This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach

Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce
Universidad Panamericana, Facultad de Ingeniería, Augusto Rodin 498, Ciudad de México 03920, México
Corresponding author: Ernesto Moya-Albor (e-mail: emoya@up.edu.mx)

Abstract—The paper presents a Fuzzy Q-Learning (FQL) and optical flow based autonomous navigation approach. The FQL method takes decisions in an unknown environment and without mapping, using motion information and through a reinforcement signal into an evolutionary algorithm. The reinforcement signal is calculated by estimating the optical flow densities in areas of the camera to determine whether they are "dense" or "thin" which has a relationship with the proximity of objects. The results obtained show that the present approach improves the rate of learning compared with a method with a simple reward system and without the evolutionary component. The proposed system was implemented in a virtual robotics system using the CoppeliaSim software and in communication with Python.

Keywords: optical flow; CoppeliaSim; fuzzy Q-learning; reinforced learning; vision-based control navigation; evolutionary algorithm

I. INTRODUCTION

Robotic navigation is a relevant topic in a world where autonomous navigation systems are starting to get implemented and could be useful in industries like automotive, logistics and industrial management. New methods in autonomous navigation could aid in automation of many tasks as well as creating efficiencies in several industries. On the other hand, these systems so far have proven unreliable when it comes to success rate and performance, which leads to them not being useful enough to be implemented. Therefore, it is important to find new methods and refine existing ones.

One way to divide robotic navigation is into motion and navigation. Thus, some researchers have been done in the motion field to make the robot movement "smoother", for example, using Bezier curves and other geometric features [1]-[4]. On the navigation front, the most common method is using ultrasonic sensors to measure distances as an input to a control scheme, frequently fuzzy logic or a geometric inference as in [5]-[10], in recent years LIDAR has become common alternative to ultrasonic sensors [11][12] these have been popularized because they provide relatively accurate distance information of their environment with simple processing, but with its high cost of as main disadvantage. Furthermore, there is currently great interest in the use of vision systems as the main input for navigation. Thus, RGB-D has been employed because it provides images that can be used for object identification and still provide depth information similar to the ultrasonic sensors and LIDAR [9][10][13][14]. However, these approaches rely on having a dedicated sensor to perceive depth and not purely on vision. The most common purely vision-based method for navigation is the use of multi-camera arrays which use techniques such as stereoscopy to estimate depth [15]-[19]. On the other hand, another less common method is through the use of structured light like the Kinect sensor [20]. Finally, there's the monocular vision methods, which are challenging since there are no other sensors involved to aid in depth estimation. Therefore, these use other algorithmic solutions to address the navigation problem [21]-[24]. In this paper, we will be exploring the use of monocular vision in conjunction with fuzzy Q-Learning (FQL) for robotic navigation in an unknown environment. On the one hand, FQL is a rules-based control approach for mobile robots with ultrasonic sensors or cameras [25]-[31]. On the other hand, instead of estimating depth we will use an optical flow estimation technique as was proposed in [32], this technique estimates the amount of movement in each pixel between two images which can be processed to calculate a direction without obstacles, the advantage of this type of system is that it does not require previous knowledge of its environment and it does not require mapping to navigate unlike the LIDAR or ultrasonic sensor approaches. In addition, we improved the FQL approach reported in [31] by adjusting the reinforcement signal, this signal comes from the measurement of optical flow density in an area

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

recorded by the camera and determines whether that region is considered "dense" or "thin", and through the addition evolutionary component, which involves time-series measures from various ultrasonic sensors on the current and previous execution loops with the objective of helping it continue learning when its performance stagnates. Finally, the proposed system was implemented and tested in a virtual robotics environment.

The remainder of the paper is organized as follows: in Section II, the virtual robotics environment, the fuzzy Q-learning approach, and the optical flow method used are presented. Section III shows the proposed vision and FQL-based autonomous navigation method. Experiments results of control to obstacle avoidance are reported in Section IV. Finally, the discussion and conclusions are given in Sections V and VI, respectively.

II. MATERIALS AND METHODS

A. COPPELIASIM SIMULATION ENVIRONMENT

CoppeliaSim is a simulation environment for robotics dynamics and control testing. The simulation environment can be modified to create multiple testing scenarios and be able to test the control systems and dynamics of robots and to track their trajectories. In this paper, we use the CoppeliaSim Edu version [33].

To test the proposed navigation autonomous control, we use the Pioneer P3-DX as our surrogate robot (Fig. 1a). Pioneer P3-DX robot is a two-wheel and two-motor differential drive robot, which includes ultrasonic sensors to measure distances to obstacles. In addition, we added an RGB camera with a resolution of 256×256 pixels and a field of view of 60°.

