We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Motion Planning for Mobile Robots

Xiangrong Xu, Yang Yang and Siyu Pan

Additional information is available at the end of the chapter

Abstract

This chapter introduces two kinds of motion path planning algorithms for mobile robots or unmanned ground vehicles (UGV). First, we present an approach of trajectory planning for UGV or mobile robot under the existence of moving obstacles by using improved artificial potential field method. Then, we propose an I-RRT* algorithm for motion planning, which combines the environment with obstacle constraints, vehicle constraints, and kinematic constraints. All the simulation results and the experiments show that two kinds of algorithm are effective for practical use.

Keywords: path planning, trajectory planning, mobile robots, unmanned ground vehicles (UGV)

1. Introduction

In recent years, the intelligent mobile robots have played an important role in industry, agriculture, aerospace, and space exploration, and it becomes a hotspot issue in many research fields. The robot is divided into two categories from the application environment, named industrial robots and special robots. The industrial robot is industrial oriented multi-joint manipulator or multi-degree of freedom robot, while the special robot is opposite to industrial robots, which is used for non-manufacturing environment or service including service robots, underwater robot, entertainment robot, military robots, agricultural robots, and robotic machines. If we want the robot autonomous mobile in configuration space, the very first thing we will do is the path planning, which can be defined as the process of finding a collision-free path for a robot from its initial position to the goal or target point by avoiding collisions with any static obstacles or other agents present in its environment. In order to solve these problems, many domestic and foreign scholars researched and put forward many theories and methods of the path planning [1–4]. The traditional path planning algorithms, such as ant

colony algorithm, genetic algorithm, and artificial potential field algorithm, dealing with some motion planning have its unique superiority. In addition, the algorithm based on potential field or heuristic function, such as A*, D*, and artificial potential field method, in addressing the problem of planning can meet the requirements of real-time and the optimality. This chapter introduces two kinds of path planning algorithm of the robot, including mobile robots and unmanned ground vehicles (UGV).

2. Path planning for robots with existence of moving obstacles

In this chapter, we present an approach of the ground mobile vehicle or robot trajectory planning under the existence of moving obstacles by using improved artificial potential field method. The potential field intensity and strength instead of force vectors was adopted in the planning. Considering the speed effect of mobile obstacles and mobile robot, the velocity information was introduced into potential field function and an "added potential field" was also applied to guide the mobile robot to be free of collision with local obstacles. Based on the new method, all the potential field intensity was considered and summed by algebraic style, then the genetic trust region algorithm was used to search the minimum sum points of potential field intensity within the motion space and scope, which the mobile robot can reach target during a sampling period, and the global optimization trajectory was consisted of all the minimum points. Simulation and experiment results show that better results of path planning for a mobile robot in a complex environment with the existence of moving obstacles can be achieved using this new approach.

The problem of mobile robot path planning under a dynamic environment in mobile robot domain is hot and difficult. Its task is to find a path, which is from the initial state to goal state. Currently, common path planning methods are artificial potential field method, grid method, neural network method, chaos genetic algorithm, aand so on [5, 6]. Artificial potential field method in the algorithm of path planning is relatively mature and efficient, because of the simplicity of the mathematical calculation, it is widely used. However, there are the issues of local minimum point and destination unreachable in traditional artificial potential field method. There are a variety of ways to escape from the local minimum point, such as random fleeing method, heuristic search, walking along the wall, Tangent bug's method, and so on [7, 8], These methods need to apply an additional control force to the robot, which could not fundamentally solve problems. An improved artificial potential field method based on genetic algorithm was proposed in [8] to solve the problem of goal unreachable and local minimum, but the convergent speed of the algorithm still needs to be improved.

In this chapter, aimed at the shortcoming of traditional artificial potential field method, an improved artificial potential field applied in a dynamic environment is proposed. Firstly, an improved artificial potential field model is established; then the genetic trust region algorithm is applied to solve the sub-target point problem of the improved artificial potential field model and then the optimal path is planned.

2.1. Artificial potential field model

2.1.1. The shortcoming of traditional potential field model

There are some problems when this method is applied to robot path planning:

1. Because the traditional artificial potential field method applies virtual force to control the movement of the robot, it is possible for a robot that it cannot go through a narrow passage when two obstacles are near to each other. Moreover, if robot, obstacles and target point are on the same straight line, the robot controlled by force can only move on the straight line repeatedly, but it cannot reach to the target point.

2. If obstacles are near to target point, it will be possible that repulsive force is greater than attractive force, leading to the problem of goal unreachable (GNRON).

3. The robot has not yet arrived at target point nearly, however, the summation of suffered forces is zero, consequently, it will fall in local minimum point and stop moving.

2.1.2. Improved measures

Aimed at the defects of traditional artificial potential field model, improved measures are proposed as below:

1. Turning the attractive force of target to the robot and the repulsive force of obstacles to the robot into a kind of potential field intensity, a method of calculating potential field is applied to replace traditional vector force control.

2. Adding a coefficient entry $\|X - X_\mu\|_2$ in the gravitational potential of obstacles, when the robot is close to the target point, gravitational potential is reducing as well as repulsion potential. Finally, they are zero until the robot arrives at the target point, so the problem of obstacles and target point being too close causing goal unreachable.

3. For a "deadlock" issue caused by local minimum point, the "added potential field" is introduced to guide the robot to walk out the local minimum point, that is, adding an extra potential field U_{add}.

2.1.3. Improved potential model

According to above improved measures, the improved artificial potential field model is proposed. The attractive force model of the target to a full range of vehicle's body is:

$$U_{att}(X) = 0.5k\rho^2(X, X_g) \tag{1}$$

where $\rho(X, X_g)$ is the distance between the current location of the central point of mobile vehicle's body and target point; k is a proportional gain coefficient; X is the position $[x, y]^T$ of robot's central point in movement space; and X_g is the target point position $\left[x_g, y_g\right]^T$.

Repulsive potential model of the i-th static obstacles on the full range of movement of the body is:

$$
U_{reps}(X_i) = \begin{cases} 0.5\eta(\dfrac{1}{\rho(X,X_i)} - \dfrac{1}{\rho_0})^2 \|X - X_g\|_2, \\ \qquad \text{if } \rho(X,Xi) \leqslant \rho o \\ 0, \\ \qquad \text{if } \rho(X,Xi) > \rho o \end{cases}
$$

where $i \in (1, 2''', n)$, n is the summation of static obstacles; $\rho(X, X_i)$ is the shortest distance of between current location of the center of mobile vehicle's body and the i-th obstacle; ρ_0 the effective effect distance of obstacle; and η is proportional position gain coefficient.

Dynamic obstacles are in motion, and it cannot reflect environment information completely if only taking location into consideration. Bring the relative speed of dynamic obstacles and robots into potential field function, Repulsive potential model of the i-th static obstacles on the full range of movement of the body is got:

$$
U_{reps}(X_i) = \begin{cases} 0.5\eta(\dfrac{1}{\rho(X,X_i)} - \dfrac{1}{\rho_0})^2 \left(\begin{array}{l} \|X - X_g\|_2 + \\ \zeta|v - v_r \sin(\varphi - \Phi)| \end{array} \right), \\ \qquad \text{if } \rho(X,Xi) \geqslant \rho o \\ 0, \\ \qquad \text{if } \rho(X,Xi) > \rho o \end{cases}
$$

where $r \in (1, 2, \ldots, m)$, m is the summation of mobile obstacles; ζ is proportional coefficient; V is the current speed of the mobile body, $v \in (2v_{\max}/3, v_{\max})$; $v_{\max}t_0$ is the current speed of the r-th dynamic obstacles; ϕ is the current movement direction of the mobile body; and φ is the current movement direction of the r-th dynamic obstacle.

