

Article

# Self-Directed Mobile Robot Navigation Based on Functional Firefly Algorithm (FFA)

Bhmeshwar K. Patle <sup>1</sup>, Brijesh Patel <sup>2,\*</sup>, Alok Jha <sup>3</sup> and Sunil Kumar Kashyap <sup>4</sup>

<sup>1</sup> Department of Mechanical Engineering, MIT Art, Design and Technology University, Pune 412201, India; balu\_patle@rediffmail.com

<sup>2</sup> Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

<sup>3</sup> Department of Mechanical Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur 440013, India; alokjha.30@gmail.com

<sup>4</sup> Department of Mathematics, MATS University, Raipur 492001, India; drsunilkk@matsuniversity.ac.in

\* Correspondence: aero.brijesh@gmail.com

**Abstract:** This paper proposes an optimized mobile robot navigation strategy using a functional firefly algorithm (FFA) and choice function. This approach has two key advantages: first, the linear objective function performs efficiently with the single degree and finite-order polynomial time operation, and second, the cartesian constraint performs compactly with the chosen degree of freedom on the finite interval. This functional approach optimizes the size of operational parameters in context with key size, operation time, and a finite range of verification. The choice function achieves parameter order (size) reduction. The attraction characteristic of fireflies is represented by the choice function for optimizing the choice between low and high intensities of fireflies. In 2D and 3D environments, the proposed robot navigation performs well in an uncertain environment with static and dynamic obstacles. This efficiency includes the robot's speed as determined by the choice function's minimum path lengths. The collision-free path is achieved by the non-void family of non-void sets. The obtained results are optimal in terms of path length and navigational time. The proposed controller is also compared with the other existing controllers, and it is observed that the FFA gives the shortest path in less time for the same environmental condition.



**Citation:** Patle, B.K.; Patel, B.; Jha, A.; Kashyap, S.K. Self-Directed Mobile Robot Navigation Based on Functional Firefly Algorithm (FFA). *Eng* **2023**, *4*, 2656–2681. <https://doi.org/10.3390/eng4040152>

Academic Editor: Antonio Gil Bravo

Received: 9 September 2023

Revised: 15 October 2023

Accepted: 16 October 2023

Published: 18 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** mobile robot navigation; firefly algorithm; choice function; path planning; obstacle avoidance

## 1. Introduction

With the growing demand for autonomous systems in household work, industry, entertainment, medical care, transportation, and especially national security, mobile robots play an important role and are heavily utilized. These mobile robots are used as UAVs, marine robots, or ground robots to perform critical tasks, especially where human interaction is impossible [1]. While performing the task in an unstructured environment, autonomous navigation is the major challenge for any mobile robot, as it involves many non-linear constraints. To enable the means for autonomous navigation, the mobile robot should be equipped with a power control unit, sensory mechanisms, and an intelligent path-planning algorithm [2]. The success of any autonomous system completely depends on selecting and implementing effective path planners. Therefore, path planners must be able to determine the best-fit parameters among all possibilities to generate an optimal path by avoiding obstacles. Autonomous navigation is not limited to a single objective function; hence, an artificially intelligent computational approach is required to deal with the multi-objective problem of fulfilling the goal of effective navigation in a complex, unstructured environment [3]. In mobile robot navigation (MRN) [4], the major challenges observed are path planning in the presence of static and dynamic obstacles, path planning in the presence of dynamic goals, and navigation in the presence of multiple robots. These challenges

become very complex when counting over-optimization (obtaining the shortest path in the minimum navigational time). Additionally, they require significant computing efforts for seemingly simpler problems, including comparison with other intelligent approaches.

This work presents the application of the FA along with the CF as a multifunctional approach to static and dynamic conditions. Real-time robots encounter challenges in real-world scenarios, including noisy sensor data, perceptual uncertainties, dynamic environmental complexities, computational efficiency constraints, limited model generalization, and the critical need for safety and collision avoidance. To address these issues comprehensively, strategies such as sensor fusion for data refinement, adaptive obstacle avoidance, efficient algorithm design, machine learning for environment adaptation, and robust safety protocols must be implemented to enhance the robot's performance and reliability in dynamic, uncertain environments. Achieving the goal of navigation requires the quick computation of efficient paths, which is presented here by introducing first the CF and then the FA. The key advantage of adopting the CF in MRN is an extension of the option for selecting the optimum path by the characteristics of the non-void family and non-void set. Another noteworthy advantage is to present the optional optimal paths into the classes by FL. The obstacles are classified into non-void sets, allowing the robot to make a quick decision and become more efficient. The choice of probability and the choice of optimality are the two major characteristics of the choice function. These characteristics are applied to improve the firefly algorithm for mobile robot navigation. The chaos of fireflies is transformed into the index set of the non-void family in this paper. The obstacle avoidance function comprises the distributed probability, and the distance–time function comprises the fuzzy logic-based index set. Thus, MRN is studied and applied using mobile robot navigation, multiple mobile robots, chaotic target seeking, multiple targets seeking, chaotic obstacle position, topological spaces, and the firefly's micro- and macro-attraction. The simulation and real-time results are provided for validation in an uncertain environment, and the obtained results are optimal compared to other navigational controllers. As per the author's belief and knowledge, very little work has been published on the path planning of mobile robots in an unstructured environment in the presence of static and dynamic (obstacle and goal) conditions using FA.

This paper is organized as follows. The introduction to the literature review is presented in Section 2. Section 3 presents the proposed functional firefly algorithm with its mathematical analysis. In Section 4, the simulation and experimental results are presented and compared. The conclusion and the future scope are presented in Section 5.

## 2. Review of Literature

From the review of the available literature on intelligent path-planning techniques such as cell decomposition [5], fuzzy logic [6], neural network [7], particle swarm optimization algorithm [8], ant colony algorithm [9], bacterial foraging optimization [10], harmony search algorithm [11], cuckoo search algorithm [12], and dragonfly algorithm [13], it is clear that the applications of metaheuristic algorithms for solving mobile robot navigational problems are growing rapidly compared to heuristic algorithms due to their high-performance capabilities. The “randomization” and “local search” features of the metaheuristic algorithm are critical. Randomization provides a good way to move away from local search to search on a global scale, and therefore, the metaheuristic algorithm is intended to be suitable for global optimization.

In 2008, Yang [14] proposed the firefly algorithm based on the behavior of fireflies for solving various optimization problems in engineering. This firefly algorithm holds two main characteristics of fireflies, i.e., flashing patterns and biological behavior. However, this firefly algorithm follows the three fundamental principles under the two characteristics as defined below:

1. Fireflies are unisex, but their attraction is based on intensity rather than gender;
2. The attraction is proportional to brightness, from lesser brightness to greater brightness;
3. The brightness interacts with the landscape of the objective function.

These three rules are necessary and sufficient for applying the FA in various behavioral applications. The generalization of these rules is possible because of the specific requirements and applications, i.e., 3D navigation of mobile robots, target-seeking applications, chaotic obstacle positions, topological spaces, etc.

The basic formula of attractiveness interacts with the intensity of light presented as follows:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (1)$$

where  $\beta$  is the variation of the attractiveness,  $\gamma$  is the light absorption coefficient, and  $\beta_0$  is the attractiveness at  $\gamma = 0$ . Here, the choice of the axiom is applied to the existing firefly algorithm. Next, the movement of a firefly ( $i$ ) attracts a firefly ( $j$ ) due to greater brightness, and it is presented as follows:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha_t \epsilon_i^t \quad (2)$$

where  $\alpha$  and  $\epsilon$  are the randomized parameters, although  $\epsilon$  is a vector of random numbers defined over the Gaussian number as uniform distribution.

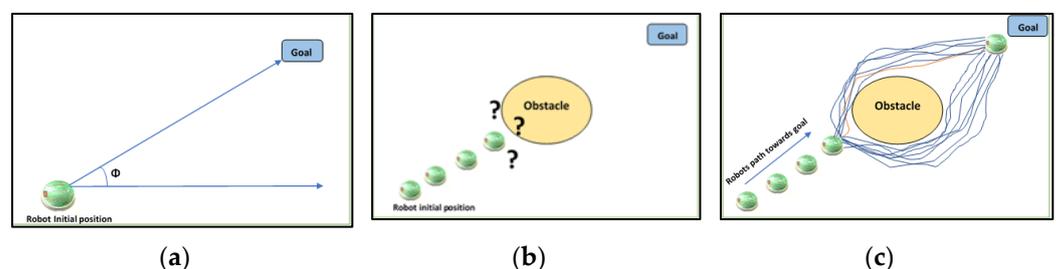
FA has been studied and implemented to solve various optimization problems in engineering and science. The fault detection in robots [15], economic emission dispatched problem [16], reliability–redundancy optimization [17], mixed variable structural optimization problem [18], cooperative networking problem [19], combinatorial optimization problem [20], learning from demonstration problem [21], and the dynamic environment problem [22] are a few of them. The FA has shown great performance and created a good impact in the category of the population-based algorithm. The FA has the ability to solve multi-model optimization and extremely non-linear problems excellently. It has a better convergence speed for finding a global solution in a complex environment and starts iteration processes without a good initial solution. As a result, FA and CF are chosen here to investigate the environment for mobile robot navigation, which includes dynamic obstacles and dynamic goals. Many researchers use the FA to solve mobile robot navigation problems. However, very few of the papers meet the requirements of the navigation. The navigation of the mobile robot using FA in the presence of a static environment was demonstrated by Liu et al. [23] and Hidalgo-Paniagua et al. [24]. However, in both approaches, the navigation strategies were presented in the simulation environment in the presence of the static obstacle. The navigation task is difficult in the presence of a moving obstacle, so Brand et al. [25] and Patle et al. [26] worked to develop the FA for a dynamic environment. In their work, the environment with single and multiple moving obstacles was tested for a single mobile robot system, respectively. The analysis of a multi-mobile robot system in the presence of multiple obstacles was demonstrated by Kim et al. [27], but the results of the navigation were limited to the simulation environment only and not to the real environment. The study of single and multiple mobile robot systems over real-time experiments was carried out by Patle et al. [28,29]. His work shows multiple mobile robots navigating multiple obstacles. The application of FA is not only limited to ground robots but also includes the navigation of aerial and underwater robots. Wang et al. [30] developed an FA-based path-planning strategy for the aerial robot. It outperformed other approaches in avoiding hazardous areas in a complex, crowded environment and reducing fuel costs. Similarly, the FA-based underwater path planning strategy was examined by Sutantyong et al. [31]. Their work primarily focused on the scheduling strategy of swarm robots to avoid interface and jamming in underwater conditioning using the principles of FA. In other work, they also presented underwater navigation in a partially known environment using a leavy-flight–firefly-based approach. To explore and enhance the performance of the mobile robot system, the FA has been introduced into a few techniques such as Q-learning [32], invasive weed optimization [33], radial basis function neural network [34], pareto-based optimization algorithm [35], and many more. Table 1 gives an overview of the related work using the firefly algorithm.

**Table 1.** Comparison of Related Work for the Firefly Algorithm.

| Reference No. | Single Robot/Multi-Robot/Aerial Robot/Underwater Robot | Simulation | Experimental | Static/Dynamic Obstacle | Hybrid Technique Used |
|---------------|--|------------|--------------|-------------------------|-----------------------|
| [20]          | Single   | Y          | N            | Static                  | N                     |
| [21]          | Single   | Y          | N            | Static                  | N                     |
| [22]          | Single   | Y          | N            | Static and dynamic      | N                     |
| [23]          | Single   | Y          | N            | Static and dynamic      | N                     |
| [24]          | Multi-robot  | Y          | N            | Dynamic                 | N                     |
| [25]          | Multi-robot  | Y          | Y            | Dynamic                 | Y                     |
| [26]          | Multi-robot  | Y          | Y            | Dynamic                 | Y                     |
| [27]          | Aerial robot   | Y          | N            | Dynamic                 | N                     |
| [28]          | Underwater   | Y          | N            | Dynamic                 | N                     |

### 3. Proposed Functional Firefly Algorithm

Various challenges exist in the real environment; some of them are the localization of position in the environment, the determination of a goal and an organized path towards it, obstacle avoidance mechanisms, and the generation of an optimal path in a minimal amount of time. Figure 1a shows that no obstacle is present in the environment between the robot and the goal position; hence, the robot reaches the goal position by using Euclidian distance, which is optimal. But in Figure 1b, when the robot moves from its initial position to the goal position, it detects the obstacle and stops. The obstacle avoidance mechanism activates, and then, the robot avoids the obstacle, as shown in Figure 1c. While avoiding the obstacles, it may produce several paths up to the goal position, while achieving the shortest path from the robot’s initial position to the robot’s goal position is the proposed study’s primary goal. The proposed algorithm addresses the issues mentioned above of robot navigation over the choice function. Any mobile robot navigation is based on the likelihood of selecting the best path, which is a function of distance and time. Therefore, probability plays an important role in executing the navigation of mobile robots. As a non-void family, CF comprises a set of probabilistic choices. Here, the axiom of choice generalizes to FA. The fireflies are defined over a finite set, and the set of fireflies with a distributed probability then generates the function of choice. The non-void family of fireflies with a distributed probability comprises the classification. The proposed FFA holds several advancements in context to theory and application. Flashing pattern feasibility is defined on the finite interval to optimize the trajectory. The basic firefly algorithm is based on a variation of the attractiveness of fireflies. The variable  $\beta$  is attractiveness.  $\beta_0$  exists if attractiveness is defined at the distance  $\gamma = 0$ .



