Obstacle Avoidance for a Robotic Navigation Aid using Fuzzy Logic Controller-Optimal Reciprocal Collision Avoidance (FLC-ORCA)

Muhammad Rabani Mohd Romlay[a,*], Azhar Mohd Ibrahim[a], Siti Fauziah Toha[a], Philippe De Wilde[b], Ibrahim Venkat[c] & Muhammad Syahmi Ahmad[a]

[a] *Department of Mechatronics Engineering, International Islamic University Malaysia, Jalan Gombak, Kuala Lumpur, Malaysia*
[b] *Divison of Natural Sciences, University of Kent, Canterbury, United Kingdom*
[c] *School of Computing and Informatics, Jalan Tunku Link Gadong, Universiti Teknologi Brunei, Bandar Seri Begawan, Brunei Darussalam*

Abstract

Robotic Navigation Aids (RNAs) assist visually impaired individuals in independent navigation. However, existing research overlooks diverse obstacles and assumes equal responsibility for collision avoidance among intelligent entities. To address this, we propose Fuzzy Logic Controller-Optimal Reciprocal Collision Avoidance (FLC-ORCA). Our FLC-ORCA method assigns responsibility for collision avoidance and predicts the velocity of obstacles using a LiDAR-based mobile robot. We conduct experiments in the presence of static, dynamic, and intelligent entities, recording navigation paths, time taken, angle changes, and rerouting occurrences. The results demonstrate that the proposed FLC-ORCA successfully avoids collisions among objects with different collision avoidance protocols and varying liabilities in circumventing obstacles. Comparative analysis reveals that FLC-ORCA outperforms other state-of-the-art methods such as Improved A* and Directional Optimal Reciprocal Collision Avoidance (DORCA). It reduces the overall time taken to complete navigation by 16% and achieves the shortest completion time of 1 minute and 38 seconds, with minimal rerouting (1 occurrence) and the smallest angle change (12°). Our proposed FLC-ORCA challenges assumptions of equal responsibility and enables collision avoidance without pairwise manoeuvres. This approach significantly enhances obstacle avoidance, ensuring safer and more efficient robotic navigation for visually impaired individuals.

Keywords: Obstacle avoidance, fuzzy logic, optimal reciprocal collision avoidance, navigation aid & electronic travel aid.
Corresponding author: banie91@gmail.com/ rabani.romlay@live.ium.edu.my

1 Introduction

For visually impaired people, the most common navigation aid is white canes. While this gives an excellent solution for a near-ground obstacle, uneven surfaces lower than the ground or obstacles higher than knee level remain undetected. Difficult circumstances such as moving within a crowd, hanging tree branches and potholes might also cause problems to visually impaired people. Hence, researchers are keen to explore scientific advancements which could help the visually impaired community with self-navigating. Mobility aids for visually impaired people known as Robotic Navigation Aids (RNAs) are equipped with measurement systems to detect

objects and avoid collision [1]. Some of the objectives and challenges of RNA includes detection of the obstacle, information on the travel surface, location of landmarks, identification information, self-familiarization and mapping of the surrounding [2]. However, one of the most fundamental and essential objectives of an RNA is to avoid collision during navigation.

The aim of navigation is to look for an optimal or suboptimal path from the starting point to the destination point with obstacle avoidance competence [3]. Within the field of mobile robotics, navigation can be classified into two categories which is the global navigation and local navigation. Global navigation requires prior knowledge of the surroundings to achieve movement, whereas local navigation determines the decision and control of its motion and orientation autonomously using equipped sensors [3]. Here, the focus will be on local navigation, which is the ability to avoid the different types of obstacles with varying states of nature.

The main focus of this research is to enable safe and collision-less self-walking for visually impaired people in environments with obstacles in different states of nature. This research classifies static, dynamic, and intelligent entities and attempts to avoid collisions of these obstacles. Hurdles with varying densities and different states of cooperativeness are also included in the experiment set-up.

Through solid collision avoidance protocol in RNA, visually impaired people can be more self-independent, less likely to be involved in accidents, improve their reachability and finally improve their living lifestyle.

The challenge faced within the field of RNA includes the conflicting issue of intelligent entity implying dissimilar obstacle avoidance strategy with non-identical responsibility of obstacle avoidance without central communication to the other agents.

In conclusion, the research results in main contributions as below:

- Formulation of a fuzzy model which predicts and updates the obstacles' nature of movement based on its characteristics of velocity, density and acceleration, moving within surroundings of obstacles with different natures of movement and varying states of cooperation.
- Hybridization of FLC-ORCA to solve the conflicting issue of intelligent entity implying dissimilar obstacle avoidance strategy with non-identical responsibility of obstacle avoidance without central communication with the other agents.
- Systematic analysis on obstacle avoidance methods with static, dynamic and intelligent entities with a different consensus of cooperation, density level and path routes in between navigation.

This section sums up the introduction of the research, followed by the related works and research gap within the area of obstacle avoidance in section 2. Section 3 presents the proposed novel method of FLC-ORCA construction and formulation. Sequentially, it is followed by results and discussion of the navigation experiment within the software and human subject navigation in section 4. And finally, the conclusion in section 5 discusses the contribution and future direction of the research.

2 Related Works

Robotic navigation techniques allow transmissions of local environment information for systems such as crop fields in agricultural production [4], exploring the unknown environments in a legacy nuclear facility or identifying locations with sources of ionising radiation [5]. The help of recent progress in robotic technologies has led to the availability of cheaper autonomous systems [6]. One of the major use of Robotic Navigation Aid (RNA) is to guide and navigate visually impaired people [7]–[11]. Within the field of mobile robotics, navigation can be classified into two categories which are global navigation and local navigation [12]. Global navigation requires prior knowledge of the surroundings to achieve movement, whereas local navigation determines the decision and control of its motion and orientation autonomously using equipped sensors [3].

In order to achieve the objective of local navigation and safely avoid collisions or obstacles, various machine-learning methods have been initiated [13]–[15]. Fuzzy logic systems have been successfully implemented within recognition technology [16], [17], fuzzy clustering [18] and complicated engineering problems [19], [20] due to their effective reasoning and learning capacity [21]. For instance, Kasmi and Hassan [22] attempted Fuzzy Logic (FL) methods and Interval Type-2 FL (IT-2FL) controllers to enable navigation and facilitate the robot within uncertainties and track optimal trajectory. Neural Network (NN) methods have also been suggested to solve prediction, recognition and image processing [23]–[27]. Chen et al., [28] demonstrated a NN-based end-to-end deep reinforcement learning for mobile robot navigation with map-based obstacle avoidance. Another variation of NN, the Adaptive Neuro Fuzzy Inference System (ANFIS) is also implemented in various fields which lack sufficient data or have vague information [29]–[32]. Studies have shown ANFIS to significantly enhance the FIS model with promising results within nature-based prediction such as solar radiation [29], air temperature [31], evapotranspiration [32] and soil temperature [30]. With results produced promising output when compared with other learning method of Artificial Neural Network (ANN) and Multiple Linear Regression (MLP). Meanwhile, Shentu et al., [33] attempted to solve the issue of obstacle avoidance through the Genetic Algorithm (GA) approach using a Mobile Parallel Robot (MPR) consisting of two phases of the genetic algorithm. Whereas, Ajeil et al., [34] have chosen Particle Swarm Optimization (PSO) with two novel algorithms for intelligent path planning within static and dynamic obstacles. As PSO have shown promising results for global optimization algorithm and clustering [35], [36], the authors decided to implement the Modified Frequency Bat (MFB) algorithm and the Hybrid Particle Swarm Optimization-Modified Frequency Bat (Hybrid PSO-MFB) algorithms for the objective of obstacle detection and avoidance.

Amongst various methods and techniques of avoiding collisions, the Optimal Reciprocal Collision Avoidance (ORCA) method is highly regarded to be collision-free in multi-agent and huge crowd situations [37]–[39]. Hajj simulation or video games with multiple units have successfully implied the ORCA technique to avoid the collision of multiple agents in a narrow area [40], [41]. This method does not only consider the state of avoiding static and moving object but also contemplate moving object with collision avoidance reasoning [42].

Berg, Lin and Manocha [42], proposed a concept of local reactive collision, namely Reciprocal Velocity Obstacle (RVO). This method does not only consider the state of avoiding static and moving object but also contemplate moving object with collision avoidance reasoning themselves. Inspired by the Velocity Obstacle (VO) [43] concept which is simple, well defined and generally applicable, RVO inherits the merits of the technique with additional capability for multi-agent navigation.

Snape et al., [37] suggest an improved model of RVO as they claimed that implementation of RVO in mobile robots often leads to reciprocal dance; a circumstance where mobile agents are unable to reach a consensus on which side to pass each other. Then RVO is further optimized with the Optimal Reciprocal Collision Avoidance (ORCA). The ORCA enhances RVO by removing the necessity of central coordination between the mobile agent. Berg et al., [44] present an approach of distributed-multi robot navigation which moves independently and simultaneously without any form of communication. Similar to RVO the model offers a solution in different circumstances such as a static object, moving object and moving obstacle which tries to avoid collision itself. These circumstances highly resemble an unknown outdoor environment.

Alonso-mora et al., [45] extend ORCA to robot applications with non-holonomic by tracking holonomic trajectory. They reduce the possibility of collision which may arise during velocity changes by enlarging the radius of the robot. The continuity of expected velocity and actuator velocity is accomplished through a trajectory tracking strategy, which enables it to be implemented in a car-like robot. Thus, it is deemed suitable to implement ORCA for navigation systems for the visually impaired. Other applications in the real world include the robot swarm exhibit [46] where real-time validation for 50 robots that move in an enclosed area. The robot was embedded with modified ORCA which gives better performance. It can detect static obstacles, move under the dynamic and physical constraints of real-world robots, and reduce sources of noise in position and velocity information.

As the ORCA method is quickly gaining attention due to its strong theoretical foundation and prominence in academic literature [46], [47], some opted for hybrid ORCA with other control methods. Cheng et al.,[48] implied the ORCA algorithm with Model Predictive Control (MPC). The ORCA-MPC combined approach incorporates the merits of both methods, with the ORCA algorithm enabling each agent to self-determine its permitted velocity and the MPC method to anticipate future trajectories of a system based on its system dynamics and various constraint.

Because of the efficiency in avoiding collision, the geometrical self-separation assurance algorithms such as the ORCA and other modified VO approaches are commonly adopted in various fields, such as in the field of robotics [49], aviation [50], marine [51] and pedestrian-assistant [52]. The modified VO algorithms are suitable to solve the problem of separation assurance for the agents with the characteristics of 'low-altitude, slow-speed, small-size'[53]. It is one of the most popular decentralized multi-agent path planning methods [54], [55]. ORCA enables each agent to efficiently construct a new collision-free velocity without communication, by first generating half-plane constraints with respect to all neighbours and next selecting within the convex region constituted by the intersection of these constraints a velocity closest to its preferred velocity [54].

Though proven to be a success in complex simulations with thousands of agents in densely packed scenarios ORCA does possess some limitations [54]. Janardan [56], states that ORCA is difficult to implement in real-life platforms. This is because research is made with the assumption of perfect sensing, where obstacles' shape, position and velocity are assumed to be known. Thus, the error in measurement obtained from the sensors is ignored. Berg et al.,[44] also state that the simulation is done assuming other robots take the same collision-avoidance protocol.

The ORCA algorithm is efficient for separation assurance, but it essentially requires pairwise manoeuvres for vehicles involved in possible collisions [53]. It assumes that other entities are implying the same protocol of manoeuvres and collision avoidance as itself. The vehicle only needs to generate the collision-free trajectory for itself, without motion planning for any other entity involved in the circumstances of the collision. In some cases, it achieves self-separation with global navigation but is unable to achieve local and specific information during mid-navigation [53]. Especially in real-time collision avoidance where information needed to be in a particular and peculiar manner.

ORCA presumes that each agent takes half of the collision avoidance responsibility [57]. That means each agent takes half effort to avoid the collision. Most ORCA variants such as [58]–[60], stick to the original ORCA's equal responsibility assigned among the agents. A few noteworthy exceptions are 3D-ORCA [61] and AVO [62], which give out responsibilities among a pair of agents according to the relative volumes of their respective velocity spaces induced by the kinematic and dynamic constraints, with the fundamental idea that agents are more dynamically constrained will receive less responsibility. In A-ORCA [63], the responsibility is shared among car-like robots by maximizing the product of the average ratio of the ORCA velocity set volume, and the initial velocity set volume together with a function indicating fairness. An agent with a larger difference between its current velocity and preferred velocity would take more responsibility, but it assumes each agent has the information of other agents' preferred velocities, which may not be practical in real-life circumstances [54].

The development of the FLC-ORCA is proposed to provide intact and secure collision-less navigation. ORCA proved to be a success in complex simulation and densely packed scenarios, while the proposed FLC complements the

system in terms of velocity computation of incoming objects and predicting responsibility of avoidance to avoid the collision. Fuzzy logic improves the limitation of ORCA in assuming all entity bears equal responsibility for avoiding collision and enables avoidance with entities having different obstacle avoidance protocol than themselves. With the integration of two concepts of Artificial Intelligence (AI) merged; Fuzzy Logic Controller and Optimum Reciprocal Collision Avoidance (FLC-ORCA), robotic navigation can achieve efficient decision and control of its motion. Therefore, reducing the possibility of harmful incidents of collision with objects passing by or the sudden emergence of foreign bodies.

Navigation is a broad subject in and of itself. The research scope is within the area of obstacle avoidance methodology, specifically local navigation which determines the decision and control of the robotic motion and orientation autonomously using equipped sensors [3]. The main focus of this research is to enable safe and collision-less robotic navigation with obstacles within different states of nature, diversified density levels and varying states of cooperativeness.

Table 1 shows the comparison of obstacle avoidance methods involving ORCA and its recent state-of-the-art variation, including its contribution and limitation. Although there are improvements in ORCA variation over the years, there are still limitations in difficulty to implement in the real-life platform [56] due to the assumption of perfect sensing within the simulation, the presupposition that another object applies similar collision avoidance protocol [44], [53], neglecting dynamic constraint [54], the presumption of current velocity as its preferred velocity [54], and for the most part, simulation are done based on simulation only without implementation on a hardware platform [54], [57], [64]. The proposed method FLC-ORCA relaxes the presumption of equal responsibility between the colliding objects to avert collision while predicting the velocity of the incoming objects to improve sensing and enable real-time application on the mobile robots. The evaluation conducted on our proposed method involves static, dynamic, intelligent entity obstacles, possessing the living and non-living subjects, having obstacles with both states of cooperative and non-cooperative to avoid collision. Additionally, irregular or flexible speed is applied to the intelligent entity object. Dissimilar obstacle avoidance protocols are implied to the scene, thus proving the ability of FLC-ORCA to steer clear of crashing regardless of the obstacle's collision-avoiding protocol. Another important factor which should be considered is that both the navigation robot and the mobile robot obstacle do not share the same host CPU. Therefore, supporting the claim that our proposed method of FLC-ORCA successfully avoids collision without prior communication with other external objects. Table 2 shows the comparison of the other novel techniques of obstacle avoidance in terms of experiments and methods of evaluation.

