Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3160

TRAJECTORY PLANNING OF ACUPUNCTURE ROBOTS

Yi Wang, DianHao Wu*, ZiChen Ma, KunLin Xie, MengDi Wang
School of Mechanical Engineering and Automation, University of Science and Technology Liaoning, Anshan 114051, Liaoning, China.
**Corresponding Author: DianHao Wu*

Abstract: For the clinical demand of sequential needling at multiple facial acupoints, needles already inserted and retained in the tissue become static obstacles to subsequent insertions. Planning a collision-free, smooth and efficient insertion path for each new needle inside the narrow facial workspace is therefore a key problem in the automation of acupuncture robots. This paper takes an acupuncture robot system composed of a UR5e six-degree-of-freedom manipulator and a self-developed acupuncture end effector as the research object, formulates multi-acupoint sequential needling as a constrained motion planning problem in which the obstacle set grows with the needling sequence, and proposes an improved APF-RRT* algorithm. The algorithm combines a bidirectional growth mechanism and goal-biased sampling on the RRT* framework to improve the convergence of the two trees, introduces Informed ellipsoidal subset sampling to reduce ineffective exploration, constructs an improved artificial potential field to alleviate the goal-unreachable and local-minimum problems, and designs a dynamic adaptive step-size strategy based on the ratio of attractive to repulsive force together with a bidirectional greedy path-pruning strategy. Simulation results in two- and three-dimensional environments show that, under the same maximum number of nodes, the proposed algorithm converges faster and yields a lower path cost than RRT* and APF-RRT*: in the 2D environment the runtime is reduced from 44.702 s to 9.538 s and the path cost from 792.702 to 750.477, while in the 3D environment the runtime is reduced from 84.594 s to 32.796 s and the path cost from 933.077 to 822.857, verifying the effectiveness and engineering applicability of the algorithm.
Keywords: Acupuncture robot; Trajectory planning; RRT*; Artificial potential field; Path pruning

1 INTRODUCTION

Acupuncture is an important component of traditional Chinese medicine, and facial acupoint selection is widely used in the treatment of facial palsy, hemifacial spasm, migraine and facial cosmesis. The conventional retained-needle method requires the practitioner to insert more than ten needles in sequence within a limited time and to select and manipulate the subsequent acupoints while the inserted needles remain in place. A large prospective practice-based survey of 37,490 consultations showed that the overall adverse events of acupuncture are dominated by mild bleeding, haematoma and pain, whereas the facial region has a dense distribution of blood vessels and nerves and only thin tissue, so improper manipulation readily causes ecchymosis and swelling [1]. Improving the positioning accuracy and operational efficiency of multi-acupoint needling while guaranteeing safety is therefore a practical problem in acupuncture practice.

In recent years, the combination of robotic technology with traditional Chinese medicine has attracted extensive attention. Visual servoing has been combined with force feedback to achieve hand-eye-force coordinated control of an acupuncture robot [2], and a force-controlled acupuncture robot has been developed that inserts needles in synchrony with the respiratory cycle so as to compensate for the patient's physiological motion [3]. Kinematic modelling and simultaneous calibration of acupuncture robots have also been studied [4]. These systems, however, are essentially single-insertion machines: they localise one acupoint, insert one needle and release it, so that the sequencing of a multi-acupoint session is still left to the practitioner.

For medical needle-insertion robots, research has focused mainly on needle-tissue interaction force modelling and insertion safety control [5]. The closest precedent to the problem addressed here is the path planner proposed by Aghdam and Liu for steerable bevel-tip needles that reach multiple targets in the presence of obstacles, motivated precisely by avoiding the tissue damage caused by repeated withdrawal and re-insertion between targets [6]. The handheld peripheral venipuncture robot developed by Leipheimer et al. achieved tip positioning accuracy on the order of 0.2 mm and stopped the insertion at the instant of puncture by force-peak detection, thereby avoiding penetration of the far vessel wall [7]. In these works, however, the obstacles are anatomical structures and the needle is generally assumed to be steerable or retractable; the obstacle formed by a rigid needle left in place after insertion has not been considered.

At the level of needle path planning, Zhang et al. applied an improved RRT* to three-dimensional path planning for flexible needles in multilayer tissue [8].

Regarding manipulator kinematics, Villalobos et al. provided a complete analytical inverse solution and singularity analysis for 6-DOF UR/TM-type manipulators [9]. Rather than repeating the derivation, this paper directly uses the closed-form inverse solution of the UR5e as the basis for the joint-space verification of path points and for selecting among multiple solutions.

As the most widely applied motion planning method in high-dimensional configuration spaces, the Rapidly-exploring Random Tree (RRT) was proposed by LaValle [10], and Kuffner and LaValle further developed the bidirectionally

© By the Author(s) 2026, under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0).

growing RRT-Connect [11]. The RRT* proposed by Karaman and Frazzoli achieves asymptotic optimality through parent reselection and rewiring [12], and the Informed RRT* proposed by Gammell et al. constrains sampling to an ellipsoidal subset focused on the start and the goal [13]. Fan et al. combined bidirectional growth, goal-biased sampling and an artificial potential field into a bidirectional APF-RRT* [14].

