

RESEARCH ARTICLE

# Single and multiple humanoid path planning using Hill valley approach applied to gravitational drift in Gravitational search algorithm

Vikas  and Dayal R. Parhi

Robotics Laboratory, Mechanical Engineering Department, National Institute of Technology, Rourkela, Odisha, India  
**Corresponding author:** Vikas; Email: [vikaspe2k8@gmail.com](mailto:vikaspe2k8@gmail.com)

**Received:** 15 December 2022; **Revised:** 19 June 2023; **Accepted:** 29 July 2023; **First published online:** 31 August 2023

**Keywords:** Humanoid navigation; Hill Valley approach; Gravity search algorithm; Hybrid techniques; Dining Philosopher's model

## Abstract

In this paper, the Hill Valley (HV) approach is applied to the drifting masses or agents in the basic Gravitational Search Algorithm (GSA) for the path planning of humanoid robots. The drift in lighter masses toward the heavier mass creates a localized area, where the probability of obtaining a globally optimal solution is very high. So, the present work is focused on exploiting the area to tackle local optima and provide the best steering angle for the humanoids to navigate. The HV approach is applied to the basic GSA model, at the later stages, to improve the overall computational time and cost. The robustness of the proposed controller was tested in both simulation and experimental environments and compared with the previous research. The results obtained from the proposed controller showed a significant improvement in the overall path length and time taken. Path smoothness was also given equal importance during path planning to ensure stability. The multi-robot navigational scheme was performed using the Dining Philosopher's model to avoid dynamic collision among the humanoids. The percentage deviation in the results was within the acceptable limits. To further check the effectiveness of the proposed technique, the proposed approach was compared with the vision-based navigation in danger space.

## 1. Introduction

Humanoid robot path planning is a complex study whose objective and constraints change over time. Working with biped robots is the most promising area in robotic research. Proper stability during locomotion is the fundamental requirement for the successful path planning of humanoid robots. For this, the path traced by the robot should be smooth and free from sudden jerks posed due to sudden turning angles. In the present work, proper care is taken toward the locomotion of the humanoid by assigning them an optimum smooth path for locomotion. The novelty of the work lies in exploiting the converged solution from the basic GSA model to obtain the global optimum solution. Since the work does not employ parallel working artificial intelligence (AI) techniques, the overall computational time is reduced. Here, the HV algorithm is applied to the converged solution from the GSA model to identify the optimal global agent from the lot.

The GSA model [1], as stated in its original work, has shown a better performance than other approaches. Swarm intelligence (SI) techniques, serving as a metaheuristic, uses the population-based search engine to explore multiple search regions in a single run. GSA, one among them, uses a similar approach based on the metaphor of the gravitational force of attraction between the agents. The approach artificially simulates Newton's law of gravity, which says that the force of attraction between the agents is directly proportional to the product of their masses and is inversely proportional to the square of the distance between them. It leads to a situation where the agents in the swarm tend to converge toward each other, which means the optimum global convergence. Several works [2, 3] done in the past have

shown great focus on the optimal solution obtained using the basic GSA technique. Ing et al. [4] applied the GSA approach for the optimal distribution of network configuration. The configuration was carried out using variable photovoltaic generation output and daily load data. Yadav et al. [5] applied the GSA model to solving constrained problems. The algorithm was tested on several state-of-art benchmark problems to evaluate the technique's performance. In a similar approach, Zibanezhad et al. [6] used the basic model for solving QoS-based web service problems. Qualitative measures were implemented to lower memory usage and ensure rapid convergence. Rashedi et al. [7] applied the model to non-linear filter modeling using various filter design parameters. Ramamoorthy et al. [8] used the approach to reduce power loss in electrical systems. Vikas et al. [9] used the approach for the path planning of the robots. The model because of the memory-less approach converged early. However, these models have shown compromised results because of several factors like an exponential decay in gravitational constant ( $G(t)$ ), lack of exploration at later stages, etc.

Moreover, several hybrid controllers [10, 11] were used to counter the drawbacks posed by using the basic GSA technique to deliver a promising result. Sun et al. [12] used the GSA model and genetic algorithm (GA) for image segmentation. The approach was implemented on two benchmark images, and the performance was compared. Other authors [13, 14] used the particle swarm optimization (PSO)-based hybridized technique to maintain a balance between exploring and exploiting the continuous search space. The methods added adaptive values to improve the efficiency of the proposed model. González et al. [15] implemented the Fuzzy controller to enhance the pattern recognition of medical images. Data from echocardiograms were used to design a modular GSA-based neural network to produce recognizable images. Further, the hybridization of the GSA was done to study the diagnosis architecture of COVID-19 disease [16]. The result successfully classified 98% of the test sets through chest X-ray analysis. Haghbayan et al. [17] used the hybrid model to solve diverse variational problems. Shen et al. [18] used the hybrid approach for the optimal function design. However, such mixed techniques are often costly and require extensive computational time. Because of the hybridization at the initial stage, the overall computational time and cost increase. So, in the current work, we have tried to overcome the drawback by using a single-window optimization technique and maximizing its efficiency. The present work is focused on exploiting the model at later stages to explore the best optimal solution.

Another step toward avoiding the local traps from the basic model is by modifying the basic structure of the algorithm. It will ensure a good amount of exploration and exploitation. Previously, several works have tried improving the basic structure of GSA by tuning the different inbuilt parameters. Their prime focus was improving computational time by avoiding using hybrid-based controllers. In their study, Sahu et al. [19] used the PID controller to tune the parameters of GSA to optimize the power control for interconnected power system distribution. The work used automatic tuning of the parameters based on the balance between the exploration and exploitation phase. Eldos et al. [20] focused on the transition functions to properly tune the different parameters of the GSA. Simultaneous comparisons were made using a genetic-based approach (GA) to match the tuning objectives. Wang et al. [21] used chaotic neural oscillators to tune the parameters of GSA. Mittal et al. [22] used a novel best-controlling parameter to balance the exploration and exploitation tradeoff. Similar work was done by other authors [23, 24], where the fuzzy-based controllers were used to tune the parameters of GSA. However, such methods suffer from higher computational requirements and more extensive memory usage. Moreover, improper tuning of the different parameters of the algorithm leads to a further increased risk of local traps.

Contrary to the abovementioned approach, the present work applies the HV approach to the converged result from the basic GSA. Like the other hybrid approach [25, 26], the current work does not simultaneously implement two or more artificial intelligence (AI) techniques to obtain an optimum result. Instead, importance is given to the converged area. The importance of the application is that: "once the agent drifts toward the optimum agent, we can easily apply the HV approach to find the optimum global agent in the lot." In the past, several authors have used the HV approach to identify the optimum solution. The search capability of the approach has helped obtain efficient convergence in different search spaces. Wang et al. [27] used the HV approach to Particle swarm optimization (PSO) technique to optimize non-linear functions. The approach could quickly identify two niching subswarms on the same hill or

merge. In another work [28], the HV approach was applied to the evolutionary algorithm for locating the global optima for a two-phase real-valued multi-modal function. Researchers use the HV approach for several other multi-modal function optimization [29, 30].

So, the research done in the past has tried to implement various hybrid and modified models for solving path planning problems. However, such approaches have the disadvantages of higher computational costs and more extensive memory usage. Also, tuning the different parameters requires careful examination to ensure a proper balance between the exploration and exploitation phases in the search space. So, based on the drawbacks of the techniques mentioned above, the present work is proposed to obtain an optimal path planning solution. The key contribution of the paper is summarized below:

- i. GSA is preferred over PSO and other swarm-based approaches because of its higher convergence rate and memory-less operations.
- ii. The proposed approach applies the HV technique to the drifted agents instead of the entire agents.
- iii. Applying the HV approach to the drifted agents helps reduce the overall computational time and memory usage.
- iv. Randomization is introduced in the HV approach to enhance the searching capability of the proposed model.

The different sections of the paper are outlined as follows. Section 2 of the paper defines the problem of robotic navigation, where the various parameters affecting path planning are studied. Section 3 shows the implementation of different individual and proposed controllers. In the next section (i.e., Section 4), the Dining Philosopher's model is discussed for application to multi-robot navigation. Section 5 discusses the single and multiple humanoid navigations in different terrains, while Section 6 shows the path planning results for the robotic navigation in Section 5. Section 7 of the paper deals with the comparison results of the proposed controller with the existing controllers. Section 8 is the discussion section, while the last section (i.e., Section 9) is the conclusion of the entire path planning strategy.

## 2. Problem definition

The present work uses a programmable medium-sized humanoid NAO [31] robot (version 4.0) for the path planning objective. The robot was developed by Aldebaran Robotics group, France, and carries a fully capable computer on-board with various sensors like infrared sensors, ultrasonic sensors, etc., mounted on the different parts of the robot. In the present work, ultrasonic range sensors are used to identify the distance of various obstacles within the environment. The angle of detection of the ultrasonic sensors is  $120^\circ$ , as shown by the cone angle in Fig. 1. Based on the distance of the obstacles, the humanoid takes different actions to achieve an optimum path for navigation.

The problem definition for humanoid navigation is formulated based on the reactive path planning approach. The following points are considered while carrying out the trajectory planning:

- i. The humanoids are aware of the start/stop locations during their journey. While the location of the obstacles is obtained using ultrasonic sensors mounted on the chest of the humanoids.
- ii. The humanoids, during their journey, will always try to align their heading angle in the direction of the goal.
- iii. The robot follows various reactive behaviors like obstacle avoidance, barrier following, and target seeking for effective path planning.
- iv. In case of conflicts between the humanoids, the Dining Philosopher's model gets evoked to ensure proper communication among the robots.

Several factors come into the light while ensuring an optimum path for navigation. So, the present work is focused on optimizing the different individual factors/objective functions affecting the overall path planning of humanoid robots. These functions are discussed below.

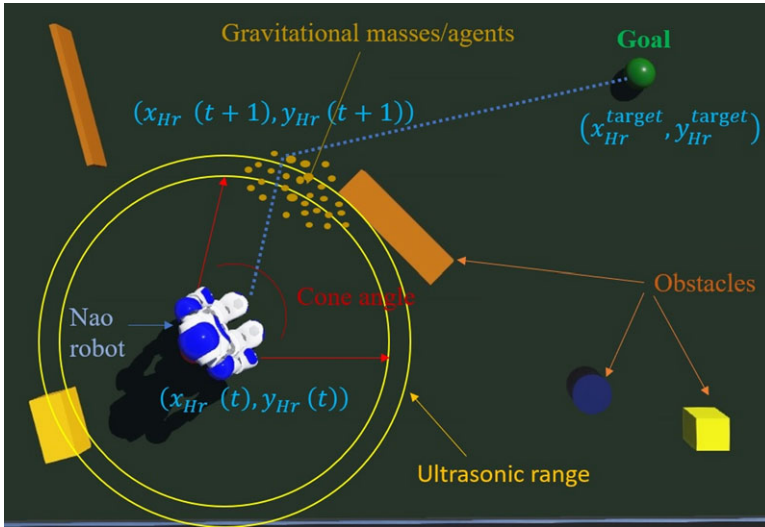


Figure 1. Top view showing NAO's interaction with the surrounding.

