



Robot Motion Control using Dual Avoidance Scheme

Kaylash Chaudhary,* Vishal Chand, Avinesh Prasad and Bibhya Sharma

Abstract

This paper presents a novel hybrid algorithm, BAT-Lyapunov-based control (LbCS), designed to enhance dynamic path planning for robots by integrating BAT optimization and LbCS. Traditional path planning methods often struggle with challenges such as local minima entrapment and slow convergence, particularly in dynamic environments with moving obstacles. BAT-LbCS addresses these issues by leveraging the strengths of both BAT optimization and LbCS, creating a robust and efficient solution for collision-free navigation. The BAT optimization algorithm is used to determine the next step by generating a point that avoids static obstacles. Meanwhile, the LbCS method ensures safe movement from the current location to the new location, effectively handling dynamic obstacles. This dual-avoidance mechanism is achieved through a feedback loop where the output of the BAT algorithm (the coordinates of the next step) is continuously fed into the LbCS. The process iterates until the robot reaches its target destination. The proposed algorithm's effectiveness was validated through two case studies involving a point-mass robot navigating environments with varying numbers of static and dynamic obstacles. Results demonstrated the robot's ability to successfully avoid both types of obstacles and reach its destination. Additionally, the algorithm was applied to a car-like robot, which also navigated successfully, further proving the versatility and robustness of BAT-LbCS. This hybrid algorithm holds significant potential for application in various dynamic environment settings, such as retail, healthcare, and agriculture. By providing an optimized and safe planned path with effective motion control, BAT-LbCS offers a substantial advancement in the field of robotic navigation.

Keywords: Heuristic; Classical; Hybrid; Robot; Path; Planning.

Received: 15 February 2024; Revised: 21 July 2024; Accepted: 21 August 2024.

Article type: Research article.

1. Introduction

Currently, several areas are using robots for certain tasks. For example, a robot can be assigned to carry out surgery in healthcare industry, improve drilling accuracy in mining, navigate through flooded areas in transportation, and package goods in the manufacturing industry, to name a few.^[1-5] Completing a task requires a robot to move from one location to another while avoiding obstacles. This research area is known as robot navigation. Robot navigation is a critical area of research due to its potential to revolutionize various industries and aspects of our lives. The ability of robots to autonomously navigate in complex environments has far-reaching implications, ranging from enhancing productivity and safety to enabling new and innovative applications.

A robot navigation problem can be solved by classical, heuristic, and machine learning approaches. Examples of classical methods include artificial potential field,^[6] cell

decomposition,^[7] road map,^[8] and virtual force field.^[9] There are many heuristic algorithms available that have been used for robot path planning.^[10-12] Likewise, machine learning has also been used for robot navigation.^[13-15] Recent algorithms are mostly hybrids of the above, which extract the advantages from both methods involved in the hybrid. This research considers a hybrid of an artificial potential field method, a Lyapunov-based Control Scheme (LbCS), and a heuristic method, BAT optimization algorithm.

LbCS is a popular method that has been used for robot motion control.^[16-18] The BAT optimization algorithm was proposed by Yang in 2010 for solving global optimization problems.^[19] This algorithm has been tested in different test functions and compared with other algorithms, which was the basis for selecting it in this research.

This paper presents a hybrid algorithm, a combination of the BAT optimization algorithm and the LbCS method, with the aim of avoiding both static and dynamic obstacles. The static obstacles are avoided using the BAT optimization algorithm, which plans the next step for a robot. The LbCS method governs the motion of the robot from one location to

School of Information Technology, Engineering, Mathematics & Physics, The University of the South Pacific, Suva 1168, Fiji.

*Email: kaylash.chaudhary@usp.ac.fj (K. Chaudhary)

another while avoiding dynamic obstacles. This algorithm will handle the situation where a step is planned without considering dynamic obstacles that can lead to collision. LbCS can also avoid static and dynamic obstacles, but drawbacks include getting trapped in local minima and slow convergence. Heuristic algorithms can only do path planning. Therefore, a motion control algorithm is needed to govern the motion of a robot. Given the limitations and strengths of each method, a hybrid approach is required. This algorithm can be applied to any robot and in any scenario where robot motion control is needed. This paper applies the proposed algorithm to point-mass and car-like robots in different environmental complexities.

The main contributions of this paper are:

- **BAT-LbCS algorithm:** The paper proposed a new hybrid method consisting of a heuristic and a classical method. According to the authors' knowledge, there is no hybrid algorithm like the one proposed in the literature. A hybrid algorithm, ACO-Kinematic,^[20] combines heuristic and classical methods and only avoids static obstacles. The proposed algorithm also avoids dynamic obstacles, a drawback of ACO-Kinematic. A classical method, LbCS, can also avoid dynamic obstacles, but it has problems of slow convergence and getting trapped in local minima, which are overcome by the proposed hybrid algorithm.
- **Application:** The proposed algorithm has been applied to a car-like robotic system, which shows its applicability in real-world situations.

2. Related work

Path planning and motion control is a well-researched subarea in the wider area of robotics and robot navigation.^[21-24] There are many instances where researchers have implemented path planning and motion control algorithms in solving robot navigation problems. This section will outline these algorithms and provide a brief comparison with the proposed algorithm.

First of all, the literature has several examples of those algorithms that carry out the path planning first and then use motion control algorithms for the robot navigation on the planned path. This method is usually referred to as path-tracking. For example, Wahhab *et al.*^[25] proposed the use of the Chaotic Particle Swarm Optimization (CPSO) algorithm, A-star algorithm, and a hybrid swarm optimization algorithm (ACPSO) and Convolutional Neural Network Trajectory Tracking (CNNTT) controller in order to control the nonlinear kinematics mobile robot system. The CNNTT controller was used to obtain the robust velocity of the left and right wheels, which are used to control the position and orientation of the mobile robot system. In another project, the researchers developed an automated robotic system of path planning and motion control to handle large components in large workspaces where high precision was required.^[26] An integrative path planning and motion control concept was developed, which simplified the commissioning of handling

systems consisting of different robots. Nejad *et al.* proposed a novel practical method for autonomous parking in a cluttered environment with narrow passages.^[27] In the project, the authors proposed a path planning method, which is based on predefined arc paths for obstacle avoidance, and control commands were generated to minimize the steering angle control error, which were executed by the vehicle actuators. Chaudhary *et al.*^[28] proposed a new hybrid system of Ant Colony Optimization algorithm and kinematic equations for path planning and motion control, respectively. This system successfully navigated from the initial point to the target in a workspace cluttered with static obstacles.

LbCS is a control scheme proposed by Sharma *et al.*^[29,30] that revolves around the governing principle of associating an attractive field with the target and a repulsive field with each obstacle. The fundamental concept driving this control scheme is the crafting of a suitable Lyapunov function, serving as an energy function. This Lyapunov function comprises the cumulative sum of all attractive and repulsive potential functions. There are various examples where Lyapunov based schemes have been used by researchers for motion control of robots. Panahandeh *et al.* used Lyapunov-like stability technique for the posture stabilization of a differentially driven wheeled mobile robot.^[31] Razzaghian investigated a novel fractional-order Lyapunov-based robust controller based on a fuzzy neural network (FNN) compensator for exoskeleton robotic systems.^[32] The author proved the finite-time stability of the closed-loop control system using the Lyapunov stability theorem, and adaptive law was derived through it. Amiri and Ramli presented a modeling, dynamic analysis, and controller tuning comparison for a five-degree-of-freedom (DoF) multi-joint robotic arm based on the Lyapunov-based Adaptive Controller (LAC).^[33] The authors designed a control law using Lyapunov theory and feedback linearization techniques to navigate a group of follower robots in a desired formation associated with the leader for the control of underwater robots.^[34]

