

ACO-Kinematic: a hybrid first off the starting block

Kaylash Chaudhary, Avinesh Prasad, Vishal Chand and Bibhya Sharma

The University of the South Pacific, Suva, Fiji

ABSTRACT

The use of robots in carrying out various tasks is popular in many industries. In order to carry out a task, a robot has to move from one location to another using shorter, safer and smoother route. For movement, a robot has to know its destination, its previous location, a plan on the path it should take, a method for moving to the new location and a good understanding of its environment. Ultimately, the movement of the robot depends on motion planning and control algorithm. This paper considers a novel solution to the robot navigation problem by proposing a new hybrid algorithm. The hybrid algorithm is designed by combining the ant colony optimization algorithm and kinematic equations of the robot. The planning phase in the algorithm will find a route to the next step which is collision free and the control phase will move the robot to this new step. Ant colony optimization is used to plan a step for a robot and kinematic equations to control and move the robot to a location. By planning and controlling different steps, the hybrid algorithm will enable a robot to reach its destination. The proposed algorithm will be applied to multiple point-mass robot navigation in a multiple obstacle and line segment cluttered environment. In this paper, we are considering a priori known environments with static obstacles. The proposed motion planning and control algorithm is applied to the tractor-trailer robotic system. The results show a collision and obstacle free navigation to the target. This paper also measures the performance of the proposed algorithm using path length and convergence time, comparing it to a classical motion planning and control algorithm, Lyapunov based control scheme (LbCS). The results show that the proposed algorithm performs significantly better than LbCS including the avoidance of local minima.

Submitted 9 September 2021

Accepted 8 February 2022

Published 15 March 2022

Corresponding author

Kaylash Chaudhary,
kaylash.chaudhary@usp.ac.fj

Academic editor

Jyotismita Chaki

Additional Information and
Declarations can be found on
page 24

DOI 10.7717/peerj-cs.905

© Copyright
2022 Chaudhary et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

Subjects Algorithms and Analysis of Algorithms, Artificial Intelligence, Robotics

Keywords Robots, Path, Planning, Optimization, Kinematic

INTRODUCTION

Over the years, the use of robots has evolved due to improved capacities and abilities to carry out complex and diverse activities in areas such as manufacturing, logistics, home, travel, health, mining, civil, military, and transportation (*Dunbabin & Marques, 2012; Khurshid & Bing-rong, 2004; Murphy et al., 2009; Napper & Seaman, 1989; Raj et al., 2018*). In almost all cases, a robot must travel from one point to another in order to complete a task. A robot should also avoid collisions and dangerous situations while navigating through its surroundings in order to reach a certain point. This is generally known as findpath or robot navigation problem which invariably has four categories: localization, path planning, motion control, and cognitive mapping (*Buniyamin et al., 2011*). This paper

focuses on the motion planning and control problem. Due to its usefulness in real-world applications, robot motion planning and control has been a widely researched topic over the past four decades. The primary goal of robot path planning and control is to identify the most efficient and safe route from point A to point B and subsequently control the robot to point B. There are two subtasks in robot motion planning and control: (1) plan a path which should be obstacle and collision free, and (2) control the robot to its destination.

Various methods are available in the literature for path planning which can be categorized into classical, heuristic and machine and deep learning. Artificial potential field ([Lee & Park, 2003](#)), cell decomposition ([Kloetzer, Mahulea & Gonzalez, 2015](#)), road map ([Lingelbach, 2004](#)), and virtual force field ([Bortoff, 2000](#)) are examples of classical or traditional approach. Optimization algorithms such as firefly ([Patle et al., 2018](#)), ant-colony ([Brand et al., 2010](#)), and particle swarm ([Wang & Zhou, 2019](#)) are examples of heuristic approaches. Algorithms such as neural networks, decision trees, Naive Bayes, and others are used in machine and deep learning techniques for robot path planning ([Otte, 2015](#)). This paper will focus on a heuristic approach, in particular, the ant colony optimization. An Ant colony optimization (ACO) is a popular algorithm that has been applied to different problems, including path planning problems in robotics ([Brand et al., 2010](#); [Reshamwala & Vinchurkar, 2013](#)). In this paper, given the nature of the problem, the authors use the ACO algorithm developed by Socha and Dorigo ([Socha & Dorigo, 2008a](#); [Socha & Dorigo, 2008b](#)) for continuous domain (ACOR); however, there are different variants of ACO algorithms such as fuzzy heuristic based ACO ([Shokouhifar, 2021](#)), continuous ACO (CACO) ([Bilchev & Parmee, 1995](#)) and continuous interacting ant colony (CIAC) ([Dréo & Siarry, 2004](#)) which have been utilized for different problems and situations. The authors chose the variant of ACO proposed by Socha and Dorigo because it clearly outperformed other continuous ACO variants like CACO, API and CIAC in eight test functions ([Socha & Dorigo, 2008a](#); [Socha & Dorigo, 2008b](#)) and it follows the original ACO formulation. Furthermore, Socha and Dorigo also compared ACOR with continuous genetic algorithm (CGA), enhanced continuous tabu search (ECTS), enhanced simulated annealing (ESA) and differential evolution (DE), where ACOR performed well in one third of the test problems and performed not much worse on other problems ([Socha & Dorigo, 2008a](#); [Socha & Dorigo, 2008b](#)).

Once a path is planned the robot can track or follow the planned path to reach its destination. There are different methods in the literature for robot motion control such as kinematics, dynamics, artificial potential field, fuzzy logic and many more ([Martinez-Alfaro & Gomez-Garcia, 1998](#); [Tanaka et al., 1998](#); [Tsai & Wang, 2005](#); [Watanabe et al., 1998](#)). This paper will focus on kinematic equations of the robots for motion control.

Majority of the studies in the literature carry out motion control in two different phases: (1) path planning, and (2) subsequent robot tracking or control. This paper will address the problem of the motion planning and control differently. A robot's next step will be planned and the robot will move to that step. At the time of the robot's step planning, the obstacles and other robots will be considered and a collision free step will be determined. By repeating this process, the robot will reach its destination. Therefore, there is path planning and control at every step to its destination.

This research will introduce a new hybrid algorithm that is a strategic combination of an ant colony optimization algorithm and kinematic equations of robot motion. The purpose of the algorithm is to plan a step (location) for the robot and the robot will use its kinematic equations to move to that step (location). This process is repeated until the final destination is reached. This is the main difference when compared to other algorithms and hybrids as most plan the entire path first and then the robot starts the journey. Selected scenarios will be used to showcase the new hybrid and a performance comparison in terms of path length and convergence time will be made with the Lyapunov based Control Scheme (LbCS), a classical approach for motion control.

The main contributions of this paper are as follows:

1. ACO-Kinematic algorithm: A new hybrid algorithm is proposed which plans a step for the robot and the robot moves to that step and this process continues until the robot reaches its destination. In literature, according to authors knowledge, there is no hybrid algorithm that plans next step location of robot and controls it to that step location. The proposed algorithm also solves the problem of local minima.
2. Multi-objective problem formulation: A new multi-objective problem is formulated in terms of path length and safety of the path. The safety objective is achieved through ant colony optimization and the path length objective is achieved by both, ant colony optimization and kinematic equations.
3. Application: The methodology derived for point-mass robots has been successfully applied to tractor-trailer robotic system to show the effectiveness of the proposed algorithm in real life applications.
4. Analysis: The performance of the new hybrid algorithm has been compared to that of lyapunov based control scheme (LbCS) in terms of path length and convergence time. Such a comparison using the LbCS has been carried out for the first time. The analysis show that ACO-Kinematic performed slightly better than LbCS including the avoidance of local minima.

‘Related Work’ presents the literature review on motion planning and control algorithms. The problem statement is discussed in ‘Problem Statement and Objectives’. ‘Ant Colony Optimization’ discusses the ant colony optimization algorithm. The three objectives of motion planning namely, short, safe and smooth path are formulated and discussed in ‘Robot Motion Planning and Control Formulation’. The proposed algorithm is presented in ‘Proposed Algorithm’. The three case studies and example scenarios, including the kinematic equations for the tractor-trailer robot are discussed in ‘Results’. ‘Discussion’ presents three different scenarios to measure performance of the proposed algorithm and the LbCS. Finally, the paper concludes in ‘Conclusion and Future Work’, discussing its contributions and recommendations for future work.

RELATED WORK

In literature, there are mostly instances of path planning and motion control being researched and implemented separately. There are a few algorithms that consider motion planning and control in parallel or simultaneously. The section will outline these algorithms and present a brief comparison with the proposed algorithm.

Firstly, there are algorithms that carry out path planning and then control the robots on these paths, usually known as path-tracking. For example, Saputro et al. implemented trajectory planning and tracking of mobile robots using a map and predictive control (Saputro, Rusmin & Rochman, 2018). A map of the environment is first created and then the collision free path is searched using A* algorithm. Then a model predictive control is used to track the robot on the reference path. Likewise, Ning et al. implemented a trajectory planning and tracking control scheme for autonomous obstacle avoidance of wheeled inverted pendulum (WIP) vehicles (Ning et al., 2020). Motion planning and trajectory control has been applied to car parking as well (Zips, Bck & Kugi, 2013), where the authors first find a path by solving a static optimization problem and then use a optimal controller for parking. The reader is referred to Chen, Peng & Grizzle (2017), Guo et al. (2018), Viana et al. (2021), Wang et al. (2020) and Zhang et al. (2019) for other examples of such algorithms.

Secondly, there are algorithms that consider path planning and motion control in parallel. Wahid et al. (2017) used artificial potential field method for motion planning and kinematics for the control to design vehicle collision avoidance assistance systems. Lyapunov based control scheme (LbCS) was developed by Sharma, Vanualailai & Singh (2014) and Sharma et al. (2008) and has been used by many researchers in different scenarios for robot motion planning and control (Devi et al., 2017; Kumar et al., 2021; Prasad, Sharma & Vanualailai, 2017; Prasad et al., 2021; Raj et al., 2021; Raj et al., 2018). LbCS is a time-invariant nonlinear method that is used to create velocity or acceleration-based controls enabling a robot to move safely in the workspace while avoiding obstacles. Raj et al. used LbCS to control a system of 1-trailer robots in a cluttered environment including a swarm of boids (Raj et al., 2021) and navigate car-like robots in 3-dimensional space (Raj et al., 2018). Prasad et al. used LbCS to derive the acceleration-based controllers for the mobile manipulator in 3-dimensional (Prasad et al., 2021) and to motion control a pair of cylindrical manipulators in a constrained 3-dimensional workspace (Prasad, Sharma & Vanualailai, 2017). Researchers also used LbCS for controlling quadrotors in different environments (Raj, Raghunwaiya & Vanualailai, 2020; Raj, Raghunwaiya & Vanualailai, 2020; Vanualailai, Raj & Raghunwaiya, 2021).

All of the motion planning and control algorithms outlined in the previous paragraph except LbCS first plan the path for the entire journey and then control the motion of the robot on the planned path. However, the proposed algorithm is different from these algorithms as it plans one step at a time and controls the motion of the robot to that step. LbCS also plans a step and then controls the motion of the robot to that step. Therefore, the proposed algorithm will be compared for performance with the LbCS.