In summary, two gaps remain in the existing research. First, research on acupuncture robots has largely solved the localisation, needle insertion and force control of a single acupoint, but has not addressed the sequencing of multi-acupoint needling, and needles already inserted and retained have never been modelled as obstacles. Second, improved RRT* algorithms for manipulators mostly take static industrial geometry as obstacles, with the obstacle set fully determined before planning, whereas in multi-acupoint needling the obstacle set is progressively generated by the robot's own previous actions and the narrow facial workspace imposes stricter requirements on path clearance. To address these problems, this paper carries out research on a UR5e manipulator platform. The main work is as follows:

1) an acupuncture robot system composed of a UR5e 6-DOF manipulator, a Kinect V2 depth camera and a self-developed acupuncture end effector is constructed; the trajectory planning problem of multi-acupoint sequential needling is formulated as a constrained motion planning problem in which the obstacle set grows with the needling sequence;

2) an improved APF-RRT* algorithm is proposed that combines six strategies, namely bidirectional growth, goal-biased sampling, Informed ellipsoidal sampling, an improved artificial potential field, a dynamic adaptive step size and path pruning;

3) comparative simulation experiments are carried out in 2D and 3D environments to verify the advantages of the proposed algorithm in convergence speed and path quality.

2 SYSTEM COMPOSITION OF THE ACUPUNCTURE ROBOT

The acupuncture robot system studied in this paper consists of four parts: a personal computer, a Kinect V2 depth camera (Microsoft, USA), a six-degree-of-freedom manipulator (UR5e, Universal Robots, Odense, Denmark) and a self-developed acupuncture end effector. During operation, the Kinect V2 camera synchronously acquires colour and depth images of the patient's face and transmits them to the computer, as shown in Figure 1. The acupoint recognition module, developed in Python, detects facial acupoints in the colour image and resolves their three-dimensional coordinates in the camera frame by combining the depth information. On this basis, the trajectory planning module computes a collision-free insertion path for each needle. The manipulator control system is implemented on the ROS platform and converts the planning result into joint-space motion commands. The human-machine interface is developed with Qt and allows the practitioner to select the acupoint prescription and monitor the needling process in real time.

The UR5e is a six-degree-of-freedom serial manipulator with a working radius of 850 mm, a rated payload of 5 kg at the end effector and a repeat positioning accuracy of 0.03 mm, and each joint integrates a torque sensor, which meets the requirements of facial needling on positioning accuracy and contact safety. The link geometry is described here by the standard D-H parameter method, with the origin of the base frame defined at the centre of the lower mounting surface and the tool frame defined at the acupuncture tip. The resulting spatial joint coordinate frames are shown in Figure 1. Because UR-series manipulators have a spherical wrist, their inverse kinematics admits an analytical solution, so that all joint-angle combinations can be obtained directly for a given end-effector pose; in the trajectory planning of this paper this closed-form inverse solution is used as the basis for the feasibility check of path points and for selecting among multiple solutions [9].



Figure 1 Conceptual Model of the Acupuncture Robot System

3 TRAJECTORY PLANNING PROBLEM AND THE RRT* ALGORITHM

3.1 Trajectory Constraints of Multi-Acupoint Sequential Needling

In acupuncture treatment, most conditions require several acupoints to be needled in sequence so as to produce synergistic stimulation and enhance the overall therapeutic effect. Taking the conventional treatment of facial palsy as an example, the physician usually needs to select acupoints sequentially on one or both sides of the face, such as Dicang (ST4), Jiache

(ST6), Sibai (ST2) and Chengjiang (CV24). In clinical needling practice, the retained-needle method is commonly used, that is, after a needle is inserted into a certain acupoint it is left in place for a period of time, during which the physician may turn to needle other acupoints. This process involves sequential movement and manipulation between different acupoints, and it requires the accurate localisation and needling of multiple acupoints within a limited time, which imposes practical demands on the physician's operating efficiency and step planning.

When an acupuncture robot replaces the physician in performing the continuous multi-acupoint needling task, its motion planning can be divided into two stages. The first stage is the initial positioning, in which the end effector of the manipulator is controlled to move to the first acupoint; at this stage the trajectory planning does not need to take the needles into account as obstacles. The second stage is the subsequent needling, in which, as the robot inserts needles into the subsequent acupoints in turn, the needles already inserted and retained in the body become static obstacles, and the trajectory planning of the manipulator must avoid these existing needle bodies while positioning the new acupoint. Together these two stages constitute a planning sequence that proceeds from obstacle-free positioning to obstacle-avoiding operation, and they represent a key technical problem for acupuncture robots in achieving automatic needling.

3.2 The RRT* Algorithm and Its Limitations

RRT is a sampling-based motion planning algorithm. By iteratively sampling and extending in the configuration space it builds a random tree that grows from the start towards the goal region, thereby avoiding the difficulty of explicitly constructing configuration-space obstacles in high-dimensional spaces [10]. RRT-Connect grows two trees simultaneously from the start and the goal and extends them towards each other, which markedly shortens the time needed to find an initial feasible solution in a single query [11]. On this basis RRT* introduces two local optimisation steps, parent reselection (ChooseParent) and rewiring (Rewire), which continuously reduce the cumulative path cost of the nodes in the tree and thereby give the algorithm asymptotic optimality: as the number of samples tends to infinity, the cost of the resulting path converges to the optimum with probability one [12].

Although RRT* is asymptotically optimal, it still has clear shortcomings in engineering applications. First, sampling is entirely random, which produces a large amount of ineffective exploration in open regions, so that the initial feasible solution appears late and convergence is slow. Second, nodes are extended with a fixed step size: an excessively large step easily crosses narrow passages or causes collision-detection failures at obstacle boundaries, whereas an excessively small step sharply increases the number of nodes and reduces planning efficiency. Third, although Informed RRT* can constrain sampling to an ellipsoidal subset, it still requires an initial path to be found first [13]. Fourth, in narrow spaces, random sampling alone provides insufficient drive for the tree to grow towards the goal, so the tree tends to wander repeatedly in local regions.