Table 1 Comparison of obstacle avoidance techniques involving ORCA.

Method	Contribution	Drawback/ Limitation	Ref
Reciprocal Velocity Obstacle (RVO)	This method does not only consider the state of avoiding static and moving object but also contemplate moving object with collision avoiding reasoning themselves	Implementation of RVO in mobile robots often leads to reciprocal dance; a circumstance where mobile agents are unable to reach a consensus on which side to pass each other.	[42]
Optimal Reciprocal Collision Avoidance (ORCA)	Enhances RVO by removing the necessity of central coordination between the mobile agents. Presents an approach of distributed-multi robot navigation which moves independently and simultaneously without any form of communication.	Janardanan [56], states that ORCA is difficult to implement in real-life platforms. This is because research is made with the assumption of perfect sensing, where obstacles' shape, position and velocity are assumed to be known. Berg *et al*. [44], state that the simulation is done assuming other robots take the same collision-avoidance protocol.	[44]
Local collision avoidance among non-holonomic robots (NH-ORCA)	Extends ORCA to robot application with non-holonomic by tracking holonomic trajectory. They reduce the possibility of collision which may arise during velocity changes by enlarging the radius of the robot. The continuity of expected velocity and actuator velocity is accomplished through a trajectory-tracking strategy.	NH-ORCA requires a combination with global path planning to look closer at the avoidance of deadlock situations.	[58]
ORCA algorithm with Model Predictive Control (ORCA-MPC).	Hybrid ORCA with another control method. The combined approach incorporates the merits of both methods, with the ORCA algorithm enabling each agent to self-determine its permitted velocity and the MPC method to anticipate future trajectories of a system based on its system dynamics and various constraint.	The proposed method was tested on a simulation. Some parameters such as the radius of the obstacles, and control input constraint are assumed to be known.	[57]
Directional ORCA	Improvement of the ORCA algorithm through the introduction of Directional ORCA (DORCA). The authors employed a vector rotation mode to construct the forbidden Velocity Obstacle (VO) in order to improve the computation efficiency.	However, in DORCA it is assumed that different agents or entities have the same motion characteristics and perfect symmetry. The deterministic and global direction selectivity of collision-free manoeuvres is all achieved using the same obstacle avoidance algorithm. All moving agents comply with the unified rules of DORCA.	[53]
Variable Responsibility Optimal Reciprocal Collision Avoidance (VR-ORCA)	Variable Responsibility ORCA (VR-ORCA) to relaxes the assumptions of equal responsibility shared between two agents within a collision path.	The proposed VR-ORCA does neglect the dynamic constraints, complex kinematics and shapes of real robots in three dimensions. Additionally, the algorithm assumes the current velocity is estimated as its preferred velocity. The algorithm has not been implemented on a hardware platform, therefore ignoring the uncertainty of acting and sensing in real robots.	[54]

Table 2 Comparison of obstacle avoidance techniques in various types of obstacles in terms of experiments conducted.

Category	Ref	Method Proposed	Obstacle Considered			Test and Evaluation		Comment
			Static	Dynamic	I.E.	Simulation	Experiment	
Fuzzy Logic	[22]	FL & Interval Type-2 FL	✓	✓	×	✓	×	Static and dynamic non-cooperative obstacles only. Software MATLAB platform without real hardware navigation.
	[65]	FLC and Optical Flow	✓	×	×	✓	×	Although there is no moving obstacle, there is a presence of a moving target predetermined to be the moving destination target within the simulation. Simulation works only.
	[66]	Same Fuzzy Logic Controller (SFLC)	✓	×	×	✓	×	Focus on solving deadlock situations among static obstacles only. Software simulation only.
	[67]	Fuzzy-based Obstacle Avoidance Controller	✓	×	✓	✓	✓	Static and slowly approaching dynamic objects, are considered but not specifically addressed to be within the class of intelligent entities. Software and hardware simulation.
	[68]	Duelling Double DQN	✓	✓	×	✓	✓	Involve static and dynamic obstacles. The intelligent entity is in the form of a pedestrian, but its separation from dynamic systems does not acknowledge & does not show characters of different behaviour.
Neural Network	[23]	Fuzzy Neural Network controller with Kalman filter	✓	×	×	✓	✓	The experiment was conducted on static, small in size obstacles within an outdoor environment.
	[69]	Behaviour-NN (BNN)	✓	×	×	✓	✓	Navigation within an enclosed area with static obstacles only.
	[25]	Recursive NN Controller (RCNN)	✓	×	×	✓	×	Have not been tested within moving obstacles and real hardware experiments.
	[70]	Multimodal deep reinforcement learning (DRL)	✓	✓	×	✓	✓	Static and dynamic obstacles are involved. The focus is on moving within a narrow area. The pedestrians are considered dynamic obstacles but did not possess decision-making criteria in the experiment.
	[34]	PSO-MFB	✓	✓	×	✓	×	Simulations are done in a free space modelling in the MATLAB environment. The objective of a moving target environment is a work in progress.
PSO	[28]	PSO-based AFP	✓	✓	×	✓	×	The motion control, path planning and physical experiment have not been done for now.
	[64]	PSO-tuned FNN	✓	×	×	✓	×	Simulation runs on MATLAB, with static obstacles only and a goal destination within a confined rectangular space.
	[33]	GA for trajectory planning	✓	×	×	✓	✓	Specified for MPR for trajectory tracking controller and move vehicle cooperatively and asymptotically.
GA	[71]	Improved GA	✓	×	×	✓	×	Simulation of QT software for cooperative collision avoidance by multiple unmanned vehicles.
	[72]	Parallel GA	✓	×	×	✓	✓	Designed to avoid inter-robot collisions using omnidirectional robots.

3 Method

In the method section, first the mathematical modelling of FLC-ORCA will be presented in subsection 3.1. The novel methods of configuring the responsibility of averting collision and prediction of moving obstacle's optimized velocity are hybridized into the mathematical modelling of ORCA. This is followed by the description of FLC 1 and FLC 2, including their fuzzy rule base, membership function, inference method and finally defuzzification to complete the proposed technique in subsection 3.2. Fig. 1 shows the framework of the proposed method.

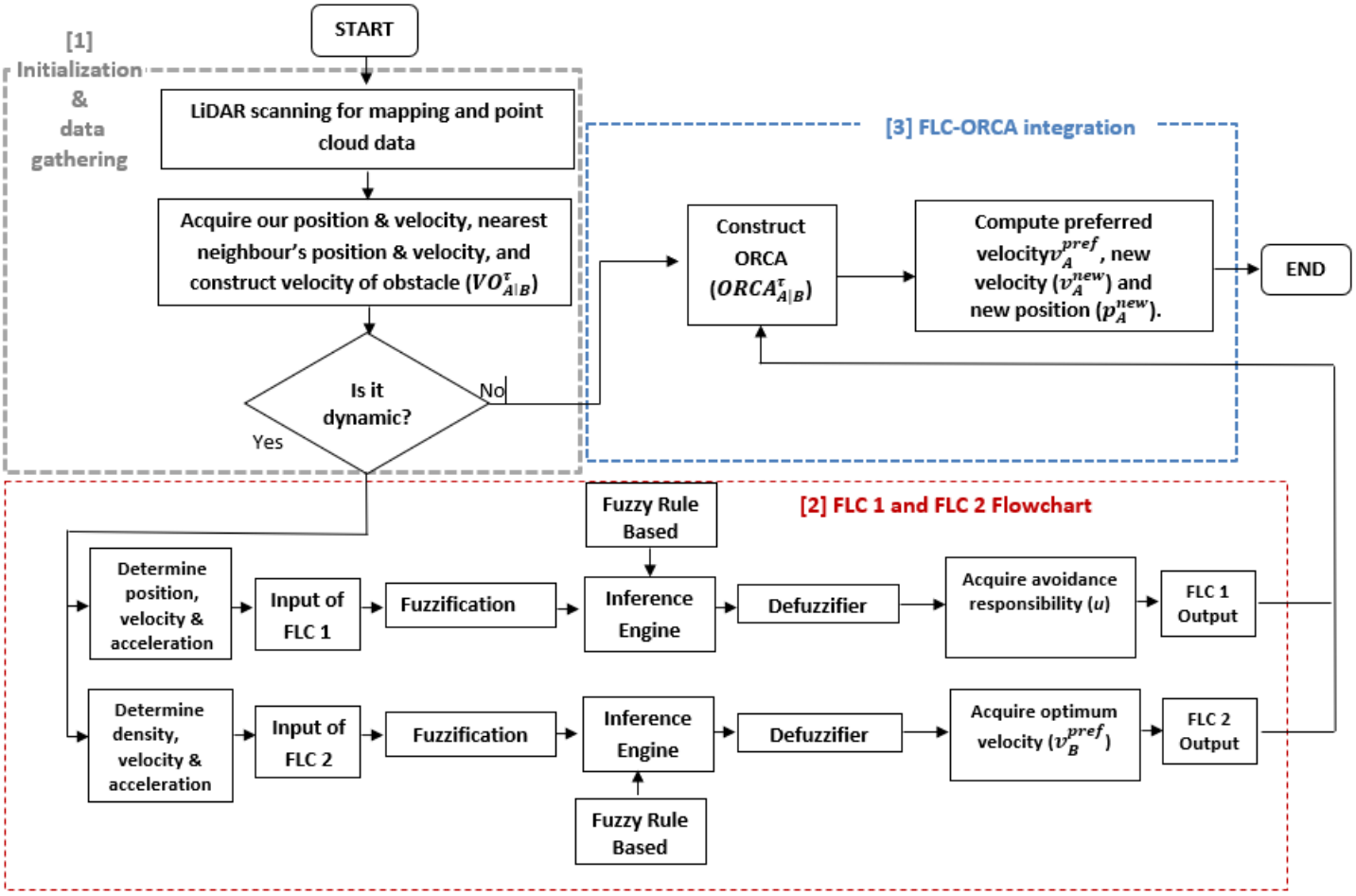


Fig. 1 Framework of the proposed FLC-ORCA.

3.1 Fuzzy Logic Controller-Optimal Reciprocal Collision Avoidance (FLC-ORCA)

In this subsection, the mathematical modelling for the integration of FLC-ORCA will be discussed. The original algorithm of ORCA can be found here [44]. Our modification of ORCA started with the fundamentals of velocity obstacle between two agents ***A*** and ***B***, where velocity obstacle $VO^{\tau}_{A|B}$ is the set of all relative velocities of ***A*** with respect to ***B*** that will result in a collision between ***A*** and ***B*** before time τ.

$$VO^{\tau}_{A|B} = \{v|\exists t \in [0,\tau] :: t\boldsymbol{v} \in D(p_B - p_A, r_A + r_B)\} \quad (1)$$

Let *D(p, r)* denote an open disc of radius *r* centred at *p* with p_A and p_B is the centre position of ***A*** and ***B.*** While r_A and r_B being the radius of object ***A*** and object ***B*** respectively.

In the proposed novel method, FLC 1 replaces the factors of responsibility $\boldsymbol{u}$, the smallest change required to avert collision in τ time frame. Which normally assumed to be $^1/_2$ to symbolize equal collision avoidance responsibility between the two agents.

$$ORCA^{\tau}_{A|B} = \left\{v|\left(v - \left(v_A^{opt} + \boldsymbol{xu}\right)\right).n \geq 0\right\} \quad (2)$$

Where the value of $\boldsymbol{x}$, will be determined by FLC 1. Here, FLC 1 determines the factors of responsibility of $\boldsymbol{u}$, based on the position, velocity and acceleration of the incoming obstacle. $\boldsymbol{ORCA^{\tau}_{A|B}}$ denotes velocities closest to preferred velocities when compared to any other pair of sets of

reciprocally collision-avoiding velocities. $\boldsymbol{v}_A^{opt}$ is the optimum velocity of $\boldsymbol{A}$. While $\boldsymbol{n}$ is the normal of boundary $VO_{A|B}^{\tau}$ at point $\left(\boldsymbol{v}_A^{opt} - \boldsymbol{v}_B^{opt}\right) + \boldsymbol{u}$.

To determine responsibility $\boldsymbol{u}$, it is usually assumed that each agent shares the information of the other agent's preferred velocity, which is not practical in real-life circumstances [54]. The obstacle optimum velocity $\boldsymbol{v}_B^{opt}$ is usually assumed to be its preferred velocity $\boldsymbol{v}_B^{pref}$. This is where the second adjustment on ORCA is initiated. The optimum velocity $\boldsymbol{v}_B^{opt}$ is estimated in FLC 2 taking into consideration of its velocity, acceleration and density of the surrounding.

$$\boldsymbol{u} = \left(\underset{\boldsymbol{v} \in VO_{A|B}^{\tau}}{\arg min} \left|\left|\boldsymbol{v} - \left(\boldsymbol{v}_A^{opt} - \boldsymbol{y}\right)\right|\right| \right) - \left(\boldsymbol{v}_A^{opt} - \boldsymbol{y}\right) \qquad (3)$$

Where $\boldsymbol{y}$ is the optimum velocity of object B v_b^{opt}, which will be determined by FLC 2.

Then, the preferred velocity of A is calculated, $\boldsymbol{v}_A^{pref}$. Within each cycle, the radius, current position and optimization velocity of other obstacles are acquired. Based on this, the robot infers the permitted half-plane of allowed velocities induced by each entity.

$$ORCA_A^{\tau} = D(0, v_A^{max}) \cap \underset{B \neq A}{\cap} ORCA_B^{\tau} \qquad (4)$$

Next, the robot selects a new velocity $\boldsymbol{v}_A^{new}$ for itself that is closest to its preferred velocity $\boldsymbol{v}_A^{pref}$

$$\boldsymbol{v}_A^{new} = \underset{\boldsymbol{v} \in ORCA_A^{\tau}}{\arg min} ||\boldsymbol{v} - \boldsymbol{v}_A^{new}|| \qquad (5)$$

Finally, the agent achieved its new position.

$$\boldsymbol{p}_A^{new} = \boldsymbol{p}_A + \boldsymbol{v}_A^{new} \Delta \boldsymbol{t} \qquad (6)$$

Then, the process of updating the position and velocity of the obstacle is repeated for the next cycle. Fig. 2 shows the flowchart of the proposed algorithm.

3.2 Fuzzy Logic Control 1 (FLC 1) and Fuzzy Logic Control 2 (FLC 2)

In this subsection, a fuzzy logic control method that anticipates the mobile robot's avoiding responsibility is constructed. The Fuzzy Logic Control 1 (FLC 1) takes the position of the obstacle, the velocity of the obstacle and its acceleration as the system inputs. For Fuzzy Logic Control 2 (FLC 2), 3 inputs are taken as its input viz., the velocity of the obstacle, the density of the surroundings and the acceleration of the obstacle. The output of the FLC 2 configures the expected velocity of the obstacle in the next scanning cycle.