When the robot is in local minimum point, the "added potential" is brought to figure the problem of local minimum out, the added potential model is:

$$
U_{add}(X) = \begin{cases} s\rho^2(X, X_g), \rho(X, X_g) > \rho_a; \\ 0, \rho(X, X_g) \leq \rho_a \end{cases}
$$

where ρ_a judgment distance of whether the mobile body reaches to a target point; s is a proportional coefficient.

Therefore, the whole potential field intensity of a range of mobile body is shown in (2). When the robot is in local minimum point, taking (2) plus the added potential as the total potential field value (as shown in (2)).

$$U = U_{att}(X) + \sum_{i=1}^{n} U_{reps}(X_i) + \sum_{r=1}^{m} U_{repm}(X_r). \tag{2}$$

$$U = U_{att}(X) + \sum_{i=1}^{n} U_{reps}(X_i) + \sum_{r=1}^{m} U_{repm}(X_r) + U_{add}(X). \tag{3}$$

Based on above model, every sampling period all regard the minimum point of potential field sum as a sub-goal point, and multiple sub-target is consisted of global optimization path. The sum of potential field is shown in (2). In order to avoid the case of target vibrating in the vicinity of local minimum point, vector synthesis method is used to judge whether the robot is in local minimum point, if in the local minimum point, potential field sum contains added potential, as shown in (3). Assuming the maximum speed of robot is v_{max}, the sampling period is t_0, so when robot in the reachable range of every sampling period, it takes current location as a center, and $v_{max}t_0$ is a radius of the circle. The speed of robot movement should not too big or too small in order to ensure the stability of robot movement and performing efficiency, so a sub-goal point can be chosen from an annular region, $R \in (2V_{max}t_0/3, V_{max}t_0), \theta \in (0, 2\pi)$.

As **Figure 1** shows, the shaded part of the annular region is the optional region of the sub-target point. Therefore, solving the sub-target issue is the key to improve artificial potential field method.

2.2. Solving target point based on genetic trust region

In order to improve the efficiency of path planning, below two situations will be considered:

(1) When the robots move in the free space far away from obstacle ($\leq \rho_0$), that solving the minimum point of the sum of annular region potential field intensity shown in **Figure 1**. The trust region method in optimal search algorithm has good reliability and fast convergence, which offers a new idea to solve the sub-target. However, its iteration speed sometimes is affected by the radial trust. When the iteration speed is affected by the radius of trust region, using genetic algorithms to solve a point, which is better than one in the current iteration point, then, based on the point the trust region is restarted until the optimized point is found, which can greatly improve the convergence speed of algorithm [9–11].

The points of the shaded part can show as $x' = x + R\cos\theta, y' = y + R\sin\theta$. Therefore, Eqs. (2) and (3) are the function about variables R and θ, $R \in (2V_{max}t_0/3, V_{max}t_0), \theta \in (0, 2*3.14)$. assuming $U_{add}(z)z_1' = x + R\cos\theta, z_2' = y + R\sin\theta, z = (z_1, z_2)$. Using genetic trust region algorithm to solve the target function of sub-goal point as (4) shows, in other words, that solving a class of linear constrained optimized problem, when that the robot is located in local minimum point, (4) and added potential $U_{add}(z)$ are regarded as the target function.

$$\min U(z) = U_{att}(z) + \sum_{i=1}^{n} U_{reps}(z_i) + \sum_{r=1}^{n} U_{repm}(z_r) \tag{4}$$

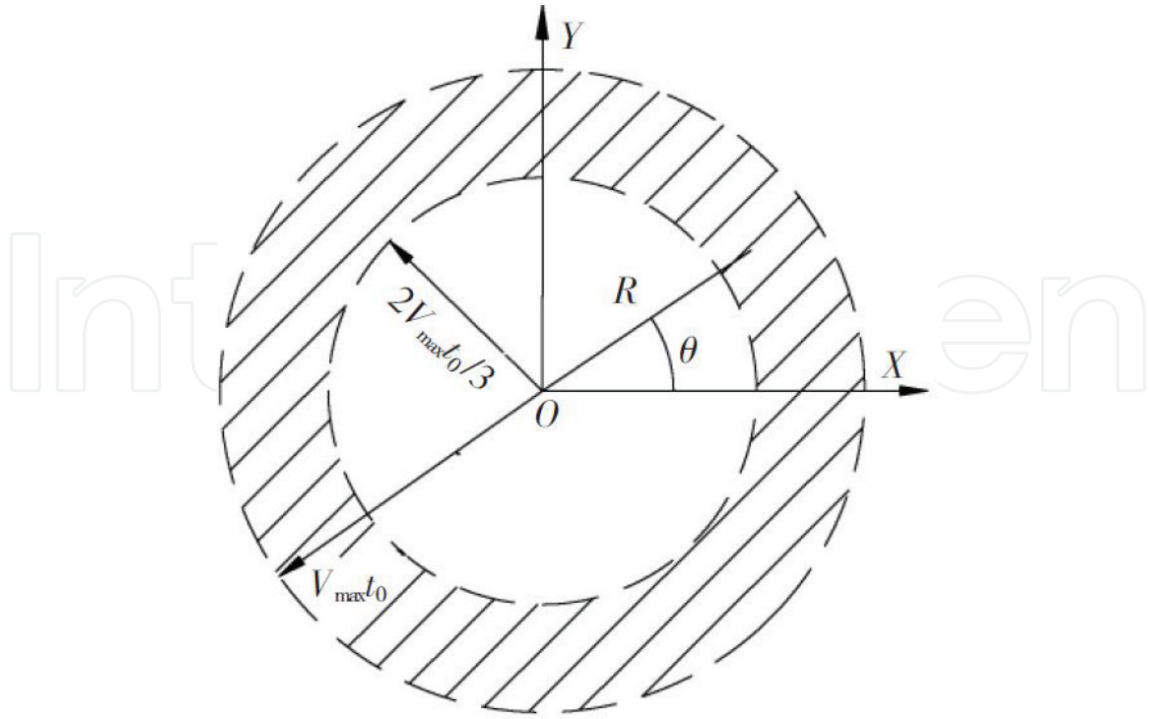Using quadratic approximation and constructing constrained trust region sub-problem:

Figure 1. The optional range of sub-target point.

$$\min q_k(d) = g_k^T d + 0.5 d^T G_k d$$
$$s.t. \|d\|_2 \le \Delta_k, z_k + d_k \in \Omega \tag{5}$$

where $g_k = \nabla U(z_k)$; Δ_k is the radius of trust region; $G_k = \nabla^2 U(z_k)$. solving G_k is very compli-cated, and using BFGS formula of quasi-Newton to structure Hessian matrix B_k, which is approximate to G_k. d_k is the decline tentative step. Ω is the range of R and θ.

