Path Planning of an Autonomous Mobile Robot in a Dynamic Environment using Modified Bat Swarm Optimization

Ibraheem Kasim Ibraheem*, Fatin Hassan Ajdilb, Rami A. Maherc

*Electrical Engineering Department, College of Engineering, University of Baghdad, Al-Jadriyah, P.O.B.: 47273, 10001 Baghdad, Iraq.
bElectrical Engineering Department, College of Engineering, University of Baghdad, Al-Jadriyah, 10001 Baghdad, Iraq.
cElectrical Engineering Department, College of Engineering, Isra University, Amman, Jordan.

ARTICLE INFO

Keywords: Autonomous navigation, path planning, obstacle avoidance, dynamic environment, Bat algorithm, mobile robot.

ABSTRACT

This paper outlines a modification on the Bat Algorithm (BA), a kind of swarm optimization algorithms with for the mobile robot navigation problem in a dynamic environment. The main objectives of this work are to obtain the collision-free, shortest, and safest path between starting point and end point assuming a dynamic environment with moving obstacles. A new modification on the frequency parameter of the standard BA has been proposed in this work, namely, the Modified Frequency Bat Algorithm (MFBA). The path planning problem for the mobile robot in a dynamic environment is carried out using the proposed MFBA. The path planning is achieved in two modes; the first mode is called path generation and is implemented using the MFBA, this mode is enabled when no obstacles near the mobile robot exist. When an obstacle close to the mobile robot is detected, the second mode, i.e., the obstacle avoidance (OA) is initiated. Simulation experiments have been conducted to check the validity and the efficiency of the suggested MFBA based path planning algorithm by comparing its performance with that of the standard BA. The simulation results showed that the MFBA outperforms the standard BA by planning a collision-free path with shorter, safer, and smoother than the path obtained by its BA counterpart.

1. Introduction

Mobile robot navigation is a challenging problem in the robotics field and numerous studies have been developed resulting in a considerable number of solutions [1]. The term navigation refers to the guidance of the mobile robot from the starting position to the target position avoiding collisions and unsafe conditions [2]. It consists of three major steps, self-localization, path planning, and map building. Three main anxieties concerning robot navigation problems are efficiency, safety, and accuracy [3]. Path planning is a vital step of the robot control and navigation of the mobile robots. The computation time of algorithms falls into two groups: P-time (Polynomial time) and NP-time (Nondeterministic Polynomial time), whereas in NP-time, the computation time required to solve such problems arises intensely when the problem dimension increases. Furthermore, NP-time is classified into NP-Hard, where problems are at least as hard to solve as any problem in NP, and NP-complete, where problems which are NP-hard and belong to NP; path planning is categorized as an NP-complete problem [4]. The research of path planning began in the late 60’s of the past century, and several algorithms have been proposed. These comprise the roadmap method [5], cell decomposition [6], Potential fields [7], and mathematical programming [8], etc., just mentioning a few. Lately, it is discovered that these algorithms are either ineffective, because of the significant computational cost; or imprecise, because of getting stuck in local minimum. To outdo these drawbacks, several heuristic based methods have been implemented, such as the application of artificial neural networks, particle swarm optimization (PSO), genetic algorithm (GA) [9], and hybridization between them [10]. One of the main benefits of heuristic based methods is that it can yield satisfactory results quickly, which is particularly appropriate to solve NP-complete problems.

The path planning is separated into two main fields, global and local path planning. On the first hand, in path planning with local path planning, the calculations of the path are achieved when the mobile robot is in motion; that means, the calculation is fit for generating new paths as the environment changes. On the other hand, with global path planning, the environment should be totally recognized and identified, while the terrain must be static [11,12].