It should be noted that multi-acupoint sequential needling imposes stricter requirements on the planner than conventional manipulator obstacle avoidance. The workspace is confined to the narrow facial region, and the obstacles are slender needles less than 0.3 mm in diameter whose clearance requirement is small but must be strictly guaranteed. Moreover, any planning failure means that the manipulator may touch a needle already inserted into the patient, so the algorithm must converge reliably within a limited number of nodes. For these reasons, this paper introduces several improvement strategies on the RRT* framework so as to guarantee path safety and smoothness while improving the convergence speed.

4 IMPROVED APF-RRT* TRAJECTORY PLANNING ALGORITHM

To address the slow convergence, strong sampling randomness and poor adaptability to a fixed step size of RRT* in trajectory planning, this paper proposes an improved APF-RRT* algorithm with coordinated multiple strategies. On the basis of conventional RRT*, the algorithm combines a bidirectional growth mechanism with a goal-biased strategy to improve the convergence of the two random trees and to enhance the search guidance; adopts an Informed sampling strategy that constrains the sampling range according to the cost of the current feasible trajectory, thereby reducing the blindness of sampling; incorporates an improved artificial potential field into the extension of new nodes so as to guide the nodes around obstacles and towards the goal; introduces a dynamic adaptive step-size mechanism that adjusts the extension step according to the attractive force, the repulsive force and the distance to the nearest obstacle; and finally removes redundant nodes from the path by bidirectional greedy pruning. The design of each strategy is described below.

4.1 Bidirectional Growth Mechanism

Bidirectional RRT* forms a parallel search architecture by building two random trees that are respectively rooted at the start point q_{start} (denoted T_a) and the goal point q_{goal} (denoted T_b). Its core strategy is to drive the two trees to grow towards each other in the configuration space and finally to connect them, thereby halving the blindness of the search and shortening the time required to find an initial path exponentially. The schematic diagram of its expansion is shown in Figure 2.

The core of the algorithm is a loop of alternating expansion and active connection attempts. In each iteration, a standard RRT* expansion process is first performed on one tree (e.g. T_a) to execute the standard RRT* expansion process and generate a new node q_{new}. Subsequently, the algorithm immediately attempts to connect the other tree T_b to this node,

that is, to search in T_b for the nearest node $q'_{nearest}$, and to check whether the connection condition is satisfied: the Euclidean distance between the two points is less than the preset connection threshold d_c (usually comparable to or slightly larger than the step size ε), and the connecting path is collision-free. This condition is formalised as:

$$\left\| q_{new} - q'_{nearest} \right\| < d_c \ \ \& \ \ CollisionFree\left(q_{new}, q'_{nearest} \right) \tag{1}$$

In the equation, the distance is usually taken as the Euclidean distance. If the condition is satisfied, the connection is realised through the edge $\left(q_{new}, q'_{nearest} \right)$. After one iteration is completed, the roles of the two trees are exchanged (T_a and T_b are swapped) so as to ensure a balanced search.

After a successful connection, the algorithm extracts a complete path composed of three segments by backtracking: the path from the start point along T_a to q_{new}, the connecting edge $\left(q_{new}, q'_{nearest} \right)$, and the path from $q'_{nearest}$ along T_b to the goal point. Thereafter, the two trees can be merged into a single unified random tree, and on this basis the standard RRT* optimisation loop of reselecting the parent node and rewiring is continued, so that the spliced path is progressively optimised and its cost decreases continuously until convergence.

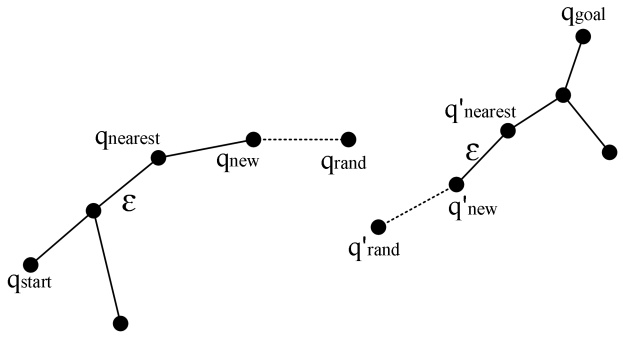


Figure 2 Schematic Diagram of Bidirectional RRT*

4.2 Goal-Biased Sampling Strategy

In sampling-based path planning algorithms, goal bias is a key heuristic strategy intended to guide the growth direction of the random tree, so as to overcome the blindness of purely random sampling when exploring open spaces and thereby significantly accelerate the convergence of the algorithm.

The core of goal bias is the improvement of the sampling function $Sample(\ )$. Standard RRT/RRT* performs uniform random sampling in the configuration space, that is, any point in free space has an equal probability of being sampled. Under the goal bias strategy, random sampling is performed as given in Eq. (2).

$$q_{rand} = \begin{cases} rand \cdot size(map), & rand > p \\ q_{goal}, & \text{otherwise} \end{cases} \tag{2}$$

In the equation, $rand$ is a random number between 0 and 1, p is the preset goal bias threshold, and $size(map)$ is the size of the space. If the random number $rand$ is greater than the goal bias threshold, the sampling point takes a random position; otherwise, the sampling point is taken as the goal point.