Meta-heuristic algorithms are very prominent in robot path planning.^[21,35-38] Kundra *et al.* proposed an integrated concept of quantum-inspired firefly algorithm with cuckoo search (IQFACS) that adapts both algorithms' expedient attributes to optimize the solution set for robot path planning.^[39] Chand *et al.* used the stepping-ahead firefly algorithm for the optimization of robot path planning with static robots and landmarks.^[40] Ju *et al.* proposed an improved A-star algorithm by combining it with the jump point search algorithm for optimized path planning.^[41] In another research, the authors of proposed an improved bat algorithm with a logarithmic decreasing strategy and Cauchy disturbance. They met the requirements of global optimal and dynamic obstacle avoidance in path planning for a mobile robot by combining the bat algorithm (BA) with the dynamic window approach (DWA).^[42] In response to the issues of suboptimal exploration efficiency and elevated optimal solution costs in current robot path planning methods, Qian *et al.* introduced an optimization

technique.^[38] This method was grounded in a heuristic multi-directional rapidly-exploring tree. Wang and Zhou proposed a heuristic elastic particle swarm algorithm that used the path planned by the A* algorithm in a large-scale grid for global guidance.^[43] The elastic particle swarm optimization algorithm used a shrinking operation to determine the globally optimal path formed by locally optimal nodes so that the particles could converge rapidly. It was also able to overcome the shortcomings of the A* algorithm.

The BAT algorithm is one of the prominent meta-heuristic algorithms used by researchers to solve various path-planning problems.^[44-47] In [48], the authors proposed a flight path planning for unmanned aerial vehicles (UAVs) based on the advanced swarm optimization algorithm of the bat algorithm (BA) in a static environment to obtain an accident-free, shorter, and safer flight path between the starting point and the endpoint in the complex three-dimensional battlefield environment. Chen *et al.* used the bat algorithm to investigate UAV flight planning in a static complex environment eight and proposed an improved spherical coordinate and truncated average stable strategy-based bat optimization algorithm. Gunji *et al.* developed a solution for the mobile robot path planning problem by hybridizing two meta-heuristic algorithms, cuckoo-search and bat algorithm (BA), in an unknown or partially known environment.^[49]

One of the drawbacks of meta-heuristic algorithms is that they are not very effective in optimizing dynamic obstacle avoidance environments.^[50] In [51], the authors mentioned that the heuristic algorithms that deal with configuration space work on the assumption that the shape and position of the obstacles are known to the mobile robot. They mention that these algorithms are not applicable when the environment is uncertain, or the map is unavailable during path planning. Yahia *et al.* also noted that heuristic algorithms alone are ineffective in dealing with dynamic environments.^[52] Yuan *et al.* used a hybrid system of dynamic window approach with a bat algorithm for global optimal and dynamic obstacle avoidance in path planning for a mobile robot. They claimed it performed better than the particle swarm optimization algorithm and the basic bat algorithm.^[42] This is one of the reasons why researchers create hybrid algorithms combining meta-heuristic algorithms with nonlinear control designs like the Lyapunov-Based Control Scheme (LbCS), a system that can interact with and manipulate their environments. The literature has several examples of such hybrid algorithms. Sang *et al.* presented a novel deterministic algorithm based on an improved Artificial Potential Field (APF) and local path planning algorithm using an improved heuristic A* algorithm.^[53] Chen *et al.* proposed an improved ant colony optimization-artificial potential field (ACO-APF) algorithm based on a grid map for local and global path planning of Unmanned Surface Vehicles (USVs) in dynamic environments.^[54] The ACO mechanism searched for a globally optimal path, and the improved artificial potential field algorithm helped avoid unknown obstacles during USV

navigation. Table 1 shows the objectives, strengths, and drawbacks of various hybrid algorithms that solve path-planning problems.

3. BAT optimization algorithm

The standard bat algorithm is based on the echolocation or bio-sonar characteristics of microbats.^[61] Microbats typically use a type of sonar called echolocation to detect prey, avoid obstacles, and locate their roosting crevices in the dark. They can emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects.^[62] The pulses primarily last a few thousandths of a second with the frequency range of 25 kHz to 150 kHz. Though, in reality, microbats can also use time delay between their ears and loudness variations to sense three-dimensional surroundings, the developers of the BAT algorithm were mainly interested in those features of the echolocation that can link them with the objective function of an optimization problem, which makes it possible to formulate a smart bat algorithm.^[61]

The bat algorithm generally follows the following rules:

1. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way;
2. Bats fly randomly with velocity \mathbf{v}_i at position \mathbf{x}_i with a fixed frequency f_{min} , varying wavelength λ , and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0,1]$, depending on the proximity of their target;
3. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min} .

The core principles of the BAT algorithm include:

1. Echolocation: Echolocation is used by the algorithm to explore the solution space. Bats emit pulses (representing potential solutions) and adjust their frequency and loudness based on the quality of solutions;
2. Frequency and Loudness: The frequency and loudness determine the ability of the algorithm to explore and exploit. Higher loudness values increase the probability of the bat's solution being accepted, while frequency determines the step size for exploration;
3. Velocity: Bats investigate the solution space by random walks, and velocity is added to allow them to adjust positions in response to the quality of the solutions;
4. Optimization: Bats that possess better solutions have the ability to attract others, resulting in a collective behavior known as swarm behavior, which aids in the process of converging toward the optimal solutions.

In simulations, we use virtual bats naturally. We have to define the rules of how their positions \mathbf{x}_i and velocities \mathbf{v}_i in a d -dimensional search space are updated. The new solutions \mathbf{x}_i^t and velocities \mathbf{v}_i^t at time step t are given by:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$\mathbf{v}_i^t = \mathbf{v}_i^{t-1} + (\mathbf{x}_i^t - \mathbf{x}_*)f_i \quad (2)$$

Table 1. The objectives and strengths/drawbacks of various path planning hybrid algorithms.

Workspace/Objective	Algorithm/Approach	Strength/Drawbacks
Dynamic path planning and motion control for microrobotic swarms to track a mobile target avoiding obstacle collisions with sufficient updating frequency in an environment with virtual obstacles and a micromaze with virtual walls. ^[55]	Hybrid - Enhanced bidirectional rapidly-exploring random tree star (EB-RRT*) algorithm	Strengths: Dynamically plan the optimal path for obstacle avoidance. The image-guided motion control algorithm including the direction control algorithm and the GA-LQR based velocity control algorithm is used to control the moving direction and velocity of swarms which guarantees the small distance error during the tracking process.
An ant-colony-based path planning algorithm that is applicable to scenic areas. The proposed algorithm modifies the ants' ending tour to achieve partial point traversal of the connected graph by eliminating the restriction of the ant colony algorithm taboo table using a temporary weight matrix. ^[56]	Improved Ant Colony Algorithm	Strengths: Avoids getting trapped in local minima Drawbacks: Functions effectively for just one scenic location at a time
Uses the hybrid of the Dynamic Window Approach (DWA) and the Teaching-Learning-Based Optimization (TLBO) technique to implement navigation on the NAO humanoid robot. ^[57]	Hybrid [Dynamic Window Approach (DWA) + Teaching-Learning-Based Optimization (TLBO) technique]	Strengths: Ability to avoid dynamic obstacles Avoids collision in a multi-robot system Drawbacks: Functions optimally on flat surfaces only
Path planning of an autonomous mobile robot using a modified bat algorithm in a time-varying situation to acquire the shortest path and avoid dynamic obstacles. ^[58]	Modified Frequency Bat (MFB) algorithm	Strengths: Ability to avoid dynamic obstacles using sensory vector information Developed and utilizes a frequency function instead of randomization to determine exploration and exploitation
Develops path planning algorithm involving point generation using hybridization between Particle Swarm Optimization and Modified Frequency Bat (PSO-MFB) Algorithm. ^[59]	Hybrid PSO-MFB algorithm	Strengths: Integrates a hybrid of two swarm-based algorithms with a linear search algorithm and utilizes 12 obstacle-detection sensors Capable of generating smooth paths for the robot Drawbacks: Simulation operates solely on stationary obstacles.
Multi-objective robotic path planning algorithm hybridizing two meta-heuristics techniques (GWO and PSO). It uses a frequency-based modification in GWO search operators to fasten the search process and improves the search strategy for collision detection and avoidance. ^[60]	Hybrid [Grey Wolf Algorithm (GWO) + Particle Swarm Optimization (PSO)]	Strengths: Avoids getting trapped in local minima Reduced computational costs when compared with other similar algorithms in the literature Ability to work on dynamic obstacle workspace

