Robotica (2023), **41**, pp. 346–369
doi:10.1017/S0263574722001357

RESEARCH ARTICLE

Navigation for multi-humanoid using MFO-aided reinforcement learning approach

Abhishek Kumar Kashyap[1,2,*], Dayal R. Parhi[1] and Vikas Kumar[1]

[1]Robotics Laboratory, Mechanical Engineering Department, National Institute of Technology, Rourkela, Odisha 769008, India and [2]Robotics Laboratory, Mechanical Engineering Department, MIT Art Design & Technology University, Pune, Maharashtra 412201, India

*Corresponding author. E-mail: Akkashyapmech@gmail.com

Received: 22 September 2021; **Revised:** 17 June 2022; **Accepted:** 5 September 2022;
First published online: 30 September 2022

Keywords: moth–flame optimization, reinforcement learning approach, humanoid robot, path planning, Petri-Net controller

Abstract

The given article emphasizes the development and modeling of a hybrid navigational controller to optimize the path length and time taken. The proposed navigational controller is developed by hybridizing the metaheuristic moth–flame optimization (MFO) approach and the reinforcement learning (RL) approach. Input parameters like obstacle and target locations are fed to the MFO controller that implements a proper navigational direction selection. It forwards to the RL controller, which exercises further refinement of the output turning angle around obstacles. The collaboration of the global MFO approach with the local-based RL approach helps to optimize the path traversed by the humanoid robot in an unknown environment. The major breakthrough in this article is the utilization of humanoid robots for navigation purposes between various checkpoints. The humanoid robots are placed in a cluttered environment and assigned specific target positions to complete the assigned tasks. In the case of a multi-humanoid robot system, to avoid self-collision, it requires a Petri-Net controller to be configured in the navigation system to prevent deadlock situations and enhance the smooth completion of tasks without inter-collision among the humanoid robots. Simulations and real-time experiments are undertaken using different controllers involving single- and multi-humanoid robot systems. The robustness of the proposed controller is also validated in dynamic environment. Comparisons are carried with an established navigational controller in a similar environmental setup, which proves the proposed hybrid controller to be robust and efficient.

1. Introduction

Humanoid robots are a prominent area of interest among current researchers due to their well-perceived ability to emulate human behavior and replicate task deportment. Various sensors and interaction devices implanted on humanoid robots help smooth proceedings between the robot and its environment, hence improvising task completion's easiness. Due to humanoid robots' consistent behavior in performing repetitive jobs, a significant chunk of production and manufacturing lines employ robots to complete monotonous work. The application of humanoid robots in various segments of the industry considerably improves the work efficiency of the process and reduces accident probability to a minimum level. The implementation of humanoid robots in perilous conditions extends the domain of its application and supersedes man's applicability in these conditions. In lieu of the given advantages of the humanoid robot, it is employed in various sectors like automobile assembly, underwater repairing, offshore drilling, and basic household services. Since the humanoid robot is being widely used in various sectors, energy minimization during tasks' performance is a major concern from an economic point of view. Various researchers have employed the use of artificial intelligence (AI) approach for the path-planning of mobile robots and humanoids, which are discussed below.

Authors [1] have provided a novel forward model-based technique for lidar analytical approach. Existing mapping methods have a number of flaws that can be traced back to an absence of a clear forward concept. The goal of Wu *et al.* [2] has been to create a generalized wavefront method for mobile robot route mapping. Multi-objective point combinations, layered grid expenses, logarithmic expansion across barriers, and route improvement are all part of the research. Liang and Xu [3] have emphasized the development of a mutated simulated annealing (SA) approach to applying on wheeled robot global navigation. They modified the conventional method since it has a demerit of slow convergence rate. Zhu *et al.* [4] have discussed the memetic approach (MA) of path planning and examined it on simulated maps and correlated with other counterpart approaches. They also described that the result came from the MA has better efficiency than the other conventional approaches. Botzheim *et al.* [5] have emphasized a bacterial MA to recognize a collision-free route for the wheeled robot by minimizing the path length and the number of divergences in the trodden path. In this paper, Ant Colony Optimization (ACO) has been used to determine the presence of intra-class pheromone secreted by other ants to determine the shortest path. Liang *et al.* [6] have emphasized the use of bacterial behavior to find out an optimal path without collision with the hindrances, with the help of bacterial foraging approach, in order to make a bio-inspired route outlining an approach for a wheeled robot.

The authors have created a navigation framework that uses an evolutionary threshold filtering technique to identify barriers in a sliding window, categories the identified barriers with a tree structure, heuristically forecasts future collisions, and uses a simplified Morphin method to select the best path [7]. Gao and Tian [8] have evaluated the refined SA mixed optimization approach and the artificial neural network and implemented it to wheeled robot motion planning in an unknown environment. This approach improves the protective measures and enhances the convergence value of the SA approach. It also minimizes the computing time of route outlining; thereby, it becomes easy and fast to obtain the optimal global solution. Jun and Qingbao [9] have presented a multi-objective mobile robot route planning approach based on an enhanced genetic algorithm. The method introduces a chaotic series and a heuristic technique focuses on ecological expertise to initialize the population in order to enhance the individuals' ergodicity and practicability in the search area. Yue and Wang [10] have focused on the use of applied SA with a compound shape method for the route outlining for the neural network of wheeled robots. Along with achieving the globally optimal result that identifies the SA approach, it has been used for an ideal declining inclination, thus refining the convergence value. Sharifi and Vinke [11] have focused on the use of a SA approach to preventing the entrapment of robots in a local minima position. Authors have applied it for local path planning in a stationary environment to increase the effectiveness of selecting path and to ensure robustness. Alfaro and Garcia [12] have proposed a fuzzy logic and SA approach to design an automated route outlining approach of wheeled robot. Proposed approach was used to regulate the velocity of the wheeled robot at the time of navigation, and it was applied in-between of the fixed polygonal hindrances by applying 49 fuzzy rules to get collision-free optimal path.

A behavior-based neural network and reactive command framework for mobile robot guidance are proposed by Pandey and Parhi [13]. As sources, two distinct reactive behaviors have been used that includes location and angle of the obstacles. Ganganath and Cheng [14] have presented an off-line path planner for which the basic was ACO approach (ACO2 Gauss) for wheeled robots. Chang *et al.* [15] has presented an advanced dynamic window approach based on Q-learning. To improve the efficiency of global guidance, the fundamental assessment algorithms are updated and enhanced by introducing two additional evaluation processes. Wei and Zhao [16] have developed a three-part technique that includes motion primitives, a Bayesian network, and an unique coupling neural network. To decouple human arm motions, motion elements are utilized. Arm motion categorization enhances the precision of human-like gestures.

Kusuma *et al.* [17] have emphasized the usage of the A* search approach for path-planning of a humanoid robot in various home assistance operations. The methodology aims for an efficient path sketching and rerouting of the humanoid robot's path in case it of unreachability to a given checkpoint. Sabe *et al.* [18] have proposed a stereo vision for path planning and obstacle detection of a QRIO humanoid robot in home environments. The grid detection methodology is implemented for obstacle

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

avoidance along the proposed path. Huang *et al.* [19] have proposed a star search navigational strategy for efficient footstep planning of humanoid robots with a major emphasis on energy consumption in the navigation process. The given methodology is based on the optimization of step length, step width, and the turning angle of the humanoid robot. Lee *et al.* [20] have proposed the concept and modeled a service-providing humanoid robot for performing day-to-day tasks in an unknown environment with an in-built 3D object detection system. Simulations in real-time environments show that the proposed robot performs the given tasks at about 24% efficiency of human beings. Lagaza *et al.* [21] have emphasized the use of the Spider Monkey algorithm for the optimization of the path traced between checkpoints in static and dynamic environments.

As evident from the study of the presented research papers, a significant amount of work has been done for path planning in mobile robots. Research work involving the navigation of humanoid robots in unknown environments is scarce. Path planning approaches involving humanoid robots have contributed a meager amount in research works with almost no proper work done for navigation of robots between multiple targets. In the current scenario, the robots employed in various industries are multidimensional, requiring various skills to complete the task. This includes the design of a mechanical engine, repair works, and various other breakdown maintenance tasks. Due to major complications in the humanoid robots' path planning, a hybrid navigational strategy combining the global moth–flame optimization approach with the accurate and local reinforcement learning (MFO-RL) approach has been proposed in this article. The MFO is a global path planning approach that cancels out the probability of local minima entrapment, whereas RL being a local path planning approach that optimizes the angle of turning of the humanoid robot around various obstacles. The paper also focuses on a multi-humanoid system's path planning in an unknown environment, which has not been done actively yet. In the case of multi-humanoid navigation, a situation of deadlock arises when a robot enters another robot's proximity zone. To avoid inter-collision among the robots, a Petri-Net controller is configured alongside the hybrid controller for the path-planning of multi-humanoid system. The Petri-Net controller, in this scenario, negotiates between the humanoid robots and cancels out any hindrance during the navigation. The various research works done on the MFO approach, RL approach, and Petri-Net controller are enlisted below.

Jalali *et al.* [22] have proposed MFO-based multilayer perceptron network for trajectory planning of an autonomous robot in an unknown environment. The proposed MFO methodology, entrusted with controlling the NN controller's weight and bias parameters compared with various evolutionary and gradient-based approaches, demonstrates its clear superiority. Abdullah *et al.* [23] have developed an MFO-based approach for the optimization of energy usage at assembly stations. The idle energy in the assembly sequencing problems is optimized using MFO-based controller, which performs much better than the genetic algorithm (GA), Particle Swarm Optimization (PSO), and ACO controllers in terms of robustness, feasibility, and computational time. Mehne and Mirjalili [24] have proposed the use of the MFO approach for feedback control on a general nonlinear problem. The methodology is used to convert a given infinite-dimensional problem to a finite one using a set of coefficients. The application of the proposed controller to certain established problems demonstrates its efficiency in achieving optimum results in this regard. Elaziz *et al.* [25] have explored the applications of MFO-based approaches in various benchmark works in conjunction with opposition based learning (OBL) and Differential evolution (DE) approaches for better initial insect population generation and exploration facilities. The results of the experiment are evidenced by the fact that the proposed approach performs better than the established ones.