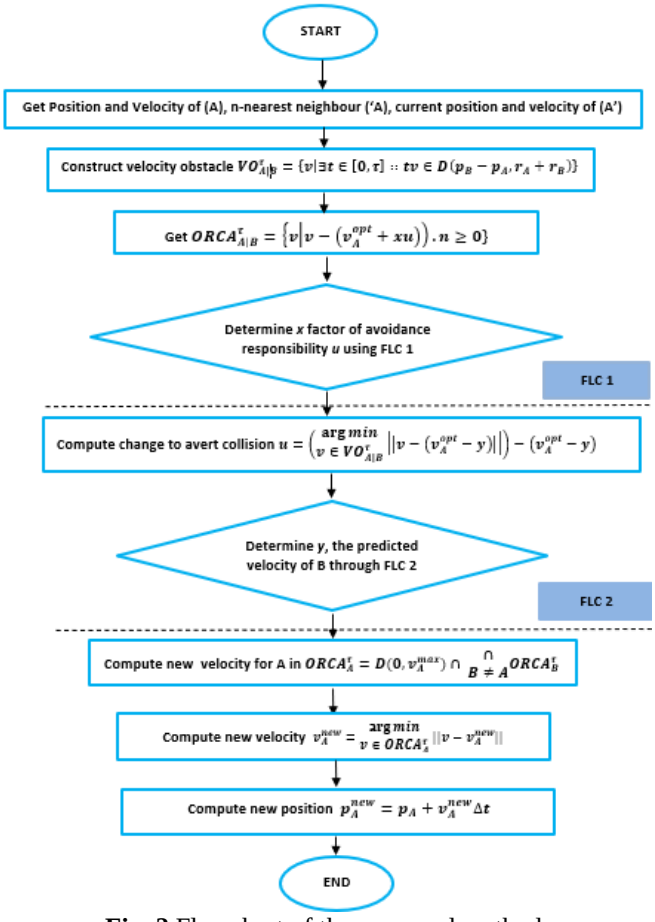


Fig. 2 Flowchart of the proposed method.

3.2.1 Fuzzy Rule Base

First, the fuzzy inputs membership functions are placed into a rule-based form to determine their output. From this, the desired relationship between the variables of the input and the output is derived.

Table 3 shows the fuzzy rule *base* for FLC 1. The range of the distance is covered by 4 fuzzy sets which are *Very Near (VN)*, *Near (N)*, *Far (F)* and *Very Far (VF)*. The range of velocity is also classified into 4 fuzzy sets namely *Very Slow (VS)*, *Slow (S)*, *Fast (F)* and *Very Fast (VF)*. The acceleration of the obstacle is divided into categories of *Decelerate (DCC)*, *Zero Acceleration (ZERO)* or *Accelerate (ACC)*. The output of the FLC 1 controls the avoidance responsibility of the mobile robot agent. As the number of rules depends on the combinations of the input membership function, there are 48 rules. For the output in Table 3, the avoidance responsibility value varies from 0 to 1 (a=0, b=0.14, c=0.29, d=0.43, e=0.57, f=0.71, g=0.86, h=1).

Table 3 Fuzzy rule base of FLC 1.

Acceleration	DCC				ZERO				ACC			
Position / Velocity	VF	F	N	VN	VF	F	N	VN	VF	F	N	VN
VS	a	a	b	b	c	c	d	d	f	f	g	g
S	a	a	b	b	c	d	e	e	f	f	g	g
F	b	b	c	c	d	e	e	f	g	g	h	h
VF	b	b	c	c	d	e	f	f	g	g	h	h

Table 4 shows the fuzzy rule base of FLC 2 with 36 rules. The velocity is categorized into *Very Slow, Slow, Fast* and *Very Fast.* Whereas for density, the surroundings are classified into 3 fuzzy sets which are *High (H), Medium (M)* and *Low (L).* For the 3rd input, similar to FLC 1 the system separates acceleration *Deceleration (DCC), Zero Acceleration (ZERO)* and *Accelerate (ACC).* The output of the system in Table 4 which is the expected velocity would be a multiplier of the current velocity of the obstacle (a=0, b=0.33, c=0.667, d=1, e=1.33, f=1.667, g=2).

Table 4 Fuzzy rule base of FLC 2.

Acceleration	DCC				ZERO				ACC			
Velocity / Density	VS	S	F	VF	VS	S	F	VF	VS	S	F	VF
High	a	a	b	b	c	c	d	d	e	e	f	f
Medium	a	b	b	c	c	d	d	e	e	f	f	g
Low	b	b	c	c	d	d	e	e	f	f	g	g

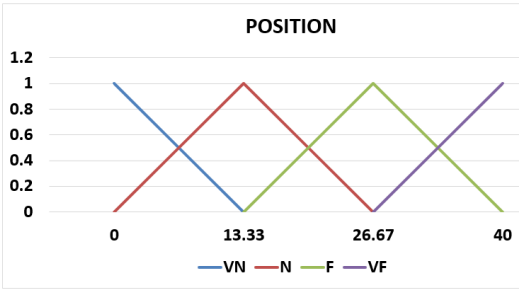
3.2.2 Membership Function

Figure 3 shows plots of membership functions of input and output of FLC 1, with all membership functions triangular in shape.

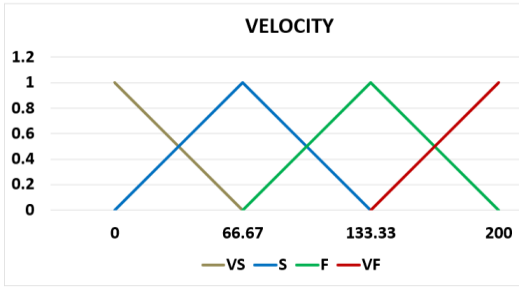
Starting from the equation (7),

$$\frac{y_2-y_1}{x_2-x_1} = \frac{y-y_1}{x-x_1}, \tag{7}$$

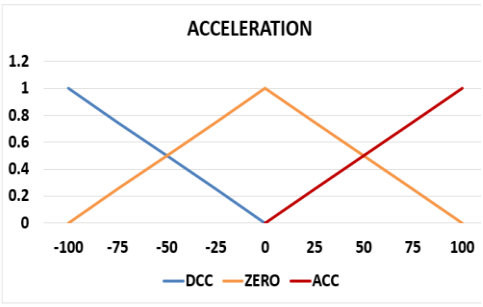
we obtain the equation of lines for all ranges of classes for each membership function.



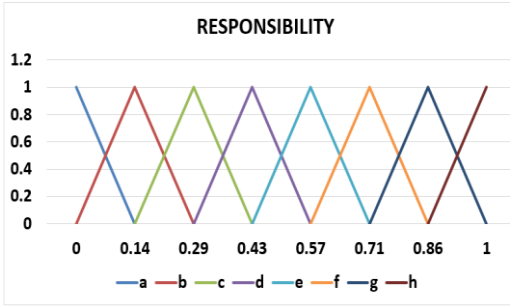
(a) Position membership function, d.



(b) Velocity membership function, v.



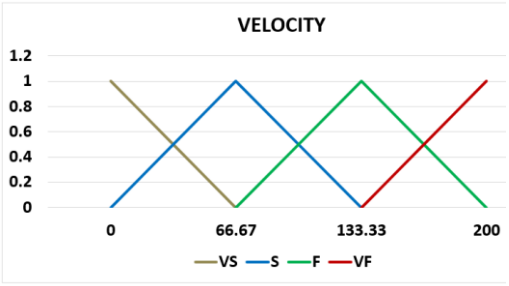
(c) Acceleration membership function, a.



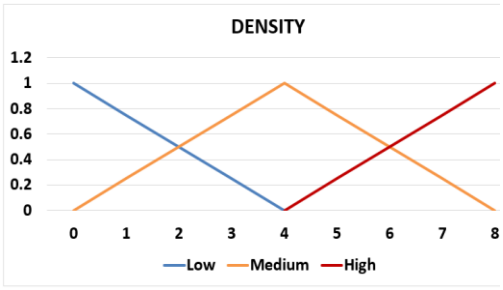
(d) Responsibility membership function, u.

Fig. 3 Membership function of the three inputs and the output of FLC 1.

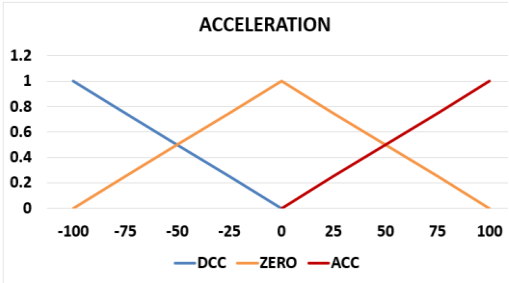
The membership function of the 3 inputs and the output of FLC 2 is represented in Fig. 4. The plots of the membership function are also selected to be triangular similar to FLC 1.



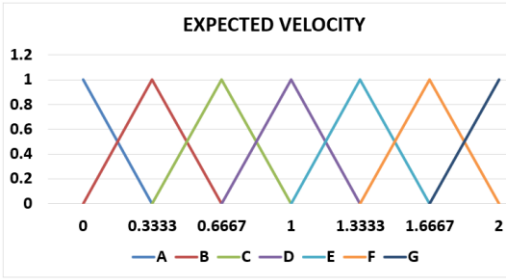
(a) Velocity membership function, v.



(b) Density membership function, ρ.



(c) Acceleration membership function, a.



(d) Expected velocity membership function, v_B^{pref}.

Fig. 4 Membership function of the three inputs and the output of FLC 2.

3.2.3 Inference Method

In this work, the fuzzy Mamdani model is employed. Mamdani-type fuzzy inference is widely used due to its advantage for decision support analysis [73]. It also enables a high level of interpretability which is required for a better understanding of the system [74], especially in our case where it involves the integration of fuzzy with another algorithm to generate collision avoidance protocol. The fuzzy rules in the inference theory are presented in the form of 'IF-THEN' which determines the avoidance strategy of the mobile robot. The mobile robot is required to explore the surroundings to make 2 crucial decisions based on FLC 1 and FLC 2 respectively, which are avoidance responsibility and obstacle anticipated velocity in the next cycle. To achieve this mechanism the inputs of the fuzzy systems have to be determined and monitored.

Thus, for the fuzzy rule base of FLC 1, implementing an 'IF-THEN' rule gives:

$$F^1 = \{IF\ v \text{ is VS and } d \text{ is VF and } a \text{ is DCC, THEN } r \text{ is a }\}$$

$$F^2 = \{IF\ v \text{ is VF and } d \text{ is VN and } a \text{ is ACC, THEN } r \text{ is h }\} \quad (8)$$

For the fuzzy rule base of FLC 2 implementing an 'IF-THEN' rule gives:

$$F^1 = \{IF\ v \text{ is VS and } \rho \text{ is H and } a \text{ is DCC, THEN } \bar{v} \text{ is a }\}$$

$$F^2 = \{IF\ v \text{ is VF and } \rho \text{ is L and } a \text{ is ACC, THEN } \bar{v} \text{ is g }\} \quad (9)$$

Figure 5 shows the fuzzy inference system structure for FLC 1 with 3 input levels, multiple-input range parameters, fuzzy rules and finally the output layer as the end product.

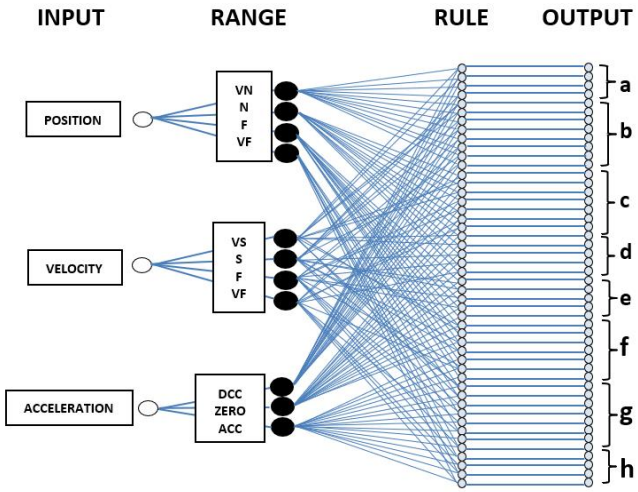
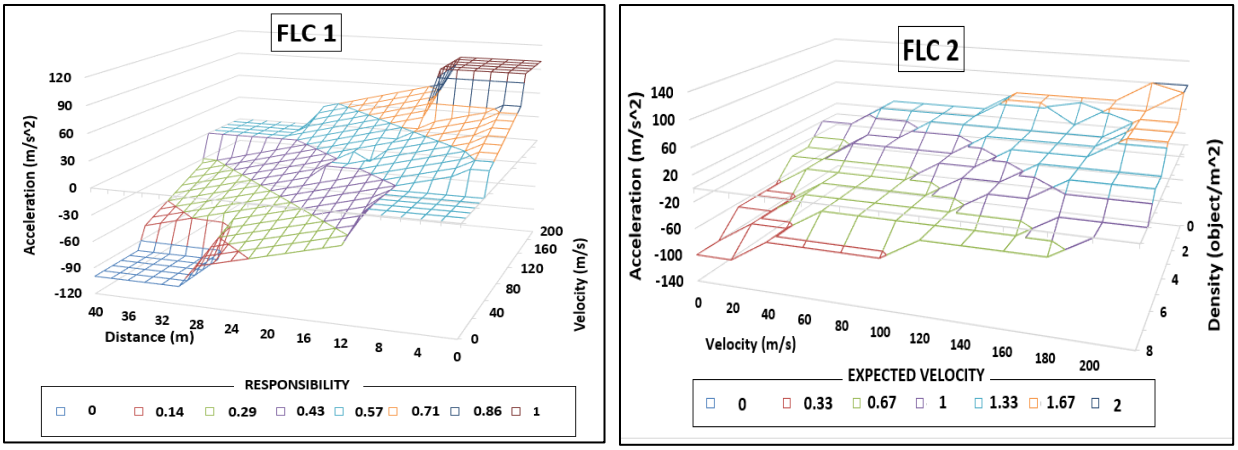


Fig. 5 Fuzzy inference system structure for FLC 1.

Fig. 6 (a) shows the result of the proposed FLC 1. In the plot, both horizontal axes denote the two inputs which are distance d and velocity v, while the vertical axis shows the acceleration a. The output avoidance responsibility u is represented through the coloured wireframe 3D surface. For instance in FLC 1; in a situation where the further distance of the obstacle (d), the slower the velocity (v) and deceleration (a), the mobile robots will incline towards less avoidance responsibility (u). Whereas in a circumstance where the nearer distance of the obstacle, higher velocity (v) and increased acceleration (a), the mobile robot will tend to have higher avoidance responsibility (u).

Fig. 6 (b) shows the plotted result of FLC 2. In the plot, both horizontal axes denote the two inputs which are density ρ and velocity v, while the vertical axis shows the acceleration a. The output expected velocity (v_B^{pref}) is represented through the coloured wireframe 3D surface. In FLC 2 when there is a slower velocity (v), higher density (ρ) and the obstacle are in a state of deceleration (a), the mobile robot will anticipate the slower velocity of the obstacle for the next cycle (v). While on the contrary, when the obstacle's velocity is higher (v), with lower surrounding density (ρ) and increasing acceleration (a), the mobile robot expects the velocity of the obstacle will be multiplied (v).