$$\mathbf{x}_i^t = \mathbf{x}_i^{t-1} + \mathbf{v}_i^t \quad (3)$$

where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. Here \mathbf{x}_* is the current global best location (solution), which is located after comparing all the solutions among all the n bats.^[63]

Once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \epsilon A^t \quad (4)$$

where $\epsilon \in [-1,1]$ is a random number, while $A_t = \langle A_i^t \rangle$ is the average loudness of all the bats at this time step.

The loudness A_i and the rate r_i of pulse emission are updated as follows as the iterations proceed:

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (5)$$

where α and γ are constants.

4. Proposed algorithm

The proposed algorithm is a hybrid of a heuristic and a

classical method. The heuristic method is the BAT optimization algorithm that is used to avoid static obstacles only. The classical method is LbCS, which is used to govern the motion of the robot from one point to another while avoiding dynamic obstacles. The hybrid algorithm is known as BAT-LbCS. The algorithm is divided into the following phases:

4.1 Initialization phase

The algorithm presented in Table 2 starts with the Euclidean distance from the robot's initial position to the target, as shown in equation (6).

$$Dist_{RT} = \sqrt{(T_1 - R_1)^2 + (T_2 - R_2)^2} \quad (6)$$

where $Dist_{RT}$ is the distance between the robot and the target, (T_1, T_2) is the coordinate of the target, and (R_1, R_2) is the coordinate of the robot.

The algorithm will continue executing until $Dist_{RT}$ becomes 0. The next steps (Steps 2 to 4) are the initialization of the BAT population with the update of each bat's position and fitness value.

4.2 Solution construction phase

Then, the main loop starts with the calculation of the new position and velocity of each bat (Steps 5 to 9). The new position is evaluated using a fitness function. The fitness value of the new position of the bat is compared with the old fitness value of the same bat. The position will be updated if the fitness value is less than the old one and a random value is greater than the loudness (Steps 10 to 12). The global best solution will be updated if the fitness value of the new solution is less than the fitness of the global best solution.

The position of the best bat will be the coordinates of the robot's new position, which will be used by the LbCS method to move the robot from the previous point to the latest point (Steps 14 to 16). This process will continue unless the robot reaches the new point, which is managed by the Euclidean distance between the robot and the new point, as shown in equation (7).

$$Dist_{RP} = \sqrt{(P_1 - R_1)^2 + (P_2 - R_2)^2} \quad (7)$$

where $Dist_{RP}$ is the distance between the robot and the new point, (P_1, P_2) is the coordinate of the target, and (R_1, R_2) is the coordinate of the robot.

The point generated is through the minimization of the fitness function. In this research, the fitness function comprises a short and safe path. The safe path is calculated using the Euclidean distance between each bat and an obstacle and is represented by:

$$d_{ik} = \sqrt{(o_{k1} - xb_i)^2 + (o_{k2} - yb_i)^2} - rad_k \quad (8)$$

where d_{ik} is the distance between i th bat and the obstacle k , (o_{k1}, o_{k2}) is the coordinate of the k th obstacle, (xb_i, yb_i) is the coordinate of the bat i and rad_k is the radius of obstacle k .

This work focuses on an environment that has multiple static obstacles. Therefore, the distance between each bat i and multiple obstacles is summed as:

$$f_{ik} = \sum_{k=1}^q d_{ik} \quad (9)$$

The short path is the Euclidean distance between a bat and the target, as shown in equation (10).

$$d_{it} = \sqrt{(xb_i - xt)^2 + (yb_i - yt)^2} \quad (10)$$

where d_{it} is the distance between the i th bat and the target, (xt, yt) are the coordinates of the target and (xb_i, yb_i) are the coordinates of the i th bat.

The point generated by the BAT-LbCS algorithm is the point where a bat is positioned at a short distance from the target and is safe from collisions with obstacles. To determine the best point in this minimization problem, the fitness of each bat i is calculated as shown in equation (11).

$$f_i = \frac{a}{f_{ik}} + b \cdot d_{it} \quad (11)$$

where a and b are control parameters.

The control parameters a and b are adjusted using brute force technique, and a proper combination represents the position of a bat at a safe and short distance from the target. Obstacle avoidance (safe) is dependent on the parameter a , while a short path is determined by parameter b . When the parameter a , if set to a high value, the bats can detect and avoid stationary circular obstacles from a larger distance. Nevertheless, a decrease in the value of a will significantly increase the likelihood of collisions with obstacles. Similarly, increasing the value of b will minimize the length of the path, while decreasing the value would maximize the length of the path. The parameters range is discussed in Table 2.^[28]

4.3 Motion control phase

This phase will use the point generated in the previous phase to move the robot from its previous location to the new location. While moving, the robot will avoid any dynamic obstacle in its way as follows:

A point-mass robot moving with a velocity of $v = u_1\mathbf{i} + u_2\mathbf{j}$ in the xy -plane has the following kinematic equations:

$$\begin{cases} \dot{x} = u_1 \\ \dot{y} = u_2 \end{cases} \quad (12)$$

Let there be n moving obstacles in the robot's workspace. Assume that the i th moving obstacle is a disk of radius rp_i centered (p_{i1}, p_{i2}) and is moving with a constant velocity of $vp_i = s_{i1}\mathbf{i} + s_{i2}\mathbf{j}$. Then the position of the i th moving obstacle at any time t can be calculated as

$$\begin{cases} p_{i1}(t) = p_{i1}(0) + s_{i1}t \\ p_{i2}(t) = p_{i2}(0) + s_{i2}t \end{cases} \quad (13)$$

For the point-mass robot to move from an initial position $(x(0), y(0))$ to another point (a, b) , and avoid moving obstacles along its path, we define a Lyapunov function (energy function) for system (12) as

$$V(x, y) = (x - a)^2 + (y - b)^2 + \sum_{i=1}^n \frac{\beta_i[(x-a)^2 + (y-b)^2]}{(x-p_{i1})^2 + (y-p_{i2})^2 - rp_i^2} \quad (14)$$

where $\beta_i > 0$ is a positive constant, known as control parameter [64]. Using the Lyapunov-based control scheme (LbCS), the velocities u_1 and u_2 are derived as equation 15.

