2j}; j = 1, 2, 3$) which are defined with three linguistic terms Negative (N), Zero (Z), Positive (P). The IT2-FSSs can be described with upper MFs ($\bar{\mu}_{\tilde{A}_{1i}}$ and $\bar{\mu}_{\tilde{A}_{2i}}$) and lower MFs ($\underline{\mu}_{\tilde{A}_{1i}}$ and $\underline{\mu}_{\tilde{A}_{2i}}$) which provides extra degree of freedom that is also known as Footprint of Uncertainty (FOU). As shown in Fig.2c, the FOU of IT2-FSSs is generated with heights of the lower MFs (m_{ij}) which is the only design parameter to be tuned. The implemented the IT2-FLC uses the center of sets type reduction method [20].

TABLE I. RULE TABLE

$\Delta E/E$	N	Z	PB
N	NB	N	Z
Z	N	Z	P
P	Z	P	PB

IV. PERFORMANCE EVALUATION OF THE T2 FUZZY MOON LANDING SYSTEMS

Here, the design of the proposed T2 fuzzy moon landing system is presented, and performance is investigated. Firstly, we will present two control system analyses to compare the control performance of the T2 fuzzy moon landing system in comparison with its Type-1 and conventional counterparts. Then, we will present experimental results that are performed in the game environment to examine its game performance. The presented results are carried out on a personal computer with an Intel Core i7 CPU - 2.60 GHz, running Windows 10 64-bit and Matlab R2015b. The IT2-FLCs have been implemented by using the Interval Type-2 Fuzzy Logic Toolbox [22].

A. Control System Performance Evaluation

In this subsection, we will present and examine the performance of the T2 fuzzy moon landing system in the framework of control theory. The proposed IT2-FLC structure

will be compared with a Type-1 Fuzzy Logic Controller (T1-FLC) structure and a conventional PD controller structure. The T1-FLC structure will use and employ the identical rule base of the IT2-FLC one with the only difference that its antecedent part will be defined with triangular type-1 fuzzy sets [19].

As it has been mentioned in Section II, the Lunar Lander is a limited type game depending on game ending condition and also includes random parameter initializations for each trial. Therefore, the parameter tuning phase of the controllers should be accomplished with several trials in the game space to design controllers that are robust for randomly generated game environments. Therefore; we will define 4 training sets, as tabulated in Table III, and tune the controllers respectively. Note that, the starting position of the spaceship is set and fixed during this phase to the value (16,20). Moreover, the terrain characteristics have also been set and fixed in the training phase to make a fair comparison between the controllers. All three controller structures were optimized with the particle swarm optimization such that to minimize the following objective function:

$$F = \sum_{k=1}^K \|e(k)\|^2 + \mathcal{C} \quad (6)$$

where K represents the total number of samples, $\|\cdot\|$ is norm operator, the $e(k)$ is defined as

$$e(k) = [\Delta x, \Delta y]_{(1 \times 2)} \quad (7)$$

moreover, \mathcal{C} is the penalty for a crash defined as:

$$\mathcal{C} = \begin{cases} 10000, & \text{if } \text{crash} = 1 \\ 0, & \text{if } \text{crash} = 0 \end{cases} \quad (8)$$

Note that, to show the superiority of T2 fuzzy sets clearly, we have not optimized the SFs of the IT2-FLCs. We have set and fixed them to the optimal values found for its type-1 counterpart.

The resulting optimal parameters according to their parts are tabulated in Table II. Moreover, we have also provided the training scenarios with their landing time for comparison in Table III. The system responses for Scenario Number 1 and 4

are illustrated in Fig.3a and Fig.3b, respectively. Firstly, it can be clearly seen that all three controller structures resulted with successful landings as their crash count is equal to zero. For Scenario Number 3, the IT2-FLC structure decreased the Landing Time value about 41% and 3% (increased the convergence speed to dock) while it also reduced the total fitness value about 30% and 16% in comparison to the PD and T1-FLC structures, respectively.