(a) (b)

Fig. 6 The result of proposed FLC 1 and FLC 2.

3.2.4 Defuzzification

Within the defuzzification stage, the fuzzy numbers are then transformed into crisp values as the final output of FLC 1 and FLC 2. The defuzzification method selected for our fuzzy system is based on the *centroid of sums* as shown in equation (9).

$$\boldsymbol{D} = \frac{\sum \boldsymbol{A_i} \times \boldsymbol{C_i}}{\sum \boldsymbol{A_i}}, \tag{9}$$

as $\boldsymbol{D}$ denotes the centroid of the combined shape, $\boldsymbol{A_i}$ represents the individual areas and $\boldsymbol{C_i}$ as the individual centroids.

4 Results and discussion

As the fundamentals of the proposed method have been projected, the research resumes with the evaluation and experiments of the system. This section is divided into four main subsections. First, subsection 4.1 discuss the experimental design of navigation simulation in detail. Then, the results of said simulation together with its discussion are presented in subsection 4.2. This is followed by subsection 4.3 which displays the construction of a mobile robot involving the hardware experiments and finally, subsection 4.4 presents the results acquired and a final discussion on the comparison of FLC-ORCA with other selected methods of Directional Optimal Reciprocal Collision Avoidance (DORCA) [53] and Improved A* [75].

.

4.1 Navigation Simulation

The test and evaluation of the proposed method involve both software simulation and hardware construction and experiment. Within both procedures, the FLC-ORCA technique is compared with Improved A* and DORCA obstacle avoidance protocol. In software simulation, the test can be done with more variables and with less experimental cost. Within the simulation, conditions can be easily varied and its outcome can be easily investigated in detail. Furthermore, the risk and danger of experiments involving human subjects can be avoided and diminished. Fig. 7 shows the experimental design of the software simulation.

Within the simulation, factorial design [76] is selected with three experimental factors of which multiple levels are tested. Results and data collected across all possible combinations of levels and factors are then collected and studied. The combined data was eventually considered as the final benchmark to measure the performance of each method as a collision-escaping mechanism. Factorial designs possess the advantage of enabling the likelihood of detecting meaningful effects [76]. It also allows the ranking of factors' importance and separation of factors with different levels of significance, providing a lower number of experiments and enabling evaluation of the interaction between varying factors [77], [78]. Additionally, it is also an effective tool for reducing bias in surveys and designs [79].

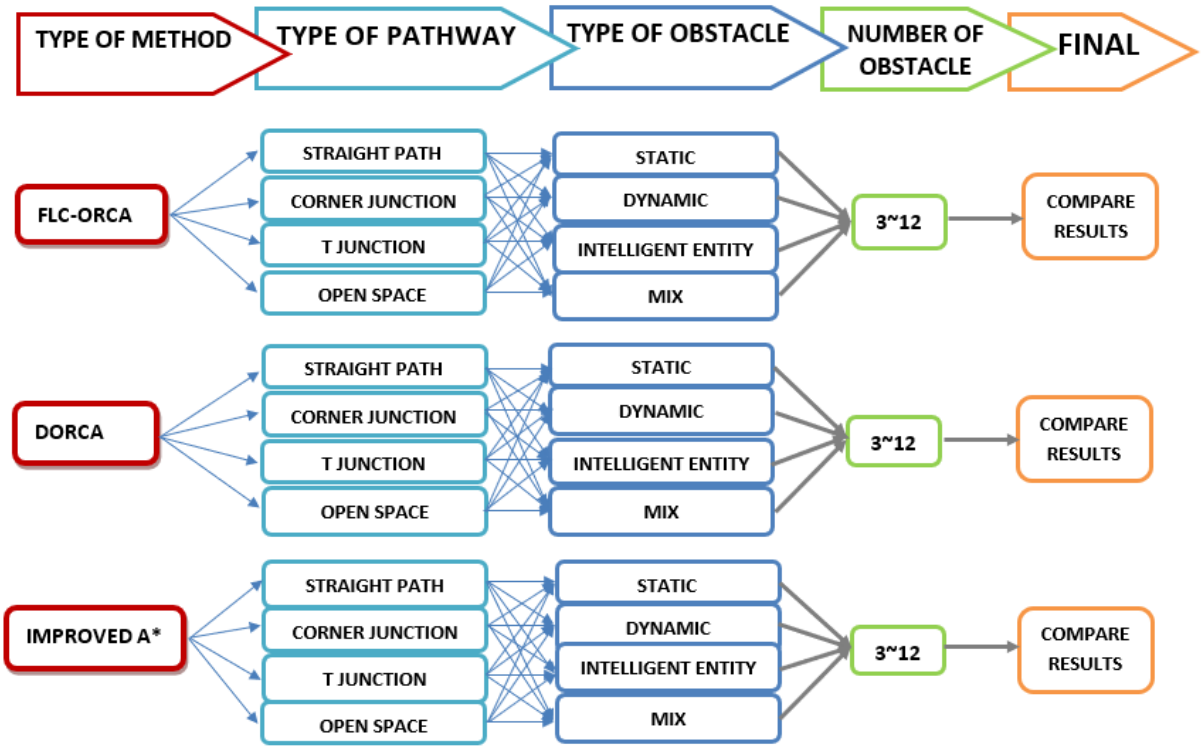
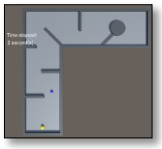


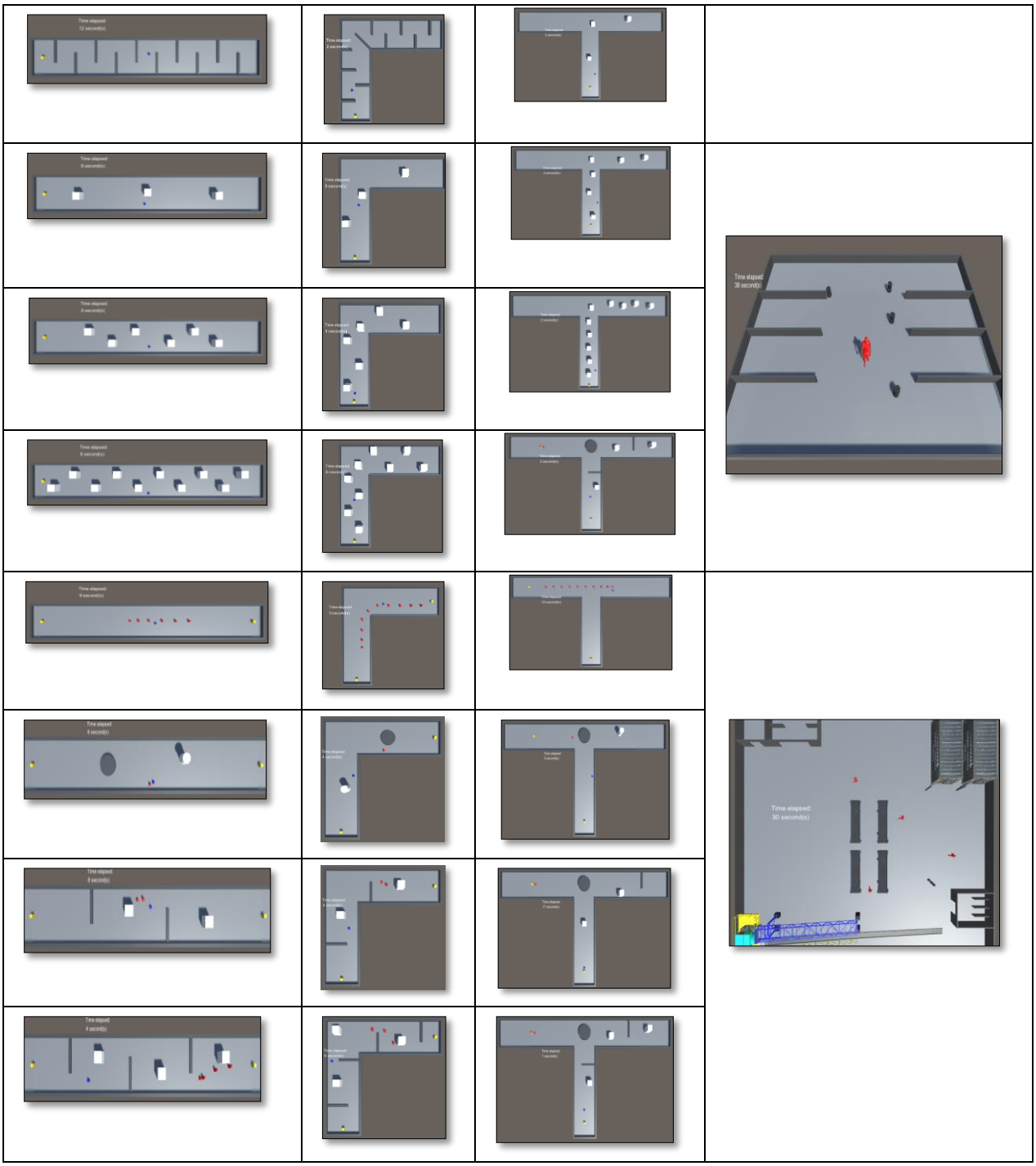
Fig. 7 Experimental design of software simulation.

An initial experiment is conducted through software simulation using Unity version 2017.3.0f3. In this simulation, the main agent, the mobile robot, is put through a scene of a straight line, a corner junction and a T-junction. The obstacles are placed between the starting point and the predetermined destination. There are three types of obstacles viz., static obstacles, dynamic obstacles and an intelligent entity with the ability to avoid collision itself. The main agent is represented as the blue-coloured sphere, and the starting point is marked with a yellow-coloured box with the destination placed at the end of the pathway. The static obstacles are represented in the form of walls, and white boxes act as dynamic obstacles. The red-coloured spheres indicate the intelligent agent.

All of the scenes are tested with the main agent adopting Improved A* [75], DORCA [53] or FLC-ORCA as the method of obstacle avoidance throughout the navigation experiment. The dynamic obstacles are constrained to move in a parallel straight line between the paths. The intelligent entity starts from the destination point of the main agent towards the starting point of the main agent. This is done on purpose so that there will be a meeting point along the pathways. There is no communication between the main agent and the obstacle, and the intelligent entity is set with the Improved A* algorithm as the default method of avoidance. Table 5 shows the navigation simulation through simulations categorically classified as a straight path, corner junction, T-junction and open space simulations. The simulation has been made available here [80].

Table 5 Navigation simulation using UNITY software.

Straight Line	Corner	T-junction	Open Space
			
			



4.2 Simulation Results

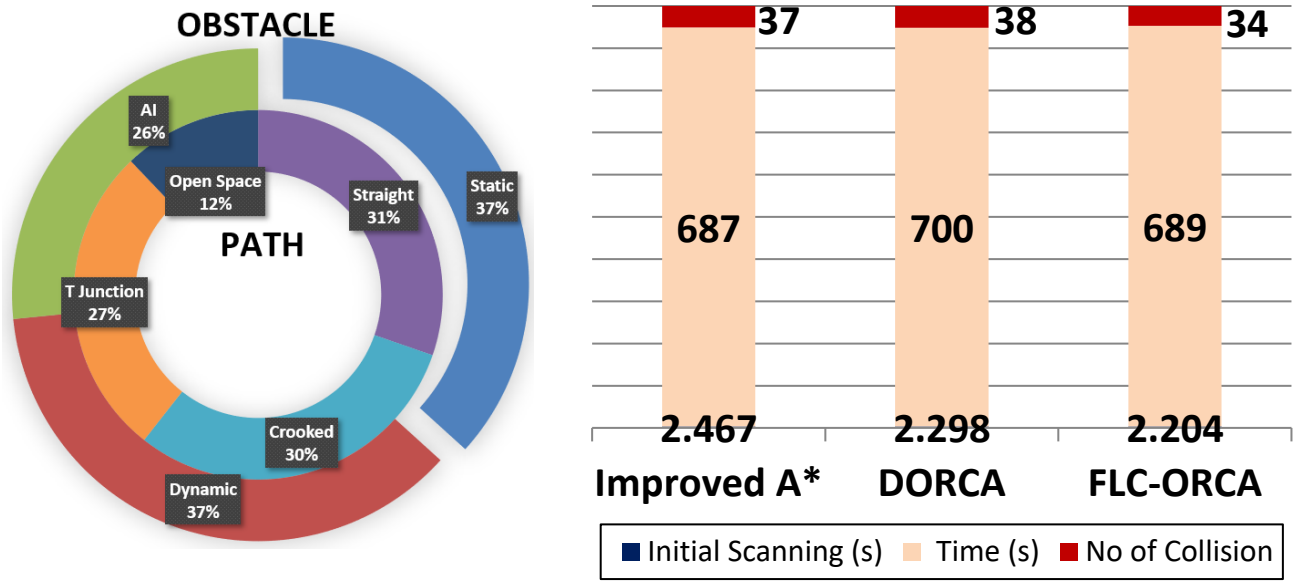
Initially, software simulations are set up to replicate navigation for the mobile robot. Depth analysis is done on the simulation data extensively. Over 3000 repetitive successions of scene simulations are done in total. For each method of obstacle avoidance, 1000 simulations are done to allow depth analysis. Each scene is differentiated based on the type of pathways with the added number of obstacles along the route. Table 7 shows the simulation results for Improved A*, DORCA and FLC-ORCA in terms of time taken for initial scanning (ms), number of collisions and total time (s) for each cycle of navigation.

Table 6 Simulation results for Improved A*, DORCA and FLC-ORCA.