PROBLEM STATEMENT AND OBJECTIVES

Suppose there is an initial position and a target position for each robot in a bounded workspace with several static obstacles of different shapes and sizes. The mobile robots will start at the initial position, avoid obstacles in path and reach their designated target position. Therefore, the main objective of the mobile robots is to reach their target by taking

a shorter and safer route (avoid collision with obstacles and other robots). The following assumptions are made to achieve the objective of motion control in this paper:

Assumption 1: The obstacles are of circular and rectangular shapes. In some cases, it is also represented as line segments. These obstacles are of different sizes and randomly distributed in the bounded workspace. The obstacles are static with known locations.

Assumption 2: The kinematic equations are used for the motion of point-mass and tractor-trailer robots. The robots can move in any direction.

Since this paper introduces first of a kind hybrid, more features such as handling uncertainties in the environment and locations of the static obstacles and the effects of noise will be added in the future work.

The research objectives of this paper are as follows:

- Design and implement a hybrid algorithm composed of a heuristic and a classical method for planning a step location of robot and controlling it to that step, and hence reach a target with a series of such steps.
- Apply the hybrid algorithm to a real life application such as tractor-trailer robots.

ANT COLONY OPTIMIZATION

Ant colony optimization (ACO) was initially developed by Dorigo et al. (*Socha & Dorigo, 2008a; Socha & Dorigo, 2008b*) in the early 90's for combinatorial optimization but later modified for continuous domains (*Dréo & Siarry, 2004; Socha & Dorigo, 2008a; Socha & Dorigo, 2008b*). ACO has been inspired from natural ants, which move out of their nest in search of food and move randomly in the surrounding area. Once an ant finds food, it evaluates and carries it on its back. On the way back to the nest, the ant deposits a pheromone trail on the ground which depends on the amount of food and quality. This pheromone trail guides other ants to the food source (*Socha & Dorigo, 2008a; Socha & Dorigo, 2008b*). In this paper, we have used the ant colony optimization for continuous domain (ACOR) proposed by Socha and Dorigo (*Socha & Dorigo, 2008a; Socha & Dorigo, 2008b*). The original ACO is based on discrete domain where pheromone and heuristic information are used to make different probabilistic choices. The main idea of ACOR is shifting from a discrete probability distribution to a continuous one that is, a probability density function (PDF). In ACOR, an ant samples a PDF instead of choosing a solution component when compared to ACO in discrete domain. ACOR closely follows the metaheuristic of ACO in discrete domain.

ACO algorithm for continuous domain has two types of population: archive and new. The pheromone information is stored as a solution archive. The solutions in the archive are ordered according to their quality (determined through objective function evaluations). Each solution has an associated weight proportion to the solution quality. ACOR uses pheromone information to make the probabilistic choice. In the case of robot motion planning and control, the next step for the robot will be determined by the fittest ant (pheromone information) and the robot will move to that step using kinematic equations.

The algorithm is divided into the following phases:

Initialization phase

The algorithm starts with the creation of archive solution during the initialization of ants. ACOR utilizes Gaussian kernel which has three vectors of parameters: weights, means and standard deviations. The time complexity for this phase is $O(n)$, where n is the number of ants.

Archive population phase

For the archive solution, the means and standard deviations are calculated using Eqs. (1) and (2):

$$\mu = \mu_1, \dots, \mu_n = s_1, \dots, s_n, \quad (1)$$

where μ_i are the means and s_i the archive solutions for $i = 1 \dots n$, and

$$sd_i = \xi \sum_{n=1}^k \frac{|s_n - s_l|}{k-1}, \quad (2)$$

where sd_i is the standard deviation, k is the archive population, ξ is the convergence speed of the algorithm and s_l is the chosen solution. ACOR stores in archive solution the values of the solutions' n variables and the value of their objective functions. This phase has a time complexity of $O(n)$, where n is the number of ants.

New population creation phase

This phase involves creation of the new population array which is subsequently randomly initialized. This phase has a time complexity of $O(1)$.

Solution construction phase

This phase starts which consists of the Gaussian kernel selection and then generating Gaussian random variable. Gaussian kernel selection is based on roulette wheel selection and probability. The probability is computed based on weights. The weights and probability are given in Eqs. (3) and (4):

$$w_i = \frac{1}{qk\sqrt{2*\Pi}} \exp\left(-0.5 \frac{(i-1)^2}{(qk)^2}\right), \quad (3)$$

where, w_i is the weight for individual solution, q is the intensification factor and k is the archive population, and

$$P_i = \frac{w_n}{\sum_{n=1}^k w_n}, \quad (4)$$

where, P_i is the probability of individual archive solution, k is the archive population and w_n is the weight of individual archive solution.

An ant chooses probabilistically one of the solutions in the archive using Eq. (4).

The position of the new population is the random variable generated using means and standard deviations from the Archive Population Phase and normally distributed random numbers. This phase has a time complexity of $O(n)$, where n is the number of ants.

Evaluation phase

The new constructed solution is then evaluated using a fitness function and merged with the archive population. Finally, the total population is sorted to get the best solution and a new set of archive solution. The time complexity for this phase is $O(1)$.

All phases except the Initialization phase will be repeated until the robot reaches its destination.

ROBOT MOTION PLANNING AND CONTROL FORMULATION

In this research, the path planning problem of robots is solved by the ACO while their motion controls by the kinematic equations. For the path planning problem, initial artificial ants and objective functions are used. For motion control, the kinematic equations will be used to control a robot to a point (step location) generated by ants. Kinematic equations which are essentially ODE's governing the motion of robot are used to control the motion of the robot. These equations are dependent on the type of the robot. For example, a point mass robot will have a set of different kinematic equations when compared that of a tractor-trailer robot because the latter may include nonholonomic constraints. This section is further divided into ants representation, multi-objective path planning problem, and the problem formulation of path planning and motion control of robots.

Multi-objective path planning problem

The robot path planning problem is formulated as the multi-objective problem. The two objectives are obtaining short path and safe path of the robots.

The following is the definition of a point-mass robot adopted from [Prasad et al. \(2020a\)](#):

Definition 1. A j th point-mass P_j is a disk of radius $rp_j \geq 0$ and is positioned at $(x_j(t), y_j(t)) \in \mathbb{R}^2$ at time $t \geq 0$. Precisely, the point-mass is the set

$$P_j = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - x_j)^2 + (z_2 - y_j)^2 \leq rp_j^2\}$$

for $j = 1, 2, \dots, n$,

Definition 2. The target for P_j is a disk of center (p_{j1}, p_{j2}) and radius rt_j which is described as

$$T_j = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - p_{j1})^2 + (z_2 - p_{j2})^2 \leq rt_j^2\} \quad (5)$$

for $j = 1, 2, \dots, n$.

Ant representation

The motion of each robot will be guided by a colony of ants which are randomly distributed in the working environment. Since we are considering n point-mass robots, there will be n colonies altogether with each colony having n_j ants. The following is the definition of a moving ant in the j th colony:

Definition 3. The i th ant in the j th colony A_{ij} is a disk of radius $ra_{ij} \geq 0$ and is positioned at $(xa_{ij}(t), ya_{ij}(t)) \in \mathbb{R}^2$ at time $t \geq 0$. Precisely, the i th ant in the j th colony is the set

$$A_{ij} = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - xa_{ij})^2 + (z_2 - ya_{ij})^2 \leq ra_{ij}^2\}$$

for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, n$.

Short path

A robot will have a target seeking behaviour, that is, a robot should always be at a minimum distance from the target while it navigates through the cluttered environment. Therefore, the total length of the robot's path should be a minimum. Ants will be used to guide the robot to the target. In case of a multi-robot environment, each robot will have a set of ants associated with it for navigation. Since ants are used to plan the path for the robot, the fittest ant will be chosen for the robot's next step location. For each ant in the j th colony, an Euclidean formula shown in Eq. (6) is used to calculate the distance between the i th ant and the target:

$$d_{ij} = \sqrt{(p_{j1} - xa_{ij})^2 + (p_{j2} - ya_{ij})^2}. \quad (6)$$

Safe path

A path is safe if it has no obstacles in it. However, the workspace considered for this research has multiple obstacles, therefore obstacle avoidance becomes necessary. There are two types of obstacles used in this paper: (1) circular obstacles and (2) line segments. The following are the definitions of a circular obstacle and a line segment, adopted from Prasad et al. (2020a):

Definition 4. The l th circular obstacle with center (o_{l1}, o_{l2}) and radius $ro_l > 0$ on the z_1z_2 plane is described as

$$FO_l = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - o_{l1})^2 + (z_2 - o_{l2})^2 \leq ro_l^2\},$$

for $l = 1, 2, \dots, q$.

Definition 5. The k th line segment in the z_1z_2 plane, from the point (a_{k1}, b_{k1}) to the point (a_{k2}, b_{k2}) is the set

$$LO_k = \{(z_1, z_2) \in \mathbb{R}^2 : (z_1 - a_{k1} - \lambda_k(a_{k2} - a_{k1}))^2 + (z_2 - b_{k1} - \lambda_k(b_{k2} - b_{k1}))^2 = 0\},$$

where $\lambda_k \in [0, 1]$, $k = 1, 2, \dots, m$.

For a circular obstacle, Euclidean formula shown in Eq. (7) is used to calculate the shortest distance between i th ant in the j colony and the l th obstacle:

$$d1_{ijl} = \sqrt{(o_{l1} - xa_{ij})^2 + (o_{l2} - ya_{ij})^2} \quad (7)$$

for $i = 1, 2, \dots, n_j$, $j = 1, 2, \dots, n$ and $l = 1, 2, \dots, q$. Since there are many obstacles, the distance between each ant and the obstacles will be calculated. The sum of the distances between i th ant in the j th colony and obstacles is given by:

$$f1_{ij} = \sum_{l=1}^q d1_{ijl}. \quad (8)$$

To avoid a line segment, the distance between an ant and several points on that line segment are calculated, and the point generating minimum distance is considered. This technique is known as *minimum distance technique* (MDT) that has been adopted from Sharma, Vanualailai & Singh (2014). Avoiding the closest point on a line segment at any

given time will result in avoiding the entire line segment. Again, the Euclidean formula is used to calculate the distance of an ant with a point on the line segment:

$$d2_{ijk} = \sqrt{(a_{k1} + \lambda_{ijk}(a_{k2} - a_{k1}) - xa_{ij})^2 + (b_{k1} + \lambda_{ijk}(b_{k2} - b_{k1}) - ya_{ij})^2}, \quad (9)$$

where $\lambda_{ijk} \in [0, 1]$, $i = 1, 2, \dots, n_j$, $j = 1, 2, \dots, n$ and $k = 1, 2, \dots, m$. Like circular obstacles, there are many line segments, therefore the distance a point on the line segment and an ant will be calculated for each line segment and will be summed as:

$$f2_{ij} = \sum_{k=1}^m d2_{ijk}. \quad (10)$$

In a multi-robot environment, the inter-collisions between robots must be avoided. Since ants determine the next step location of a robot, each ant in the j th colony must avoid other robots which is considered as a moving obstacle. The Euclidean distance as shown in Eq. (11) is used to avoid robots from colliding with each other:

$$d3_{ijh} = \sqrt{(xa_{ij} - x_h)^2 + (ya_{ij} - y_h)^2}, \quad (11)$$

for $i = 1, 2, \dots, n_j$, $j = 1, 2, \dots, n$, $h = 1, 2, \dots, n$, $h \neq j$.