Many types of research have been studied path planning problem in dynamic environments, authors of [13] proposed a new method to decide the optimum route of the mobile robot in an unknown dynamic environment by the ant colony optimization (ACO) algorithm, they used ACO algorithm to decide the optimal rule table of the system. The angle variance to the nearest obstacle and the distance between the starting and end positions are the factors that affected the fuzzy decision-making process. The work in [14] presented the navigation approach for ground vehicle in a dynamic environment by using two types of fuzzy logic controllers. The first controller called the Target Reaching Fuzzy Inference Controller (TR-FIC) to guarantee to have the robot reaches the goal (by means of the angle difference between vehicle heads and target). The second controller is an Obstacle Avoidance Fuzzy Inference Control (OA-FIC). The switching between these controllers is done by obstacle sensing bit from the environment. While in [15] researchers introduced a new method to deal with the static and dynamic obstacles, if the robot encounters a static obstacle, it avoids the obstacle by using a fuzzy logic controller. While in the case of dynamic obstacles, the mobile robot estimates the direction and the velocity of all the moving obstacles and generates a corresponding trajectory prediction table of each obstacle for estimating the obstacle’s future trajectory. In [16], people proposed a new fitness function taking into account the distances between robot-obstacles,
robot, robot-goal. This work deals with obtaining the optimal path for multiple mobile robots using PSO algorithm. In [17] obstacle avoidance for wheeled mobile robot in a dynamic environment is achieved by considering a virtual disc in front of the mobile robot centered at the head angle of the mobile robot, then computing the intersection angle between the disc and the obstacles and changing the direction of the mobile robot according to specific rules. The fuzzy logic system and laser scan sensor were implemented in [18] for tuning mobile robot velocities in an unknown dynamic environment, the noise of the laser sensor has been eliminated with a suitable filter. The movement of the dynamic obstacles has been predicted using artificial neural networks (ANN) and radial basis functions neural networks (RBANN) to solve the problem of the motion planning in [19]. While in [20] a new method has been suggested for tackling the motion planning problem in a dynamic environment using the synergism of neural network, and fuzzy system. This work combined Artificial Potential Field (APF) and Fuzzy Neural Network (FNN) for globally optimized path planning on a dynamic environment in a real-time fashion. This technique used Fuzzy Logic to construct the space model, then integrated APF and FNN exploring for the shortest route while efficiently evades moving obstacles. The authors of [21] implemented a PSO algorithm for dynamic obstacle avoidance and target reaching by adding a penalty function to the fitness function to avoid obstacles. Grid-based methods for mobile robot path planning can be found in [22]. It should be remembered that mobile robot navigation including the path planning can be considered as the upper layer for the motion planning of the mobile robot through which the mobile receive data and react to its environment. This layer is built on the lower layer, namely, the motion control layer which operates the actuators of the mobile robot in response to the upper layer. Motion control layer can be designed using one of the linear or nonlinear control design methods [23–29].

The objective of this paper is to find the collision-free and shortest path (if it exists) from an initial position to an end position in a dynamic environment with moving obstacles using the proposed Modified Frequency Bat Algorithm (MFBA). This proposed algorithm is integrated with a novel local search technique to detect dynamic obstacles near the mobile robot using sensory vector information. Then proposing a new collision-free position for the mobile robot for the mobile robot to resume its trip to its goal position using obstacle avoidance procedure based on gap vector principle.

The rest of this paper is organized as follows. Section 2 presents the problem statement and assumptions. The swarm based Navigation algorithm is introduced in section 3. While section 4 suggests the obstacle detection and avoidance algorithm. The simulations and the results are discussed in section 5. Finally, the conclusions are given in Section 6.

2. Problem Statement and Assumptions

Assuming a 2-D world frame as shown in Fig.1. Initially, the mobile robot is at point (SP) and has to reach the point (GP). There are dynamic obstacles in the environment. The objective is to find the shortest route from SP to GP without colliding with any of the dynamic obstacles in the workspace by finding the best and feasible next position for the mobile robot from the current one. Before discussing and suggesting the solution to this problem, there are some assumptions made in this work:

1. The obstacles are represented by a circular shape. While the mobile robot is considered as a point in the free space.
2. No kinematic constraints affect the motion of the mobile robot. The only effect source is the motion of the obstacles.
3. The speed of the obstacles is different.
4. The mobile robot movement is omnidirectional at any time.
5. All the obstacles have the same size.
6. The shortest distance is obtained by minimizing the distance function \(f(x, y)\),

\[
f(x, y) = \sqrt{(x_{i+1} - x_1)^2 + (y_{i+1} - y_1)^2}
\]

where \(x_{i+1}, y_{i+1}\) represent the next position, \(x_1, y_1\) represent the current position.