TABLE II. CONTROLLER PARAMETERS

Method	Controller	Parameter	Value
PD	Thrust	K_p	5.22
		K_d	13.14
	Angle	K_p	0.51
		K_d	6.12
T1-FLC	Thrust	G_E	$1/e_{kmax}$
		G_{CE}	0.18
		G_U	39.51
	Angle	G_E	$1/e_{kmax}$
		G_{CE}	0.61
		G_U	41.37
IT2-FLC	Thrust	m_{11}, m_{13}	0.27
		m_{12}	0.91
		m_{21}, m_{23}	0.11
		m_{22}	0.67
	Angle	m_{11}, m_{13}	0.15
		m_{12}	0.61
		m_{21}, m_{23}	0.47
		m_{22}	0.91

It should be noted that the performances of the T1 and IT2 fuzzy moon landing systems are quite similar. That lies because the design of the IT2-FLC has been accomplished as an extension of its T1 counterpart. Moreover, this also coincides with the results presented in [23] where it has been stated that the IT2-FLCs result in smoother control surfaces in comparison with its T1 counterpart. Thus, the resulting system response might be relatively slower but potentially more robust against uncertainties. Similar comments can be made for the other two reference variations.

TABLE III. CONTROL PERFORMANCE EVALUATION ON TRAINING AND TESTING SCENARIOS

	Positions			PD			T1-FLC			IT2-FLC		
	Scenario Number	Initial Point	Set Point	Fitness Value	Landing Time	Crash	Fitness Value	Landing Time	Crash	Fitness Value	Landing Time	Crash
Training	1	(16,20)	(2,5.4)	1612	17.6	–	1508	14.2	–	1520	14.6	–
	2		(10,4.2)	1577	21.2	–	1278	16.7	–	1307	17.0	–
	3		(18,5.6)	1284	19.9	–	1087	12.1	–	905	11.7	–
	4		(25,4.3)	1626	20.7	–	1394	17.2	–	1372	16.5	–
Testing	5	(35,20)	(2,5.4)	13093	–	1	3938	23.5	–	3775	22	–
	6		(5,3.6)	12932	–	1	13107	–	1	3500	22.4	–
	7		(10,4.2)	12394	–	1	12607	–	1	2793	20.4	–
	8		(35,5.5)	1368	20.5	–	10915	–	1	1317	20.6	–

We have also tested the controllers for a different initialization of the spaceship (35,20) to investigate how they will cope with uncertainty and different operating regions. The resulting performance values are also tabulated in testing part of Table III. The system responses for Scenario Number 5 and 7 are illustrated in Fig.3c and Fig.3d, respectively. It can clearly observe that the T2 fuzzy moon landing system was able to pilot the spaceship to the dock without a crash for all scenarios while its T1 and conventional counterparts crashed the spaceship in three of them. The PD structure was not able to handle the uncertainty and thus, in the first three testing scenarios, the crash occurred because the 3rd condition of the successful landing (presented in Section III-A) was violated. The T1 fuzzy structure crashed the spaceship in Scenario Numbers 6 and 7 since it hit/touched the terrain. The last crash of the T1-FLC structure occurred due to the fact the required angle condition for a successful landing could not be satisfied as it resulted with oscillating system response. The presented results clearly show that the proposed T2 fuzzy moon landing system is capable of handling uncertainties and various operating points in comparison with its T1 and conventional counterparts successfully.

B. Game Performance Evaluation

In this subsection, we will examine the performance of the controllers on the game logic. As the nature of the game environment, the randomization phase of the game can create various unique terrains, starting points, and landing points. Therefore, here we will use the successful landing criteria for the comparison. Thus, we have employed 200 times each controller structure to game where starting point (x_s, y_s) , landing point (x_d, y_d) and also terrain characteristics are randomly generated by the game. The success rates of the controllers are given in Table IV. It can be concluded that the game performance of the IT2-FLC structure, with respect to game logic, is better than its T1 and conventional parts by almost 14% and 22.5%, respectively.

TABLE IV. GAME PERFORMANCE EVALUATION ON 200 TRIALS

	<i>Crash Count</i>	<i>Success Rate</i>
PD	78	61%
T1-FLC	61	69.5%
IT2-FLC	33	83.5%

