IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 February 2024, accepted 18 March 2024, date of publication 21 March 2024, date of current version 27 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380004

RESEARCH ARTICLE

Robot Motion Control Using Stepping Ahead Firefly Algorithm and Kinematic Equations

KAYLASH CHAUDHARY[1], (Member, IEEE), ALVIN PRASAD[2], AVINESH PRASAD[1], AND BIBHYA SHARMA[1], (Member, IEEE)
[1]School of Information Technology, Engineering, Mathematics and Physics, The University of the South Pacific, Suva, Fiji
[2]School of Science and Technology, The University of Fiji, Lautoka, Fiji
Corresponding author: Kaylash Chaudhary (kaylash.chaudhary@usp.ac.fj)

ABSTRACT The use of robots to carry out activities in various fields has evolved over the years. Robots can move from one location to another according to a plan and have a good understanding of the environment to perform their assigned tasks. Robot movements are based on motion planning and control algorithms. This paper proposes a new hybrid motion planning and control algorithm called FAStep-Kinematic (Stepping Ahead Firefly algorithm and Kinematic equations). This algorithm includes the Firefly Stepping Ahead algorithm to avoid obstacles, the equation of motion to control the movement of the robot, and a method to detect obstacles. The algorithm starts with the equation of motion and switches to the stepping ahead firefly algorithm when an obstacle is detected. The stepping ahead firefly algorithm then plans the robot's next step to avoid obstacles. The robot moves to this step. This procedure is repeated until the robot successfully avoids an obstacle and reaches its target. The obstacles used in this article are static and are known to robots. The hybrid algorithm will be used to navigate robots in a cluttered environment. The new algorithm's effectiveness will be seen with an application to tractor-trailer robotic systems. The results show that the robots are able to reach their destination safely and using a shorter route. Additionally, this research compared the new hybrid algorithm's performance with that of the ACO-Kinematic (Ant Colony Optimization and Kinematic equations) algorithm in terms of path length and convergence time. The analysis show that the hybrid algorithm is superior to the ACO-Kinematic algorithm. The proposed algorithm improves the path length by 0.37%, 9.22%, and 5.79% compared to the ACO-Kinematic algorithm in three different scenarios.

INDEX TERMS Robot, motion, control, optimization, algorithm.

I. INTRODUCTION

Robots are devices that perform functions in the form of humans. Today, robots are utilised in many sectors to perform tasks, including healthcare, travel, mining, transportation, military, civilian, manufacturing, agriculture and forestry [1], [2], [3], [4], [5], [6], [7], [8], [9].

In most industries, robots need to move from their current location to another location to complete their task. When moving to a certain location, the robot must avoid collisions with objects or people. This field of study is called robot navigation problems and is divided into route planning, localization, cognitive mapping and motion control [10]. The focus of this paper is on robot path planning and motion control. The two tasks involved are planning a safe path free of obstacles and collisions, and following that path to the robot's goal. In this paper, the robot moves toward the target while continuing to look for obstacles (motion control). When an obstacle is detected, the robot's next steps are planned to avoid this obstacle (path planning). The robot then moves to that step (motion control).

Classical, heuristic, machine and deep learning are the path planning approaches available in the literature. Classical approach includes artificial potential field [11], cell decomposition [12], road map [13], and virtual force field [14]. The major drawback of artificial potential field is the possibility of the robot getting stuck in the local minima and unable to access the target even if it is within the vicinity of

The associate editor coordinating the review of this manuscript and approving it for publication was Zhong Wu.

the target [15]. Another drawback of artificial potential field is the inability to find the optimal path when facing symmetric obstacles [16]. Cell decomposition algorithms that use smaller predefined cells increase the number of cells in a larger environment, producing a shorter path but making it slower, which leads to scalability issues [17]. The problem with cell decomposition is that it becomes more complex and generates infeasible solutions [18] when decomposing a continuous space into cells. Roadmap planners require precise information on the shapes of the obstacles in the environment, which leads to difficulty in the representation of the environment, potentially leading to increased computational cost [19]. One of the shortcomings of virtual force field is that it requires intensive calculations in a dynamic environment. Virtual force field also leads to the local minima, where the robot gets trapped in suboptimal configurations, preventing it from finding a more optimal path [20]. In general, classical methods suffer from either the problem of local minima, the inability to find the optimal path, or scalability issues.

Algorithms such as ant colony optimization [21], particle swarm [22], firefly [23] and artificial bee colony [24] are examples of heuristic approaches. Machine and deep learning methods have also been recently used in robot path planning. These include neural networks, decision trees, Naive Baiyes, to name a few [25]. This paper focuses on a heuristic technique known as the Stepping Ahead Firefly Algorithm (FAStep). The FAStep algorithm was chosen because it has been successfully tested against 65 benchmark test functions. The FAStep algorithm's performance was compared to the standard Firefly and other modified versions and was verified to be superior to variants as well as the standard algorithm [26].

To reach the destination, a robot needs to use a motion control method such as kinematic equations, artificial potential field, and many more [27], [28], [29], [30]. This paper uses kinematic equations for robot motion control.

Compared to the literature where most studies deal with path planning and motion control separately, this paper introduces a hybrid algorithm where a heuristic algorithm (stepping ahead firefly algorithm) is used to avoid obstacles while kinematic equations used to govern the motion of the robot. The proposed hybrid algorithm starts with a kinematic equation for moving the robot to the target. The stepping ahead firefly algorithm is activated only when an obstacle is detected. Next, steps are planned to avoid this obstacle. The robot moves to the new step position and continues the process until the obstacle is completely avoided. This hybrid is a new algorithm compared to others in the literature.

This paper contributes the following to the literature:

1) FAStep-Kinematic algorithm: A new hybrid algorithm consisting of classical and heuristic methods. To the best of the authors' knowledge, no such hybrid algorithm exists. There is an algorithm called ACO-Kinematic, but unlike this algorithm, the algorithm plans steps even when there are no obstacles.
2) Application: To illustrate its usefulness in real-world applications, the hybrid algorithm was successfully applied to tractor-trailer robot systems.
3) Analysis: The new hybrid algorithm has been compared to the ACO-Kinematic algorithm using path length and convergence time as performance metrics. The comparison show that the FAStep-kinematic algorithm outperforms the ACO-kinematic algorithm.

Section II presents the related work from the literature. The objectives of this paper are listed in section III. The stepping ahead firefly algorithm used to avoid obstacles is discussed in section IV. Section V presents the motion control formulation problem. The new hybrid algorithm and its strategic formulation are discussed section VI. The simulation results and application to tractor trailer robotic system are presented in section VII. The comparison of FAStep-Kinematic and ACO-Kinematic with scenarios is discussed in section VIII. This paper concludes with the recommendation for future work in section IX.

II. RELATED WORK

In order to plan and control the robot on the path, the environment in which the robot will be navigating should be captured and analyzed. This section will highlight other bio-inspired algorithms that is proposed to solve motion planning and control problems. Firefly algorithm is implemented in several research [23], [31], [32] to solve the path planning problem. This research paper will highlight selected studies from the literature.