2.1. Shortest path length

Consider an environment, see Fig. 1, where the humanoid is at any random location  $(x_{Hr}(t), y_{Hr}(t))$ . The humanoid has to travel to the next position  $(x_{Hr}(t + 1), y_{Hr}(t + 1))$  to reach the target at  $(x_{Hr}^{target}, y_{Hr}^{target})$ . However, in its journey, the humanoid encounters an obstacle placed at  $(x_o, y_o)$ . So, the objective function  $f_1(x, y)$ , defining the path length, is given by the expression shown in Eq. (1).

$$f_1(x, y) = \sum_{Hr=1}^n \left\{ \sqrt{((x_{Hr}(t) - x_{Hr}(t + 1))^2 + (y_{Hr}(t) - y_{Hr}(t + 1))^2)} + \sqrt{((x_{Hr}(t + 1) - x_{Hr}^{target})^2 + (y_{Hr}(t + 1) - y_{Hr}^{target})^2)} \right\} \tag{1}$$

Here,  $Hr \in [1, n]$ , where  $n$  is the maximum number of humanoids involved in path planning.

2.2. Obstacle avoidance behavior

The obstacle avoidance behavior [32] is generally based on the repulsive action of the humanoid as soon as it encounters an obstacle in its journey. The minimum threshold distance to avoid collision is assumed to be  $d_{min}(x_{Hr-o})$  to avoid collision with obstacles. So, the expression shown in Eq. (2) gives the function defining the repulsive behavior.

$$f_2(x, y) = \alpha \left( \frac{1}{d_{min}(x_{Hr-o})} \right) \tag{2}$$

Here,  $\alpha$  is a positive constant, and it is desired to keep a safe distance of the robot from the obstacle.

2.3. Path smoothness

Path smoothness [32] is essential for gaining the stability of the humanoids and is defined as the angle formed between two hypothetical lines which connect the target location and two consecutive robot's positions in each iteration,  $j$ , given by  $global\_best^j$  and  $global\_best^{j-1}$ . The objective function is expressed as shown in Eq. (3).

$$f_3 = \frac{\cos^{-1} \left[ (x_{Hr}(t) - x_{Hr}^{target}) \cdot (x_{Hr}^{global\_best^{j-1}} - x_{Hr}^{target}) + (y_{Hr}(t) - y_{Hr}^{target}) \cdot (y_{Hr}^{global\_best^{j-1}} - y_{Hr}^{target}) \right]}{\sqrt{(x_{Hr}(t) - x_{Hr}^{target})^2 + (y_{Hr}(t) - y_{Hr}^{target})^2} \times \sqrt{(x_{Hr}^{global\_best^{j-1}} - x_{Hr}^{target})^2 + (y_{Hr}^{global\_best^{j-1}} - y_{Hr}^{target})^2}} \tag{3}$$

**2.4. Overall objective function**

The overall objective function is the weighted sum of the individual functions representing the shortest path, static obstacle repulsion, and path smoothness defined by  $w_1$ ,  $w_2$ , and  $w_3$ , respectively.

$$f = w_1f_1 + w_2f_2 + w_3f_3 \tag{4}$$

The above fitness function is a minimization problem to achieve an optimum trajectory path. The Humanoid path planning is a very challenging task that needs the simultaneous optimization of the different individual functions to obtain a final optimized result. Focusing on optimizing a single variable may compromise the overall result and act as a threat to the overall path planning of the Humanoid. So, the different individual functions are combined into a single multi-objective optimization function with weights assigned to the individual functions. For a shorter path length, it is desired to select  $w_1$  closer to 1. In a similar way, for the robots to avoid the obstacle from far away, it is desired to select the value of  $w_2$  closer to 1. Similar to other weights, the weight associated with smoothness value was also varied, but was found to be effective at 0.1. The analysis using the different values of weights analyzed that their preminent values are  $w_1 = 0.5$ ,  $w_2 = 0.4$ ,  $w_3 = 0.1$  for an optimal fitness function.

**3. Proposed controller**

**3.1. Gravity search approach**

GSA, a Swarm intelligence (SI) technique [33], was introduced by Rashedi et al. in 2009 to solve several continuous non-linear functions. The algorithm is based on the metaphor of mass interaction due to gravitational force. The algorithm can efficiently solve various optimization problems as a global search optimization technique. What makes the algorithm different is the rules that apply to the agents while updating their positions. Each agent in the population can be represented by Eq. (5).

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^N) \quad i = 1, 2, \dots, N \tag{5}$$

where  $x_i^d$  is the location of  $i$ th masses/agents in  $d$ th dimensional search space, and  $N$  represents the agents.

As per Newton’s law, the gravitational force acting on any  $i$ th agent because of any  $j$ th agent separated by  $R_{ij}(t)$  is given by Eq. (6).

$$F_{ij}^d(t) = G(t) \frac{M_j(t)M_i(t)}{\varepsilon + R_{ij}(t)} (x_j^d(t) - x_i^d(t)) \tag{6}$$

where  $\varepsilon$  is a constant, and  $G(t)$  is the gravitational constant given by Eq. (7).

$$G(t) = G_0 e^{\frac{-\beta t}{t_{max}}} \tag{7}$$

where  $\beta$  is a user-defined value, and  $G_0$  is the initial value. The Euclidean distance between the two agents,  $i$  and  $j$ , is given by Eq. (8) below.

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \tag{8}$$

To give randomization to the overall problem, Eq. (6) is can be further re-written as:

$$F_i^d(t) = \sum_{j=1, j \neq i}^N \text{rand}_j F_{ij}^d(t) \tag{9}$$

where  $\text{rand}_j$  is a random number in the interval  $[0,1]$ . Because of the force of attraction between the agents, the small mass agents start drifting toward the heavier masses, whose acceleration can be

obtained using Newton's law of motion.

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (10)$$

where  $M_{ii}(t)$  is the inertial mass of agent  $i$  at  $t$ th iteration given by Eq. (11) below.

$$M_i(t) = \frac{\frac{(\text{fit}_i(t) - \text{worst}(t))}{(\text{best}(t) - \text{worst}(t))}}{\sum_{j=1}^N m_j(t)} \quad (11)$$

where  $\text{fit}_i(t)$  represents the fitness function of  $i$ th agent at  $t$ th iteration. For the minimization problem, the best functions yield a minimum value of the overall function shown by Eq. (4).

The velocity and position of the agents can be further updated as shown in Eqs. (12) and (13) below.

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t) \quad (12)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (13)$$

Thus, the lighter mass gets displaced toward the heavier ones leading to drift in population. The steering angle is proposed based on identifying the heavier mass in the population. The position of the different masses gets updated, with subsequent iterations, until the best optimum mass is found. The present work is focused on this drift and is explained in the coming section of the paper.

### 3.2. Hill Valley approach

The HV approach was proposed [17] for solving complex functions, keeping in mind the various objective functions. The approach emphasized the fitness of two solutions to check if they lie in the same niche. This allows to determine whether the solution is part of a hill or valley and concludes the optimization scheme accordingly.

The functional approach [34] of the HV technique can be easily understood from Fig. 2. Firstly, two input points are supplied to the algorithm, and then the HV technique generates new points (called identity/test points), either randomly or sequentially, as per Eqs. (14) and (15), respectively. Here, the objective is to minimize the overall fitness function described in Eq. (4). For a set of identity points generated, if their (e.g.,  $i_4$ ,  $i_5$ , and  $i_6$ ) fitness value is higher than the minimum fitness of the input points (e.g.,  $i/p_3$  and  $i/p_4$ ), then input points are assumed to be in the same niche. In other words, input points (e.g.,  $i/p_3$  and  $i/p_4$ ) can be claimed to belong to the same peak. However, if there is at least one identity point (e.g.,  $i_3$ ), whose fitness is minimum, then the fitness of the input points (e.g.,  $i/p_1$  and  $i/p_2$ ) can be claimed to be in different niches. Hence, these two input points do not belong to the same niche. The overall concepts are described in Section 3.3, where the proposed controller is implemented.

#### 3.2.1. Random HV approach

The approach uses random identity points generated from the two input points. In the present section, we discuss the working of the random HV approach, and later on, we find its effectiveness w.r.t. the counterparts. The identity point  $q_i$  [17] is computed according to Eq. (14) below.

$$q_i = a + \text{rand}_i \times (b - a) \quad (14)$$

Here,  $a$  and  $b$  are two input points, and  $\text{rand}_i$  is a random number in the interval  $[0,1]$ .

#### 3.2.2. Sequential HV approach

Like the random HV approach, the sequential approach uses test points based on the two inputs supplied. However, because of the implementation of a sequential operator  $\Delta_i$ , the test points [17] are sequentially generated.

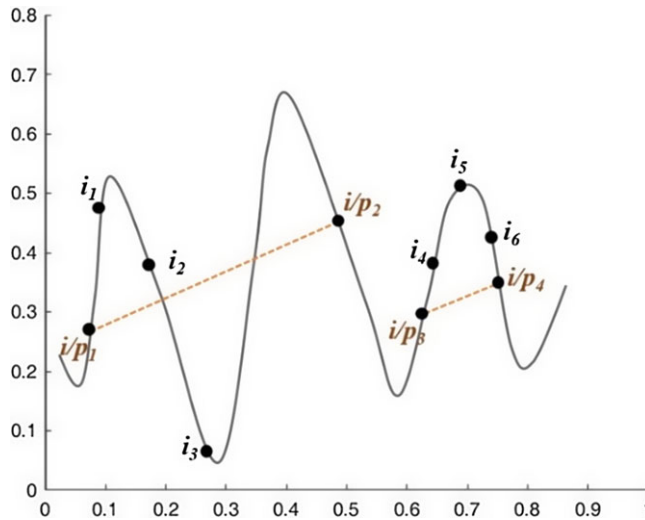


Figure 2. Working of the Hill Valley approach.

$$r_i = a + \Delta_i \times (b - a) \tag{15}$$

$$\text{where } \Delta_i = \frac{i}{(m + 1)}, \text{ and } i = 1, 2, \dots, m \tag{16}$$

Here,  $a$  and  $b$  are again the input points supplied.

### 3.3. Implementation of the proposed controller

The proposed technique, implemented in the current study, is a cumulative analysis of both GSA and HV approaches.

Unlike the previous hybrid techniques, the current approach applies the HV to the drifted mass, not the entire population. It helps in improving the overall computational time and the cost of operation. Fig. 3 shows the working mechanism of the proposed controller [18]. As the humanoid starts its journey (location marked by a red dot in Fig. 3) toward the goal, it encounters an obstacle in its path. As a result, the GSA controller gets activated, and artificial mass/agents are deposited close to the obstacle. With iterations, the agents align themselves, so the lighter mass gets drifted toward the heavier ones, leading to a population drift. With successive iterations, the agents align themselves, and the location of the best agent (or heavier mass) (i.e.,  $X_{GSA}$ ) is obtained. This agent (i.e.,  $X_{GSA}$ ) serves as a local optimum solution obtained using the GSA model.