3. SWARM BASED NAVIGATION ALGORITHM

3.1. Bat Algorithm (BA)

The BA is a bio-inspired algorithm developed by Yang in 2010, it is based on the echolocation or biosonar characteristics of micro-bats [30,31]. Echolocation is an important feature of the bat behavior, which means the bats emit a sound pulse and listen to the echo bouncing back from the obstacles while flying. By utilizing the time difference between its ears, the loudness of the response, and the delay time, it can figure up the velocity, the shape, and the size of the prey. In addition, the bat has the capability to change the way it works, if it sends the sound pulses with a high rate, they won't fly longer but give thorough details about its nearby surroundings which help bats to distinguish the prey position exactly. Another characteristic of bat's echolocation is its loudness; when the bats are near from the victim, it transmits sound pulsations silently while amid the searching process they send noisy sound pulses. Bats hunting methodology can be summarized in the next:

- All the bats utilize the echolocation to detect the distance. In addition, they distinguish the distinction between nutrient/prey and background obstacles in some supernatural way.
- Bats fly at random with speed \(v(t)\) at posture \(x_i\) with a frequency \(f_{min}\), changing wavelength \(\lambda\) and loudness \(A_i\) to scan for the prey. They can consequently change the wavelength (or frequency) of their transmitted pulsations and alter the rate of pulses transmission \(r \in [0, 1]\), contingent upon the nearness of their objective.
- Even though the loudness can change in many ways, we presume that the loudness decreases from a large (positive) \(A_i\) to a minimum constant value \(A_{min}\).

1) The Movement of Artificial Bats

In a D-dimensional searching or solution space, each bat is related with a velocity \(v(t)\) and a location \(x(t)\) at iteration \(t\). Amid all the bats, there exists a current best solution. As a result, the three guidelines aforementioned above can be converted into the updating equations for the positions and velocities. Since the frequency \(f(t)\) controls the range and the pace of the movement, the updating procedure of bats' positions/solutions is as follows:

\[
f(t) = f_{min} + (f_{max} - f_{min}) * \beta
\]

\[
v(t) = v(t-1) + (x(t-1) - x) * f(t)
\]

\[
x(t) = x(t-1) + v(t)
\]
where $\beta \in [0, 1]$ is a random vector of a uniform distribution. Here $x^*$ is the present global best location (solution), which is found after comparing all the solutions among all the $n$ bats. For the locally searching stage, once a solution is chosen from the best current solutions, a new solution for each bat is locally produced using the random walk principle,

$$x_{\text{new}} = x_{\text{old}} + \alpha A(t)$$  \hspace{1cm} (6)

where $\alpha \in [-1, 1]$ is a random number and represents the direction and intensity of random walk, $A(t)$ is the average loudness of all the bats at iteration step $t$. From the practical point of view, it is better to provide a scaling parameter $\sigma$ to control the step size.

2) Loudness and Pulse Emission

The loudness $A_i$ and the rate of pulse emission $r_i$ have to be updated accordingly as the iterations continue. Typically, the loudness $A_i$ decreases once a bat has detected its prey, whereas the rate of pulse emission $r_i$ rises according to the following equations:

$$A_i(t + 1) = \alpha A_i(t)$$  \hspace{1cm} (7)

$$r_i(t + 1) = r_i(0) [1 - \exp(-\gamma t)]$$  \hspace{1cm} (8)

where $\alpha$ and $\gamma$ are constants. For any $0 < \alpha < 1$ and $\gamma > 0$. Notwithstanding the success of the BA and its diverse fields and applications, there are still a few fundamental issues that require more investigation as described next.