The algorithm contains, and remark shows: $z \in R^2$, $z_k = (z_{k1}, z_{k2})$, $z_{k+1} = (z_{(k+1)1}, z_{(k+1)2})$, $z^e{}_{k+1} = (z^e{}_{(k+1)1}, z^e{}_{(k+1)2})$, $y_k = g_{k+1} - g_k$, := value assignment, $B_{k+1}^{BFGS} = B_k + y_k y_k^T / y_k^T d_k - B_k d_k B_k^T / d_k^T B_k d_k$, the amount of actual decline in the ratio of the amount of pre-estimated decline is:

$$r_k = \Delta U_k / \Delta q_k = \frac{U(z_k) - U(z_k + d_k)}{q_k(0) - q_k(d_k)}.$$

The first step in the algorithm.

Step 1: Initialization, given: $z_0, \Delta_0 > 0, \varepsilon_1 > 0, \varepsilon_2 > 0, \varepsilon_3 > 0, a > 0, b > 0 M > 1$.

$$0 < \eta_1 < \eta_2 < 1, B_0 =: I_{2x2}, 0 < \beta_1 < \beta_2 < 1 \le \beta_3, k := 0$$

Step 2: Calculating g_k, if $\|g_k\|_2 < \varepsilon_1$, stop calculating, and output the result.

Step 3: Solving sub-problem in Eq. (5), and get the decline tentative step d_1.

Step 4: Calculating r_k, if $r_k > \eta_1$ meanwhile $z_k + d_k \in \Omega$, so, $z_{k+1} := z_k + d_k$ rectifies B_{k+1}, or $z_{k+1} := z_k, B_{k+1} := B_k$.

Step 5: Choosing Δ_{k+1}, let it meet:

$$\Delta_{k+1} \begin{cases} \left\{ [\Delta_k, \beta_3 \Delta_k], r_k > \eta_2; \right. \\ [\beta_2 \Delta_k, \Delta_k], r_k \in [\eta_1, \eta_2]; \\ [\beta_1 \Delta_k, \beta_2 \Delta_k], r_k < \eta_1. \end{cases}$$

Step 6: If $\Delta_{k+1} < \varepsilon_2, \|U_{k+1} - U_k\| < \varepsilon_3$, and using genetic algorithm to quickly solve min $U(z_{k+1}^e)$ so that an iteration point which is better than the current point, go to step 7, Or back to step 8.

Step 7: If $\|U_{k+1} - U_k\| > M\|U_{k+1} - U_k\|$, so order $z_{k+1} := z_{k+1}^e G_{k+1} := \Delta_0$, back to step 2; or back to step 8.

Step 8: Using BFGS formula to modify B_k and gets $B_{k+1}, k := k + 1$, back to step 2.

$\varepsilon_2, \varepsilon_3, M$ values can be adjusted to control the number of times the call number of sub-issues of a genetic algorithm to achieve the optimization algorithm speed, regulating ε_1 also can regulate the speed of convergence of the algorithm, but there are trade-offs to optimize the value obtained, need to be considered in accordance with the actual situation.

When the convergence speed is influenced by the radius of the trust region in step 6 of algorithm 1, using a genetic algorithm to figure $\min U(z_{k+1}^e)$ out must meet:

$$z_{k+1}^e \in \Omega, c < z_{k+1}^2, l < z_{k+1}^e < n.$$

where

$$c = z_{k1}^e - a, f = z_{k1}^e + a, l = z_{k2}^e - b, n = z_{k2}^e + b,$$

when solving $z_{(k+1)1}^e, z_{(k+1)2}^e, c, f, l, n$ are all known quantity.

Fitness function: establishing the mapping relationship between the objective function and the function of moderate: $G(z) = 0.618^{U(z)}$; adopting the binary code to encode; replication strategy to preserve the best individual mixed roulette selection; crossover operator is single point crossover; and mutation operator for the basic bit mutation.

The procedure of algorithm 2:

Step 1: Parameter initialization: population size N, crossover probability P_c, mutation probability P_m, current algebraic $T_c = 0$, maximum algebraic $T_{\max}$, and $0 < k < 0.618$. The coding initial solution produces a unit corresponding to the initial solution. And it randomly produces two l-bit binary string working as one unit, and it does not stop until it generates N units.

Step 2: Solving adaption value of every individual in the current group, to get the optimization z_{1b}, z_{2b}: described as:

If

$$T_c \leq T_{\max}$$

If

$$G(z_b) - G(z_k) > k \text{ and } z \in \Omega, \text{ then } z^e_{(k+1)1} := z_{1b},$$

$z^e_{(k+1)2} := z_{2b}$, stopping calculating, and backing to the step1 of algorithm 1.

ELSE backing to step 3.

ELSE updating initial parameters, baking to step 1.

Step 3: Replacing the two worst individuals in the two optimizations of the current generation, then selecting N individuals by using roulette selection.

Step 4: Manipulating cross mutation to produces a new population. Backing to step 2.

2.3. Simulation and experiment

We select the initial parameters of the algorithm:

$\triangle_0 = 0.05 \|U(z_0)\|$, $\varepsilon_1 = 0.1$, $\varepsilon_2 = 0.03$, $\varepsilon_3 = 0.05$, $a = b = 0.5$, $M = 1.5$, $\eta_1 = 0.15$, $\eta_2 = 0.3$, $B_0 = I_{2x2}$, $\beta_1 = 0.35$, $\beta_2 = 0.75$, $\beta_3 = 1.25$, .

$N = 20$, $P_c = 0.99$, $P_m = 0.05$, $T_{\max} = 100$, $k = 0.1$, coding length$l = 32.v_{\max} = 0.3m/s$, $t_0 = 3s$, $z_0 = (5v_{\max}t_0/6.0)$, $k = 1$, $\eta = 2$, $\zeta = 0.1$, $n = 4$, $m = 1$, .

$s = 0.3$. The speed of mobile obstacles.

$v_r = 0.2(0.5) + r$ and $(m)/s$,

$$\varphi = \pi/2, \rho_0 = 2m, \rho_a = 0.15m$$

The original point of robots is (0, 0), the target point is (20, 24), and the unit is m (meter).

The simulation results are shown in **Figures 4** and **5**. It can be seen from simulation, that to combine trust region algorithm and the improved artificial potential field method can optimize mobile robot path planning, and get a better solution to local minima and the unreachable target problem. Under the same conditions, a pure genetic algorithm for solving the sub-goal problem, the average convergence time comes to the second level, and the individual points solvers need more than 30 s, therefore, the proposed algorithm is much fast.

Robot trajectory simulation resulting potential field on a typical point 1 seen from **Table 1**, the optimal path in the intensity values obtained are shown in table potential field strength decreases rapidly, and the strong field in the times to reach the target point, the basic value to 30 s.

To verify the validity of the path planning algorithm, an experiment was made based on the current conditions as shown in **Figures 2** and **3**. The equipment used in the experiment include mobile robot, laser tracker sensor, distance detector sensor, digital compass, dynamics obstacles, and still obstacles.

Point	x/m	y/m	strength of field
1	0. 63	0. 64	460. 440
2	2. 33	2. 74	382. 110
3	4. 48	5. 60	313. 620
4	4. 43	9. 00	273. 710
5	6. 89	11. 61	216. 690
6	9. 54	14. 04	142. 790
7	10. 42	17. 26	108. 600
8	13. 30	19. 41	72. 980
9	16. 27	21. 41	38. 310
10	19. 97	23. 97	0. 001

Table 1. Magnitude of field of the point at the trajectory.