Obstacle Avoidance Method / Pathways		Improved A*			DORCA			FLC-ORCA		
Type of Pathways	**Types of Obstacles**	**Initial Scanning (ms)**	**No of Collision**	**Total Time (s)**	**Initial Scanning (ms)**	**No of Collision**	**Total Time (s)**	**Initial Scanning (ms)**	**No of Collision**	**Total Time (s)**
Straight Road	3 Static	21	0	16	21	0	18	40	0	18
	6 Static	20	0	18	22	0	20	21	0	18
	10 Static	22	0	23	20	0	25	36	0	24
	3 Dynamic	129	2	17	24	1	16	21	1	16
	6 Dynamic	22	2	19	21	4	16	21	4	16
	10 Dynamic	21	2	18	20	7	17	20	5	20
	6 IA	24	6	9	35	0	18	22	0	18
Crooked	3 Static	40	0	14	63	0	16	39	0	15
	6 Static	46	0	16	39	0	17	38	0	16
	10 Static	38	0	17	39	0	18	40	0	18
	3 Dynamic	47	0	14	47	0	13	45	2	12
	6 Dynamic	41	0	15	40	3	14	39	2	13
	9 Dynamic	40	0	20	44	7	14	39	6	13
	10 IA	14	10	12	14	0	15	15	0	15
T Junction	3 Static	60	0	17	101	0	19	59	0	18
	6 Static	60	0	18	69	0	20	59	0	19
	9 Static	60	0	20	60	0	22	58	0	21
	3 Dynamic	57	0	17	60	0	18	57	0	18
	6 Dynamic	69	0	17	59	3	13	59	3	17
	10 Dynamic	59	0	19	61	4	19	59	4	19
	10 AI	21	0	17	20	0	18	20	0	17
Mixture	Crooked/ 1 Static 1 Dynamic/ 1 AI	130	1	12	48	1	13	39	1	12
	Straight / 1 Static 1 Dynamic/ 1 AI	21	1	17	21	0	16	23	0	16
	T-junction/ 1 Static 1 Dynamic/ 1AI	59	1	15	59	1	16	62	1	16
	Crooked/ 2 Static 2 Dynamic/ 2 AI	45	0	14	47	0	13	40	0	13
	Straight/ 2 Static 2 Dynamic/ 2 AI	22	2	18	24	0	17	21	1	16
	T-junction/ 2 Static 2 Dynamic/ 2 AI	60	0	16	60	1	17	78	0	16
	Cooked/ 4 Static 3 Dynamic/ 4 AI	48	4	17	49	1	16	39	0	16
	Straight/ 3 Static 3 Dynamic/ 4 AI	94	0	14	58	1	17	26	0	17
Open Space Simulation	4 AI	76	1	28	73	0	25	80	0	25
	4 AI/ 1 Human	130	4	32	129	4	32	129	4	32
	4 Human/ 4 Container/ 4 Robot	871	1	151	851	0	152	860	0	149
Total		**2467**	**37**	**687**	**2298**	**38**	**700**	**2204**	**34**	**689**

Fig. 8 (a) presents a sunburst chart showing the percentage of obstacles and paths in the simulation. Numerically, static obstacles account for 76 out of the total obstacles (37%), with dynamic obstacles recording another 76 (37%) and the remaining 55 designated by the intelligent entity (26%). In terms of pathways, a straight route makes up for 37% with 10 routes, followed by the corner junction with 10 (30%), the T junction with 9 (27%) and finally open space surrounding with 4 (12%). The pie chart below shows the types of obstacles and categories of pathways included within the simulation. Fig. 8 (b) shows the total time for initial scanning, time of simulation and number of collisions for each method. In total, FLC-ORCA recorded the least number of collisions with 34, followed by Improved A* with 37 and

DORCA with 38. The proposed method FLC-ORCA also achieved decent time for completion with 689s, which is only exceeded by DORCA with 687s and finally Improved A* with 700s time taken. In terms of time taken for initial scanning, the FLC-ORCA also consumes the least amount of consumption with 2.204ms, in comparison to DORCA with 2.298ms and Improved A* with 2.467ms.



(a) Types of pathways and obstacles.

(b) Initial scanning, time of simulation and no of collision for each method.

Fig 8. Simulation pathways and their results

Next, an analysis of the obstacle avoidance method will be done in terms of the number of paths, computational time (ms), searched nodes and path length. For comparison purposes, we will look into a specific scene of open space simulation in Table 9, where there are static, dynamic and intelligent entity obstacles. The path length across the simulation is the longest while also recording the longest time taken to complete the simulation.

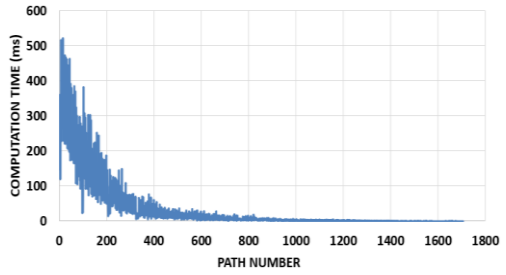
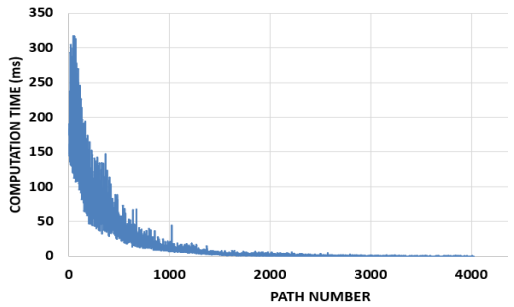
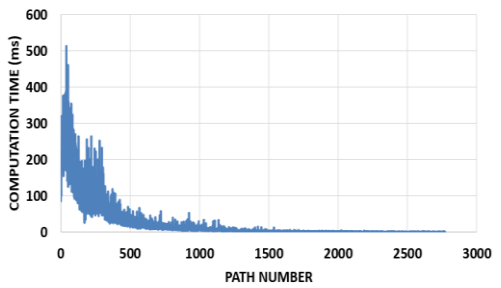
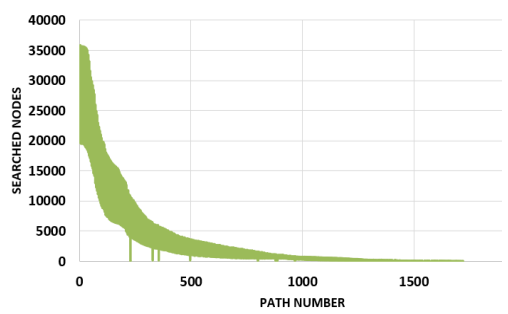
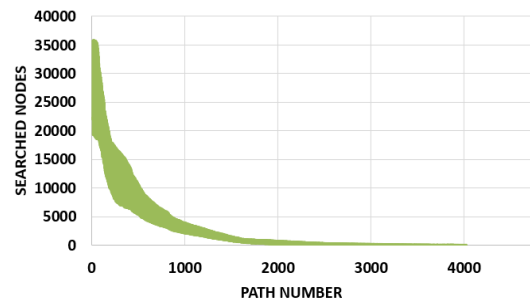
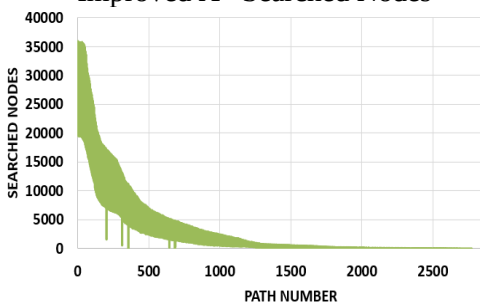
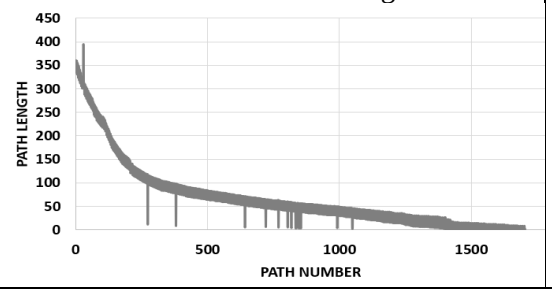
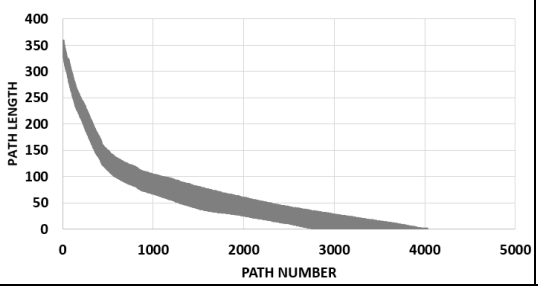
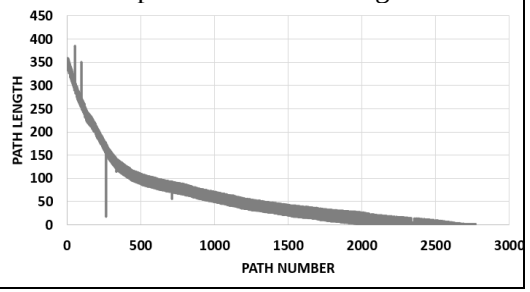
For the number of paths, FLC-ORCA recorded the lowest value with 1706, followed by Improved A* with 2772 and DORCA with 4025. The mean of searched nodes also shows that the proposed method FLC-ORCA recorded the lowest number of searched nodes with 5.01×10^6, followed by Improved A* with 7.8×10^6 and DORCA with 1.0×10^7. For path length, the FLC-ORCA method attained the lowest value with 1.15×10^5, followed by Improved A* with 1.76×10^5 and finally DORCA with 2.53×10^5. Even though FLC-ORCA recorded a higher mean path length, due to having a lower number of paths it is still able to attain the lowest time for completion with 149s, followed by Improved A* with 151s and DORCA with 152s. Table 8 shows the total number of paths, computational time, searched nodes and path length of all obstacle avoidance methods. The mean score, variance and standard deviation of computational time, searched nodes and path length are shown as well.

Table 7 Number of paths, computational time, searched nodes and path length of all obstacle avoidance methods.

Value / Method	Number of Paths	Computational Time (ms)				Searched Nodes				Path Length			
		Total	Mean	Variance (σ^2)	Std Dvt (σ)	Total	Mean	Variance (σ^2)	Std Dvt (σ)	Total	Mean	Variance (σ^2)	Std Dvt (σ)
FLC- ORCA	1706	56379.41	33.0477	5448.324	73.81276	5013979	2939.026377	32419303.64	5693.795188	115300	67.58499414	5277.781	72.64834
DORCA	4025	69795.41	17.3447	1629.88	40.3718	10948099	2720.7	28547430.91	5342.979	253573	62.999	5412.13	73.5672
Improved A*	2772	61293.46	22.10366	2679.008	51.75913	7761913	2799.103	31457825.85	5608.72765	175549	63.32937	5495.745	74.13329

Therefore, based on the performance throughout the whole software simulation, it can be concluded that the proposed method of FLC-ORCA has achieved the best initial time for scanning, the lowest number of collisions and the total time for completion. Based on the specifically selected scene, it can be deduced that FLC-ORCA also performs better than DORCA and Improved A* in terms of searched nodes and computational time. Although FLC-ORCA has a higher value of path length, it possesses the lowest number of paths, therefore deriving the shortest time for scene simulation. Table 8 on the next page shows the computational time, searched nodes and path length for all obstacle avoidance methods. This dataset is made freely available here [81].

Table 8 Computational time, searched nodes and path length along the number of paths for all obstacle avoidance methods.

	FLC-ORCA	DORCA	Improved A*
COMPUTATION TIME	FLC-ORCA Computation Time (ms) 	DORCA Computation Time (ms) 	Improved A* Computation Time (ms) 
SEARCHED NODES	FLC-ORCA Searched Nodes 	DORCA Searched Nodes 	Improved A* Searched Nodes 
PATH LENGTH	FLC-ORCA Path Length 	DORCA Path Length 	Improved A* Path Length 

4.3 Construction of Mobile Robot

In this subsection, the hardware construction involved in the experimental design is discussed thoroughly. In the robotic base model, important components such as the microprocessor and type of sensors used are presented. The capacity of the hardware items is also listed for reference.

TurtleBot3 Burger has been selected as the platform for the mobile robot in this research. On the original TurltleBot3 hardware, a Garmin LiDAR Lite v3 sensor is attached with an additional acrylic base stacked on top of the mobile robot. Raspberry Pi 3 is used as the Single-Board Computer (SBC) to enable high data computing capability and wireless data transmission. Open CR1.0 is implemented as the embedded robot controller and DYNAMIXEL (XL 430) as the motor driver.

As for the software integration, the TurtleBot3 uses the Robot Operating Software (ROS) framework compatible with the C# and Python programming languages used for object recognition and obstacle avoidance. Another crucial improvement with TurtleBot3 is that it enables the implementation of Simultaneous Localization and Mapping (SLAM) which will be used heavily in the real-time navigation experiments with human subjects which will be explained later on in the experiment section. Fig. 9 shows the structure of the prototype with additional sensor systems on Turtlebot3.

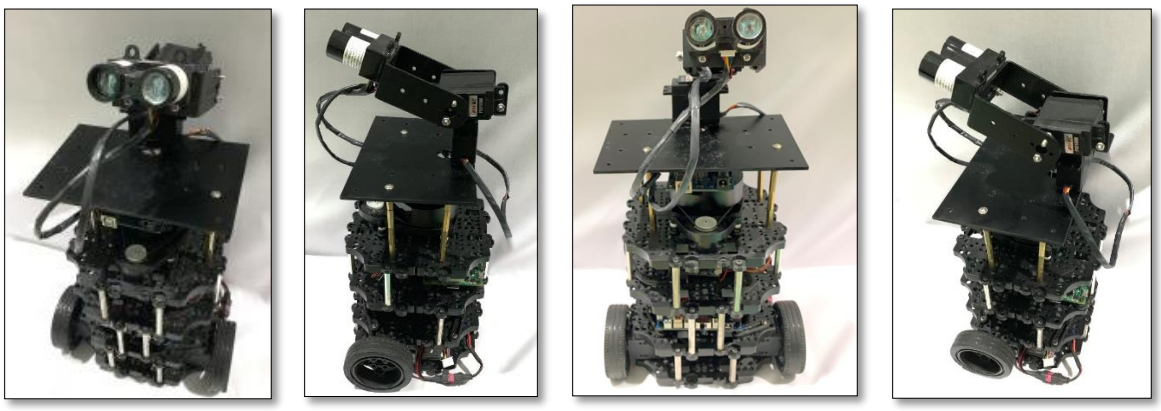


Fig. 9 Prototype structure and configurations.

As the prototype is completely assembled, The LiDAR Lite V3 sensor acts as the sensor for early object detection and recognition with the servo motors attached to it enabling 3D scanning up to 40 meters. The 360 Laser Distance Sensor LDS-01 contributes to the navigation, 2D scanning of the environment and tracking of the surroundings with a scanning range of approximately 3.5 meters, a sampling rate of 1.8 kHz and a scanning rate of 300 ± 10 rpm. The angular range of 360° is most useful for all-around scanning and inspection. As the hardware construction of the mobile robot has been completed, now the formulation of the proposed method of FLC-ORCA is initiated.

4.4 Real-life Navigation Experiment

For final validation, the navigation of a blindfolded subject moving across a living room passing through static, dynamic and intelligent agents is presented here. Similar to the software simulation above, the FLC-ORCA will be compared with both Improved A* and DORCA methods. The venue setting of the experiment is conducted within an indoor environment of a living room with a width and length of $8m \times 10m$. There are 7 human subject participants in total, with both gender, ages ranging from 23 years old to 68 years old. These volunteers are blindfolded throughout the experiment, with navigation solely depending on the mobile robot. They are equipped with a blind stick on one hand, and a string attached to the mobile robot on the other hand, to mimic self-navigation with the assistance of a guide dog. Information about the participants is shown in Table 9.

Table 9 Information on the participants selected for the experiments.