3.2. Modified Frequency Bat Algorithm (BA)

There are two significant divisions in current metaheuristics: exploration and exploitation. Exploration is the inspection ability of mysterious different spaces to sense the global optimal target, while exploitation points to finding the optimal target by utilizing the past best fitness's information. A decent reply of a metaheuristics algorithm relies upon good coordination of these modules. In the case of little Exploration with escalated exploitation, the procedure could be stuck into local optima [32]. While a considerable Exploration with little exploitation could bring the algorithm to converge gradually and reduces the overall searching performance. The BA is an effective optimization algorithm in “exploitation” (i.e., local search), but at certain times it may get stuck at local optima and consequently it cannot accomplish the global search efficiently. For BA, the searching depends totally on “random walks”, so a rapid convergence cannot be secured [33].

Since the frequency controls the pace and the range of the Bat movement, the sound pulses with high frequency will not travel longer and vice versa. In addition, the BA loses its exploration capability when through time. Therefore, we suggest a method for balance between exploration and exploitation using frequency tuning by assuming that the algorithm starts with low frequency to increase the global search capability and the frequency increases gradually as iterations proceed. This can be achieved by considering the factor $\beta$ as a function, instead of being a random number. In this way, it guarantees that the frequency will increase through iterations. The factor $\beta$ in (3) will be modified as follows:

$$\beta = t \times e^{(-\rho + r)}$$  \hspace{1cm} (9)

where $t$ is iteration number, $r$ is a random number (0,1), the value of $\rho$ is an application dependent, for path planning problem the value of $\rho$ is chosen to be (0.01) as shown in the Fig. 2.

4. OBSTACLE DETECTION AND AVOIDANCE: LOCAL SEARCH

The mobile robot navigates from its SP to GP using BA or MFBA until it detects an obstacle, then the mobile robot switches to local search mode. The local search is implemented by surrounding the mobile robot by twelve virtual sensors as shown in Fig. 3.

**Fig. 2. Increasing frequency parameter of BA with iterations and different values of $\rho$.**

Since the mobile robot is a physical body, the obstacles are enlarged by the dimension of the mobile robot and considering the robot as a particle during the simulations as depicted in Fig. 4.

**Fig. 3. Mobile robot sensors deployment.**

**Fig. 4. Enlarging obstacles size corresponding to mobile robot size.**
4.1. Obstacles Detection

The obstacles are detected by using sensory vector (binary vector provides information about the obstacle, where logic (1) represents the existence of the obstacle while logic(0) represent free space within the sensor range). The number of sensors is twelve each with a sensor range (SR) of 0.8 m and the angle range of 30°. For each obstacle inside a certain sensor range (i.e. in the range between S) draw the tangent lines to the expanded obstacle (dotted circles in Fig. 5). Then compute the angle between the mobile robot point and the obstacle tangent lines that belong to a certain sensor range. For example, consider Fig. 5, in this case, the sensory vector $V_s$ is given as:

$$V_s = \{1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \}$$

4.2. Obstacles avoidance

The obstacles avoidance is done by using gap vector concept. Gap vector $V_g$, which is a binary vector, where logic(1) represent the occupancy gap, while logic(0) represent a free gap. The length of the vector is given as:

$$\text{length} (V_g) = \text{length} (V_s)$$

The mobile robot chooses the nearest gap to the goal. The gap vector $V_g$ can be derived from the sensory vector $V_s$ (each consecutive zeros represent free gap (logic(0)), otherwise (logic(1)) which is equivalent to OR gate as shown in Table 1.

| $V_s(i)$ | $V_s(M|H)$ | $V_g(i)$ |
|----------|------------|----------|
| 0        | 0          | 0        |
| 0        | 1          | 1        |
| 1        | 0          | 1        |
| 1        | 1          | 1        |

From the previous example (Fig. 5) we get,

$$V_s = \{1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \}$$

$$V_g = \{1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \}$$

The proposed path planning algorithm can be clarified in the pseudo-code listed in Algorithm 1.

Algorithm 1: Pseudo code for Proposed modified BA

```plaintext
while (mobile robot position is not equal to GP)
    collect data from virtual twelve sensors
    if detected obstacles within SR
        switch to local search mode (OA mode)
    else
        navigate toward goal using MFBA of (3)-(9)
end while
```

5. Simulation Results

The simulations of the proposed MFBA is divided into two parts. The first part includes the simulation of the test of the MFBA on benchmark functions. While the second one is devoted to the simulations for the path planning of the mobile robot on a dynamic environment.