The laser tracker can precisely measure the center point of the coordinate (x, y) when mobile robot is in motion. Then it can send the distance information to the robot controller through a local network and get the location of the robot in the space. The mobile robot is shaped like a cylinder, with 0.6 m in diameter, 12 ultrasonic distance detect sensors with 0.1–6.0 m detect distance, and 32 infrared distance sensors with 0.1–0.8 m detect distance. The angle between two adjacent sensors is 30°. The measurement time is 0.015 s. The initial position of the robot is (1.375, −2.328), and the target position is (1.248, 1.353), the velocity of moving obstacle is V_r = 0.08 m/s, and $\varphi = \pi/6$. The velocity of the mobile robot is V = 0.05 m/s, the moving direction can be measured in real-time by the digital compass.

The initial parameters of the experiment are k = 1, n = 2, m = 1 $\eta = 2, \zeta = 0.1, \rho_0 = 1m$, $\rho_a = 0.15m$. The laser tracker recorded many points of the mobile robot, and the approximated trajectory can be obtained from these points as shown in **Figures 4** and **5**. And the relationship between the field strength and algorithm implementation time was listed in **Table 2**. From the experiment, we can see that the algorithm has some advantages compared with those traditional methods. It is faster in implementation and computation.

2.4. Commits

We presented an approach of mobile robot trajectory planning under the existence of moving obstacles by using improved artificial potential field method. To further verify the improved artificial potential field method for moving machines effectiveness of robots path planning, the use of the existing conditions, design path planning experiment = experimental scene shown in "laboratory equipment including full orientation of the mobile robot laser tracker digital
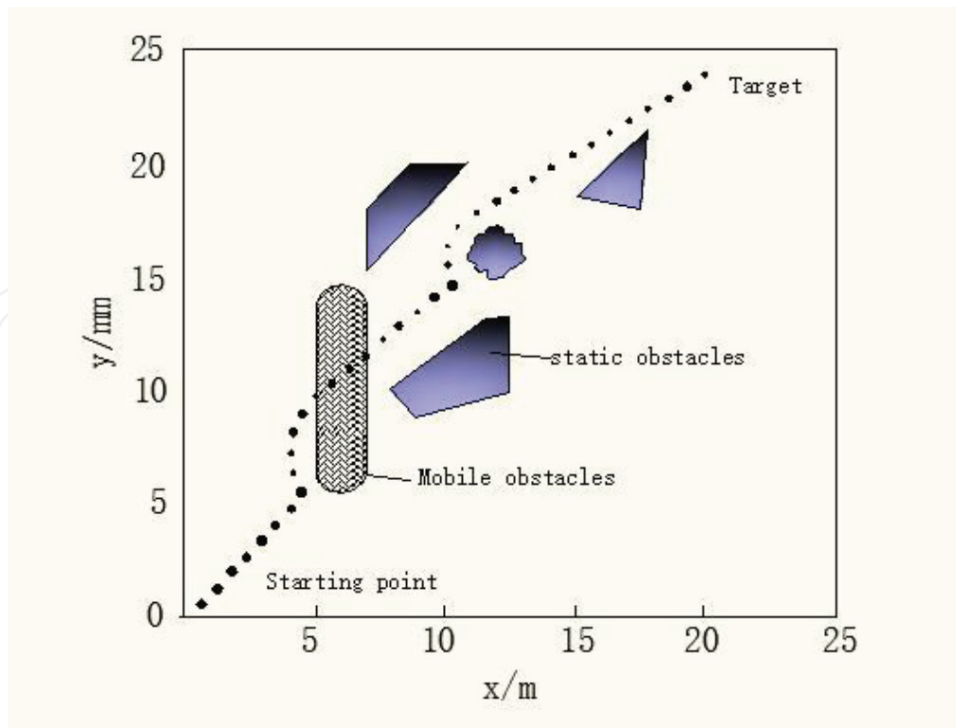
Figure 2. The simulation result of improved artificial potential field method.
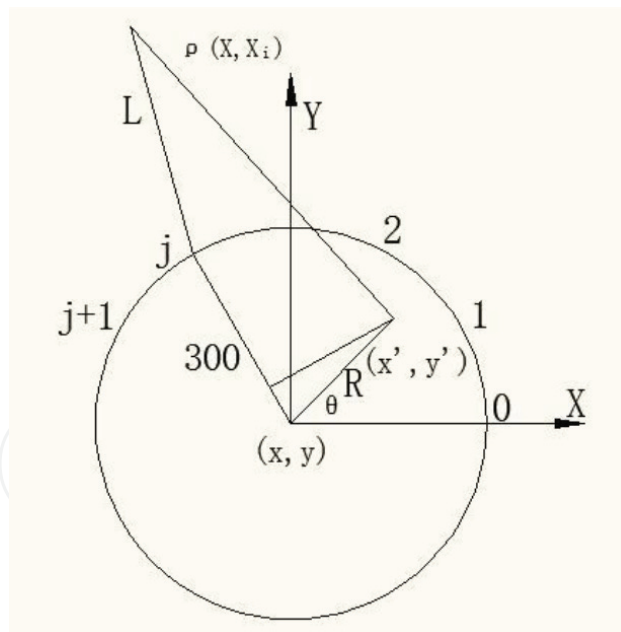

Figure 3. Geometric representation for computing target potion.

