Enhanced Path Planning for Industrial Robot: Integrating Modified Artificial Potential Field and A* Algorithm

Fan Rui[1,2], Muhammad Azmi Ayub[1], Mohd Nor Azmi Ab Patar[1], Sukarnur Che Abdullah[1], Fazlina Ahmat Ruslan[1]*

[1] College of Engineering, Universiti Teknologi MARA, 40450, Shah Alam, Selangor, MALAYSIA
[2] Hebei Institute of Mechanical and Electrical Technology, 054000, Xingtai, Hebei, CHINA
**fanrui0321@foxmail.com*

ABSTRACT

The study proposes a modified Artificial Potential Field (APF) method integrated with the A algorithm to enhance industrial robot path planning for obstacle avoidance. This approach addresses issues of local minima and unreachable targets within APF, mitigates the A* algorithm's poor real-time performance, and enhances obstacle avoidance success rates. Kinematic and workspace analyses of the robot utilize the Denavit-Hartenberg and Monte Carlo methods. The study analyses the principles and limitations of classical algorithms. The study introduces a modified APF algorithm to address issues of local minima and path oscillation, which is integrated with A* to guide movement towards the virtual target. After getting rid of local minima, the algorithm reverts to the APF method for further searching. Introducing a safe distance to restrict the repulsive field's influence resolves the issue of unreachable targets. Simulation results demonstrate that the modified algorithm efficiently plans obstacle-free paths in multi-obstacle environments, with target error controlled within 0.0121 m.*

Keywords: *Path Planning; Kinematic; Artificial Potential Field; A* Algorithm; Industrial Robot*

ISSN 1823-5514, eISSN 2550-164X

https://doi.org/10.24191/jmeche.v13i1.3761

Received for review: 2024-06-23
Accepted for publication: 2024-10-09
Published: 2024-11-15

Introduction

With the rapid advancement of smart manufacturing technologies, this study uses digital twin technology as the guiding idea to innovate and enhance the path planning strategy of industrial robots in smart manufacturing [1]. Industrial robots with six rotary joints (6R) are standard advanced equipment in smart manufacturing, effectively assisting in labour-intensive tasks such as machining, handling, and sorting, thus reducing the burden of tedious physical labour [2]. Given the complexity and dynamics of the robot's working environment, as well as the variability in the number and location of surrounding obstacles, it is imperative to find a safe obstacle avoidance path that can bypass all obstacles and reach the target smoothly [3].

An effective path planning algorithm significantly enhances robots' autonomous obstacle avoidance capability, thereby profoundly impacting their efficiency, work quality, and service life [4]. Generally, robot path planning methods can be broadly classified into two categories: global path planning and local path planning. A* algorithm [5], genetic algorithm [6], ant colony optimization algorithm [7], and dynamic programming algorithm are examples of global path planning algorithms, which are suitable for less environment change or static environment. The artificial Potential Field (APF) method [8], particle swarm optimisation algorithm [9], and simulated annealing algorithm are local path algorithms, which are suitable for dynamic environments or completely unknown environments. Due to the complex working environments of industrial robots and the high requirements for real-time control, the APF method is often chosen for robot obstacle avoidance path planning [10]. Ji et al. [11] introduced virtual obstacles into the APF method to disrupt the equilibrium and solve the local minima, but it is still a challenge to select appropriate obstacles in practical applications. Abadlla et al. [12] combined the APF method with the fuzzy logic techniques, which alleviated certain local minimum challenges inherent in the classical APF method, although it could not completely alleviate the problem of the robot's repetitive oscillations. Ding et al. [13] proposed to integrate the positive hexagonal guidance method into the APF method to efficiently circumvent the local minima by navigating around obstacles; however, this greatly prolongs the operation time of robots in complex scenarios.

In this paper, an obstacle avoidance path planning algorithm for industrial robots is proposed on the basis of previous research [10]-[13]. The efficiency of the algorithm is enhanced by refining the APF method, which represents the potential field in joint space. Additionally, the repulsive field function is strengthened by incorporating a safe distance, addressing the issue of the robot being unapproachable near the target. The APF methods were then integrated with the A* algorithm. When the robot's path encounters a local minimum, the A* algorithm guides the robot toward a virtual goal. Once the local minimum is resolved, the algorithm reverts to the enhanced APF method,

and path exploration is restarted until the destination is reached and the shortest path is determined. Using the digital twin idea, the path planning algorithm is simulated and optimised in a virtual environment, laying the foundation for validation on a physical robot.