5.1. Standard Benchmark Functions

The assessment of validation, efficiency, and reliability of a certain optimization algorithm is normally implemented by utilizing a set of basic standard benchmarks or test functions. Then the performance of such an algorithm is approved by comparing it with its counterparts after utilizing these test benchmark functions in the calculations. Among the most popular ones are given in Table 2. The proposed MFBA has been tested with 15 runs for each benchmark function. In order to analyze the performance of both algorithms, the “mean” values in Table 3 are considered. The sign “+” indicates that MFBA is better than BA, “-” means that the two approaches relatively give equal results, and “.” Represents MFBA is worse than BA.

| Fun Name       | Formula                                      | C     | $F_{min}$ | Search range   |
|----------------|----------------------------------------------|-------|-----------|----------------|
| F1: Sphere     | $\sum_{i=1}^{d} x_i^2 \cos(x_i) \cos(x_j)$ | U     | 0         | [-5.12, 5.12]  |
| F2: Easom      | $\exp(-x_1^2/2 - x_2^2/2) \sin(x_1) \cos(x_2)$ | U     | -1        | [-100, 100]    |
| F3: Hump Camel | $2x_1^2 - 1.051x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$ | M     | 0         | [-5.5]         |
| F4: Booth      | $(x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$   | U     | 0         | [-10, 10]      |
| F5: Rastrigin  | $\exp(-x_1^2/2 - x_2^2/2) \sin(x_1) \cos(x_2)$ | M     | 0         | [-5.12, 5.12]  |
| F6: Michalewicz| $\sin(x_1) \sin(2\cdot x_2 \cos(x_1))$     | M     | -1.8013   | [0, n]         |

Where $C$ characteristic, $U$: unimodal, $M$: multimodal, $d$: dimension, $m$: steepness parameter =10.
Table 3. The comparison performance of the BA and MFBA on benchmark functions

| Fun No | Alg.  | Best     | Worst   | Mean   | SD    | significant |
|--------|-------|----------|---------|--------|-------|-------------|
| F1     | BA    | 0.00028437 | 0.048626 | 0.0146 | 0.0145 | +           |
|        | MFBA  | 0.00011976 | 0.0285  | 0.0048 | 0.0078 | -           |
| F2     | BA    | -0.8286   | -9.64e-32 | -0.1241 | 0.2476 | +           |
|        | MFBA  | -0.8691   | 0       | -0.2090 | 0.3132 | -           |
| F3     | BA    | 1.1407e-05 | 0.3050  | 0.0398 | 0.0857 | +           |
|        | MFBA  | 1.1983e-05 | 0.3034  | 0.0549 | 0.1061 | -           |
| F4     | BA    | 1.7676e-04 | 0.4544  | 0.1256 | 0.1533 | -           |
|        | MFBA  | 2.7530e-05 | 1.1854  | 0.1629 | 0.3005 | -           |
| F5     | BA    | 0.0121    | 5.6793  | 2.4541 | 1.8399 | -           |
|        | MFBA  | 0.0642    | 5.3864  | 2.0278 | 1.3476 | +           |
| F6     | BA    | -1.9880   | -1.5022 | -1.7743 | 0.1008 | -           |
|        | MFBA  | -1.9695   | -1.7993 | -1.8762 | 0.0617 | -           |

5.2. Simulation Results on Mobile Robot Path Planning

The simulation parameters for all case studies are population size $= 5$, $A(0)=1$, $r(0)=0.5$, $\alpha=0.98$, $\gamma=0.8$, $f_{\min}=0$, $f_{\max}=10$, $\sigma=0.3$, $SR=0.8$, the starting point for mobile robot is $SP=(0, 0)$, and goal is $GP=(12, 12)$. The simulations have been run under MATLAB Environment on a computer system with 2.76 GHz Core i7 CPU, and 4G RAM. In these case studies, the solutions (shortest paths) are obtained after executing the proposed algorithm ten times in order to find out the best path. In all the simulations, the optimal path from start to target points without any obstacles is equal to 16.9705. The best path is the one nearest to the optimal path.