**Figure 1.** Navigational challenges for a mobile robot in (a) exploring the environment for the shortest distance, (b) obstacle avoidance, and (c) selection of an optimal path.

The key idea of this paper is to define attractiveness at the neighborhood of the distance of not  $\gamma = 0$  exactly but very close to zero. This approach is called calculus. The neighborhood of  $\beta_0$  is defined by an interval  $(\beta_0 - \delta, \beta_0 + \delta)$ , where  $\delta$  is the small, positive real number.  $\beta_0 - \delta$  is the point at the left-hand side from  $\gamma = 0$ , and  $\beta_0 + \delta$  is the point

at the right-hand side from  $\gamma = 0$ . The reason for applying this concept is to achieve optimization. This is an optimal input that gives the optimal output correspondingly. For example, if  $\beta_0$  is 2, and  $\delta$  is 0.0001, then the neighborhood will be  $(2 - 0.0001, 2 + 0.0001)$  or  $(1.9998, 2.0001)$ . Thus, this is very close to 2 but not exactly equal to 2, and hence, the resultant output  $\beta$  will be approaching the corresponding value. Hence, Equation (3) holds the limit ( $L$ ) represented by  $|\beta - L| < \varepsilon$ , where  $\varepsilon$  is a small, real positive number. Hence, Equation (3) is represented by a limit as follows:

$$\lim_{\gamma \rightarrow 0} \beta = L \tag{3}$$

The attractiveness is measured from the left-hand side (from  $\beta_0 - \delta$  to  $\beta_0$ ) and denoted by  $\beta_e$  for attractiveness at the point  $\beta_0 - \delta$ , represented as follows:

$$\lim_{\gamma \rightarrow 0^-} \beta^- = \beta_e e^{-\gamma r^2} \tag{4}$$

Similarly, the attractiveness is measured from the right-hand side (from  $\beta_0 + \delta$  to  $\beta_0$ ) and denoted by  $\beta_l$  for attractiveness at the point  $\beta_0 + \delta$ , represented as follows:

$$\lim_{\gamma \rightarrow 0^+} \beta^+ = \beta_l e^{-\gamma r^2} \tag{5}$$

If  $\lim_{\gamma \rightarrow 0^-} \beta = \lim_{\gamma \rightarrow 0^+} \beta$ , then Equation (1) exists, which will be unique.

The source of brightness and its area are also formulated in the proposed FFA. It depends on the size of the firefly, although the difference is minor but measurable. The source of brightness ( $S$ ) is defined as a function of  $\beta_0$ ,  $\beta_e$ , and  $\beta_l$ , which is defined as follows:

$$S = f(\beta_0, \beta_e, \beta_l) \tag{6}$$

The distance between two fireflies that are attracted to each other is also reviewed as an advantage of the FFA. A firefly attracts those whose distance is less than other fireflies, although brightness is the same. The deviation in the distance is  $\delta$ , although the intensity remains the same, presented as follows:

$$\text{Left-Hand Deviation : } \beta = \beta_e e^{-\gamma \delta^2} \tag{7}$$

$$\text{Right-Hand Deviation : } \beta = \beta_l e^{-\gamma \delta^2} \tag{8}$$

The position of the firefly is measured over its central tendency in the proposed FFA. This lies in the displacements, i.e., up, down, diagonal, horizontal, etc., although brightness is the same. The deviation in the position of the fireflies exists when the following is true:

$$\beta_e e^{-\gamma \delta^2} \neq \beta_l e^{-\gamma \delta^2} \tag{9}$$

The discrete approach of attractiveness is studied in this paper. The proposed FFA establishes robot navigation based on the natural conjugation of fireflies. The probability of attraction is only by brightness, but its discrete and continuous distributions are also generalized in the proposed FFA. The choice function plays a crucial role in executing the idea of a functional firefly algorithm for robot navigation. This approach achieves optimum navigation. The application of the choice function receives the dynamic decision. The choice function is a mathematical rule applied as the association of the elements of the two non-empty sets such that each element of the first set has the unique image of the element of the second set. Notable is the self-map mechanism, where the pre-image and image are identical. This identity characteristic establishes a distinct path for navigation. The self-image approach is associated with the firefly position, and its pre-image selects the unique and optimal path. The mathematical definition of the choice function is presented as follows:

*Choice Function:* Let  $\Lambda$  be a non-void set and  $\lambda \in \Lambda$ .  $f$  is said to be a choice function if  $f(\lambda) \in \lambda, \forall \lambda \in \Lambda$ . Its application to robot navigation is presented as follows:  $\Lambda$  is the non-empty finite set of the position of fireflies, which is coordinated in three dimensions. The brighter firefly is denoted by  $\lambda \in \Lambda$ , and the movement is defined by the displacement from the less-bright firefly as the pre-image or the domain as the first set to the image as the co-domain of the second set. This displacement is referred to as the range  $f(\lambda) \in \lambda$  of the function, which is the real number obtained by the choice rule.

*Axiom of Choice Function:* There exists a choice function ( $f$ ) for each non-void family of non-void sets if  $\{X_\lambda : \lambda \in \Lambda\}$  is a family of sets such that  $\Lambda \neq \emptyset$  and  $x_\lambda$  for each  $\lambda \in \Lambda$ ; then, there exists  $f$  on  $\Lambda$  such that  $f(\lambda) \in X_\lambda$  for each  $\lambda \in \Lambda$ .

*Cartesian-Choice Operator:* Let  $\{X_\lambda : \lambda \in \Lambda\}$  be an arbitrary collection of sets induced by  $\Lambda$ . Then, the cartesian product of this collection is the set of all mapping.

$$X : \Lambda \rightarrow \lambda = \{X_\lambda : \lambda \in \Lambda\} : X(\lambda) \in X_\lambda \tag{10}$$

For all,  $\lambda \in \Lambda$ , and it is denoted by the following:

$$\Lambda\{X_\lambda : \lambda \in \Lambda\} \text{ or by; } x\{X_\lambda : \lambda \in \Lambda\} \tag{11}$$

The set  $X_\lambda$  is called the  $\lambda^{th}$  coordinate set of the product. It is used as a symbol  $X_\lambda$  for the image  $X(\lambda)$  of  $X$  under the mapping  $X$ . Here,  $\Lambda$  is an index set, and  $\lambda \in N$  is the set of natural numbers.

Then,

$$X_\lambda = \{x : x \in N, x \text{ is the multiple of } N\}$$

Hence,

$$X_1 = \{1, 2, 3, \dots\}$$

$$X_2 = \{2, 4, 6, \dots\}$$

$$X_n = \{n, 2n, 3n, \dots\}$$

Here,  $f$  is the function of choice on  $\Lambda$  such that

$$f(\lambda) \in X_\lambda; \lambda \in \Lambda$$

Thus, the axiom of the choice function is generalized to the resultant formula of firefly for the attractiveness over the initial position, defined as follows:

$$x_i^{t+1} : \Lambda \rightarrow U \{x_\beta : \beta \in \Lambda\} : x(\lambda) \in x_\lambda \tag{12}$$

where  $\Lambda$  is the index set of existing properties of the FA, and  $U$  is the proposed transformation over the seven characteristics of fireflies. Hence, the resultant formula of attraction over the axiom of choice is presented as follows:

$$x_i^{t+1} = x_i^t + \frac{\Lambda}{U} [(\beta_0 e^{-\gamma r_{ij}^2}) (x_j^t - x_i^t)] + \alpha_i \epsilon_i^t \tag{13}$$

Here,  $\Lambda$  is an index set.

$\lambda \in N$  (the set of natural numbers)

Then,

$$x_\lambda = \{x : x \in N, x \text{ is the multiple of } N\}$$

The robot navigation architecture is presented in Figure 2. In an uncertain environment, the finite set of obstacles is  $O = \{o_1, \dots, o_n\}$ , the finite set of fireflies of lower and higher intensity of light is  $X = \{x_1^l, x_1^h, \dots, x_n^l, x_n^h\}$ , and the robots' initial position is  $R(x_1^l)$  and the

goal position  $G(x_n^h)$ . The robot (less-bright firefly)  $R(x_1^l)$  is attracted towards the brighter firefly ( $x_1^h$ ) with obstacle avoidance by the optimal choice function  $f(\lambda_1); \lambda_1 \in \Lambda$ . Similarly,  $R(x_1^l)$  follows the preceding rule and reaches the goal  $G(x_n^h)$  by the optimal choice function navigation:  $C : R(x_1^l) \xrightarrow{f(\lambda_n)} G(x_n^h)$ .

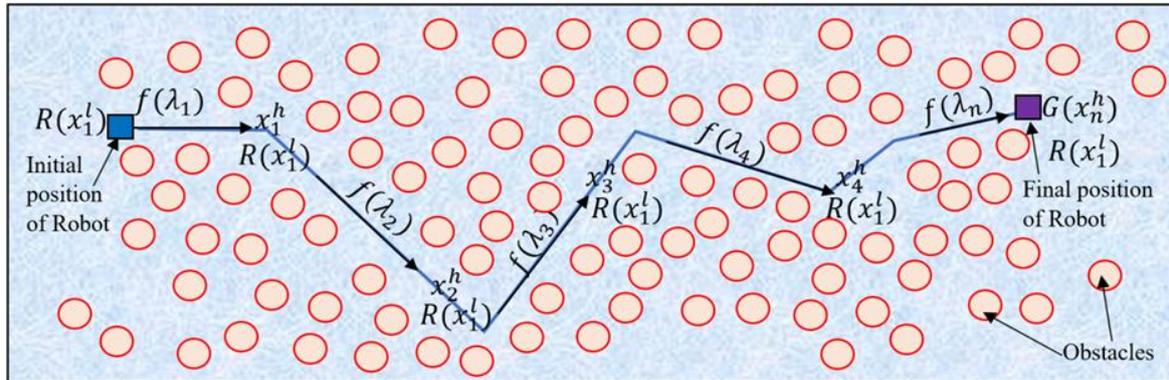


Figure 2. Functional firefly navigation space.

The robot navigation follows the attraction behavior of fireflies as the difference ( $I^H - I^L \geq 0$ ) of the intensity of light of fireflies by ( $I^H > I^L$ ), where  $I^H$  and  $I^L$  are high intensity and low intensity, respectively. Thus, the objectives of optimal navigation, i.e., the shortest path in the minimal time and collision-free path, are achieved by the CF simultaneously. Let the non-void family of the non-void set be  $X = \{ \{x_1^l, x_1^h\}, \dots, \{x_n^l, x_n^h\} \}$ ; then, the choice function is  $f(x) = \{ \{x_1^l\}, \{x_1^l, x_2^h\}, \{x_2^l, x_3^h\}, \dots, \{x_n^h\} \}$ . Thus, the firefly-choice function-based robot navigation is presented as FFA:  $f : R \rightarrow G$ .

The set of lower-intensity fireflies =  $X_1 = X_\lambda^l = \{x_1^l, \dots, x_n^l\}$ ;

The set of higher-intensity fireflies =  $X_2 = X_\lambda^h = \{x_1^h, \dots, x_n^h\}$ .

Then, the set of functional fireflies is defined by the choice function as follows:

$$C = \{C_l, C_h\} = \{ \{x_{n-1}^l, x_n^l\}, \{x_{n-1}^h, x_n^h\} \} = \{X_\lambda : \lambda \in \Lambda\} \tag{14}$$

Here,  $f$  is a function defined on  $\Lambda$  such that  $f(\lambda) \in X_\lambda$  for each  $\lambda \in \Lambda$ . This choice function is applied to the above-defined sets of fireflies and modified again as follows:

$$f : \lambda \rightarrow \lambda \text{ Or, } f : X_\lambda \rightarrow X_\lambda \text{ Or, } f : X_1 \rightarrow X_2 \text{ Or, } X_2 = f(X_1) \tag{15}$$

Next, the objective function is formulated. The decision variables are as given: The choice-function-based navigational path function is  $f(\lambda_1), \dots, f(\lambda_n)$ ; respective time is  $t_1, \dots, t_n$ ; and the navigation rate is  $\frac{x_n^h - x_1^l}{t_n}$ .

Hence, the objective function is given:

$$\min f(\lambda) = \frac{(x_1^h - x_1^l)}{t_1} + \dots + \frac{(x_n^h - x_n^l)}{t_n}, f(\lambda) \in X_\lambda, \lambda \in \Lambda \tag{16}$$

This is subject to  $C_{m1} \frac{1}{t_1} + \dots + C_{mn} \frac{1}{t_n} = f(\lambda_n)$  and  $t_1, \dots, t_n \geq 0$ . The navigation function table is presented as follows in Table 2.