Table 2. BAT-LbCS Algorithm

<p>Step 1: Calculate distance between robot and target (DistRT) while DistRT > 0 do Step 2: Initialize bat population, loudness, frequency, gamma, pulse rate and alpha foreach i = 1 : n (all n Bat) do Step 3: Calculate the position of each bat within certain distance of the robot Step 4: Calculate the fitness of each solution (fSol) end while it ≤ ITmax do Step 5: Calculate pulse rate (r) using gamma and it Step 6: Calculate loudness (A) based on alpha and initial loudness foreach i = 1 : n (all n Bat) do Step 7: Generate random frequency Step 8: Calculate the new position of each bat Step 9: Update the velocity of each bat Step 10: Evaluate the fitness of new position (solution) if fNewSol ≤ fSol && rand > A then Step 11: Update the position of ith bat with the new position Step 12: Update the fitness of ith bat with the new fitness end if fNewSol < fbest then Step 13: Update the current best solution (generated point) end end end Step 14: Calculate distance between robot and generated point (DistRP) while DistRP > 0 do Step 15: Move the robot to the best location using LbCS and avoid any dynamic obstacles Step 16: Update distance between robot and generated (DistRP) end Step 17: Update distance between robot and target (DistRT) end</p>

$$\begin{cases} u_1 = -\frac{1}{\alpha_1} \frac{\partial V}{\partial x} \\ u_2 = -\frac{1}{\alpha_2} \frac{\partial V}{\partial y} \end{cases} \quad (15)$$

where $\alpha_1 > 0$ and $\alpha_2 > 0$ are called convergence parameters.

Stability Theorem 1: When the point-mass robot moves from an initial position $(x(0), y(0))$ to another point (a, b) , its motion described by system the (12) is stable.

Proof: The Lyapunov function $V(x, y)$ given in (14) is continuous and has continuous partial derivatives for all $(x, y) \in \mathbf{R}^2$. It is clear that $V(x, y) > 0$ for all $(x, y) \neq (a, b)$ and $V(a, b) = 0$. Moreover,

$$\dot{V}(x, y) = \frac{\partial V}{\partial x} u_1 + \frac{\partial V}{\partial y} u_2 = -\alpha_1 u_1^2 - \alpha_2 u_2^2 \quad (16)$$

Since α_1 and α_2 are positive real numbers, it follows that $\dot{V}(x, y) \leq 0$. Thus, we conclude that the motion of the point mass robot described by the system (12) is stable.

Figure 1 shows the flowchart of the proposed algorithm with the place where the BAT algorithm and LbCS method have been integrated.

In summary, the three-phase hybrid algorithm, which

consists of the BAT optimization algorithm and LbCS, avoids both static and dynamic obstacles. The BAT optimization algorithm avoids static obstacles by generating points (a location for a robot to move to) at a minimum distance from an obstacle and the target. The LbCS method governs the robot's motion to that location and will avoid any dynamic obstacle.

5. Results

5.1 Case studies

The proposed algorithm was applied to two case studies. The first case study consists of a point mass robot, seven static obstacles of different sizes, and one dynamic obstacle. Fig. 2 shows the point mass robot moving towards the target location (5, 45) from the initial location (45, 5). It also shows that the proposed method generates points, and the point mass robot moves from one point to another. The BAT optimization algorithm generates the points, and the robot moves from one point to another using the LbCS method, which also avoids the dynamic obstacle (in blue). The obstacle initially positioned at (16,16) is moving at a constant velocity of $vp = 0.03i +$

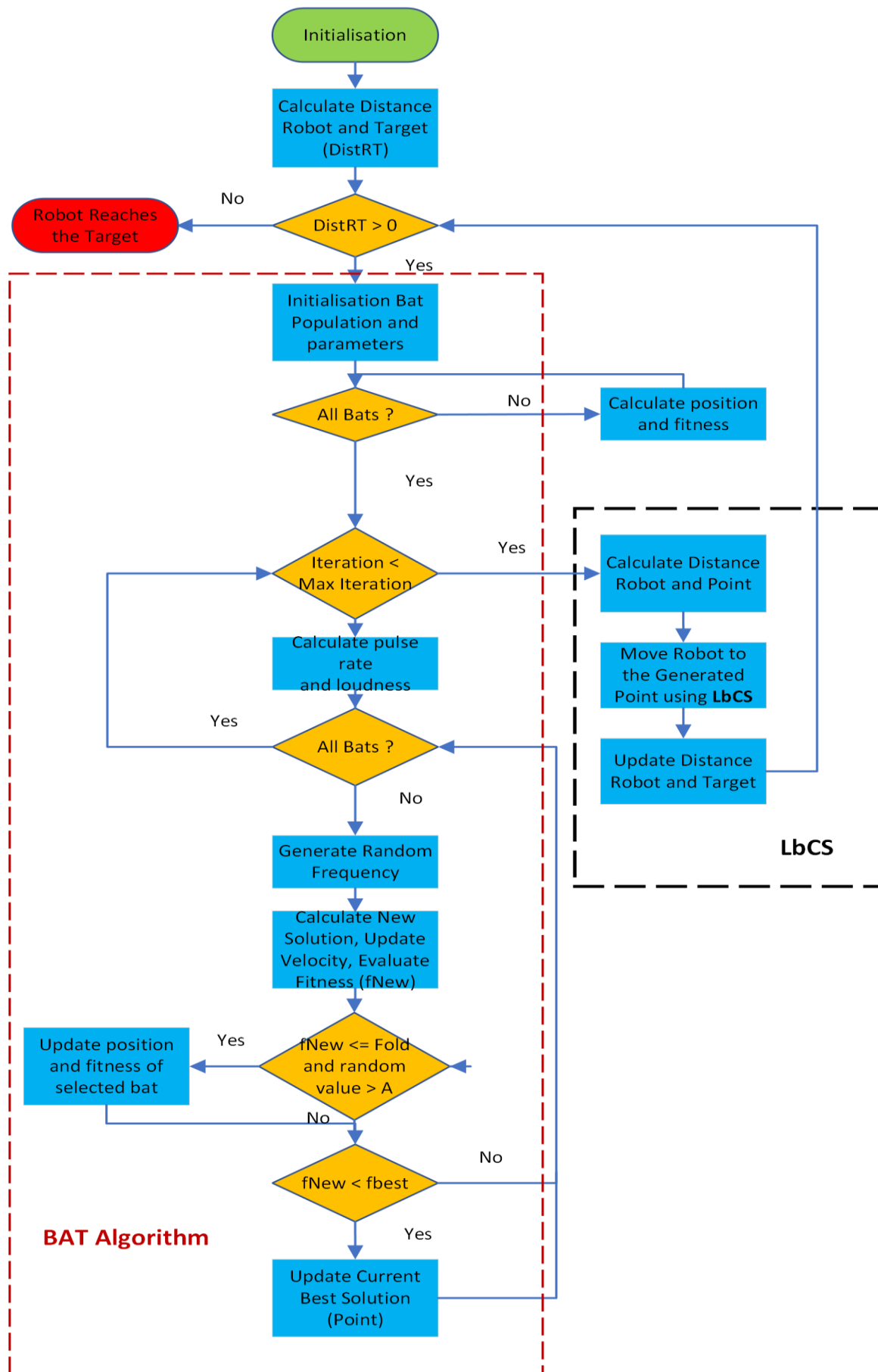


Fig. 1 Flowchart of the proposed algorithm.

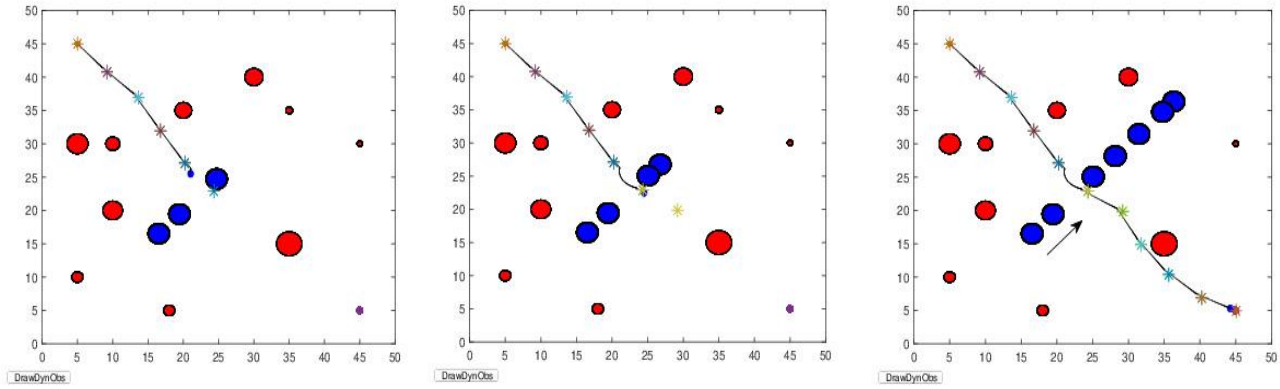


Fig. 2 Robot path with initial location (5, 45) and goal location (45, 5) before and after dynamic obstacle avoidance.