The majority of the proposed studies focus only on path planning with bio-inspired algorithms. In [33], an optimal path is identified using convolutional neural network controller with off-line and on-line tuning Back-Propagation algorithms resulted into a controller. In the same a hybrid swarm optimization algorithm is used to avoid collision and find the shortest path. A hybrid solution to path planning problem is also proposed in [34] which combines the new bio-inspired grey wolf algorithm and particle swarm optimization. Another bio-inspired path planning algorithm is proposed in [35], inspired by plant growth and uses the plant growth route planning algorithm to solve the path planning problem in 3D dynamic environment. In [36] a new path planning algorithm is proposed where the A* algorithm is used to plot the cost effective path points. The adaptive window approach is combined afterwards, to perform the real-time path planning and obstacle avoidance. Similarly, [37] proposed a novel fusion algorithm, a hybrid path planning technique by integrating jump-A* with dynamic windows approach. The combination of jump point search and dynamic windows approach plans the motion fast and avoid obstacles smoothly. In [38], the authors proposed a method to solve time-varying non-linear optimization problem in motion control. A discrete-time kinematics equation for robot motion control is presented in [39].

Further, literature also contains techniques for both path planning and motion control. A two stage motion planning and control method is proposed in a dynamic environment in [40]. Initially a plan is generated using RRT* algorithm for the movement, followed by a CVaR method that assesses the safety risk and designs a constrained receding controller to track the path. Both motion planning and motion control module is proposed in [41] where, the new proposed model combines the advantages of the A* algorithm and the fuzzy analytic hierarchy process to plan the path. The bio-inspired brain limbic system (BLS)-based control method is utilized to control the motion as per plan in an efficient manner. A hybrid algorithm is also proposed in [42] where the ant colony optimization is combined with kinematic equation.

The algorithms discussed above are different from the proposed algorithm since the latter combines the kinematics equation with stepping ahead firefly algorithm. As per the knowledge of the authors, the literature do not contain similar combinations.

III. OBJECTIVES

The objectives of this paper are:

- Create and construct a hybrid algorithm that combines stepping ahead firefly algorithm with kinematic equations (Firefly algorithm for planning steps to avoid obstacles only and kinematic equations for governing the motion of a robot).
- Utilise the new hybrid algorithm to govern the control a mechanical system in a cluttered environment.
- Evaluate and contrast the efficacy of the new hybrid algorithm with a similar method in literature.

IV. STEPPING AHEAD FIREFLY ALGORITHM

The firefly algorithm is classified as swarm intelligence and is a metaheuristic, inspired by the bioluminescent communication of fireflies [43]. The algorithm assumes that all fireflies are unilateral, attracted to bright fireflies, and fireflies having same brightness will move randomly.

A. BRIGHTNESS OF THE FIREFLY

The brightness, I, of a firefly i on firefly j resembles attractiveness and is based on the degree of the brightness and the distance r_{ij} between the two fireflies given as

$$I(r) = \frac{I_s}{r^2} \tag{1}$$

where $I(r)$ is the light intensity and I_s is the source intensity.

Fireflies that are not bright (attractive) are attracted and move to bright fireflies. Every firefly have a distinct attraction value β. The distance between the fireflies determines β and is calculated using

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

B. MOVEMENT OF THE FIREFLY

The movement of firefly i from position x_i to position x_j of the brighter firefly j is accomplished by

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r^2}(x_i - x_j) + \alpha\varepsilon_i, \tag{3}$$

where $x_i(t+1)$ is the new position of firefly i, $x_i(t)$ is the current position of the firefly i, γ is the absorption coefficient, r is the distance between firefly i and j, $\beta_0 e^{-\gamma r^2}$ is the distinct attraction value, $\beta_0 e^{-\gamma r^2}(x_i - x_j)$ is the distance measure based on attraction, and $\alpha\varepsilon_i$ is the random movement.

In the event that the new location has a greater attraction value, the firefly will relocate. Otherwise, the firefly will remain where it is.

C. MODIFICATION OF THE FIREFLY ALGORITHM

In an effort to enhance the firefly algorithm's performance and solve different types of problems, a number of changes have been suggested in the literatures [44], [45], [46], and [26]. This paper uses a modified firefly algorithm proposed by Nand et al. in 2021 [26]. The algorithm is called stepping ahead Firefly algorithm (FAStep). We chose FA for this study because of its simple structure and success in a variety of application areas, including widespread use to solve continuous problems. The reason for choosing FAStep is the ability to avoid local minima and improve performance.

FAStep differs from the standard firefly algorithm in terms of firefly movement equation and having a preference based system for position update. If the fitness of new position is not better, then update the position of the firefly using equation (4).

The modified movement equation is shown below:

$$x_i(t+1) = x_j(t) + \beta_0 e^{-\gamma r^2}(x_i - x_j) + \alpha\varepsilon_i \tag{4}$$

where $x_i(t+1)$ is the new position of firefly i, x_j, is the current position of the best firefly j, $\beta_0 e^{-\gamma r^2}(x_i - x_j)$ is the distance measure based on attraction, and $\alpha\varepsilon_i$ is the random movement.

V. MOTION CONTROL PROBLEM FORMULATION

In this study, the fireflies avoid obstacles and the motion of the robot is determined by its kinematic equations. The obstacle avoidance problem is formulated as a multi-objective path planning problem. The robot needs to know the route planned by the fireflies to avoid obstacles. If there are no obstacles in the robot's path, the robot uses the equation of motion to move towards the target. The fireflies determine the step to avoid an obstacle, and the equation of motion guides the robot to that step. The step planned by the fireflies are safe from collision with that obstacle and shortest with respect to the target. These are the two objectives of the algorithm while planning a step.

A. SHORT PATH

The robot will perform homing movements even if it avoids obstacles. The robot must be within the minimum distance of the target from any step determined by the fireflies. When using multiple robots in the environment, each robot is assigned a set of fireflies to avoid obstacles. The fittest firefly from the swarm will determine the robot's next step. Equation (5) is used to calculate the Euclidean distance between the firefly and the target.

$$d_{ij} = \sqrt{(xf_{ij} - p_{j1})^2 + (yf_{ij} - p_{j2})^2} \quad (5)$$

where d_{ij} is the distance between ith firefly in jth colony and the target of the jth colony, (p_{j1}, p_{j2}) are the coordinates of the target for the jth colony, and (xf_{ij}, yf_{ij}) are the coordinates of the ith firefly in the jth colony.

B. SAFE PATH

If a robots path is unobstructed, it is considered safe. This research has used circular obstacles which can be avoided using the Euclidean distance formula

$$d1_{ijl} = \sqrt{(o_{l1} - xf_{ij})^2 + (o_{l2} - yf_{ij})^2} \quad (6)$$

where $d1_{ij}$ is the distance between ith firefly in jth colony and the obstacle l, (o_{l1}, o_{l2}) are the coordinates of the obstacle l, and (xf_{ij}, yf_{ij}) are the coordinates of the ith firefly in the jth colony. Since the environment consists of multiple obstacles, the total distance between the ith firefly in the jth colony and the lth obstacle can be calculated with

$$f1_{ij} = \sum_{l=1}^{q} d1_{ijl}. \quad (7)$$

In addition, the environment is made up of multiple robots, and collisions between robots must be avoided. A robot is considered a mobile obstacle by another robot. If the proposed algorithm detects that the robot is within the safety parameters of another robot, the algorithm switches to collision avoidance mode. The algorithm then spawns fireflies to avoid this mobile robot, taking into account other obstacles in the environment. Again, the Euclidean distance formula used to calculate the distance between the ith firefly in the jth colony and another robot is

$$d2_{ijh} = \sqrt{(xf_{ij} - x_h)^2 + (yf_{ij} - y_h)^2}, \quad (8)$$

where $d2_{ijh}$ is the distance between ith firefly in jth colony and the robot h, and (x_h, y_h) are the coordinates of the robot h.