Methodology

In the context of robotics, the section mainly discusses the theoretical basis for obtaining robot obstacle avoidance path planning through mathematical modeling and algorithm improvement.

Robot DH model

The 6R industrial robot comprises six rotating joints and mechanical links. Denavit-Hartenberg (DH) notation, as described by Hartenberg and Denavit [14], defines the spatial relationship between adjacent joint axes. Figure 1 illustrates the standard DH links for the 6R industrial robot. To describe the pose of the robot's endpoint in Cartesian space, each joint is sequentially translated and rotated relative to a fixed coordinate frame. Table 1 summarizes the DH parameters of the 6R industrial robot, comprising four parameters θ_i, d_i, α_i, a_i for each link in DH notation.

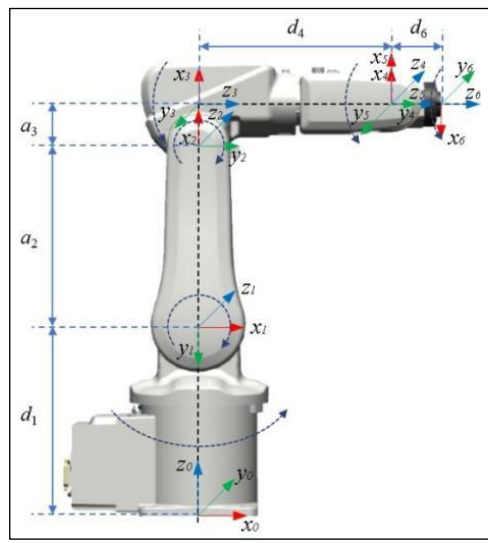

Figure 1: Coordinate frame of 6R industrial robot

Table 1: DH parameters of 6R industrial robot

Link	θ_i /°	d_i /m	α_i /°	a_i /m	Joint Range/°
1	θ_1	0.29	-90	0	-165°~+165°
2	θ_2 - 90	0	0	0.27	-110°~+110°
3	θ_3	0	-90	0.07	-110°~70°
4	θ_4	0.302	90	0	-160°~160°
5	θ_5	0	-90	0	-120°~+120°
6	θ_6 - 180	0.072	0	0	-400°~+400°

Robot Kinematics and Workspace Strategy

According to the DH notation, the homogeneous matrix ${}^{i-1}_{i}T$ with the elementary rotations and translations describes the transformation from the two consecutive link coordinate frames [15], which can be expanded as:

$$
\begin{aligned}
{}^{i-1}_{i}T &= \mathrm{Rot}(z,\theta_i)\times Trans(0,0,d_i)\times Trans(a_i,0,0)\times Trans(x_i,\alpha_i) \\
&= \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{1}
$$

The positional relationships of the links relative to the reference coordinate frame are derived by matrix operations [16]. Multiply the homogeneous matrices sequentially, and the homogeneous matrix ${}^{i-1}_{i}T$ from the link coordinate frame $\{0\}$ to frame $\{6\}$ is expressed as:

$$
{}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \tag{2}
$$

In robotics, the location and orientation of the robot endpoint are precisely captured by two matrices R and P, which are closely related to the robot's joint angles $\theta_i(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)$ [17]. Through fine-grained calculations in kinematics, the robot's workspace is revealed, which consists of all realizable points reachable by the robot's end. The workspace is a well-defined boundary that delimits the full extent of the robot's kinematic capabilities and is a prerequisite for the robot to achieve collision-free path planning [18]. The Monte Carlo method [19] is a numerical computation method that solves mathematical problems by iterative random sampling, which is fast and easy to compute and facilitates the use of computer-aided

methods for accurate graphing. Therefore, the Monte Carlo method is usually employed to assess the availability of points in the robot's workspace by leveraging random sampling techniques. The mathematical model of the robot implemented with the help of the MATLAB Robotics Toolbox is shown in Figure 2. The model is used to compute the robot's kinematics and workspace and can also be used to simulate complex path planning scenarios to advance robotics.