The sum of the distances between each ant and different robots is given by:

$$f3_{ij} = \sum_{\substack{h=1 \\ h \neq j}}^n d3_{ijh}. \quad (12)$$

Problem formulation

The problem is the minimization optimization problem which finds the optimal path for mobile robots in a cluttered environment. The fitness equation is designed by summing all the objective functions defined in this section:

$$f_{ij} = a \cdot \frac{1}{f1_{ij}} + b \cdot \frac{1}{f2_{ij}} + c \cdot \frac{1}{f3_{ij}} + d \cdot d_{ij} \quad (13)$$

which is the fitness of i th ant for robot j and a , b , c and d are control parameters. The ant having the minimum f_{ij} will be the fittest ant in the j th colony. It means that the ant is located at a safe distance from obstacles and at a minimum distance from the target. The j th robot will move to the fittest ant of the j th colony and this process will continue until the j th robot reaches its target. The control parameters a , b and c are the fitting parameters that decide path safety. With a high value of parameter a the ants will avoid the stationary circular obstacles from the greater distance. Similarly, a small value of b will mean that the ants will avoid the line segments from a closer distance which can compromise safety. However, when there is a decrease in the value of a , b and c , the chances of collision with obstacles, line segments and other robots are high. Likewise, a higher value of d will minimize the path length and a smaller value will maximize the path length. Therefore, a proper selection of these parameters decides the success of the objective function in planning the next step for a robot.

Motion control

The kinematic equations are used to control the robot to the step location generated ants. The kinematic equations will be used to generate smooth path for the robots in between the nodes.

PROPOSED ALGORITHM

The proposed new algorithm is a hybrid of the ant colony optimization and kinematic equations, named *ACO-Kinematic*. The choice of ACO variant was made from the results of [Socha & Dorigo \(2008a\)](#); [Socha & Dorigo \(2008b\)](#) where they showed that it outperformed other ACO variants and was equivalent to some heuristic algorithms for continuous domain. The robot's next step will be planned using the ant colony optimization algorithm and the robot will move to that step using its kinematic equations. In the literature, there are various hybrid models but according to the authors knowledge, there is none of this kind. The ACO-Kinematic pseudocode is shown in Algorithm 1.

Algorithm 1 ACO-Kinematic

Objective function $f(x)$

Initialize population of ants (Archive size)

Initialize robots initial and target positions

Initialize the weights and selection probabilities

While (robots current position < target position) do

 For all Ants in Archive Size do

 Calculate means

 End For

 For all Ants in Archive Size do

 Calculate standard deviation

 End For

 Create New Population Array (Sample Size)

 For all Ants in Sample Size do

 Construct solution based on Gaussian Kernal

 Evaluate new solutions

 End For

 Merge main population (archive) and new population (sample size)

 Rank ants and find the new best position

 Move the robot from current position to the new best position using Kinematic equations

 Update current position to new best

End While

Post-processing the results and visualization;

RESULTS

Table 1 shows the initial parameters of the ACO-Kinematic algorithm used in the three case studies discussed in the following subsections. Parameters 1–8 are used for path planing while parameters 9–10 are used for motion control. Safety parameters a , b and c will be used by users to fine tune avoidance of obstacles, line segments and other robots, respectively. Convergence parameter d determines the time it will take for a robot to reach its destination. Faster convergence means compromising the safety of the robot while the slower convergence means that the operation can be costly. Therefore, the parameters in this research have been adjusted with precaution. In this research the authors have deployed the brute force method to generate the values of the control parameters.

Case study 1: single point-mass robot and multiple obstacles

The proposed algorithm has been used to navigate the point-mass robot from source to destination in a cluttered environment. The kinematic equations governing the motion of a point-mass robot from its initial position (x_0, y_0) to another point (p_1, p_2) are

$$\dot{x} = \alpha_1(p_1 - x), \quad \dot{y} = \alpha_2(p_2 - y) \quad (14)$$

where α_1 and α_2 are positive real numbers.

Figures 1 and **2** show two scenarios where the point-mass robot avoids circular and line obstacles to reach its target. The path of the point-mass robot consists of points that have been generated by the ants. The robot moves from one step to another using its kinematic equations.

Case study 2: multiple point-mass robots and multiple obstacles

The proposed algorithm has been used to plan and control motion of multiple point-mass robots in a multiple obstacles (circular and rectangular shapes) environment. **Figure 3** shows the paths of three point-mass robots. The first robot (R1) has a initial position of $(5, 45)$ and the target placed at $(45, 5)$. The second robot (R2) is placed at the initial position $(5, 5)$ and has to reach the target at $(45, 45)$. The initial and target positions for the third robot (R3) are $(45, 25)$ and $(5, 25)$. The three robots start journey from their initial positions and have a goal to achieve, that is, to reach their target positions safely by taking a shortest route moving from one step to another. The three robots avoid all obstacles in their paths and also avoid colliding with each other. **Figure 4** shows the first robot (R1) and the second robot (R2) avoiding each other.

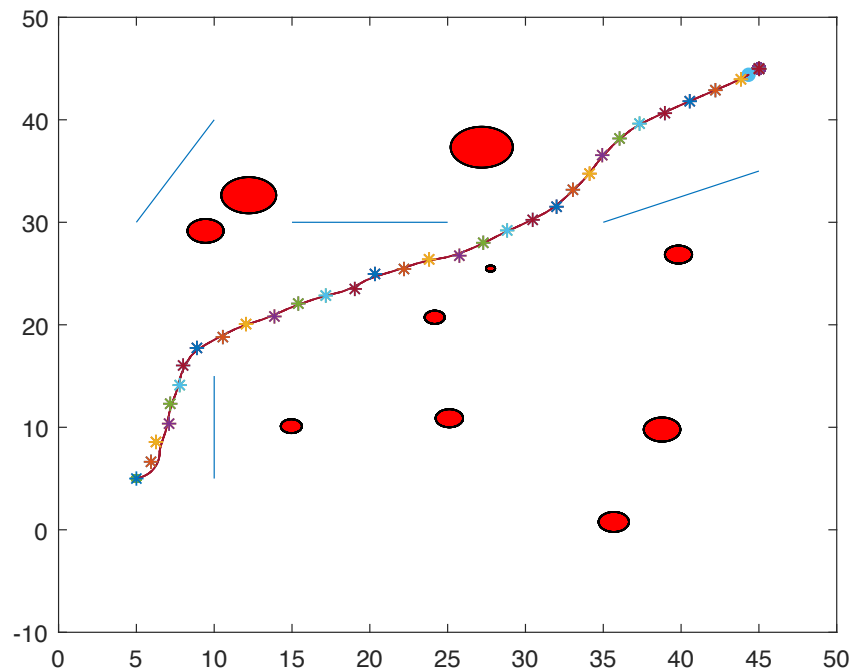
Application: tractor-trailer robot

The proposed algorithm has been used for motion planning and control of a tractor-trailer robot system. We consider a non-standard tractor-trailer robot which comprises of a rear wheel driven car-like vehicle and a hitched two-wheeled passive trailer attached to the rear axel of the vehicle (**Fig. 5**).

Let (x, y) represent the cartesian coordinates of the tractor robot, θ_0 be its orientation with respect to the x -axis, while ϕ gives the steering angle with respect to its longitudinal axis. Similarly, let θ_1 denote the orientation of the trailer with respect to the x -axis. Letting

Table 1 ACO-Kinematic parameters.

No.	Parameter	Value
1	Population size (Archive)	300
2	Sample size	10000
3	Deviation-distance ratio	1
4	Intensification factor	0.5
5	a	0.18
6	b	0.18
7	c	0.18
8	d	0.01
9	α_1	0.1
10	α_2	0.1

**Figure 1** A point-mass robots path with initial position (5, 5) and target position (45, 45).

Full-size  DOI: [10.7717/peerjcs.905/fig-1](https://doi.org/10.7717/peerjcs.905/fig-1)

L and L_t be the lengths of the mid-axle of the tractor and trailer, respectively, the motion of the tractor-trailer robot is governed by the following kinematic equations (*Prasad et al., 2020b*)

$$\begin{aligned}
 \dot{x} &= v \cos \theta_0 - \frac{v}{2} \tan \phi \sin \theta_0, \\
 \dot{y} &= v \sin \theta_0 + \frac{v}{2} \tan \phi \cos \theta_0, \\
 \dot{\theta}_0 &= \frac{v}{L} \tan \phi, \\
 \dot{\theta}_1 &= \frac{v}{L_t} \left(\sin(\theta_0 - \theta_1) - \frac{c}{L} \tan \phi \cos(\theta_0 - \theta_1) \right),
 \end{aligned} \tag{15}$$

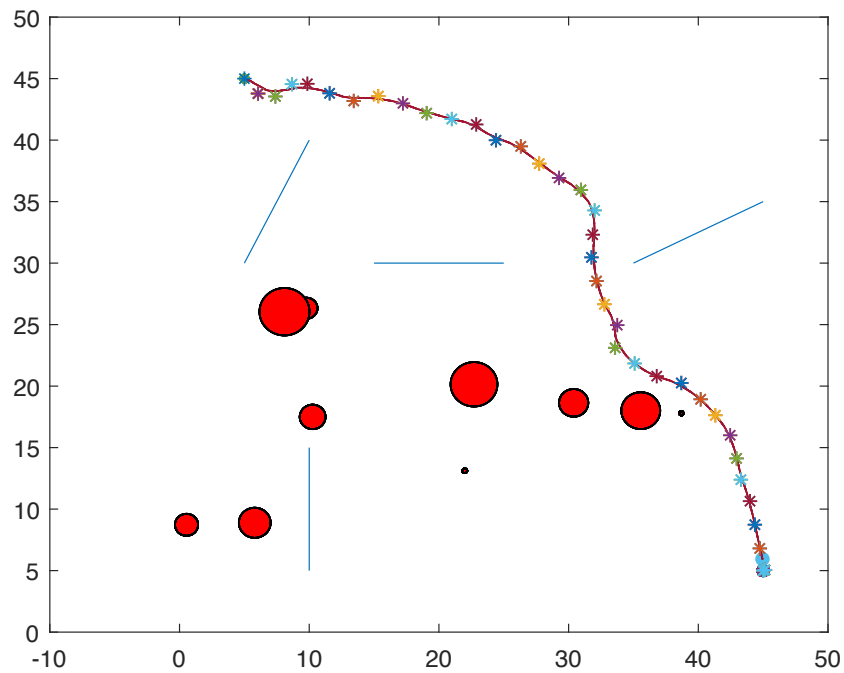


Figure 2 A point-mass robot's path with initial position (5, 45) and target position (45, 5).

Full-size  DOI: [10.7717/peerjcs.905/fig-2](https://doi.org/10.7717/peerjcs.905/fig-2)

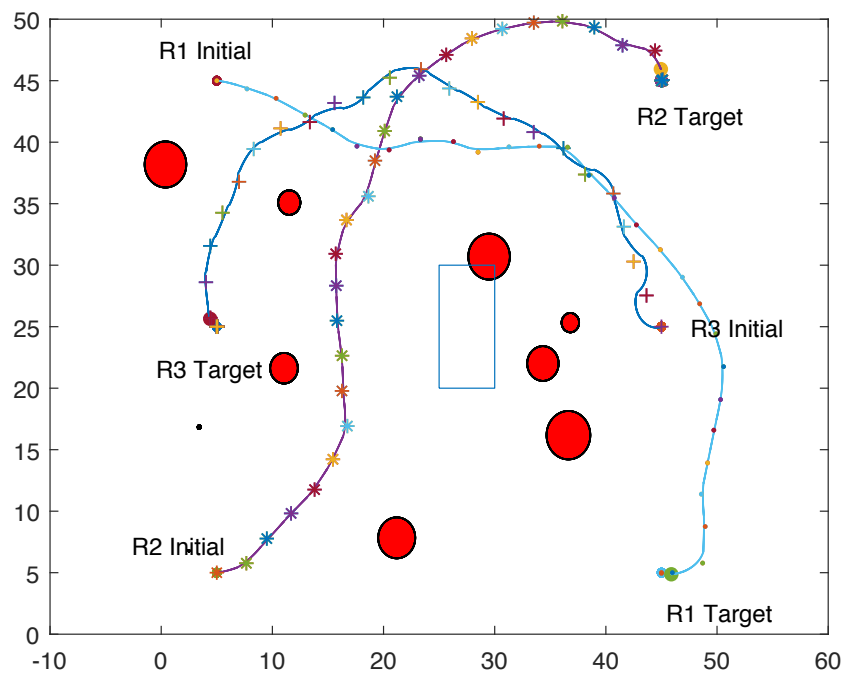


Figure 3 Paths of three point-mass robots.