The sum of the distances between ith firefly in the jth colony and the hth robot $(f2_{ij})$ is given as

$$f2_{ij} = \sum_{\substack{h=1 \\ h \neq j}}^{n} d2_{ijh} \quad (9)$$

C. PROBLEM FORMULATION

This problem is formulated as a minimization optimization problem that aims to find the best steps to avoid obstacles. The fitness of the ith firefly for robot j is given as

$$f_{ij} = a.\frac{1}{f1_{ij}} + b.\frac{1}{f2_{ij}} + c.d_{ij} \quad (10)$$

which a, b, and c are control parameters.

The position of fittest firefly, calculated by equation (10), will be chosen as next step for robot j. This process will continue until robot j successfully avoids an obstacle. Avoiding an obstacle safely is dependent on parameters, a and b. High values for these parameters mean that the robot can safely avoid obstacles. Reducing the values of these parameters increases the likelihood of collisions. For parameter, c, increasing the value minimizes the path length and decreasing the value maximizes the path length. The range for parameters a and b is [0.1, 1] and the range for parameter c is [0.00001, 0.01] [23].

VI. PROPOSED ALGORITHM

The proposed algorithm is a combination of the firefly's step-ahead algorithm and an equation of motion called FA *Step-Kinematic*. The robot will start its journey with the kinematic equations. While moving, the robot will continuously check for obstacles using the following equation:

$$Dist_{RO} = \sqrt{(o_1 - R_1)^2 + (o_2 - R_2)^2} \quad (11)$$

where $Dist_{RO}$ is the distance between the robot and an obstacle, (o_1, o_2) are the coordinates of an obstacle and (R_1, R_2) are the coordinates of the robot.

For an obstacle, if $Dist_{RO}$ is greater than the radius of that obstacle plus ε (radius of the protective region), then the firefly optimization algorithm will be activated (Steps 2.2.1 to 2.2.3 of Algorithm 1). The firefly optimization algorithm will generate points around that obstacle and the robot will move from one point to another using its kinematic equations. Once the obstacle has been avoided, only the kinematic equations will run. Algorithm 1 shows the FA *Step-Kinematic* algorithm.

Figure 1 shows the flowchart of the FAStep-Kinematic algorithm. The flowchart shows the logic in the algorithm and the integration points of the FAStep algorithm and kinematic equations. The algorithm starts with the calculation of the distance between the robot and the target. If the robot has not reached the target, the algorithm will continue executing either the kinematic equations for motion control or stepping ahead firefly algorithm for path planning. The FAStep algorithm will only execute if there are obstacles in the robots path and to avoid an obstacle, the algorithm will generate a point which the robot will see as immediate target and will move towards it using kinematic equations. If there are no obstacles in the robots path, the robot will move towards the target using its kinematic equations.

VII. RESULTS

The proposed algorithm was applied to three case studies with different environment complexities. The level of environment complexity can be based on the obstacles (number, size, shapes and types of obstacles with different singularities), robots (number and types of robots) and real life scenarios. This paper uses case studies which includes multiple obstacles of different sizes and shapes and multiple point mass robots with an application to a mechanical system.

Algorithm 1 FAStep-Kinematic Algorithm

Step 1: Calculate distance between robot and target ($Dist_{RT}$)
while $Dist_{RT} > 0$ **do**
  Step 2: Calculate distance between robot and each obstacles ($Dist_{RO_i}$)
  if ($Dist_{RO_i} \leq ro_i + \varepsilon$) **then**
    Step 2.1: Move robot using kinematic equations
  end
  else
    Step 2.2.1: Initialize firefly parameters and population of fireflies
    Random partition $x_i (i = 1, 2, \ldots, n)$
    Step 2.2.2: Define Cost Function f_x associated with light intensity I
    Step 2.2.3: Define absorbtion coefficient γ
    foreach $i = 1 : n$ *(all n fireflies)* **do**
      foreach $j = 1 : n$ *(n fireflies)* **do**
        if $I_j \leq I_i$ **then**
          Vary attractiveness with distance r via $e^{-\gamma r^2}$
          Step 4: Stepping Ahead
          Calculate new position ($NewPos$) using Equation .... Evaluate $f(NewPos)$
          Evaluate new solutions and update light intensity
          foreach $i = 1 : n$ **do**
            if $f(Pos_i) \leq f(BestSol)$ **then**
              $BestSol = Pos_i$
            end
          end
          if $f(NewPos) \leq f(Pos_i)$ **then**
            $Pos_i = NewPos$
            if $f(NewPos) \leq f(BestSol)$ **then**
              $BestSol = NewPos$
            end
          end
        end
      end
    end
    Rank the fireflies and find the current best location
    Step 2.2.4: Move robot to the best location using kinematic equations
  end
end

The new hybrid algorithm was able to avoid obstacles and safely navigate the robot to its destination in all case studies. This algorithm uses the motion planning and control parameters shown in Table 1. Parameters 1-8 are used by the Stepping Ahead Firefly algorithm for robot path planning, and parameters 9 and 10 are used by the kinematic equations for motion control. Parameters a and b are safety parameters used to avoid obstacles, and parameter c is a convergence parameter used to determine the time it takes for the robot to reach its target. Slow convergence increases operating costs, and fast convergence impairs the safety of the robot and its environment. Other than brute force methods, there is no way to set these parameters (a, b and c) in the literature. Parameter 1 is the swarm size which has been set to 50. Parameters 2 - 5 use the default values of the stepping ahead firefly algorithm.

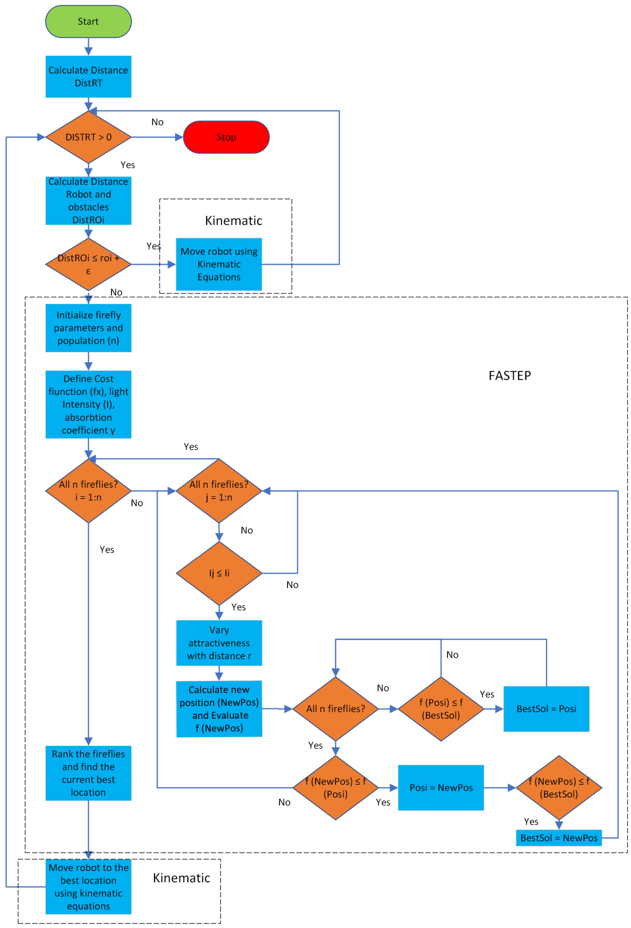


FIGURE 1. Flowchart of the FAStep-Kinematic algorithm.