The HV approach is then applied to the drifted mass, where a circle of radius,  $\rho$  is generated about the local best agent ( $X_{GSA}$ ) [18]. The two ends of the loop,  $(X_{GSA} + \rho)$  and  $(X_{GSA} - \rho)$ , serve as the two inputs to the HV approach. Based on the location of these inputs, the identity points are obtained, either randomly or sequentially, and their fitness is evaluated as per Section 3.2. If the fitness of the randomly or sequentially generated identity points lies in the same niche, then the radius,  $\rho$  can be further reduced to generate new identity points. However, if they lie in a different niche, then the fitness of the identity point lying in different niche (having position say,  $X_{HV1}$ ,  $X_{HV2}$ ) is compared with the fitness of the solution obtained using the GSA (i.e.,  $X_{GSA}$ ). If  $X_{HV1}$  provides a more optimal solution than  $X_{GSA}$ , then our new  $X_{GSA}$  will become  $X_{HV1}$  (i.e.,  $X_{GSA} = X_{HV1}$ ). The next search operation will be performed about this new optimum solution (i.e.,  $X_{GSA} = X_{HV1}$ ). The steps are repeated, the radius is decreased, and the comparison scheme is followed until the best solution ( $X_{GSA-HV}$ ) is achieved. To reach the goal, the humanoid takes the steering angle as per the location of the final global optimum solution ( $X_{GSA-HV}$ ). The flowchart of the

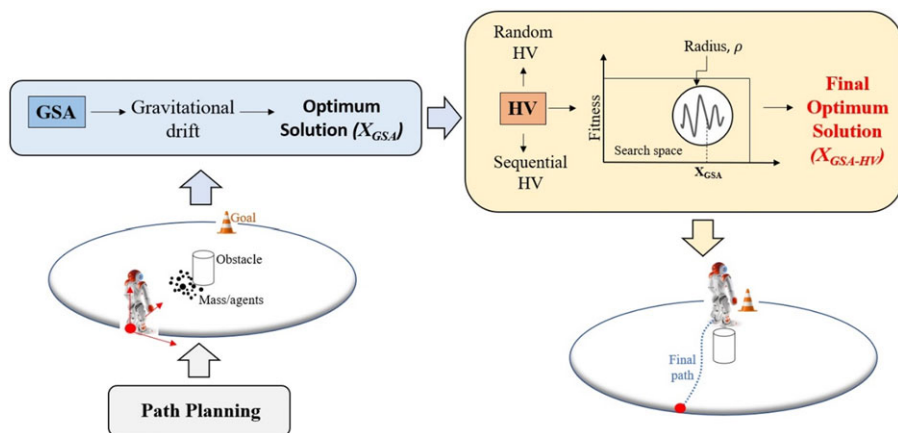


Figure 3. Implementation of the proposed controller.

proposed technique is shown in Fig. 4, where the left side of the flowchart represents the GSA technique while the right side represents the HV approach.

#### 4. Dining Philosopher’s model for multi-robot navigation

Humanoid path planning becomes a complex task when multi-robot navigation is performed. The system requires accurate decisions among the humanoids to avoid collisions during navigation. The Dining Philosopher’s model [35, 36] was introduced to counter this challenge and achieve a collision-free path on a common platform. The model runs on a set of rules, where the individual robot treats the other as a dynamic obstacle. The robot’s different stages are represented by the humanoid’s guiding position to reach its goal.

The Dining Philosopher’s model is an approach, that helps integrate the proposed model in circumstances dealing with multiple Humanoid locomotion. The multiple Humanoid locomotion on a single terrain is a very complex and challenging path planning task, that needs proper communication and supervision among the Humanoids to avoid any dynamic collision among them. Also, the Dining Philosopher’s model is aimed to improve the overall path planning by assigning priority to the robots, based on reducing the overall path time. Thus, as soon as the path planning begins, the humanoids begin to move toward their respective goal location. However, as the static obstacles are encountered, the proposed controller then searches for the optimal mass/agent which can have an optimal fitness value. Thus, the process continues unless situations of dynamic obstacle are encountered. In our work, one NAO acts as a dynamic obstacle to another. Thus, in such a situation (when the NAOs are within the threshold distance), the Petri-Net model is evoked. As a result, the priority is assigned to the robots based on the distance to goal and presence of any further obstacle in the coming journey. The less prioritized robot stops, and the robot with a higher priority walk to the respective target. Once the prioritized robot comes out of the threshold distance of the other robot, the other robot starts to walk toward its goal location. The process continues till the different robots reach their respective goal.

The systematic diagram of the configuration is shown in Fig. 5. The red dot represents the initial guide stage of the individual robots from where they start their journey. This is called the guiding stage 1 of the robots. Here, the humanoids are positioned at random locations and unaware of others’ locations. This initial stage is where the robot waits for its journey to start. The next guiding position (i.e., guide stage 2) is the stage where the robot starts its journey avoiding static obstacles to reach the goal. On further ascending in the journey (i.e., guide stage 3), the humanoids may encounter other humanoids in their path. This is called the dead-lock situation, where the next stage, that is, guide position-4, is initiated. In the next stage (i.e., guiding stage 4), the robots start negotiations based on the priority of the overall path. Here, negotiations are done based on the robot’s distance from the target or any further



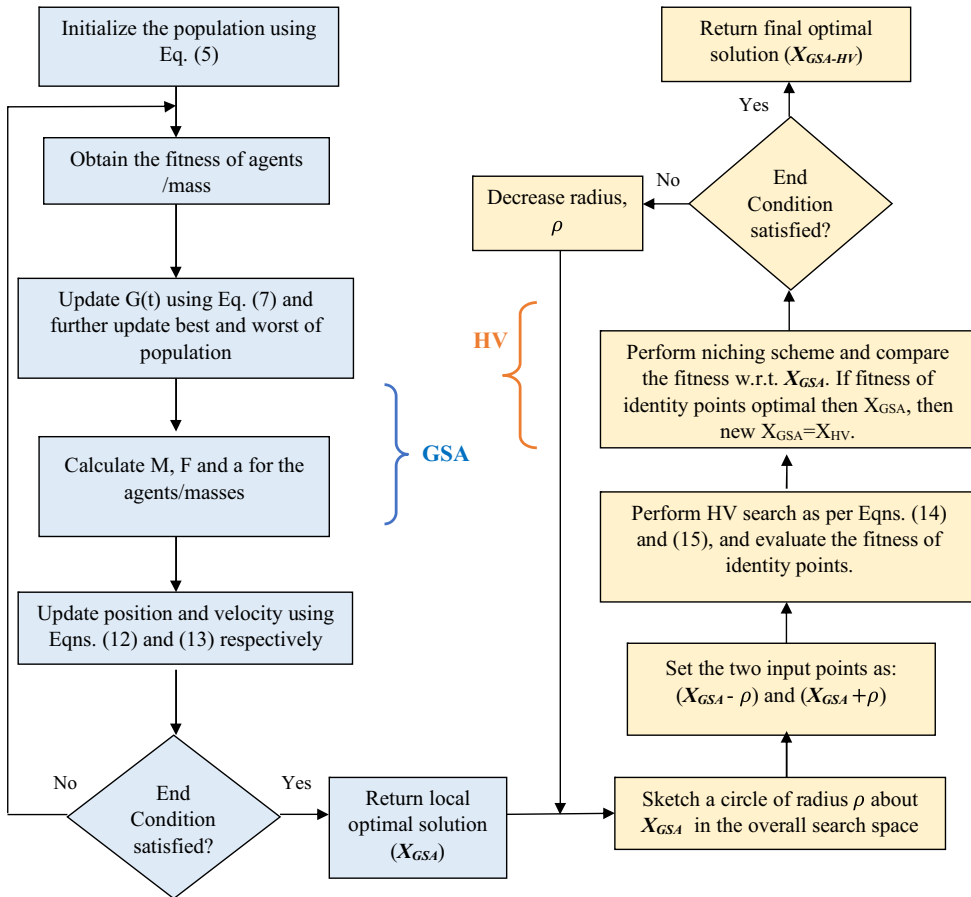


Figure 4. Flowchart of the proposed technique.

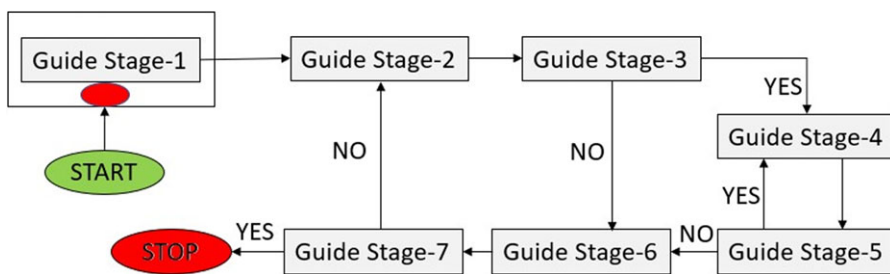


Figure 5. Systematic diagram of Dining Philosopher's model.

obstacles in the path. Next (i.e., guide stage 5), the prioritized robot starts to walk toward the goal while the non-prioritized robot waits. In guide stage 6, the non-prioritized robot starts moving toward the goal. If any further obstacle is reencountered, the guide stage 2 is initiated. The process continues till the robot reaches its respective targets.

### 5. Humanoid navigation

Humanoid navigation becomes complex when the situation of dynamic collision is encountered. In such a situation, the system seeks the Dining Philosopher's model to avoid dynamic collision by ensuring an

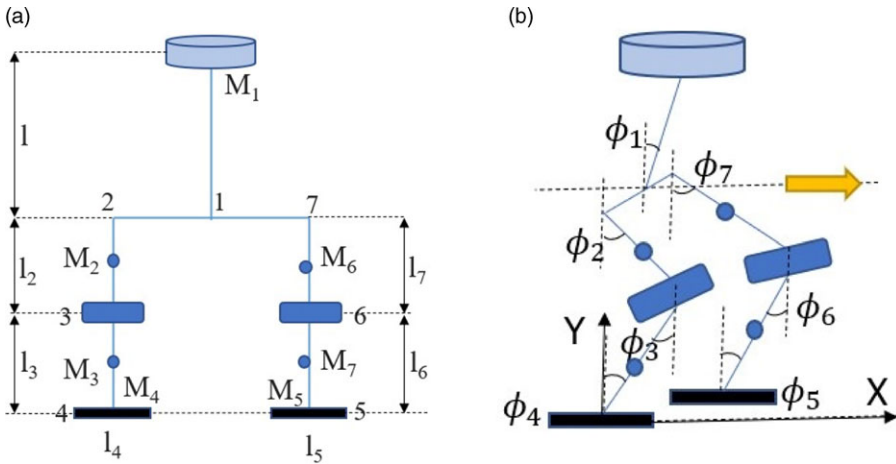


Figure 6. Analysis of humanoid locomotion.