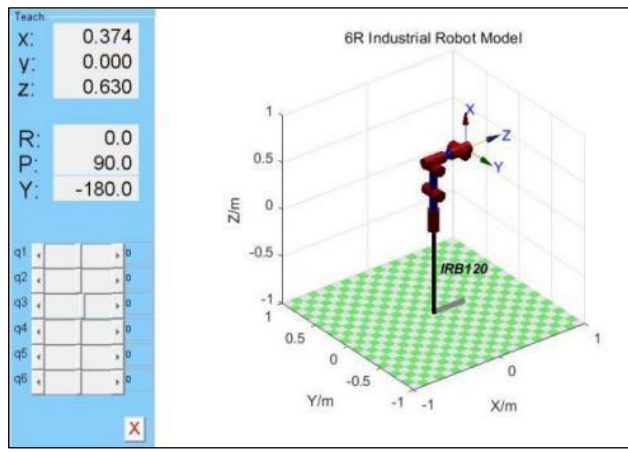

Figure 2: Mathematical model of 6R industrial robot

Robot collision detection model

Effective collision detection between robots and obstacles is essential for path planning [20], and it is imperative to consider potential collisions between the various links of the robot. 6R industrial robots always operate in complex environments and encounter obstacles with irregular shapes in smart manufacturing. To simplify collision detection modeling, hierarchical bounding box techniques [21] are commonly utilized for obstacle approximation. Common types of bounding boxes include spheres, cylinders, Axis-Aligned Bounding Boxes (AABBs), Oriented Bounding Boxes (OBBs), K-Discrete Oriented Polytope (K-DOPs), etc [22]. Establishing complex bounding boxes and conducting collision detection requires more time. To satisfy the real-time demands of robot motion control, the paper chooses spheres to model the envelopes of obstacles due to their fewer parameters and faster calculation speed. By projecting spatial obstacles onto the xoy xoy, yoz, and xoz planes, the maximum and minimum values of the obstacle's projections on the x, y and z axes can be obtained as x_{max}, x_{min}, y_{max}, y_{min}, z_{max}, and z_{min}. Consequently, the center $O(x_j, y_j, z_j)$ and radius R_j of the bounding sphere are determined as:

$$x_j = \frac{1}{2}(x_{\max} + x_{\min}), y_j = \frac{1}{2}(y_{\max} + y_{\min}), z_j = \frac{1}{2}(z_{\max} + z_{\min})$$
$$R_j = \frac{1}{2}\sqrt{(x_{\max} + x_{\min})^2 + (y_{\max} + y_{\min})^2 + (z_{\max} + z_{\min})^2} \quad (3)$$

Collision detection does not need to be considered during the robot's movement because the robot's base position remains fixed. Based on the mechanical structure and motion characteristics of the IRB120 robot, the cylindrical bounding box was utilized to represent the robot and obtain three cylindrical bounding boxes, as illustrated in Figure 3.

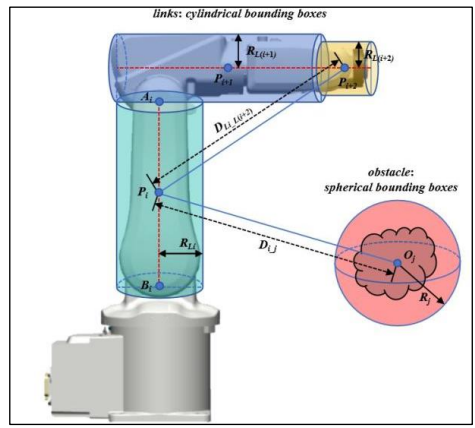

Figure 3: Collision detection model of 6R industrial robot

The position of a cylinder can be determined by its axis endpoint, which can be obtained through the homogeneous transformation matrix. Let R_L represents the maximum radial extent of the robotic link, and denotes the endpoint of the cylindrical axis as $A(x_A, y_A, z_A)$, $B(x_B, y_B, z_B)$. Then, a point $P(x_P, y_P, z_P)$ located on the axis of the cylinder, which corresponds to the line segment connecting points A and B, can be represented as:

$$\begin{cases} x_P = x_A + s(x_B - x_A) \\ y_P = y_A + s(y_B - y_A) \\ z_P = z_A + s(z_B - z_A) \end{cases} \quad (4)$$

where s is a proportional coefficient. Therefore, the collision detection issue of the 6R robot is transformed into the issue of comparing the shortest distance $d(P_i, O_j)$ between a cylindrical bounding box and a spherical bounding box,

or the shortest distance $d(P_i, P_{i+2})$ between two cylindrical bounding boxes, with the sum of their respective. Whether a collision between each link and obstacle is determined as:

$$d(P_i, O_j) = D_{i\_j} = \left\| P_i - O_j \right\| < \left(R_{Li} + R_j \right), i = 1,2,3,\ j = 1,2,\ldots,n \tag{5}$$