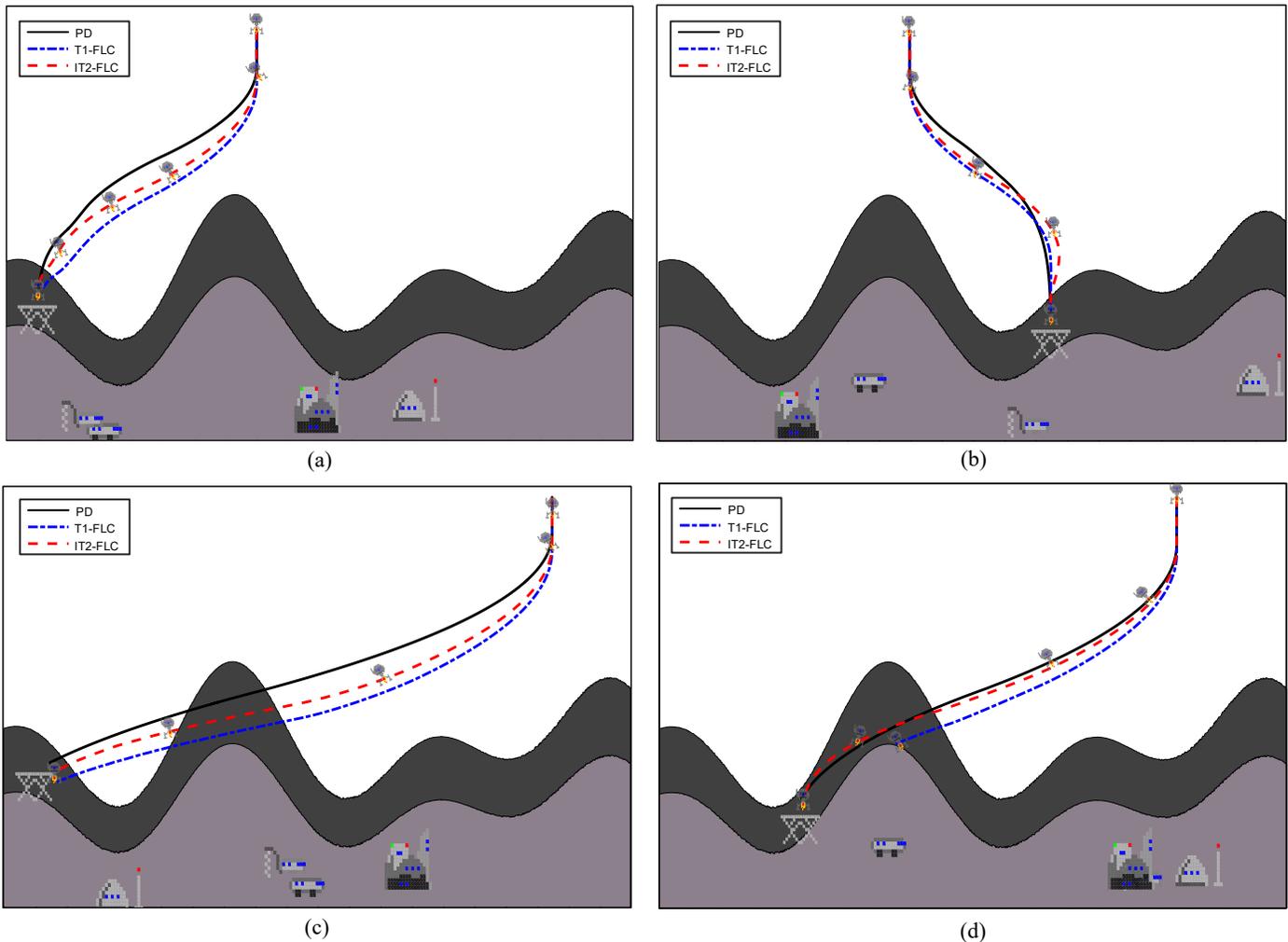


Fig. 3. Illustration of some training and testing cases. (a) scenario-1 and (b) scenario-4 for training, (c) scenario-5 and (d) scenario-7 for testing.

V. CONCLUSIONS AND FUTURE WORK

In this study, we presented a novel T2 fuzzy moon landing system to solve landing problem in the lunar lander game. To the best of our knowledge, our work is the first deployment of the widely used T2 fuzzy control structure to this game. The proposed IT2-FLC structure is composed of the error signal generator and the IT2-FLC structure which gave the opportunity to transform the moon landing problem of the spaceship as a multivariable tracking control problem. The landing problem of the game is one of the classical position control problems including several uncertainties depending on the randomization process on the game environment. Thus, by employing T2 fuzzy logic and PSO, we have shown that the resulting T2 fuzzy moon landing system resulted with an adequate control and game performance in the presence of the uncertainties, disturbances and nonlinear system dynamics. For our future work, we aim to extend the application area of T2 fuzzy logic in real world games.