Participant	Age	Height (m)	Weight (kg)	Gender	Eyesight
1	24	1.62	58	F	Normal (short-sightedness)
2	30	1.68	90	M	Normal
3	31	1.56	51	F	Normal (short-sightedness)
4	34	1.60	66	F	Normal
5	44	170	78	M	Normal (short-sightedness)

6	**61**	**162**	**61**	**F**	**Normal (far-sightedness)**
7	**68**	**1.65**	**78**	**M**	**Normal (short-sightedness)**

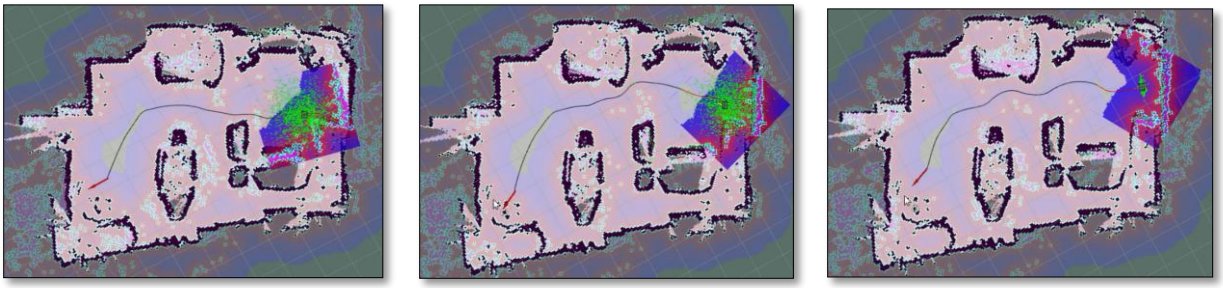
The static obstacles are made of numerous sets of tables and chairs situated in the living rooms. The dynamic object is represented in the form of a TurtleBot3, moving with a predetermined velocity, direction, starting point and destination. And finally, the intelligent agent is characterized by a moving child along the pathway. To analyse the performance of each method, 90 repeated scenes are organized. For each obstacle avoidance method, 30 scenes are allocated with 10 scenes each for static obstacles only, static and dynamic obstacles and ultimately static, dynamic and intelligent agent obstacles.

For initial mapping, the mobile robot moves around the allocated space within the living room on its own, without any person for guidance. This is done through simultaneous localization and mapping (SLAM) for the initialization of the starting point and pre-determined destination. Within the area, there are four sets of furniture, with a total number of 16 chairs, one dining table and two side tables. There are staircases, two enclosed doors and a single open route towards another room. The experiments are done with the starting point situated at one of the doors towards the corner of the room furthest from its destination. Figure 10 shows the surrounding environment of the living room through the view of a camera and SLAM mapping.

(a) Image of the surrounding environment (b) Initial mapping from SLAM

Fig. 10 Image and mapping of experiment environment.

90 repeated simulations are done with the seven participants involved as blindfolded subjects. The experiments are done with the mobile robot implementing the obstacle avoidance of Improved A*, DORCA and FLC-ORCA one by one. The trajectory, time of completion, rerouting occurrences, change of angle, stoppage occurrences and time in stoppages are accumulated. Then, the performance of the proposed FLC-ORCA is validated and compared with the other state-of-the-art obstacle avoidance methods of Improved A* and DORCA. Fig. 11 below shows the trajectory navigation of each method moving towards the predetermined destination.



(a) Improved A* (b) DORCA (c) FLC-ORCA

Fig. 11 The trajectory of navigation for each method.

On the path route, the blindfolded human subject will pass in between three sets of furniture characterized as static obstacles. In the meantime, a moving TurtleBot 3 will approach the trajectory of the navigated blindfolded subject acting as a dynamic obstacle, and eventually a child will come into the scene with the destination set in the opposite direction of the mobile robot.

Fig. 12 shows the navigation trajectory of the mobile robot using FLC-ORCA. The SLAM mapping displays the room's surrounding from the initial scanning. The irregular curving line represents the trajectory of the mobile robot, with the red arrow showcasing its determined destination. From the initial scanning, black-coloured borders are drawn, representing the mapping memorized within the mobile robot. In the mapping, there is a blue box surrounding the mobile robot representing the range of detection for the LiDAR for about a 3.5m radius. The turquoise-coloured dots exhibit the real-time obstacle detected during the navigation. Within sequential SLAM mapping, it can be seen that there are three moving objects with three different colour identifications moving along with the objects. The first is the green-colour base, representing the mobile robot agent itself, followed by the red-colour base representing the dynamic object and finally white-colour base showing the detection of the intelligent entity.

In Fig. 12, the dynamic object with a red-coloured base enters the scene in Fig. 12 (b) until Fig. 12 (e) and the intelligent agent with a white-coloured base can be seen in Fig. 12 (e) until Fig. 12 (g). Within the scene, there is an occurrence of stoppage with the intelligent agent in Fig. 12 (f). The SLAM navigation recording has been made available here [82].

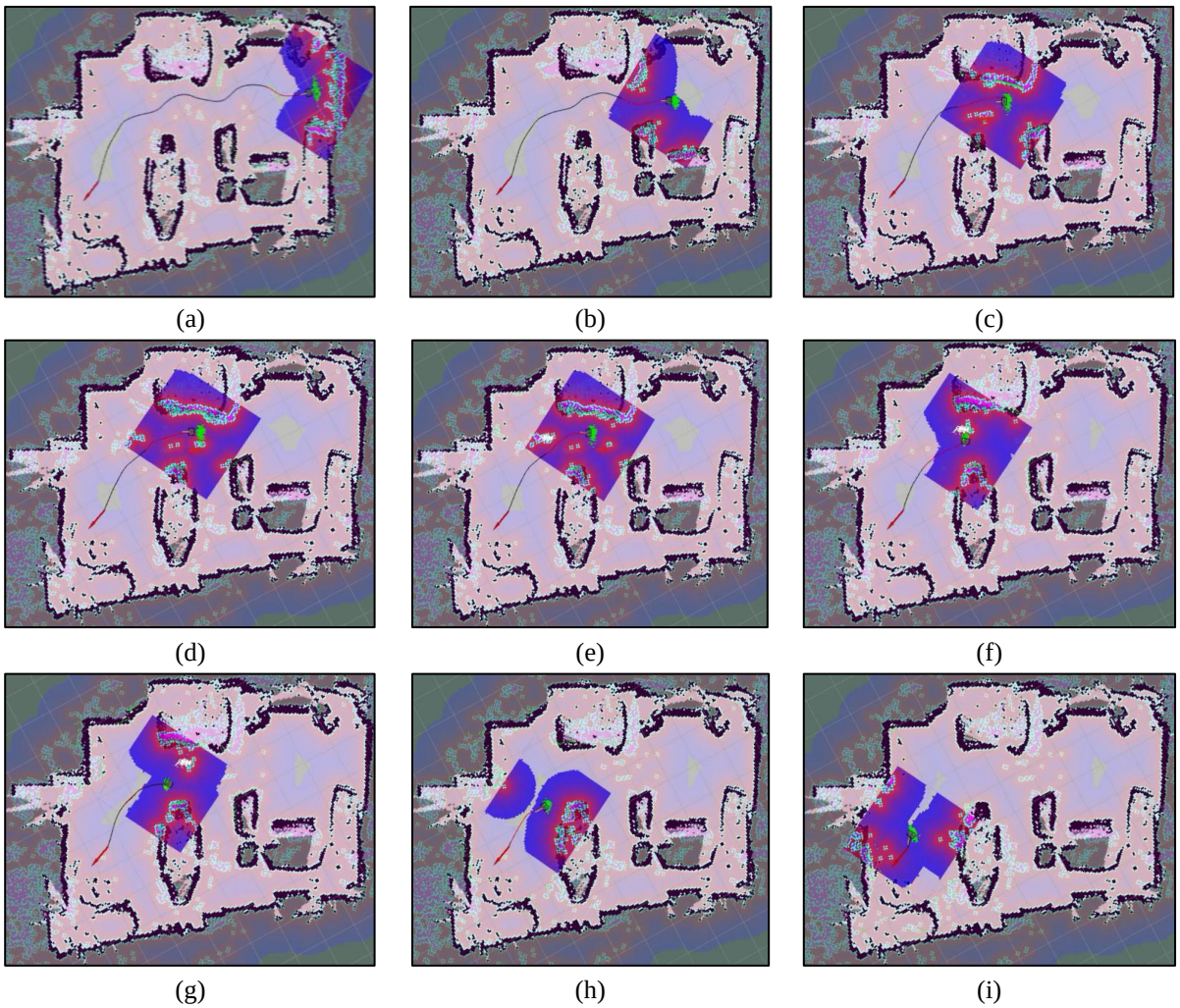


Fig. 12 Robotic navigation motion for FLC-ORCA.

Fig. 13 shows scene-by-scene shot images of navigation by the mobile robot. Fig. 13 (b) until Fig. 13 (e) show the avoidance of a dynamic obstacle in the form of a mobile robot, and Fig. 13 (f) until Fig. 13 (h) indicate obstacle avoidance of an intelligent agent characterised by a small kid walking on the pathway.

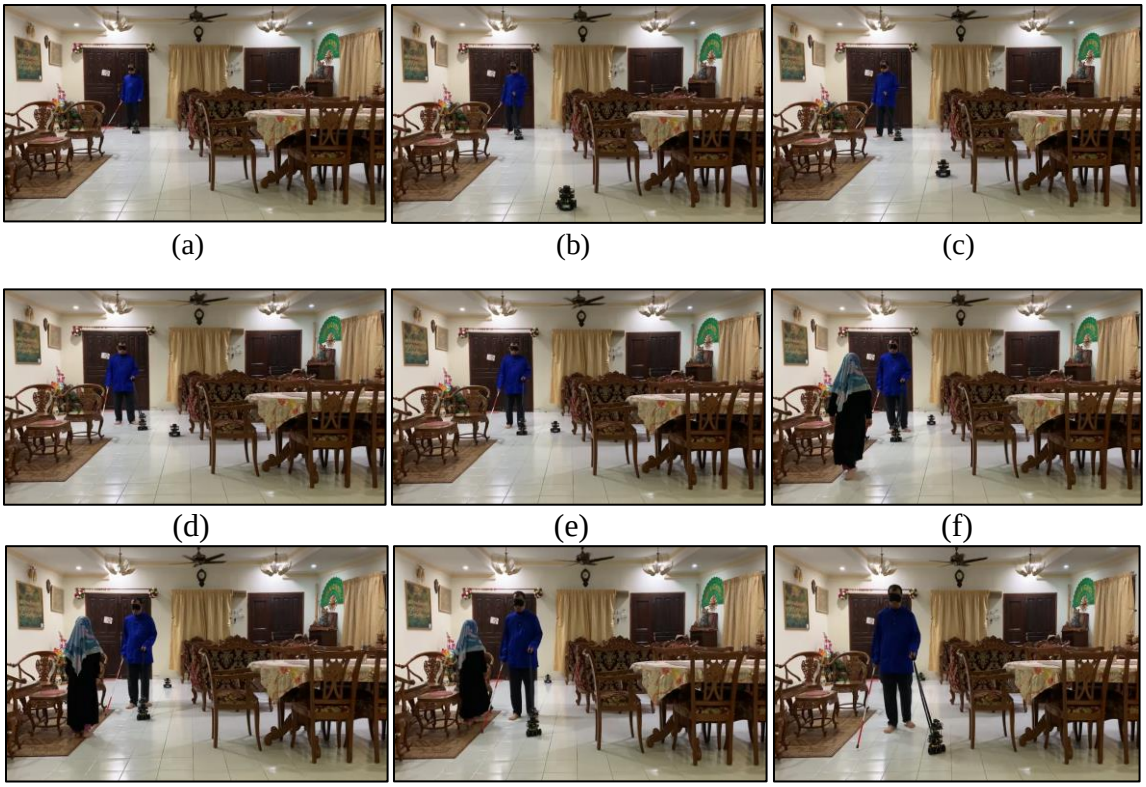


Fig. 13 Navigation experiment with a blindfolded human subject.

Table 10 shows the performance of each obstacle avoidance method. With static obstacles only, FLC-ORCA recorded the shortest time to complete navigation with 1min 19s, followed by DORCA with 1min 35 s and A* with 1min 52s. There is no occurrence of rerouting for Improved A* and FLC-ORCA, whereas there are four times recorded for DORCA. For all rerouting incidents, the accumulated change of angle heading sums up to 65°**.** Concurrently, each method recorded stoppage at least one time for DORCA and FLC-ORCA, and twice for the Improved A* algorithm.

Table 10 Performance of obstacle avoidance methods during navigation.

Method	Obstacle	Total Experiment	Time Taken	Rerouting Occurrence	Angle of Heading Change	Total No of Stoppage	Total Time in Stoppage
Static	Improved A*	10	1min 52s	0	0	2	7.55
	DORCA	10	1min 35s	4	65°	1	4.02
	FLC-ORCA	10	1min 19s	0	0°	1	2.76
Static/ Dynamic	Improved A*	10	1min 24s	1	16°	1	27.39
	DORCA	10	1min 40s	1	34°	1	4.65
	FLC-ORCA	10	1min 14s	2	17°	2	27.2
Static/ Dynamic/ Intelligent Entity	Improved A*	10	1min 43s	2	40°	2	11.9
	DORCA	10	1min 45s	2	14	1	5.11
	FLC-ORCA	10	1min 38s	1	12°	2	9.10

Next, the performance with the added dynamic obstacle is examined. FLC-ORCA performs best in terms of time taken with 1min 14s, followed by A* with 1 min 24 s and eventually DORCA with 1 min 40 s. This time around, there are two incidents of rerouting for FLC-ORCA resulting in a 17° angle change from the initial trajectory. The A* method experienced a single rerouting with an almost similar angle change of 16° and the DORCA method recorded a single rerouting with a 34° change in heading angle.

In the end, an intelligent entity enters the scene adding to the presence of the static and dynamic obstacles.

FLC-ORCA once again excels in the time taken to complete the navigation (1min 38s), with the least rerouting occurrence (1), and least change of angle heading (12°). However, the method does experience two times of stoppages with an accumulated time of 9.10 s. Improved A* comes in second with 1 min 43s, 2 rerouting incidents, 40° angle of diversion and two stoppages. And finally, DORCA recorded 1 min 45s time taken for navigation, 2 times diversion with 14° cumulative angle change and a single stoppage along the route.

Fig. 14 shows the chart of distance travel against the time taken for the navigation with Fig. 14 (a) static obstacle, Fig. 14 (b) static and dynamic obstacle and Fig. 14 (c) static, dynamic and intelligent agent. From here, it can be deduced that the stoppages are only associated with the appearance of moving objects be it dynamic or intelligent entity obstacles. Most obviously for the Improved A* method in Fig. 14 (c), a long period of stoppage can happen due to the inability to arrive at a consensus on which side to go through.