Gao *et al.* [26] have illustrated the use of the Q-learning reinforcement approach for the path-planning of mobile robots in cluttered environments. The proposed methodology involves the generation of an obstacle-free path in the first step and optimization of the generated path in the next step. The emphasis was to establish the proposed method, superior to the BFS breadth-first search (BFS) and rapidly exploring random trees (RRT) methods. Fakoor *et al.* [27] have solved the latter challenge by using a unique method wherein the speed is decided using fuzzy Markov decision process (MDP).

Trinh *et al.* [28] have emphasized the usage of dipole interaction systems for the path-planning of humanoid robots in cluttered environments. Furthermore, a Petri-Net framework with a dynamic window approach has been used for dynamic obstacle avoidance in unknown environment navigation. Parhi and Mohanta [29] have emphasized the use of fuzzy logic controller combined with the Petri-Net framework, which helps in avoidance of dynamic obstacles during the navigation of humanoid robots in the given terrain. Kumar *et al.* [30] have proposed the usage of a hybrid RA-fuzzy logic controller for hassle-free travel of humanoid robots in unknown environments. The Petri-Net controller attached to the proposed framework helps in the prevention of inter-collision among the humanoid robots during task completion.

Implementation of the sole AI approach for the path-planning of a robot takes an enormous amount of time and renders the strategy uneconomical. Furthermore, the single technique used often leads to the robot being captive in a local minima situation where the conditions created are unresolvable. The hybrid navigational strategy proposed in the article deals with the entrapment of the robot in a local minima condition and guides in faster convergence toward the target with minimum deviations. The hybrid controller is implemented in single- and multi-humanoid robots in a common platform, which is still a topic for research. It is assessed against various standalone approaches based on path length. The hybrid controller leads to the successful completion of the task and has not been used anywhere yet by author's knowledge. The MFO approach is a global search strategy that is based on the generation of fitness function of various sample spaces in the unknown environment to sketch out an optimum path for the humanoid robot navigation. The RL approach is a more accurate local search strategy that helps in the determination of the optimum turning angle of the humanoid for collision-free navigation. RL optimizes the footstep planning to efficiently avoid obstacles with any shapes. The algorithm also helps the robot to find out the prioritized target to decrease the overall length of the path. The hybridization of these two approaches are implemented in a single controller and configured alongside a Petri-Net controller for dynamic obstacle clearance. The proposed controller is also capable of solving the problem of avoidance of dynamic obstacles (multi-humanoid robots). The layout of the article is demonstrated as follows. Section 2 emphasizes the MFO approach, whereas the RL approach is described in Section 3. The Petri-Net controller, which is used to solve the conflict problem, is presented in Section 4. The hybrid controller based on the MFO and RL approach is presented in Section 5. Various simulations and experimental results using the proposed hybrid controller are shown in Section 6, whose comparisons with previously established approaches are discussed in Section 7. Section 8 demonstrates the conclusion of the presented work and gives the scope for future improvisations in the given context.

2. Moth–flame optimization

It is a bio-inspired swarm optimization method based on moth's navigational strategy during the night. This paper's navigational mechanism is referred to as the transverse orientation in which moths can travel in a straight line for longer distances by maintaining a fixed angle with respect to the moon. However, the presence of artificial light sources in the environment, moths, due to their proximity to these artificial sources, is influenced on a larger scale and maintains a certain angle with them, resulting in the travel path being spiral. In this approach, the moths traverse the spiral path while exploiting the search space to reach the maximum fitness function value attainable by the flame position with respect to other moths.

In this optimization approach, the moths are randomly placed in the solution space, where each moth is assigned a solution in the sample space. Moths are assigned a specific fitness function and a flame that stores the best solution found by the moth. With every iteration, the moth traverses a spiral path around the flame, thus updating its fitness function and new positions. The position matrix for'x' number of moths in'n 'dimensional solution space are as follows:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,n} \\ m_{2,1} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ m_{n,1} & \dots & \dots & m_{n,n} \end{bmatrix} \tag{1}$$

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

The fitness function for'x 'number of moths are as follows:

$$MF = \begin{bmatrix} MF_1 \\ MF_2 \\ \cdots \\ MF_x \end{bmatrix} \tag{2}$$

The position matrix for 'x' number of flames in 'n' dimensional solution space are as follows:

$$F = \begin{bmatrix} F_{1,1} & \cdots & F_{1,n} \\ \cdots & \cdots & \cdots \\ F_{x,1} & \cdots & F_{x,n} \end{bmatrix} \tag{3}$$

The complete fitness function of flames based on fitness function of each flame is given as as follows:

$$FF = \begin{bmatrix} FF_1 \\ FF_2 \\ \cdots \\ FF_3 \end{bmatrix} \tag{4}$$

where F_j is the j^{th} flame. The fitness function refers to the evaluation of the positions based on the suitability of the given path for travel, that is, the higher the fitness function value, the higher are the chances of treading via that path. The functions used for position updation of moths that contains M denotes random position of moths, and E presents the termination search operation.

The random positioning of moths is done by the M function which is given as:

$$M(i,j) = (U(i) - L(j)) * rand() + L(i) \tag{5}$$

where U and L are arrays defining upper and lower bounds of variables, respectively.

The movement of moths in the search space is based on traverse orientation, and the path model used is the logarithmic spiral curve. The curve has the following properties:

(i) Spiral's initial point is the starting point of the moth and the final point is the next position of the flame.

(ii) Fluctuations in the spiral curve's range should be within the search space domain.

The movement of moths in the search space is given by the H function whose equation is given as:

$$H\left(M_i, F_j\right) = D_i e^{pt} \cos(2\Pi t) + F_j \tag{6}$$

where H represents the movement of moths in search space, M_i is the i^{th} moth, F_j is the j^{th} flame, p is the defining parameter for the spiral, t is the time between [–1,+1], and D_i is the distance between i^{th} moth and j^{th} flame and is given by:

$$D_i = \left|F_j - M_i\right| \tag{7}$$

Thus, with the completion of each iteration, the moths and flames' fitness value get updated, which ensures that the exact location of the best solution is found. The terminal function E is based on the termination criteria as proposed by the programmer which is limited by the number of epochs or a certain minimum fitness function value. The flow diagram of the working of an MFO approach is presented in Fig. 1.

Steps for carrying out the MFO approach:

Step 0: Set the number of moth and flame and the maximum no. of iterations 'δ' to be carried out. Set the upper and lower bound of the function and initialize random generation of the moths.

Step 1: Calculate the fitness function for each moth and flame and tag the best position by comparing the fitness function of flames.

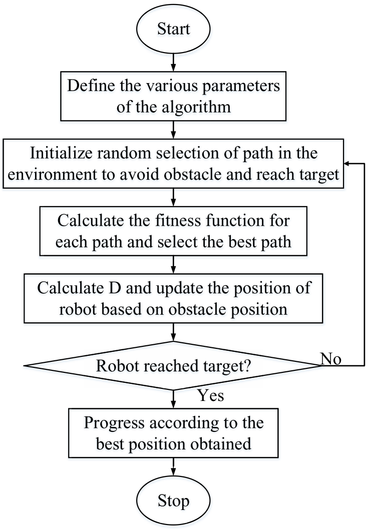
Start
Define the various parameters of the algorithm
Initialize random selection of path in the environment to avoid obstacle and reach target
Calculate the fitness function for each path and select the best path
Calculate D and update the position of robot based on obstacle position
Robot reached target?
No
Yes
Progress according to the best position obtained
Stop

Figure 1. *Flow chart of MFO approach in humanoid robot navigation.*

Step 2: Update the flame position by the fitness function criteria.

Step 3: Calculate the value of D_i for the corresponding moth and update it's $M(i,j)$ using the relation:

$$M_{i+1} = H\left(M_i, F_j\right) \tag{8}$$

Step 4: If no. of iterations$\geq \delta$, terminate the process or else, reiterate the process.

2.1. Description of the optimal navigational fitness function

Assume a humanoid NAO going through an environment having a starting location (X_s, Y_s) and a goal location (X_t, Y_t), and a barrier at (X_o, Y_o). The environment is made up of a number of robots that behave as dynamic barriers to one another. The main purpose is to develop the humanoid NAO smart enough to evade static and dynamic barriers (another humanoid NAO) and reach the goal in the least period of time. In addition, the humanoid NAO should create the shortest and best route possible while also ensuring its smoothness. In this part, the objective navigational mechanism is built while keeping these factors into account. As there is a possibility of inter-collision, the motion of another robot also disrupts their activities. As a result, in order to comprehend the situation and evade self-collision, the robot's movement is also evaluated.

The objective feature is based on the minimum length to the goal, the evasion of barriers, and the smooth path. These aims should be met in order to complete the assignments with the minimal computational effort.