0.03j. Fig. 2b shows that the point mass robot has successfully avoided the dynamic obstacle. Likewise, Fig. 2c shows that the point mass robot has safely reached the target. The point mass robot took 187.82s to reach the target with a path length of 57.78 cm.

In the second case study, a point mass robot avoids multiple static obstacles (eleven obstacles of different sizes) and multiple dynamic obstacles (two dynamic obstacles). Fig. 3a shows the point-mass robot avoiding the static obstacles and the first dynamic obstacle on its path from the initial location (-5, -5) to the target location (45, 45). The first dynamic obstacle moves from location (30, 0) at a constant velocity of $vp_1 = -0.07i + 0.07j$. Fig. 3b shows the point-mass robot avoiding the second dynamic obstacle that is moving at a constant velocity of $vp_2 = 0.03i - 0.03j$ from location (20, 45). Fig. 3c shows the robot's full path. The path length was 94.06 cm, and the time was 146.65 s. Overall, the point mass robot successfully avoided the static and dynamic obstacles in both case studies.

5.2 Application

Consider a 2-dimensional rear wheel-driven car-like robot (adopted from [64]) as shown in Fig. 4. The car-like robot considered in this research is subject to non-holonomic constraints, which are due to the pure rolling and non-slipping nature of the wheels.^[64] Referring to the figure, (x, y) denotes the center of mass (CoM) of the robot, θ denotes its orientation with respect to the z_1 -axis, while ϕ gives the steering wheel's angle with respect to the car's longitudinal axis.

Let L be the distance between the two axles and l the length of each axle, as shown in Fig. 4, then the kinematic model of the car-like robot adopted from [64] is given by

$$\begin{cases} \dot{x} = v \cos \theta - \frac{L}{2} \omega \sin \theta \\ \dot{y} = v \sin \theta + \frac{L}{2} \omega \cos \theta \\ \dot{\theta} = \omega \end{cases} \quad (17)$$

where v and ω are the translational and rotational velocities of the car-like robot. For the car-like robot to move from an initial position $(x(0), y(0))$ to another point (a, b) , and avoid moving obstacles along its path, we define a Lyapunov function for system (17) as

$$V(x, y) = (x - a)^2 + (y - b)^2 + \sum_{i=1}^n \frac{\beta_i [(x - a)^2 + (y - b)^2]}{(x - p_{i1})^2 + (y - p_{i2})^2 - (rp_i + r_c)^2} \quad (18)$$

where r_c is the radius of the smallest possible circle that centered at (x, y) that encloses the car-like robot (as shown in Fig. 4). Using LbCS, the translational and rotational velocities are derived as

$$\begin{cases} v = -\alpha_1 \left(\frac{\partial V}{\partial x} \cos \theta + \frac{\partial V}{\partial y} \sin \theta \right) \\ \omega = -\alpha_2 \left(\frac{\partial V}{\partial y} \cos \theta \pm \frac{\partial V}{\partial x} \sin \theta \right) \end{cases} \quad (19)$$

where $\alpha_1 > 0$ and $\alpha_2 > 0$ are the convergence parameters [sharma08d].

Stability Theorem 2: When the car-like robot moves from an initial position $(x(0), y(0))$ to another point (a, b) , its motion described by system (17) is stable.

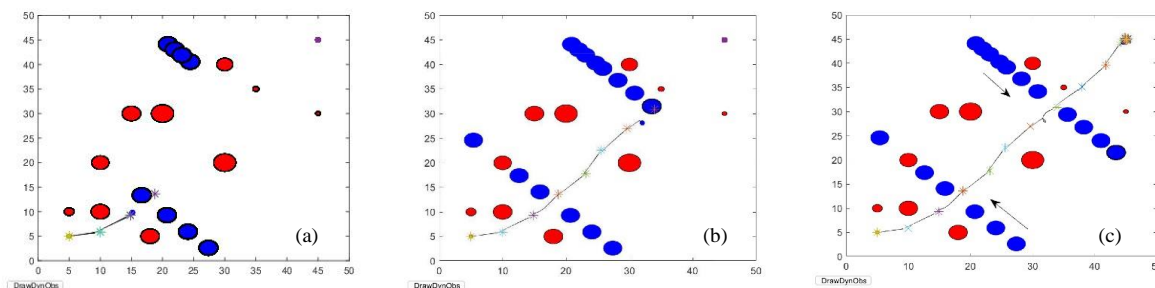


Fig. 3 Robot path with initial location (-5, -5) and goal location (45, 45) before and after dynamic obstacle avoidance.

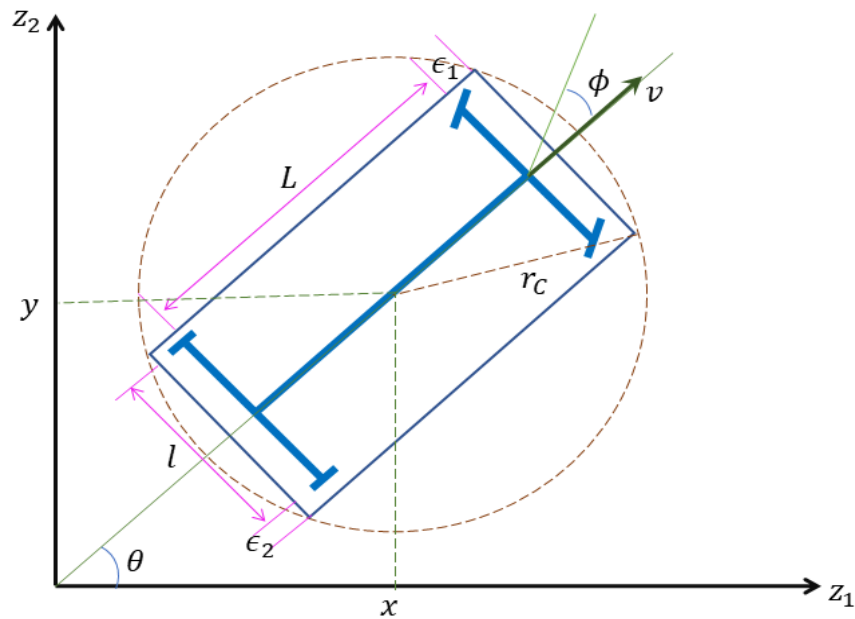


Fig. 4 A rear wheel driven vehicle. Reproduced with the permission from [30].

Proof: The Lyapunov function $V(x, y)$ given in (18) is continuous and has continuous partial derivatives for all $(x, y) \in \mathbb{R}^2$. Moreover, $V(x, y) > 0$ for all $(x, y) \neq (a, b)$ and $V(a, b) = 0$. Next, note that:

$$\dot{V}(x, y) = \frac{\partial V}{\partial x} \dot{x} + \frac{\partial V}{\partial y} \dot{y} + \frac{\partial V}{\partial \theta} \dot{\theta} \quad (20)$$

Substituting (17) and (19) into (20) and simplifying:
 $\dot{V}(x, y) = -\frac{1}{\alpha_1} v^2 - \frac{1}{\alpha_2} \omega^2 \leq 0$.