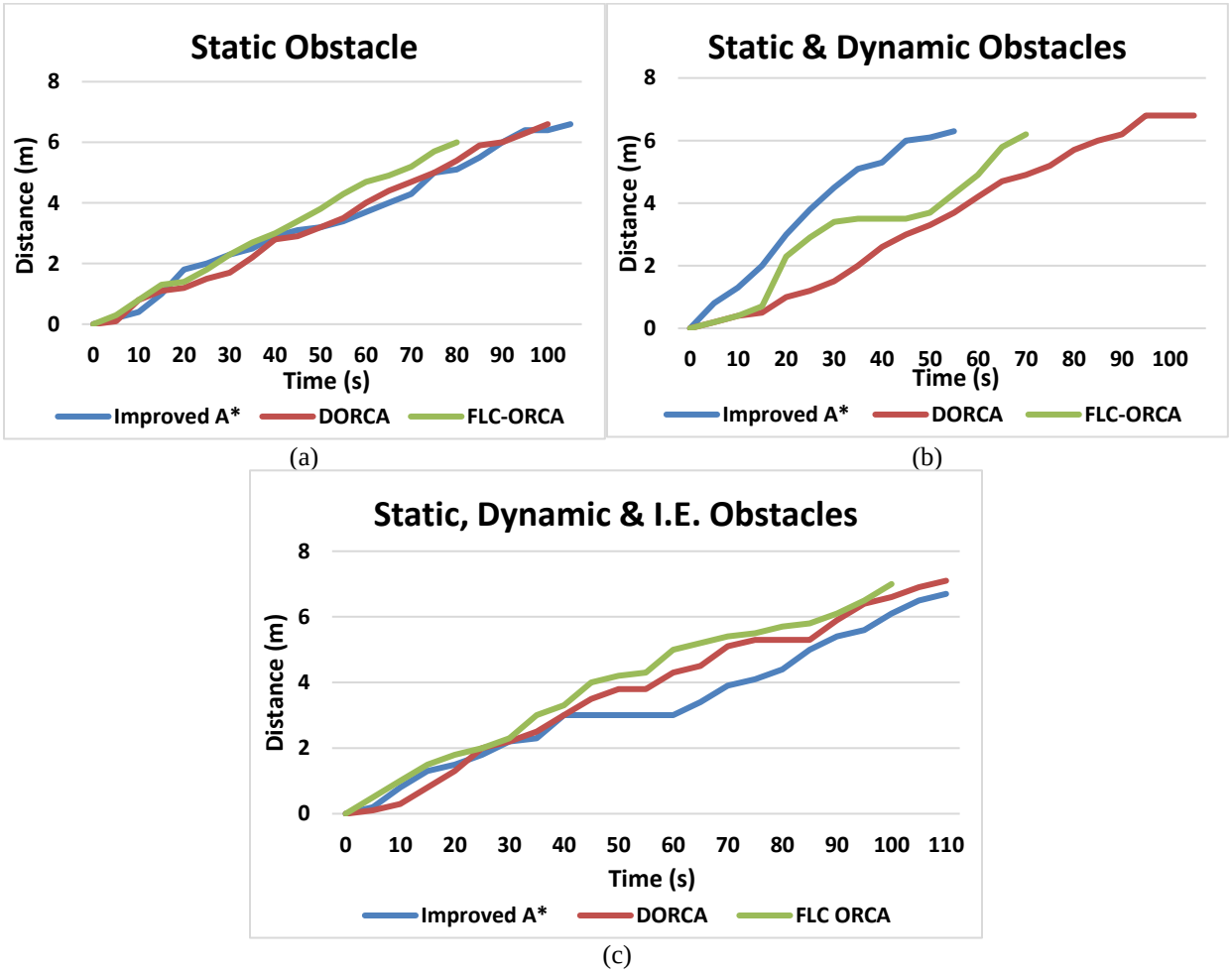


Fig 14. Distance travelled vs time taken for each method and scene during navigation.

5 Conclusion

This research presents the development of an electronic travel aid in the form of a navigating mobile robot for visually impaired people. The study aims to enable robotic navigation within an environment of obstacles under different states of nature. This aim needs to be accomplished while arriving at the destination with the minimum, if not zero number of collisions.

In this research, a method is proposed to solve the conflicting issue of intelligent agents implying dissimilar obstacle avoidance strategies with non-identical obstacle avoidance without central communication with the other agents. Results for two fuzzy models which predict and update obstacle movements based on characteristics of position, velocity, density and acceleration are derived. The experiment is completed with the mobile robot navigating surrounded by obstacles with different types of movement and varying states of cooperation.

The performance of the proposed method using a high-end LiDAR sensor with a high-density point cloud and longer detection range remains to be seen, as it should assist the navigation system to predict collision at earlier stages. Furthermore, obstacle avoidance is not a stand-alone application. Other navigation applications can also benefit from this research as obstacle avoidance is a fundamental of many essential core modules which can seamlessly be integrated into each other, such as wayfinding, localization and position tracking.

Appendix 1: Range and Equation of Membership Function

Table 11 shows the input of position, velocity and acceleration of the FLC 1 with the given output of avoidance responsibility. The range of the obstacle's distance is ($0 < x \leq$ 40m), whereas the range of velocity is ($0 < v \leq$ 200m/s) and acceleration is $(-100m/s^2 < a \leq 100m/s^2)$.

Table 11 Range and equation of each membership function input for FLC 1.

Position		Velocity		Acceleration	
Range	Equation	Range	Equation	Range	Equation
Very Near ($0 < x \leq 13.33$)	$y = \frac{-x}{13.33} + 1$	Very Slow ($0 < x \leq 66.7$)	$y = \frac{-x}{66.7} + 1$	Deceleration ($-100 < x \leq 0$)	$y = {}^{-x}/_{100}$
Near ($0 < x \leq 13.33$)	$y = \frac{x}{13.33}$	Slow ($0 < x \leq 66.7$)	$y = \frac{x}{66.7}$	Zero Acceleration ($-100 < x \leq 0$)	$y = {}^{x}/_{100} + 1$
Near ($13.33 < x \leq 26.67$)	$y = \frac{-x}{13.33} + 2$	Slow ($66.7 < x \leq 133.33$)	$y = \frac{-x}{66.7} + 2$	Zero Acceleration ($0 < x \leq 100$)	$y = {}^{-x}/_{100} + 1$
Far ($13.33 < x \leq 26.67$)	$y = \frac{x}{13.33} - 1$	Fast ($66.7 < x \leq 133.33$)	$y = \frac{x}{66.7} - 1$	Acceleration ($0 < x \leq 100$)	$y = {}^{x}/_{100}$
Far ($26.67 < x \leq 40$)	$y = \frac{-x}{13.33} + 3$	Fast ($133.33 < x \leq 200$)	$y = \frac{-x}{66.7} + 3$		
Very Far ($26.67 < x \leq 40$)	$y = \frac{x}{13.33} - 2$	Very Fast ($133.33 < x \leq 200$)	$y = \frac{x}{66.7} - 2$		

The range and equations of each membership function of FLC 2 are described in Table 12. Within the table, the inputs of FLC 2 are shown to be the velocity, density and acceleration. The range of the velocity is ($0 < v \leq$ 200m/s), density is (0 object/$m^2 < \rho \leq 8$ objects/m^2), and acceleration is $(-100m/s^2 < a \leq 100m/s^2)$.

Table 12 Range and equations of each membership function for the input of FLC 2.

Velocity		Density		Acceleration	
Range	Equation	Range	Equation	Range	Equation
Very Slow ($0 < x \leq 66.7$)	$y = \frac{-x}{66.7} + 1$	Low ($0 < x \leq 4$)	$y = {}^{-1}/_{4}x + 1$	Decelerate ($-100 < x \leq 0$)	$y = {}^{-x}/_{100}$
Slow ($0 < x \leq 66.7$)	$y = \frac{x}{66.7}$	Medium ($0 < x \leq 4$)	$y = {}^{1}/_{4}x$	Zero Acceleration ($-100 < x \leq 0$)	$y = {}^{x}/_{100} + 1$
Slow ($66.7 < x \leq 133.33$)	$y = \frac{-x}{66.7} + 2$	Medium ($4 < x \leq 8$)	$y = {}^{-1}/_{4}x + 2$	Zero Acceleration ($0 < x \leq 100$)	$y = {}^{-x}/_{100} + 1$
Fast ($66.7 < x \leq 133.33$)	$y = \frac{x}{66.7} - 1$	High ($4 < x \leq 8$)	$y = {}^{1}/_{4}x - 1$	Accelerate ($0 < x \leq 100$)	$y = {}^{x}/_{100}$
Fast ($133.33 < x \leq 200$)	$y = \frac{-x}{66.7} + 3$				
Very Fast ($133.33 < x \leq 200$)	$y = \frac{x}{66.7} - 2$				

Appendix 2: Input Variables

Table 13 shows the input variable of FLC 1. All possible combinations of antecedents and the corresponding consequents result in 48 total rules enumerated.

Table 13 Input variables of FLC 1

No	Input Parameter	Ranges	Sign
1	Position	$0 < x \leq 13.33$	Very Near (VN)
2		$13.33 < x \leq 26.67$	Near (N)
3		$26.67 < x \leq 40$	Far (F)

4		x > 40	Very Far (VF)
5		0 < x ≤ 66.7	Very Slow (VS)
6	Velocity	66.7 < x ≤ 133.33	Slow (S)
7		133.33 < x ≤ 200	Fast (F)
8		x > 200	Very Fast (VF)
9		-100 < x ≤ 0	Decelerate (DCC)
10	Acceleration	0 < x ≤ 100	No Acceleration (ZERO)
11		x > 100	Accelerate (ACC)

For the FLC 2, calculating all possible combinations assemble 36 rules in total. Table 14 shows the input variable of FLC 2.

Table 14 Input variables of FLC 2.

No	Input Parameter	Ranges	Sign
1		0 < x ≤ 66.7	Very Slow (VS)
2	Velocity	66.7 < x ≤ 133.33	Slow (S)
3		133.33 < x ≤ 200	Fast (F)
4		x > 200	Very Fast (VF)
5	Density	0 < x ≤ 4	Low
6		4 < x ≤ 8	Medium
7		x > 8	High
8		-100 < x ≤ 0	Decelerate (DCC)
9	Acceleration	0 < x ≤ 100	No Acceleration (ZERO)
10		x > 100	Accelerate (ACC)

Acknowledgement

This research is supported by the IRAGS 2018 Grant: IRAGS18-014-0015 awarded by the International Islamic University Malaysia (IIUM).

Declarations

Conflict of Interest: The authors declare that they have no conflict of interest concerning the publication of this manuscript.

References

1. U. Roijezon, M. Prellwitz, D. I. Ahlmark, J. van Deventer, G. Nikolakopoulos, and K. Hyyppa, "A Haptic Navigation Aid for Individuals with Visual Impairments : Indoor and Outdoor Feasibility Evaluations of the LaserNavigator," *J. Vis. Impair. Blind.*, vol. 113, no. 2, pp. 194–201, 2019, doi: 10.1177/0145482X19842491.
2. E. Cardillo and A. Caddemi, "Insight on Electronic Travel Aids for Visually Impaired People : A Review on the Electromagnetic Technology," 2019.
3. A. Pandey, "Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review," *Int. Robot. Autom. J.*, vol. 2, no. 3, 2017, doi: 10.15406/iratj.2017.02.00023.
4. J. Gai, L. Xiang, and L. Tang, "Using a depth camera for crop row detection and mapping for under-canopy navigation of agricultural robotic vehicle," *Comput. Electron. Agric.*, vol. 188, no. June, p. 106301, 2021, doi: 10.1016/j.compag.2021.106301.
5. K. Groves, E. Hernandez, A. West, T. Wright, and B. Lennox, "Robotic exploration of an unknown nuclear environment using radiation informed autonomous navigation," *Robotics*, vol. 10, no. 2, pp. 1–15, 2021, doi: 10.3390/robotics10020078.
6. E. Soria, F. Schiano, and D. Floreano, "Predictive control of aerial swarms in cluttered environments," *Nat. Mach. Intell.*, vol. 3, no. 6, pp. 545–554, 2021, doi: 10.1038/s42256-021-00341-y.
7. M. R. Mohd Romlay, A. Mohd Ibrahim, S. F. Toha, P. De Wilde, and I. Venkat, "Novel CE-CBCE feature extraction method for object classification using a low-density LiDAR point cloud," *PLoS One*, vol. 16, no. 8, 2021, doi: 10.1371/journal.pone.0256665.
8. M. Afif, R. Ayachi, Y. Said, E. Pissaloux, and M. Atri, "An Evaluation of RetinaNet on Indoor Object Detection for Blind and Visually Impaired Persons Assistance Navigation," *Neural Process. Lett.*, vol. 51, no. 3, pp. 2265–2279, 2020, doi: 10.1007/s11063-020-10197-9.
9. H. Zhang, L. Jin, and C. Ye, "An RGB-D camera based visual positioning system for assistive navigation by a robotic navigation aid," *IEEE/CAA J. Autom. Sin.*, vol. 8, no. 8, pp. 1389–1400, 2021, doi: 10.1109/JAS.2021.1004084.
10. A. N. Angelopoulos, H. Ameri, D. Mitra, and M. Humayun, "Enhanced Depth Navigation Through Augmented Reality Depth Mapping in Patients with Low Vision," *Sci. Rep.*, vol. 9, no. 1, p. 11230, 2019, doi: 10.1038/s41598-019-47397-w.
11. M. R. M. Romlay, S. F. Toha, A. M. Ibrahim, and I. Venkat, "Methodologies and evaluation of electronic travel aids for the visually impaired people: A review," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, pp. 1747–1758, 2021, doi: 10.11591/eei.v10i3.3055.
12. M. N. A. Wahab, C. M. Lee, M. F. Akbar, and F. H. Hassan, "Path Planning for Mobile Robot Navigation in Unknown Indoor Environments Using Hybrid PSOFS Algorithm," *IEEE Access*, vol. 8, pp. 161805–161815, 2020, doi: 10.1109/ACCESS.2020.3021605.
13. X. Zhang, X. Yao, Y. Zhu, and F. Hu, "An ARCore Based User Centric Assistive Navigation System for Visually Impaired People," *Appl. Sci.*, vol. 9, no. 5, 2019, doi: 10.3390/app9050989.
14. E. Meyer, H. Robinson, A. Rasheed, and O. San, "Taming an Autonomous Surface Vehicle for Path following and Collision Avoidance Using Deep Reinforcement Learning," *IEEE Access*, vol. 8, pp. 41466–41481, 2020, doi: 10.1109/ACCESS.2020.2976586.
15. S. Martinez-Cruz, L. A. Morales-Hernandez, G. I. Perez-Soto, J. P. Benitez-Rangel, and K. A. Camarillo-Gomez, "An Outdoor Navigation Assistance System for Visually Impaired People in Public Transportation," *IEEE Access*, 2021, doi: 10.1109/access.2021.3111544.
16. Z. Pourtousi *et al.*, "Ability of neural network cells in learning teacher motivation scale and prediction of