2.1.1. Minimum route length

The design of the minimum route length is considered as the primary goal of humanoid NAO's trajectory mapping. Between the starting position and the goal, the robot must take the quickest path. To find the optimal path, it ought to be a Euclidean length. At each cycle, the robot's location ought to be accurately adjusted to obtain the shortest trip interval between it and the destination. The module is based on the humanoid NAO's optimal positioning and goal. It is defined as [31]:

$$f_1(X, Y) = d[(X_{Hr}(i)\,, \mathrm{Y}_{Hr}(i))\,, (X_t, \mathrm{Y}_t)] \tag{9}$$

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

where $(X_{Hr}(i)\,, \mathrm{Y}_{Hr}(i))$ is the humanoid NAO's location at i^{th} point. There could be n locations between starting and goal, with the coordinate of humanoid NAO at n^{th} location is $(X_{Hr}(n)\,, \mathrm{Y}_{Hr}(n))$. The overall path distance is equivalent to the total of the lengths between the humanoid NAO's various sites. It is defined as [31]:

$$E_{spl} = \sum_{i=1}^{n} d[(X_{Hr}(i)\,, \mathrm{Y}_{Hr}(i))\,, (X_t, \mathrm{Y}_t)] = \sum_{i=1}^{n} spl \tag{10}$$

$$spl = \sqrt{(Y_{Hr}(i+1) - Y_{Hr}(i))^2 + (X_{Hr}(i+1) - X_{Hr}(i))^2} + \sqrt{(Y_{Hr}(n) - Y_t)^2 + (X_{Hr}(n) - X_t)^2} \tag{11}$$

2.1.2. Barrier evasion

For safe trajectory mapping, barrier evasion should be considered in addition to the minimal path length. The utility is determined by the location of j^{th} barrier and the humanoid NAO at i^{th} location. The following is a representation of the framework [31]:

$$f_2(X, Y) = \left[(X_o(j)\,, \mathrm{Y}_o(j))\,, (X_{Hr}(i)\,, \mathrm{Y}_{Hr}(i))\right] \tag{12}$$

The range between the humanoid NAO at location i^{th} and the barrier at location j^{th} must be kept to a minimum (safe). The humanoid NAO must travel to the closest, secured location to the barrier. The entire length of the route between them is denoted by [31]:

$$E_{oa} = \sum_{j=1}^{m}\sum_{i=1}^{n} d\left[(X_o(j)\,, \mathrm{Y}_o(j))\,, (X_{Hr}(i)\,, \mathrm{Y}_{Hr}(i))\right] = \sum_{j=1}^{m}\sum_{i=1}^{n} oa \tag{13}$$

The above equation represents the discrete situation while achieving the target. It is the summation of the distance between the robot at and i^{th} position and barrier at j^{th} position. Likewise, it draws a line between the start point and the target by considering the barrier.

where, m is the count of barriers.

$$oa = \sqrt{(Y_{Hr}(i) - Y_o(j))^2 + (X_{Hr}(i) - X_o(j))^2} \tag{14}$$

2.1.3. Soothing of route

The purpose of performance management is to maintain a smooth path while evading barriers. It denotes the reduction of angle fluctuation from the Euclidean route to the smallest possible value (from the humanoid NAO at i^{th} location to the goal). It is formulated as [31]:

$$f_3(X, Y) = |\alpha[H(i)\,, H(i+1)]\,, \alpha[H(i)\,, \mathrm{T}]| \tag{15}$$

where $\alpha[H(i)\,, H(i+1)]$ and $\alpha[H(i)\,, \mathrm{T}]$ are the angles between the humanoid NAO's route at i^{th} and $(i+1)^{th}$ spots and the angle between the humanoid NAO's route at i^{th} location and the goal, respectively.

$$\alpha[H(i)\,, H(i+1)] = \tan^{-1}\left[\frac{Y_{hr}(i+1) - Y_{hr}(i)}{X_{hr}(i+1) - X_{hr}(i)}\right], \text{ and } \alpha[H(i)\,, \mathrm{T}] = \tan^{-1}\left[\frac{Y_t - Y_{hr}(i)}{X_t - X_{hr}(i)}\right]$$

$$E_{ts} = \sum_{i=1}^{n} |\alpha[H(i)\,, H(i+1)]\,, \alpha[H(i)\,, \mathrm{T}]| \tag{16}$$

The multi-objective activity is calculated as follows [31]:

$$f(mof) = w_1 \sum_{i=1}^{n} E_{spl}(i) + w_2 \sum_{i=1}^{n}\sum_{j=1}^{m} E_{oa}(i)\,(j) + w_3 \sum_{i=1}^{n} E_{ts}(i) \tag{17}$$

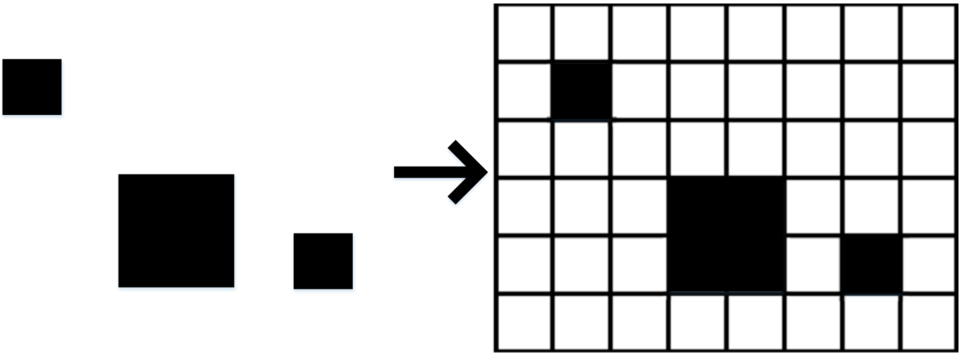

Figure 2. Representation of global space example with cell system.

where the weight functions w_1, w_2, and w_3 represent the relative importance of specific objectives. They must adhere to the following restrictions:

$$w_1 + w_2 + w_3 = 1 \tag{18}$$

The multi-objective equations' ultimate fitness is expressed as:

$$f_{fitness} = \frac{1}{f(mof) + \varepsilon} \tag{19}$$

where ε is a small value that is inserted to ensure that it is not divided by zero. Because all factors must be reduced to produce the best solution, the multi-objective expression is a minimization task.

To achieve our objective, it is needed to create a genotype layout that could be easy for the MFO approach agents to interpret while yet preserving the necessary routing path for autonomously moving robot. As a result, as illustrated in Fig. 2, a simple cell layout has been employed for the global space. The parameters (length and width) of the global space are presumed to be available. The following hypotheses are used in this concept:

1. There is a layout of the space where navigation occurs. The exploration space's length and width will be determined by the route organizer, who will then implement a cell layout to the space. As a result, the space is split into rows and columns.

The strategy of assuming that the quantity of rows equals the number of columns is used. The cell's populated area represents the positions of recognized barriers.

2. The source and target of the planned robot's motion are also specified in predefined dimensions.
3. The robot is able to travel on all viable cells, with its center moving down an imaginary path connecting the centers of one cell and the centers of adjacent cell.

A route in that domain is defined by a genotype [32] having N genes, provided a navigable scenario defined by N rows. Every gene frequency correlates with a column reference inside that row, while each gene location correlates with a row reference. Let's say there is chromosome 2,3,3,6,8,8. This genotype denotes a route that begins in row 1, column 2 (1,2) and finishes in row 6, column 8 (6,8). This route's interim locations are (2,3), (3,3), (4,6), and (5,8). The passage through this route in world space is shown in Fig. 3(a), with point (1,1) presumed to be in the upper left corner.

The interim phases, or vertices, of a route are represented by the directional data of a chromosome. Moving the robot in a continuous trajectory from one cell's center to another cell's center, on the other hand, might result in the robot moving in a diagonal route over several neighboring cells. If any neighboring cells on the robot's path through one row to another have a barrier, it will create issues. A better solution is to divide the diagonally route section into a horizontally and a vertically portion, allowing the robot to navigate around barriers. As a result, an orientation information to the chromosomal architecture has been added to signify the robot's initial turning as it moves to another vertex.

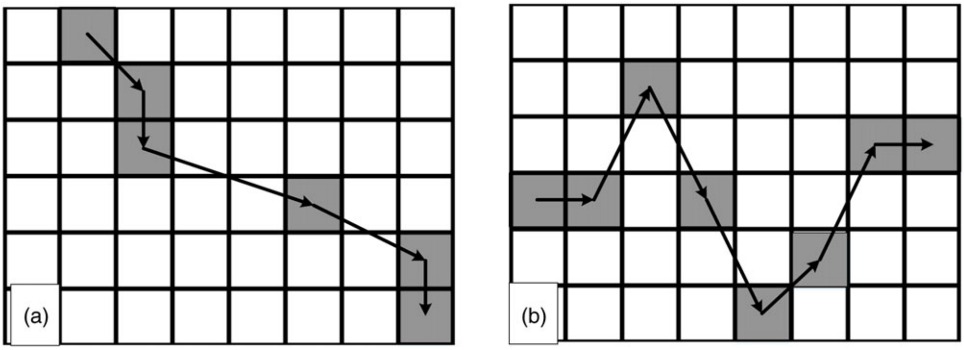

Figure 3. *Representation of global space navigation: (a) row wise and (b) column wise.*

The genotype architecture has been enhanced by introducing a guidance bit to every chromosome: 0 indicates column-wise configuration and 1 indicates row-wise configuration. Each locus, on the other hand, correlates with a column reference on a column-wise chromosome, whereas every gene correlates with a row reference. The movement for the column-wise chromosome 4,4,2,4,6,5,3,3 is shown in Fig. 3(b). This genotype shows a route that begins in row 4, column 1 (4,1) and continues to row 3, column 8 (3,8) via intermediary locations (4,2), (2,3), (4,4), (6,5), (5,6), and (6,3).

The process has one limitation that the genetic individual representation disallows moving back simultaneously in both rows and columns. The proposed optimization technique's parameters are the number of search agents (50), number of features (400), number of iterations (200), and dimension (400).

3. Reinforcement learning

RL is a learning approach implemented in unknown environments for assessment and improvisation of the agent's learning skills when an incomplete output data set is provided. The approach is based on the development of a continuous space scenario into a discrete space scenario by the implementation of sampling criteria used in the MDP and the Q-learning algorithm. This random sampling of the given environment occurs unless the final target is attained. The approach considers obtaining a specific Q-value from a specific action and moving on from the current state to a newer state. The obtained Q-value is compared to the maximum Q-value desired by carrying out value iterations. The policy network is the major deciding criteria, which determines the course of action taken by the agent in the current state. Based on evaluating the Q-value received from a certain iteration, the network either gives a reward or imposes a penalty on the agent's action, which governs its future actions.

The penalty imposed on the actions reduces the probability of a similar action occurring in the future, thus, filtering out the action from the agent's behavior pattern. Similarly, a rewarded action has a greater probability of occurrence in the future in similar states. The agent has two primary behaviors: exploration and exploitation. Exploration refers to assessing the immediate surroundings by the agent's inquisitive behavior and drawing out a certain conclusion, while exploitation is the calculation of optimum action based on provided information to generate a suitable path for locomotion in the environment. Q-learning is based on the agent's exploration behavior in the sample space and discovers optimum solutions that are not provided in the input data set. The schematic diagram for the RL approach has been presented in Fig. 4.