Hence, the motion of the car-like robot described by the system (17) is stable.

Figure 5a shows the path of the car-like robot while avoiding the first dynamic obstacle. The first dynamic obstacle is moving from the initial location (37, 0) at a constant velocity of $vp_1 = -0.07i + 0.07j$. Fig. 5b shows the path of the car-like robot while avoiding the first dynamic obstacle. The second dynamic obstacle is moving from the initial location (40, 30) at a constant velocity of $vp_1 = -0.03i - 0.03j$. The full path of the car-like robot is shown in Fig. 5c. The robot took 236.15 s to reach the target with a path length of 68.64 cm.

Overall, the car-like robot reached the target by avoiding static and dynamic obstacles. In real-world navigation scenarios, the obstacles are of different types and sizes. This algorithm only avoids circular obstacles, and a potential improvement would be to include different types of obstacles for avoidance.

6. Discussion

Figure 6 shows the cost function graph for case study 1. The BAT optimization algorithm uses the objective function denoted as the cost function to determine the next best step for a robot. Every step calculation has a cost value associated with it. The cost value will be higher at the robot's initial point, which is farther from the target, than at a position where the robot is closer to the target. The robot travels between points to reach the target, consisting of 15 points, as shown in Fig. 2. Robot path with initial location (5, 45) and goal location (45, 5) before and after dynamic obstacle avoidance. The cost function graph shows the cost associated with each point generation.

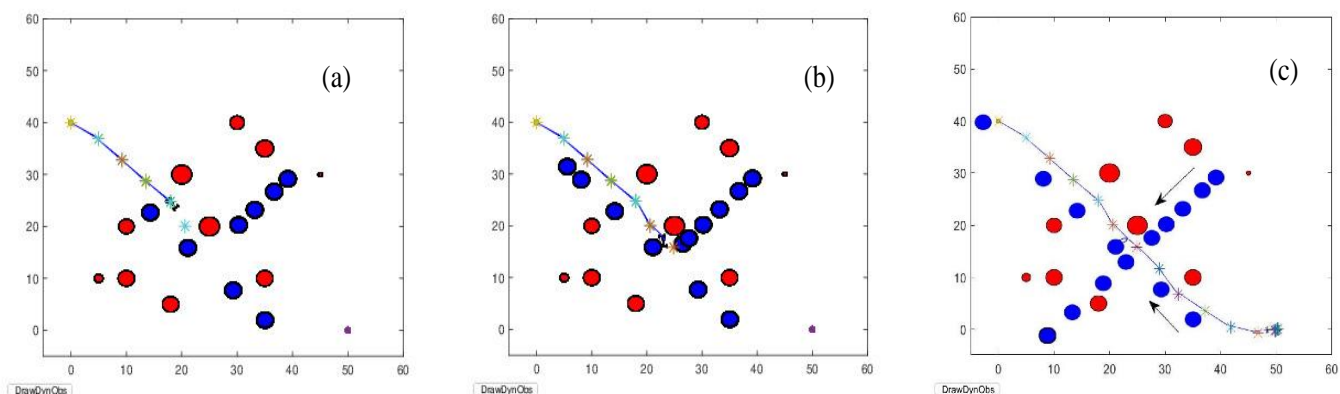


Fig. 5 Robot path with initial location (0, 40) and goal location (50, 0) before and after dynamic obstacle avoidance.

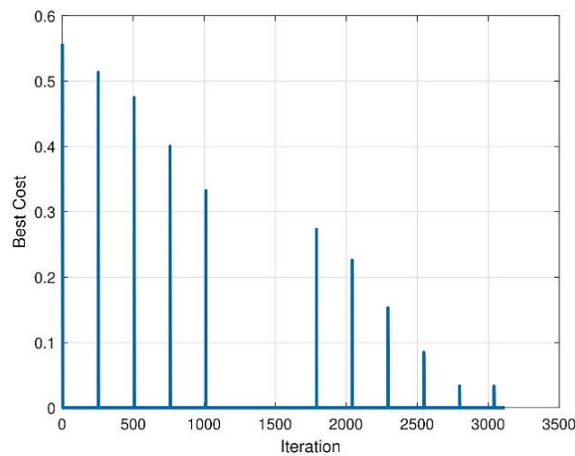


Fig. 6 The cost graph for case study 1.

Figure 7 shows the convergence of controllers (\dot{x} and \dot{y} of system (12)) graph for case study 1. The correlation between the decreasing and increasing portions of the dot x and y represents the robot's motion from one generated point to another. An illustration of this can be seen in Fig. 7, which depicts the robot's path from the initial point to the first generated point from approximately $t = 0$ to approximately $t = 250$ units. The only dynamic obstacle in the case study is circumvented by the robot between $t = 1000$ and $t = 1750$. Likewise, the convergence of \dot{x} and \dot{y} occurs when $t = 2750$ units have passed; this convergence pertains to the robot's path from the final generated point to the target. In general, the graph illustrates that the proposed algorithm converges successfully, as evidenced by the robot's safe arrival at the target.

The above two graphs show that the algorithm is able to avoid static and dynamic obstacles and converge successfully.

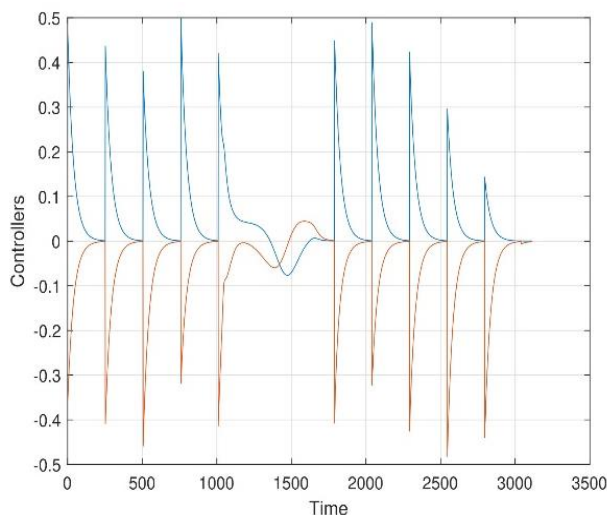


Fig. 7 The convergence of controllers graph for case study 1.

7. Conclusion

This paper introduced a novel hybrid algorithm combining the LbCS method and the BAT optimization algorithm for robot navigation in environments with both static and dynamic

obstacles. The hybrid algorithm involves two main components: planning a step for the robot and motion control of the robot to that step. The BAT optimization algorithm plans the next step for a robot, while the LbCS method governs its motion to that step. While planning the next step, the BAT optimization algorithm avoids only static obstacles in the robot's path. The LbCS method avoids dynamic obstacles.

The hybrid algorithm has been tested on two case studies consisting of a point-mass robot and different numbers and sizes of static and dynamic obstacles. The robot was able to reach its destination successfully. The proposed algorithm has also been applied to a car-like robotic system. The car-like robot was able to avoid static and dynamic obstacles to reach the target. Overall, in both case studies and the application, the robots were able to avoid multiple static and dynamic obstacles to reach the target by taking a shorter and safer route. The application of the algorithm in a mechanical system suggests that the algorithm is not only for a point-mass robot but can also be used in different mechanical systems.

In future work, the authors will test the algorithm in more complex environments with multiple robots and various types of obstacles. Also, the authors will perform experimental verification and compare with the simulated result. Experimental verification is essential for validating the algorithm which has been only used in simulations, optimizing performance, ensuring safety, and improving the overall functionality of robotic systems in real-world applications. The authors will also conduct experiments on optimizing the current system to consider the speed of an obstacle during avoidance.