As for spatial multi-degree-of-freedom tandem robots, self-collision between two adjacent links is typically prevented by joint limitations or mechanical structures. Therefore, if a self-collision occurs, it will only happen between two links that are not directly joined. The determination of a collision between each link is assessed as:

$$d(P_i, P_{i+2}) = D_{Li\_L(i+2)} = \left\| P_i - P_{i+2} \right\| < \left(R_{Li} + R_{L(i+2)} \right), i = 1 \tag{6}$$

Modified algorithm for robot path planning

The modified algorithm combines the benefits of the APF method for smooth navigation around obstacles with the A* algorithm search for more effective obstacle avoidance paths.

Artificial potential field method

The APF method is a virtual force-based approach introduced by American scientist Khatib in the 1980s [23], which enables the robot to compute a feasible collision-free path during the moving process. As a classical robot path planning algorithm, the model is relatively simple, with fewer operations, high algorithmic efficiency, and most importantly strong adaptability to unknown environments. The force analysis of the APF method is illustrated in Figure 4.

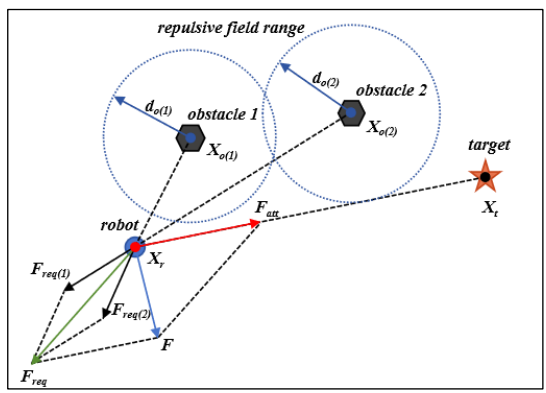

Figure 4: The force analysis of the robot in the APF method

where the circle, hexagons, and pentagon represent the locations of the robot, obstacles, and target respectively, which assumed as X_r, $X_{O(i)}(i = 1,2,\ldots,n)$, X_t. A virtual attractive potential field U_{att} is established at the target location, while a virtual repulsive potential field $U_{rep(i)}$ is created at the positions of obstacles. The total force F directs the robot's movement, resulting from the attractive force, F_{att} and the repulsive force, $F_{rep(i)}$. These forces result from the influence of the potential fields on the robot, prompting it to advance toward the target along the path of decreasing potential energy, thereby ensuring a collision-free trajectory. The conventional attractive field function can be represented as:

$$U_{att}(X_r) = \frac{1}{2} k_a d\left(X_r - X_t\right)^2 \tag{7}$$

where k_a is the correction factor of the attractive potential field. $d(X_r - X_t)$ is a vector which is an Euclidean distance from the position of the robot end effector to the target. The conventional repulsive field function can be written as:

$$U_{rep}(X_r) = \sum_{i=1}^{n} U_{rep(i)}(X_r) \tag{8}$$

$$U_{rep(i)}(X_r) = \begin{cases} \frac{1}{2} k_{r(i)} \left(\frac{1}{d(X_r - X_{o(i)})} - \frac{1}{d_{o(i)}}\right)^2, 0 \le d(X_r - X_{o(i)}) \le d_{o(i)} \\ 0, d(X_r - X_{o(i)}) \ge d_{o(i)} \end{cases} \tag{9}$$
$$i = 1,2,\ldots n$$

where $k_{r(i)}$ is the correction coefficient of the repulsive potential field generated by each obstacle. $d\left(X_r - X_{o(i)}\right)$ is a vector representing the Euclidean distance from the position of the robot end effector to each obstacle. $d_{o(i)}$ is a constant that indicates the range of repulsion influence.