3.1. State representation and reward function

The state comprises of ultrasonic sensors data that include range statistics from the working region, its forward speed, and the respective ranges of x and y from the robot to the goal location:

$$z = z_{ultrasonic} + z_{t\arg et} + z_{velocity} \tag{20}$$

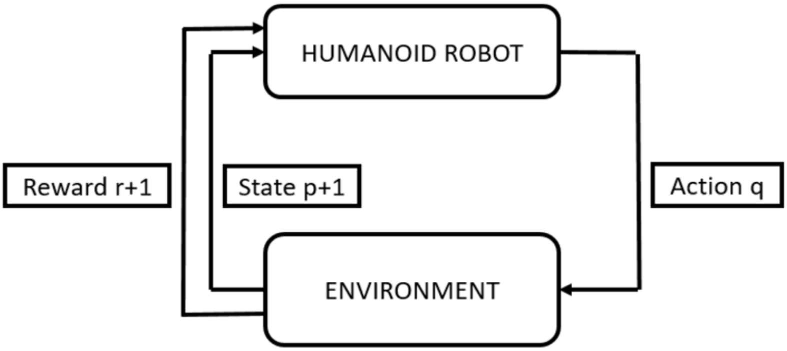

Figure 4. *Schematic diagram of reinforcement learning approach [33].*

where $z_{ultrasonic}$ data is ultrasonic sensor information that measures range and recognizes obstacles to identify the interaction between barriers and the robot. Additionally, the ultrasonic information has been utilized to explicitly estimate the object's motion position and velocity over three step. The comparative distance between the robot and the goal point is $z_{t\arg et}$, and moving orientation may be efficiently determined by's unambiguous data on how the moving route is correct. The robot's speed data is displayed via $z_{velocity}$. Furthermore, $z_{velocity}$ can be used to determine the robot's maximum velocity and inertia, as well as the robot's evasion technique based on velocity.

The robot's objective in this investigation would have been to achieve the goal point (X_t, Y_t). Traveling with collision avoidance using RL and staying within the efficiency limit is required to achieve the goal location. As a result, the reward has also been taken into account independently. The combination of following two reward functions is considered:

$$R_t = R_G + R_{CA} \tag{21}$$

The system earns a huge reward of 20 if the robot achieves the goal location. Furthermore, if the range to the goal grows shorter than previously while advancing to the goal location, the robot is rewarded as it moves in the correct route:

$$R_G = \begin{cases} 20 & if\ p_c < 0.5 \\ p_p - p_c & otherwise \end{cases} \tag{22}$$

where p_c is the current position of robot and p_p is the previous position of robot.

$$p = \sqrt{(Y_{Hr}(i) - Y_t(j))^2 + (X_{Hr}(i) - X_t(j))^2} \tag{23}$$

Walking in the reverse way from the desired location results in a penalty equal to the distance traveled in one step, while traveling in the other direction results in a reward equal to the distance traveled in one step.

Whenever the robot collides with a barrier, the reward R_{CA} applies a severe penalty of –20, which is a condition that the robot must be trained to escape:

$$R_{CA} = \begin{cases} -20 & if\ collisions\ occur \\ 0 & otherwise \end{cases} \tag{24}$$

The random sampling carried out in the Q-learning approach is economical as it significantly reduces the number of grids to be computed for the generation of an optimum path. After the sampling, paths that interfere with the obstacles are removed, and the remaining connections are considered for the next stage. The prime purpose is to generate a smaller and smoother path based on the current location according

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

to the given state of the function. The Q-learning approach is based on the existence of a Q-function $Q(p_t, q_t)$ described as:

$$Q(p_t, q_t) = \max(R_{t+1}) \tag{25}$$

where R_t is the total future reward from any time point t and is defined as:

$$R_t = r_t + wr_{t+1} + w^2 r_{t+2} + \ldots.. + w^{n-t} r_n \tag{26}$$

And p_t is the state at time step t and q_t is the action at time step t. Substituting the value of R_{t+1} in Eq. (25) yields the equation as:

$$Q(p_t, q_t) = w + \left[\delta^* \{\max Q(p_{t+1}, q_{t+1})\}\right] \tag{27}$$

where w is the discount factor ($0<w<1$)

The action value function is denoted as $Q(p_t, q_t)$ which is improvised by the following given rule:

$$Q(p_{t+1}, q_{t+1}) = Q(p_t, q_t) + \beta^* \left[a_t + \delta^* \{\max Q(p_{t+1}, q_{t+1})\} - Q(p_t, q_t)\right] \tag{28}$$

where β is the learning rate ($0<\beta<1$), which decides the learning convergence rate, and δ is the discount rate ($0<\delta<1$), which determines the relative ratios between awards at two consecutive states.

The steps for carrying out the Q-learning process in path planning are as follows:

1. The employed humanoid robot evaluates the input data provided to it and decides upon exploration or exploitation of the environment.
2. An action is determined by the humanoid robot based on the policy network by the usage of policy gradients.
3. The action is carried out in the given environment.
4. Based on the action undertaken, the agent receives an award or penalty as a reinforcement and a new state is formed.
5. If $Q(q_t, p_t)<Q_{\max}\left(q_{t,}p_t\right)$, reiterate the process or else, stop the iteration and feed the final data to the path planning approach.

With the increase in the number of iterations, if the Q-value also increases, then the path planning approach is in a learning state. As the value of Q reaches an optimum value, the iterations stop, and the stable state is reached. This data is fed to the path planning approach as the optimized path is obtained. The RL algorithm has been presented in Fig. 5.

3.2. Collision detection problem

When humanoid robot navigates in a global space to complete tasks, it utilizes its own gadgets to evaluate if the range between robot and obstacle is smaller than the threshold distance T_r, and subsequently if there is a risk of collision. If a collision is possible, the humanoid robot must execute a collision prevention strategy to maintain safe navigation. Due to the limitations of movement of humanoid robots, they have a safety zone, and whenever an obstacle enters in its safety zone, avoiding movements are required. As a result, an avoidance movement is necessary before any obstacle enter the robot's security zone.

The detection zone for the humanoid robot is a part of circular region with threshold radius T_{safe} in its present location $\left(x_{r,}y_r\right)$, as depicted in Fig. 6. In the safe region, the humanoid robot will move safely, if there is no obstacle, else it has high risk of collision.

In the Fig. 6, $x_p o y_p$ is the relative coordinate system and XOY is the absolute coordinate system, with the location of humanoid robot taken as the origin. The safe range of the robot is denoted by T_{safe} and the threshold range is represented as T_r. The humanoid robot in the present location $\left(x_{r,}y_r\right)$ to identify the range between the barriers and the humanoid robot is being R. Once the sensor radar $T_r < R$ demonstrates that the humanoid robot has identified the barriers, and a possibility collision prevention operation decision should be considered. If $R < T_{safe}$ is not present, it signifies that the barrier has reached the robots's safe area and cannot be efficiently avoided.

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

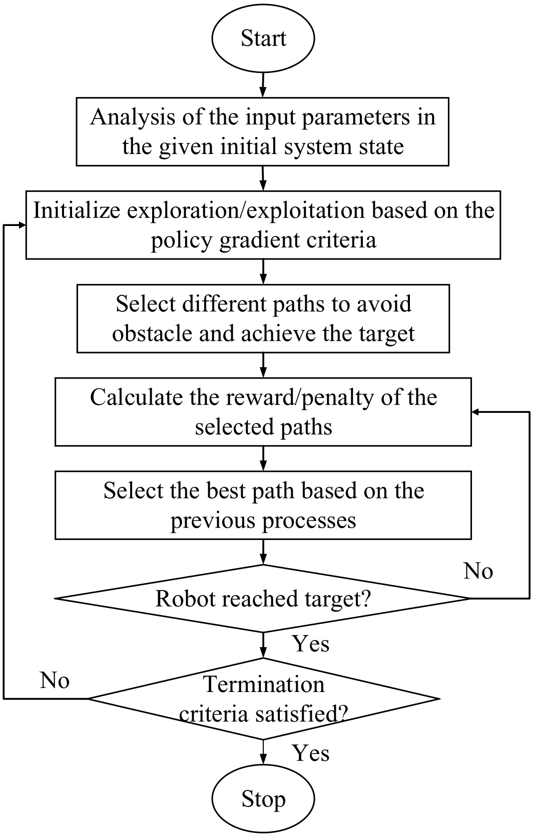
Start
Analysis of the input parameters in the given initial system state
Initialize exploration/exploitation based on the policy gradient criteria
Select different paths to avoid obstacle and achieve the target
Calculate the reward/penalty of the selected paths
Select the best path based on the previous processes
Robot reached target?
No
Yes
Termination criteria satisfied?
No
Yes
Stop

Figure 5. *Flow chart of reinforcement learning approach in humanoid robot navigation.*

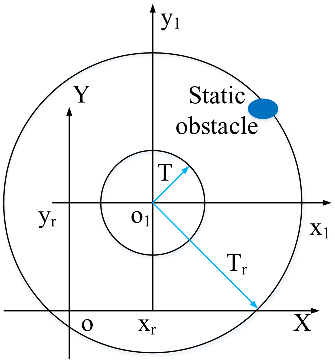

Figure 6. *Representation of safety range of the robot and detection of static obstacle.*

4. Petri-Net controller for dynamic obstacle avoidance

Petri-Net, consisting of places, transitions and arcs, was first used for modeling of systems in 1981 by Peterson [34]. A Petri-Net setup is often characterized by marks known as a token, which creates a configuration called marking. The enabling of a certain token in a Petri-Net is termed as firing, which leads to the consumption of the required input token and the creation of an output token. Unless defined, the firing order in a Petri-Net is often nondeterministic, that is, in any random order. The transfer of tokens from one state to another occurs via a transition move between them.

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

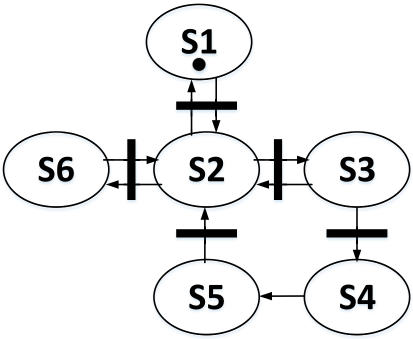

Figure 7. *Petri-Net framework for multi-humanoid robot navigation system.*

In the scenario of obstacle avoidance among humanoid robots, where they act as dynamic obstacles to each other, Petri-Net is used for the employment of supervised firing order. The property of concurrent firing of humanoid robots is utilized in this controller. The proposed Petri-Net network for obstacle avoidance is shown in Fig. 7. The configuration net consists of six places and eight possible transitions. It shows the initial position of robot using a circular black circle and transition is denoted by black bar.