A. CASE STUDY 1

The new algorithm was employed to facilitate the navigation of a point mass robot from its initial position to a designated destination in an environment containing many obstacles. The equations governing the motion of a point mass robot, as it traverses from its current location (x_0, y_0) to another point (x_1, y_1), can be expressed as follows:

$$\begin{aligned} \dot{x} &= \alpha_1(x_1 - x), \quad & \dot{y} &= \alpha_2(y_1 - y), \\ x(0) &= x_0, & y(0) &= y_0, \end{aligned} \tag{12}$$

where α_1 and α_2 are positive real numbers.

The Stability of Robot Motion:

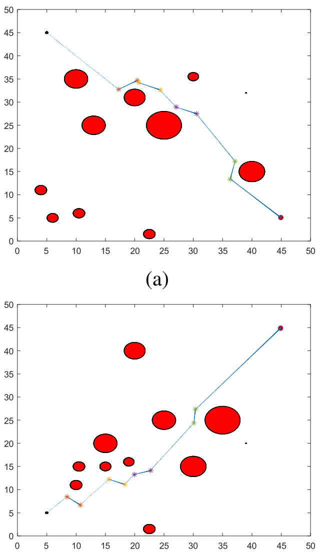
Theorem 1: When the point mass robot moves from an initial position (x_0, y_0) to another point (x_1, y_1), its motion described by equation (12) is asymptotically stable.

Proof: We use the Direct method of Lyapunov [47] to proof Theorem 1. Consider a Lyapunov function of the form $V(x, y) = \frac{1}{2}[(x_1 - x)^2 + (y_1 - y)^2]$ which is continuous and has continuuous partial derivatives for all $(x, y) \in \mathbb{R}^2$. Moreover, it is clear that $V(x, y) > 0$ for all $(x, y) \neq (x_1, y_1)$ and $V(x_1, y_1) = 0$. Next, note that $\dot{V}(x, y) = -(x_1 - x)\dot{x} - (y_1 - y)\dot{y}$. Substituting (12) into $\dot{V}(x, y)$ gives

$$\dot{V}(x, y) = -\alpha_1(x_1 - x)^2 - \alpha_2(y_1 - y)^2.$$

TABLE 1. FAStep-kinematic parameters.

No.	Parameter	Value
1	No. of Fireflies, n	50
2	Light Absorption Coefficient, γ	1
3	Attraction Coefficient Base Value, β	2
4	Mutation Coefficient, α	0.2
5	Mutation Coefficient Damping Ratio, α_0	0.8
6	Safety Parameter 1, a	0.18
7	Safety Parameter 2, b	0.18
8	Convergence Parameter 1, c	0.01
9	Convergence Parameter 2, α_1	0.1
10	Convergence Parameter 3, α_2	0.1



(a)

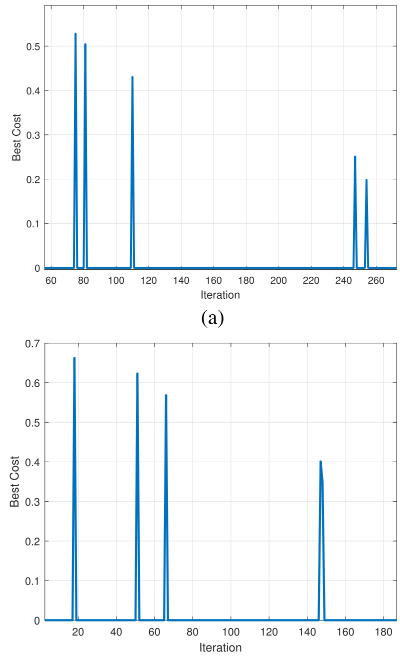
(b)

FIGURE 2. Path of a point-mass robot with: (a) initial position (5, 45) and target position (45, 5) and (b) initial position (5, 5) and target position (45, 45).

Since α_1 and α_2 are positive real numbers, it follows that $\dot{V}(x, y) < 0$ for all $(x, y) \neq (x_1, y_1)$ and $\dot{V}(x_1, y_1) = 0$. Thus we conclude that the motion of the point mass robot from an initial position (x_0, y_0) to another point (x_1, y_1) is asymptotically stable.

The paths of the point-mass robot generated in two scenarios are shown in figure 2. The point-mass robot starts its journey with the equations of motion and continuously checks for obstacles. If the robot is within the parameter of an obstacle then the algorithm generates fireflies which generate the points. The robot then moves to the generated point using kinematic equations. Figure 2 further shows that the points are only generated around obstacles that falls in the path of the robot. When the robot has safely avoided the obstacle, it will switch back to kinematic equations and continue monitoring further obstacles in its path.

Figures 3(a) and 3(b) display the graph of the cost function for the scenarios depicted in figures 2(a) and 2(b), respectively. Typically, the cost function value is zero, indicating that the algorithm is now executing the kinematic equations section. The cost function value will be nonzero during the execution of the stepping ahead firefly algorithm



(a)

(b)

FIGURE 3. Cost function graphs for case study 1.

section. Note that when a robot comes across an obstacle in its trajectory, it will navigate around it by employing the stepping ahead firefly algorithm. The peaks observed in figures 3(a) and 3(b) indicate that the algorithm successfully circumvented obstacles by generating waypoints that the robot subsequently followed to prevent collisions.

The Feasibility of the Algorithm: Figure 4 shows the convergence graph of the controllers ($\dot{x}$ and $\dot{y}$ of equation 12) for figure 2(a). The decreasing portion of the $\dot{x}$ and the increasing portion of the $\dot{y}$ indicate the movement of the robot from one generated point to another. For example, from $t = 0$ to approximately $t = 90$ units, the robot travels from the initial point towards the target but encounters an obstacle which it avoids by generating a point, from which the robot travels to that generated point (refer to figure 2(a)). Similarly, the convergence of $\dot{x}$ and $\dot{y}$ after $t = 900$ units is for the movement of the robot from the last generated point to the target. Overall, the graph shows that the proposed algorithm converges effectively as the robot safely reaches the target. Similar graphs were also obtained for the other case studies. Also, the major portion of the proposed algorithm is stepping ahead firefly algorithm which has been tested for its effectiveness in a number of benchmark functions by Nand et al. [26].