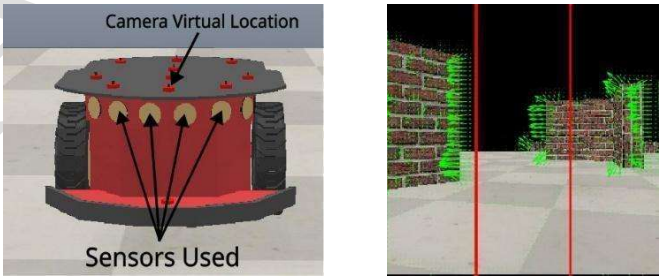
The simulation environment of CoppeliaSim is configured to run on real time in a Ryzen 7750x up to 5.5GHz CPU with 64GB of RAM, it is also tested in a i7-10750H up to 2.8 GHz CPU and 16GB of RAM and did not use all system resources, therefore the simulation can be done in lower capacity equipment.

B. OPTICAL FLOW ESTIMATION METHOD

The optical flow methods estimate the spatial displacement between the pixels of two consecutive images. These displacements generally are associated with intensity variations of local structures in the image sequence [32] and generally a larger optical flow indicates objects that are closer in proximity to the camera since they would shift more pixels. This is estimated by assuming that the change between images in time will be close to zero. The optical flow constraint equation holds that $E_x u+E_y v+E_t=0$, where $E=E(x,y,t)$ relates to the intensity of the image in the spatial coordinates x,y and time t, E_* is the derivative of E with respect to *, and u,v are the displacements from a time t to a time $t+1$. Due to this equation having two unknown variables so a constraint is needed to approximate the rate of change, in [32] the smoothness constraint is proposed as seen in (1).

$$\int\int [\alpha^2(||\nabla u||^2 + ||\nabla v||^2) + \left(E_x u + E_y v + E_t\right)]dxdy = 0, \quad (1)$$

where α is a weighting factor which aids in areas where the brightness gradient is small and ∇^2 indicates the Laplacian [32]. The result of solving (1) are the motion vectors $(u,v)^\top$ for each pixel which represent the perceived movement between two consecutive images as seen on Fig. 1b.



a **b**

Fig. 1. (a) P3-DX Robot. **(b)** Optical flow vectors between two consecutive images.

Optical flow however has some limitations due to it using assumptions to be able to calculate the optical flow, one of these assumptions is that the movements between images are slow, therefore if there are sudden movements or the robot moves too fast, it can lead to some inaccuracies, however it allows for a simple model to be implemented in real time. On this work this method was selected due to ease of implementation and due to its novelty as unlike most other methods, it does not require measuring distances in its environment.

C. FUZZY Q-LEARNING

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

Fuzzy Q-Learning is a combination of fuzzy-logic and Q-learning, where the fuzzy portion defines the rules and takes actions based on the states of the system [25][26], while the Q-Learning is in charge of generating a reward which alters the functions of the inference made by the fuzzy-logic portion to make it adapt to new situations, these rewards are based on another set of functions and rules that defines them. Thus, the system over time "learns" by estimating the discounted future rewards given the actions performed from states. The Q function value estimation equation is shown in (2) [27]:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \epsilon \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right], \quad (2)$$

Where S_t is the current state, A_t is the action in S_t, $Q(S_t,At)$ is the estimated sum of discounted reward carried out by an action A given a state S, ϵ is the learning rate, γ is the discounting factor, $S_{(t+1)}$ is next state, $R_{(t+1)}$ is the reward in the new state and a is the action with the largest q-value in a given S_t.

III. VISION-FQL BASED OBSTACLE AVOIDANCE CONTROL

A. CONTROL SCHEME

Fig. 2 shows the overall schema of the obstacle avoidance control based on vision. It uses the optical flow method described in Section II.B as input and a closed loop FQL controller described in Section II.C.

The inputs for the proposed obstacle avoidance system consist of motion densities. These densities were obtained from the optical flow estimation through three horizontal areas of the images (R_{i+n}) corresponding to left ($n = 0$), center ($n = 1$) and right ($n = 2$) areas of the images as it is shown in Fig. 1-b. Then, they were fuzzified into "dense" and "thin" fuzzy sets, dense meaning that there is more perceived moment and thin the contrary [31]. The procedure to calculate these densities is described in (3).