The steps for working of the Petri-Net controller in a humanoid robot navigation system are as follows:

Step 1: Initially, the token is enabled at state 1, which is the state of the humanoid robot's initialization. The humanoid robot is at its initial position waiting for the signal to start the task.

Step 2: As the first transition occurs, the token is transferred to state 2, which is the state of navigation of the humanoid in the cluttered environment while avoiding static obstacles by following the path decided by the proposed path planning approach. The robot navigates in an unknown environment unless it comes in proximity of another humanoid robot.

Step 3: The proximity issue leads to the transition of the token to state 3, where the humanoid robot comes into conflict scenario which another robot, hence, creating dynamic obstacles in its pathway.

Step 4: The token then undergoes another transition from state 3 to state 4, which involves the implementation of negotiation strategy among the humanoid based on various deciding parameters and assignment of priority to the preferred robot. The priority parameters often prioritize the robot, which is at a lesser distance from its goal and has lesser obstacles left in its path to overcome.

Step 5: The token thereafter transfers to state 5, where the situation is similar to state 2, that is, navigation in the unknown environment consisting of static environment and treading of the optimal path decided by the path planning approach. The robot which has been prioritized earlier undertakes this task and moves forward, whereas the less priority robot's state is transferred to state 6.

Step 6: State 6 involves waiting for the lesser prioritized robot in its original place till the superior robot leaves its proximity zone. Once the higher priority robot clears out of its proximity zone, the state of the lesser prioritized humanoid robot transitions to state 2.

Step 7: In case a third robot enters the conflict zone in between, it is assigned as state 6.

This proposed configuration net helps in the smooth completion of each humanoid robots' task without causing any conflict between them.

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

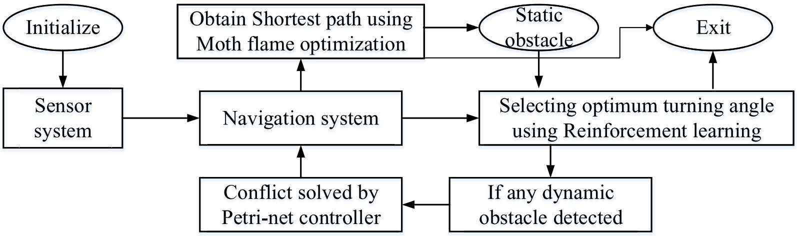
Initialize
Obtain Shortest path using Moth flame optimization
Static obstacle
Exit
Sensor system
Navigation system
Selecting optimum turning angle using Reinforcement learning
Conflict solved by Petri-net controller
If any dynamic obstacle detected

Figure 8. *Block diagram of proposed hybrid MFO-RL controller.*

5. Hybridization of MFO and RL (hybrid MFO-RL) controller

The proposed path planning approach in the hybrid controller is a combination of the bio-inspired MFO approach and the RL approach. From the literature review section, it is evident that various classical approaches have been earlier used for the path-planning of the humanoid robot, which is usually faster but less accurate while convergence to final solutions. On the other hand, AI approaches are precise but are less efficient in providing optimum solutions at a steady rate. Furthermore, usage of a single approach often results in the entrapment of the humanoid robot in the local minima, which presses the need for another approach to remove this complication. For overcoming this problem, we have utilized the culmination of a classical approach to strengthening the benefits of a metaheuristic approach for efficient path planning.

In the proposed hybrid controller, input parameters of the humanoid robot's surroundings are fed to the sensor system implanted on the humanoid robot's body. The sensor system forwards this information to the navigation system, which is controlled by the MFO controller. The MFO controller targets the obstacles and the final goal and designs a proper path for the navigation of the humanoid robot. In case of conflict with an obstacle, the RL controller is activated. This data further fed to the RL controller helps further refine the optimal angles during turning, as shown in Fig. 8. This process continues till the humanoid robot reaches the final goal. The hybrid controller works in a similar fashion as the feedback loop mechanism to obtain optimum results.

The steps of the MFO-RL hybrid controller for humanoid robot navigation are listed below:

1. The start and end positions of the humanoid robot are intialized.
2. The MFO controller is activated, which decides the obstacles' fitness function and the target and sketches out an optimal path for the robot.
3. The robot is navigated freely through the cluttered environment until it encounters an obstacle.
4. MFO controller has activated again. In case of any obstacle, input parameters (left obstacle distance, right obstacle distance, and front obstacle distance) are fed to the robot and initial turning angle is determined to avoid getting stuck in local minima, the reward significance for every particular optimum approach as well as the global best location in the trajectory planning process is computed and recorded.
5. RL controller is activated, and final turning angle is determined by improvisation using the RL approach. It is utilized to determine the cumulative reward depending on all global best locations and the reward value of particular optimum solution.
6. The ideal particle location and searching vector of the MFO approach, as well as the fitness of every individual's optimum response and the global optimum position, are modified.
   The mechanism of correction is as follows:

$$\vec{G}(x) = G(x) - \pi$$

For j^{th} particle, $g\left(x_j^{j+1}\right) = g(x_i) - \pi\,(i)$

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

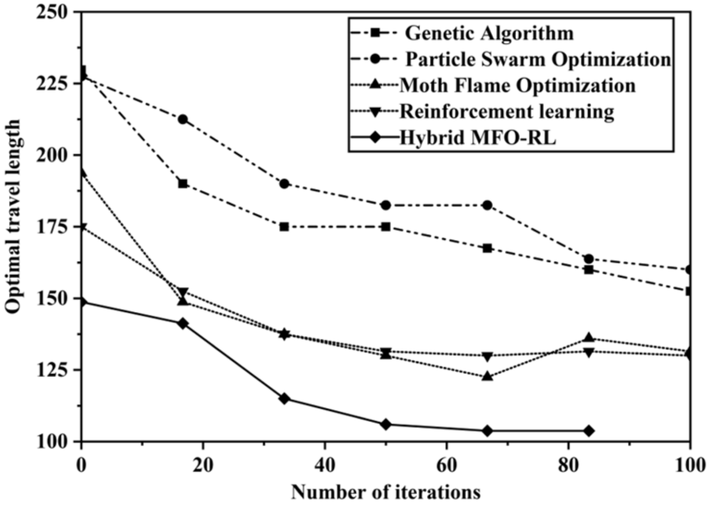

Figure 9. Representation of convergence curve between standalone algorithms and hybrid MFO-RL approach.

7. If there comes a situation where, multi-humanoid robots (dynamic obstacle) come at a single coordinate rise the situation of conflict. In this situation, Petri-Net comes into play and priotize the robot which can reach the target quickest (as described in Fig. 7).
8. Final turning angle is implemented to steer the humanoid out of the conflict zone.
9. Once out of the conflict zone, MFO is activated to sketch out the humanoid's new optimal path.
10. Steps 3–9 are repeated till the humanoid robot reaches the end position.

The optimum route is chosen using the optimal route assessment variable system, with the predefined weight limit variable as the most important criteria.

The RL method will be used to train the MFO approach in this section. The adjustment parameter generated from the total reward, which is called as hybrid MFO-RL approach, maintains the optimum particles and global ideal location fitness of the MFO. Then, using the variables in the optimal route assessment mechanism, this method is implemented to the trajectory-tracking of autonomous navigation robot and an effective optimum route is established. The trajectory planning for the robot is done by using following steps.

The hybrid MFO-RL approach has been developed. Further, it has been tested against few standalone approaches with respect to optimal travel length in number on interations. Five algorithms (genetic algorithm, particle swarm optimization, MFO, RL, and hybrid MFO-RL approach) are tested with reference to travel length. As presented in Fig. 9, graph shows that the hybrid MFO-RL approach provides optimal travel length and too in the least number of iterations. Therefore, further research has been proceeded by taking hybrid MFO-RL approach in consideration.

6. Simulation and experimental results

6.1. Robotic platform

The robotic platform used in the given study is NAO V4 developed by Aldebaran Robotics of France. It is a compact, automated, and configurable robot with 25 degrees of freedom and weighs about 5.2 kg. Various specifications of the robot are given in Table I.

Table I. Specification of robotic platform.

Specifications	NAO V4
Dimensions (h * w * d)	573mm * 275mm * 311mm
Power source	Lithium battery (27.6Wh at 21.6V)
CPU	Intel Atom Z530@1.6Ghz
RAM	1 GB
Storage	2 GB Flash memory + 8 GB Micro SDHC
Programming languages	C++, Python, Java, MATLAB, Urbi, C, .Net
Sensors	4 microphones, 9 tactile sensors, inertial board, 2 IR emitters and receivers, 8 pressure sensors, sonar rangefinder
Connectivity	Ethernet, Wi-Fi
Cameras	2 MT9M114 72.6 degrees DFOV camera

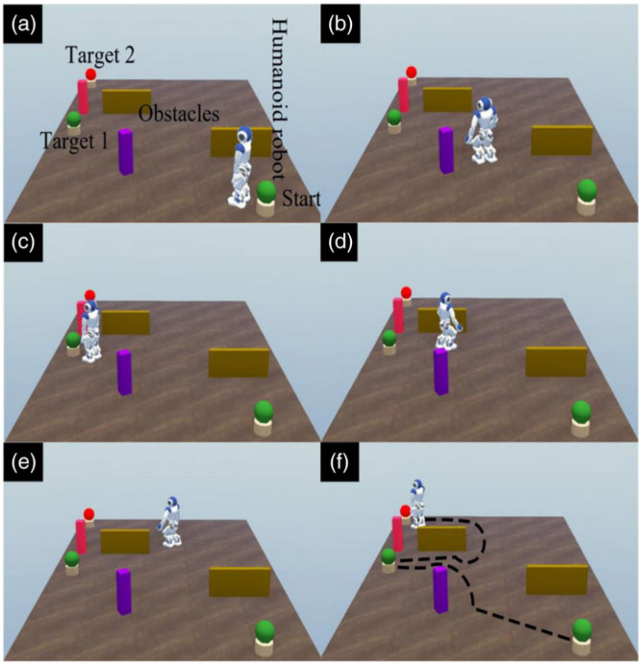

Figure 10. *Simulation result of navigation of humanoid robot using hybrid MFO-RL approach for single robot.*