**Table 2.** Navigation Functions.

| SN. | Navigation Direction | Representations                  |
|-----|----------------------|----------------------------------|
| 1   | Left Move            | $\beta_0 + \delta$               |
| 2   | Straight Move        | $\beta_0 = \delta$               |
| 3   | Right Move           | $\beta_0 - \delta$               |
| 4   | Up Move              | $\beta_0 / \delta$               |
| 5   | Down Move            | $\delta / \beta_0$               |
| 6   | Constant             | $\beta_0$                        |
| 7   | Left Curve Move      | $(\beta_0 + \delta)^n; n \geq 2$ |
| 8   | Right Curve Move     | $(\beta_0 - \delta)^n; n \geq 2$ |

Hence, by the Cartesian product of the  $X_1$  and  $X_2$ , we find the move according to the position of obstacles. There are finite options to choose the move, as defined by the choice function.

$$f : X_1 \times X_2 = \left\{ (x_1^l, x_1^h), \dots, (x_n^l, x_n^h) \right\} \tag{17}$$

The distance vector is defined for controlling the obstacle position. Let the set of the position point of the obstacle be  $\{D_1, \dots, D_n\}$ . Each point has the vector by the characteristic of this set of finite sums of intervals that hold the probabilistic decision of optimal path. The probability of optimal path is  $\{p(D_1), \dots, p(D_n)\}$ . The domain of probability for the rule (choice function) to decide the optimal path is represented by the following matrix:

$$D = \begin{bmatrix} d_{11} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & d_{nn} \end{bmatrix}$$

Thus, the rule (choice function) is set by the co-domain of the probable vectors as  $\{(d_{11}, \dots, d_{1n}), \dots, (d_{91}, \dots, d_{9n})\}$ . The permutation of the vectors of the co-domain for generating the range as a rule by the following vector spaces is illustrated in Tables 3–5.

**Table 3.** Vector Space-Based Navigational Decision.

| Obstacle Position | Distance Classification | Choice of Axiom of (II)     | Metric Decision Vector          |
|-------------------|-------------------------|-----------------------------|---------------------------------|
|                   | $D_1 \dots D_N$         | $A(D_1) \dots A(D_N)$       | $C_i [A(D_1), A(D_N)]$          |
| LOD               | $d_{11} \dots d_{1n}$   | $A(d_{11}) \dots A(d_{1n})$ | $[A(d_{11}), \dots, A(d_{1n})]$ |
| FOD               | $d_{21} \dots d_{2n}$   | $A(d_{21}) \dots A(d_{2n})$ | $[A(d_{21}), \dots, A(d_{2n})]$ |
| ROD               | $d_{31} \dots d_{3n}$   | $A(d_{31}) \dots A(d_{3n})$ | $[A(d_{31}), \dots, A(d_{3n})]$ |
| UOD               | $d_{41} \dots d_{4n}$   | $A(d_{41}) \dots A(d_{4n})$ | $[A(d_{41}), \dots, A(d_{4n})]$ |
| DOD               | $d_{51} \dots d_{5n}$   | $A(d_{51}) \dots A(d_{5n})$ | $[A(d_{51}), \dots, A(d_{5n})]$ |
| ULDOD             | $d_{61} \dots d_{6n}$   | $A(d_{61}) \dots A(d_{6n})$ | $[A(d_{61}), \dots, A(d_{6n})]$ |
| URDOD             | $d_{71} \dots d_{7n}$   | $A(d_{71}) \dots A(d_{7n})$ | $[A(d_{71}), \dots, A(d_{7n})]$ |
| DLDOD             | $d_{81} \dots d_{8n}$   | $A(d_{81}) \dots A(d_{8n})$ | $[A(d_{81}), \dots, A(d_{8n})]$ |
| DRDOD             | $d_{91} \dots d_{9n}$   | $A(d_{91}) \dots A(d_{9n})$ | $[A(d_{91}), \dots, A(d_{9n})]$ |

where LOD, left obstacle distance; FOD, front obstacle distance; ROD, right obstacle distance; UOD, up obstacle distance; DOD, down obstacle distance; ULDOD, up left obstacle distance; URDOD, up right obstacle distance; DLDOD, down left obstacle distance; DRDOD, down right obstacle distance.

Hence, the turning or transformation or direction function of the mobile robot is given:

$$T = f\{X_D, X_S; D, \lambda \in \Lambda\} \tag{18}$$

The conditional rule for controlling the MRN is given as follows in Tables 6 and 7.

**Table 4.** Euclidean Metric Grading for Optimal Navigation.

| Obstacle Position | Distance Classification<br>( $D_1, \dots, D_N$ ) | Choice of Axiom<br>of (II)<br>$[A(D_1), \dots, A(D_N)]$ | Distance Decision<br>Matrix<br>$C_i [A(D_1), \dots, A(D_N)]$  |
|-------------------|--|---|---|
| LOD               |  | $[A(d_{11}), \dots, A(d_{1n})]$                         | $\begin{bmatrix} A(d_{11}) & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & A(d_{9n}) \end{bmatrix}$ |
| FOD               | VVN-VN-N-F-VF-VVF                                | $[A(d_{21}), \dots, A(d_{2n})]$                         |   |
| ROD               | VVN—Very Very Near                               | $[A(d_{31}), \dots, A(d_{3n})]$                         |   |
| UOD               | VN—Very Near                                     | $[A(d_{41}), \dots, A(d_{4n})]$                         |   |
| DOD               | N—Near   | $[A(d_{51}), \dots, A(d_{5n})]$                         |   |
| ULDOD             | F—Far  | $[A(d_{61}), \dots, A(d_{6n})]$                         |   |
| URDOD             | VF—Very Far                                      | $[A(d_{71}), \dots, A(d_{7n})]$                         |   |
| DLDOD             | VVF—Very Very Far                                | $[A(d_{81}), \dots, A(d_{8n})]$                         |   |
| DRDOD             |  | $[A(d_{91}), \dots, A(d_{9n})]$                         |   |

**Table 5.** Speed Grading Rule for Fast Navigation.

| Obstacle Position | Speed Classification<br>( $S_1, \dots, S_N$ ) | Choice of Axiom<br>of (II)<br>$A(S_1), \dots, A(S_N)$ | Speed Decision<br>Matrix<br>$C_i [A(S_1), \dots, A(S_N)]$   |
|-------------------|---|---|---|
| LOD               |   | $A(S_{11}), \dots, A(S_{1n})$                         | $\begin{bmatrix} A(S_{11}) & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & A(S_{9n}) \end{bmatrix}$ |
| FOD               | VVF-VF-F-S-VS-VVS                             | $A(S_{21}), \dots, A(S_{2n})$                         |   |
| ROD               | VVF—Very Very Fast                            | $A(S_{31}), \dots, A(S_{3n})$                         |   |
| UOD               | VF—Very Fast                                  | $A(S_{41}), \dots, A(S_{4n})$                         |   |
| DOD               | F—Fast  | $A(S_{51}), \dots, A(S_{5n})$                         |   |
| ULDOD             | S—Slow  | $A(S_{61}), \dots, A(S_{6n})$                         |   |
| URDOD             | VS—Very Slow                                  | $A(S_{71}), \dots, A(S_{7n})$                         |   |
| DLDOD             | VVS—Very Very Slow                            | $A(S_{81}), \dots, A(S_{8n})$                         |   |
| DRDOD             |   | $A(S_{91}), \dots, A(S_{9n})$                         |   |

**Table 6.** Linear If-Then Rule.

| If  | Then  |
|---|---|
| $LOD = \frac{(LOD)_i}{HA}; FOD = \frac{(FOD)_i}{HA}; ROD = \frac{(ROD)_i}{HA}; HA = \frac{(HA)_i}{(LD)(RD)}; UOD = \frac{(UOD)_i}{HA};$<br>$ULDOD = \frac{(ULDOD)_i}{HA}; URDOD = \frac{(URDOD)_i}{HA}; DOD = \frac{(DOD)_i}{HA};$<br>$DLDOD = \frac{(DLDOD)_i}{HA}; DRDD = \frac{(DRDOD)_i}{HA}$ | $\frac{HV}{HV_{ijkl}} \cdot \frac{VV}{VV_{ijkl}}$ |

where HV, horizontal velocity; VV, vertical velocity.

**Table 7.** Non-Linear If-Then Rule.

| If   | Then  |
|--|---|
| $\frac{(LOD)_i}{LOD} \cdot \frac{(FOD)_i}{FOD} \cdot \frac{(ROD)_i}{ROD} \cdot \frac{(HA)_i}{HA} \cdot \frac{(UOD)_i}{UOD} \cdot \frac{(ULDOD)_i}{ULDOD} \cdot \frac{(URDOD)_i}{URDOD} \cdot \frac{(DOD)_i}{DOD} \cdot \frac{(DLDOD)_i}{DLDOD}$<br>$\frac{(LOD)_i}{(LOD)_i} \cdot \frac{(FOD)_i}{(FOD)_i} \cdot \frac{(ROD)_i}{(ROD)_i} \cdot \frac{(HA)_i}{(HA)_i} \cdot \frac{(UOD)_i}{(UOD)_i} \cdot \frac{(ULDOD)_i}{(ULDOD)_i} \cdot \frac{(URDOD)_i}{(URDOD)_i} \cdot \frac{(DOD)_i}{(DOD)_i} \cdot \frac{(DLDOD)_i}{(DLDOD)_i}$ | $\frac{HV}{HV_{ijkl}} \cdot \frac{VV}{VV_{ijkl}}$ |

Thus, the compact rule for the effective navigation of robots is defined as follows:

$$W_{ijkl} = Dis_{ijkl} \left[ \left( \frac{X_{LOD}}{LOD_i} \right) \left( \frac{X_{FOD}}{FOD_i} \right) \left( \frac{X_{ROD}}{ROD_i} \right) \left( \frac{X_{HA}}{HA_i} \right) \left( \frac{X_{UOD}}{UOD_i} \right) \left( \frac{X_{ULDOD}}{ULDOD_i} \right) \left( \frac{X_{URDOD}}{URDOD_i} \right) \left( \frac{X_{DOD}}{DOD_i} \right) \left( \frac{X_{DLDOD}}{DLDOD_i} \right) \right] \quad (19)$$

Similarly, the velocity function is presented:

$$(Vel)^{XLV}_{ijkl} = \frac{W_{ijkl}}{(vel_{LV})X_{LV}_{ijkl}}; (Vel)^{XRV}_{ijkl} = \frac{W_{ijkl}}{(vel_{RV})X_{RV}_{ijkl}};$$

$$(Vel)^{XHV}_{ijkl} = \frac{W_{ijkl}}{(vel_{VV})X_{HV}_{ijkl}} \text{ and } (Vel)^{XVV}_{ijkl} = \frac{W_{ijkl}}{(vel_{VV})X_{VV}_{ijkl}}$$

Hence,  $LV = \left( \frac{\sum (Vel) \frac{X_{LV}}{(Vel)X_{LV}}}{\sum (Vel)X_{LV}} \right)$ ;  $RV = \left( \frac{\sum (Vel) \frac{X_{RV}}{(Vel)X_{RV}}}{\sum (Vel)X_{RV}} \right)$ ;  $HV = \left( \frac{\sum (Vel) \frac{X_{HV}}{(Vel)X_{HV}}}{\sum (Vel)X_{HV}} \right)$ ; and  $VV = \left( \frac{\sum (Vel) \frac{X_{VV}}{(Vel)X_{VV}}}{\sum (Vel)X_{VV}} \right)$ .

The continuous representation of the above is given as follows:

$$LV = \frac{\int (Vel) \frac{X_{LV}}{(Vel)X_{LV}} d(Vel)}{\int (Vel)X_{LV}d(Vel)}; RV = \frac{\int (Vel) \frac{X_{RV}}{(Vel)X_{RV}} d(Vel)}{\int (Vel)X_{RV}d(Vel)}; HV = \frac{\int (Vel) \frac{X_{HV}}{(Vel)X_{HV}} d(Vel)}{\int (Vel)X_{HV}d(Vel)}; \text{ and } VV = \frac{\int (Vel) \frac{X_{VV}}{(Vel)X_{VV}} d(Vel)}{\int (Vel)X_{VV}d(Vel)}.$$

The pseudocode for Functional Firefly Algorithm 1 is described as:

---

**Algorithm 1** Functional Firefly (FFA).

---

```
#Function FunctionalFireflyAlgorithm():
  Initialize robot navigation
  Initialize the population of fireflies  $X_\lambda = \{x_1, \dots, x_n\}$ .
  Initialize the objective function  $f(\lambda) \in X_\lambda; \lambda \in \Lambda$ .
  Initialize light intensity of fireflies (I)
  Initialize absorption coefficient ( $\gamma$ )
  Initialize the distance between two fireflies ( $r$ )
  Vary attractiveness ( $e^{-\gamma r}$ ).
  Classify fireflies into two groups based on intensity:
   $X_\lambda^l = \{x_1^l, \dots, x_n^l\}$ , l: less intensity.
   $X_\lambda^h = \{x_1^h, \dots, x_n^h\}$ , h: high intensity.
  t = 0
  maxiterations = max( $X_\lambda$ )
  While t < max $X_\lambda$  :
    # Update lowintensity fireflies
    For i in  $X_\lambda^l$ :
      If  $C_l \xrightarrow{f_1(\lambda)} C_h$ ;
         $x_1^l \rightarrow x_1^h (x_1^l < x_1^h)$ ;
      If  $C_l \xrightarrow{f_2(\lambda)} C_h$ ;
         $x_2^l \rightarrow x_2^h (x_2^l < x_2^h)$ ;
      If  $C_l \xrightarrow{f_3(\lambda)} C_h$ ;
         $x_3^l \rightarrow x_3^h (x_3^l < x_3^h)$ ;
      If  $C_l \xrightarrow{f_4(\lambda)} C_h$ ;
         $x_4^l \rightarrow x_4^h (x_4^l < x_4^h)$ ;
      If  $C_l \xrightarrow{f_5(\lambda)} C_h$ ;
         $x_{35}^l \rightarrow x_5^h (x_5^l < x_5^h)$ ;
    # Calculate the objective function
     $f(\lambda) = f_1(\lambda) + f_2(\lambda) + f_3(\lambda) + f_4(\lambda) + f_5(\lambda)$ 

    # Update fireflies based on the objective function
     $R \xrightarrow{f(\lambda)} G$ .
    t = t + 1
  # Optimized f( $\lambda$ )
  Optimized function f( $\lambda$ )
  End robot navigation
```

---

## 4. Simulation and Experimental Result Analysis

### 4.1. Mobile Robot Navigation Simulation Results

To demonstrate the effectiveness of our developed approach across diverse environmental conditions, we conducted numerous trials in both static and dynamic environments featuring various obstacles. Simulation analysis was carried out using MATLAB R2021a software, providing the flexibility to customize the environment with different obstacle positions, robot placements, and goals. In static environments, obstacle positions remained fixed, allowing adjustments only in the initial robot and goal positions. Conversely, dynamic environments featured variable obstacle and goal positions. Our program was designed to accommodate a variable number of robots and goals. Figures 3 and 4 demonstrate the navigation of a single mobile robot in simulated environments with static configurations, presented in both 3D and 2D formats. During navigation, the robot prioritizes path safety, avoiding obstacles by maintaining a safe distance. We also demonstrated multi-robot navigation strategies in a complex environment with four robots (Figure 5a–d). Each robot had a distinct starting position and a predefined common goal. The paths created by individual robots were uniquely color-coded, illustrating our approach's efficiency in finding optimal collision-free paths even in dynamic environments. Figure 6 presents an environment with two moving obstacles (green and pink) and a fixed goal. The robot autonomously identifies approaching obstacles and adjusts its position to maintain a safe distance. Figure 7 illustrates mobile robot navigation when the goal itself is in motion. Our approach consistently generates optimal pathways in both cases, effectively addressing uncertainties in dynamic environments. These results highlight the robustness of our proposed approach.