distance sensor compass static obstacles and dynamic obstacles. Laser tracker can accurately measure mobile robot motion moving the coordinate value of the center point of the process (x, y), through a wireless local area network location information measured in real-time sent to the robot controller, robot positioning cylindrical omnidirectional mobile robot structure with a diameter 0.6 m, and surrounded by 12 uniform ultrasonic ranging.
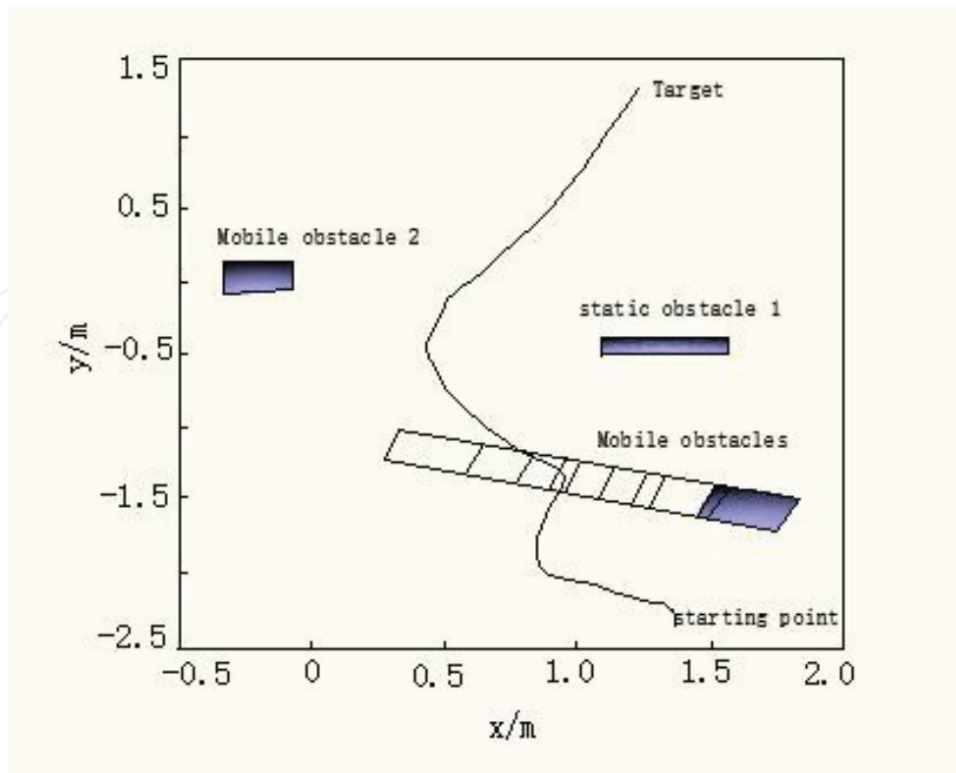
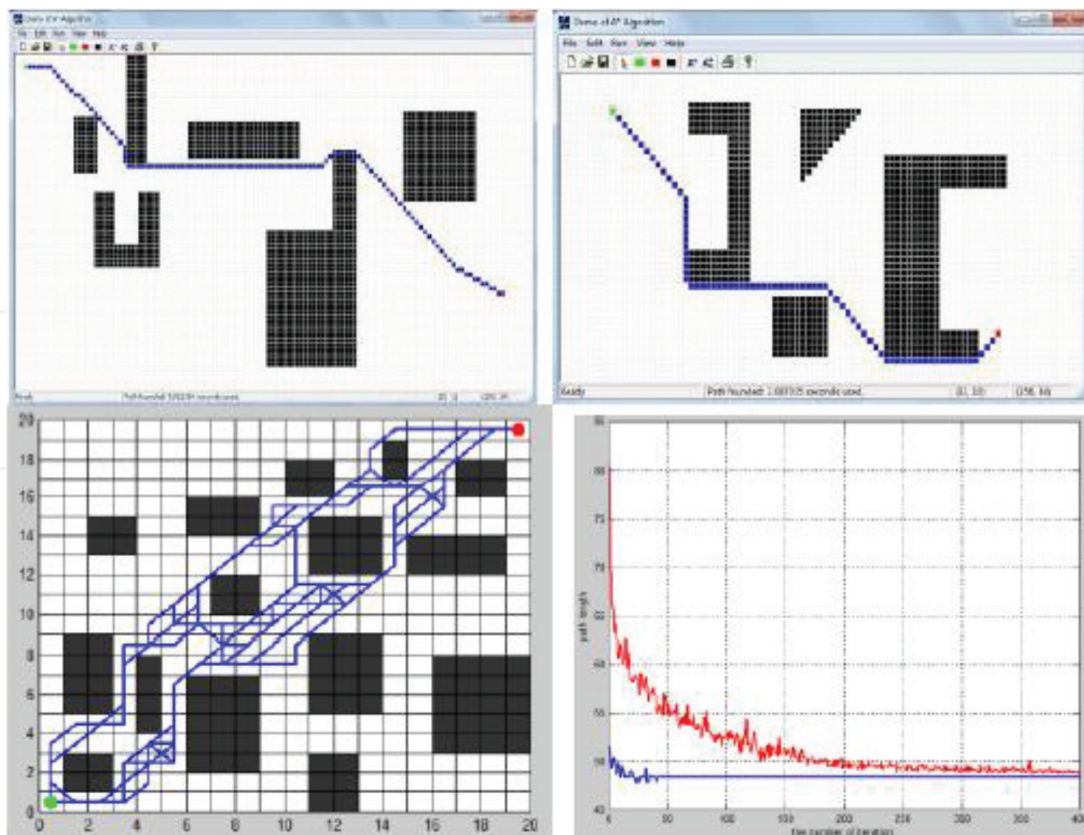**Figure 4.** Trajectory of the mobile robot.

Figure 5. Trajectory planning and simulation (1–3).

Point	x/m	y/m	t/ms	Field strength
1	1. 375	− 2. 328	13	6. 783
2	0. 845	− 1. 795	11	5. 439
3	0. 947	− 1. 330	15	4. 247
4	0. 694	− 1. 075	12	3. 703
5	0. 475	− 0. 612	19	2. 831
6	0. 504	− 0. 145	14	1. 932
7	0. 686	0. 118	9	1. 241
8	0. 822	0. 363	21	0. 981
9	1. 031	0. 807	12	0. 273
10	1. 217	1. 319	10	0. 001

Table 2. Algorithm implementation time and field strength.

3. Improved RRT* motion planning algorithm for mobile robot

Some path planning algorithms such as ant colony algorithm, genetic algorithm, artificial potential field algorithm, and dealing with some planning have its unique superiority. However, in the complex environment and high-dimensional space, the complex of the algorithm will increase sharply, which lead the convergence time too long to solve the problem. Because the algorithm based on potential field or heuristic function does not consider the kinematic and dynamic constraints, its result cannot be executed by the real case [12–17].

In order to solve the problem of high-dimensional path planning, the sampling algorithms have been introduced [18]. Compared with another advanced algorithm, the main advantage of the algorithm based on sampling is avoiding constructing the explicit configuration space, and it has been shown to be an effective solution to the path planning [19]. RRT algorithm based on sampling with rapid convergence rate can be used in unknown obstacle environment [20]. However, there are still some shortcomings in RRT algorithm [21–23]:

1. Its convergence rate is very slow in achieving the optimal solution;

2. Its memory requirements are significantly large due to a large number of iterations used to calculate the optimal path;

3. The rejection of samples, which might not be connectable directly with the existing nodes in the tree, but may lie closer to the goal region and hence could aid the algorithm in determining an optimal path much more rapid.

The domestic and foreign scholars have been studied on these deficiencies and proposed various RRT algorithms to adapt to different application scenarios. In order to improve the efficiency of the expansion of the node, Kuffner and La Valle proposed the RRT-connect [24]. Karaman and Frazzoli first proposed RRT* algorithm, which convergence to the optimal solution [25]. A.H. Qureshi and S. Mumtaz proposed the TG-RRT*, using triangular geometry to select node, in order to reduce the number of iterations required for the optimal solution, so as to make the algorithm rapid convergence [26]. C. Wouter Bac and Tim Roorda proposed by

RRT algorithm on ROS platform for mobile robot path planning research and applied to the motion planning problem in the dense obstacles environment [27].

3.1. Nonholonomic constraints for mobile robot

The integrity constraints include a system of generalized coordinates derivative and integral constraints. The mobile robot is a typical nonholonomic constraint system. The mobile robot's simplified moving model is shown in the **Figure 6**.

Mobile robot's state variables in configuration space (x, y, θ, v, ϕ), let (x, y) represent the center of mobile robot rear axle in the system coordinates, θ represent the angle between the forward direction of the mobile robot and the x-axis, φ represent the angle between the forward direction of the mobile robot and the front wheel direction, v as the speed of the mobile robot, due to the nonholonomic constraints, the wheel is point contact with the ground, and only pure roll no relative sliding at contact.