6.2. Path-planning of a single-humanoid robot using hybrid MFO-RL approach

For the justification of the efficiency of the proposed controller, it is employed for the navigational purpose of a single-humanoid robot in a cluttered environment. The controller is employed in simulated and real NAO. WEBOT, a 3D simulator, is utilized first to test the developed controller. Further, it is verified in real NAO. The comparison of outcomes from simulation and experiment are performed based on navigational parameters. The navigational parameters are recorded from the interface in WEBOT and using measuring instruments like measuring tape and stopwatch for recording the parameters. In Fig. 10, the robot is placed at its initial Start position and proceeds toward the first checkpoint Target T1. Henceforth, it traverses from Target 1 to Target 2. Similar environment conditions are being created for real-time experiments, as shown in Fig. 11. The hybrid model works according to the steps as directed earlier. The path length and time taken are directly measured through the WEBOT interface in

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

Table II. *Comparison of travel length (Tl) and travel time (Tt) for simulations and experiments using hybrid MFO-RL approach for single robot.*

	Simulation		**Experiment**		**Deviation (%)**	
Sl. No.	**Tl (cm)**	**Tt (s)**	**Tl (cm)**	**Tt (s)**	**Tl (cm)**	**Tt (s)**
1.	150.4	60.45	155.4	63.36	3.32	4.81
2.	151.1	61.51	157.4	64.42	4.17	4.73
3.	150.8	61.09	154.3	61.94	2.32	1.39
4.	149.8	59.92	156.1	62.71	4.21	4.66
5.	150.2	60.2	155.8	61.71	3.73	2.51
6.	152.2	60.98	157.8	63.77	3.68	4.58
7.	153.01	62.51	159.9	65.19	4.5	4.29
8.	153.3	62.77	160.1	65.23	4.44	3.92
9.	149.2	59.83	156.4	61.62	4.83	2.99
10.	151	61.38	155.5	63.51	2.98	3.47
Avg.	151.101	61.064	156.87	63.346	3.818	3.735

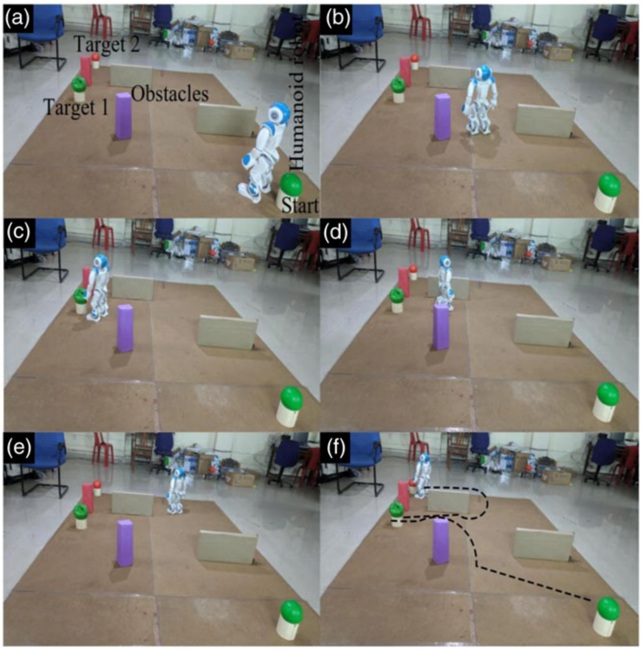

Figure 11. *Experimental result of navigation of humanoid robot using hybrid MFO-RL approach for single robot.*

simulation. While, in experimental demonstration, measuring tape and stopwatch are used to measure path length and travel time.

A tabular data of path length and time taken by the humanoid robot during simulation and experiment show a minimal deviation of about 5% between them as shown in Table II. Hence, it justifies the usage of the proposed hybrid approach. Figures 10 and 11 show that the robot reaches both targets conveniently in simulation and experimental setups.

The deviation of 5% indicates that the result from the simulation is validated and demonstrates the robustness of the proposed controller. The deviation is due to some external disturbance in experimental demonstration that are absent in simulation.

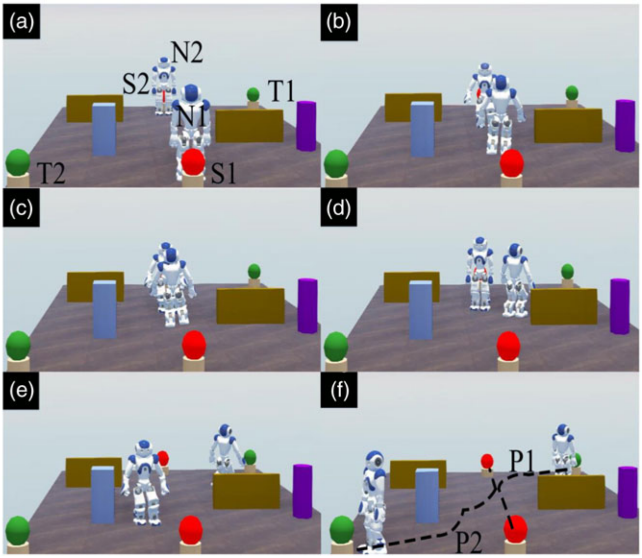

Figure 12. *Simulation result of navigation of humanoid robot using hybrid MFO-RL approach for two robots.*

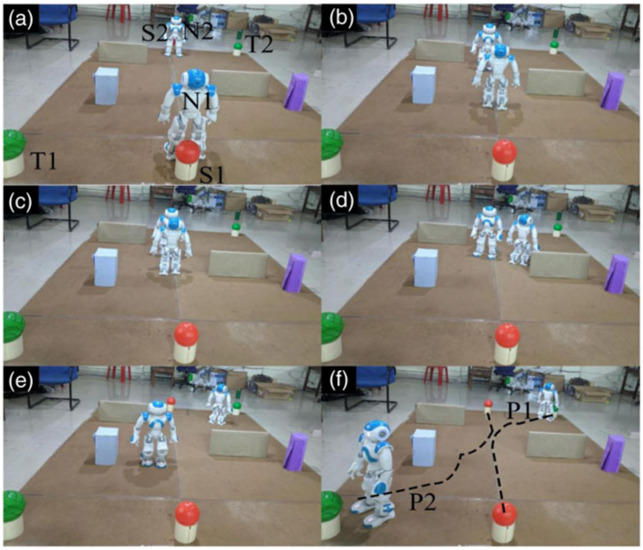

Figure 13. *Experimental result of navigation of humanoid robot using hybrid MFO-RL approach for two robots.*

6.3. Path-planning of multi-humanoid robot using hybrid MFO-RL approach

For the acceptance of the proposed hybrid path planning controller, the approach is used as the prime navigational strategy in the path-planning of two humanoid robots N1 and N2, in an unknown environment. The environment is modeled with various static obstacles, and similar conditions are created for both simulation and real-time experiments, as shown in Figs. 12 and 13. The robots are placed at their initial start positions S1 and S2 and are navigated through the environment to Target T1 and Target T2, respectively. It is observed that although the hybrid approach proves its worth in static obstacle clearance, it reaches a stalemate in the scenario of a dynamic obstacle.

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

Table III. *Comparison of travel length for simulations and experiments using hybrid MFO-RL approach for two robots.*

	Simulation		Experiment		Deviation (%)	
Sl. No.	**N1**	**N2**	**N1**	**N2**	**N1**	**N2**
1.	153.2	159.7	159.5	166.9	4.11	4.51
2.	153.4	159.4	159.6	166.5	4.04	4.45
3.	152.8	159.9	158.9	166.3	3.99	4
4.	152.9	157.2	160.09	164.8	4.7	4.83
5.	153.7	158.3	159.9	165.7	4.03	4.67
6.	153.5	160.2	159.8	166.8	4.1	4.12
7.	153	159.4	159.2	165.7	4.05	3.95
8.	153.1	160.5	158.6	166.3	3.59	3.61
9.	153.8	158.9	160.1	165.4	4.1	4.09
10.	152.6	161.4	158.8	167.7	4.06	3.9
Avg.	153.2	159.49	159.449	166.21	4.077	4.213

Table IV. *Comparison of travel time for simulations and experiments using hybrid MFO-RL approach for two robots.*

	Simulation		Experiment		Deviation (%)	
Sl. No.	**N1**	**N2**	**N1**	**N2**	**N1**	**N2**
1.	62.65	67.81	64.78	70.02	3.4	3.26
2.	63.01	67.25	65.85	69.51	4.51	3.36
3.	61.96	68.12	64.29	70.37	3.76	3.3
4.	62.07	64.95	64.42	66.84	3.79	2.91
5.	63.25	66.12	66.01	68.71	4.36	3.92
6.	63.1	68.55	65.42	70.89	3.68	3.41
7.	62.31	67.28	64.91	70.01	4.17	4.06
8.	62.45	68.7	64.71	71.41	3.62	3.94
9.	63.48	66.03	65.98	68.81	3.94	4.21
10.	61.75	69.28	64.74	71.6	4.84	3.35
Avg.	62.603	67.409	65.111	69.817	4.007	3.572

Thus, the hybrid approach combined Petri-Net controller is encoded in the humanoid robot for dynamic obstacle clearance. On entering the proximity zone of each other, the robots activate the Petri-Net controller, which prioritizes the robots for further navigation based on a given set of rules. The data values about the path length traversed and time taken to complete the two humanoid robots' tasks are recorded in tabular form using a similar method discussed in the previous section. Further analysis confirms that the simulation and experimental values agree with each other, with a deviation of about 6%, which confirms the hybrid approach's robustness and feasibility in multi-humanoid navigation. Tables III and IV show the travel length and time taken obtained through these simulations and experimental results, respectively.