Full-size  DOI: [10.7717/peerjcs.905/fig-3](https://doi.org/10.7717/peerjcs.905/fig-3)

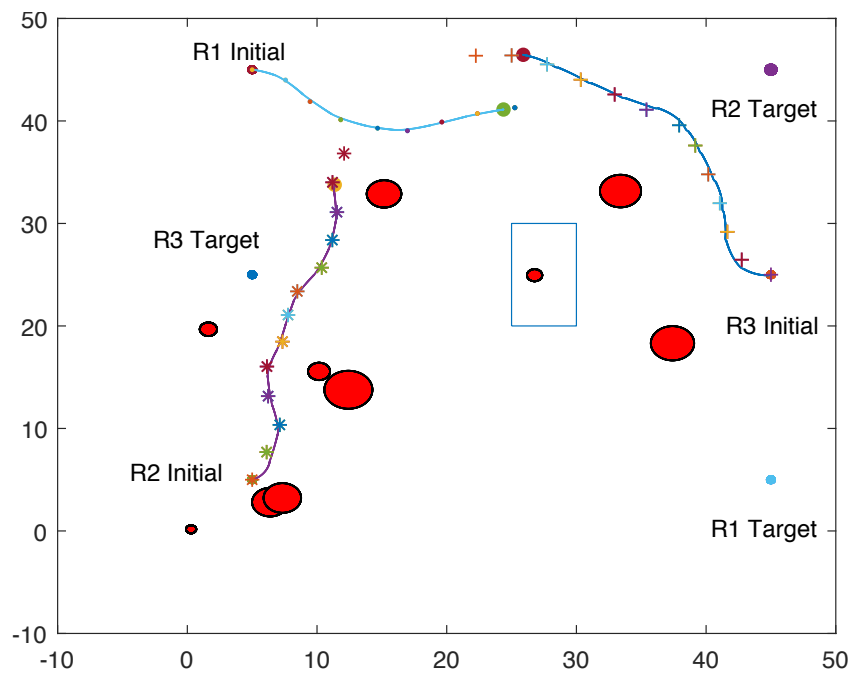


Figure 4 Partial paths of three point-mass robots with two robots avoiding each other.

Full-size  DOI: [10.7717/peerjcs.905/fig-4](https://doi.org/10.7717/peerjcs.905/fig-4)

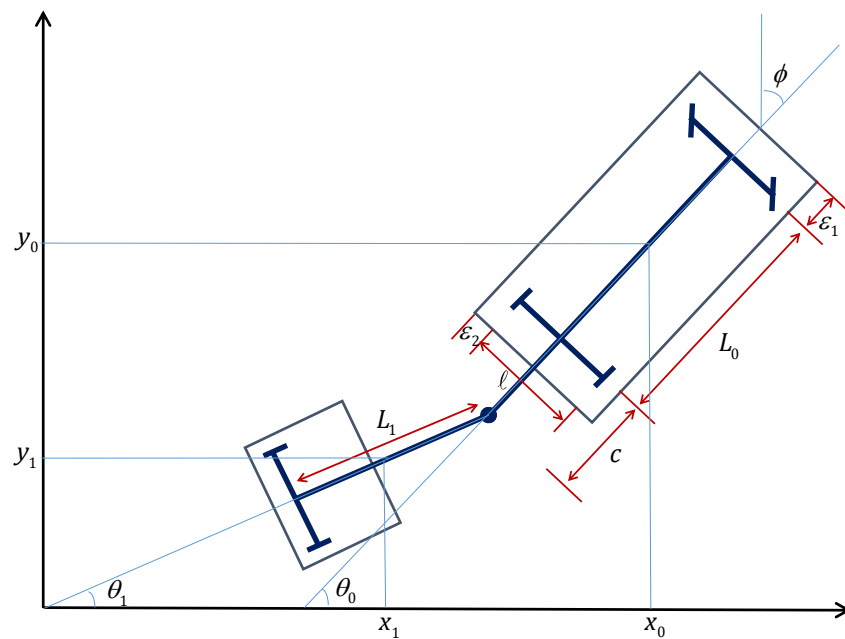


Figure 5 Schematic representation of the tractor-trailer robot.

Full-size  DOI: [10.7717/peerjcs.905/fig-5](https://doi.org/10.7717/peerjcs.905/fig-5)

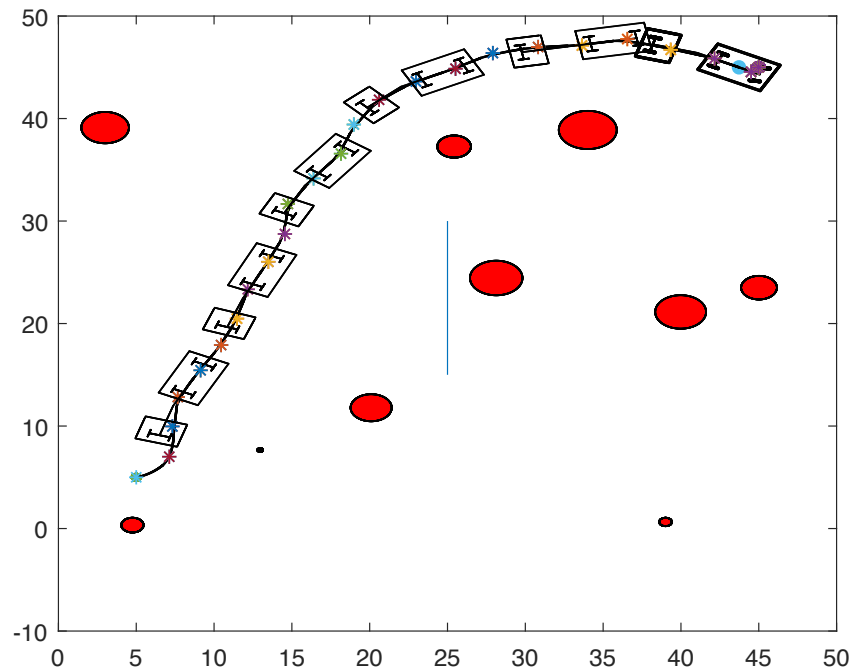


Figure 6 Path with initial position (5, 5) and target position (45, 45).

Full-size  DOI: [10.7717/peerjcs.905/fig-6](https://doi.org/10.7717/peerjcs.905/fig-6)

where v and ϕ which are the translational velocity and the steering angle, respectively, of the tractor robot, given as [Prasad et al. \(2020b\)](#)

$$v = \alpha \sqrt{(p_2 - y)^2 + (p_1 - x)^2},$$

$$\phi = \frac{7}{9} \tan^{-1} \left(\xi + \frac{\beta}{\cos|\theta_0 - \theta_1|} \right)$$

where α is a positive real number and $\beta = \max\{0, 0.5 - \cos|\theta_1 - \theta_0|\} \cdot \text{sign}(\theta_1 - \theta_0)$. Note that (p_1, p_2) is the next step for the robot generated by ant colony optimization and (x, y) is the current position of the robot. ξ is obtained by numerically solving the differential equation

$$\dot{\xi} = \frac{(p_2 - y) \cos \theta_0 - (p_1 - x) \sin \theta_0}{\sqrt{(x - p_1)^2 + (y - p_2)^2 + 0.01}} - \text{atan2}(p_2 - y, p_1 - x) + \theta_0,$$

$$\xi(0) = \text{atan2}(p_2 - y(0), p_1 - x(0)) - \theta_0(0)$$

[Figures 6](#) and [7](#) show the trajectories of one tractor-trailer robot in two different scenarios.

[Figures 8, 9, 10](#) and [11](#) show the paths of the three tractor-trailer robots. [Table 2](#) shows the initial and target positions for each robot.

[Figure 8](#) shows that R1 is avoiding R3 by stopping and waiting for R3 to pass. R1 has successfully avoided R3 and R1 resumes its journey towards the target, as shown in [Fig. 9](#). [Figure 10](#) shows that R2 and R3 avoid each other. The complete paths for the three robots are shown in [Fig. 11](#). The robots avoid each other using the fitness function given in

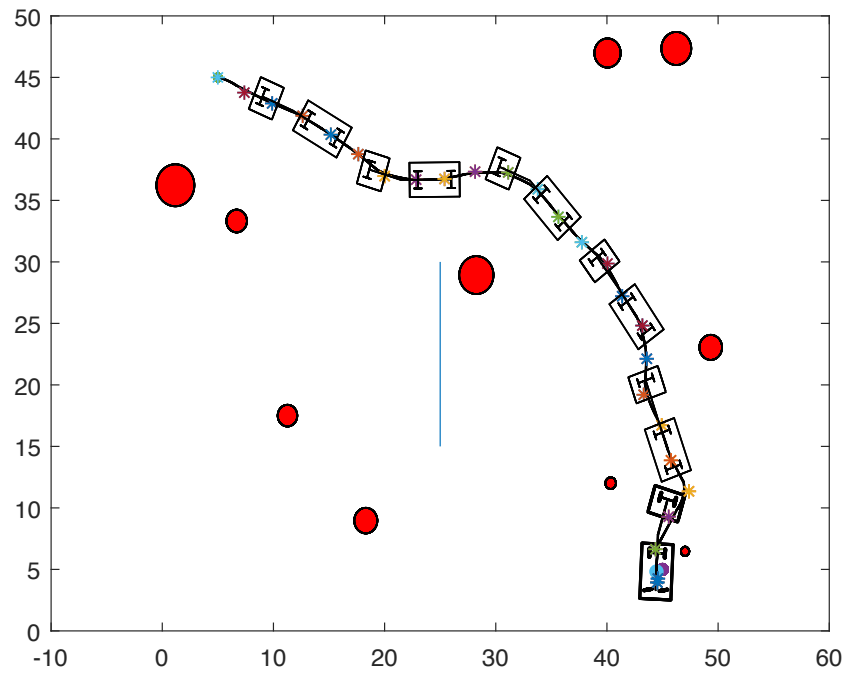


Figure 7 Path with initial position (5, 45) and target position (45, 5).