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = v \tan \phi / L \\ \dot{v} = u_0 \\ \dot{\phi} = u_1 \end{cases} \tag{4.1}$$

And the constraint equations:

$$\begin{pmatrix} dx/dt \\ dy/dt \\ d\theta/dt \\ dv/dt \\ d\phi/dt \end{pmatrix} = \begin{pmatrix} v \cos \theta \\ v \sin \theta \\ v \tan \phi / L \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_0 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_1 \tag{4.2}$$
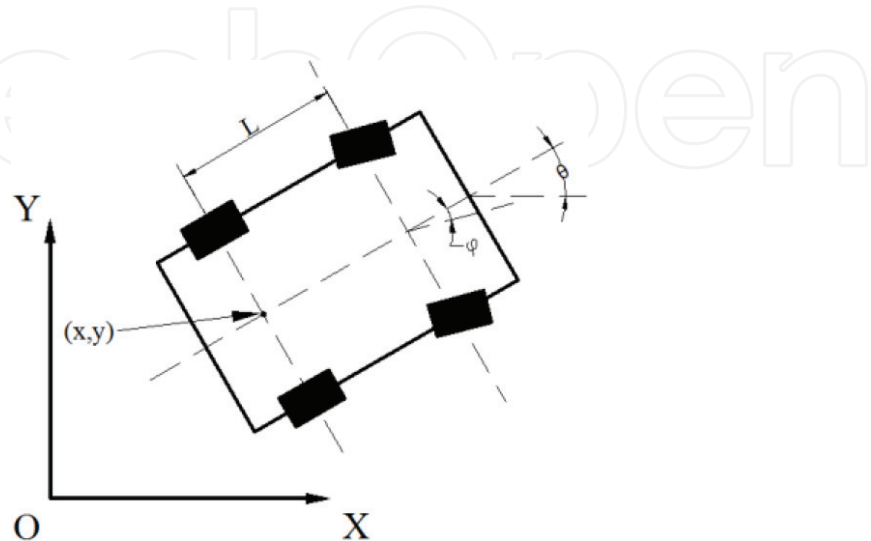

Figure 6. Geometrics of the mobile robot.

u_0 (acceleration) and u_1 (angular acceleration) of the mobile robot control variables. The minimum turning radius (maximum curvature) of the mobile robot is:

$$K_{\max} = \frac{\tan \phi_{\max}}{L} = \frac{1}{R_{\min}} \tag{4.3}$$

Therefore, it is unreasonable without considering the nonholonomic constraints of the robot planning algorithm.

3.2. Description and implementation of the I-RRT*

I-RRT* algorithm showed in **Table 3** is specifically designed for motion planning in complex, cluttered environments where exploration of configuration space is difficult. Let the sets of near vertices from the tree T_a and T_b be denoted by X_{near}^a and X_{near}^b, respectively. The path is connecting P_{init}^a and P_{rand} is denoted by $\sigma_a' : [0, s_a]$, while the path is connecting P_{init}^b and P_{rand} is denoted by $\sigma_b' : [0, s_b]$.

It starts by picking a random sample P_{rand} from the obstacle-free configuration space X_{free} that is, $x_{rand} \in X_{free}$. It then populates the set of near vertices, X_{near}^b for both trees using the *NearVertices* procedure. It should be noted that a ball region centered at P_{rand} of radius r is formed, and the sets of the near vertices from both trees are computed that is:

1. $V_a \leftarrow \{x_{init}^a\}; E_a \leftarrow \phi; T_a \leftarrow (V_a, E_a);$
2. $V_b \leftarrow \{x_{init}^b\}; E_b \leftarrow \phi; T_b \leftarrow (V_b, E_b);$
3. $\sigma_f \leftarrow \infty; E \leftarrow \phi;$
4. $Connection \leftarrow True$
5. $for \quad i \leftarrow 0 \quad to \quad N \quad do$
6. $x_{rand} \leftarrow Sample(i)$
7. $\{X_{near}^a, X_{near}^b\} \leftarrow NearVertices(x_{rand}, T_a, T_b)$
8. $if \quad X_{near}^a = \varphi \quad \&\& \quad X_{near}^b = \varphi \quad then$
9. $\{X_{near}^a, X_{near}^b\} \leftarrow NearestVertex(x_{rand}, T_a, T_b)$
10. $Connection \leftarrow False$
11. $L_s^a \leftarrow GetSortedList(x_{rand}, X_{near}^a)$
12. $L_s^b \leftarrow GetSortedList(x_{rand}, X_{near}^b)$
13. $\{x_{\min}, flag, \sigma_f\} \leftarrow GetBestTreeParent(L_s^a, L_s^b, Connection)$
14. $if(flag)then$
15. $T_a \leftarrow InsertVertex(x_{rand}, x_{\min}, T_a)$
16. $T_a \leftarrow RewireVertex(x_{rand}, x_{\min}, T_a)$
17. $else$
18. $T_b \leftarrow InsertVertex(x_{rand}, x_{\min}, T_b)$
19. $T_b \leftarrow RewireVertex(x_{rand}, x_{\min}, T_b)$
20. $E \leftarrow E_a \cup E_b$
21. $V \leftarrow V_a \cup V_b$
22. $return \quad (\{T_a, T_b\} = V, E)$

Table 3. The algorithm of I-RRT*.

$$X_{near}^a := \left\{ v \leftarrow V_a : v \in B_{x_{rand},r} \right\} \qquad (4.4)$$

$$X_{near}^b := \left\{ v \leftarrow V_b : v \in B_{x_{rand},r} \right\} \qquad (4.5)$$

In the case of both sets of near vertices being found empty, these sets are filled with the closest vertex from their respective trees instead. The procedure Best Selected Tree returns the nearest vertex on the Best Selected Tree, which is eligible to become the parent of the random sample.

Hence, unlike the connect heuristic, the I-RRT* is not greedy since the connection is only made inside the ball region. Finally, the tree connection generates the end-to-end global path.

3.3. Experimental results

3.3.1. The algorithm simulation

This section presents simulations performed on a 2.4 GHz Intel Core i5 processor with 4 GB RAM. Here, performance results of our I-RRT* algorithm are compared with RRT* and B-RRT*. For proper comparison, experimental conditions and size of the configuration space were kept constant for all algorithms. Since randomized sampling-based algorithms exhibit large variations in results, the algorithms were run up to 50 times with different seed values for each type of environment. Maximum, minimum, and an average number of iterations i as well as time t utilized by each algorithm to reach the optimal path solution is presented in **Table 4**.

In the second environment simulation, chooses some representative to verify the algorithm performance.

Figures 7–9 are three kinds of the optimization algorithm for the same problems. And we can see the I-RRT* is better than the others.

In the third environment model simulation test, using the three different kinds of obstacle environment simulation experiment, we tested the performance of the algorithm in **Figure 10**.