$$\underline{R_i} = \frac{\underline{R_i}}{max\{\underline{R_i}, \underline{R_{i+1}}, \underline{R_{i+2}}\}}, \quad (3)$$

where $\underline{R_i}$ are the normalized magnitudes of each region (left, center and right) with respect to the area with the maximum average registered. The fuzzy input membership functions have been changed from the ones used in [31] to the ones shown in Fig. 3. To determine the state of the robot, the three fuzzified inputs compared to one of eight states or rules (Ru) in Table I, which are the different combinations of "thin" and "dense" (T or D) for R_{i+n} sections, in turn these states can take one of four actions $a1$,$a2$,$a3$ and $a4$ which correspond to going forward, left, right and back and then turning right. The proposed FQL architecture is shown in Fig. 4, where the fuzzy system used operates with eight fuzzy rules, whose inputs are the densities mentioned in (3) as knowledge base, and as it is shown in Table I, where | represents logical OR and max represents the maximum q value present on a given row of a fuzzy rule. The q values with which the actions are selected change over time through the reward function or reinforcement signal and an error signal, for this application we propose a reinforcement signal that measures the distance to the objects using the four central ultrasonic sensors of the P3-DX robot (Fig.1a). These will provide rewards in real time depending on whether the robot is getting closer or farther to a collision and the space it has to maneuver so it registers the maximum distances from the ultrasonic sensors, if the sensors do not detect anything the distance defaults to 999. This is designed to consider the maximum and minimum distances read by the sensors.

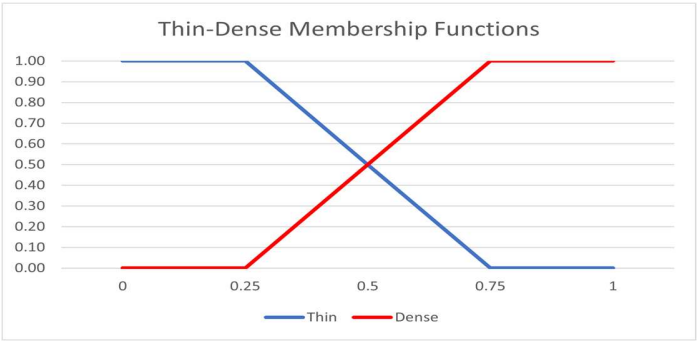


Fig. 2. Obstacle avoidance control overview.

Fig. 3. Input membership functions.

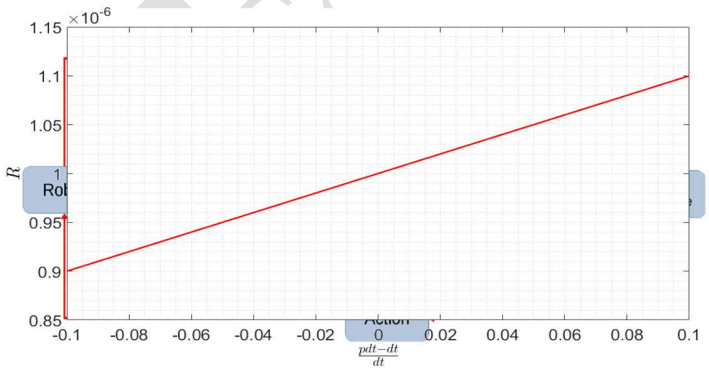
This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

Table I. Fuzzy rules implemented in the controller.

Ru	R_i	R_{i+1}	R_{i+2}	Action	*q(i,j)*
1	T	T	T	*a1|a2|a3|a4*	*mx q1,j*
2	T	T	D	*a1|a2|a3|a4*	*mx q2,j*
3	T	D	T	*a1|a2|a3|a4*	*mx q3,j*
4	T	D	D	*a1|a2|a3|a4*	*mx q4,j*
5	D	T	T	*a1|a2|a3|a4*	*mx q5,j*
6	D	T	D	*a1|a2|a3|a4*	*mx q6,j*
7	D	D	T	*a1|a2|a3|a4*	*mx q7,j*
8	D	D	D	*a1|a2|a3|a4*	*mx q8,j*

The reward system is shown on Table II, where the distance obtained from the ultrasonic sensors is considered. In this manner, the minimum distance registered by the sensors d_{min} is compared to define the reward value. Thus, $dl = 0.14$ is the collision threshold distance when a sensor registers a distance lesser than d_l, the algorithm enters into the first condition shown in Table II and therefore the robot gets a fixed negative reward signal and the simulation ends. pd_{min} is the distance registered by the sensors on the previous execution loop, dt is the distance to the target, pdt is the distance to target (circle green in the scenarios shown in Fig. 7) on the previous execution loop, d_{max} is the maximum distance registered by the sensors, and $\alpha = 1 \times 10^{-5}$.

The idea of considering d_{min}, pd_{min} as conditions is to "reward" when the robot gets farther from an obstacle and therefore enters into the second condition and to "punish" the robot when it gets closer to an obstacle and enters into the third condition, d_{max} it is used to moderate the reward signal when it's a "punishment" by considering how much distance it has to maneuver. Finally, the logic behind using pdt and dt as inputs is to direct the robot towards the target, it is used in the second and third conditions to modify slightly the reward by taking into account how much it approached the target regardless of being closer or farther



from an obstacle, the rewards associated with getting closer to the target are approximately an order of magnitude lower, so the robot prioritizes avoiding collision, but still gets "rewarded" for getting closer to the target and "punished" when getting farther from it.