Full-size  DOI: 10.7717/peerjcs.905/fig-7

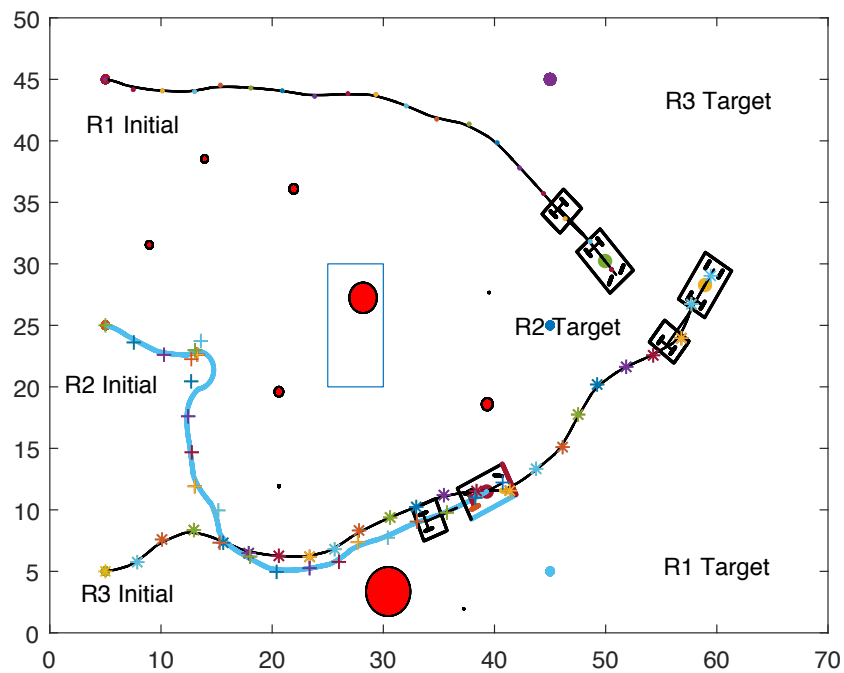


Figure 8 A tractor-trailer robot (R3) avoiding another tractor-trailer robot (R1).

Full-size  DOI: 10.7717/peerjcs.905/fig-8

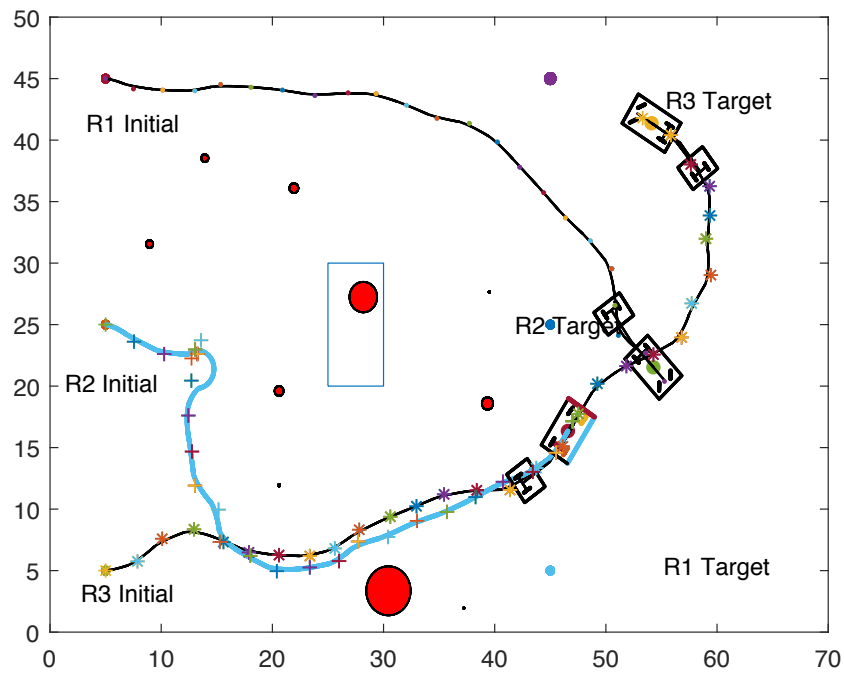


Figure 9 A tractor-trailer robot (R3) avoiding another tractor-trailer robot (R1).

Full-size  DOI: [10.7717/peerjcs.905/fig-9](https://doi.org/10.7717/peerjcs.905/fig-9)

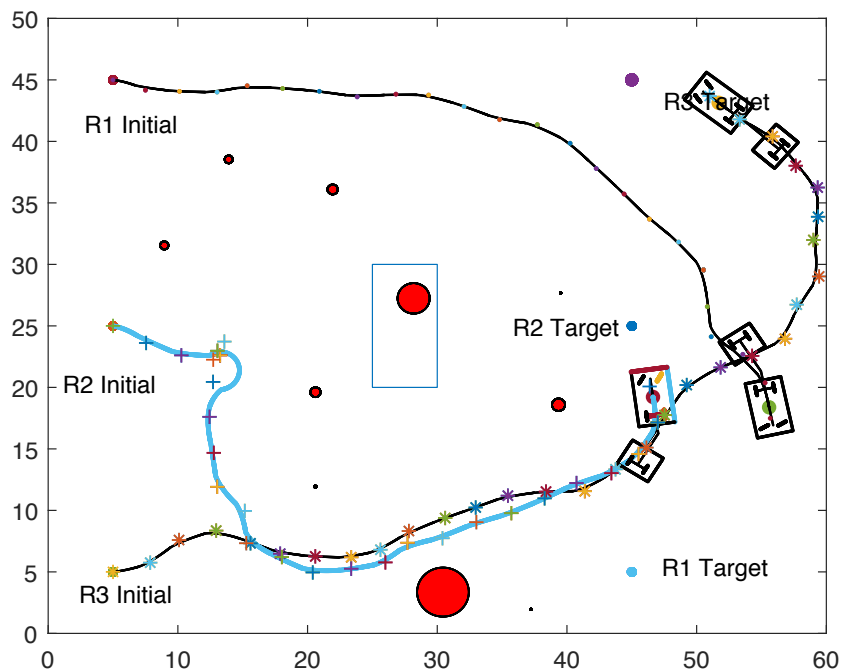


Figure 10 A tractor-trailer robot (R1) avoiding another tractor-trailer robot (R2).

Full-size  DOI: [10.7717/peerjcs.905/fig-10](https://doi.org/10.7717/peerjcs.905/fig-10)

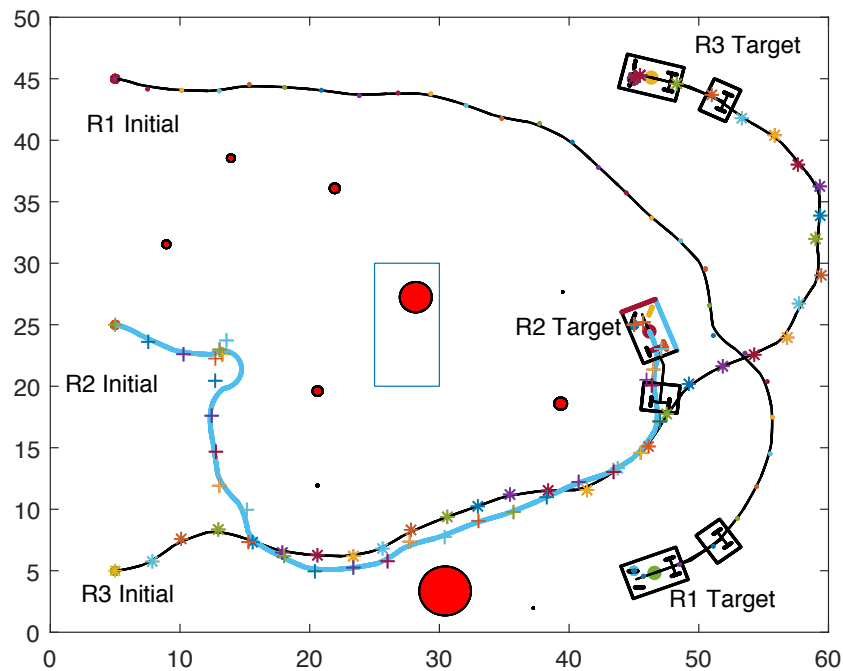


Figure 11 Complete paths of three tractor-trailer robots.

Full-size DOI: [10.7717/peerjcs.905/fig-11](https://doi.org/10.7717/peerjcs.905/fig-11)

Table 2 Robots' initial and target positions.

Robot	Initial position	Target position
R1	(5, 45)	(45, 5)
R2	(5, 25)	(45, 25)
R3	(5, 5)	(45, 45)

Eq. (13) which is used in ACO to determine the fittest ant. The robots are then controlled to the location of the fittest ants by the kinematic equations of the robots.

DISCUSSION

In this section, the performance of the proposed ACO-Kinematic algorithm will be compared with the Lyapunov-based Control Scheme (LbCS), which is a popular potential field-based method used to solve motion planning and control problem (*Sharma et al., 2018*). The performance will be measured in terms of path length and convergence time. Both algorithms have convergence and safety parameters. For LbCS, a larger convergence parameter value increases convergence time whereas a smaller convergence parameter value will decrease the convergence time. For ACO-Kinematic, a larger convergence parameter value decreases convergence time whereas a smaller convergence parameter value will increase the convergence time. Note that a quicker time to converge can affect a robot's safety. Therefore, the parameters need to be adjusted. There is no method in literature to adjust these parameters apart from the brute-force method. The authors have also used

Table 3 ACO-Kinematic parameters.

Parameter	Point-mass robot	Tractor-trailer	Range	Range source
a	0.02	0.02	0.01–1	
c	0.05	0.18	0.01–1	
d	0.01	0.01	0.0001–0.01	[20]

Table 4 LbCS parameters.

Parameter	Point-mass robot	Tractor-trailer	Range
beta	2	2	0.1–2
beta1	0.1	0.1	0.1–2
beta3	n/a	0.2	0.1–2
beta4	n/a	0.1	0.1–2
gamma	n/a	0.01	0.01–2
delta1	10	10	10–20
delta2	10	10	10–20

Table 5 Average path lengths and convergence time of two motion planning and control algorithms.

Scenario	Algorithms			
	LbCS		ACO-Kinematic	
	Time (s)	Path length (cm)	Time (s)	Path length (cm)
1	251.02	58.57	208.72	57.98
2	309.44	66.3	124.99	65.35
3	240.92	68.34	195.79	66.27

the brute-force method to obtain optimal parameters for the two algorithms that have been used to measure performance as shown in [Tables 3](#) and [4](#). The LbCS equations and parameters for the point-mass robot and tractor-trailer has been used from [Vanualailai, Ha & Nakagiri \(1998\)](#) and [Sharma et al. \(2008\)](#), respectively. The parameter values depend on the type of the robot which is also shown in the two tables. The convergence parameters d for ACO-Kinematic, and $delta1$ and $delta2$ for LbCS have the same values for both robots. Other parameters may differ in their values and are dependent on the type of the robots.

[Table 5](#) shows the average path lengths and the time it takes for a robot to reach the destination using ACO-Kinematic algorithm and LbCS. It shows the results for three scenarios. Robot motion planning and control with ACO-Kinematic algorithm for each scenario was iterated 30 times to calculate the average path length and time. LbCS inherently provides the same results for any run, therefore only 1 run was considered.