B. CASE STUDY 2

The proposed algorithm was applied to control the movement of two point-mass robots in an environment with multiple obstacles. The movement of a robot is like a dynamic obstacle to another robot. Therefore, in addition to avoiding static obstacles, the proposed algorithm also avoids moving obstacles. To avoid the moving obstacles, the algorithm

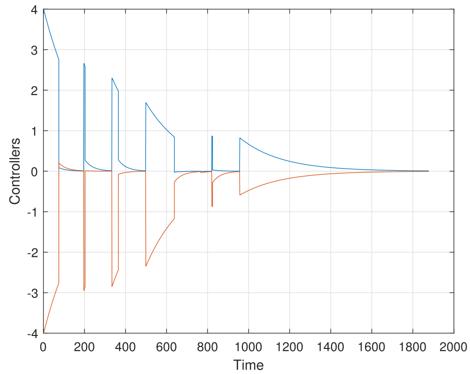
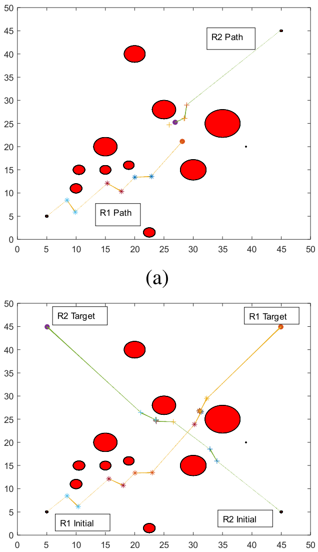


FIGURE 4. **Convergence of the controllers.**



(a)

(b)

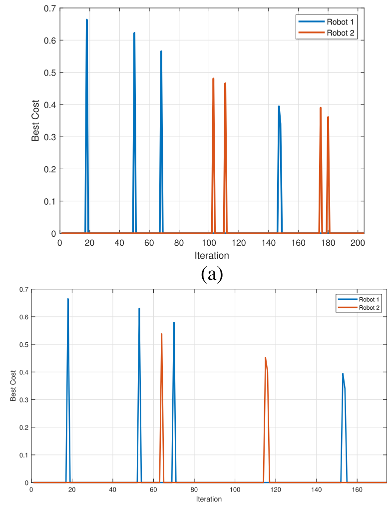
FIGURE 5. **Paths of two point-mass robots.**

generates fireflies which will determine a point that will avoid collision with the moving obstacle. This is the same process that has been discussed in section VII-A. Figure 5 (a) shows the two robots (R1 and R2) avoiding each other. R1 starts its journey from initial position (5, 5) and moves towards target (45, 45) while avoiding obstacles. Likewise, R2 starts from position (45, 45) and moves towards target at position (5, 5). The figure shows that R2 is avoiding an obstacle and R2 using points generated by fireflies. Figure 5 (b) shows the full paths of the two robots (R1 and R2). The two robots try to avoid collision at point (30, 22) using the same technique as described above.

Figures 6(a) and 6(b) display the graph of the cost function for the scenarios depicted in figures 5(a) and 5(b), respectively. The graphs depict the cost function values for two individual robots, as each robot has its own distinct cost function. Once more, the graphs clearly demonstrate that whenever a robot successfully avoids an obstacle, the value of the cost function will not be zero.

C. APPLICATION

The new hybrid algorithm has been used to govern the motion of a tractor-trailer robot. Consider a non-standard tractor



(a)

(b)

FIGURE 6. **Cost function graphs for case study 2.**

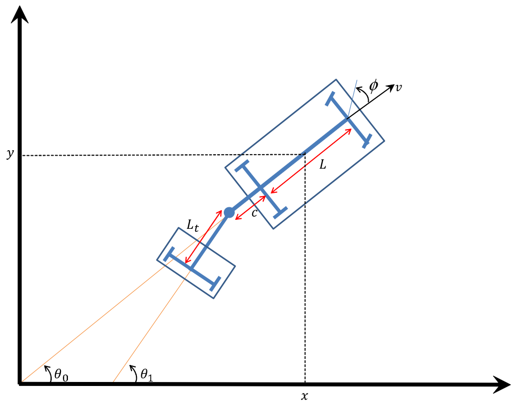


FIGURE 7. **Kinematics of the tractor-trailer robot.**

trailer robot consisting of a car-like rear-wheel drive vehicle and a two-wheel passive trailer mounted on the rear axle of the vehicle (Figure 7). Let (x, y), θ_0, and ϕ represent the cartesian coordinates of the tractor-trailer robot, orientation with respect to the x-axis, and the steering angle with respect to its longitudinal axis, respectively. Likewise, θ_1, L and L_t denotes the orientation of the trailer with respect to the x-axis, the length of the mid-axle of the tractor and the length of the mid axle of the trailer, respectively. The tractor-trailer robot's motion is controlled by the kinematic equations [48]

$$\left.\begin{aligned} \dot{x} &= v\cos\theta_0 - \tfrac{v}{2}\tan\phi\sin\theta_0, \\ \dot{y} &= v\sin\theta_0 + \tfrac{v}{2}\tan\phi\cos\theta_0, \\ \dot{\theta}_0 &= \tfrac{v}{L}\tan\phi, \\ \dot{\theta}_1 &= \tfrac{v}{Lt}\left(\sin(\theta_0-\theta_1) - \tfrac{c}{L}\tan\phi\cos(\theta_0-\theta_1)\right), \end{aligned}\right\} \tag{13}$$

where v and ϕ are the translational velocity and the steering angle, respectively, of the tractor robot. If we want the robot to move from its current position (x_0, y_0) to another point (x_1, y_1), then we define v and ϕ as

$$v = \alpha\sqrt{(y_1-y_0)^2 + (x_1-x_0)^2},$$
$$\phi = \frac{7}{9}\tan^{-1}\left(\xi + \frac{\beta}{\cos|\theta_0-\theta_1|}\right)$$

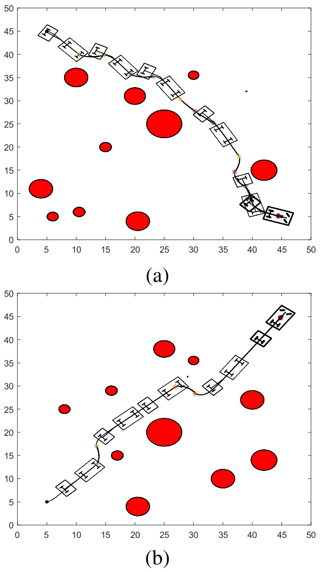


FIGURE 8. Path of a tractor-trailer robot with: (a) initial position (5, 45) and target position (45, 5) and (b) initial position (5, 5) and target position (45, 45) (Source: Authors own work).

where α is a positive real number, $\beta = \max\{0, 0.5 - \cos|\theta_1 - \theta_0|\} \cdot sign(\theta_1 - \theta_0)$ and ξ is obtained by numerically solving the differential equation

$$\dot{\xi} = \frac{(y_1 - y_0)\cos\theta_0 - (x_1 - x_0)\sin\theta_0}{\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + 0.01}} - \text{atan2}(y_1 - y, x_1 - x_0) + \theta_0,$$
$$\xi(0) = \text{atan2}(y_1 - y_0, x_1 - x_0) - \theta_0(0).$$

The tractor-trailer robot, governed by the kinematic equations (13) will initially move towards a designated target. During its motion, if an obstacle is detected within a distance of ε (from the robot's position) then Steps 2.2.1 to 2.2.3 of Algorithm 1 will be activated to generate point(s) around that obstacle so that the robot's motion is deviated towards those generated point(s). The generated point(s) should be at a distance of at least $\sqrt{L^2 + b^2}/2$ (where b represents the width of the tractor robot) away from the obstacle so that when the robot moves to the generated point, it does not collide with the obstacle. This strategy is repeated until all the obstacles that lie along the robot's route are avoided and the robot reaches its designated target.