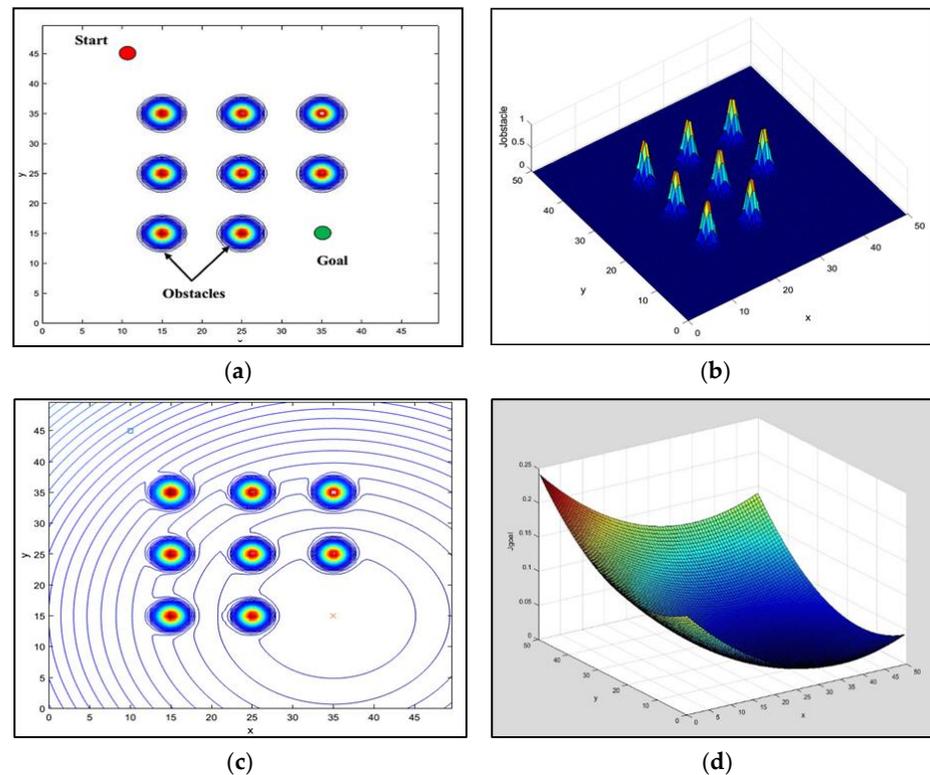


Figure 3. Cont.

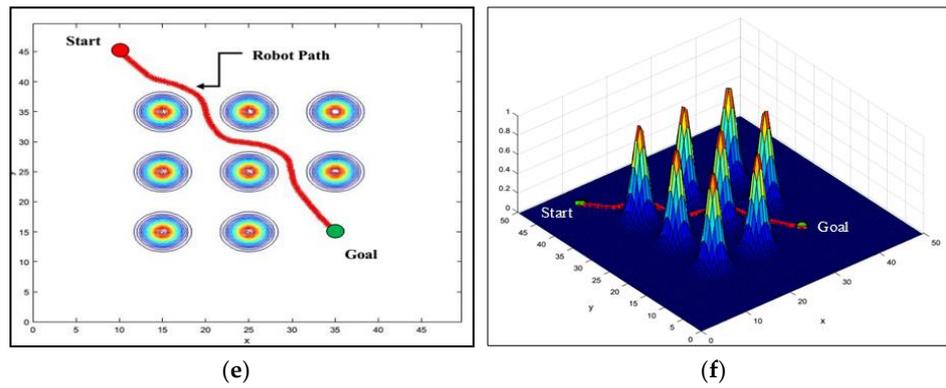


Figure 3. Mobile robot navigation in a static environment. (a) Two-dimensional representation of the environment. (b) Obstacle function. (c) Total cost function. (d) Goal function. (e) Navigation in 2D environment. (f) Navigation in 3D.

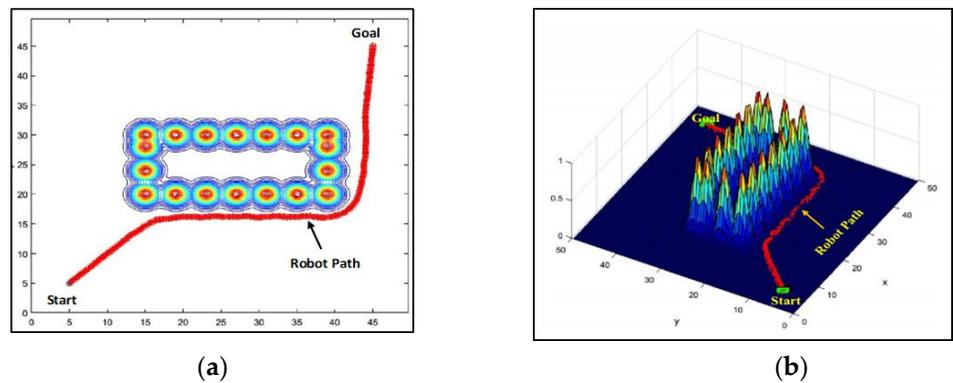


Figure 4. Mobile robot navigation in static environment. (a) Robot navigation in 2D environment. (b) Robot navigation in 3D.

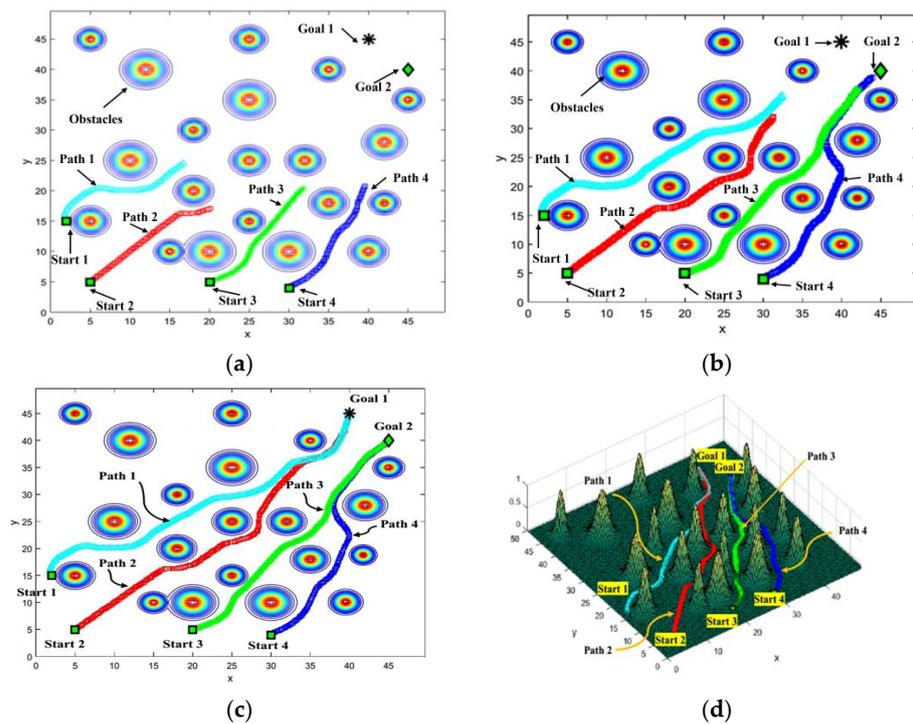


Figure 5. Demonstration of multiple mobile robot navigation in a static environment, illustrated in (a–d).

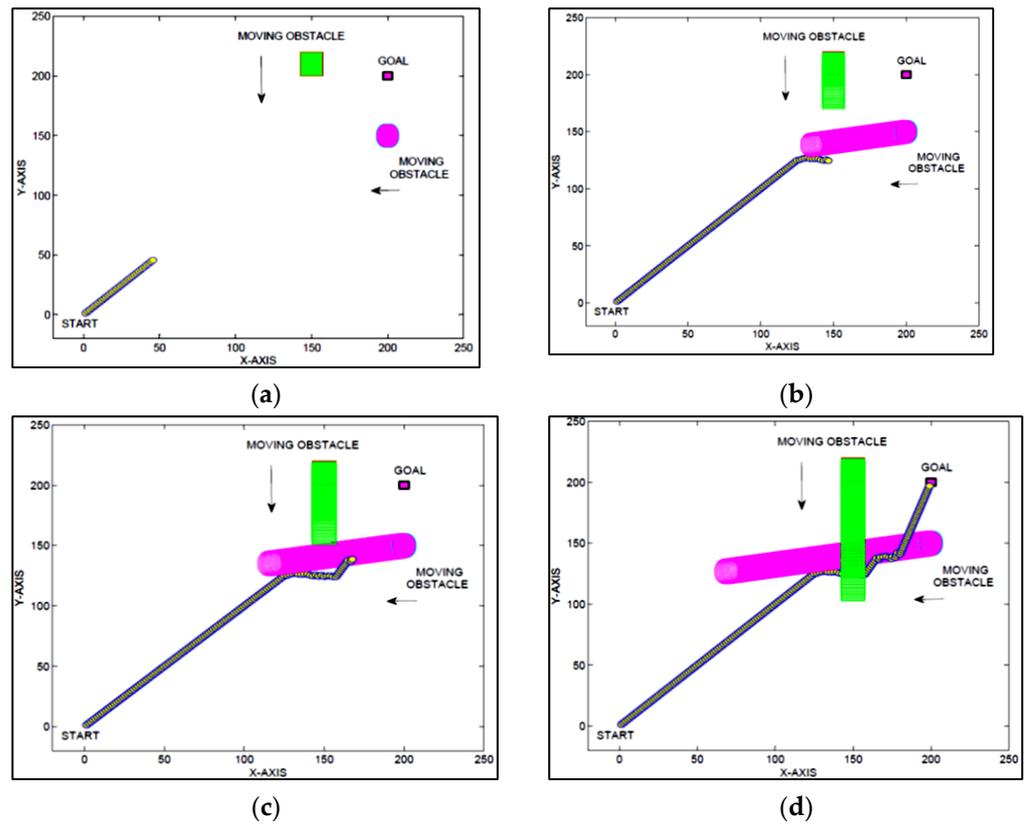


Figure 6. Demonstration of mobile robot navigation in a dynamic environment with two moving obstacles: illustrated in (a–d).

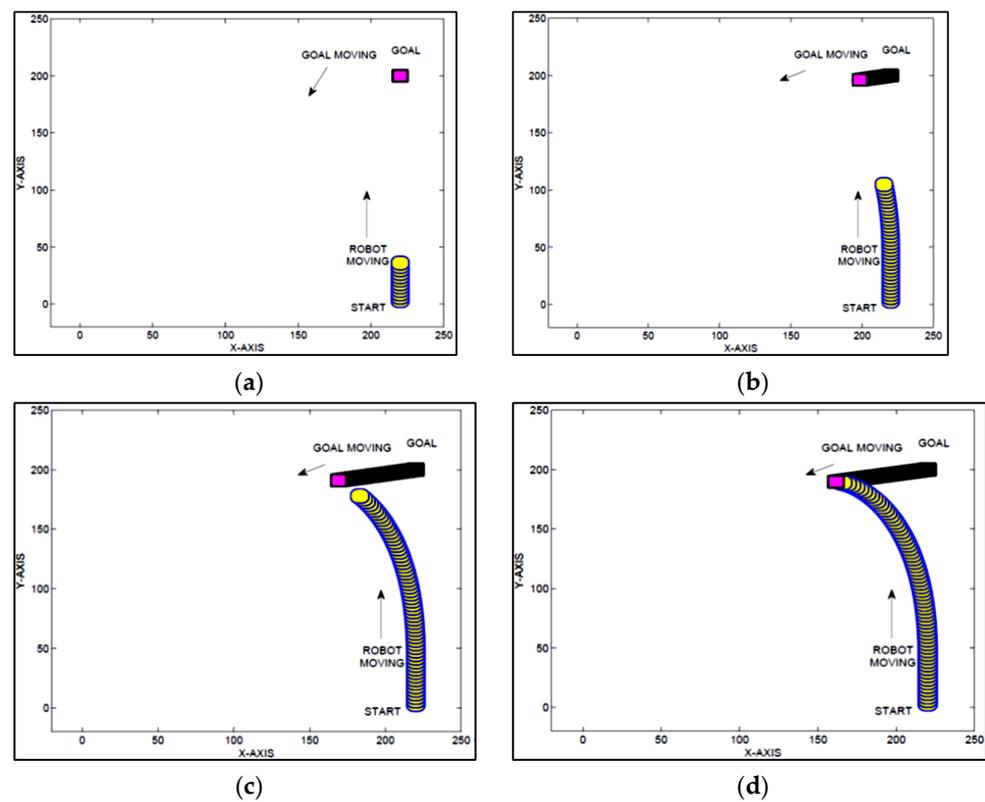


Figure 7. Demonstration of mobile robot navigation in a dynamic environment with moving goal: illustrated in (a–d).