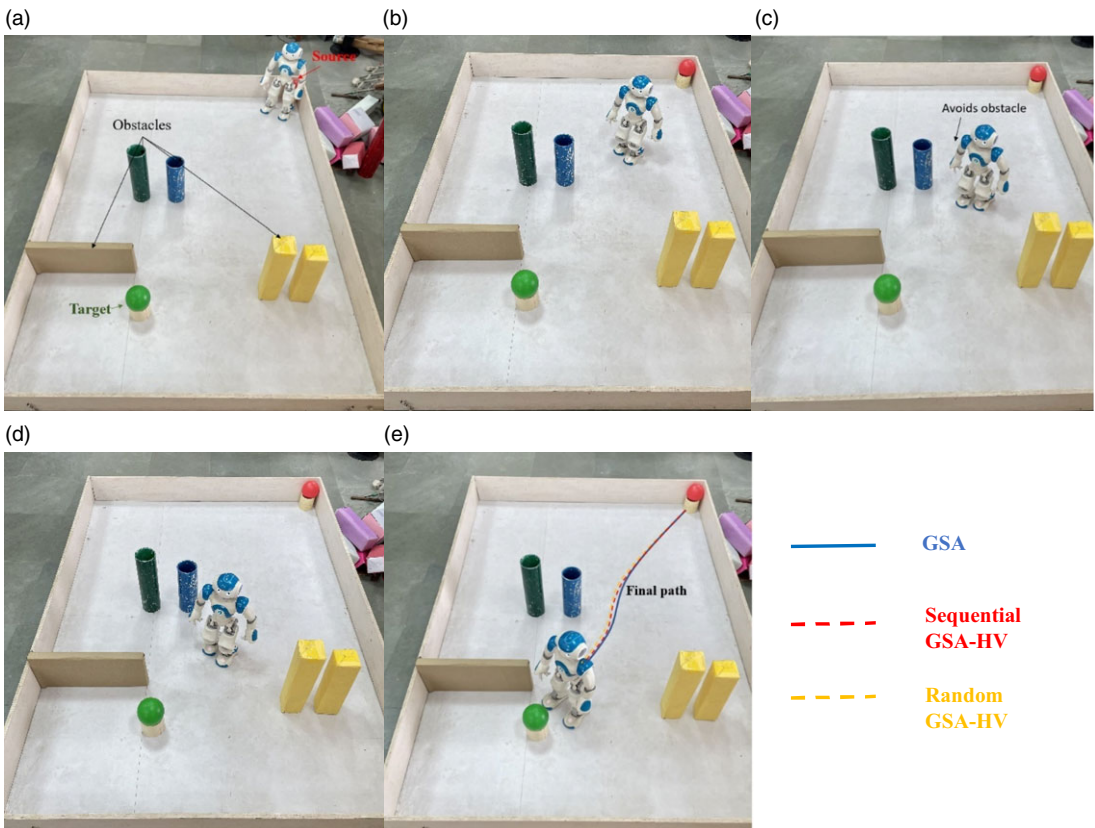


Figure 7. Experimental path traced using single-robot navigation.

optimal decision between the robots. The various navigational schemes are discussed in this section for the path planning of the humanoid. An arena of dimension  $250 \times 200 \text{ cm}^2$  is used as the navigational platform, with several random obstacles. The navigational schemes are tested on a common platform, with and without the occurrence of dynamic obstacles. However, the static obstacles of different shapes and sizes are located randomly on the platform. Path planning is carried out on both simulation and

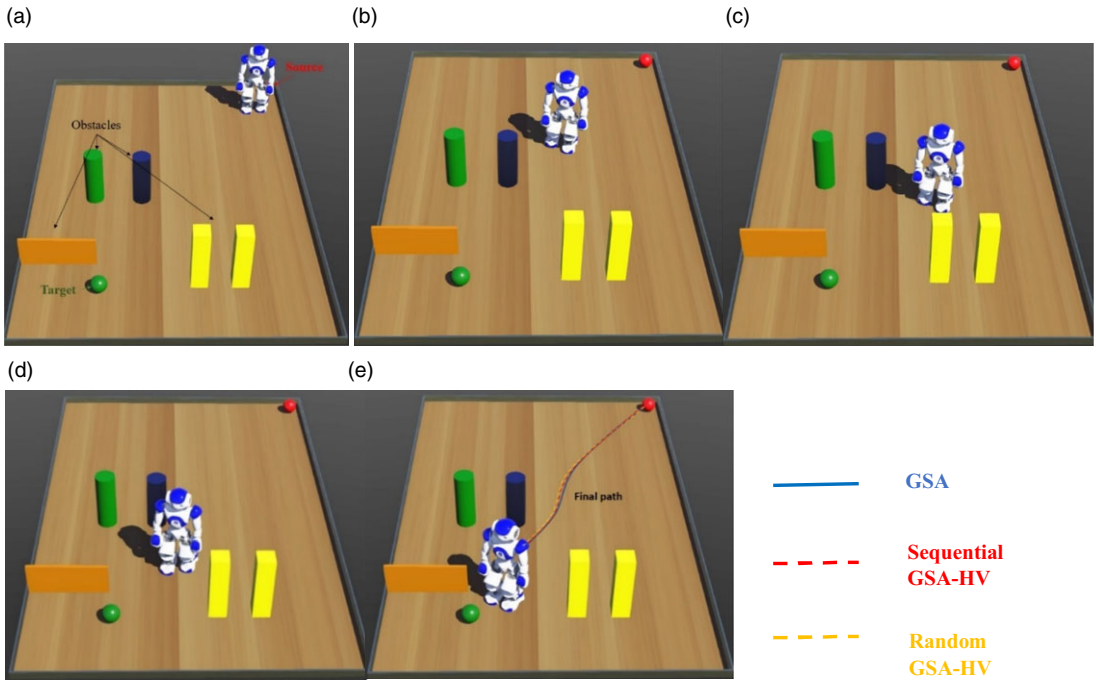


Figure 8. Simulation path traced using single-robot navigation.

experimental platforms. The simulation is done using the Webot R2021a simulation tool, and the coding is done in MATLAB R2019b software. Moreover, in the lab conditions, the humanoid NAO connects to the Choregraphe 2.8.6.23 experimental tool, and the coding is done using Python software.

The Humanoid robot is a complex multiple DOF system, which demands a proper stability during walking. The NAO walking comprises of single support phase and double support phase, depending upon whether both the feet are in contact with the ground, or only one foot is in contact with ground, while other is lifted. The entire cycle helps in the walking of the NAO robots in different terrains. The mass of the different joints along with their length and angles w.r.t. the reference axis plays an important role in deciding its stability on different complex terrains. Consider a Humanoid NAO, with length and masses of different joints, as shown in Fig. 6(a). The masses of the left upper and lower limbs are denoted as  $M_7$  and  $M_6$ , respectively. Similarly, the right upper and lower limbs are denoted by  $M_2$  and  $M_3$ , respectively. Further, the torso mass is given by  $M_1$ . The masses of the right and left feet are given by  $M_4$  and  $M_5$ . Also, the length of torso, left upper and lower limb, right upper and lower limbs, right and left feet is given by  $l, l_7, l_6, l_2, l_3, l_4$ , and  $l_5$ , respectively. The angle between the different joints of the Humanoid NAO is obtained using the inverse kinematics. The different angle posed w.r.t. the vertical axis is shown in Fig. 6(b). The expression for the static and dynamic analysis of the Humanoid is inspired from the work done by Kashyap et al. [36]. So, for the stable locomotion of the NAO robot, the different joint angles are obtained, as shown below.

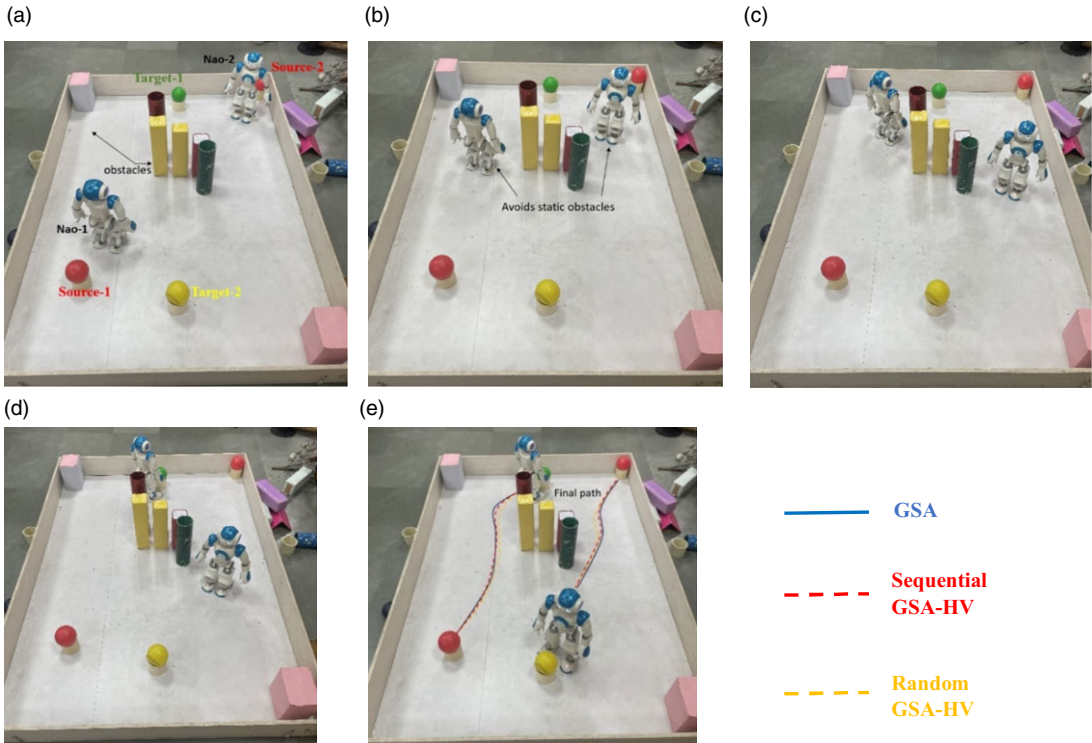
$$N_1 = (l_3 \sin \phi_3 + l_2 \sin \phi_2) \left( l_3 + l_2 \left( \frac{(l_3 \cos \phi_3 + l_2 \cos \phi_2)^2 + (l_3 \sin \phi_3 + l_2 \sin \phi_2)^2 - l_3^2 - l_2^2}{2l_2l_3} \right) \right) \quad (17)$$

$$N_2 = (l_2 \sin \phi_1)(l_3 \sin \phi_3 + l_2 \sin \phi_2) \quad (18)$$

$$D_1 = \left\{ l_3 + l_2 \left( \frac{(l_3 \cos \phi_3 + l_2 \cos \phi_2)^2 + (l_3 \sin \phi_3 + l_2 \sin \phi_2)^2 - l_3^2 - l_2^2}{2l_2l_3} \right) \right\} \quad (19)$$

**Table I.** Navigation results for single-robot system.

| Section 5.1    | Path length (cm) |                     |                         |
|----------------|------------------|---------------------|-------------------------|
|                | GSA              | Random-based GSA-HV | Sequential-based GSA-HV |
| Experiment     | 350.2            | 330.1               | 332.3                   |
|                | 351.5            | 331.5               | 333.4                   |
|                | 349.8            | 331.8               | 333.8                   |
| Simulation     | 336.8            | 315.5               | 317.5                   |
|                | 335.2            | 316.1               | 318.8                   |
|                | 335.8            | 315.8               | 318.4                   |
| Avg. Exp. path | 350.8            | 331.1               | 333.2                   |
| Avg. Sim. path | 335.9            | 315.8               | 318.2                   |
| % Deviation    | 4.4              | 4.8                 | 4.7                     |
| Avg. Exp. time | 47.5             | 40.2                | 41.6                    |
| Avg. Sim. time | 45.1             | 38.4                | 39.5                    |
| % Deviation    | 5.3              | 4.7                 | 5.3                     |