motivation with fuzzy logic system," *Sci. Rep.*, vol. 11, no. 1, pp. 1–17, 2021, doi: 10.1038/s41598-021-89005-w.
17. W. Ren, G. S. Member, O. U. Ma, and H. Ji, "Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches," *IEEE Access*, vol. 8, pp. 135628–135639, 2020, doi: 10.1109/ACCESS.2020.3011697.
18. H. E, Y. Cui, W. Pedrycz, and Z. Li, "Enhancements of rule-based models through refinements of Fuzzy C-Means," *Knowledge-Based Syst.*, vol. 170, pp. 43–60, 2019, doi: 10.1016/j.knosys.2019.01.027.
19. M. Babanezhad, S. Zabihi, I. Behroyan, A. T. Nakhjiri, A. Marjani, and S. Shirazian, "Prediction of gas velocity in two-phase flow using developed fuzzy logic system with differential evolution algorithm," *Sci. Rep.*, vol. 11, no. 1, pp. 1–14, 2021, doi: 10.1038/s41598-021-81957-3.
20. M. R. M. Romlay, M. I. Azhar, and S. F. Toha, "Two-wheel Balancing Robot ; Review on Control Methods and Experiment," vol. 5, 2017.
21. K. V. Shihabudheen and G. N. Pillai, "Recent advances in neuro-fuzzy system: A survey," *Knowledge-Based Syst.*, vol. 152, pp. 136–162, 2018, doi: 10.1016/j.knosys.2018.04.014.
22. B. Kasmi and A. Hassam, "Comparative Study between Fuzzy Logic and Interval Type-2 Fuzzy Logic Controllers for the Trajectory Planning of a Mobile Robot," *Eng. Technol. Appl. Sci. Res.*, vol. 11, no. 2, pp. 7011–7017, 2021, doi: 10.48084/etasr.4031.
23. C. Zong, Z. Ji, Y. Yu, and H. Shi, "Research on Obstacle Avoidance Method for Mobile Robot Based on Multisensor Information Fusion," *Sensors Mater.*, vol. 32, no. 4, pp. 1159–1170, 2020.
24. Z. Sui, Z. Pu, J. Yi, and S. Wu, "Formation Control with Collision Avoidance through Deep Reinforcement Learning Using Model-Guided Demonstration," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 6, pp. 2358–2372, 2021, doi: 10.1109/TNNLS.2020.3004893.
25. K. Khnissi, C. Ben Jabeur, and H. Seddik, "A Smart Mobile Robot Commands Predictor using Recursive Neural Network," *Rob. Auton. Syst.*, vol. 131, no. September, 2020, doi: 10.1016/j.robot.2020.103593.
26. A. Bouguettaya and H. Zarzour, "Deep learning techniques to classify agricultural crops through UAV imagery : a review," *Neural Comput. Appl.*, vol. 9, 2022, doi: 10.1007/s00521-022-07104-9.
27. K. Stergiou and T. E. Karakasidis, "Application of deep learning and chaos theory for load forecasting in Greece," *Neural Comput. Appl.*, vol. 33, no. 23, pp. 16713–16731, 2021, doi: 10.1007/s00521-021-06266-2.
28. Z. Lin, M. Yue, G. Chen, and J. Sun, "Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles," *Trans. Inst. Meas. Control*, no. July, 2021, doi: 10.1177/01423312211024798.
29. H. Citakoglu, "Comparison of artificial intelligence techniques via empirical equations for prediction of solar radiation," *Comput. Electron. Agric.*, vol. 118, pp. 28–37, 2015, doi: 10.1016/j.compag.2015.08.020.
30. H. Citakoglu, "Comparison of artificial intelligence techniques for prediction of soil temperatures in Turkey," *Theor. Appl. Climatol.*, vol. 130, no. 1–2, pp. 545–556, 2017, doi: 10.1007/s00704-016-1914-7.
31. M. Cobaner, H. Citakoglu, O. Kisi, and T. Haktanir, "Estimation of mean monthly air temperatures in Turkey," *Comput. Electron. Agric.*, vol. 109, pp. 71–79, 2014, doi: 10.1016/j.compag.2014.09.007.
32. H. Citakoglu, M. Cobaner, T. Haktanir, and O. Kisi, "Estimation of Monthly Mean Reference Evapotranspiration in Turkey," *Water Resour. Manag.*, vol. 28, no. 1, pp. 99–113, 2014, doi: 10.1007/s11269-013-0474-1.
33. S. Shentu, F. Xie, X. Liu, and Z. Gong, "Motion Control and Trajectory Planning for Obstacle Avoidance of the Mobile Parallel Robot Driven by Three Tracked Vehicles," *Robotica*, vol. 39, no. 6, pp. 1037–1050, 2020, doi: 10.1017/S0263574720000880.
34. F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 3, pp. 1–15, 2020, doi: 10.1177/1729881420929498.
35. A. Mortazavi and M. Moloodpoor, "Enhanced Butterfly Optimization Algorithm with a New fuzzy Regulator Strategy and Virtual Butterfly Concept," *Knowledge-Based Syst.*, vol. 228, p. 107291, 2021, doi: 10.1016/j.knosys.2021.107291.
36. P. Rawat and S. Chauhan, "Particle swarm optimization-based energy efficient clustering protocol in wireless sensor network," *Neural Comput. Appl.*, vol. 7, 2021, doi: 10.1007/s00521-021-06059-7.
37. J. Snape, S. Member, S. J. Guy, and D. Manocha, "The Hybrid Reciprocal Velocity Obstacle," *IEEE Trans. Robot.*, vol. 27599, no. August, pp. 696–706, 2011.
38. J. Liang, U. Patel, A. J. Sathyamoorthy, and D. Manocha, "Crowd-Steer: Realtime Smooth and Collision-Free Robot Navigation in Densely Crowded Scenarios Trained using High-Fidelity Simulation," in *IJCAI International Joint Conference on Artificial Intelligence*, 2020, pp. 4221–4228, doi: 10.24963/ijcai.2020/583.
39. S. Yao, G. Chen, Q. Qiu, J. Ma, X. Chen, and J. Ji, "Crowd-Aware Robot Navigation for Pedestrians with Multiple Collision Avoidance Strategies via Map-based Deep Reinforcement Learning," *arXiv Prepr.*, 2021, [Online]. Available: https://github.com/snape/RVO2.
40. N. P. M. Murugan, "Natural Disaster Resilience Approach (NDRA) to Online Social Networks," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, 2020, doi: 10.1007/s12652-020-02644-1.
41. B. Kleinmeier, "Modeling of Behavioral Changes in Agent-Based Simulations psychological processes," 2021.
42. J. Van Den Berg, M. Lin, and D. Manocha,

"Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation," *IEEE Internatioal Conf. Robot. Autom.*, pp. 1928–1935, 2008.
43. P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *Int. J. Robot. Res.*, no. July, 1998, doi: 10.1177/027836499801700706.
44. J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n -Body Collision Avoidance," in *Springer Tracts in Advanced Robotic*, 2011, pp. 3–19.
45. J. Alonso-mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal Collision Avoidance for Multiple Car-like Robots," *IEEE Int. Conf. Robot. Autom.*, pp. 360–366, 2012.
46. A. Levy, C. Keitel, S. Engel, and J. Mclurkin, "The Extended Velocity Obstacle and Applying ORCA in the Real World *," *IEEE Int. Conf. Robot. Autom.*, pp. 16–22, 2015.
47. J. Godoy, S. J. Guy, M. Gini, and I. Karamouzas, "C-Nav: Distributed coordination in crowded multi-agent navigation," *Rob. Auton. Syst.*, vol. 133, p. 103631, 2020, doi: 10.1016/j.robot.2020.103631.
48. H. Cheng, Q. Zhu, Z. Liu, T. Xu, and L. Lin, "Decentralized Navigation of Multiple Agents Based on ORCA and Model Predictive Control *," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 3446–3451, 2017.
49. X. Zhong, X. Zhong, and X. Peng, "Velocity-Change-Space-based dynamic motion planning for mobile robots navigation," *Neurocomputing*, vol. 143, pp. 153–163, 2014, doi: 10.1016/j.neucom.2014.06.010.
50. M. Choi, A. Rubenecia, T. Shon, and H. H. Choi, "Velocity obstacle based 3D collision avoidance scheme for low-cost micro UAVs," *Sustainability*, vol. 9, no. 1174, 2017, doi: 10.3390/su9071174.
51. Y. Huang, P. H. A. J. M. van Gelder, and Y. Wen, "Velocity obstacle algorithms for collision prevention at sea," *Ocean Eng.*, vol. 151, no. September, pp. 308–321, 2018, doi: 10.1016/j.oceaneng.2018.01.001.
52. T. Lisini Baldi, S. Scheggi, M. Aggravi, and D. Prattichizzo, "Haptic Guidance in Dynamic Environments Using Optimal Reciprocal Collision Avoidance," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 265–272, 2018, doi: 10.1109/LRA.2017.2738328.
53. H. Niu, C. Ma, and P. Han, "Directional optimal reciprocal collision avoidance," *Rob. Auton. Syst.*, vol. 136, p. 103705, 2021, doi: 10.1016/j.robot.2020.103705.
54. K. Guo, D. Wang, T. Fan, and J. Pan, "VR-ORCA: Variable Responsibility Optimal Reciprocal Collision Avoidance," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4520–4527, 2021, doi: 10.1109/LRA.2021.3067851.
55. S. H. Arul and D. Manocha, "V-RVO : Decentralized Multi-Agent Collision Avoidance using Voronoi," 2021.
56. J. K. Janardanan, "Decentralized Collision Avoidance," *Comput. Sci. Eng. Thesis, Diss. Student Res.*, vol. 61, 2013.
57. R. Mao, H. Gao, and L. Guo, "A Novel Collision-Free Navigation Approach for Multiple Nonholonomic Robots Based on ORCA and Linear MPC," *Math. Probl. Eng.*, 2020, doi: 10.1155/2020/4183427.
58. J. Alonso-mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots," in *Distributed Autonomous Robotic Systems*, 2013, pp. 203–216.
59. L. He and J. van den Berg, "Meso-Scale Planning for Multi-Agent Navigation," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2839–2844.
60. D. Bareiss and J. Van Den Berg, "Generalized reciprocal collision avoidance," *Int. J. Rob. Res.*, vol. 34, no. 12, pp. 1501–1514, 2015, doi: 10.1177/0278364915576234.
61. J. Snape and D. Manocha, "Navigating multiple simple-airplanes in 3D workspace," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 3974–3980, doi: 10.1109/ROBOT.2010.5509580.
62. J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3475–3482, doi: 10.1109/ICRA.2011.5980408.
63. Y. Wang and A. Cavallaro, "Active visual tracking in multi-agent scenarios," in *14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, no. August, pp. 1–6.
64. A. Pandey, V. S. Panwar, M. Ehtesham Hasan, and D. R. Parhi, "V-REP-based navigation of automated wheeled robot between obstacles using PSO-tuned feedforward neural network," *J. Comput. Des. Eng.*, vol. 7, no. 4, pp. 427–434, 2020, doi: 10.1093/jcde/qwaa035.
65. M. Nadour, M. Boumehraz, L. Cherroun, and V. Puig, "Mobile robot visual navigation based on fuzzy logic and optical flow approaches," *Int. J. Syst. Assur. Eng. Manag.*, vol. 10, no. 6, pp. 1654–1667, 2019, doi: 10.1007/s13198-019-00918-2.
66. A. Aouf, L. Boussaid, and A. Sakly, "Same Fuzzy Logic Controller for Two-Wheeled Mobile Robot Navigation in Strange Environments," *J. Robot.*, vol. 1–11, 2019, doi: 10.1155/2019/2465219.
67. N. M. Nakrani and M. M. Joshi, "A human-like decision intelligence for obstacle avoidance in autonomous vehicle parking," *Appl. Intell.*, 2021, doi: 10.1007/s10489-021-02653-3.
68. G. Chen *et al.*, "Deep Reinforcement Learning of Map - Based Obstacle Avoidance for Mobile Robot Navigation," *SN Comput. Sci.*, vol. 2, no. 417, 2021, doi: 10.1007/s42979-021-00817-z.
69. K. K. Pandey and D. R. Parhi, "Trajectory Planning and the Target Search by the Mobile Robot in an Environment Using a Behavior-Based Neural Network Approach," *Robotica*, vol. 38, no. 9, pp. 1627–1641, 2019, doi: 10.1017/S0263574719001668.
70. H. Song, A. Li, T. Wang, and M. Wang, "Multimodal Deep Reinforcement Learning with Auxiliary Task," *Sensors*, vol. 21, no. 1363, 2021.

71. H. Wang, Z. Fu, J. Zhou, M. Fu, and L. Ruan, “Cooperative collision avoidance for unmanned surface vehicles based on improved genetic algorithm,” *Ocean Eng.*, vol. 222, no. 108612, 2021, doi: 10.1016/j.oceaneng.2021.108612.
72. A. Lopez-Gonzalez, J. A. M. Campaña, E. G. H. Martínez, and P. P. Contro, “Multi robot distance based formation using Parallel Genetic Algorithm,” *Appl. Soft Comput. J.*, vol. 86, no. 105929, 2019, doi: 10.1016/j.asoc.2019.105929.
73. W. E. Sari, O. Wahyunggoro, and S. Fauziati, “A Comparative Study on Fuzzy Mamdani-Sugeno-Tsukamoto for the Childhood Tuberculosis Diagnosis,” in *AIP Conference Proceeding*, 2016, vol. 1755, doi: 10.1063/1.4958498.
74. S. C. V. A. Julio Barón Velandia, Jonathan Steven Capera Quintana, “Environment humidity and temperature prediction in agriculture using Mamdani inference systems agriculture using Mamdani inference systems,” *Int. ournalof Electr. andComputer Eng.*, vol. 11, no. 4, pp. 3502–3509, 2021, doi: 10.11591/ijece.v11i4.pp3502-3509.
75. S. Erke, D. Bin, N. Yiming, Z. Qi, X. Liang, and Z. Dawei, “An improved A-Star based path planning algorithm for autonomous land vehicles,” *Int. J. Adv. Robot. Syst.*, vol. 17, no. 5, pp. 1–13, 2020, doi: 10.1177/1729881420962263.
76. C. Tannenbaum, R. P. Ellis, F. Eyssel, J. Zou, and L. Schiebinger, “Sex and gender analysis improves science and engineering,” *Nature*, vol. 575, no. January, pp. 137–146, 2019, doi: 10.1038/s41586-019-1657-6.
77. M. A. Omar, H. M. Ahmed, H. A. Batakoushy, and M. A. Abdel, “Spectrochimica Acta Part A : Molecular and Biomolecular Spectroscopy New spectro fl uorimetric analysis of empagli fl ozin in its tablets and human plasma using two level full factorial design,” *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.*, vol. 235, p. 118307, 2020, doi: 10.1016/j.saa.2020.118307.
78. M. Mihăilescu *et al.*, “Full factorial design for gold recovery from industrial solutions,” *Toxics*, vol. 9, no. 5, pp. 1–17, 2021, doi: 10.3390/toxics9050111.
79. S. Walzenbach, “Hiding Sensitive Topics by Design? An Experiment on the Reduction of Social Desirability Bias in Factorial Surveys,” *Surv. Res. Methods*, vol. 13, no. 1, pp. 103–121, 2019.
80. M. R. M. Romlay, A. M. Ibrahim, S. F. Toha, and M. S. Ahmad, “UNITY Simulation for Navigation using FLC-ORCA, Improved A-Star & Directional ORCA,” *IEEE Dataport*, 2022. .
81. M. R. M. Romlay, A. M. Ibrahim, S. F. Toha, and M. S. Ahmad, “Computation Time , Searched Nodes and Path Length for Navigation Using Improved A-Star, Directional ORCA and FLC-ORCA,” *Zenodo*, 2021, doi: 10.5281/zenodo.5786618.
82. M. R. M. Romlay, A. M. Ibrahim, S. F. Toha, and M. S. Ahmad, “SLAM Recording for Navigation using FLC-ORCA, Improved A Star and Directional ORCA,” *IEEE Dataport*, 2022.