- Case Study 1: Three dynamic obstacles.

In this case study, the positions, velocities, and directions of the obstacles are listed in Table 4 below:

Table 4. Characterization of the obstacles motion

| Obs | center | radius | Vobs(m/s) | Theta(deg) |
|-----|--------|--------|-----------|------------|
| 1   | (1, 4.5) | 0.3    | 0.3       | 0°         |
| 2   | (10.5, 6) | 0.3    | 0.2       | 180°       |
| 3   | (6, 12)  | 0.3    | 0.15      | 270°       |

The mobile robot navigates from (0, 0) as a start point using MFBA, the magenta circle around the mobile robot in Fig. 6 (a) represents the sensing region. When the moving obstacles being sensed in this region, the path planning algorithm is switched into the local search mode and consequently triggers OA procedure as shown in Fig. 7 (b). Then the mobile robot continues its searching process until it reaches the goal as shown in Fig. 6 (c, d). The best path with the shortest distance and shortest time using MFBA algorithm was achieved in experiment no. eight among the ten runs. The total distance was equal to 17.0925m with a run time equal to 8.151829 sec. While the best path with the shortest distance using BA algorithm was achieved in experiment no. 8 too. The total distance was equal to 17.0932m with a run time equals 7.905828 sec.
Another comparison is made by summarizing both algorithms results after executing the program ten times. The MFBA obtained the largest fitness as the smallest standard deviation achieved by BA, the results are tabulated in Table 5.

|          | Standard BA | MFBA       |
|----------|-------------|------------|
| minimums | 0.056170939 | 0.05830325 |
| maximum  | 0.058502796 | 0.0585072461 |
| standard deviation | 0.00103782 | 0.05315157 |
| mean     | 0.05751639825 | 0.05844756632 |

- Case Study 2: Five dynamic obstacles

In this case study, the positions, velocities, and directions of obstacles are listed in Table 6 below.

| Obs | center | radius | $v_{dist}$ (m/s) | theta(deg) |
|-----|--------|--------|-----------------|------------|
| 1   | (4,2)  | 0.3    | 0.3             | 111.8°     |
| 2   | (3.7)  | 0.3    | 0.2             | 315°       |
| 3   | (9,4)  | 0.3    | 0.2             | 126.8°     |
| 4   | (7,9)  | 0.3    | 0.25            | 315°       |
| 5   | (11.2,7) | 0.3  | 0.22            | 150°       |

The mobile robot navigates in between five dynamic obstacles with different velocities as shown in Fig. 7, where the red dotted lines refer to the direction of the obstacle’s movement. It’s initially moving toward GP using MFBA until two of the obstacles (the first is shown in Fig. 7 (b), while the second is shown in Fig. 7 (c) enter the sensing region. At that time, the local search mode has been enabled to avoid obstacles as shown in Fig. 7 (b, c), the mobile robot returns to its normal mode (i.e., path generation using MFBA or BA) as shown in Fig. 7 (d, e). The best path with the shortest distance and shortest time using MFBA algorithm was achieved in experiment no. five among the ten runs. The total distance was equal to 18.3533 m with run time 9.0903 sec. While the best path using BA algorithm was achieved in experiment no. 8, which was equal to 18.6239 m with a run time equal to 9.10554 sec.
6. Conclusion
In this paper, the navigation approach for the mobile robot in a dynamic environment has been introduced using MFBA optimization combined with the local search technique and obstacle avoidance. The modified algorithm allowed the mobile robot to move from its start location to destination without colliding any of the available moving obstacles in the dynamic environment. The modification of the frequency parameter in the standard BA guaranteed that the searching was more directed and faster. The simulations on the test benchmark functions together with that on the path planning of a mobile robot proved the validity and the effectiveness of the proposed MFBA as compared to the standard BA. It can be concluded that the proposed MFBA based path planning achieves the optimal goal with the smoothest, shortest and less time path as compared to its BA based path planning counterpart.