**Figure 9.** Experimental path traced using multi-robot navigation without dynamic obstacle.

$$D_2 = (l_2 \sin \phi_1) \tag{20}$$

So,

$$\phi_3 = \cos^{-1} \left( \frac{N_1 + N_2}{D_1^2 + D_2^2} \right) \tag{21}$$

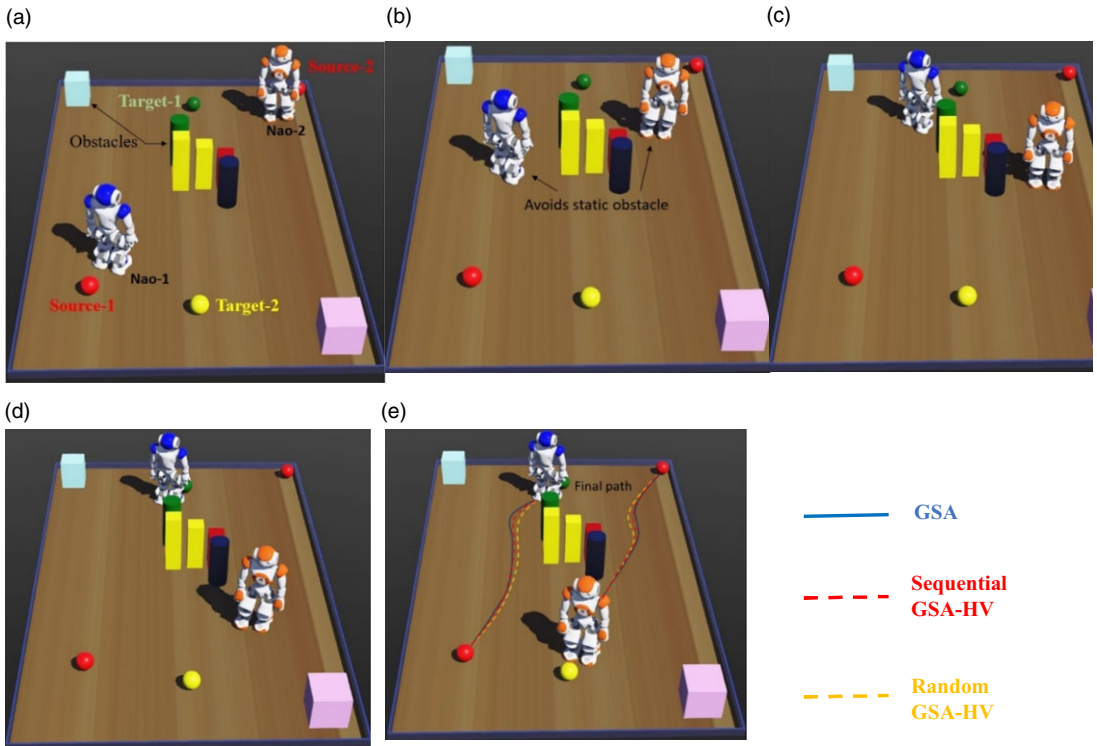


Figure 10. Simulation path traced using multi-robot navigation without dynamic obstacle.

Similarly,

$$N_3 = (l_7 \sin \phi_7 + l_6 \sin \phi_6) \left( l_6 + l_7 \left( \frac{(l_7 \sin \phi_7 + l_6 \sin \phi_6)^2 + (l_7 \sin \phi_7 + l_6 \sin \phi_6)^2 - l_6^2 - l_7^2}{2l_6l_7} \right) \right) \quad (22)$$

$$N_4 = (l_7 \sin \phi_7)(l_7 \cos \phi_7 + l_6 \cos \phi_6) \quad (23)$$

$$D_3 = \left\{ l_6 + l_7 \left( \frac{(l_3 \cos \phi_3 + l_2 \cos \phi_2)^2 + (l_3 \sin \phi_3 + l_2 \sin \phi_2)^2 - l_3^2 - l_2^2}{2l_6l_7} \right) \right\} \quad (24)$$

$$D_4 = (l_7 \sin \phi_7) \quad (25)$$

$$\phi_6 = \cos^{-1} \left( \frac{N_3 + N_4}{D_3^2 + D_4^2} \right) \quad (26)$$

Here,  $\phi_4 = \phi_5 = 0$

$$\phi_7 = \phi_6 - \cos^{-1} \left( \frac{(l_7 \sin \phi_7 + l_6 \sin \phi_6)^2 + (l_7 \sin \phi_7 + l_6 \sin \phi_6)^2 - l_6^2 - l_7^2}{2l_6l_7} \right) \quad (27)$$

and,

$$\phi_2 = \phi_3 - \cos^{-1} \left( \frac{(l_2 \sin \phi_2 + l_3 \sin \phi_3)^2 + (l_2 \sin \phi_2 + l_3 \sin \phi_3)^2 - l_3^2 - l_2^2}{2l_2l_3} \right) \quad (28)$$

The stability of the Humanoid NAO can be evaluated based on the Zero Moment Point (ZMP) equations while walking on different terrains. The ZMP parameter is used to evaluate the Dynamic Balance Margin that ensures the stability of the robot while walking on different terrains. It is calculated from

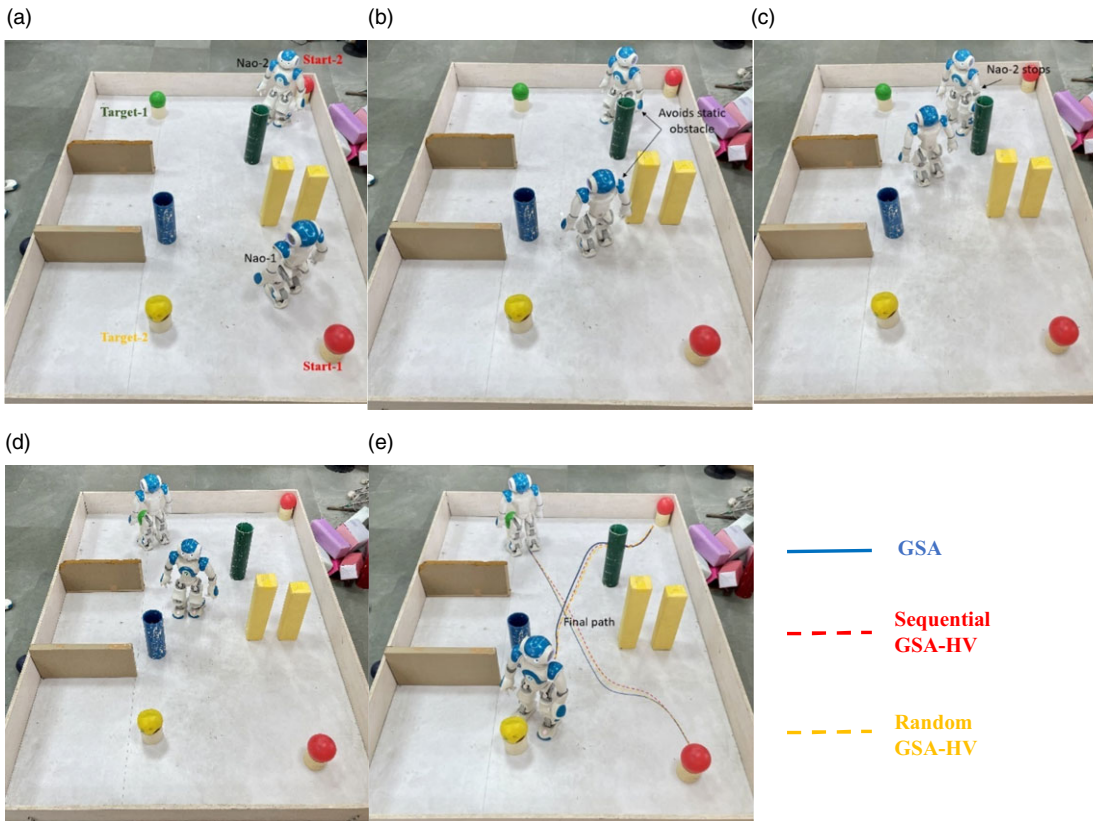


Figure 11. Experimental path traced using multi-robot navigation with dynamic obstacle.

Eqs. (29) and (30), respectively.

$$DM_x = \left( 0.5I_5 - \frac{\sum_{l_1}^{l_n} \kappa_{l_n} - \sum_{l_1}^{l_n} x_n(m_n \ddot{x}_n) - \sum_{l_1}^{l_n} z_n m_n (\ddot{x}_n - g)}{\sum_{l_1}^{l_n} m_n \ddot{x}_n} \right) \tag{29}$$

$$DM_y = \left( 0.5h_{\text{foot}} - \frac{\sum_{l_1}^{l_n} \kappa_{l_n} - \sum_{l_1}^{l_n} y_n m_n (\ddot{x}_n - g) - \sum_{l_1}^{l_n} z_n (m_n \ddot{y}_n)}{\sum_{l_1}^{l_n} m_n (\ddot{x}_n - g)} \right) \tag{30}$$

where  $h_{\text{foot}}$  is height of NAO’s foot,  $l_i$  denoted the  $i$ th link of NAO, and  $(\ddot{x}, \ddot{y}, \ddot{z})$  denotes the acceleration in the  $x, y,$  and  $z$  directions, respectively, and  $g$  is the acceleration due to gravity.

### 5.1. Path planning using a single-robot system

In the current section, the path planning of a single humanoid is carried out in the presence of several static obstacles located at random locations. The humanoid starts its journey from the source (marked by a red ball) and avoids obstacles (see Fig. 7(c)) to reach the target (marked by a green ball). As the humanoid encounters an obstacle in its path, the proposed controller gets activated, and the robot takes the optimal steering angle to avoid the obstacle. The humanoids then advance toward the goal/target. Similar instances can be found in Fig. 8.

A similar environment, as used during experimentation, is recreated for carrying out the simulation-based path planning of the humanoid robot. The results from the simulation and experimental platforms are then tabulated in Table I to understand the variation in the overall path planning strategies.

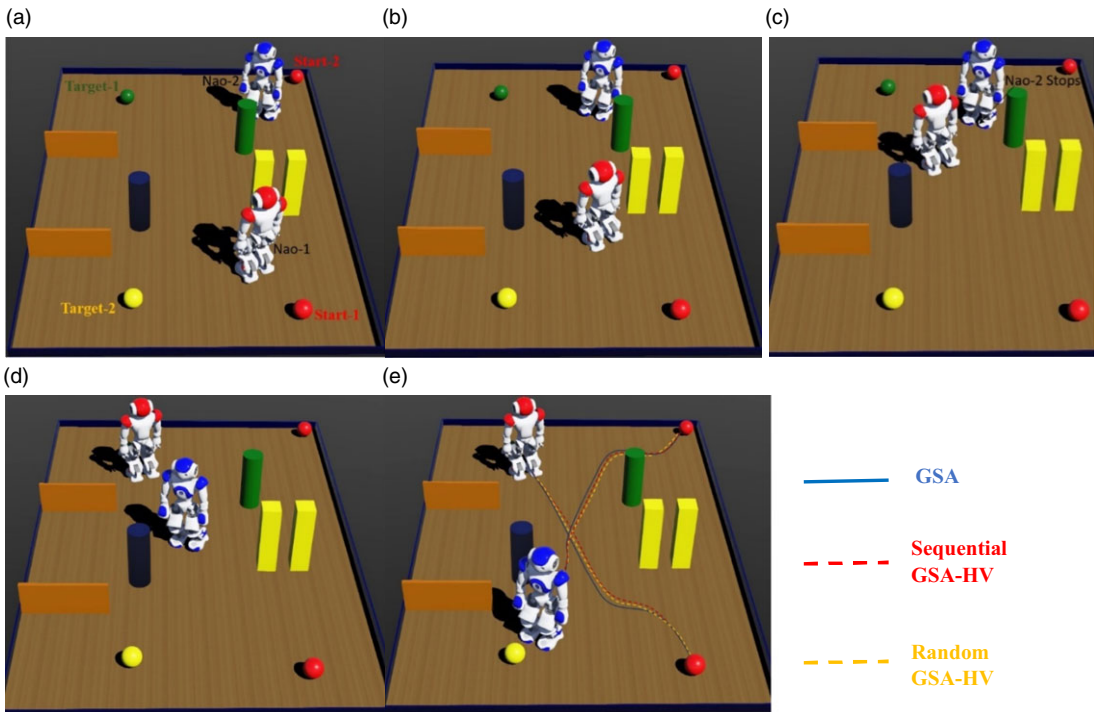


Figure 12. Simulation path traced using multi-robot navigation with dynamic obstacle.