For Scenario 1, the trajectories of the point-mass robot for ACO-kinematic and LbCS are shown in [Figs. 12](#) and [13](#). On average, for ACO-Kinematic algorithm the point-mass robot took 208.72 s to reach the destination with the average path length of 57.98 cm. The best path length was 57.81 cm with the time of 201.4s while the worst path length was 58.29 cm with the time of 210.57s. For LbCS, the point-mass robot took 251.02 s to reach the

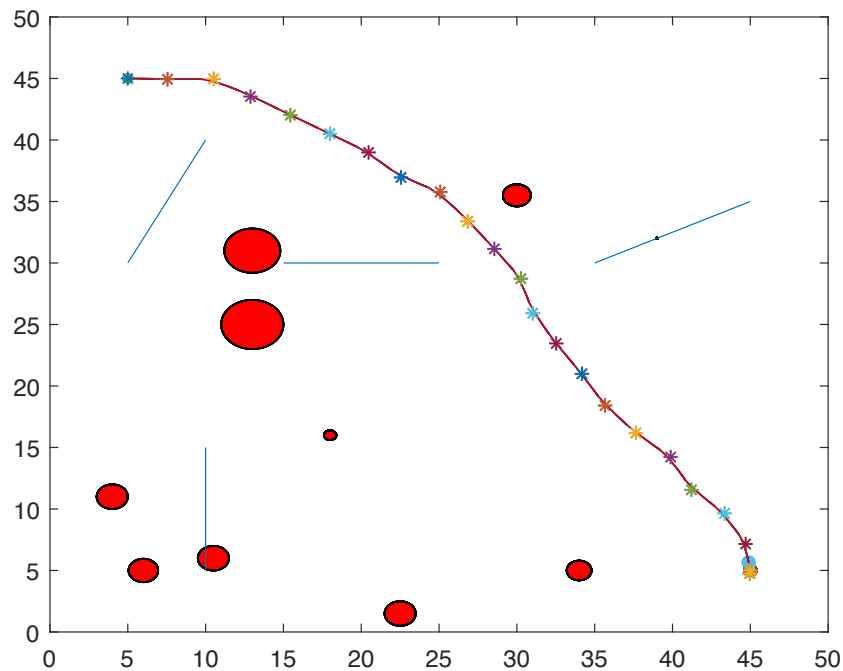


Figure 12 Scenario 1: A point-mass robot's path generated by ACO-Kinematic with initial position (5, 45) and target position (45, 5).

Full-size  DOI: [10.7717/peerjcs.905/fig-12](https://doi.org/10.7717/peerjcs.905/fig-12)

destination with the path length of 58.57 cm. The results of scenario 1 further show that the ACO-Kinematics worst-case path length and time are better than LbCS.

For Scenario 2, the trajectories of the point-mass robot ACO-kinematic and LbCS are shown in Figs. 14 and 15. The point-mass robot with LbCS had a path length of 66.3 cm and took 309.44s to reach the destination while the point-mass robot with ACO-Kinematic took 124.99s to cover the path length 65.35 cm. The best-case path length for ACO-Kinematic was 64.5 cm but it took the robot 138.14s to reach the destination. The worst-case path length for ACO-Kinematic was 66.28 cm and had a time of 101.28s.

For Scenario 3, the point-mass robots was replaced by a non-standard tractor-trailer robotic systems. Figures 16 and 17 show the trajectories of the tractor-trailer robot for ACO-Kinematic and LbCS, respectively. The tractor-trailer robot with the LbCS had a path length of 68.34 cm and time of 240.92s. The robot with ACO-Kinematic had the average path length of 66.27 cm and time of 195.79s. The ACO-Kinematic also provided the best-case path length of 62.99 cm with a time of 230.02s while the worst-case path length was 68.06 cm with a time of 238.05s.

Overall, in all 3 scenarios, ACO-Kinematic was able to achieve the shorter path in a time lesser than that of LbCS.

Figures 18 and 19 show the trajectories of the point-mass robots using ACO-Kinematic and LbCS for Scenario 4. While, the point-mass robot controlled by ACO-Kinematic was able to reach the target, the point-mass robot through LbCS controllers could not. This is because the LbCS system had entered into the local minima, which is one issue that most

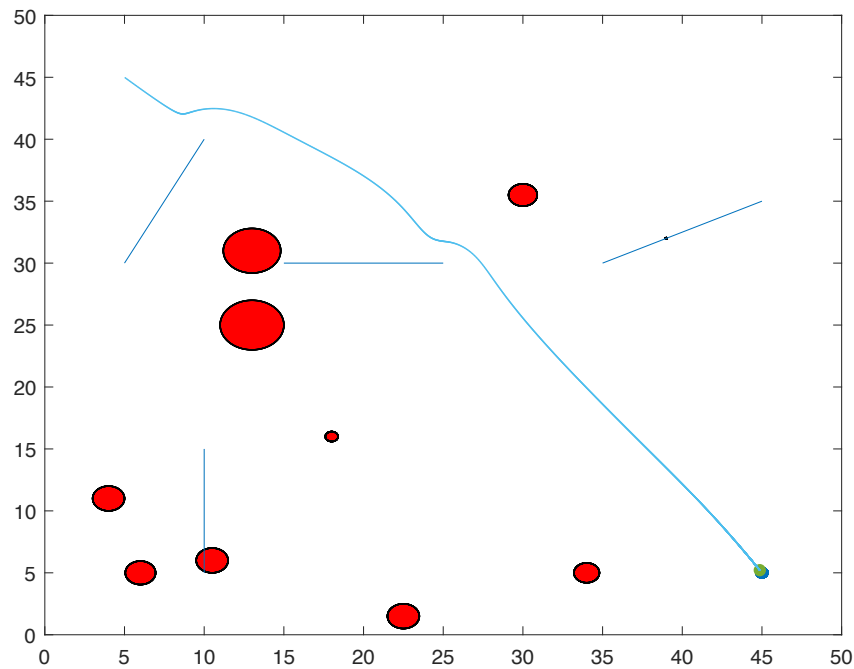


Figure 13 Scenario 1: A point-mass robot's path generated by LbCS with initial position (5, 45) and target position (45, 5).

Full-size  DOI: [10.7717/peerjcs.905/fig-13](https://doi.org/10.7717/peerjcs.905/fig-13)

artificial potential field methods face. The ACO-Kinematic algorithm was able to solve the problem of local minima and hence a better performer than a traditional motion planning and control algorithms like LbCS.

CONCLUSION AND FUTURE WORK

In this paper, a unique approach is proposed for solving motion planning and control problems. In particular, a new hybrid algorithm, ACO-Kinematic strategically made up of ant colony optimization and kinematic equations is presented. The new algorithm plans a step for a robot using ant colony optimization and the robot moves to that step using its kinematic equations. The algorithm is inherently capable of making the robot avoid obstacles and other robots while moving from initial to the target position. In the hybrid algorithm, the ACO plans the robot's next step while the kinematic equations controls the robot to that step location. In the authors' belief, this is the first time a hybrid algorithm of this kind is proposed for motion planning and control problem. The algorithm solves a multi-objective problem that consists of finding the safest and shortest path, successfully using kinematic equations to move the robot from one step to another and finally conveys to the final destination.

The ACO-Kinematic algorithm was used in three case studies. In the first case study, a point-mass robot navigated from initial position to target while avoiding multiple circular and line obstacles. The obstacles were static and the locations were known to robots. The second case study extended to multiple point-mass robots in a cluttered environment.

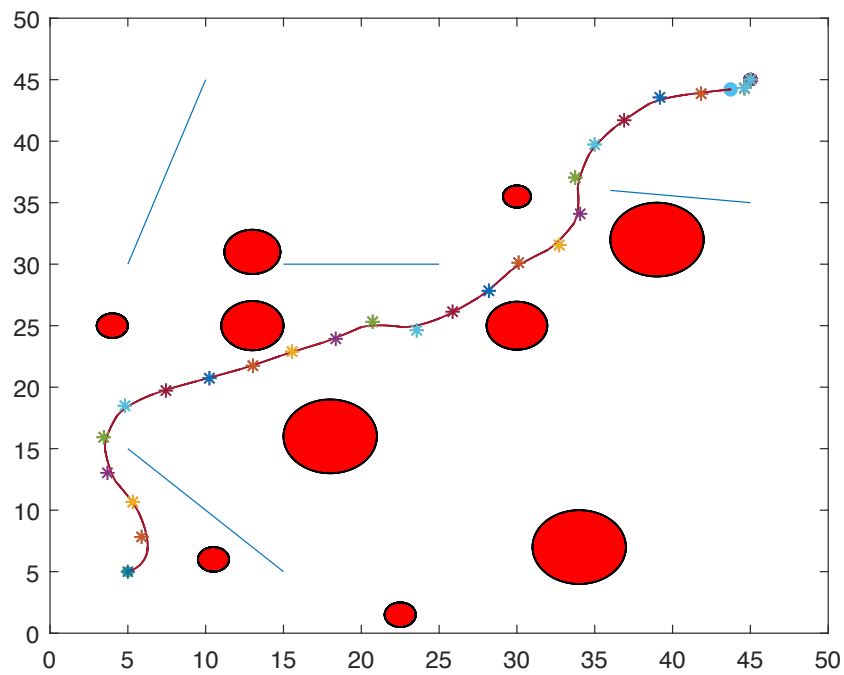


Figure 14 Scenario 2: A point-mass robot's path generated by ACO-Kinematic with initial position (5, 5) and target position (45, 45).

Full-size  DOI: [10.7717/peerjcs.905/fig-14](https://doi.org/10.7717/peerjcs.905/fig-14)

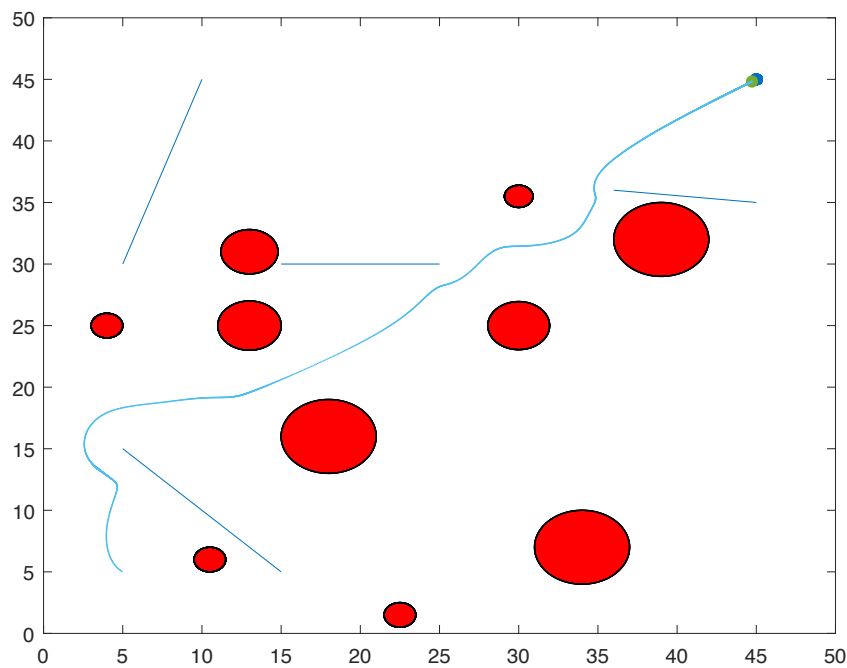


Figure 15 Scenario 2: A point-mass robot's path generated by LbCS with initial position (5, 5) and target position (45, 45).