The proposed controller has been examined in the terrain having a single robot and multiple robots. In the multiple robots situation, one robot act as a dynamic obstacle to the other. But in this situation, the programmers do not know which will act as the robot and which will act as a dynamic obstacle. Therefore, a different scenario has been taken where one robot (Knepra III with one obstacle mounted on it) will act as a dynamic obstacle. The proposed controller has been checked in both simulation and experimental scenarios, as shown in Fig. 14 and 15, respectively. The situation shows that the dynamic

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

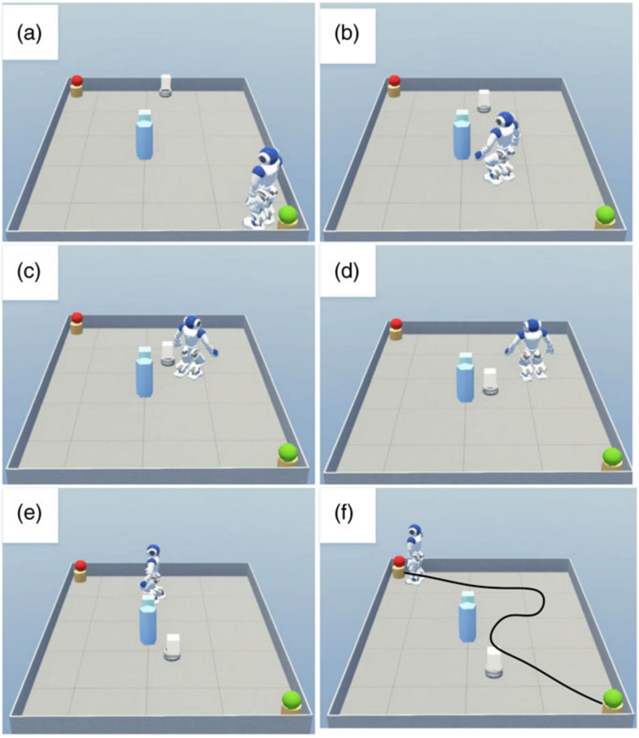

Figure 14. *Simulation result of navigation of humanoid robot using hybrid MFO-RL approach for a dynamic environment.*

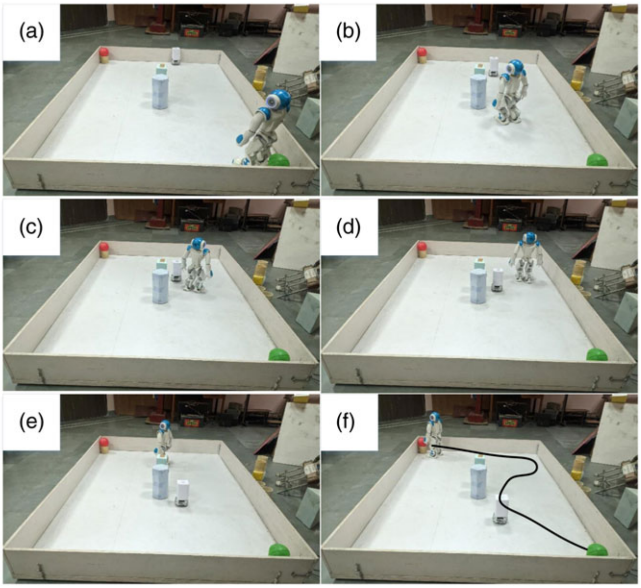

Figure 15. *Experimental result of navigation of humanoid robot using hybrid MFO-RL approach for a dynamic environment.*

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

Table V. *Comparison of travel length (Tl) and travel time (Tt) for simulations and experiments using hybrid MFO-RL approach for a dynamic environment.*

	Simulation		**Experiment**		**Deviation (%)**	
Sl. No.	**Tl (cm)**	**Tt (s)**	**Tl (cm)**	**Tt (s)**	**Tl (cm)**	**Tt (s)**
1.	176.2	73.88	182.5	76.79	3.58	3.94
2.	174.5	72.35	179	74.48	2.58	2.94
3.	175.9	73.46	179.4	74.31	1.99	1.16
4.	174.9	72.29	181.2	75.08	3.6	3.86
5.	175.3	72.57	180.9	74.08	3.19	2.08
6.	175.7	71.95	181.3	74.74	3.19	3.88
7.	176.51	73.48	183.4	76.16	3.9	3.65
8.	175.5	72.82	180.5	75.73	2.85	4
9.	176.8	73.74	183.6	76.2	3.85	3.34
10.	172.7	70.8	179.9	72.59	4.17	2.53
Avg.	175.401	72.734	181.17	75.016	**3.29**	**3.138**

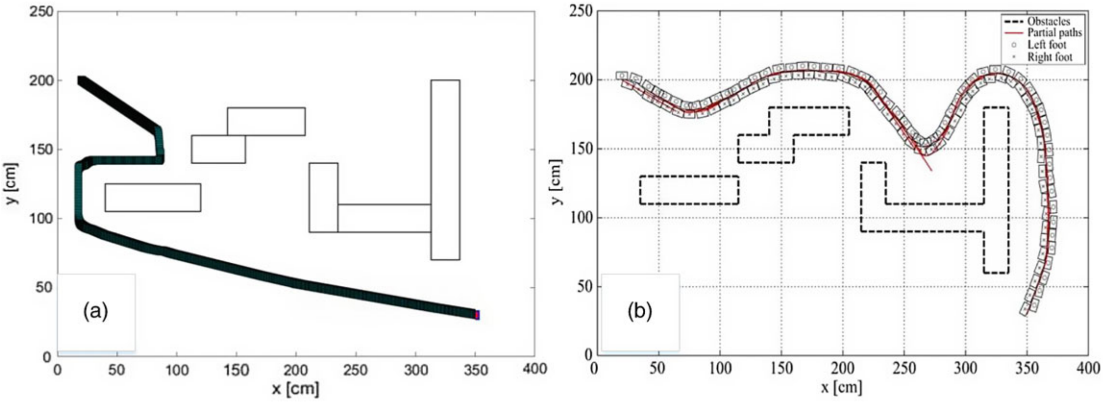

Figure 16. *Comparison of path adaptation of humanoid robot using (a) proposed hybrid MFO-RL approach against the (b) existing approach [35].*

obstacle moves in its path and the humanoid robot need to change its direction to find a safe path to the target. The results displated in tabular form (Table V) demonstrate that the simulation and experimental results are in good relation with each other with deviation under 5% for both travel length and travel time.

7. Comparison

The proposed hybrid MFO-RL controller, in culmination with the Petri-Net controller, is configured into the humanoid robot NAO. The results obtained in lieu of the simulations and real-time experiments show the proposed hybrid approach's efficacy. To analyze the effectiveness and profundity of the path planning approach, it is weighed up against previous works in path planning research. The paper used in lieu of comparison is the online multi-objective evolutionary approach for navigation of humanoid robots [35]. A similar environmental scenario has been created to check the efficiency of the proposed controller. The previously used method and the proposed hybrid approach has been tested on, as shown in Fig. 16. The results obtained show the successful clearance of the obstacles and the attainment of the final goal by both strategies.

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

Table VI. *Comparison of previously developed approach [35] and proposed hybrid MFO-RL approach based on path length.*

Sl. No.	Approach	Path length	Deviation (%)
1.	Online multi-objective evolutionary approach	563 units	**9.8**
2.	Hybrid MFO-RL approach	508 units	

Table VII. *Comparison of previously developed approach [36] and proposed hybrid MFO-RL approach based on path length.*

Sl. No.	Approach	Path length	Deviation (%)
1.	Improved ACO and MDP approach	360.6 units	**11.3**
2.	Hybrid MFO-RL approach	324 units	

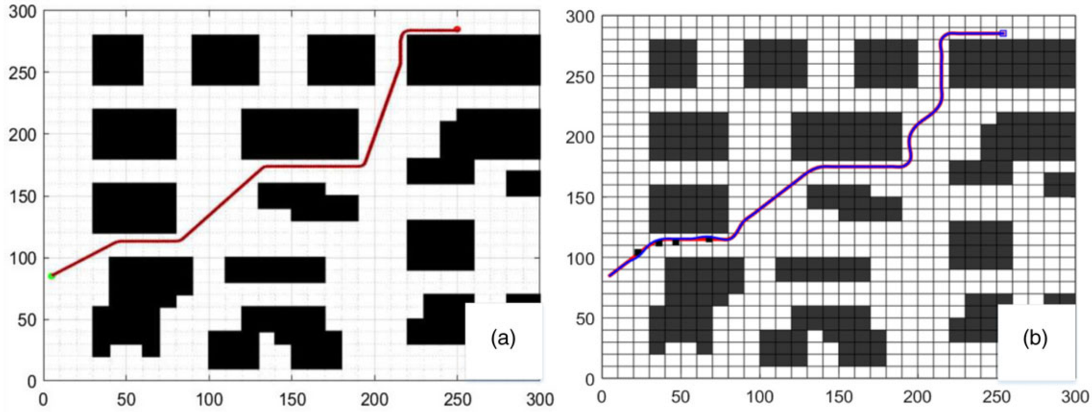

Figure 17. *Comparison of path adaptation of humanoid robot using (a) proposed hybrid MFO-RL approach against the (b) existing approach [36].*

However, the data report's detailed analysis shows that the proposed hybrid approach edges over the past-established method by 9.8% in terms of path length traversed between the initial and final position. The results ensue in lesser computational complexity, making the proposed controller economical. These comparisons prove the method to be highly feasible and robust while using humanoid robot path planning in an unknown environment. Table VI shows the deviation between the two path planning strategies.

The proposed approach has been evaluated with reference to improved ACO and MDP [36] in regard to path length. Just like the previous comparison, utmost care has been taken to replicate the environment. The process has been started by feeding the proposed controller to the robot. It finds its path and reaches the target by avoiding obstacles in its path. Figure 15 (b) shows the path adaption by robot using the improved ACO and MDP approach. The proposed approach is compared in the same environment and presented in Fig. 17(a). The comparison has been made also in tabular form and presented in Table VII. It shows that the overall improvement of path length can be obtained, that is, about 11% to the path length obtained by previously improved ACO and MDP approach.