5.2. Path planning using multi-robot system

5.2.1. Multi-robot path planning using static obstacles

Similar to single-robot navigation, multi-robot path planning is carried out using two humanoids. Path planning becomes complex compared to single humanoid navigation, as the system demands proper communication to avoid dynamic collision. However, in the present subsection, the humanoids do not encounter any dynamic barriers and hence do not evoke the Dining Philosopher’s model. The overall path planning is shown in Figs. 9 and 10.

5.2.2. Multi-robot path planning using static and dynamic obstacles

A dynamic obstacle is introduced in the present subsection to complicate the overall path planning. Here, one robot serves as a dynamic obstacle for another and demands proper communication among them to avoid a collision. Similar to Sections 5.1 and 5.2.1, path planning is carried out using randomly located static obstacles. Different paths traced using the individual and proposed controllers are shown in Figs. 11 and 12 and further tabulated.

A similar environment is recreated using the simulation interface, and the experimental results are validated. From Figs. 11(c) and 12(c), it can be seen that NAO-2 stops on the passage (NAO-2 serves as a dynamic obstacle for Nao-1) to give way to NAO-1 using the Petr-Net model.

6. Path planning results of humanoid navigation

The results from the experiment and simulation environments are shown in Tables I, II, and III. It accounts for, both, single and multiple humanoid navigations in Section 5 of the paper. The solid blue line indicates the path obtained using the GSA approach, while the dotted lines indicate the path obtained

**Table II.** Navigation results for multi-robot system with static obstacles.

|                      | Path length (cm) |       |                     |       |                         |       |
|----------------------|------------------|-------|---------------------|-------|-------------------------|-------|
|                      | GSA              |       | Random-based GSA-HV |       | Sequential-based GSA-HV |       |
|                      | Nao-1            | Nao-2 | Nao-1               | Nao-2 | Nao-1                   | Nao-2 |
| <b>Section 5.2.1</b> |                  |       |                     |       |                         |       |
| Experiment           | 305.9            | 322.8 | 282.2               | 298.1 | 286.3                   | 301.8 |
|                      | 305.5            | 322.2 | 282.9               | 299.5 | 286.8                   | 302.3 |
|                      | 306.2            | 323.5 | 281.8               | 298.8 | 287.1                   | 302.1 |
| Simulation           | 293.1            | 307.9 | 268.4               | 284.5 | 274.8                   | 288.2 |
|                      | 293.9            | 308.5 | 269.5               | 284.1 | 274.1                   | 289.6 |
|                      | 294.5            | 309.4 | 269.1               | 284.8 | 273.8                   | 287.8 |
| Avg. Exp. path       | 305.9            | 322.8 | 282.3               | 298.8 | 286.7                   | 302.1 |
| Avg. Sim. path       | 293.8            | 308.6 | 268.7               | 284.5 | 274.2                   | 288.5 |
| % Deviation          | 4.1              | 4.6   | 5.1                 | 5.0   | 4.6                     | 4.7   |
| Avg. Exp. time       | 38.4             | 39.5  | 28.5                | 34.5  | 29.5                    | 35.1  |
| Avg. Sim. time       | 36.6             | 37.8  | 26.9                | 32.8  | 27.9                    | 33.4  |
| % Deviation          | 4.9              | 4.5   | 5.9                 | 5.2   | 5.7                     | 5.1   |

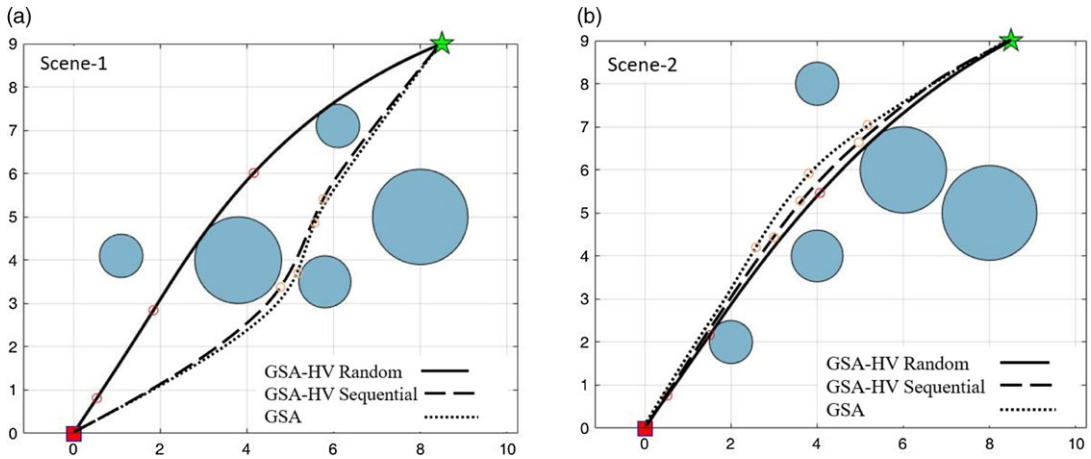
**Table III.** Navigation results for a multi-robot system with static and dynamic obstacles.

|                      | Path length |       |                     |       |                         |       |
|----------------------|-------------|-------|---------------------|-------|-------------------------|-------|
|                      | GSA         |       | Random-based GSA-HV |       | Sequential-based GSA-HV |       |
|                      | Nao-1       | Nao-2 | Nao-1               | Nao-2 | Nao-1                   | Nao-2 |
| <b>Section 5.2.2</b> |             |       |                     |       |                         |       |
| Experiment           | 353.1       | 361.2 | 330.5               | 335.1 | 328.5                   | 339.5 |
|                      | 352.8       | 360.5 | 331.2               | 334.2 | 329.4                   | 339.1 |
|                      | 352.1       | 360.8 | 331.9               | 334.5 | 330.4                   | 340.4 |
| Simulation           | 337.5       | 346.8 | 315.8               | 319.4 | 313.5                   | 323.5 |
|                      | 336.8       | 345.4 | 315.2               | 318.8 | 313.8                   | 323.1 |
|                      | 337.8       | 345.9 | 316.1               | 319.1 | 312.8                   | 322.5 |
| Avg. Exp. path       | 352.7       | 360.8 | 331.2               | 334.6 | 329.4                   | 339.7 |
| Avg. Sim. path       | 337.4       | 346.0 | 315.7               | 319.1 | 313.4                   | 323   |
| % Deviation          | 4.5         | 4.3   | 4.9                 | 4.9   | 5.1                     | 5.2   |
| Avg. Exp. time       | 47.8        | 48.1  | 41.2                | 43.1  | 39.2                    | 44.2  |
| Avg. Sim. time       | 43.7        | 45.6  | 39.1                | 41.4  | 37.5                    | 41.8  |
| % Deviation          | 5.5         | 5.5   | 5.4                 | 4.1   | 4.5                     | 5.7   |

using GSA-HV. Ten runs are carried out for both simulation and experimental platforms to avoid errors in the humanoids' overall path planning. However, only the three best runs are shown in the table.

The results from Table I show that the random-based proposed technique outperformed the other approach. The difference in the results obtained using the random and sequential GSA-HV approach was very small, but a clear distinction can be seen in the results. Similarly, for the path planning carried





**Figure 13.** Comparison of work done using basic GSA vs. proposed technique.

out in Section 5.2.1, it is clear that the random approach was more effective than the other two for the navigation of both Nao-1 and Nao-2. Again, in Section 5.2.2, the sequential-based proposed technique was more effective for Nao-1 locomotion, while random-based GSA-HV was more effective for Nao-2 locomotion. The reason lies in how the approach is applied to the path planning of the humanoids. For a given number of iterations, the random-based search algorithm can capture the optimal global solution, which is impossible for a given iteration for the sequential-based technique. Moreover, the percentage deviation is within the acceptable limits for the different path planning techniques.

## 7. Comparison of the proposed model with other approaches

Different robots possess different kinematics during path planning. The kinematics of wheeled robots differs from that of biped robots. As a result, comparing the two in a similar environment is impossible. So, the effectiveness of the proposed model is tested via the simulation done in the MATLAB interface. The effectiveness in terms of the path length and time can be easily identified using the MATLAB tool. So, in the present section, we have tried to simulate the path planning in MATLAB R2019b software and compared the proposed technique with its counterparts.

### 7.1. Comparison with the basic GSA model

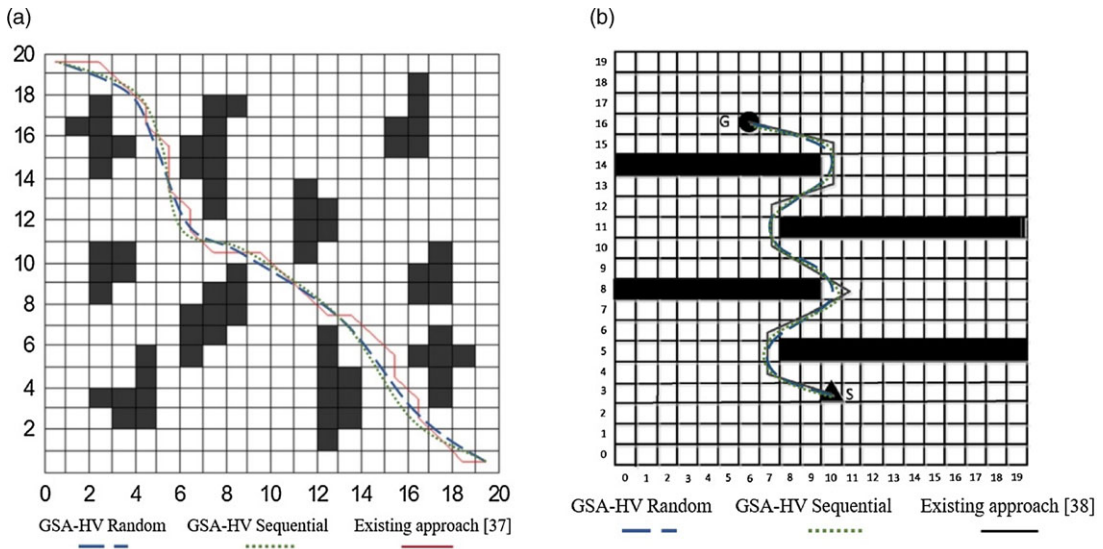
In the present subsection, a simulation-based comparison is made using the MATLAB interface. An arena of dimension  $10 \times 9$  unit<sup>2</sup> is used for obtaining the best possible path from the source (0,0) to the goal (8,2,9). The red square box represents the position of the source, while a green star represents the goal. Figure 13 shows that the path traced by the proposed technique is optimal than that obtained using the basic GSA technique. While the path length obtained by the basic GSA was 15.4 units in scene 1, the path length obtained in scene 2 was 14.1 units. A minimal percentage improvement of more than 7% was achieved using the proposed techniques and tabulated in Table IV.