Environment	ALG	i_{min}	i_{max}	i_{avg}	t_{min}	t_{max}	t_{avg}	C	fail
3D-Random obstacles	RRT*	107,810	110,851	108,965	17.8	19.3	18.8	**	6
	B-RRT*	30,901	38,086	36,128	6.4	8.1	7.4	**	4
	I-RRT*	19,507	22,525	21,290	4.4	5.3	5.1	**	1
3D-Multiple barriers	RRT*	1,430,381	1,440,619	1,437,342	242.1	247.4	244.1	205.6	14
	B-RRT*	495,961	503,240	498,972	102.1	106.5	105.1	205.6	6
	I-RRT*	97,885	112,857	111,139	22.3	27.3	26.2	205.6	3
3D-Narrow passage	RRT*	1,277,376	1,301,698	1,290,674	216.9	221.9	218.5	345.1	9
	B-RRT*	533,276	561,347	551,771	110	117.1	115.9	345.1	3
	I-RRT*	127,363	143,806	134,421	30.3	35.2	31.4	345.1	0

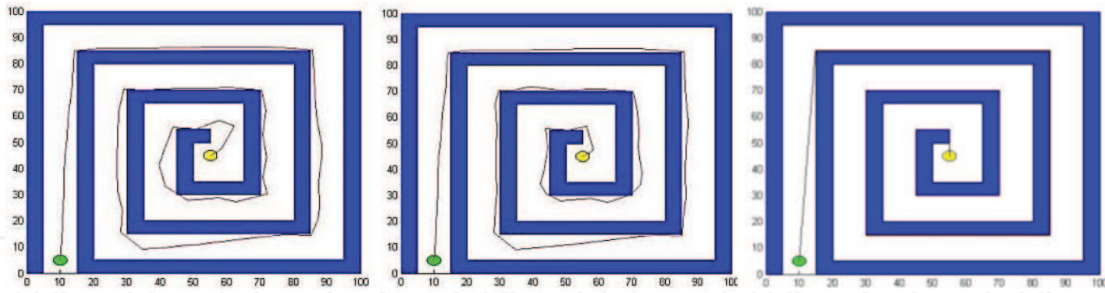**Table 4.** Experimental results for computing optimal path solution.

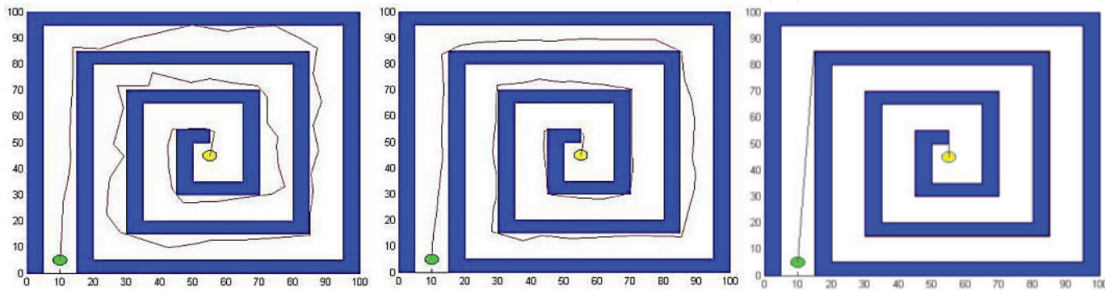**Figure 7.** RRT* performance in 2-D environment.

Figure 8. B-RRT* performance in 2-D environment.
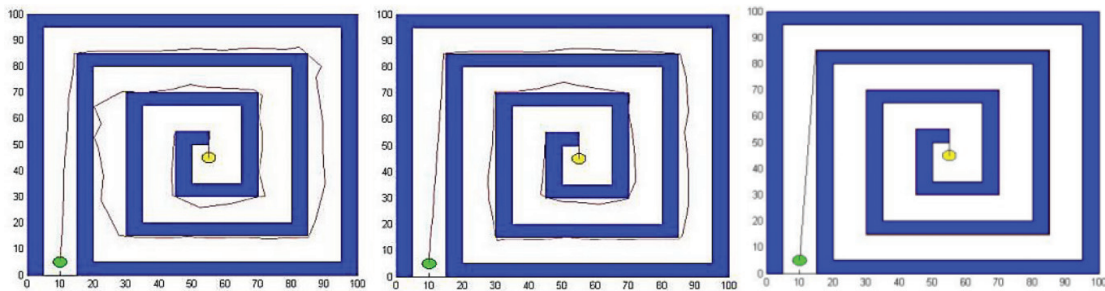

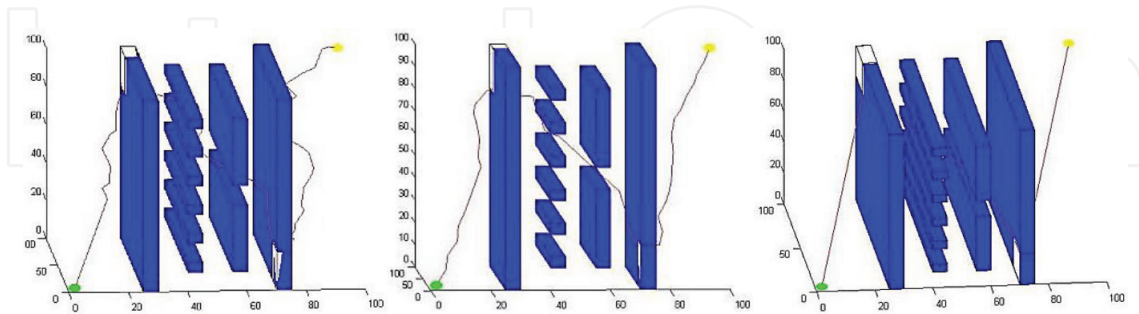**Figure 9.** I-RRT* performance in a 2-D environment.

Figure 10. I-RRT* performance in 3-D environment.

To restrain the computational time within reasonable limits, the maximum limit for the number of tree nodes was kept at 3 million.

Although both I-RRT* and B-RRT* were successful in finding the optimal solution, B-RRT* took an extremely large number of iterations to converge in comparison with I-RRT*. B-RRT*
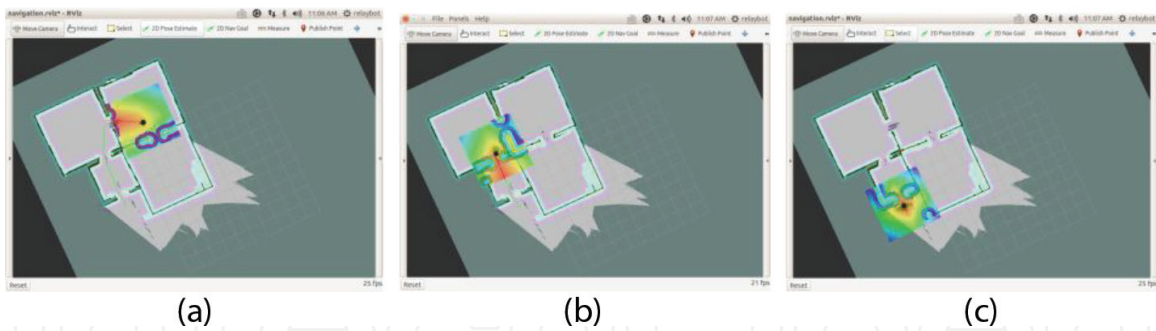
utilizes the partial greedy heuristic approach as discussed earlier, which significantly reduces its ability of convergence to the optimal path solution. **Table 4** shows the convergence from the initial path solution to the optimal path solution by I-RRT* and RRT*, respectively. For determination of the optimal path, the I-RRT* algorithm takes the least number of average iterations (iavg = 111,139) as compared to B-RRT*(iavg = 498,972) and the extraordinarily large number of iterations taken up by RRT*(iavg = 1,437,342).

3.4. Mobile robot simulation experiment on ROS

The simulation of the environment was established under the gazebo. In building a model, a mobile robot called ROS mapping packages build implementation scenario map in **Figures 11–13**.

(a)   (b)

Figure 11. The process of SLAM. (a) SLAM show under gazebo (b) SLAM show under RVIZ.

Figure 12. Map information in RVIZ.

Figure 13. The process of planning. (a) Start planning, (b) the process of planning, and (c) planning complete.