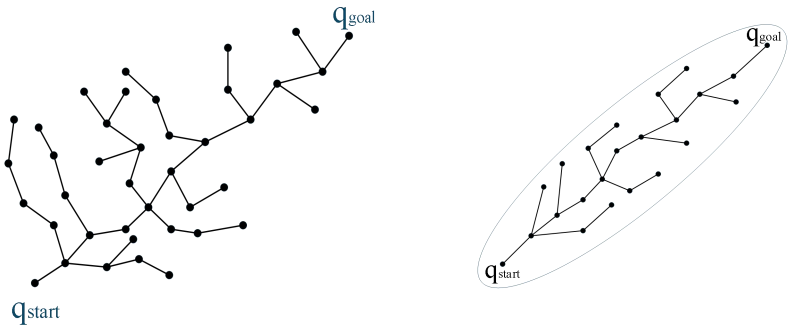
4.3 Informed Sampling Strategy

The core of the Informed RRT* sampling strategy lies in constraining the subsequent global random sampling to a specific elliptical subspace once an initial path has been obtained. The ellipse takes the start point q_{start} and the goal point g_{goal} as its foci, and the length of its major axis is equal to the cost of the current optimal path C_{best}. Any path node that may be better than the current solution must lie within this elliptical region, as expressed in Eq. (3).

$$\left\| q_{rand} - q_{start} \right\|_2 + \left\| q_{rand} - q_{goal} \right\|_2 \le C_{best} \tag{3}$$

The implementation of this strategy is divided into two stages. The first stage is initial exploration, whose behaviour is consistent with standard RRT: uniform random sampling is carried out in the global space until the first feasible path is found and its cost C_{best} is recorded. The second stage is focused optimisation, in which the sampling function is switched to uniform random sampling only within the above elliptical region. As the iterative optimisation reduces C_{best}, the elliptical region contracts accordingly, and the sampling process adaptively focuses on the region where better solutions may exist. This strategy directly avoids sampling in invalid regions and improves the convergence efficiency.

Its effect is shown in Figure 3: compared with the global scattered sampling of traditional RRT, the nodes produced by Informed sampling exhibit an obvious spatial orientation.



(a) (b)

Figure 3 Schematic Diagram of Node Expansion with the Informed Sampling Strategy: (a) Schematic Diagram of RRT* Expansion; (b) Schematic Diagram of Node Expansion with the Informed Sampling Strategy

4.4 Improved Artificial Potential Field Guidance

By introducing the artificial potential field (APF) method into RRT*, an attractive field is constructed for the goal point and a repulsive field for the obstacles; the robot is regarded as a particle moving in the field, and its direction of motion is determined by the resultant force acting on it, so that it is guided to reach the goal without collision. The APF method involves an attractive field $U_{att}(q)$ and a repulsive field $U_{rep}(q)$, which are given in Eq. (4) and Eq. (5) respectively.

$$U_{att}(q)=\frac{1}{2}\xi d^2\left(q,q_{goal}\right) \tag{4}$$

$$U_{rep}(q)=\begin{cases}\frac{1}{2}\eta\left(\frac{1}{d\left(q,q_{obs}\right)}-\frac{1}{d_0}\right)^2, & if\ d\left(q,q_{obs}\right)\le d_0\\ 0, & if\ d\left(q,q_{obs}\right)>d_0\end{cases} \tag{5}$$

$$U(q)=\sum U_{att}(q)+\sum U_{rep}(q) \tag{6}$$

In Eq. (4), ξ is the attractive gain coefficient, and $d\left(q,q_{goal}\right)$ is the Euclidean distance from the current expanded node $q_{nearest}$ to the goal point q_{goal}. In Eq. (5), η is the repulsive gain coefficient, and $d\left(q,q_{obs}\right)$ is the Euclidean distance from the current expanded node $q_{nearest}$ to the nearest obstacle q_{obs}, and d_0 is the influence radius of the obstacle. In Eq. (6), $U(q)$ is the total potential field. Since the negative gradients of the attractive and repulsive functions are the attractive and repulsive forces respectively, the magnitudes of the attractive force, the repulsive force and the resultant force can be obtained directly as given in Eq. (7), Eq. (8) and Eq. (9).

$$F_{att}(q)=-\Delta U_{att}(q)=\xi d\left(q,q_{goal}\right) \tag{7}$$

$$F_{rep}(q)=\begin{cases}\eta\left(\frac{1}{d\left(q,q_{obs}\right)}-\frac{1}{d_0}\right)\frac{1}{d^2\left(q,q_{obs}\right)}\frac{\partial d\left(q,q_{obs}\right)}{\partial q}, & if\ d\left(q,q_{obs}\right)\le d_0\\ 0, & if\ d\left(q,q_{obs}\right)>d_0\end{cases} \tag{8}$$

$$F(q)=-\nabla U(q)=\sum F_{att}(q)+\sum F_{rep}(q) \tag{9}$$

When APF is integrated into RRT*, and expansion is carried out from the expanded node $q_{nearest}$, the virtual resultant force $F(q)$ acting on it is the vector sum of the attractive force of the random sampling point, the attractive force of the goal point and the repulsive forces of the obstacles, and the direction of this resultant force determines the expansion direction of the new node q_{new}. In order to solve the local minimum problem caused by the mutual cancellation of the attractive and repulsive forces at a certain point, as well as the problem that the goal becomes unreachable because the repulsive force of obstacles near the goal point is too strong, this section makes the following improvements to APF. A distance function with respect to the goal point is introduced into the repulsive function, so that the repulsive force decreases when the end effector approaches an obstacle, as given in Eq. (10).

$$U_{rep}(q)=\begin{cases}\frac{1}{2}\eta\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)^2 d(q,q_{goal})^n, d(q,q_{obs})\le d_0\\ 0, \qquad d(q,q_{obs})>d_0\end{cases} \tag{10}$$

Then the obstacle repulsive force is as expressed in Eq. (11):

$$F_{rep}(q)=\eta\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)\frac{d(q,q_{goal})^n}{d(q,q_{obs})^2} \tag{11}$$