REFERENCES

- [1] S. M. Lucas and G. Kendall, "Evolutionary computation and games," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 10-18, 2006.
- [2] N. Cole, S. J. Louis and C. Miles, "Using a genetic algorithm to tune first-person shooter bots," in *Proc. IEEE Congr. on Evol. Comput.*, 2004, pp.139-145.
- [3] A. Sahin, E. Atici and T. Kumbasar, "Type-2 fuzzified flappy bird control system," in *Proc. IEEE Int. Conf. on Fuzzy Syst.*, 2016, pp.1578-1583.
- [4] S. M. Lucas, "Evolving a neural network location evaluator to play ms. pac-man," in *Proc. IEEE Symp. on Comput. Intell. and Games*, 2005, pp. 203-210.
- [5] S. Ontanón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill and M. Preuss, "A survey of real-time strategy game ai research and competition in starcraft," *IEEE Trans. on Comput. Intell. and AI in games*, vol. 5' no. 4, pp. 293-311, 2013.
- [6] S. Karakovskiy and J. Togelius, "The mario ai benchmark and competitions," *IEEE Trans. Comput. Intell. and AI in Games*, vol. 4, no. 1, pp. 55-67, 2012.
- [7] S. Samothrakis, S. A. Roberts, D. Perez and S. M. Lucas, "Rolling horizon methods for games with continuous states and actions," in *Proc. IEEE Conf. on Comput. Intell. in Games*, 2014, pp. 1-8.
- [8] S. A. Roberts and S. M. Lucas, "Measuring interestingness of continuous game problems," in *Proc. IEEE Conf. on Comput. Intell. in Games*, 2013, pp. 1-8.
- [9] J. MacGlashan and M. L. Littman, "Between imitation and intention learning," in *Proc. Int. Conf. on Artificial Intell.*, 2015, pp. 3692-3698.
- [10] T. Schaul, "A video game description language for model-based or interactive learning," in *Proc. IEEE Conf. on Comput. Intell. in Games*, 2013, pp. 1-8.
- [11] M. Ebner, J. Levine, S. M. Lucas, T. Schaul, T. Thompson and J. Togelius, "Towards a video game description language," DOI: 10.4230/DFU.Vol6.12191.85, 2013.
- [12] B. Edwards, "Forty Years of Lunar Lander", *Technologizer*, 2009. [Online]. Available: <http://www.technologizer.com/2009/07/19/lunar-lander/>. [Accessed: 02- Jan- 2017].
- [13] J. M. Mendel, H. Hagnas, W.W Tan, W.W. Melek, and H. Ying, "Introduction to type-2 fuzzy logic control", John Wiley and IEEE Press, Hoboken, NJ, 2014.
- [14] J. M. Mendel and R. B. John, " Type-2 fuzzy sets made simple," *IEEE Trans. on Fuzzy Syst.*, vol. 10, no. 2, pp. 117-127, 2002.
- [15] H. Hagnas, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524-539, 2004.
- [16] M. Mehndiratta, E. Kayacan and T.Kumbasar, "Design and experimental validation of single input type-2 fuzzy PID controllers as applied to 3 DOF helicopter testbed," in *Proc. IEEE Int. Conf. on Fuzzy Syst.*, 2016, pp. 1584-1591.
- [17] J. M. Mendel, "Computing with words and its relationships with fuzzistics," *Inform. Sci.*, vol. 177, no. 4, pp. 988-1006, 2007.
- [18] S. M. Chen andL. W. Lee, "Fuzzy multiple attributes group decision-making based on the interval type-2 TOPSIS method," *Expert Syst. with Applicat.*, vol. 37, no. 4, pp. 2790-2798, 2010.
- [19] A. Sakalli, T. Kumbasar, E. Yesil and H. Hagnas, "Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system," in *Proc. IEEE Int. Conf. on Fuzzy Syst.*, 2014, pp. 1859-1866.
- [20] T. Kumbasar and H. Hagnas, H. "Big Bang-Big Crunch optimization based interval type-2 fuzzy PID cascade controller design strategy," *Inform. Sci.*, vol. 282, pp. 277-295, 2014.
- [21] H. Corte, "Moon lander matlab game", mathworks.com, 2012. [Online]. Available: <https://uk.mathworks.com/matlabcentral/fileexchange/38927>. [Accessed: 02- Jan- 2017].
- [22] A. Taskin and T. Kumbasar, "An open source Matlab/Simulink toolbox for interval type-2 fuzzy logic systems," in *Proc. IEEE Symp. Series on Comput. Intell.*, 2015, pp. 1561-1568.
- [23] T. Kumbasar and H. Hagnas, "A self-tuning zSlices based general type-2 fuzzy PI controller," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 991-1013, 2015.