Full-size  DOI: [10.7717/peerjcs.905/fig-15](https://doi.org/10.7717/peerjcs.905/fig-15)

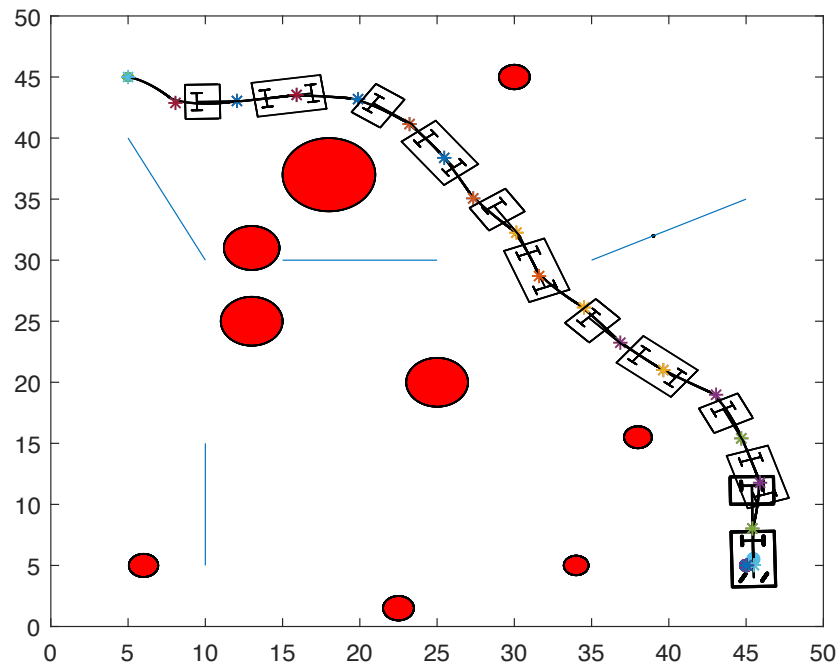


Figure 16 Scenario 3: A tractor-trailer robot's path generated by ACO-Kinematic with initial position (5, 45) and target position (45, 5).

Full-size  DOI: [10.7717/peerjcs.905/fig-16](https://doi.org/10.7717/peerjcs.905/fig-16)

In addition, the robots also avoided each other. The third case study was an application involving the non-standard 1-trailer robots. The tractor-trailer robots avoided obstacles and other tractor-trailer robots to reach the target from initial position.

The proposed algorithm, was also compared for performance with the Lyapunov based control scheme (LbCS) that has been used in a number of research on motion planning and control problems. Three scenarios were used for both algorithms and path lengths and convergence times were noted. The results showed that ACO-Kinematic algorithm was able to achieve shorter path lengths in relatively shorter convergence times in all three scenarios. It was also noted that LbCS was trapped in the local minima in the fourth scenario and the point-mass robot was not able to reach the target. The point-mass robot controlled by ACO-Kinematic algorithm was able to reach the target in the same scenario. This shows an arterial advantage of the ACO-Kinematic algorithm.

In the future work, the authors will apply ACO-Kinematic to static obstacles with unknown locations, dynamic obstacles and also to other mechanical systems. The authors will also carry out experimental analysis for verifications, comparisons and theoretical proofs of the new algorithm. Since this paper is a first for ACO-Kinematic hybrid, the authors will also upgrade the algorithm by adding new features including heuristic information to be part of ACO selection rule, handle noise in robot control and sensing, and handling imperfect knowledge of the environment.

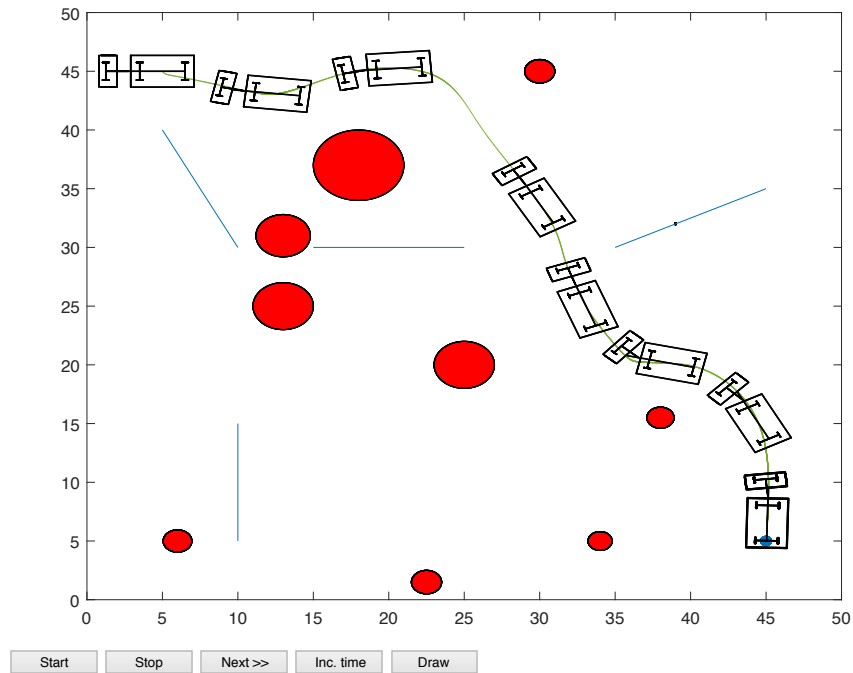


Figure 17 Scenario 3: A tractor-trailer robot's path generated by LbCS with initial position (5, 45) and target position (45, 45).

Full-size DOI: [10.7717/peerjcs.905/fig-17](https://doi.org/10.7717/peerjcs.905/fig-17)

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Kaylash Chaudhary conceived and designed the experiments, performed the experiments, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Avinesh Prasad performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Vishal Chand performed the experiments, prepared figures and/or tables, and approved the final draft.
- Bibhya Sharma performed the experiments, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The MATLAB code used for simulations are available in the [Supplementary Files](#).

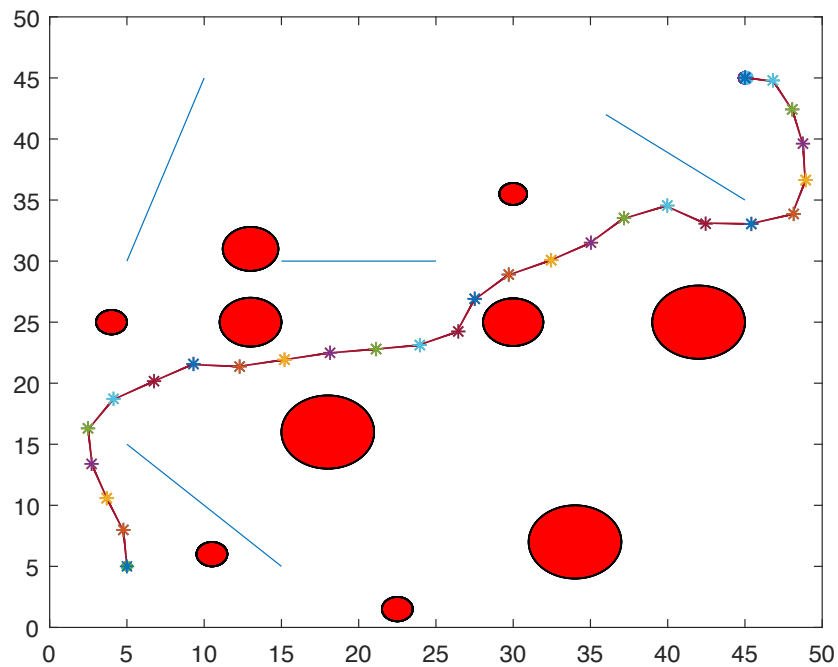


Figure 18 Scenario 4: A point-mass robot's path generated by ACO-Kinematic with initial position (5, 5) and target position (45, 45).

Full-size  DOI: [10.7717/peerjcs.905/fig-18](https://doi.org/10.7717/peerjcs.905/fig-18)

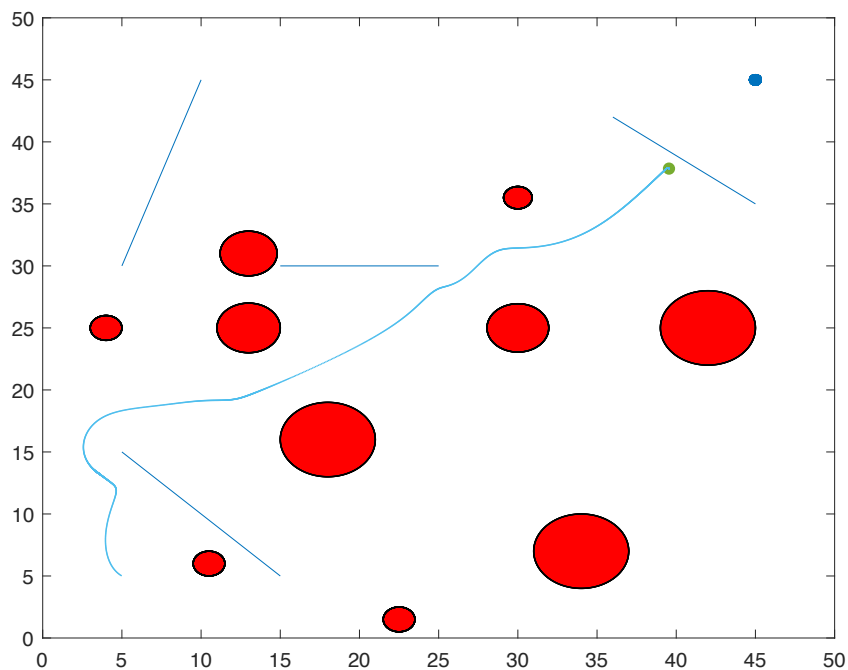


Figure 19 Scenario 4: A point-mass robot's path generated by LbCS with initial position (5, 5) and target position (45, 45).

Full-size  DOI: [10.7717/peerjcs.905/fig-19](https://doi.org/10.7717/peerjcs.905/fig-19)

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj-cs.905#supplemental-information>.

REFERENCES

- Bilchev G, Parmee IC. 1995.** The ant colony metaphor for searching continuous design spaces. In: *AISB workshop on evolutionary computing*. 25–39.
- Bortoff S. 2000.** Path planning for UAVs. In: *Proceedings of the 2000 american control conference. ACC (IEEE Cat. No.00CH36334), vol.1*. Piscataway: IEEE, 364–368
DOI [10.1109/ACC.2000.878915](https://doi.org/10.1109/ACC.2000.878915).
- Brand M, Masuda M, Wehner N, Yu X-H. 2010.** Ant colony optimization algorithm for robot path planning. In: *2010 international conference on computer design and applications, volume 3*. V3–436.
- Buniamin N, Sariff N, Wan Ngah W, Mohamad Z. 2011.** Robot global path planning overview and a variation of ant colony system algorithm. *International Journal of Mathematics and Computers in Simulation* 5(1):9–16.
- Chen Y, Peng H, Grizzle JW. 2017.** Fast trajectory planning and robust trajectory tracking for pedestrian avoidance. *Ieee Access* 5:9304–9317
DOI [10.1109/ACCESS.2017.2707322](https://doi.org/10.1109/ACCESS.2017.2707322).
- Devi A, Vanualailai J, Kumar SA, Sharma B. 2017.** A cohesive and well-spaced swarm with application to unmanned aerial vehicles. In: *Proceedings of the 2017 international conference on unmanned aircraft systems*. Miami, FL, USA, 698–705.
- Dréo J, Siarry P. 2004.** Continuous interacting ant colony algorithm based on dense heterarchy. *Future Generation Computer Systems* 20(5):841–856
DOI [10.1016/j.future.2003.07.015](https://doi.org/10.1016/j.future.2003.07.015).
- Dunbabin M, Marques L. 2012.** Robots for environmental monitoring: significant advancements and applications. *IEEE Robotics Automation Magazine* 19(1):24–39
DOI [10.1109/MRA.2011.2181683](https://doi.org/10.1109/MRA.2011.2181683).
- Guo H, Shen C, Zhang H, Chen H, Jia R. 2018.** Simultaneous trajectory planning and tracking using an MPC method for cyber-physical systems: a case study of obstacle avoidance for an intelligent vehicle. *IEEE Transactions on Industrial Informatics* 14(9):4273–4283
DOI [10.1109/TII.2018.2815531](https://doi.org/10.1109/TII.2018.2815531).
- Khurshid J, Bing-rong H. 2004.** Military robots - a glimpse from today and tomorrow. In: *ICARCV 2004 8th control, automation, robotics and vision conference, 2004, vol. 1*. 771–777
DOI [10.1109/ICARCV.2004.1468925](https://doi.org/10.1109/ICARCV.2004.1468925).
- Kloetzer M, Mahulea C, Gonzalez R. 2015.** Optimizing cell decomposition path planning for mobile robots using different metrics. In: *2015 19th international conference on system theory, control and computing (ICSTCC)*. 565–570
DOI [10.1109/ICSTCC.2015.7321353](https://doi.org/10.1109/ICSTCC.2015.7321353).
- Kumar SA, Vanualailai J, Sharma B, Prasad A. 2021.** Velocity controllers for a swarm of unmanned aerial vehicles. *Journal of Industrial Information Integration* 22:100198
DOI [10.1016/j.jii.2020.100198](https://doi.org/10.1016/j.jii.2020.100198).