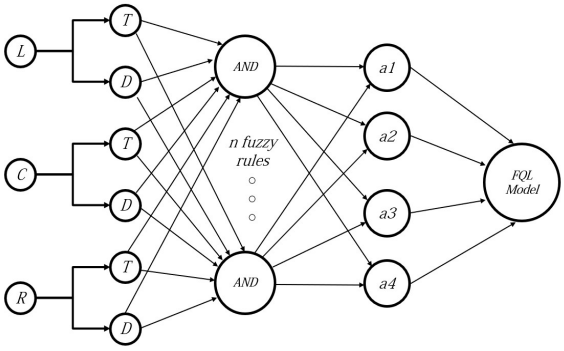


Fig. 4. FQL Architecture.

Table II. Fuzzy rules implemented in the controller.

Condition	Reward Function
$d_{min} \leq dl$	-2α
$d_{min} \geq pd_{min}$ or $d_{min} \geq 999$	$\alpha + \alpha\left(\frac{pdt - dt}{dt}\right)$
$d_{min} \leq pd_{min}$	$-\alpha\left(\frac{d_{max}}{d_{min}}\right) + \alpha\left(\frac{pdt - dt}{dt}\right)$

The error signal is calculated through the quality value $V(x,a)$ and the calculated q value $Q(S_t,A)$ using (4) [12]:

$$\Delta Q = r + \gamma V(x,a) - Q(x,a), \quad (4)$$

where r is the calculated reward and γ is a discount factor. With that error signal the new q value, for the action taken, is updated as it is shown in (5):

$$(i,j^0) = q(i,j^0) + \Delta q(i,j^0), \quad (5)$$

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

where $q(i,j^0)$ is the q value of the action taken, and the value of $\Delta q(i,j^0)$ can be obtained through (6):

$$\Delta q(i,j^0) = \epsilon \times \Delta Q(i,j^0) \times \frac{a_i(x)}{\Sigma_{i=1}^{N} a_i(x)}, \quad (6)$$

where ϵ is a discount factor, and for our tests it was fixed to 0.2.

The behavior of the reward function depends on the corresponding condition of Table II, for the first condition the value of the reward function is fixed, for the second condition, where the robot is rewarded, the behavior is linear as shown on Fig 5. The third condition depends mostly on the relation between d_{max} and d_{min}, maintaining a constant d_{min}, pdt and dt, its behavior is linear as seen on Fig. 6 (red plot). However, when d_{max} is constant instead of d_{min}, its behavior becomes of the form $-\frac{n}{x}$ seen on Fig. 6 (blue plot).

Compared to the method explored in [31] the reinforcement signal proposed on this paper is able to judge not only if the robot has collided or not but also how well it's navigating based on the reward responses shown in (5) and (6) providing a proportional signal instead of a fixed reward.

B. EVOLUTIONARY CORRECTION

In the experiments carried out, it is observed that after certain number of iterations, the system tends to stop improving, therefore, we implemented an evolutionary system based on conditions so in the case that the system's performance during training decreases it can use existing Q tables from previous iterations, distinguish the ones which had the best performances, combine and mutate them so it can better retain previous knowledge.

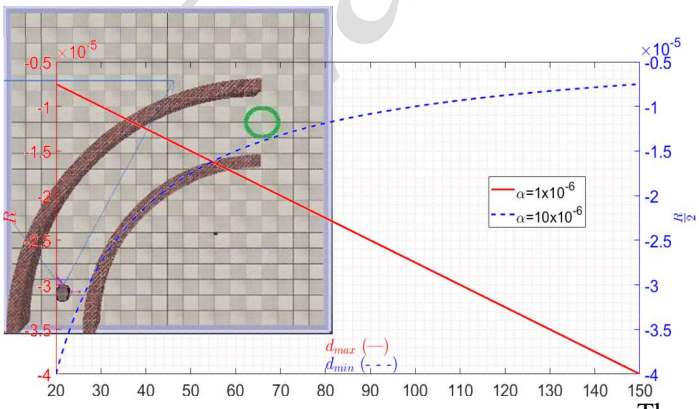


Fig. 5. Reward response on condition 2.

The method implemented is a genetic algorithm, where the "parents" are the three best Q-tables, whose score is based on the minimum distance reached to a set target in the scenario, and the "child" is the average of the parents, where each element of the resulting "child" Q-table is mutated according to (7), where R_{rand} is the mutation matrix where each cell contains a random value within the range of 0.95 and 1.05.

$$\frac{Q_{table} + Q_{table} + Q_{table}}{3} \times R_{rand}, \quad (7)$$

where Q_{tablei}, with $i=1,...3$, are the three best performing models at the end of the iteration.

IV. TESTING AND RESULTS

In this section, we show the tests performed to evaluate the proposed vision-FQL based obstacle avoidance control and compare them to the one proposed in [31]. The main differences between the proposed method and that reported in [31], correspond to the reward function shown in Table II and the evolutionary behavior reported in (7).