The attractive functions of the goal point and of the random point are given in Eq. (12) and Eq. (13) respectively, while the total attractive function acting on the end effector is given in Eq. (14).

$$U_{att1}(q)=\frac{1}{2}\xi\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)^2 d^n(q,q_{goal}) \tag{12}$$

$$U_{att2}(q)=\frac{1}{2}\xi\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)^2 d^n(q,q_{rand}) \tag{13}$$

$$U_{att}(q)=U_{att1}(q)+U_{att2}(q) \tag{14}$$

In Eq. (12), $d(q,q_{goal})$ is the Euclidean distance from the current expanded node $q_{nearest}$ to the goal point q_{goal}. In Eq. (13), $d(q,q_{rand})$ is the Euclidean distance from the current expanded node $q_{nearest}$ to the random point q_{rand}. $U_{att}(q)$ is the total attractive force. Then the attractive forces of the goal point and of the random point are expressed as:

$$F_{att1}=\frac{n}{2}\xi\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)^2 d^2(q,q_{goal}) \tag{15}$$

$$F_{att2}=\frac{n}{2}\xi\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)^2 d^2(q,q_{rand}) \tag{16}$$

$$F_{att}(q)=F_{att1}(q)+F_{att2}(q) \tag{17}$$

In the equations, $F_{att1}(q)$ is the attractive force exerted by the goal point on the node $q_{nearest}$, $F_{att2}(q)$ is the attractive force exerted by the random point on the node $q_{nearest}$, and $F_{att}(q)$ is the sum of the attractive forces acting on the node $q_{nearest}$. The direction of the resultant force is obtained by the parallelogram law, and the new node q_{new} is then expanded along this direction. The schematic diagram of the process is shown in Figure 4.

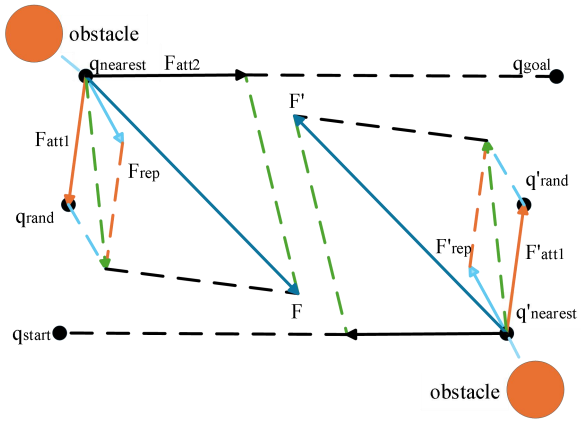


Figure 4 Schematic Diagram of Improved Bidirectional APF-RRT* Expansion

4.5 Dynamic Adaptive Step Size Strategy

Aiming at the problems of low search efficiency and slow convergence caused by the fixed step size used for node expansion in the traditional APF-RRT* algorithm, this section proposes a dynamic adaptive step size strategy. This strategy comprehensively considers the strength of the attractive field and of the repulsive field at the current node as well as the distance between the node and the nearest obstacle, so as to achieve dynamic adjustment of the step size and thereby improve the search efficiency and safety of the algorithm.

Based on the dynamic variation characteristics of the attractive and repulsive fields in the improved artificial potential field method, as well as the safety constraint on the distance between the node and obstacles, this paper designs the dynamic adaptive step size function as follows:

$$\begin{cases} \varepsilon = \left(1-\left(1-\dfrac{F_{att_now}}{F_{att_max}}\right)^2\right)\times\varepsilon_{now}\times lg\left(F_{att_now}\right), d\left(q,q_{obs}\right)\geq\varepsilon_{new} \\ \varepsilon = \left(1-\left(1-\dfrac{F_{att_now}}{F_{att_max}}\right)^2\right)\times\varepsilon_{now}\times e^{-F_{rep_now}}, \quad d\left(q,q_{obs}\right)<\varepsilon_{new} \end{cases} \tag{18}$$

In the equation, ε_{new} is the step size of the parent node of the current node, F_{att_now} is the attractive force acting on the current expanded node $q_{nearest}$, F_{att_max} is the attractive force acting on the start point from the goal point, and F_{rep_now} is the repulsive force acting on the current expanded node $q_{nearest}$. The dynamic step size function is regulated by an adjustment coefficient under different distances between the current node and the nearest obstacle. When $d\left(q,q_{obs}\right)\geq\varepsilon_{new}$, that is, when the distance between the current node and the nearest obstacle is greater than the current step size, the end effector is in a relatively safe region, so the adjustment coefficient adopts a logarithmic $lg\left(F_{att_now}\right)$ form that grows relatively slowly, making the step size increase steadily as the attractive force increases and avoiding excessive aggressiveness in the safe region. When $d\left(q,q_{obs}\right)<\varepsilon_{new}$, that is, when the distance between the current node and the nearest obstacle is less than the current step size, the end effector is close to the obstacle, and the adjustment coefficient adopts an exponential $e^{-F_{rep_now}}$ form that decreases sharply as the repulsive force increases, making the step size decrease significantly with the repulsive force so as to guarantee safe obstacle avoidance.