In conclusion, the proposed algorithm for robot navigation represents a significant advancement in the field. It addresses key challenges and offers valuable contributions to real-world applications. The algorithm's importance lies in its ability to enhance the efficiency, adaptability, and safety of robot navigation in dynamic environments.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

- [1] M. Dunbabin, L. Marques, Robots for environmental monitoring: significant advancements and applications, *IEEE Robotics & Automation Magazine*, 2012, **19**, 24-39, doi: 10.1109/MRA.2011.2181683.
- [2] J. Khurshid, B.-R. Hong, Military robots - a glimpse from today and tomorrow, ICARCV 2004 8th Control, Automation, Robotics and Vision Conference. Kunming, China. IEEE, 2004.
- [3] R. Müller, M. Esser, M. Janssen, Integrative path planning and motion control for handling large components. *Lecture Notes in Computer Science*, 2011, 93-101, doi: 10.1007/978-3-642-

25486-4_10.

- [4] S. A. Napper, R. L. Seaman, Applications of robots in rehabilitation, *Robotics and Autonomous Systems*, 1989, **5**, 227-239, doi: 10.1016/0921-8890(89)90047-x.
- [5] J. Raj, K. Raghuwaiya, J. Vanualailai, B. Sharma, Navigation of car-like robots in three-dimensional space, 2018 5th Asia-Pacific World Congress on Computer Science and Engineering, 2018.
- [6] M. C. Lee, M. G. Park, Artificial potential field based path planning for mobile robots using a virtual obstacle concept, Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003). Kobe, Japan. IEEE, 2003.
- [7] M. Kloetzer, C. Mahulea, R. Gonzalez, Optimizing cell decomposition path planning for mobile robots using different metrics, 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), 2015.
- [8] F. Lingelbach, Path planning using probabilistic cell decomposition, IEEE International Conference on Robotics and Automation, 2004.
- [9] S. A. Bortoff, Path planning for UAVs, American Control Conference, 2000.
- [10] M. M. Gangadharan, A. Salgaonkar, Ant colony optimization and firefly algorithms for robotic motion planning in dynamic environments, *Engineering Reports*, 2020, **2**, e12132, doi: 10.1002/eng2.12132.
- [11] H. Kundra, W. Khan, M. Malik, K. P. Rane, R. Neware, V. Jain, Quantum-inspired firefly algorithm integrated with cuckoo search for optimal path planning, *International Journal of Modern Physics C*, 2022, **33**, 2250018, doi: 10.1142/s0129183122500188.
- [12] P. Wu, Z. Wang, H. Jing, P. Zhao, Optimal time-jerk trajectory planning for delta parallel robot based on improved butterfly optimization algorithm, *Applied Sciences*, 2022, **12**, 8145, doi: 10.3390/app12168145.
- [13] V. Rouss, W. Charon, G. Cirrincione, Neural model of the dynamic behaviour of a non-linear mechanical system, *Mechanical Systems and Signal Processing*, 2009, **23**, 1145-1159, doi: 10.1016/j.ymsp.2008.09.004.
- [14] X. Song, H. Fang, X. Jiao, and Y. Wang, Autonomous mobile robot navigation using machine learning, International Conference on Information and Automation for Sustainability, 2012.
- [15] X. Xiao, B. Liu, G. Warnell, P. Stone, Motion planning and control for mobile robot navigation using machine learning: a survey, *Autonomous Robots*, 2022, **46**, 569-597, doi: 10.1007/s10514-022-10039-8.
- [16] R. Chand, R. P. Chand, S. A. Kumar, Switch controllers of an n -link revolute manipulator with a prismatic end-effector for landmark navigation, *PeerJ Computer Science*, 2022, **8**, e885, doi: 10.7717/peerj-cs.885.
- [17] S. A. Kumar, K. Chand, L. I. Paea, I. Thakur, M. Vatikani, Herding predators using swarm intelligence, 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2021, 1-6, doi: 10.1109/CSDE53843.2021.9718476.
- [18] S. A. Kumar, J. Vanualailai, A Lagrangian UAV swarm formation suitable for monitoring exclusive economic zone and for search and rescue, 2017 IEEE Conference on Control Technology and Applications (CCTA), 2017.
- [19] X.-S. Yang, A new metaheuristic bat-inspired algorithm, Nature Inspired Cooperative Strategies for Optimization, Berlin, 2010.
- [20] K. Chaudhary, A. Prasad, V. Chand, B. Sharma, ACO-Kinematic: a hybrid first off the starting block, *PeerJ Computer Science*, 2022, **8**, e905, doi: 10.7717/peerj-cs.905.
- [21] J. A. Abdulsahab, D. J. Kadhim, Classical and heuristic approaches for mobile robot path planning: a survey, *Robotics*, 2023, **12**, 93, doi: 10.3390/robotics12040093.
- [22] M. N. A. Wahab, S. Nefti-Meziani, A. Atyabi, A comparative review on mobile robot path planning: classical or meta-heuristic methods, *Annual Reviews in Control*, 2020, **50**, 233-252, doi: 10.1016/j.arcontrol.2020.10.001.
- [23] C. S. Tan, R. Mohd-Mokhtar, M. R. Arshad, A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms, *IEEE Access*, 2021, **9**, 119310-119342, doi: 10.1109/ACCESS.2021.3108177.
- [24] X. Wang, Z. Liu, and J. Liu, Mobile robot path planning based on an improved A* algorithm, International conference on computer graphics, artificial intelligence, and data processing, 2023.
- [25] O. Wahhab, A. Al-Araji, Path planning and control strategy design for mobile robot based on hybrid swarm optimization algorithm, *International Journal of Intelligent Engineering and Systems*, 2021, **14**, 565-579, doi: 10.22266/ijies2021.0630.48.
- [26] R. Müller, M. Esser, M. Janssen, Integrative path planning and motion control for handling large components. Lecture Notes in Computer Science, 2011.
- [27] H. Tehrani Nick Nejad, Q. H. Do, R. Sakai, L. Han, S. Mita, Real time localization, path planning and motion control for autonomous parking in cluttered environment with narrow passages, 2012 15th International IEEE Conference on Intelligent Transportation Systems, 2012.
- [28] K. Chaudhary, A. Prasad, V. Chand, B. Sharma, ACO-Kinematic: a hybrid first off the starting block, *PeerJ Computer Science*, 2022, **8**, e905, doi: 10.7717/peerj-cs.905.
- [29] B. Sharma, J. Vanualailai, S. Singh, Lyapunov-based nonlinear controllers for obstacle avoidance with a planar-link doubly nonholonomic manipulator, *Robotics and Autonomous Systems*, 2012, **60**, 1484-1497, doi: 10.1016/j.robot.2012.07.014.
- [30] B. Sharma, J. Vanualailai, S. Singh, Motion planning and posture control of multiple n -link doubly nonholonomic manipulators, *Robotica*, 2017, **35**, 1-25, doi: 10.1017/s0263574714002604.
- [31] P. Panahandeh, K. Alipour, B. Tarvirdizadeh, A. Hadi, A kinematic Lyapunov-based controller to posture stabilization of wheeled mobile robots, *Mechanical Systems and Signal Processing*, 2019, **134**, 106319, doi: 10.1016/j.ymsp.2019.106319.
- [32] A. Razzaghian, A fuzzy neural network-based fractional-order Lyapunov-based robust control strategy for exoskeleton