- Lee MC, Park MG. 2003.** Artificial potential field based path planning for mobile robots using a virtual obstacle concept. In: *Proceedings 2003 IEEE/ASME international conference on advanced intelligent mechatronics (AIM 2003)*, vol.2. Piscataway: IEEE, 735–740 DOI [10.1109/AIM.2003.1225434](https://doi.org/10.1109/AIM.2003.1225434).
- Lingelbach F. 2004.** Path planning using probabilistic cell decomposition. In: *IEEE international conference on robotics and automation, 2004. proceedings. ICRA '04. 2004, vol. 1.* 467–472 DOI [10.1109/ROBOT.2004.1307193](https://doi.org/10.1109/ROBOT.2004.1307193).
- Martinez-Alfaro H, Gomez-Garcia S. 1998.** Mobile robot path planning and tracking using simulated annealing and fuzzy logic control. *Expert Systems with Applications* **15**(3–4):421–429 DOI [10.1016/S0957-4174\(98\)00055-4](https://doi.org/10.1016/S0957-4174(98)00055-4).
- Murphy RR, Kravitz J, Stover SL, Shoureshi R. 2009.** Mobile robots in mine rescue and recovery. *IEEE Robotics Automation Magazine* **16**(2):91–103 DOI [10.1109/MRA.2009.932521](https://doi.org/10.1109/MRA.2009.932521).
- Napper SA, Seaman RL. 1989.** Applications of robots in rehabilitation. *Robotics and Autonomous Systems* **5**(3):227–239 DOI [10.1016/0921-8890\(89\)90047-X](https://doi.org/10.1016/0921-8890(89)90047-X).
- Ning Y, Yue M, Yang L, Hou X. 2020.** A trajectory planning and tracking control approach for obstacle avoidance of wheeled inverted pendulum vehicles. *International Journal of Control* **93**(7):1735–1744 DOI [10.1080/00207179.2018.1530455](https://doi.org/10.1080/00207179.2018.1530455).
- Otte MW. 2015.** *A survey of machine learning approaches to robotic path-planning.* Boulder: University of Colorado at Boulder.
- Patle B, Pandey A, Jagadeesh A, Parhi DR. 2018.** Path planning in uncertain environment by using firefly algorithm. *Defence Technology* **14**(6):691–701 DOI [10.1016/j.dt.2018.06.004](https://doi.org/10.1016/j.dt.2018.06.004).
- Prasad A, Sharma B, Vanualailai J. 2017.** Motion control of a pair of cylindrical manipulators in a constrained 3-dimensional workspace. In: *2017 4th Asia-Pacific World congress on computer science and engineering (APWC on CSE)*. 75–81 DOI [10.1109/APWC on CSE.2017.00022](https://doi.org/10.1109/APWC on CSE.2017.00022).
- Prasad A, Sharma B, Vanualailai J, Kumar S. 2020a.** Stabilizing controllers for landmark navigation of planar robots in an obstacle-ridden workspace. *Journal of Advanced Transportation* **2020**:1–13 DOI [10.1155/2020/8865608](https://doi.org/10.1155/2020/8865608).
- Prasad A, Sharma B, Vanualailai J, Kumar S. 2021.** Motion control of an articulated mobile manipulator in 3D using the Lyapunov-based control scheme. *International Journal of Control* Epub ahead of print Apr 30 2021 DOI [10.1080/00207179.2021.1919927](https://doi.org/10.1080/00207179.2021.1919927).
- Prasad A, Sharma B, Vanualailai J, Kumar SA. 2020b.** A geometric approach to target convergence and obstacle avoidance of a nonstandard tractor-trailer robot. *International Journal of Robust and Nonlinear Control* **30**(13):4924–4943 DOI [10.1002/rnc.5021](https://doi.org/10.1002/rnc.5021).
- Raj J, Raghuwaiya K, Sharma B, Vanualailai J. 2021.** Motion control of a flock of 1-trailer robots with swarm avoidance. *Robotica* **39**(11):1926–1951 DOI [10.1017/S0263574721000060](https://doi.org/10.1017/S0263574721000060).
- Raj J, Raghuwaiya K, Vanualailai J. 2020.** Collision avoidance of 3D rectangular planes by multiple cooperating autonomous agents. *Journal of Advanced Transportation* **2020**:1–13 DOI [10.1155/2020/4723687](https://doi.org/10.1155/2020/4723687).

- Raj J, Raghuwaiya K, Vanualailai J, Sharma B. 2018.** Navigation of car-like robots in three-dimensional space. In: *2018 5th Asia-Pacific world congress on computer science and engineering (APWC on CSE)*. 271–275 DOI [10.1109/APWCOnCSE.2018.00051](https://doi.org/10.1109/APWCOnCSE.2018.00051).
- Raj J, Raghuwaiya KS, Vanualailai J. 2020.** Novel Lyapunov-based autonomous controllers for quadrotors. *IEEE Access* **8**:47393–47406 DOI [10.1109/ACCESS.2020.2979223](https://doi.org/10.1109/ACCESS.2020.2979223).
- Reshamwala A, Vinchurkar DP. 2013.** Robot path planning using an ant colony optimization approach: a survey. *International Journal of Advanced Research in Artificial Intelligence* **2**(3):65–71.
- Saputro JS, Rusmin PH, Rochman AS. 2018.** Design and implementation of trajectory tracking motion in mobile robot skid steering using model predictive control. In: *2018 IEEE 8th international conference on system engineering and technology (ICSET)*. Piscataway: IEEE, 73–78 DOI [10.1109/ICSEngT.2018.8606361](https://doi.org/10.1109/ICSEngT.2018.8606361).
- Sharma B, Singh S, Vanualailai J, Prasad A. 2018.** Globally rigid formation of n-link doubly nonholonomic mobile manipulators. *Robotics and Autonomous Systems* **105**:69–84 DOI [10.1016/j.robot.2018.02.006](https://doi.org/10.1016/j.robot.2018.02.006).
- Sharma B, Vanualailai J, Singh S. 2014.** Tunnel passing maneuvers of prescribed formations. *International Journal of Robust and Nonlinear Control* **24**(5):876–901 DOI [10.1002/rnc.2923](https://doi.org/10.1002/rnc.2923).
- Sharma BN, Vanualailai J, Raghuwaiya K, Prasad A. 2008.** New potential field functions for motion planning and posture control of 1-trailer systems. *International Journal of Mathematics and Computer Science* **3**(1):45–71.
- Shokouhifar M. 2021.** FH-ACO: Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing. *Applied Soft Computing* **107**:107401 DOI [10.1016/j.asoc.2021.107401](https://doi.org/10.1016/j.asoc.2021.107401).
- Socha K, Dorigo M. 2008a.** Ant colony optimization for continuous domains. *European Journal of Operational Research* **185**(3):1155–1173 DOI [10.1016/j.ejor.2006.06.046](https://doi.org/10.1016/j.ejor.2006.06.046).
- Socha K, Dorigo M. 2008b.** Ant colony optimization for continuous domains. *European Journal of Operational Research* **185**(3):1155–1173 DOI [10.1016/j.ejor.2006.06.046](https://doi.org/10.1016/j.ejor.2006.06.046).
- Tanaka Y, Tsuji T, Kaneko M, Morasso PG. 1998.** Trajectory generation using time scaled artificial potential field. In: *Proceedings. 1998 IEEE/RSJ international conference on intelligent robots and systems. Innovations in theory, practice and applications (Cat. No. 98CH36190), vol. 1*. Piscataway: IEEE, 223–228.
- Tsai C, Wang T. 2005.** Nonlinear control of an omnidirectional mobile robot. In: *Proceedings of the 8th Intl. Conf. Autom Technol, volume 732*. 727.
- Vanualailai J, Ha JH, Nakagiri S-i. 1998.** A solution to the two-dimensional findpath problem. *Dynamics and Stability of Systems* **13**(4):373–401 DOI [10.1080/02681119808806270](https://doi.org/10.1080/02681119808806270).
- Vanualailai J, Raj J, Raghuwaiya K. 2021.** Autonomous quadrotor maneuvers in a 3D complex environment. In: *Advances in computer, communication and computational sciences*. Vol: 1158. Singapore: Springer, 221–231.

- Viana IB, Kanchwala H, Ahiska K, Aouf N. 2021.** A comparison of trajectory planning and control frameworks for cooperative autonomous driving. *Journal of Dynamic Systems, Measurement, and Control* **143**(7):071002 DOI [10.1115/1.4049554](https://doi.org/10.1115/1.4049554).
- Wahid N, Zamzuri H, Abdul Rahman MA, Kuroda S, Raksincharoensak P. 2017.** Study on potential field based motion planning and control for automated vehicle collision avoidance systems. In: *2017 IEEE international conference on mechatronics (ICM)*. Piscataway: IEEE, 208–213 DOI [10.1109/ICMECH.2017.7921105](https://doi.org/10.1109/ICMECH.2017.7921105).
- Wang H, Zhou Z. 2019.** A heuristic elastic particle swarm optimization algorithm for robot path planning. *Information* **10**(3):99 DOI [10.3390/info10030099](https://doi.org/10.3390/info10030099).
- Wang Z, Tan D, Ge G, Liu S. 2020.** Optimal trajectory planning and control for automatic lane change of in wheel motor driving vehicles on snow and ice roads. *Automatic Control and Computer Sciences* **54**(5):432–445 DOI [10.3103/S0146411620050090](https://doi.org/10.3103/S0146411620050090).
- Watanabe K, Shiraishi Y, Tzafestas SG, Tang J, Fukuda T. 1998.** Feedback control of an omnidirectional autonomous platform for mobile service robots. *Journal of Intelligent and Robotic Systems* **22**(3):315–330 DOI [10.1023/A:1008048307352](https://doi.org/10.1023/A:1008048307352).
- Zhang C, Chu D, Liu S, Deng Z, Wu C, Su X. 2019.** Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control. *IEEE Intelligent Transportation Systems Magazine* **11**(2):29–40 DOI [10.1109/MITS.2019.2903536](https://doi.org/10.1109/MITS.2019.2903536).
- Zips P, Bck M, Kugi A. 2013.** Fast optimization based motion planning and path-tracking control for car parking. *IFAC Proceedings* **46**(23):86–91 DOI [10.3182/20130904-3-FR-2041.00042](https://doi.org/10.3182/20130904-3-FR-2041.00042).