4.6 Path Pruning Optimisation Strategy

In the initial trajectory planning stage, owing to the random sampling characteristics of the APF-RRT* algorithm, the generated initial trajectory often contains a large number of redundant intermediate nodes. These redundant nodes not only increase the path length but also cause frequent acceleration and deceleration during the motion of the manipulator, reducing the execution efficiency and smoothness. Therefore, it is necessary to optimise the topological structure of the path by pruning while ensuring the safety of the trajectory. The schematic diagram is shown in Figure 5, and the specific procedure is as follows:

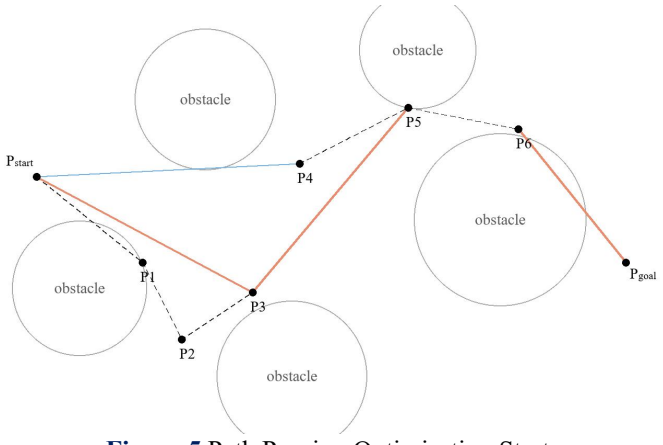


Figure 5 Path Pruning Optimisation Strategy

Step 1: The initial path generated by the improved APF-RRT* algorithm is P_{start} - P_1 - P_2 - P_3 - P_4 - P_5 - P_6 - P_{goal};

Step 2: Set P_{start} as the initial point and P_1 as the node in the initial path that is connected to P_{start}; connect P_{start} and P_2, and check whether the straight line P_{start} - P_2 contacts an obstacle. If it does not, remove the P_1 node from

the initial path and keep the straight line P_{start} - P_2 as the new path; then connect P_{start} and P_3 and check whether the straight line P_{start} - P_3 contacts an obstacle, and so on. When a straight line contacts an obstacle, for example when the straight line P_{start} - P_4 contacts an obstacle, the path P_{start} - P_3 is retained;

Step 3: Set the node P_3 as the initial point, repeat Step 2, check whether a new path exists, and update the initial path;

Step 4: Repeat Step 3 until the node P_{goal} is connected, and the final path is P_{start} - P_3 - P_5 - P_{goal}.

4.7 Overall Algorithm Flow

The pseudocode of the improved APF-RRT* algorithm is shown below. The algorithm initialises two rapidly exploring random trees respectively at the start point q_{start} and the goal point q_{goal}, namely T_a and T_b. First, the current optimal path length c_{best} is set to infinity, and the straight-line distance between the start point and the goal point is calculated as c_{min}. The $informed$ flag is initialised to $false$ (Line 2). The main loop is entered, and the loop counter i is incremented (Lines 3-5). The main loop is entered and the loop counter i is incremented (Lines 3-5). According to the parity of i, the current expansion tree T and the other tree T_{other} are selected alternately (Lines 6-7). In the sampling stage, if $informed$ is $true$ (that is, a feasible path has already been found), the $InformedSample$ function is called to perform uniform sampling within the ellipsoid whose foci are the start point and the goal point and whose major axis length is c_{best}, yielding the sampling point q_{rand}; otherwise the $Sample$ function is called to sample randomly in the whole free space (Lines 8-9). The $Nearest$ function is used to find, in the current tree T, the node q_{rand} nearest to q_{near} (Line 10), and then the $SteerAPF$ function is used to incorporate the artificial potential field method: while expanding towards the random point, the node is subject to the combined action of the attractive field of the goal and the repulsive fields of the obstacles, and an adaptive step size is adopted to generate the new node q_{new} (Line 11). After that, collision detection is performed between q_{near} and q_{new} (Line 12). If there is no collision, the new node q_{new} is added to the current tree T (Line 13). Subsequently, every K iterations an attempt is made to connect the two trees: if there exist nodes $n_a \in T_a$ and $n_b \in T_b$ satisfying Euclidean distance ≤ threshold τ and the line segment between the two points is collision-free (Lines 15-16), the $BuildPath$ function is called to generate a complete path by backtracking $path$ (Line 17). If the length of this path is less than the current optimal path length c_{best}, then c_{best} and the optimal path best are updated, and the $informed$ flag is set to $true$, after which sampling enters the $informed$ optimisation stage (Lines 18-21). After the main loop ends, if a feasible path has been found (best is not empty), the $PrunePath$ function is called to perform bidirectional pruning optimisation on the path: starting from the start point, the algorithm jumps as directly as possible to the farthest collision-free point, then performs the same operation once in reverse from the goal point, and takes the shorter of the two as the final path and returns it (Lines 28-29); otherwise it returns a path search failure (Line 31) (Table 1).

Table 1 Improved APF-RRT* Pseudocode

1. $T_a \leftarrow \{q_{start}\}$; $T_b \leftarrow \{q_{goal}\}$;
2. $c_{best} \leftarrow \infty$; $c_{min} \leftarrow \|q_{goal} - q_{start}\|$; $informed \leftarrow false$;
3. $i \leftarrow 0$;
4. **while** $I \le N$ **do**
5. $i \leftarrow i+1$;
6. **if** $i \ mod \ 2 = 1$ **then** $(T, T_{other}) \leftarrow (T_a, T_b)$;
7. **else** $(T, T_{other}) \leftarrow (T_b, T_a)$;
8. **if** $informed$ **then** $q_{rand} \leftarrow InformedSample(q_{start}, q_{goal}, c_{best}, c_{min})$;
9. **else** $q_{rand} \leftarrow Sample(T)$;
10. $q_{near} \leftarrow Nearest(T, q_{rand})$;

$1.\ T_a \leftarrow \{q_{start}\};\ T_b \leftarrow \{q_{goal}\};$

$11.\quad q_{new} \leftarrow SteerAPF\left(q_{near}, q_{rand}, q_{goal}, obstacles\right);$