Figures 9(a) and 9(b) display the graph of the cost function for the scenarios depicted in figures 8(a) and 8(b), respectively.

The path of a tractor trailer robot in two distinct scenarios is depicted in Figure 8.

VIII. DISCUSSION

This section will compare the performance of the new hybrid algorithm with a similar algorithm, ACO-Kinematic [42]. The path length and convergence time will be used to measure the performance of the both algorithms. A shorter path length and low convergence time are preferred from any motion control algorithm. Both the algorithms have been tested on three scenarios. For each scenario, both algorithms were executed for 30 runs. The average path length and convergence time has been used to compare the two algorithms as shown in Table 2.

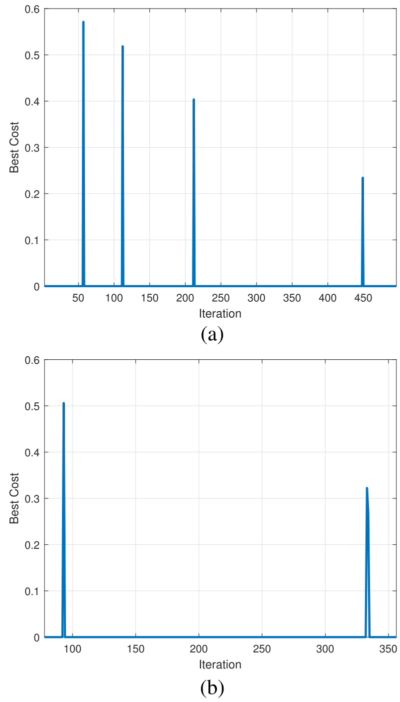


FIGURE 9. Cost function graphs for the Application scenario.

TABLE 2. The convergence time and average path lengths of the two algorithms.

	Algorithms			
	FAStep-Kinematic		ACO-Kinematic	
Scenario	Time (s)	Path Length (cm)	Time (s)	Path Length (cm)
1	69.62	56.56	133.76	56.77
2	154.04	59.69	283.25	65.75
3	95.80	45.73	173.27	48.54

The results in Table 2 show that FAStep-Kinematic algorithm outperformed ACO-Kinematic algorithm. The point-mass robot under the control of the FAStep-Kinematic algorithm exhibited a shorter time to reach its destination in comparison to the point-mass robot controlled by the ACO-Kinematic method. The average path length achieved by FAStep-Kinematic algorithm is less than ACO-Kinematic.

Figure 10 shows the path taken by point-mass robots controlled by FAStep-Kinematic and ACO-Kinematic in scenario 1. Figure 10 (a) further shows that the point-mass robot controlled by FAStep-Kinematic algorithm only moves through kinematic equations without using the firefly optimization algorithm. This is because there are no obstacles in its path. In comparison to the ACO-kinematic algorithm, point-mass robots move from one point to another. These points are generated by the fireflies. Generation of each point takes time and this is evident from the results shown in Table 2. This is a drawback of the ACO-kinematic

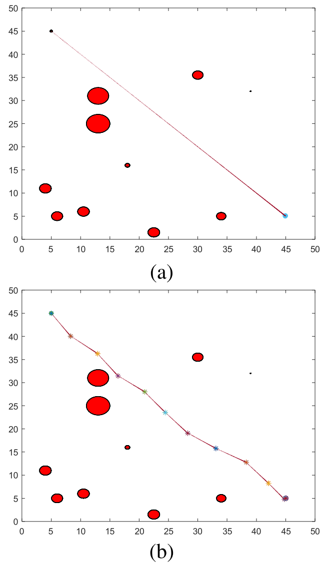


FIGURE 10. Path of a point-mass robot generated by (a) FAStep-Kinematic and (b) ACO-Kinematic with initial position (5, 45) and target position (45, 5).

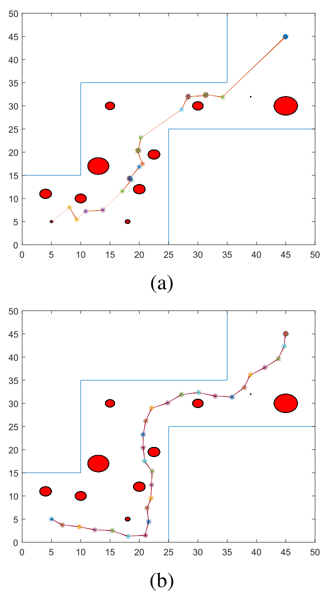


FIGURE 11. Path of a point-mass robot generated by (a) FAStep-Kinematic and (b) ACO-Kinematic with initial position (5, 5) and target position (45, 45).

algorithm. For scenario 2, the paths taken by point-mass robots controlled by FAStep-Kinematic and ACO-Kinematic algorithms are shown in figure 11. In addition to circular obstacles, this scenario also has line obstacles which can be avoided by a technique known as minimum distance discussed in [42]. The point-mass robot, governed by the FAStep-Kinematic algorithm, successfully reached the goal in a time span of 107.8 seconds. Throughout its trajectory, the robot covered an average distance of 65.97cm. In the ACO-Kinematic experiment, the point-mass robot exhibited a total time of 152.99 seconds, while the average path length covered by the robot was measured to be 76.2cm. Figure 12 depicts the motion paths of the point-mass robots

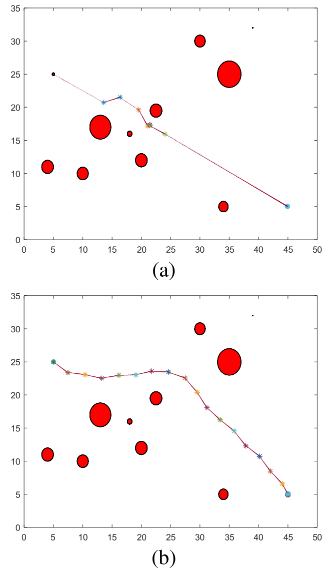


FIGURE 12. Path of a point-mass robot generated by (a) FAStep-Kinematic and (b) ACO-Kinematic with initial position (5, 25) and target position (45, 5).

using FAStep-Kinematic and ACO-Kinematic for scenario 3. The point-mass robot using the FAStep-Kinematic method required a total time of 95.80s to successfully arrive at its designated location. Throughout its journey, the robot covered an average route length of 45.73cm. The robot controlled by the ACO-Kinematic exhibited an average route length of 48.54cm, accompanied by a time duration of 173.27s.

The enhancement percentage technique obtained from [49] is used to calculate the enhancement of path length and time of FAStep-Kinematic when compared to ACO-Kinematic. FAStep-Kinematic enhanced the path length by 0.37%, 9.22%, and 5.79% and time by 47.95%, 45.62% and 44.71% in the three scenarios above. This findings indicate that the FAStep-kinematic algorithm was capable of attaining a shorter route within a shorter duration compared to the ACO-Kinematic algorithm. Therefore, it can said that FAStep-Kinematic is a better algorithm when compared to ACO-Kinematic.