For testing, we used three scenarios defined in the CoppeliaSim environment and shown in Fig. 7. The scene in Scenario 1 consists of a curved path, Scenario 2 is a curved path with two consecutive turns and Scenario 3 which is an open maze. For scenarios 1 and 2, we placed a target at the end of their paths and in scenario 3 we placed it on an arbitrary location, the targets are located in the green circles of Fig. 7. The targets have two purposes, to end the simulation if the robot reaches it and start a new iteration, and its location is used to calculate the distance between it and the robot to provide the feedback for the evolutionary system proposed in section III. Three experiments are performed. The first experiment is to compare the ability of the response system and evolutionary methods proposed in Sections II.C and III.B to the ones proposed in [31] to reach the targets of each scenario, the second test will evaluate their ability to find multiple solutions to the same scenario and the third will evaluate whether or not it is effective to train the robot in one scenario with a model from another scenario as a starting point. Because scenario 3 is the most complex, the successful attempt will be the first attempt to get within 10% of the distance to the target.

Fig. 6. Reward response on condition 3. With fixed d_{min} (red plot) and with fixed d_{max} (blue plot).

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

In Tables III and IV, we report the results obtained after performing the three experiments, where R0 corresponds to the response system proposed in [31], R1 corresponds to the response system proposed in Section III.C, and "non evo" and "evo" designations indicate if that algorithm used the evolutionary correction mentioned in Section III.B.

Table III shows the results for experiment 1, reporting the amount of attempts it took each algorithm variant to reach the target in each scenario.

Table III. Results for Experiment 1: Attempts to reach a solution.

Model	Scenario 1	Scenario 2	Scenario 3
R0 non evo	5	14	N/A
R1 non evo	1	5	N/A
R0 evo	2	28	15
R1 evo	1	12	N/A

From Table III, we can see that the proposed response system tends to reach a solution in a lower number of attempts than the one proposed in [31]. However, the evolutionary version of the original response system was the only one to get within 10% of the distance in the third scenario. Fig. 8 shows some successful attempts.

The second experiment focused on the ability of the models to learn multiple solutions to the scenarios, for the case of the third scenario we will only consider those that reached within 10% of the initial distance since none were able to reach the target. Thus, Table IV shows how many solutions were reached by each algorithm.

Table IV. Results for Experiment 2: Ability to continue learning.

Model	Scenario 1	Scenario 2	Scenario 3
R0 non evo	4	3	0
R1 non evo	3	2	0
R0 evo	7	3	3
R1 evo	7	3	0

From Table IV, we can see that the evolutionary models show greater ability to learn multiple solutions than the non-evolutionary approaches.

The last experiment is done to determine if it's better to train the model from another model trained in another scenario, in this case we used the previous scenario (for example a model trained in scenario 1 used in scenario 2), and the R1 response system, the results are shown in Table V.

Table V. Results for Experiment 4: Attempts to reach a solution from a model trained in a previous scenario.

Model	Scenario 1 to 2	Scenario 2 to 3
R1 non evo	12	N/A
R1 evo	46	N/A

Table V, shows that it took a larger number of attempts than the regular starting point to reach a solution to scenario 2, significantly reducing its performance and did not reach a solution for scenario 3. Therefore, there is no evidence that starting from a model trained in another scenario improves training, this may be due to the scenarios not being similar enough to benefit the training of the models.

V. DISCUSSION

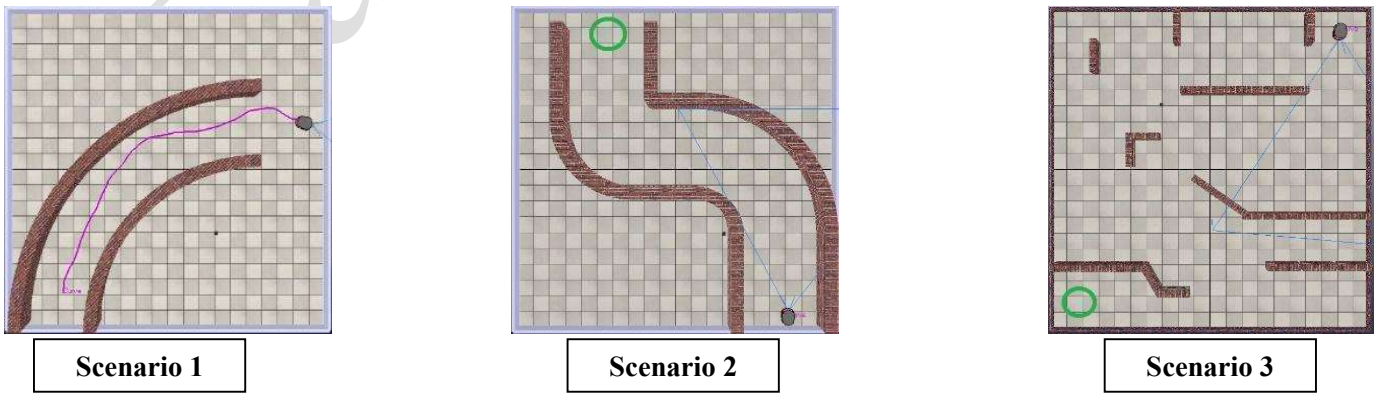


Fig. 7. Testing scenarios defined in the CoppeliaSim environment.