$12.\quad \textbf{if } CollisionFree\left(q_{near}, q_{new}\right) \textbf{ then}$

$13.\quad T.Add\left(q_{new}\right);$

$14.\quad \textbf{if } i \bmod K = 0 \textbf{ then}$

$15.\quad \textbf{if } \exists\ n_a \in T_a, n_b \in T_b \textbf{ with } \left\|na - nb\right\| \le \tau;$

$16.\quad \textbf{and } CollisionFree\left(q_{near}, q_{new}\right) \textbf{ then}$

$17.\quad path \leftarrow BuildPath\left(T_a, T_b, n_a, n_b\right);$

$18.\quad \textbf{if } Length\left(path\right) < c_{best} \textbf{ then}$

$19.\quad c_{best} \leftarrow Length\left(path\right);$

$20.\quad best \leftarrow path;$

$21.\quad informed \leftarrow true;$

$22.\quad \textbf{end}$

$23.\quad \textbf{end}$

$24.\quad \textbf{end}$

$25.\quad \textbf{end}$

$26.\ \textbf{end}$

$27.\ \textbf{if } best = \emptyset \textbf{ then}$

$28.\quad best \leftarrow PrunePath\left(best, obstacles\right);$

$29.\quad \textbf{return } best$

$30.\ \textbf{else}$

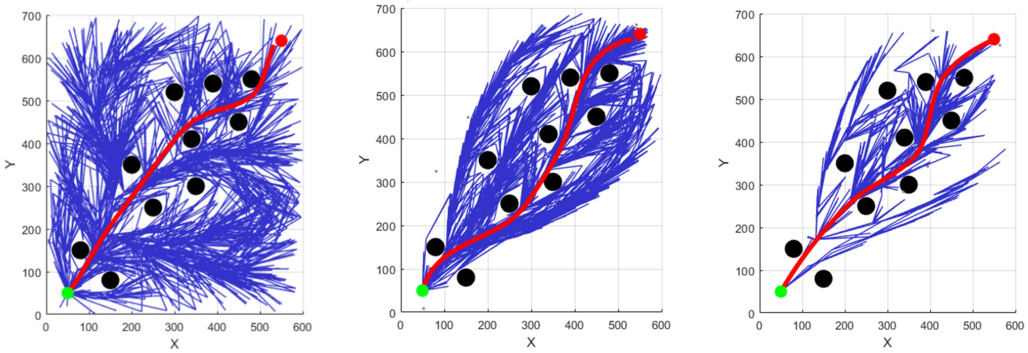
$31.\quad \textbf{return } Failure$

5 SIMULATION AND ANALYSIS

5.1 Two-Dimensional Environment Simulation

To verify the effectiveness of the improved APF-RRT* algorithm proposed in this paper, this section compares the performance of RRT*, APF-RRT* and the improved algorithm of this paper in a two-dimensional environment and a three-dimensional environment respectively.

First, the two-dimensional simulation environment is set as the XOY plane with a range of 600 mm x 700 mm; the coordinates of the start point are (50, 50) and those of the goal point are (550, 640). The coordinates of the obstacles are set to (80, 150), (200, 350), (400, 200), (480, 550), (150, 80), (300, 450), (350, 300), (420, 450), (220, 200) and (410, 510), the radius is set to 20 mm, and the maximum number of nodes is 1000. The results are shown in Figure 6.



RRT* APF-RRT* Improved APF-RRT*

Figure 6 Comparison of Simulation Results in the 2D Environment

5.2 Three-Dimensional Environment Simulation

Second, the three-dimensional simulation environment is 600 mm x 700 mm x 500 mm; the coordinates of the start point are (50, 50, 100) and those of the goal point are (550, 640, 100). The (x, y) coordinates of the obstacles are the same as those in the two-dimensional environment, their heights are all 200 mm, their radius is 20 mm, and the maximum number of nodes is 2000 (Figure 7).