IX. CONCLUSION AND FUTURE WORK

This paper presented a unique hybrid algorithm composed of stepping ahead firefly algorithm and kinematic equations of a robot. Uniqueness is that the firefly stepping ahead algorithm is only used to avoid obstacles. If there are no obstacles in robot's path, the stepping ahead firefly algorithm is not activated and only the equation of motion are executed. The authors have also deployed a relatively new method to detect obstacles, that is, a way for the algorithm to know when to execute stepping ahead firefly algorithm or kinematic equations. If the robot is not within the obstacle safety parameter, use the equations of motion to move the robot towards its target. Otherwise, use the stepping ahead firefly algorithm to avoid the obstacle. According to the authors knowledge, this is the first time this type of hybrid has been proposed. Obstacle avoidance was done by solving a multi-objective optimization problem consisting of finding the safest and shortest route. By avoiding obstacles and using the equations of motion, the robot moves towards the destination.

The new hybrid algorithm was applied to three case studies.The findings show that the robots successfully reached their target by selecting a more efficient and secure route.

The new hybrid algorithm was also compared to a similar hybrid, ACO-Kinematic, in terms of path length and convergence time. Mean path lengths and convergence times were recorded in three scenarios. The results show that FAStep-Kinematic achieves shorter paths and convergence times than ACO-Kinematic in all three scenarios. The results also show a vast difference in convergence time between FAStep kinematics and ACO kinematics. The main reason for this is the architecture of the two algorithms. ACO-Kinematic uses Ant-Colony optimization to plan the robot's steps until the robot reaches its target, while FAStep-Kinematic uses the Stepping Ahead Firefly algorithm to plan a step to avoid obstacles only. This demonstrates the arterial benefits of FAStep kinematic over the ACO kinematic algorithm. The proposed algorithm improved the path length by 0.37%, 9.22%, and 5.79% compared to the ACO-Kinematic algorithm in three different scenarios. It also improved the time taken to reach the destination by 47.95%, 45.62% and 44.71% compared to the ACO-Kinematic algorithm in three scenarios.

The proposed algorithm was applied only to static obstacles where the robot knew its position. In future research, the authors intend to employ the algorithm in the context of static obstacles characterised by uncertain positions, dynamic obstacles, and various other mechanical systems. Also, current system of adjusting parameters is using brute force technique. The authors, in the future, will explore the relationship between the different parameters, develop and use an intelligent mathematical method to adjust the parameters. Due to the scope of this paper, the application of this algorithm in an emergency system where robot needs to arrive quickly will be part of future work.

SOURCE CODE AVAILABILITY

The MATLAB code used for simulations are available at http://repository.usp.ac.fj/id/eprint/13815.

REFERENCES

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, Mar. 2012.

[2] J. Khurshid and H. Bing-rong, "Military robots—A glimpse from today and tomorrow," in *Proc. 8th Control, Autom., Robot. Vis. Conf.*, 2004, pp. 771–777.

[3] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 91–103, Jun. 2009.

[4] S. A. Napper and R. L. Seaman, "Applications of robots in rehabilitation," *Robot. Auto. Syst.*, vol. 5, no. 3, pp. 227–239, 1989.

[5] J. Raj, K. Raghuwaiya, J. Vanualailai, and B. Sharma, "Navigation of car-like robots in three-dimensional space," in *Proc. 5th Asia–Pacific World Congr. Comput. Sci. Eng.*, Dec. 2018, pp. 271–275.

[6] S. A. Kumar and J. Vanualailai, "A Lagrangian UAV swarm formation suitable for monitoring exclusive economic zone and for search and rescue," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2017, pp. 1874–1879.

[7] S. A. Kumar, K. Chand, L. I. Paea, I. Thakur, and M. Vatikani, "Herding predators using swarm intelligence," in *Proc. IEEE Asia–Pacific Conf. Comput. Sci. Data Eng. (CSDE)*, Dec. 2021, pp. 1–6.

[8] R. Chand, R. P. Chand, and S. A. Kumar, "Switch controllers of an *n*-link revolute manipulator with a prismatic end-effector for landmark navigation," *PeerJ Comput. Sci.*, vol. 8, p. e885, Feb. 2022.

[9] A. Holzinger, A. Saranti, A. Angerschmid, C. O. Retzlaff, A. Gronauer, V. Pejakovic, F. Medel-Jimenez, T. Krexner, C. Gollob, and K. Stampfer, "Digital transformation in smart farm and forest operations needs human-centered AI: Challenges and future directions," *Sensors*, vol. 22, no. 8, p. 3043, Apr. 2022.

[10] N. Buniyamin, N. Sariff, W. Wan Ngah, and Z. Mohamad, "Robot global path planning overview and a variation of ant colony system algorithm," *Int. J. Math. Comput. Simul.*, vol. 5, no. 1, pp. 9–16, 2011.

[11] M. Cheol Lee and M. Gyu Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jun. 2003, pp. 735–740.

[12] M. Kloetzer, C. Mahulea, and R. Gonzalez, "Optimizing cell decomposition path planning for mobile robots using different metrics," in *Proc. 19th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2015, pp. 565–570.

[13] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *Proc. IEEE Int. Conf. Robot. Autom.*, Oct. 2004, pp. 467–472.

[14] S. A. Bortoff, "Path planning for UAVs," in *Proc. Amer. Control Conf.*, 2000, pp. 364–368.

[15] S. M. H. Rostami, A. K. Sangaiah, J. Wang, and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–19, Dec. 2019.

[16] H. M. Jayaweera and S. Hanoun, "A dynamic artificial potential field (D-APF) UAV path planning technique for following ground moving targets," *IEEE Access*, vol. 8, pp. 192760–192776, 2020.

[17] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107376.

[18] M. Šeda, "Roadmap methods vs. cell decomposition in robot motion planning," in *Proc. 6th WSEAS Int. Conf. Signal Process., Robot. Autom.*, Athens, Greece, 2007, pp. 127–132.

[19] Y. Liu and Y. Jiang, "Robotic path planning based on a triangular mesh map," *Int. J. Control, Autom. Syst.*, vol. 18, no. 10, pp. 2658–2666, Oct. 2020.

[20] J. Ni, W. Wu, J. Shen, and X. Fan, "An improved VFF approach for robot path planning in unknown and dynamic environments," *Math. Problems Eng.*, vol. 2014, pp. 1–10, Jan. 2014.

[21] M. Brand, M. Masuda, N. Wehner, and X.-H. Yu, "Ant colony optimization algorithm for robot path planning," in *Proc. Int. Conf. Comput. Design Appl.*, vol. 3, Jun. 2010, pp. V3-436–V3-440.

[22] H. Wang and Z. Zhou, "A heuristic elastic particle swarm optimization algorithm for robot path planning," *Information*, vol. 10, no. 3, p. 99, Mar. 2019.

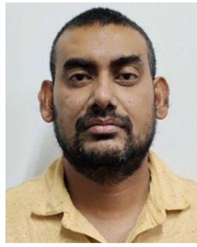
[23] B. K. Patle, A. Pandey, A. Jagadeesh, and D. R. Parhi, "Path planning in uncertain environment by using firefly algorithm," *Defence Technol.*, vol. 14, no. 6, pp. 691–701, Dec. 2018.