### 7.2. Comparison with Improved Ant Colony model

In the current section, the proposed technique is compared with the work of Liu et al. [37]. Using the MATLAB tool, the author applied the Improved Ant Colony algorithm (IACA) to obtain the optimal path in an arena of dimension  $20 \times 20$  unit<sup>2</sup>. Further, the work was also tested with the work done by Qu et al. [38] in a similar arena, with parameters of the IACA tuned differently. The environment was

**Table IV.** Comparison of the proposed approach with basic GSA.

| Techniques used   | Path length (units) |         | Percentage improvement w.r.t. GSA |         |
|-------------------|---------------------|---------|-----------------------------------|---------|
|                   | Scene 1             | Scene 2 | Scene 1                           | Scene 2 |
| GSA-HV Sequential | 14.3                | 13.1    | 7.14 %                            | 7.1 %   |
| GSA-HV Random     | 13.2                | 12.5    | 14.29 %                           | 11.35 % |



**Figure 14.** Comparison with work done by (a) Liu et al. [37] and (b) Qu et al. [38].

reproduced, and the proposed approach was applied. Figure 14 shows that the locus of the path traced by the proposed technique is optimal compared to the existing approach.

A percentage improvement of more than 13 % is achieved using the proposed plan. The results are tabulated in Tables V and VI.

### 7.3. Comparison with the vision-based approach

The results are further compared with the work done using a vision-based approach [39] using a linguistic method for walking NAO on uneven terrain. The vision-based approach entirely depends upon the color identification scheme of the humanoid NAO to avoid the obstacle. From Fig. 15, it can be seen that the vision-based approach took a long time to reach the goal compared to the sensor-based proposed technique.

## 8. Discussion

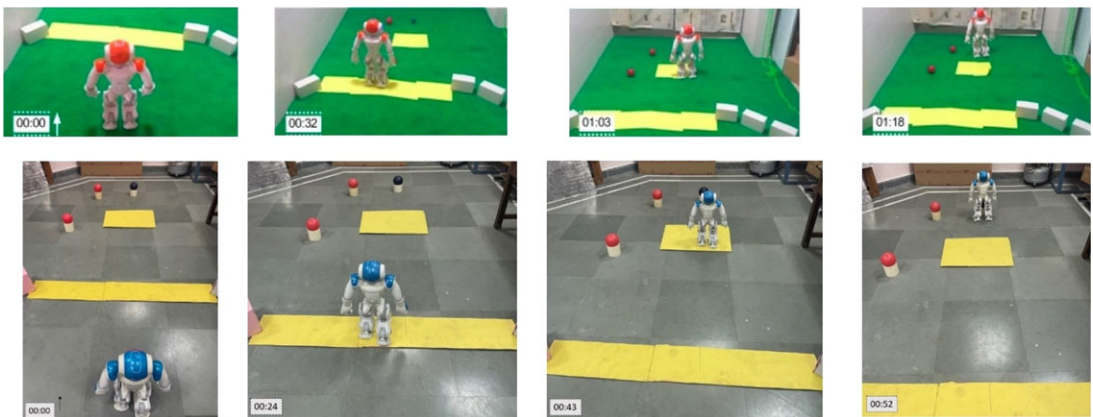
The performance of the proposed approach is demonstrated using single and multiple humanoid navigations with static and dynamic obstacles. The proposed approach is found to be robust when compared with the basic GSA technique. The basic GSA tends to get trapped in optimal local conditions, resulting in the optimal global solution being compromised. Using the HV approach to the drifted mass helps reduce the overall computational time and ensures that an optimal path is achieved. The navigational

**Table V.** Comparison of the proposed technique with the work done by Liu et al. [37].

| Techniques used   | Path length (units) | Maximum percentage improvement |
|-------------------|---------------------|--------------------------------|
| Liu et al. [37]   | 30.38               | 16.66 %                        |
| GSA-HV Sequential | 26.48               |                                |
| GSA-HV Random     | 25.32               |                                |

**Table VI.** Comparison of the proposed technique with the work done by Qu et al. [38].

| Techniques used   | Path length (units) | Maximum percentage improvement |
|-------------------|---------------------|--------------------------------|
| Qu et al. [38]    | 22.72               | 13.64 %                        |
| GSA-HV Sequential | 20.51               |                                |
| GSA-HV Random     | 19.62               |                                |



**Figure 15.** Path traced using vision-based approach [39] vs. proposed technique.

parameters are calculated for individual and proposed controllers. A percent improvement of 5.9% and 6.4% is seen in the overall path length for single-robot navigation in experimental and simulation environments. In the same run, the overall improvement in time was found to be 18.2% and 17.4% for experimental and simulation-based navigation, respectively.

For multi-robot navigation without any dynamic obstacle, it was found that the percent improvement in experimental path length and time for both the humanoids was more than 8% and 14%, respectively. Similar improvements were seen for multi-robot navigation with dynamic obstacles. Figures 16 and 17 show the path length and time graph for navigation in Section 5 of the paper. To obtain the stability of the Humanoid while avoiding the different obstacles, the ZMP obtained using the different individual and proposed approach was compared with the ideal ZMP. While the average deviation using the proposed model showed a maximum deviation of 5.8% when compared with the ideal ZMP, the deviation was around 11.4% while using the GSA technique. Thus, the proposed approach was found to be more stable while avoiding the obstacle in the journey.

The effectiveness of the proposed technique was further compared using the MATLAB simulation software, where a percent improvement of more than 7% was observed compared to the basic GSA technique. Further comparison with the existing approach showed a minimal 13% improvement in the MATLAB-based simulation result. The effectiveness of the model was further compared with the vision-based approach, and it was found that the proposed model was able to show more than 30% improvement in the overall path planning of the humanoid.

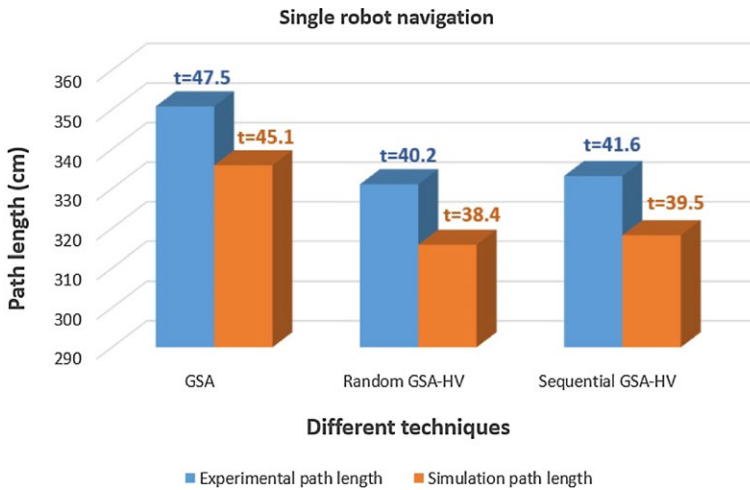


Figure 16. Single-robot navigation results in a static obstacle-prone environment.

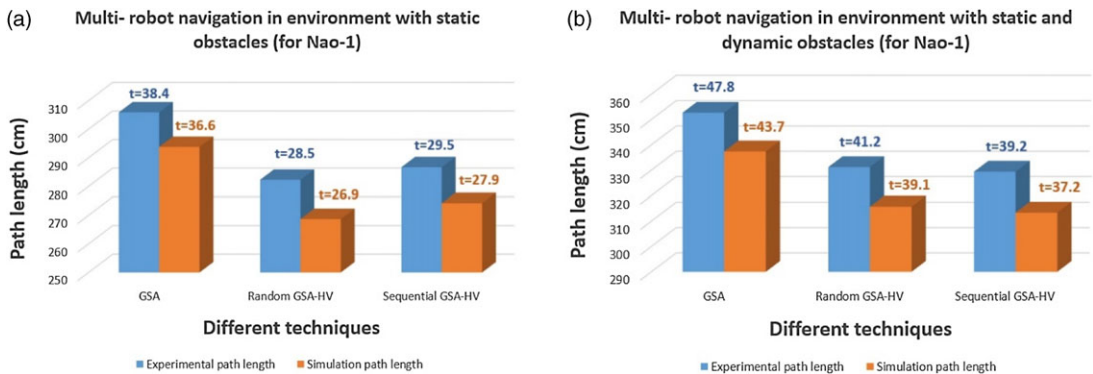


Figure 17. Multi-robot navigation results in an environment with (a) only static obstacles and (b) static and dynamic obstacles.

### 9. Conclusion

Path planning is a complex task that needs proper tuning of the different parameters. The work is significant, as it can replace human efforts in the coming time. The modification to the basic structure of GSA has helped improve the overall path-based navigation in different complex terrains. From the work done in the current paper, the following conclusions can be drawn:

- The proposed technique successfully avoids local traps and obtains optimal global solutions.
- The proposed model was tested in both simulation and experimental platforms and was found to be robust.
- The efficiency of the proposed controller, when compared with the work done by authors in the past, produced optimal results.
- Implementing the proposed technique helped reduce the overall computational time, thereby optimizing the overall navigation time.

The proposed technique, in the future, can be applied to newer areas related to navigation on uneven terrains with different floor inclinations or walking on stairs, etc. The approach can be further implemented along with vision-based navigation to improve overall path planning and coordination.

**Acknowledgments.** This research received no specific grant from any funding agency, commercial or not-for-profit sectors.