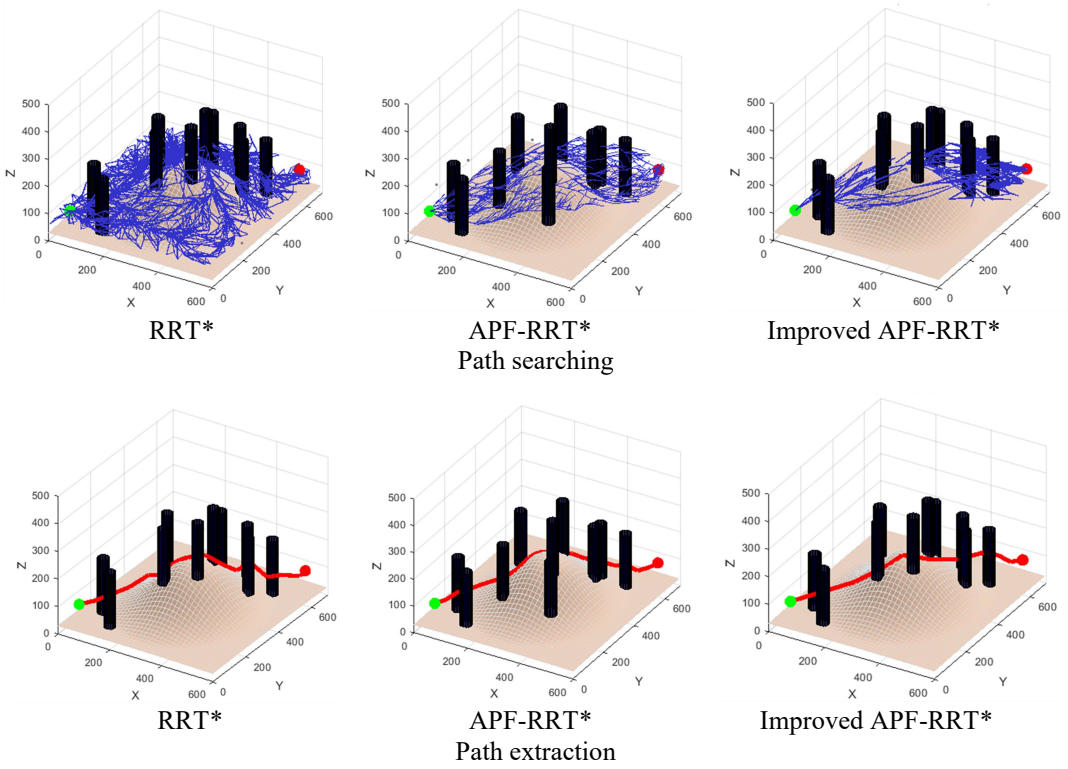


Figure 7 Comparison of Simulation Results in the 3D Environment

5.3 Comparison and Analysis of Simulation Results

In the two-dimensional and three-dimensional environments, the simulations of RRT*, APF-RRT* and the improved algorithm of this paper are carried out 10 times each, and the average results of the simulation data are shown in Table 2.

Table 2 Comparison of Simulation Results

Environment	Algorithm	Runtime (s)	Path cost
2D	RRT*	44.702	792.702
	APF-RRT*	29.798	775.796
	Improved APF-RRT*	9.538	750.477
3D	RRT*	84.594	933.077
	APF-RRT*	45.591	877.434
	Improved APF-RRT*	32.796	822.8568

6 CONCLUSIONS

This paper addresses the trajectory planning problem of facial multi-acupoint sequential needling on a UR5e manipulator platform, in which needles already inserted act as a growing obstacle set. The main conclusions are as follows.

1) An acupuncture robot system composed of a UR5e six-degree-of-freedom manipulator, a Kinect V2 depth camera and a self-developed acupuncture end effector was constructed, and the technical route of using the closed-form inverse solution for the joint-space verification of path points was established.

2) Multi-acupoint sequential needling was formulated as a constrained motion planning problem in which the obstacle set grows with the needling sequence, which makes explicit the modelling premise that an inserted needle is a static obstacle.

3) An improved APF-RRT* algorithm was proposed that combines bidirectional growth, goal-biased sampling, Informed ellipsoidal sampling, an improved artificial potential field, a dynamic adaptive step size and bidirectional greedy pruning.

4) The simulation results show that, under the same maximum number of nodes, the proposed algorithm reduces the runtime in the 2D environment from 44.702 s to 9.538 s and the path cost from 792.702 to 750.477, and reduces the runtime in the 3D environment from 84.594 s to 32.796 s and the path cost from 933.077 to 822.857, so that both the convergence speed and the path quality are better than those of RRT* and APF-RRT*. Future work will carry out facial multi-acupoint sequential needling experiments on a physical acupuncture robot platform and will further introduce insertion-force constraints and compensation for the patient's respiratory motion.

COMPETING INTERESTS

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

REFERENCES

[1] Won J, Lee J H, Bang H, et al. Safety of acupuncture by Korean Medicine Doctors: a prospective, practice-based survey of 37,490 consultations. BMC Complementary Medicine and Therapies, 2022, 22(1): 300.
[2] Su J, Zhu Y, Zhu M. Hand-eye-force coordination of acupuncture robot. IEEE Access, 2019, 7: 82154-82161.
[3] Chen P, Wen H, Jing H, et al. AcuFlexBot: a force-controlled robot for breathing-synchronized acupuncture. IFAC-PapersOnLine, 2025, 59(35): 326-331.
[4] Zhang C, Han Y, Liu W, et al. Kinematic modeling and simultaneous calibration for acupuncture robot. Applied Mathematical Modelling, 2024, 130: 438-456.
[5] Yang C, Xie Y, Liu S, et al. Force modeling, identification, and feedback control of robot-assisted needle insertion: a survey of the literature. Sensors, 2018, 18(2): 561.
[6] Aghdam A N, Liu P X. A novel path planner for steerable bevel-tip needles to reach multiple targets with obstacles. IEEE Transactions on Instrumentation and Measurement, 2020, 69(10): 7636-7645.
[7] Leipheimer J, Balter M, Chen A, et al. Design and evaluation of a handheld robotic device for peripheral catheterization. Journal of Medical Devices, 2022, 16(2): 021015.
[8] Zhang Y, Ju Z, Zhang H, et al. 3-D path planning using improved RRT* algorithm for robot-assisted flexible needle insertion in multilayer tissues. IEEE Canadian Journal of Electrical and Computer Engineering, 2022, 45(1): 50-62.
[9] Villalobos J, Sanchez I Y, Martell F. Singularity analysis and complete methods to compute the inverse kinematics for a 6-DOF UR/TM-type robot. Robotics, 2022, 11(6): 137.
[10] LaValle S M. Rapidly-exploring random trees: a new tool for path planning. Technical Report TR 98-11, Department of Computer Science, Iowa State University, 1998.
[11] Kuffner J J, LaValle S M. RRT-connect: an efficient approach to single-query path planning. Proc. IEEE International Conference on Robotics and Automation (ICRA), 2000: 995-1001.
[12] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 2011, 30(7): 846-894.
[13] Gammell J D, Srinivasa S S, Barfoot T D. Informed RRT*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014: 2997-3004.
[14] Fan J, Chen X, Liang X. UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased. Expert Systems with Applications, 2023, 213: 119137.