Table 7. Comparison results for case study 2

| Fitness     | Standard BA | MFBA   |
|-------------|-------------|--------|
| minimum     | 0.052038612 | 0.0532810468 |
| maximum     | 0.053694446 | 0.054486114 |
| Standard deviation | 0.10387507 | 0.0777658 |
| mean        | 0.052918954 | 0.05399353627 |

REFERENCES
[1] S.G. Tzafestas, Introduction to Mobile Robot Control, Elsevier, 2014. doi:10.1016/B978-0-12-417049-0.00001-8.
[2] Peter Cork, Robotics, Vision and Control. Fundamental Algorithms in MATLAB, Springer: Berlin, 2011. doi:10.1108/ir.2012.04939fba.005.
[3] Kyung min Han. Collision Free Path Planning Algorithms for Robot Navigation Problem, University of Missouri-Columbia, 2007.
[4] X.-H.Y. Michael Brand, Michael Masuda, Nicole Wehner, Ant Colony Optimization Algorithm for Robot Path Planning, in: Int. Conf. Comput. Des. Appllications (ICCDA 2010) Ant, IEEE, Qinhuangdao, China, n.d.: pp. 436–440.
[5] P.B.A.M.L. GAVRILOVA, Roadmap-Based Path Planning - Using the Voronoi Diagram for a Clearance-Based Shortest Path, IEEE Robot. Autom. Mag. (2008) 58–66. doi:10.1109/MRA.2008.921540.
[6] G.I. Delgado, R.O. Casallas, G.E. Jacinto, Path planning in static scenarios using image processing and cell decomposition, in: IEEE Int. Autumn Meet. Power, Electron. Comput. ROPEC, Ixtapa, Mexico, n.d. doi:10.1109/ROPEC.2014.7036309.
[7] B. Kovács, G. Szayer, F. Taji, M. Burdelis, P. Korondi, A. novel potential field method for path planning of mobile robots by adapting animal motion attributes, Rob. Auton. Syst. 82 (2016) 24–34. doi:10.1016/j.robot.2016.04.007.
[8] P. Abichandani, G. Ford, H.Y. Benson, M. Kam, Mathematical programming for Multi-Vehicle Motion Planning problems, in: IEEE Int. Conf. Robot. Autom., IEEE, Saint Paul, MN, USA, n.d.: pp. 3315–3322. doi:10.1109/IROBOT.2012.6225141.
[9] A. Atyabi, D.M.W. Powers, Review of classical and heuristic-based navigation and path planning approaches, Int. J. Adv. Comput. Technol. 5 (2013) 1–14.
[10] J.K. Ibraheem, F.H. Ajeil, Multi-Objective Path Planning of an Autonomous Mobile Robot in Static and Dynamic Environments using a Hybrid PSO-MFB Optimization Algorithm, ArXiv:1805.00224v2. (2018).
[11] I.R.N. and D.S. Roland Siegwart, Introduction to Autonomous Mobile Robots. 2nd ed., MIT Press, 2011.
[12] F. Fahimi, Autonomous Robots Modeling, Path Planning, and Control, Springer-Verlag US, 2009. doi:10.1007/978-0-387-09538-7.
[13] F.K. Purian, E. Sadeghian, Mobile robots path planning using ant colony optimization and Fuzzy Logic algorithms in unknown dynamic environments, in: Int. Conf. Control. Autom. Robot. Embed. Syst., Jabalpur, India, n.d.: pp. 1–6. doi:10.1109/CARE.2013.6737118.
[14] A.A.-M.W. Wang, Fuzzy inference approach for autonomous ground vehicle navigation in dynamic environment, in: Int. Conf. Control. Syst. Comput. Eng. (ICSCSE 2014), IEEE, Batu Ferringhi, Malaysia, n.d.: pp. 29–34.
[15] C.D. Cheng-hsiung Chinag, Robot navigation in dynamic environments using fuzzy logic and trajectory prediction table, in: Int. Conf. Fuzzy Theory Its Appl., IEEE, Kaohsiung, Taiwan, n.d.: pp. 99–104.
[16] M.K.R.B.V.L. Deepak, PSO based system architecture for path planning of mobile robot in dynamic environment, in: Glob. Conf. Commun. Technol., IEEE, Thuckalay, India, n.d.: pp. 797–801.
[17] Faten Cherni; Yassine Bouterea; Chokri Rekik; Nabil Derbel, Autonomous mobile robot navigation algorithm for planning collision-free path designed in dynamic environments, in: 9th Jordanian Int. Electr. Electron. Eng. Conf., IEEE, Amman, Jordan, n.d.: pp. 1–6.
[18] Y.Y.Y. Li, Mobile robot autonomous path planning based on fuzzy logic and filter smoothing in dynamic environment, in: 12th World Congr. Intell. Control Autom., IEEE, Guilin, China, n.d.: pp. 1479–1484.
[19] Y.L. Liping Sun, *, X. Ding, and L. Wu, Path Planning and Obstacle Avoidance for Mobile Robots in a Dynamic Environment, Open Autom. Control Syst. J. 6 (2016) 77–83.
[20] Q.-Q.W.Y.-J.LX.-Q. Liu, Real-Time globally optimized path planning in a dynamic environment combining artificial potential field and fuzzy neural network, in: Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process., IEEE, Chengdu, China, n.d.: pp. 114–118.
[21] Maryam Yarmohamadi, Improvement of Robot Path
Planning Using Particle Swarm Optimization in Dynamic Environments with Mobile Obstacles and Target, Adv. Stud. Biol. 3 (2011) 43–53.