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

In general, the proposed response system tends to get to a solution in a lesser number of attempts than the one proposed in [31]. Although the results are mixed, the evolutionary correction tends to help the system in finding more possible solutions. Starting the training using a model trained in another scenario does not reduce the number of iterations it takes to find a solution.

Implementation is relatively straightforward, however there are some issues especially with the simulation environment. There is some variability on the behavior of the simulation depending on which system runs it. CoppeliaSim struggles with imported models therefore we limited the complexity of the environments, also CoppeliaSim is not as widely used as some other programs and there is a lack of resources to consult on how to implement features, limiting the variety of experiments and environments we could test, and finally the simulator does not work natively with Python and it has to be implemented through the use of an API, therefore we think that it may be beneficial to use a game engine such as Unity or Godot which would allow for more flexibility and have more resources with the disadvantage that the algorithm would have to be translated to another programming language and create new simulation scenes.

The results obtained showed that the reinforced learning, with a vision-based control strategy and an evolutionary reward signal improves the traditional Fuzzy Q-learning strategy by taking into account the time behavior of the robot. These findings are relevant to develop new applications where the continuous learning of the robot is a critical factor, for example, autonomous vehicles that use low-cost and real-time vision-based methods instead of computationally expensive ones such as LiDAR-based methods. On the other hand, it is observed that with the current optical flow system the robot can struggle with change of direction due to new objects appearing on its visual field, when a new object appears the area in which it appears is considered "dense" and overshoots the response as seen on Fig. 9a, this overshoot tends to get the robot in a situation where it's facing directly against a wall where the flow density tends to be low and therefore does not identify it as an obstacle and eventually collides. Another identified failure mode is related to the angle of view of the robot, in some iterations the robot initially avoids the obstacle, however the 60° field of view of the camera is sometimes insufficient to detect objects that are near and to the side of the robot and therefore navigates too close to an obstacle as seen on Fig. 9b.

Fig. 8. Examples of solutions reached by the algorithms, in scenario 1 and 2 by R1 evolutionary algorithm and for scenario 3 the R0 evolutionary algorithm.

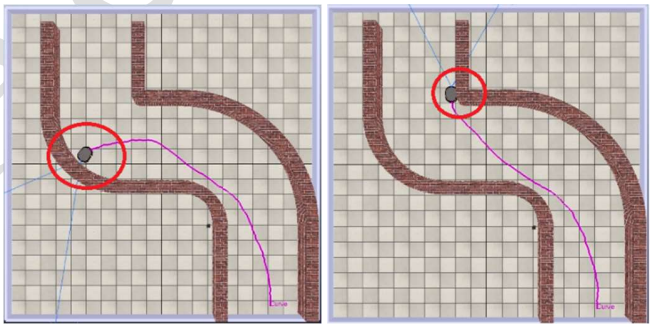
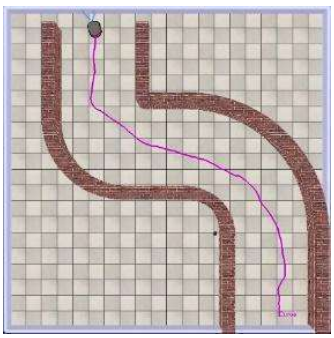
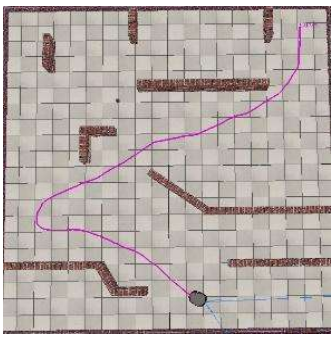


Fig. 9. Identified failure modes. (a) Appearance of new objects. (b) Objects not detected given the viewing angle.





To address these issues there are two proposals, the first would be to add more regions to the optical flow and therefore the rules of the fuzzy controller, five regions

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

instead of three would allow for a more precise control and prevent overshooting when a new obstacle is detected. The second proposal is increasing the field of view of the camera to prevent the second identified failure mode, another proposal is a hybrid approach in which the ultrasonic sensors are used to determine a state in which the robot is too close to an obstacle and does a routine to get the robot in a state where it can continue its navigation using the camera, this would help in both identified failure modes. It would also be possible to use the optical flow as input for another machine learning model such as Primal Policy Optimization.