[24] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *J. Comput. Appl. Math.*, vol. 236, no. 11, pp. 2741–2753, May 2012.

[25] M. W. Otte, "A survey of machine learning approaches to robotic path-planning," Dept. Comput. Sci., Univ. Colorado, Boulder, CO, USA, Tech. Rep., 2015.

[26] R. Nand, B. N. Sharma, and K. Chaudhary, "Stepping ahead firefly algorithm and hybridization with evolution strategy for global optimization problems," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107517.

[27] H. Martınez-Alfaro and S. Gómez-Garcıa, "Mobile robot path planning and tracking using simulated annealing and fuzzy logic control," *Expert Syst. Appl.*, vol. 15, nos. 3–4, pp. 421–429, Oct. 1998.
[28] Y. Tanaka, T. Tsuji, M. Kaneko, and P. G. Morasso, "Trajectory generation using time scaled artificial potential field," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Innov. Theory, Pract. Appl.*, Mar. 1998, pp. 223–228.
[29] C. Tsai and T. Wang, "Nonlinear control of an omnidirectional mobile robot," in *Proc. 8th Int. Conf. Autom. Technol.*, vol. 732, 2005, p. 727.
[30] K. Watanabe, Y. Shiraishi, S. G. Tzafestas, J. Tang, and T. Fukuda, "Feedback control of an omnidirectional autonomous platform for mobile service robots," *J. Intell. Robot. Syst.*, vol. 22, no. 3, pp. 315–330, 1998.
[31] H. Kundra, W. Khan, M. Malik, K. P. Rane, R. Neware, and V. Jain, "Quantum-inspired firefly algorithm integrated with cuckoo search for optimal path planning," *Int. J. Mod. Phys. C*, vol. 33, no. 2, Feb. 2022, Art. no. 2250018.
[32] M. M. Gangadharan and A. Salgaonkar, "Ant colony optimization and firefly algorithms for robotic motion planning in dynamic environments," *Eng. Rep.*, vol. 2, no. 3, p. e12132, Mar. 2020.
[33] O. Wahhab and A. Al-Araji, "Path planning and control strategy design for mobile robot based on hybrid swarm optimization algorithm," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 3, pp. 565–579, Jun. 2021.
[34] F. Gul, I. Mir, D. Alarabiat, H. M. Alabool, L. Abualigah, and S. Mir, "Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning," *J. Parallel Distrib. Comput.*, vol. 169, pp. 171–184, Nov. 2022.
[35] Y. Zhou, Y. Su, A. Xie, and L. Kong, "A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV," *Chin. J. Aeronaut.*, vol. 34, no. 9, pp. 199–209, Sep. 2021.
[36] X. Zhong, J. Tian, H. Hu, and X. Peng, "Hybrid path planning based on safe A algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *J. Intell. Robotic Syst.*, vol. 99, no. 1, pp. 65–77, Jul. 2020.
[37] L. Liu, J. Yao, D. He, J. Chen, J. Huang, H. Xu, B. Wang, and J. Guo, "Global dynamic path planning fusion algorithm combining jump—A algorithm and dynamic window approach," *IEEE Access*, vol. 9, pp. 19632–19638, 2021.
[38] Y. Zhang, L. He, C. Hu, J. Guo, J. Li, and Y. Shi, "General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization," *J. Comput. Appl. Math.*, vol. 347, pp. 314–329, Feb. 2019.
[39] M. M. Yahaya, P. Kumam, A. M. Awwal, and S. Aji, "A structured quasi-Newton algorithm with nonmonotone search strategy for structured NLS problems and its application in robotic motion control," *J. Comput. Appl. Math.*, vol. 395, Oct. 2021, Art. no. 113582.
[40] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using CVaR-constrained optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3924–3931, Oct. 2019.
[41] C. Kim, J. Suh, and J.-H. Han, "Development of a hybrid path planning algorithm and a bio-inspired control for an omni-wheel mobile robot," *Sensors*, vol. 20, no. 15, p. 4258, Jul. 2020.
[42] K. Chaudhary, A. Prasad, V. Chand, and B. Sharma, "ACO-kinematic: A hybrid first off the starting block," *PeerJ Comput. Sci.*, vol. 8, p. e905, Mar. 2022.
[43] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. U.K.: Luniver Press, 2010.
[44] A. Yelghi and C. Köse, "A modified firefly algorithm for global minimum optimization," *Appl. Soft Comput.*, vol. 62, pp. 29–44, Jan. 2018.
[45] J.-S. Chou and N.-T. Ngo, "Modified firefly algorithm for multidimensional optimization in structural design problems," *Struct. Multidisciplinary Optim.*, vol. 55, no. 6, pp. 2013–2028, Jun. 2017.
[46] H. Wang, W. Wang, H. Sun, J. Zhao, H. Zhang, J. Liu, and X. Zhou, "A new firefly algorithm with local search for numerical optimization," in *Proc. Int. Symp. Comput. Intell. Intell. Syst.* Cham, Switzerland: Springer, 2015, pp. 13–22.
[47] A. M. Lyapunov, "The general problem of the stability of motion," *Int. J. Control*, vol. 55, no. 3, pp. 531–534, 1992.
[48] A. Prasad, B. Sharma, J. Vanualailai, and S. A. Kumar, "A geometric approach to target convergence and obstacle avoidance of a nonstandard tractor-trailer robot," *Int. J. Robust Nonlinear Control*, vol. 30, no. 13, pp. 4924–4943, Sep. 2020.
[49] Z. E. Kanoon, A. S. Al-Araji, and M. N. Abdullah, "Enhancement of cell decomposition path-planning algorithm for autonomous mobile robot based on an intelligent hybrid optimization method," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 3, pp. 161–165, 2022.



KAYLASH CHAUDHARY (Member, IEEE) received the master's and Ph.D. degrees in computer science from The University of the South Pacific. He is currently a Senior Lecturer of computer science. He has published more than 50 articles in the field of crypto-currency, model checking, robot navigation using artificial intelligence, educational data mining, and targeted interventions for higher education.

ALVIN PRASAD is currently pursuing the Ph.D. degree with The University of Fiji. He is a Computer Science and Information Technology Lecturer with The University of Fiji. He has published multiple research articles in ranked journals and presented his research at various conferences. Apart from academia, his interest is in community engagement with the intent to educate and create awareness of various developments in information and communication technologies. His research interests include robotics, software engineering, machine learning, and educational technology.

AVINESH PRASAD received the master's and Ph.D. degrees in applied mathematics from The University of the South Pacific, Fiji, in 2008. He is currently a Senior Lecturer of mathematics with the School of Information Technology, Engineering, Mathematics and Physics. His research interest includes the dynamics of nonlinear systems and robotics.

BIBHYA SHARMA (Member, IEEE) received the Ph.D. degree. He was an Associate Dean for Learning and Teaching with the Faculty of Science, Technology and Environment, The University of the South Pacific (USP), from 2010 to 2019. He is currently a Professor of mathematics with USP. He is a member of several professional mathematics societies and unions and has published more than 100 articles and book chapters in the fields of mathematics education, robotics, biologically inspired processes, and the TLPs of higher education. He champions adaptive works and innovations in the areas of eLearning and mLearning for higher education.