[22] I.K. Ibraheem, F.H. Ajell, Path Planning of an autonomous Mobile Robot using Swarm Based Optimization Techniques Technique, Al-Khwarizmi Eng. J. 12 (2016) 12–25.

[23] I.K.I. Aws Abdulssalam Najm, On The Design of Nonlinear PID Controller for Nonlinear Quadrotor System, ArXiv:1806.05153. (2018).

[24] I.K. Ibraheem, On The Design of a Novel Finite-Time Nonlinear Extended State Observer for Class of Nonlinear Systems with Mismatch Disturbances and Uncertainties, ArXiv:1805.00257. (n.d.).

[25] State-Space Based H– Robust Controller Design for Boiler-Turbine System, Arab. J. Sci. Eng. 37 (2012) 1767–1776.

[26] I.K. Ibraheem, A Digital-Based Optimal AVR Design of Synchronous Generator Exciter Using LQR Technique, Al-Khwarizmi Eng. J. 7 (2011) 82–94.

[27] G.A.J. Ibraheem Kasim Ibraheem, Motion Control of An Autonomous Mobile Robot using Modified Particle Swarm Optimization Based Fractional Order PID Controller, Eng. Technol. J. 34 (2016) 2406–2419.

[28] I.K. Ibraheem, Robust Governor Design for Hydro machines using H– Loop-Shaping Techniques, in: 6th Eng. Conf., Baghdad University, Baghdad Iraq, n.d.: pp. 403–414.

[29] I.K. Ibraheem, On the frequency domain solution of the speed governor design of non-minimum phase hydro power plant, Mediterr. J. Meas. Control. 8 (2012) 422–429.

[30] X.-S. Yang, Nature-Inspired Optimization Algorithms, 2014. doi:10.1016/B978-0-12-416743-8.00016-6.

[31] X.-S. Yang, Z. Cui, R. Xiao, A.H. Gandomi, M. Karamanoglu, Swarm Intelligence and Bio-Inspired Computation Theory and Applications, 2013. doi:10.1016/B978-0-12-405163-8.00025-9.

[32] S. Yilmaz, E.U. Kucuksille, Y. Cengiz, Modified bat algorithm, Elektron. İr Elektrotechnika 20 (2014) 71–78. doi:10.5755/j01.eee.20.2.4762.

[33] and L.G. GaigeWang, A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization, J. Appl. Math. 2013 (2013) 1–21. doi:10.1155/2013/696491.