VI. CONCLUSIONS

In this paper, an FQL and optical flow based autonomous navigation approach was presented. The proposed method incorporated a new reward system including the ultrasonic sensors and an evolutionary component which enables it to keep learning indefinitely without having prior knowledge of its environment or mapping. The results obtained show that the present approach improves the rate of learning compared with a method with a simple reward system and without the evolutionary component. The findings show that the proposal enables us to test more solutions and help us find a model better suited to the scenario it is trained on, allowing the robot to navigate using only one camera while keeping the model relatively light on processing resources.

In future work, we will explore adding more regions to the optical flow and to modify the architecture of the fuzzy system or using a different algorithm such as primal policy optimization. Also, we will incorporate a hybrid approach using the ultrasonic sensors to determine a state in which the robot is too close to an obstacle. In addition, we will consider changing the simulation platform, for example, Unity and Godot are popular alternatives due to abundance of documentation, large community and flexibility.

CONFLICT OF INTEREST STATEMENT

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

[1] L. Claussmann, M. Revilloud, D. Gruyer, and S.Glaser, "A Review of Motion Planning for Highway Autonomous Driving," IEEE Transactions on Intelligent Transportation Systems 21 (5): 1826–48, 2020.

[2] J. Villagra, V. Milanés, J. Pérez Rastelli, J. Godoy, and E. Onieva, "Path and speed planning for smooth autonomous navigation," in IV 2012 - IEEE Intelligent Vehicles Sympo-sium, Alcala de Henares, Madrid, Spain, Jun. 2012.

[3] F. von, Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller and H.-J. Wuensche, "Driving with Tentacles: Integral Structures for Sensing and Motion," Journal of Field Robotics 25 (9): 640–73, 2008.

[4] S. Mitsch, K. Ghorbal and A. Platzer, "On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles," In Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, June 24 - June 28, 2013, edited by Paul Newman, Dieter Fox, and David Hsu. Berlin, Germany.

[5] F. Lachekhab, M. Tadjine, and M. Kesraoui, "Experimental Evaluation of New Navigator of Mobile Robot Using Fuzzy q-Learning," Article. International Journal of Engineering Systems Modelling and Simulation 11 (2): 50–59, 2019.

[6] H. A. Hussein and M. H. Salih, "FPGA implementation of enhanced obstacles avoidance system for robotics," ARPN Journal of Engineering and Applied Sciences, vol. 14, no. 9, p. 1820 – 1830, 2019.

[7] N. A. A. Khalid and M. H. Salih, "Design And Implementation Of Embedded Obstacle Avoidance System On Fpga," ARPN Journal of Engineering and Applied Sciences, vol. 15, no. 6, p. 776 – 783, 2020.

[8] A. A. Aldair, M. T. Rashid, and A. T. Rashid, "Navigation of Mobile Robot with Polygon Obstacles Avoidance Based on Quadratic Bezier Curves," Iranian Journal of Science and Technology - Transactions of Electrical Engineering, vol. 43, no. 4, p. 757 – 771, 2019.

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

[9] M. Nadour and L. Cherroun, "Using flood-fill algorithms for an autonomous mobile robot maze navigation," International Journal of System Assurance Engineering and Management, vol. 13, no. 1, p. 546 – 555, 2022.

[10] M. Samadi Gharajeh and H. B. Jond, "An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system," Ain Shams Engineering Journal, vol. 13, no. 1, 2022.

[11] C. Zhou, F. Li, and W. Cao, "Architecture design and implementation of image based autonomous car: THUNDER-1," Multimedia Tools and Applications, vol. 78, no. 20, p. 28557 – 28573, 2019.

[12] P. G. Luan and N. T. Thinh, "Real-time hybrid navigation system-based path planning and obstacle avoidance for mobile robots," Applied Sciences (Switzerland), vol. 10, no. 10, 2020.

[13] R. Fareh, T. Rabie, S. Grami, and M. Baziyad, "A vision-based kinematic tracking control system using enhanced-prm for differential wheeled mobile robot," International Journal of Robotics and Automation, vol. 34, no. 6, p. 654 – 667, 2019.

[14] B. Li, J. P. Munoz, X. Rong, Q. Chen, J. Xiao, Y. Tian, A. Arditi, and M. Yousuf, "Vision-Based Mobile Indoor Assistive Navigation Aid for Blind People," IEEE Transactions on Mobile Computing, vol. 18, no. 3, p. 702 – 714, 2019.

[15] Y. Shou, "Obstacle Avoidance Path Planning Algorithm of an Embedded Robot Based on Machine Vision," Mobile Information Systems, vol. 2022, 2022.

[16] S. Wang, L. Wang, X. He, and Y. Cao, "A monocular vision obstacle avoidance method applied to indoor tracking robot," Drones, vol. 5, no. 4, 2021.