#### 4.2. Mobile Robot Navigation Experimental Results

To evaluate the effectiveness of the developed approach in real-time scenarios, we employed two distinct types of robots: an in-house developed robot (Figure 8) and the Khepera-II robot (Figure 9). Detailed specifications for these robots can be found in the Appendix A, specifically outlined in Table A1 for the in-house developed robot and Table A2 for the Khepera-II robot. As illustrated in Figures 10–12, we established a consistent experimental setup to assess the approach's performance for single- and multiple-robot systems. This setup confirms that the approach can generate optimal paths comparable to those achieved in simulation, as demonstrated in Figures 3–5.

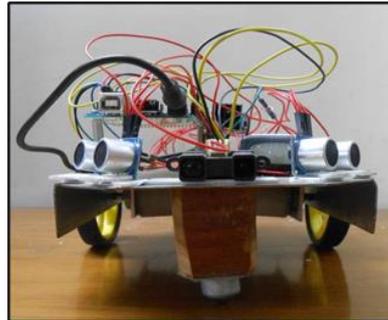


Figure 8. In-house developed a robot.



Figure 9. Khepera-II robot.

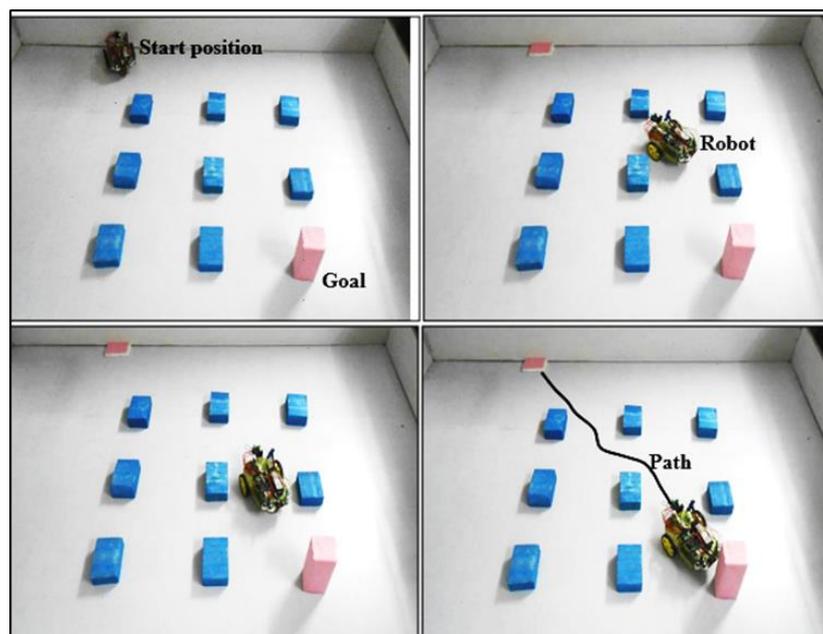


Figure 10. Mobile robot navigation in a real environment (Scenario-1).

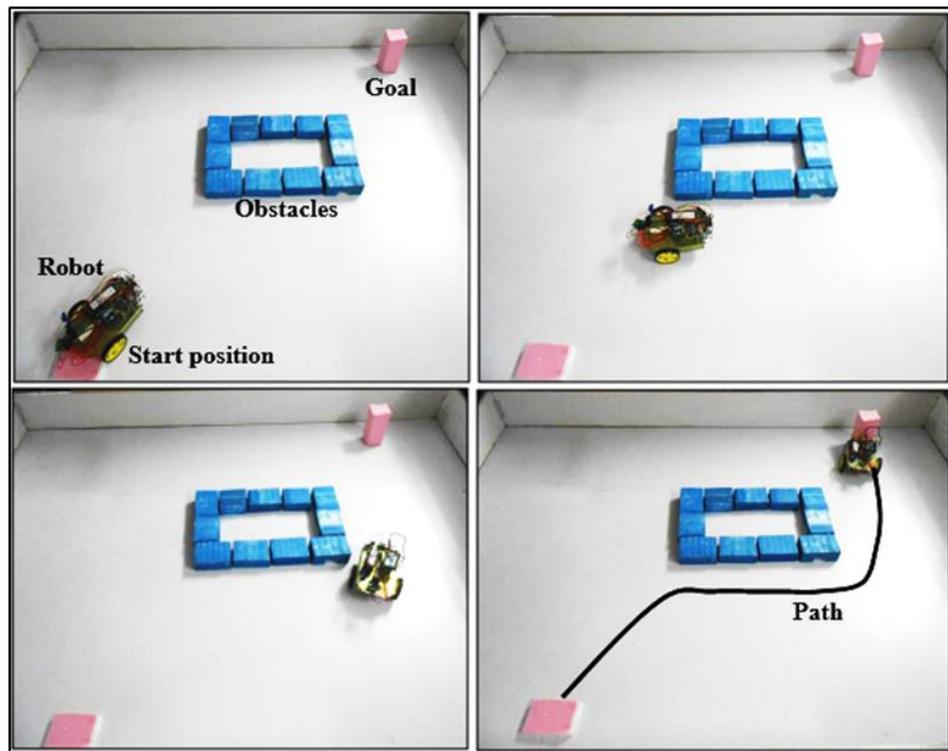


Figure 11. Mobile robot navigation in a real environment (Scenario-2).

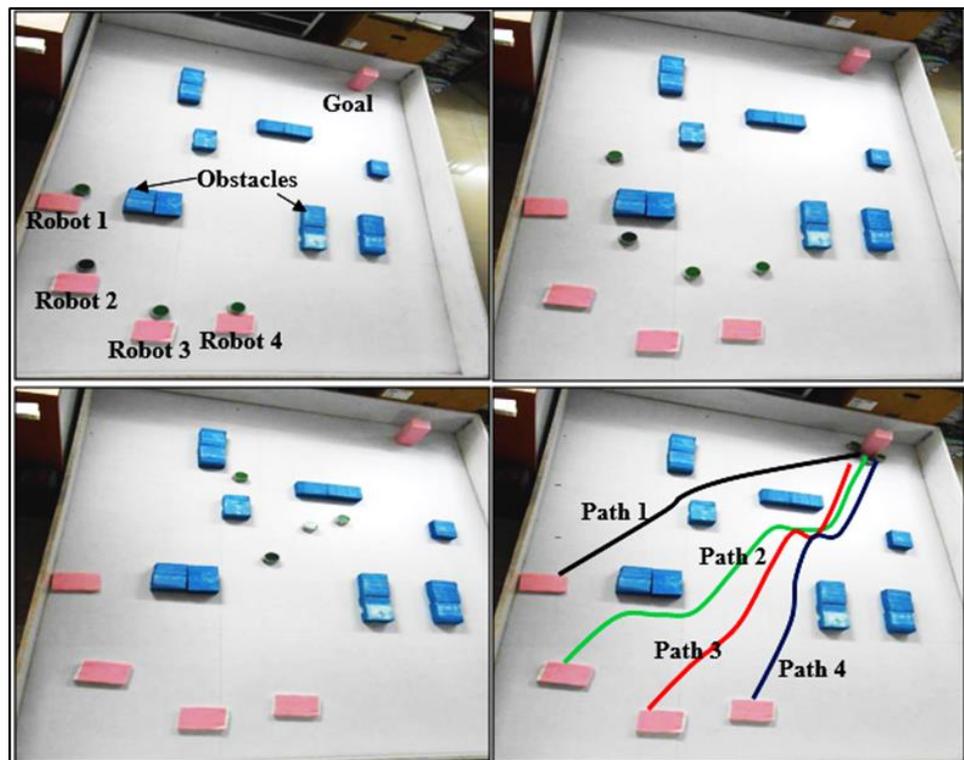


Figure 12. Mobile robot navigation in a real environment in the presence of multiple robots.

Both robots are three-wheeled, with the rear two wheels being active and the front one passive. The rear wheels can move independently to achieve the desired navigation angle. The path was traced on the platform using a pencil during the robot's movement between points. Extensive experimentation involving more than twenty trials for each environment

was conducted with a maximum velocity of 0.06 m/s. The approach's robustness was further tested with various obstacle shapes and sizes. The results affirm the approach's suitability for real-time applications, showcasing minimal deviation when compared to the simulation environment. The experimental results demonstrate the robot's successful obstacle avoidance and timely attainment of its goal.

#### 4.3. Comparative Analysis of Experimental vs. Simulation

On comparison of real-time results with simulation results, it seems that the path developed by the proposed FFA controller safely avoids static and dynamic obstacles. The observed path length and navigational time tabulated in Tables 8–11 confirm the appropriateness of the developed approach, as the percentage of deviation between real-time results and simulation results is less than 4.5%. Hence, the goal of navigation was achieved successfully in an unknown environment. The developed approach deals with the single mobile robot system that effectively and efficiently handles the multiple robot systems for crowded environments. The velocity profile for the left and right wheels in meters per second is shown in Figure 13. For a detailed analysis of path length and navigational time for single- and multiple-robot navigation systems, 20 trials and 5 trials were undertaken, respectively.

**Table 8.** Path length comparison for a single mobile robot system.

| Sl. No.    | Experimental Path Length (cm) | Simulation Path Length (cm) | % of Error |
|------------|-------------------------------|-----------------------------|------------|
| Scenario-1 | 133.61 (Figure 8)             | 127.69 (Figure 3)           | 4.43       |
| Scenario-2 | 261.97 (Figure 9)             | 250.95 (Figure 4)           | 4.20       |

**Table 9.** Navigational time comparison for a single mobile robot system.

| Sl. No.    | Experimental Time during MRN (s) | Simulation Time during MRN (s) | % of Deviation |
|------------|----------------------------------|--------------------------------|----------------|
| Scenario-1 | 15 (Figure 8)                    | 14.2 (Figure 3)                | 4.40           |
| Scenario-2 | 27.6 (Figure 9)                  | 26.4 (Figure 4)                | 4.34           |

**Table 10.** Path length comparison for multiple mobile robot systems.

| Sl. No.    | Experimental Time during MRN (s) (Figure 10) | Simulation Time during MRN (s) (Figure 5) | % of Deviation |
|------------|--|---|----------------|
| Scenario-3 | Robot 1                                      | 133.12                                    | 4.22           |
|            | Robot 2                                      | 147.59                                    | 4.25           |
|            | Robot 3                                      | 163.11                                    | 4.17           |
|            | Robot 4                                      | 122.76                                    | 4.11           |

**Table 11.** Navigational time comparison for multiple mobile robot systems.

| Sl. No.    | Experimental Time during MRN (s) (Figure 10) | Simulation Time during MRN (s) (Figure 5) | % of Deviation |
|------------|--|---|----------------|
| Scenario-3 | Robot 1                                      | 13.60                                     | 4.26           |
|            | Robot 2                                      | 15.01                                     | 4.33           |
|            | Robot 3                                      | 16.48                                     | 4.24           |
|            | Robot 4                                      | 13.31                                     | 4.13           |

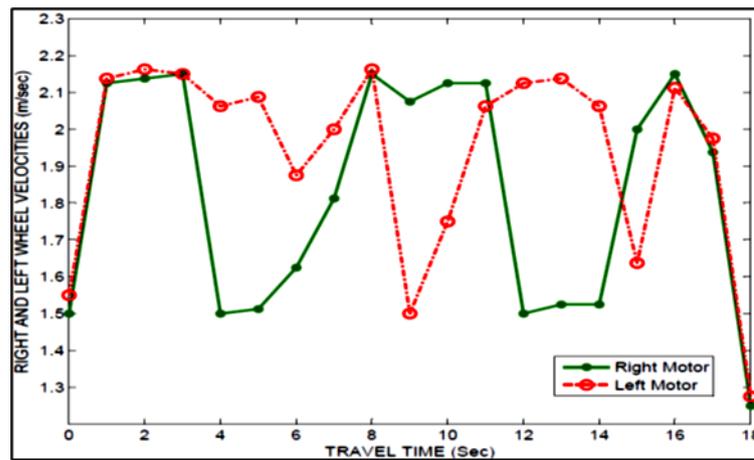


Figure 13. Left wheel velocity versus right wheel velocity profile in m/s.