3.5. Static environment experiment

Using the I-RRT* algorithm for mobile robot path planning, we choose a random target on the map and set the position and posture in **Figure 13**. I-RRT* algorithm can generate a barrier-free path, as shown in the green feed, where the trajectory planning algorithm.

Figures show the I-RRT* dynamic planning process, which is real-time planning. I-RRT* algorithm programming device receives the message, if there are new obstacles, in order to avoid obstacles, I-RRT* real-time planning. In this picture perceived color area for mobile robot around obstacles of information, the green line is the initial path planning, the red line for real-time planning path. As a result, the algorithm can be applied to the dynamic environment of the path planning problem.

4. Conclusion

Solving the path planning problems of the ground robot and mobile robot are the premise of more efficient use of the robot. Due to the different types of robots, the problems in the path planning need to be considered, from 2D to 3D. Three path planning algorithms for different types of robots are proposed in this chapter. The problem of path planning for existing industrial robots and mobile robots, including unmanned vehicles and unmanned aerial vehicles, are solved. All the simulations results show that the algorithms proposed in this paper are feasible.

Acknowledgements

Author details

Xiangrong Xu*, Yang Yang and Siyu Pan

*Address all correspondence to: xuxr88@yahoo.com

School of Mechanical Engineering, Anhui University of Technology, Ma'anshan, Anhui, China

References

[1] Fu KS, Gonzales RC, Lee CS. Robotics: Control, Sensing, Vision, and Intelligence. Singapore: McGrawHill. Inc; 1987. ISBN:0070226253

[2] Jung D, Tsiotras P. On-line path generation for small unmanned aerial vehicles using B-spline path templates. AIAA Guidance, Navigation and Control Conference, AIAA. 2008, Vol. 7135. DOI: 10.2514/6.2008–7135

[3] Chandler P, Rasmussen S, Pachter M. UAV cooperative path planning. AIAA Guidance, Navigation, and Control Conference and Exhibit. 2000. pp. 1255-1265. DOI: 10.2514/6.2000-4370

[4] Oscar M, Ulises O, Roberto S. Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles. Expert Systems with Applications. July 2015;**42**(12):5177-5191

[5] Park M, Jeon J, Lee M. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. Proceedings of the 2001 IEEE International Symposium on Industrial Electronics, Pusan. 2001. pp. 1530-1535

[6] Kitamura Y, Tanaka T, Kishino F. 3-D path planning in a dynamic environment using an octree and an artificial potential field. Proceedings of the 1995 IEEE International Conference on Intelligent Robots and Systems. Piscataway: IEEE; 1995. 2474-2481

[7] Janabi S, Vinke D. Integration of the artificial potential field approach with simulated annealing for robot path planning. Proceedings of the 1993 IEEE International Symposium on Intelligent Control Piscataway: IEEE. 1993. 536-541

[8] Hsu CK. Variable structure control design for uncertain dynamic systems with sector nonlinearities. Automatica. 1998;**34**(4):505-508

[9] Ashiru I, Czarnecki C, Routen T. Characteristics of a genetic based approach to path planning for mobile robots. Journal of Network and Computer Applications, 1996;**19**(2):149-169

[10] Chen L, You B. Dynamic robot motion tracking and obstacle avoidance based on artificial potential field method. Control Theory and Application. 2007;**26**(4):8-10

[11] Vadakkepat P, Tan WK. Evolutionary artificial potential fields and their application in real time robot path. Proceedings of the 2000 IEEE Conference on Evolutionary Computation Evolutionary Computation. Piscataway: IEEE. 2000. 256-263

[12] Xu X, Shi DQ, Lu M, et al. Study on mechanical mechanics with a method for minimum-time path planning of robots in cartesian space. Advanced Materials Research. 2013;**703**: 181-185. DOI: 10.4028/www.scientific.net/AMR.703.181

[13] Pellazar MB. Vehicle route planning with constraints using genetic algorithms. Aerospace and Electronics Conference, 1994. NAECON 1994. Proceedings of the IEEE 1994 National. IEEE, 1994: 111-118. DOI: 10.1109/NAECON.1994.333010

[14] Duan H, Yu Y, Zhang X, et al. Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm. Simulation Modelling Practice and Theory. 2010;**18**(8): 1104-1115. DOI: 10.1016/j.simpat.2009.10.006

[15] Wang H, Liu YH, Chen W. Visual tracking of robots in uncalibrated environments. Mechatronics. 2012;**22**(4):390-397. DOI: 10.1016/j.mechatronics.2011.09.006

[16] He J, Hou Z. Ant colony algorithm for traffic signal timing optimization. Advances in Engineering Software. 2012;**43**(1):14-18. DOI: 10.1016/j.advengsoft.2011.09.002

[17] Xu X, Li Y, Yang Y, et al. A method of trajectory planning for Ground Mobile Robot based on ant colony algorithm. IEEE International Conference on Robotics and Biomimetics. IEEE, 2017:2117–2121. DOI: 10.1109/ROBIO.2016.7866642

[18] Elbanhawi M, Simic M. Sampling-based robot motion planning: A review. IEEE Access. 2014;**2**(1):56-77. DOI: 10.1109/ACCESS.2014.2302442

[19] Qureshi AH, Ayaz Y. Intelligent bidirectional rapidly exploring random trees for optimal motion planning in complex cluttered environments. Robotics and Autonomous Systems. 2015;**68**:1-11. DOI: 10.1016/j.robot.2015.02.007

[20] Kavraki L, Latombe JC. Randomized preprocessing of configuration for fast path planning. IEEE International Conference on Robotics and Automation, 1994. Proceedings 3, 2138-2145. IEEE. 1994. DOI: 10.1109/ROBOT.1994.350966

[21] Vonásek V, Saska M, Winkler L, Přeučil L. High-level motion planning for cpg-driven modular robots. Robotics & Autonomous Systems. 2015;**68**:116-128. DOI: 10.1016/j. robot.2015.01.006

[22] Doshi AA, Postula AJ, Fletcher A, Singh SPN. Development of micro-uav with integrated motion planning for open-cut mining surveillance. Microprocessors and Microsystems. 2015;**39**(8):829-835. DOI: 10.1016/j.micpro.2015.07.008

[23] Park KJ, Won M. People tracking and accompanying algorithm for mobile robot using kinect sensor and extended kalman filter. Transactions of the Korean Society of Mechanical Engineers A. 2014;**38**(4):345-354. DOI : 10.3795/KSME-A.2014.38.4.345

[24] Kuffner JJ, Lavalle SM. RRT-connect: An efficient approach to single-query path planning. IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA (Vol. 2, pp. 995-1001). 2000. IEEE. DOI: 10.1109/ROBOT.2000.844730

[25] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. International Journal of Robotics Research. 2011;**30**(7):846-894. DOI: 10.1177/0278364911406761

[26] Qureshi AH, Mumtaz S, Iqbal KF, Ayaz Y. Triangular geometry based optimal motion planning using RRT*-motion planner. IEEE, International Workshop on Advanced Motion Control. 2014:380-385. DOI: 10.1109/AMC.2014.6823312

[27] Bac CW, Roorda T, Reshef R, Berman S, Hemming J, Henten EJV. Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. Biosystems Engineering. 2016;**146**:85-97. DOI: 10.1016/j.biosystemseng.2015.07.004