- robots: application in upper-limb rehabilitation, *Mathematics and Computers in Simulation*, 2022, **193**, 567-583, doi: 10.1016/j.matcom.2021.10.022.
- [33] M. S. Amiri, R. Ramli, Utilisation of initialised observation scheme for multi-joint robotic arm in Lyapunov-based adaptive control strategy, *Mathematics*, 2022, **10**, 3126, doi: 10.3390/math10173126.
- [34] A. Keymasi Khalaji, R. Zahedifar, Lyapunov-based formation control of underwater robots, *Robotica*, 2020, **38**, 1105-1122, doi: 10.1017/s0263574719001279.
- [35] X. Dai, S. Long, Z. Zhang, D. Gong, Mobile robot path planning based on ant colony algorithm with A* heuristic method, *Frontiers in Neurorobotics*, 2019, **13**, 15, doi: 10.3389/fnbot.2019.00015.
- [36] W. Gao, Q. Tang, B. Ye, Y. Yang, J. Yao, An enhanced heuristic ant colony optimization for mobile robot path planning, *Soft Computing*, 2020, **24**, 6139-6150, doi: 10.1007/s00500-020-04749-3.
- [37] R. Kumar, L. Singh, R. Tiwari, Comparison of two meta-heuristic algorithms for path planning in robotics, 2020 International Conference on Contemporary Computing and Applications (IC3A), 2020.
- [38] K. Qian, Y. Liu, L. Tian, J. Bao, Robot path planning optimization method based on heuristic multi-directional rapidly-exploring tree, *Computers & Electrical Engineering*, 2020, **85**, 106688, doi: 10.1016/j.compeleceng.2020.106688.
- [39] H. Kundra, W. Khan, M. Malik, K. P. Rane, R. Neware, V. Jain, Quantum-inspired firefly algorithm integrated with cuckoo search for optimal path planning, *International Journal of Modern Physics C*, 2022, **33**, 2250018, doi: 10.1142/s0129183122500188.
- [40] V. Chand, K. C. Chaudhary, B. N. Sharma, and A. Prasad, Landmark aided robot path planning using stepping-ahead firefly algorithm, Pacific Asia Conference in Information Systems, 2022.
- [41] C. Ju, Q. Luo, X. Yan, Path planning using an improved A-star algorithm, 2020 11th International Conference on Prognostics and System Health Management, 2020.
- [42] X. Yuan, X. Yuan, X. Wang, Path planning for mobile robot based on improved bat algorithm, *Sensors*, 2021, **21**, 4389, doi: 10.3390/s21134389.
- [43] H. Wang, Z. Zhou, A heuristic elastic particle swarm optimization algorithm for robot path planning, *Information*, 2019, **10**, 99, doi: 10.3390/info10030099.
- [44] N. Lin, J. Tang, X. Li, L. Zhao, A novel improved bat algorithm in UAV path planning, *Computers, Materials & Continua*, 2019, **61**, 323-344, doi: 10.32604/cmc.2019.05674.
- [45] L. Liu, S. Luo, F. Guo, S. Tan, Multi-point shortest path planning based on an Improved Discrete Bat Algorithm, *Applied Soft Computing*, 2020, **95**, 106498, doi: 10.1016/j.asoc.2020.106498.
- [46] Z. Wang, Z. Wu, L. Si, K. Tong, C. Tan, A novel path planning method of mobile robots based on an improved bat algorithm, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2021, **235**, 3071-3086, doi: 10.1177/0954406220963148.
- [47] G. Xin, L. Shi, G. Long, W. Pan, Y. Li, J. Xu, Mobile robot path planning with reformative bat algorithm, *PLoS One*, 2022, **17**, e0276577, doi: 10.1371/journal.pone.0276577.
- [48] X. Zhou, F. Gao, X. Fang, Z. Lan, Improved bat algorithm for UAV path planning in three-dimensional space, *IEEE Access*, 2021, **9**, 20100-20116, doi: 10.1109/ACCESS.2021.3054179.
- [49] B. Gunji, D. B.B.v.l., S. M.B.l., U. R. Mogili, Optimal path planning of mobile robot using the hybrid cuckoo-bat algorithm in assorted environment, *International Journal of Intelligent Unmanned Systems*, 2019, **7**, 35-52, doi: 10.1108/ijius-07-2018-0021.
- [50] Q. Wu, Z. Chen, L. Wang, H. Lin, Z. Jiang, S. Li, D. Chen, Real-time dynamic path planning of mobile robots: a novel hybrid heuristic optimization algorithm, *Sensors*, 2019, **20**, 188, doi: 10.3390/s20010188.
- [51] S. K. Pattnaik, D. Mishra, S. Panda, A comparative study of meta-heuristics for local path planning of a mobile robot, *Engineering Optimization*, 2022, **54**, 134-152, doi: 10.1080/0305215x.2020.1858074.
- [52] H. S. Yahia, A. S. Mohammed, Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: a systematic review, *Environmental Monitoring and Assessment*, 2022, **195**, 30, doi: 10.1007/s10661-022-10590-y.
- [53] H. Sang, Y. You, X. Sun, Y. Zhou, F. Liu, The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations, *Ocean Engineering*, 2021, **223**, 108709, doi: 10.1016/j.oceaneng.2021.108709.
- [54] Y. Chen, G. Bai, Y. Zhan, X. Hu, J. Liu, Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning, *IEEE Access*, 2021, **9**, 40728-40742, doi: 10.1109/ACCESS.2021.3062375.
- [55] Q. Zou, X. Du, Y. Liu, H. Chen, Y. Wang, J. Yu, Dynamic path planning and motion control of microrobotic swarms for mobile target tracking, *IEEE Transactions on Automation Science and Engineering*, 2023, **20**, 2454-2468, doi: 10.1109/TASE.2022.3207289.
- [56] W. Zhang, X. Gong, G. Han, Y. Zhao, An improved ant colony algorithm for path planning in one scenic area with many spots, *IEEE Access*, 2017, **5**, 13260-13269, doi: 10.1109/ACCESS.2017.2723892.
- [57] A. K. Kashyap, D. R. Parhi, M. K. Muni, K. K. Pandey, A hybrid technique for path planning of humanoid robot NAO in static and dynamic terrains, *Applied Soft Computing*, 2020, **96**, 106581, doi: 10.1016/j.asoc.2020.106581.
- [58] F. H. Ajeil, I. K. Ibraheem, A. J. Humaidi, Z. H. Khan, A novel path planning algorithm for mobile robot in dynamic environments using modified bat swarm optimization, *The Journal of Engineering*, 2021, **2021**, 37-48, doi: 10.1049/tje2.12009.
- [59] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, A. J. Humaidi, Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm, *Applied Soft Computing*, 2020, **89**, 106076, doi: 10.1016/j.asoc.2020.106076.
- [60] F. Gul, I. Mir, D. Alarabiat, H. M. Alabool, L. Abualigah, S.

Mir, Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning, *Journal of Parallel and Distributed Computing*, 2022, **169**, 171-184, doi: 10.1016/j.jpdc.2022.06.014.

[61] X. S. Yang, X. He, Bat algorithm: literature review and applications, *International Journal of Bio-Inspired Computation*, 2013, **5**, 141, doi: 10.1504/ijbic.2013.055093.

[62] R. Moratelli, The secret lives of bats: my adventures with the world's most misunderstood mammals, *Journal of Mammalogy*, 2016, **97**, 1783-1785, doi: 10.1093/jmammal/gyw143.

[63] X. Yang, A new metaheuristic bat-inspired algorithm, Nature Inspired Cooperative Strategies for Optimization, Berlin, 2010.

[64] B. Wang, S. Li, J. Guo, Q. Chen, Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm, *Neurocomputing*, 2018, **282**, 42-51, doi: 10.1016/j.neucom.2017.12.015.

Publisher's Note: Engineered Science Publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.