4.4. Proposed FFA Controller versus Another Intelligent Controller over the Same Environmental Setup

To prove the effectiveness of the developed controller, it is necessary to check its performance with other artificially intelligent controllers over the same environmental condition; i.e., the number of obstacles, its position, and the robot position must be the same. The performance parameters considered are the path length and navigational time. For comparison, fuzzy logic (FL), particle swarm optimization (PSO), and genetic algorithm (GA) were considered. The optimized path was selected for each controller after performing more than 20 trials. While analyzing, all the positions of robots and obstacles in a static environment are shown in Figures 14 and 15, whereas Figures 16 and 17 deal with the navigation in a dynamic environment with a moving obstacle system and dynamic goal system, respectively. The movement of obstacle and goal in a dynamic environment are shown in Figures 18 and 19. The simultaneous comparison of path length and navigational time is tabulated in Tables 12–15, and from the data, it is clear that the path produced by using the proposed FFA controller in all terrain is short, and the required navigational time was greatly decreased as compared to FL, PSO, and GA.

Table 12. Path length and navigational time comparison for a single-robot system (Figures 16 and 20).

| Sl. No. | Name of Controllers | Simulation Path Length (cm) | Simulation Time (s) | Real-Time Path Length (cm) | Real-Time (s) |
|---------|---------------------|-----------------------------|---------------------|----------------------------|---------------|
| 1       | FL                  | 291.06                      | 30.93               | 307.23                     | 32.65         |
| 2       | PSO                 | 278.12                      | 29.55               | 297.528                    | 31.62         |
| 3       | GA                  | 270.03                      | 28.69               | 287.82                     | 30.58         |
| 4       | <b>FFA</b>          | <b>261.97</b>               | <b>27.84</b>        | <b>281.35</b>              | <b>29.90</b>  |

Table 13. Path length and navigational time comparison for multiple-robot system (Figures 17 and 21).

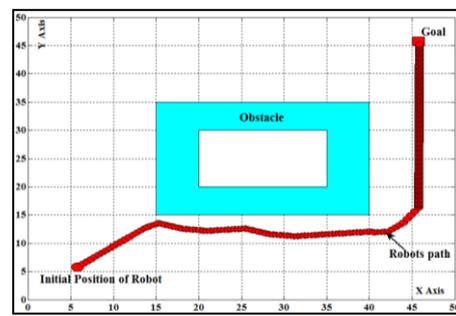
| Sl. No. | Name of Controllers | Simulation Path Length (cm) | Simulation Time (s) | Real-Time Path Length (cm) | Real-Time (s) |              |
|---------|---------------------|-----------------------------|---------------------|----------------------------|---------------|--------------|
| 1       | FL                  | Robot-1                     | 245.78              | 26.12                      | 252.2         | 26.802       |
|         |                     | Robot-2                     | 213.44              | 22.68                      | 226.38        | 24.05        |
| 2       | PSO                 | Robot-1                     | 239.08              | 25.40                      | 247.08        | 26.25        |
|         |                     | Robot-2                     | 210.50              | 22.37                      | 216.67        | 23.02        |
| 3       | GA                  | Robot-1                     | 229.61              | 24.40                      | 232.84        | 24.74        |
|         |                     | Robot-2                     | 207.21              | 22.02                      | 210.21        | 22.34        |
| 4       | <b>FFA</b>          | <b>Robot-1</b>              | <b>223.14</b>       | <b>23.71</b>               | <b>230.38</b> | <b>24.48</b> |
|         |                     | <b>Robot-2</b>              | <b>205.97</b>       | <b>21.89</b>               | <b>208.74</b> | <b>22.18</b> |

**Table 14.** Path length and navigational time comparison in the presence of moving obstacles (Figure 18).

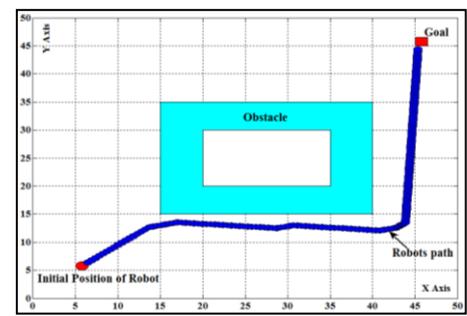
| Sl. No. | Name of Controllers | Simulation Path Length (cm) | Simulation Time (s) |
|---------|---------------------|-----------------------------|---------------------|
| 1       | FL                  | 244.55                      | 26                  |
| 2       | PSO                 | 237.69                      | 25.62               |
| 3       | GA                  | 232.84                      | 24.74               |
| 4       | FFA                 | 228.76                      | 23.87               |

**Table 15.** Path length and navigational time comparison in moving goal situations (Figure 19).

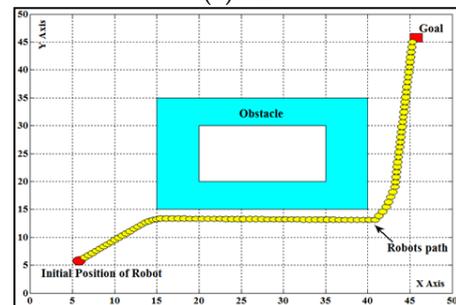
| Sl. No. | Name of Controllers | Simulation Path Length (cm) | Simulation Time (s) |
|---------|---------------------|-----------------------------|---------------------|
| 1       | FL                  | 161.70                      | 17.18               |
| 2       | PSO                 | 153.61                      | 16.32               |
| 3       | GA                  | 142.90                      | 15.18               |
| 4       | FFA                 | 139.29                      | 14.80               |



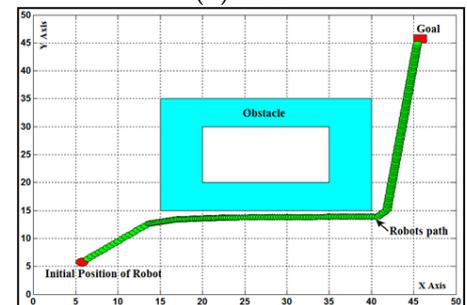
(a) FL



(b) PSO

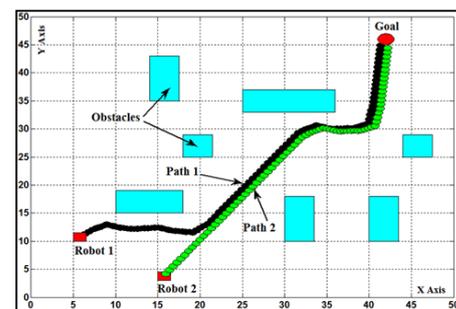


(c) GA

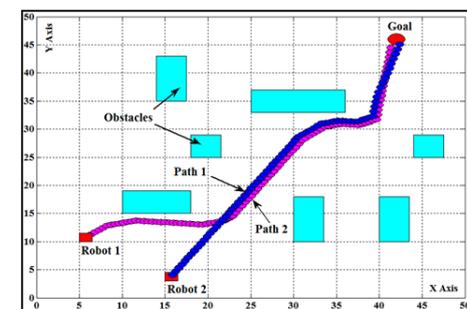


(d) FFA

**Figure 14.** Mobile robot (single) navigation in static simulation environment by various intelligent approaches.



(a) FL



(b) PSO

**Figure 15.** Cont.

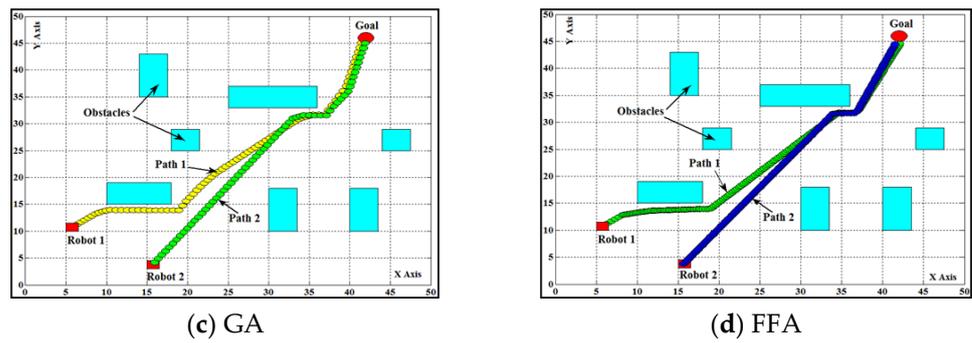


Figure 15. Multiple mobile robot navigation in a static simulation environment by various intelligent approaches.

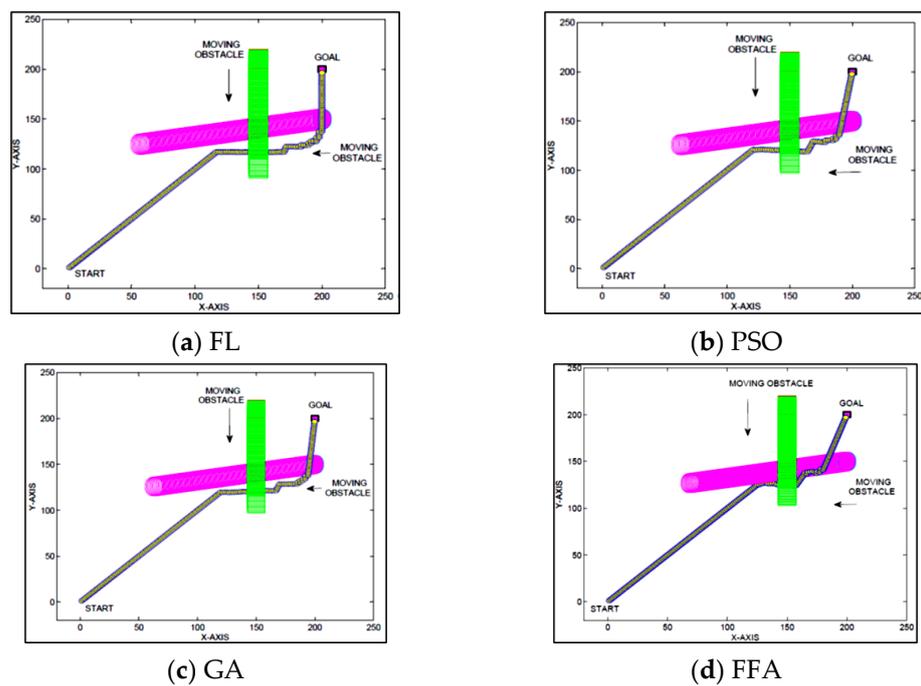


Figure 16. Navigation in a dynamic obstacle environment by various intelligent approaches.

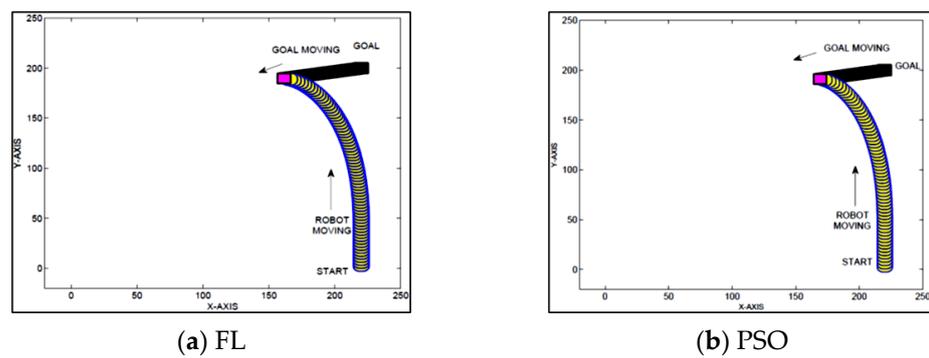


Figure 17. Cont.

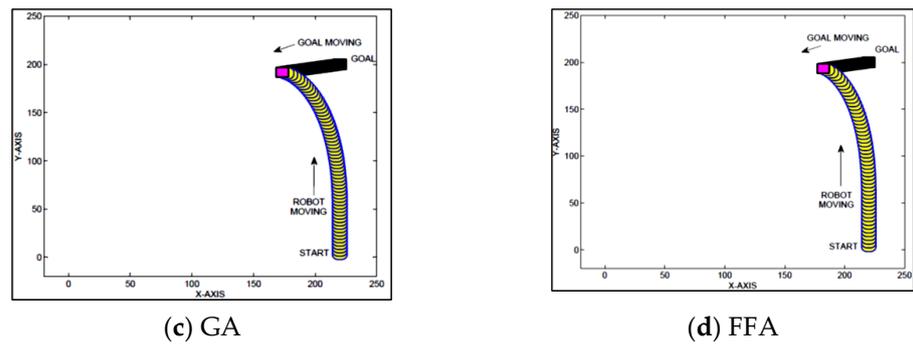


Figure 17. Navigation in a dynamic goal environment by various intelligent approaches.