**Author's contribution.** **Vikas:** Methodology, Conceptualization, original draft preparation, Writing. **D. R. Parhi:** Formal analysis and investigation, review and editing

**Competing interests.** The authors have no relevant financial or non-financial interests to disclose.

## References

- [1] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "GSA: A gravitational search," *Inf. Sci.* **179**(13), 2232–2248 (2009). doi: [10.1016/j.ins.2009.03.004](https://doi.org/10.1016/j.ins.2009.03.004).
- [2] C. Purcaru, R. E. Precup, D. Iercan, L. O. Fedorovici, R. C. David and F. Dragan, "Optimal robot path planning using gravitational search algorithm," *Int. J. Artif. Intell.* **10**(13), 1–20 (2013).
- [3] G. Sahoo, "A review on gravitational search algorithm and its applications to data clustering & classification," *Int. J. Intell. Syst. Appl.* **6**(6), 79–93 (2014). doi: [10.5815/ijisa.2014.06.09](https://doi.org/10.5815/ijisa.2014.06.09).
- [4] K. Gia, H. Mokhlis, H. A. Illias, M. M. Aman and J. J. Jamian, "Gravitational search algorithm and selection approach for optimal distribution network configuration based on daily photovoltaic and loading variation," *J. Appl. Math.* **2015**, 1–11 (2015). doi: [10.1155/2015/894758](https://doi.org/10.1155/2015/894758).
- [5] A. Yadav and K. Deep, "Constrained optimization using gravitational search algorithm," *Natl. Acad. Sci. Lett.* **36**(5), 527–534 (2013). doi: [10.1007/s40009-013-0165-8](https://doi.org/10.1007/s40009-013-0165-8).
- [6] B. Zibanezhad, K. Zamanifar, R. S. Sadjady and Y. Rastegari, "Applying gravitational search algorithm in the QoS-based Web service selection problem," *J. Zhejiang Univ. Sci. C* **12**(9), 730–742 (2011). doi: [10.1631/jzus.C1000305](https://doi.org/10.1631/jzus.C1000305).
- [7] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "Filter modeling using gravitational search algorithm," *Eng. Appl. Artif. Intell.* **24**(1), 117–122 (2011). doi: [10.1016/j.engappai.2010.05.007](https://doi.org/10.1016/j.engappai.2010.05.007).
- [8] A. Ramamoorthy and R. Ramachandran, "Reactive Power Optimization Using GSA," *In: 2014 6th IEEE Power India International Conference (PIICON)* (IEEE, 2014) pp. 1–4. doi: [10.1109/POWERI.2014.7117680](https://doi.org/10.1109/POWERI.2014.7117680).
- [9] D. R. Parhi, "Chaos-based optimal path planning of humanoid robot using hybridized regression-gravity search algorithm in static and dynamic terrains," *Appl. Soft Comput.* **140**, 110236 (2023). doi: [10.1016/j.asoc.2023.110236](https://doi.org/10.1016/j.asoc.2023.110236).
- [10] P. B. Kumar and D. R. Parhi, "Intelligent hybridization of regression technique with genetic algorithm for navigation of humanoids in complex environments," *Robotica* **38**(4), 565–581 (2020). doi: [10.1017/S0263574719000869](https://doi.org/10.1017/S0263574719000869).
- [11] D. R. Parhi and A. K. Kashyap, "Humanoid robot path planning using memory-based gravity search algorithm and enhanced differential evolution approach in a complex environment," *Expert Syst. Appl.* **215**, 119423 (2023). doi: [10.1016/j.eswa.2022.119423](https://doi.org/10.1016/j.eswa.2022.119423).
- [12] G. Sun and A. Zhang, "A Hybrid Genetic Algorithm and Gravitational Search Algorithm for Image Segmentation Using Multilevel Thresholding," *In: Iberian Conference on Pattern Recognition and Image* (Springer, Berlin/Heidelberg, 2013) pp. 707–714. doi: [10.1007/978-3-642-38628-2\\_84](https://doi.org/10.1007/978-3-642-38628-2_84).
- [13] S. Mirjalili, G. G. Wang and L. D. S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Comput. Appl.* **25**(6), 1423–1435 (2014). doi: [10.1007/s00521-014-1629-6](https://doi.org/10.1007/s00521-014-1629-6).
- [14] S. Mallick, S. P. Ghoshal, P. Acharjee and S. S. Thakur, "Optimal static state estimation using improved particle swarm optimization and gravitational search algorithm," *Int. J. Electr. Power Energy Syst.* **52**, 254–265 (2013). doi: [10.1016/j.ijepes.2013.03.035](https://doi.org/10.1016/j.ijepes.2013.03.035).
- [15] B. Gonzalez, F. Valdez, P. Melin and G. Prado-Arechiga, "Fuzzy logic in the gravitational search algorithm for the optimization of modular neural networks in pattern recognition," *Expert Syst. Appl.* **42**(14), 5839–5847 (2015). doi: [10.1016/j.eswa.2015.03.034](https://doi.org/10.1016/j.eswa.2015.03.034).
- [16] D. Ezzat and H. A. Ella, "GSA-DenseNet121-COVID-19: A hybrid deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization algorithm," *arXiv preprint arXiv:2004.05084* (2020). doi: [10.48550/arXiv.2004.05084](https://doi.org/10.48550/arXiv.2004.05084).
- [17] P. Haghbayan, H. Nezamabadi-Pour and S. Kamyab, "A niche GSA method with nearest neighbor scheme for multimodal optimization," *Swarm Evol. Comput.* **35**, 78–92 (2017).
- [18] D. Shen, T. Jiang, W. Chen, Q. Shi and S. Gao, "Improved Chaotic Gravitational Search Algorithms for Global Optimization," *In: 2015 IEEE Congress on Evolutionary Computation (CEC)* (IEEE, 2015) pp. 1220–1226. doi: [10.1109/CEC.2015.7257028](https://doi.org/10.1109/CEC.2015.7257028).
- [19] R. K. Sahu, S. Panda and S. Padhan, "Optimal gravitational search algorithm for automatic generation control of interconnected power systems. Ain shams engineering," *Ain Shams Eng. J.* **5**(3), 721–733 (2014). doi: [10.1016/j.asej.2014.02.004](https://doi.org/10.1016/j.asej.2014.02.004).
- [20] T. Eldos and R. Al Qasim, "On the performance of the gravitational search algorithm," *Int. J. Adv. Comput. Sci. Appl.* **4**(8), 74–78 (2013). doi: [10.14569/IJACSA.2013.040811](https://doi.org/10.14569/IJACSA.2013.040811).
- [21] Y. Wang, S. Gao, Y. Yu, Z. Wang, J. Cheng and T. Yuki, "A gravitational search algorithm with chaotic neural oscillators," *IEEE Access* **8**, 25938–25948 (2020). doi: [10.1109/ACCESS.2020.2971505](https://doi.org/10.1109/ACCESS.2020.2971505).
- [22] H. Mittal, R. Pal, A. Kulhari and M. Saraswat, "Chaotic Kbest Gravitational Search Algorithm (CKGSA)," *In: 2016 Ninth International Conference on Contemporary Computing (IC3)* (IEEE, 2016) pp. 1–6. doi: [10.1109/IC3.2016.7880252](https://doi.org/10.1109/IC3.2016.7880252),

- [23] T. S. Sieb, A. Singh, L. Guidos and H. A. Hashim, “FLC Tuned with Gravitational Search Algorithm for Nonlinear Pose Filter,” **In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)** (2020) pp. 2864–2869. doi: [10.1109/SMC42975.2020.9283387](https://doi.org/10.1109/SMC42975.2020.9283387).
- [24] D. Pelusi, R. Mascella and L. Tallini, “A fuzzy gravitational search algorithm to design optimal IIR filters,” *Energies* **11**(4), 736 (2018). doi: [10.3390/en11040736](https://doi.org/10.3390/en11040736).
- [25] K. Kang, C. Bae, H. W. F. Yeung and Y. Y. Chung, “A hybrid gravitational search algorithm with swarm intelligence and deep convolutional feature for object tracking optimization,” *Appl. Soft Comput.* **66**, 319–329 (2018). doi: [10.1016/j.asoc.2018.02.037](https://doi.org/10.1016/j.asoc.2018.02.037).
- [26] R. K. Sahu, S. Panda and S. Padhan, “A novel hybrid gravitational search and pattern search algorithm for load frequency control of non-linear power system,” *Appl. Soft Comput.* **29**, 310–327 (2015). doi: [10.1016/j.asoc.2015.01.020](https://doi.org/10.1016/j.asoc.2015.01.020).
- [27] J. Wang, D. Liu and H. Shang, “Hill Valley Function Based Niching Particle Swarm Optimization for Multi-modal Functions,” **In: 2009 International Conference on Artificial Intelligence and Computational Intelligence**, vol. 1 (IEEE, 2009) pp. 139–144. doi: [10.1109/AICI.2009.250](https://doi.org/10.1109/AICI.2009.250).
- [28] S. C. Maree, T. Alderliesten and P. A. Bosman, “Real-Valued Evolutionary Multi-modal Multi-objective Optimization by Hill-Valley Clustering,” **In: Proceedings of the Genetic and Evolutionary Computation Conference** (2019) pp. 568–576. doi: [10.48550/arXiv.2010.14998](https://doi.org/10.48550/arXiv.2010.14998).
- [29] M. M. Ellabaan and Y. S. Ong, “Valley-Adaptive Clearing Scheme for Multi-modal Optimization Evolutionary Search,” **In: 2009 Ninth International Conference on Intelligent Systems Design and Applications** (IEEE, 2009) pp. 1–6. doi: [10.1109/ISDA.2009.115](https://doi.org/10.1109/ISDA.2009.115).
- [30] J. Zhang, D. S. Huang, T. M. Lok and M. R. Lyu, “A novel adaptive sequential niche technique for multi-modal function optimization,” *Neurocomputing* **69**(16–18), 2396–2401 (2006). doi: [10.1016/j.neucom.2006.02.016](https://doi.org/10.1016/j.neucom.2006.02.016).
- [31] SoftBank Robotics, NAO the Humanoid and Programmable Robot (n.d.). Retrieved May 26, 2022, from <https://www.softbankrobotics.com/emea/en/nao>
- [32] P. K. Das, H. S. Behera, S. Das, H. K. Tripathy, B. K. Panigrahi and S. K. Pradhan, “A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment,” *Neurocomputing* **207**, 735–753 (2016). doi: [10.1016/J.NEUCOM.2016.05.057](https://doi.org/10.1016/J.NEUCOM.2016.05.057).
- [33] S. Gao, C. Vairappan, Y. Wang, Q. Cao and Z. Tang, “Gravitational search algorithm combined with chaos for unconstrained numerical optimization,” *Appl. Math. Comput.* **231**, 48–62 (2014). doi: [10.1016/j.amc.2013.12.175](https://doi.org/10.1016/j.amc.2013.12.175).
- [34] R. K. Ursem, “Multinational Evolutionary Algorithms,” **In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**, vol. 3 (1999) pp. 1633–1640. doi: [10.1109/CEC.1999.785470](https://doi.org/10.1109/CEC.1999.785470).
- [35] E. W. Dijkstra, “Hierarchical Ordering of Sequential Processes,” **In: The Origin of Concurrent Programming** (Springer, New York, 1971) pp. 198–227. doi: [10.1007/BF00289519](https://doi.org/10.1007/BF00289519).
- [36] A. K. Kashyap and D. R. Parhi, “Dynamic walking of multi-humanoid robots using BFGS Quasi-Newton method aided artificial potential field approach for uneven terrain,” *Soft Comput.* **27**(9), 5893–5910 (2023). doi: [10.1007/s00500-022-07606-7](https://doi.org/10.1007/s00500-022-07606-7).
- [37] J. Liu, J. Yang, H. Liu, X. Tian and M. Gao, “An improved ant colony algorithm for robot path planning,” *Soft Comput.* **21**(19), 5829–5839 (2017). doi: [10.1007/s00500-016-2161-7](https://doi.org/10.1007/s00500-016-2161-7).
- [38] H. Qu, K. Xing and T. Alexander, “An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots,” *Neurocomputing* **120**, 509–517 (2013). doi: [10.1016/j.neucom.2013.04.020](https://doi.org/10.1016/j.neucom.2013.04.020).
- [39] H. Jahanshahi, M. Jafarzadeh, N. N. Sari, V. T. Pham, V. V. Huynh and X. Q. Nguyen, “Robot motion planning in an unknown environment with danger space,” *Electronics* **8**(2), 201 (2019). doi: [10.3390/electronics8020201](https://doi.org/10.3390/electronics8020201).