[17] P. Qian, N. Xu, C. Fu, and S. Deng, "Mapping and autonomous obstacle avoidance of mobile robot based on ros platform," Manufacturing Technology, vol. 23, no. 4, p. 504–512, Sep. 2023.

[18] H. Zhang, J. Li, R. Shu, H. Wang, and G. Li, "Research on dynamic obstacle avoidance method of manipulator based on binocular vision," Recent Patents on Engineering, vol. 16, no. 6, Nov. 2022.

[19] J. Yan, L. Zhang, X. Yang, C. Chen, and X. Guan, "Communication-aware motion planning ofauv in obstacle-dense environment: A binocular vision-based deep learning method," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 12, p. 14927–14943, Dec. 2023.

[20] H. Farahat, S. Farid, and O. E. Mahmoud, "Adaptive Neuro-Fuzzy control of Autonomous Ground Vehicle (AGV) based on Machine Vision," Journal of Engineering Research, vol. 163, p. 1 – 15, 2019.

[21] R. Miyamoto, M. Adachi, H. Ishida, T. Watanabe, K. Matsutani, H. Komatsuzaki, S. Sakata, R. Yokota, and S. Kobayashi, "Visual navigation based on semantic segmentation using only a monocular camera as an external sensor," Journal of Robotics and Mechatronics, vol. 32, no. 6, p. 1137 – 1153, 2020.

[22] T.-V. Dang, D.-M.-C. Tran, and P. X. Tan, "IRDC-Net: Lightweight Semantic Segmentation Network Based on Monocular Camera for Mobile Robot Navigation," Sensors, vol. 23, no. 15, 2023.

[23] T.-V. Dang and N.-T. Bui, "Obstacle Avoidance Strategy for Mobile Robot Based on Monocular Camera," Electronics (Switzerland), vol. 12, no. 8, 2023.

[24] Y. Zhang, "Cooperative control method of robot formation movement path based on machine vision," Journal of Computational Methods in Sciences and Engineering, vol. 22, no. 6, p. 2093 – 2105, 2022.

[25] P. Y. Glorennec, and L. Jouffe, "Fuzzy Q-learning," In IEEE International Conference on Fuzzy Systems, 2:659–62, 1997.

[26] K. Anam, P. Prihastono, H. Wicaksono, R. Effendi, IA. Sulistijono, S. Kuswadi, A. Jazidie, and M. Sampei, "Embedded Learning Robot with Fuzzy q-Learning for Obstacle Avoidance Behavior," In LSP-conference proceeding, 2009.

[27] E. Lopez-Lozada, E. Rubio-Espino, J. H. H. Sossa-Azuela, and V. H. Ponce-Ponce, "Reactive Navigation under a Fuzzy Rules- Based Scheme and Reinforcement Learning for Mobile Robots," PeerJ Computer Science 7: 1–25, 2021.

This article has been accepted for publication in a future issue of this journal, but it is not yet the definitive version. Content may undergo additional copyediting, typesetting and review before the final publication.
Citation information: Roberto Cuesta-Solano, Ernesto Moya-Albor, Jorge Brieva and Hiram Ponce, A Vision-based Robotic Navigation Method Using an Evolutionary and Fuzzy Q-Learning Approach, *Journal of Artificial Intelligence and Technology* (2024), DOI: https://doi.org/10.37965/jait.2024.0511

[28] A. Singh, M. Shakeel, V. Kalaichelvi, and R. Karthikeyan, "A Vision-Based Bio-Inspired Reinforcement Learning Algorithms for Manipulator Obstacle Avoidance," Electronics (Switzerland), vol. 11, no. 21, 2022.

[29] M. R. Mohd Romlay, A. Mohd Ibrahim, S. F. Toha, P. De Wilde, I. Venkat, and M. S. Ahmad, "Obstacle avoidance for a robotic navigation aid using Fuzzy Logic Controller-Optimal Reciprocal Collision Avoidance (FLCORCA)," Neural Computing and Applications, vol. 35, no. 30, p. 22405 – 22429, 2023.

[30] Moya-Albor, S. L. Coronel, H. Ponce, J. Brieva, R. Chavez-Dominguez and A. E. Guadarrama-Munoz, "Bio-Inspired Optical Flow-Based Autonomous Obstacle Avoidance Control," In 2019 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE). IEEE, 2019.

[31] E. Moya-Albor, J. Brieva, H. Ponce, and S. L. Gomez-Coronel, "Optical Flow-Hermite and Fuzzy q-Learning Based Robotic Navigation Approach," In 2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE). IEEE, 2021.

[32] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," Artificial Intelligence 17: 185–203, 1981.

[33] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: a Versatile and Scalable Robot Simulation Framework," 2013 in Proc. Of The International Conference on Intelligent Robots and Systems (IROS), 2013.