8. Conclusion

Path planning approach using a hybrid MFO-RL controller is successfully applied and tested on single- and multi-humanoid robot navigation in an unknown environment. A total of 150 epochs were carried

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

out to optimize the training weights in the given approach used for finding out the optimal turning angle of the humanoid robot. The reward function is used to optimize the trajectory selection by each moth. For the path-planning of multi-humanoid robot system in an unknown environment, a Petri-Net controller is configured alongside the MFO-RL controller to prevent deadlock situations during navigation in the terrain. This is accomplished by prioritization among the humanoid robots using the Petri-Net controller. For the analysis of the applied approach's robustness and effectiveness, various factors like the easiness of obstacle avoidance, turning angle, clearance, and task completion time are considered. The hybrid controller is then assessed based on path length with reference to other standalone approaches that displays the superiority of the hybrid MFO-RL approach. The simulation and experimental results in both single- and multi-humanoid robot navigation yield a deviation of around 6%, which is acceptable by common standards. In the case of single-humanoid robot navigation, comparison with a previously established approach demonstrates the superiority of the proposed approach by over 9% reduction in path length covered. Comparison with one more environment has been done based on the path length. In that environment also, the proposed controller comes out as a clear winner. It justifies that the proposed hybrid controller is efficient and feasible for the path-planning of single- and multi-humanoid robots in an unknown environment having multiple targets. The proposed approach can be improvised in the future by combining the proposed controller with a classical path planning approach for navigation over uneven terrains.

Authors contributions. **Abhishek Kumar Kashyap:** Conceptualization, Methodology, Software, Investigation, Software, and Validation. **Dayal R. Parhi**: Supervision, and Writing – Reviewing and Editing. **Vikas Kumar:** Writing – Original draft preparation and Visualization.

Financial support. The authors do not have any financial support

Conflict of interest. The authors declare that they have no conflicts of interest.

Ethical contributions. This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Y. Berquin and A. Zell, "A physics perspective on lidar data assimilation for mobile robots," *Robotica* **40**(4), 862–887 (2022).

[2] S. Wu, Y. Du and Y. Zhang, "Mobile robot path planning based on a generalized wavefront algorithm," *Math. Probl. Eng.* **1**(2), 2020–2012 (2020).

[3] Y. Liang and L. Xu, "Global Path Planning for Mobile Robot Based Genetic Algorithm and Modified Simulated Annealing Algorithm," **In:** *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation* (2009) pp. 303–308.

[4] Z. Zhu, F. Wang, S. He and Y. Sun, "Global path planning of mobile robots using a memetic algorithm," *Int. J. Syst. Sci.* **46**(11), 1982–1993 (2015).

[5] J. Botzheim, Y. Toda and N. Kubota, "Bacterial memetic algorithm for offline path planning of mobile robots," *Memetic Comput.* **4**(1), 73–86 (2012).

[6] X. D. Liang, L. Y. Li, J. G. Wu and H. N. Chen, "Mobile robot path planning based on adaptive bacterial foraging algorithm," *J. Cent. South Univ.* **20**(12), 3391–3400 (2013).

[7] M.-Y. Chen, Y.-J. Wu and H. He, "A novel navigation system for an autonomous mobile robot in an uncertain environment," *Robotica* **40**(3), 421–446 (2022).

[8] M. Gao and J. Tian, "Path Planning for Mobile Robot Based on Improved Simulated Annealing Artificial Neural Network," **In:** *Third International Conference on Natural Computation (ICNC)*, vol. 3 (IEEE, 2007) pp. 8–12.

[9] H. Jun and Z. Qingbao, "Multi-objective Mobile Robot Path Planning Based on Improved Genetic Algorithm," **In:** *International Conference on Intelligent Computation Technology and Automation*, vol. 2 (IEEE, 2010) pp. 752–756.

[10] H. Yue and Z.-M. Wang, "Path Planning of Mobile Robot Based on Compound Shape and Simulated Annealing Hybrid Algorithm," **In:** *IEEE International Conference on Robotics and Biomimetics-ROBIO* (IEEE, 2005) pp. 186–189.

[11] F. Janabi-Sharifi and D. Vinke, "Integration of the Artificial Potential Field Approach with Simulated Annealing for Robot Path Planning," **In:** *Proceedings of 8th IEEE International Symposium on Intelligent Control* (IEEE, 1993) pp. 536–541.

https://doi.org/10.1017/S0263574722001357 Published online by Cambridge University Press

[12] H. Martınez-Alfaro and S. Gomez-Garcıa, “Mobile robot path planning and tracking using simulated annealing and fuzzy logic control,” *Expert Syst. Appl.* **15**(3-4), 421–429 (1998).

[13] K. K. Pandey and D. R. Parhi, “Trajectory planning and the target search by the mobile robot in an environment using a behavior-based neural network approach,” *Robotica* **38**(9), 1627–1641 (2020).

[14] N. Ganganath and C.-T. Cheng, “A 2-dimensional ACO-based Path Planner for Off-Line Robot Path Planning,” **In:** *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (IEEE, 2013) pp. 302–307.

[15] L. Chang, L. Shan, C. Jiang and Y. Dai, “Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment,” *Auton. Robots* **45**(1), 51–76 (2021).

[16] Y. Wei and J. Zhao, “Designing Human-like behaviors for anthropomorphic arm in humanoid robot NAO,” *Robotica* **38**(7), 1205–1226 (2020).

[17] M. Kusuma, Riyanto, C. Machbub, “Humanoid Robot Path Planning and Rerouting Using A-Star Search Algorithm,” **In:** *Proceedings - 2019 IEEE International Conference on Signals and Systems, ICSigSys* 2019 (IEEE, 2019) pp. 110–115.

[18] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto and T. Yoshigahara, “Obstacle Avoidance and Path Planning for Humanoid Robots Using Stereo Vision,” **In:** *IEEE International Conference on Robotics and Automation, Proceedings. ICRA 2004* (IEEE, 2004) pp. 592–597.

[19] W. Huang, J. Kim and C. G. Atkeson, “Energy-based Optimal Step Planning for Humanoids,” **In:** *IEEE International Conference on Robotics and Automation* (IEEE, 2013) pp. 3124–3129.

[20] M. Lee, Y. Heo, J. Park, H. D. Yang, H. D. Jang, P. Benz, H. Park, I. S. Kweon and J. H. Oh, “Fast Perception, Planning, and Execution for a Robotic Butler: Wheeled Humanoid M-Hubo,” **In:** *IEEE International Conference on Intelligent Robots and Systems* (2019) pp. 5444–5451.

[21] K. P. Lagaza, A. K. Kashyap and A. Pandey, “Spider Monkey Optimization Algorithm Based Collision-Free Navigation and Path Optimization for A Mobile Robot in the Static Environment,” **In:** *Advances in Mechanical Engineering* (2020) pp. 1459–1473.

[22] S. M. J. Jalali, R. Hedjam, A. Khosravi, A. A. Heidari, S. Mirjalili and S. Nahavandi, “Autonomous robot navigation using Moth-Flame-Based neuroevolution,” **In:** *Evolutionary Machine Learning Techniques* (Springer 2020) pp. 67–83.

[23] A. Abdullah, M. F. F. A. Rashid, S. G. Ponnambalam and Z. Ghazalli, “Energy efficient modeling and optimization for assembly sequence planning using moth flame optimization,” *Assem. Autom* **39**(2), 356–368 (2019).

[24] S. H. H. Mehne and S. Mirjalili, “Moth-Flame optimization algorithm: theory, literature review, and application in optimal nonlinear feedback control design,” *Nat.-Inspir. Optim.* **811**, 143–166 (2020).

[25] M. A. Elaziz, A. A. Ewees, R. A. Ibrahim and S. Lu, “Opposition-based moth-flame optimization improved by differential evolution for feature selection,” *Math. Comput. Simul.* **168**(4), 48–75 (2020).

[26] P. Gao, Z. Liu, Z. Wu and D. Wang, “A Global Path Planning Algorithm for Robots Using Reinforcement Learning,” **In:** *IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2019) pp. 1693–1698.

[27] M. Fakoor, A. Kosari and M. Jafarzadeh, “Humanoid robot path planning with fuzzy Markov decision processes,” *J. Appl. Res. Technol* **14**(5), 300–310 (2016).

[28] L. A. Trinh, M. Ekström and B. Cürüklü, “Petri Net Based Navigation Planning with Dipole Field and Dynamic Window Approach for Collision Avoidance,” **In:** *6th International Conference on Control, Decision and Information Technologies (CoDIT)* (IEEE, 2019) pp. 1013–1018.

[29] D. R. Parhi and J. C. Mohanta, “Navigational control of several mobile robotic agents using Petri-potential-fuzzy hybrid controller,” *Appl. Soft Comput. J* **11**(4), 3546–3557 (2011).

[30] P. B. Kumar, M. K. Muni and D. R. Parhi, “Navigational analysis of multiple humanoids using a hybrid regression-fuzzy logic control approach in complex terrains,” *Appl. Soft Comput.* **89**(3), 106088 (2020).

[31] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib and A. J. Humaidi, “Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm,” *Appl. Soft Comput.* **89**(4), 106076 (2020).

[32] A. Hosseinzadeh and H. Izadkhah, “Evolutionary approach for mobile robot path planning in complex environment,” *IJCSI Int. J. Comput. Sci. Issues* **7**(8), 1–9 (2010).

[33] F. Alfaverh, M. Denaï and Y. Sun, “Demand response strategy based on reinforcement learning and fuzzy reasoning for home energy management,” *IEEE Access* **8**, 39310–39321 (2020).

[34] J. L. Peterson, “Petri nets,” *ACM Comput. Surv.* **9**(3), 223–252 (1977).

[35] K. B. Lee, H. Myung and J. H. Kim, “Online multiobjective evolutionary approach for navigation of humanoid robots,” *IEEE Trans. Ind. Electron.* **62**(9), 5586–5597 (2015).

[36] H. Ali, D. Gong, M. Wang and X. Dai, “Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in Grid-Based environment,” *Front. Neurorobot.* **14**(July), 1–13 (2020).

Cite this article: A. K. Kashyap, D. R. Parhi and V. Kumar (2023). “Navigation for multi-humanoid using MFO-aided reinforcement learning approach”, Robotica **41**, 346–369. https://doi.org/10.1017/S0263574722001357