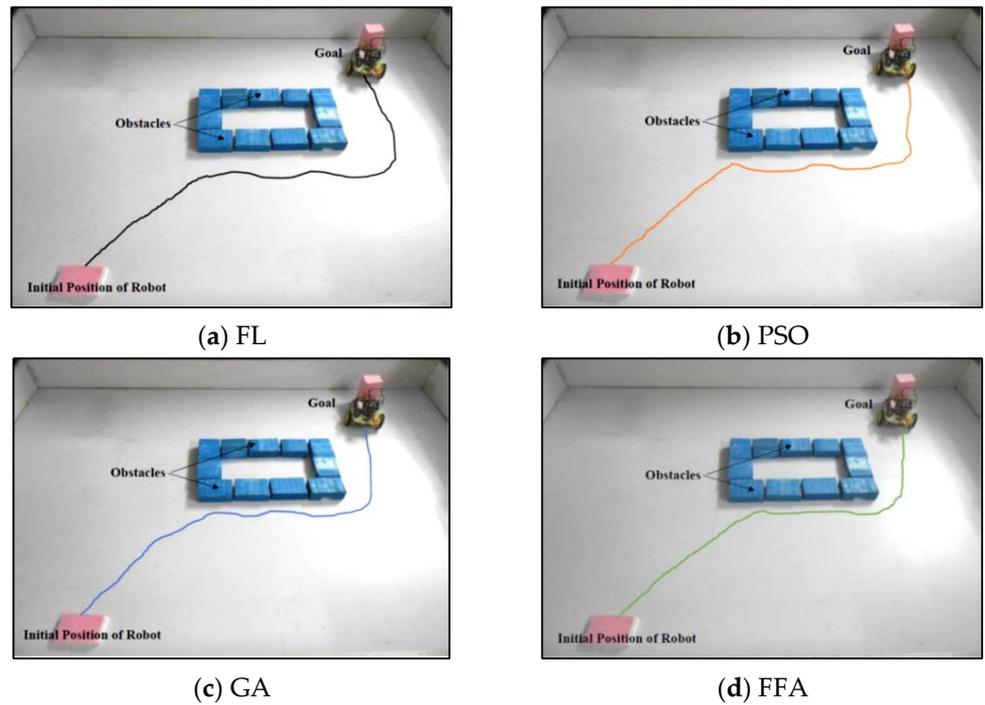


Figure 18. Mobile robot (single) navigation in the static, real-time environment by various intelligent approaches.

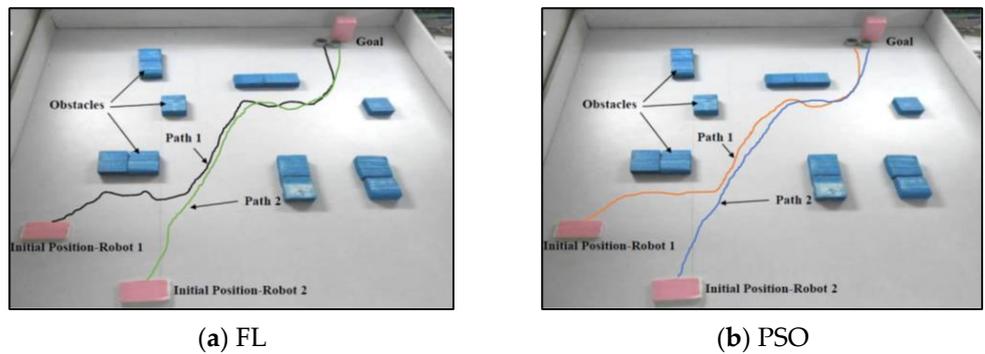


Figure 19. Cont.

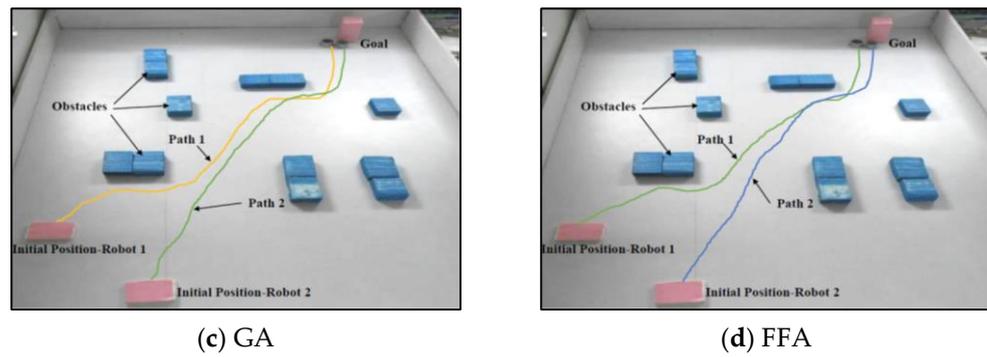


Figure 19. Multiple mobile robot navigation in the static, real-time environment by various intelligent approaches.

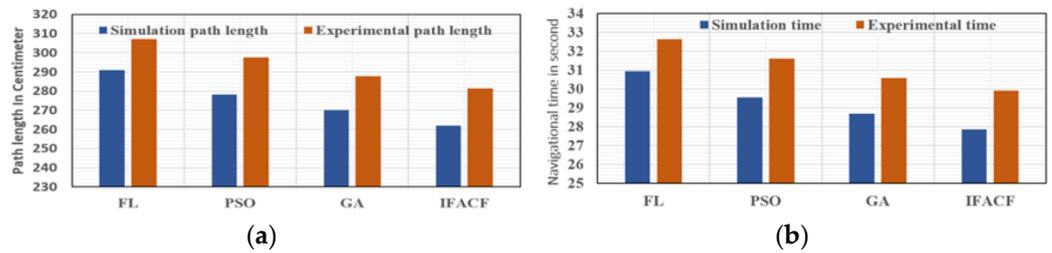


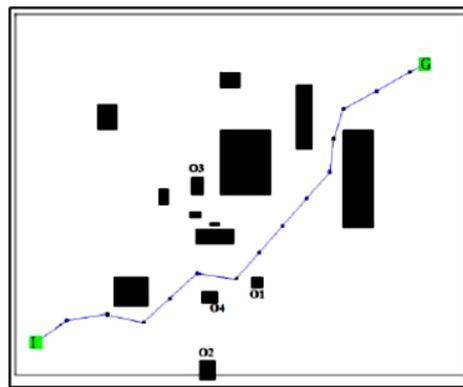
Figure 20. Moving goal environment: Comparison between FL, PSO, GA, and FFA. (a) Path length and (b) navigational time.

4.5. Proposed FFA Controller versus Published Work

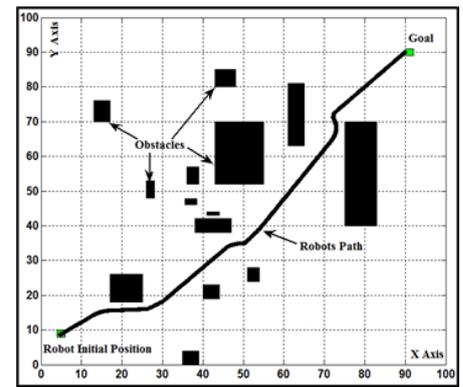
In this section, the work developed by the other researchers, i.e., Singh et al. [36], Montiel et al. [37], Zheng et al. [38], and Orozco-Rosas et al. [39], is considered for comparison with the proposed FFA controller. The comparison was carried out in the simulation environment only, and the performance was evaluated on the basis of path length. In [36], the navigation of robots was presented using the neural network (NN) in a complex, crowded environment where the obstacles had a rectangular shape (Figure 21a). The application of artificial potential field (APF) for the navigation of robots in a static environment was developed by [37] in the presence of three circular obstacles in a static environment (Figure 21c). Similarly, the navigation based on the Elman neural network (ENN) training technique in the presence of three rectangular obstacles (Figure 21e) and the parallel bacterial potential field algorithm (PBPFA) over a set of circular obstacles (Figure 21g) were presented by [38] and [39], respectively. Figure 21 demonstrates that the proposed FFA controller generates a smoother and shorter path than the respective AI controllers. Table 16 and Figure 22 show that there is a huge gap between the path developed by the proposed controller and other controllers, and the percentage of path length saved by using a proposed controller reached a maximum of 35.38% and a minimum of 5.7%.

Table 16. FFA versus other AI controllers.

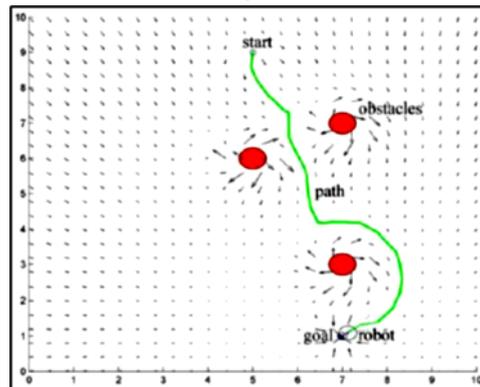
| S. N.      | Start Point | Goal Point | Path Length (cm) by Other AI Controllers | Path Length (cm) by FFA Controller | % of Path Length Saved by FFA |
|------------|-------------|------------|--|------------------------------------|-------------------------------|
| Scenario-4 | (4,8)       | (90,89)    | 7.9 (Figure 21a)                         | 6.6 (Figure 21b)                   | 16.45                         |
| Scenario-5 | (5,9)       | (7,1)      | 6.5 (Figure 21c)                         | 4.2 (Figure 21d)                   | 35.38                         |
| Scenario-6 | (1.7,1.5)   | (16.9,16)  | 5.2 (Figure 21e)                         | 4.9 (Figure 21f)                   | 5.7                           |
| Scenario-7 | (5,9)       | (5,1)      | 7.6 (Figure 21g)                         | 5.7 (Figure 21h)                   | 25                            |



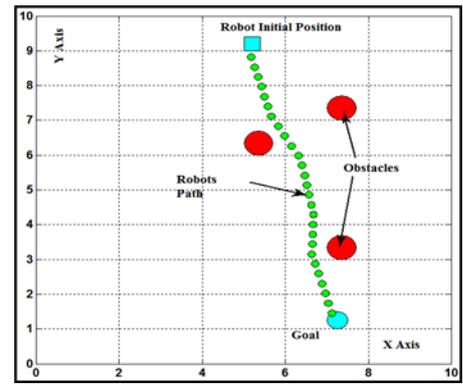
(a) Singh et al.



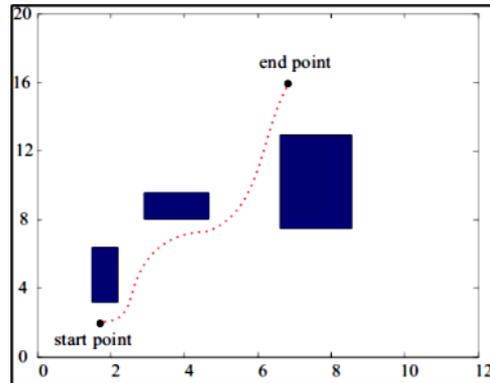
(b) FFA



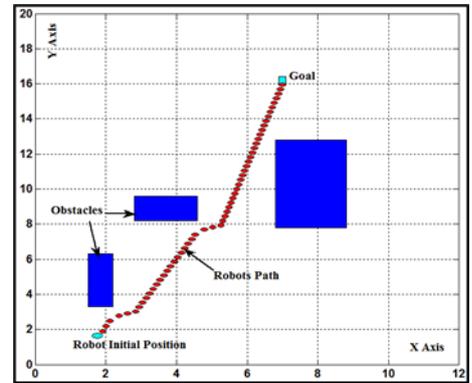
(c) Montiel et al.



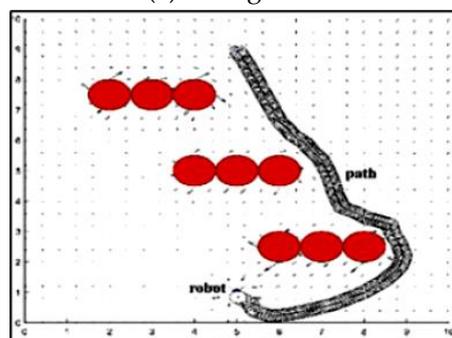
(d) FFA



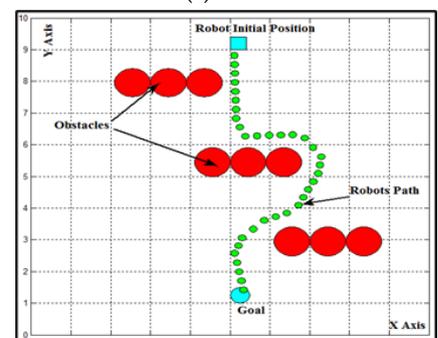
(e) Zheng et al.



(f) FFA

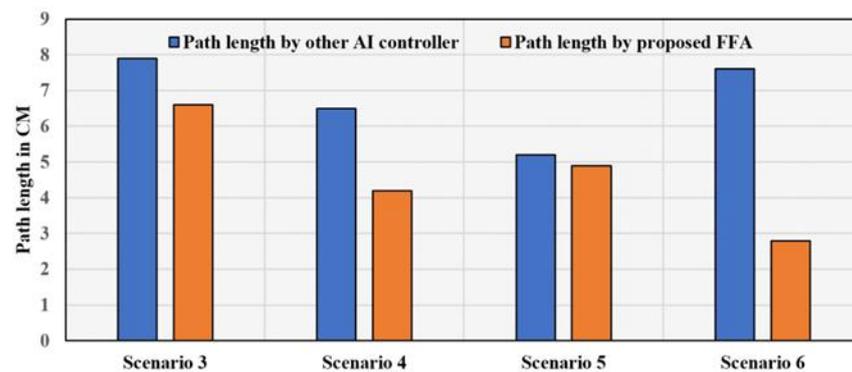


(g) Orozco-Rosas et al.



(h) FFA.

Figure 21. Other published work versus the proposed FFA controller [36–39].



**Figure 22.** Path length comparison between another AI controller and FFA.

## 5. Conclusions

The proposed functional firefly algorithm provides the smallest size (order) parameters for optimizing the mobile robot navigation. The FFA is based on the reduction order technique from  $n$ -interval to  $(n-m)$  interval by applying the choice function. The axiom of choice function reduces the order of domain, co-domain, and range of FFA to minimize the navigation. FFA is presented with a choice function for the navigation of a mobile robot in a completely unknown environment in the presence of a static obstacle, a moving obstacle, and a dynamic goal. The choice function identifies these limitations by the non-void family of the fireflies as a set. The movement of the fireflies is classified by fuzzy, and the probability function sets the optimization of the path. The developed controller achieved the goals of navigation, i.e., obstacle avoidance and path optimization in static and dynamic environments for single and multiple mobile robot systems. The obtained results show that the robot provides a smoother trajectory with a shorter path in less navigational time. The observed percentage of deviation between simulation and real-time results is less than 4.5%, and it yielded the optimal path length with minimum navigational time when compared to FL, PSO, and GA over similar environmental conditions. In comparison with the other researcher's work on NN, APF, ENN, and PBPFA, it saves a maximum path length of 36% and a minimum of 5%. In the future, the developed approach will aim to apply real-time navigation for on-road traffic conditions (real dynamic situations). The proposed controller can be tested by hybridizing with newly developed intelligent algorithms for developing a new path planner. It can be implemented in the development of an autonomous robot in an uncertain environment. The proposed approach can also be implemented to navigate aerial vehicles and underwater robots.

**Author Contributions:** Conceptualization, B.K.P. and A.J.; methodology, B.K.P. and S.K.K.; software, B.K.P. and B.P.; formal analysis, B.P. and B.K.P.; investigation, B.K.P. and B.P.; writing—original draft preparation, B.K.P. and A.J.; writing—review and editing, B.P.; visualization, S.K.K. and B.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

**Table A1.** Specification of the in-house robot used in the experiment.

| Elements |                                       | Technical Specification   |
|----------|---------------------------------------|---|
| 1        | Processor                             | ATmega2560 (Arduino Mega 2560, Arduino UNO, Olatus Systems, Guwahati, India)                    |
| 2        | RAM                                   | 8 KB, EEROM-4 KB  |
| 3        | Flash                                 | 256 KB (8 KB used for boot loader)  |
| 4        | Motors                                | 2-DC gear motors with incremental encoders  |
| 5        | Distance sensors                      | (a) Infrared sensors with up to 150 cm range<br>(b) Ultrasonic sensors with up to 400 cm range  |
| 6        | Speed                                 | Max: 0.47 m/s, Min: 0.03 m/s  |
| 7        | Power                                 | Power adapter or Rechargeable NiMH Battery (2000 mAh)   |
| 8        | Communication                         | USB connection to the computer  |
| 9        | Size                                  | Length: 25 cm, Width: 19 cm, Height: 12 cm  |
| 10       | Weight                                | Approx. 1100 g  |
| 11       | Payload                               | Approx. 4000 g  |
| 12       | Remote control Software via USB cable | C/C++17 <sup>®</sup> (on PC, MAC OS 12)<br>MATLAB R2021a <sup>®</sup> (on PC, MAC OS 12, Linux) |

**Table A2.** Specification of the Khepera-II robot used in the experiment.

| Elements |   | Technical Specification  |
|----------|---|--|
| 1        | Processor                                   | Motorola 68331 CPU, 25 MHz   |
| 2        | RAM   | 512 KB   |
| 3        | Flash                                       | 512 KB   |
| 4        | Motors                                      | 2-DC brushed Servo motors with incremental encoders  |
| 5        | Sensors                                     | 8 Infrared proximity and ambient light sensors with up to 100 mm range   |
| 6        | Speed                                       | Max: 0.5 m/s, Min: 0.02 m/s  |
| 7        | Power                                       | Power adapter or Rechargeable NiMH Batteries   |
| 8        | Communication                               | Standard Serial Port, up to 115 KB/S   |
| 9        | Size  | Diameter: 70 mm, Height: 30 mm   |
| 10       | Weight                                      | Approx. 80 g   |
| 11       | Payload                                     | Approx. 250 g  |
| 12       | Remote control software via tether or radio | LabVIEW <sup>®</sup> (on PC, MAC OS 12) using RS232<br>MATLAB <sup>®</sup> (on PC, MAC OS 12, Linux) using RS232<br>Sys Quake <sup>®</sup> (on PC, MAC OS 12, Linux) using RS232<br>Freeware Any other software capable of RS232 communication |

## References

- Patle, B.K.; Patel, B.; Jha, A. Rule-Based Fuzzy Decision Path Planning Approach for Mobile Robot. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCCUBEA), Pune, India, 16–18 August 2018; pp. 1–7.
- Dubey, V.; Patel, B.; Barde, S. Path Optimization and Obstacle Avoidance using Gradient Method with Potential Fields for Mobile Robot. In Proceedings of the 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 14–16 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1358–1364.
- Patle, B.K. Intelligent Navigational Strategies for Multiple Wheeled Mobile Robots Using Artificial Hybrid Methodologies. Ph.D. Thesis, National Institute of Technology, Rourkela, India, 2016.
- Patle, B.K.; Pandey, A.; Parhi DR, K.; Jagadeesh, A. A review: On path planning strategies for navigation of mobile robot. *Def. Technol.* **2019**, *15*, 582–606. [[CrossRef](#)]
- Kanoon, Z.E.; Al-Araji, A.S.; Abdullah, M.N. Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid Optimization Method. *Int. J. Intell. Eng. Syst.* **2022**, *15*, 161–175. [[CrossRef](#)]
- Wang, D.; Chen, S.; Zhang, Y.; Liu, L. Path planning of mobile robot in dynamic environment: Fuzzy artificial potential field and extensible neural network. *Artif. Life Robot.* **2021**, *26*, 129–139. [[CrossRef](#)]
- Wang, G.; Zhou, J. Dynamic robot path planning system using neural network. *J. Intell. Fuzzy Syst.* **2021**, *40*, 3055–3063. [[CrossRef](#)]
- Chen, Z.; Wu, H.; Chen, Y.; Cheng, L.; Zhang, B. Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm. *Appl. Soft Comput.* **2022**, *116*, 108192. [[CrossRef](#)]

9. Hou, W.; Xiong, Z.; Wang, C.; Chen, H. Enhanced ant colony algorithm with communication mechanism for mobile robot path planning. *Robot. Auton. Syst.* **2022**, *148*, 103949. [[CrossRef](#)]
10. Muni, M.K.; Parhi, D.R.; Kumar, P.B. Improved motion planning of humanoid robots using bacterial foraging optimization. *Robotica* **2021**, *39*, 123–136. [[CrossRef](#)]
11. Quan, Y.; Wei, W.; Ouyang, H.; Lan, X. New Harmony Search Algorithm for Mobile Robot Path Planning. In *Advances in Guidance, Navigation and Control*; Springer: Singapore, 2022; pp. 3463–3473.
12. Singh, S.; Sharma, K.; Doriya, R. Minimizing Computation Time for Robot Path Planning Using Improved Cuckoo Search Algorithm. In *Advances in Electrical and Computer Technologies*; Springer: Singapore, 2021; pp. 199–209.
13. Syama, R.; Mala, C. A Multi-objective Optimal Trajectory Planning for Autonomous Vehicles Using Dragonfly Algorithm. In Proceedings of the 3rd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing, Online, 18–19 December 2020; Springer: Cham, Switzerland, 2022; pp. 21–36.
14. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: London, UK, 2008.
15. Christensen, A.L.; O’Grady, R.; Dorigo, M. Synchronization and fault detection in autonomous robots. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 4139–4140.
16. Liaquat, S.; Fakhar, M.S.; Kashif SA, R.; Saleem, O. Statistical Analysis of Accelerated PSO, Firefly and Enhanced Firefly for Economic Dispatch Problem. In Proceedings of the 2021 6th International Conference on Renewable Energy: Generation and Applications (ICREGA), Al Ain, United Arab Emirates, 2–4 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 106–111.
17. Nalluri, M.R.; Kannan, K.; Roy, D.S. A Novel Discrete Firefly Algorithm for Constrained Multi-Objective Software Reliability Assessment of Digital Relay. In *Machine Vision Inspection Systems, Volume 2: Machine Learning-Based Approaches*; Wiley: Hoboken, NJ, USA, 2021; pp. 287–321.
18. Cheng, Z.; Song, H.; Chang, T.; Wang, J. An improved mixed-coded hybrid firefly algorithm for the mixed-discrete SSCGR problem. *Expert Syst. Appl.* **2022**, *188*, 116050. [[CrossRef](#)]
19. Xing, H.X.; Wu, H.; Chen, Y.; Wang, K. A cooperative interference resource allocation method based on improved firefly algorithm. *Def. Technol.* **2021**, *17*, 1352–1360. [[CrossRef](#)]
20. Sivaranjani, P.; Senthil Kumar, A. Hybrid Particle Swarm Optimization-Firefly algorithm (HPSOFF) for combinatorial optimization of non-slicing VLSI floorplanning. *J. Intell. Fuzzy Syst.* **2017**, *32*, 661–669. [[CrossRef](#)]
21. Dash, S.; Thulasiram, R.; Thulasiraman, P. Modified firefly algorithm with chaos theory for feature selection: A predictive model for medical data. *Int. J. Swarm Intell. Res.* **2019**, *10*, 1–20. [[CrossRef](#)]
22. Feng, Y.; Wang, G.G.; Wang, L. Solving randomized time-varying knapsack problems by a novel global firefly algorithm. *Eng. Comput.* **2018**, *34*, 621–635. [[CrossRef](#)]
23. Liu, C.; Gao, Z.; Zhao, W. A new path planning method based on firefly algorithm. In Proceedings of the 2012 Fifth International Joint Conference on Computational Sciences and Optimization, Harbin, China, 23–26 June 2012; IEEE: Piscataway, NJ, USA, 2012.
24. Hidalgo-Paniagua, A.; Vega-Rodríguez, M.A.; Ferruz, J.; Pavón, N. Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach. *Soft Comput.* **2017**, *21*, 949–964. [[CrossRef](#)]
25. Michael, B.; Yu, X.-H. Autonomous robot path optimization using firefly algorithm. In Proceedings of the 2013 International Conference on Machine Learning and Cybernetics, Tianjin, China, 14–17 July 2013; IEEE: Piscataway, NJ, USA, 2013; Volume 3.
26. Patle, B.K.; Pandey, A.; Jagadeesh, A.; Parhi, D.R. Path planning in uncertain environment by using firefly algorithm. *Def. Technol.* **2018**, *14*, 691–701. [[CrossRef](#)]
27. Kim, H.C.; Kim, J.S.; Ji, Y.K.; Park, J.H. Path planning of swarm mobile robots using firefly algorithm. *J. Inst. Control. Robot. Syst.* **2013**, *19*, 435–441. [[CrossRef](#)]
28. Patle, B.K.; Parhi, D.R.; Jagadeesh, A.; Kashyap, S.K. On firefly algorithm: Optimization and application in mobile robot navigation. *World J. Eng.* **2017**, *14*, 65–76. [[CrossRef](#)]
29. Patle, B.K.; Parhi, D.; Jagadeesh, A.; Sahu, O.P. Real Time Navigation Approach for Mobile Robot. *J. Comput.* **2017**, *12*, 135–142. [[CrossRef](#)]
30. Wang, G.; Guo, L.; Hong, D.; Duan, H.; Liu, L.; Wang, H. A modified firefly algorithm for UCAV path planning. *Int. J. Hybrid Inf. Technol.* **2012**, *5*, 123–144.
31. Sutantyo, D.; Levi, P. Decentralized underwater multi robot communication using bio-inspired approaches. *Artif. Life Robot.* **2015**, *20*, 152–158. [[CrossRef](#)]
32. Sadhu, A.K.; Konar, A.; Bhattacharjee, T.; Das, S. Synergism of Firefly Algorithm and Q-Learning for Robot Arm Path Planning. *Swarm Evol. Comput.* **2018**, *43*, 50–68. [[CrossRef](#)]
33. Panda, M.R.; Dutta, S.; Pradhan, S. Hybridizing invasive weed optimization with firefly algorithm for multi-robot motion planning. *Arab. J. Sci. Eng.* **2018**, *43*, 4029–4039. [[CrossRef](#)]
34. Huang, H.C.; Lin, S.K. A Hybrid Metaheuristic Embedded System for Intelligent Vehicles Using Hypermutated Firefly Algorithm Optimized Radial Basis Function Neural Network. *IEEE Trans. Ind. Inform.* **2018**, *15*, 1062–1069. [[CrossRef](#)]
35. Duan, P.; Sang, H.; Li, J.; Han, Y.; Sun, Q. Solving multi-objective path planning for service robot by a pareto-based optimization algorithm. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; IEEE: Piscataway, NJ, USA, 2018.
36. Singh, N.H.; Thongam, K. Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. *Intell. Serv. Robot.* **2019**, *12*, 55–67. [[CrossRef](#)]

37. Montiel, O.; Sepúlveda, R.; Orozco-Rosas, U. Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field. *J. Intell. Robot. Syst.* **2015**, *79*, 237–257. [[CrossRef](#)]
38. Zheng, W.; Wang, H.; Zhang, Z.; Lu, X. Hybrid position/virtual-force control for obstacle avoidance of wheeled robots using Elman neural network training technique. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417710460. [[CrossRef](#)]
39. Orozco-Rosas, U.; Montiel, O.; Sepúlveda, R. Parallel Bacterial Potential Field Algorithm for Path Planning in Mobile Robots: A GPU Implementation. In *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*; Springer: Cham, Switzerland, 2018; pp. 207–222.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.