Therefore, the resultant potential field acting on the robot consists of gravitational and repulsive potential and can be expressed as:

$$U(X_r) = U_{att}(X_r) + U_{rep}(X_r) \tag{10}$$

The gravitational influence $F_{att}(X_r)$ and the repulsion $F_{rep}(X_r)$ is induced by the potential energy can then be expressed as the opposite gradient of the potential energy function. $F_{att}(X_r)$ is given as:

$$F_{att}(X_r) = -\nabla U_{att}(X_r) = k_a d\left(X_r - X_t\right) \tag{11}$$

The resultant repulsion, $F_{rep}(X_r)$ is the sum of repulsion generated by each obstacle, derived from the opposite gradient of the repulsive potential field, given by:

$$F_{rep}(X_r) = -\nabla U_{rep}(X_r) = -\nabla(\sum_{i=1}^{n} U_{rep(i)}(X_r)) = \sum_{i=1}^{n} F_{rep(i)}(X_r) \quad (12)$$

$$F_{rep(i)}(X_r) = \begin{cases} k_{r(i)}(\frac{1}{d(X_r - X_{o(i)})} - \frac{1}{d_{o(i)}})\frac{1}{d(X_r - X_{o(i)})^2} \times \\ \nabla d(X_r - X_{o(i)}), \quad 0 \le d(X_r - X_{o(i)}) \le d_{o(i)} \\ 0, \quad d(X_r - X_{o(i)}) \ge d_{o(i)} \end{cases} \quad (13)$$

$i = 1,2,...n$

The resultant force $F(X_r)$ acts on the robot is:

$$F(X_r) = -\nabla U(X_r) = F_{att}(X_r) + F_{rep}(X_r) \quad (14)$$

Based on the functional expression of the APF method it can be analyzed that it also has drawbacks and limitations in some aspects. One of the limitations is the occurrence of local minima. When the robot reaches a specific position in space, the gravitational force acting on it is canceled by the repulsive force exerted by nearby obstacles, resulting in a net force of zero. In such cases, the robot is unable to determine where to go next, causing the robot to stop or wander. To address this issue, the A* algorithm can be initiated to perform a heuristic search until the local minimum is jumped out and a valid route from the initial position to the target position is obtained.

An additional constraint emerges when the obstacle is situated near the target, hindering the robot's access to it. According to Equations (11) and (13), as the robot approaches the target, its distance $d(X_r - X_t)$ from the target position and $d(X_r - X_o)$ from the obstacle position decreases at the same time, causing the repulsive force to increase instead when the gravitational force on the robot decreases. When the repulsive force significantly outweighs the gravitational force, the robot often drifts away from the target position due to the dominant repulsive influence, preventing it from reaching the target. Therefore, this can be reduced by modifying the potential field function.

A* Algorithm

A widely used heuristic search algorithm, the A* algorithm is designed for path planning in global and static environments [24]. Based on the Best-First-Search algorithm and Dijkstra's algorithm, the A* algorithm incorporates a

heuristic function to assess the cost of getting from the current node to the goal node, thus measuring the path's quality, and selecting a neighbouring node with the lowest estimated value as the best node to move to. The evaluation function of the A* algorithm can be expressed as:

$$f(X_r) = g(X_m) + h(X_m) \tag{16}$$

where function $f(X_r)$ represents the total estimated cost of the path in three-dimensional space, spanning from the initial node X_s, passing through the intermediate node X_m, and reaching the target node X_t. while $h(X_m)$ indicates the estimated cost of reaching the target node from the intermediate node. It's important to choose an appropriate Heuristic Function $h(X_m)$ to ensure that the A* algorithm provides the best path-finding effect. The costs of $g(X_m)$ and $h(X_m)$ should be as close as possible to ensure optimal path searchability and efficiency. Widely utilized heuristic functions include Manhattan distance, Chebyshev distance, and Euclidean distance. In this research, the Euclidean distance is chosen as the heuristic function, which is expressed as:

$$h(X_m) = \sqrt{(x_m - x_t)^2 + (y_m - y_t)^2 + (z_m - z_t)^2} \tag{17}$$

The algorithm guides the search in the most promising direction, traverses fewer intermediate nodes, and makes it widely applicable for finding the shortest path in state space. However, the number of nodes traversed also rises, resulting in an abundance of nodes in the Openlist. This reduces the speed of the algorithm and consumes significant memory space.

Modified artificial potential method

The 6R industrial robot is a multi-degree-of-freedom tandem robot with strongly coupled motion characteristics and a three-dimensional workspace. When using the traditional APF to search for paths in Cartesian space, the path points must be converted to joint space using the inverse kinematics solution in order to determine the corresponding joint angles needed to drive the robot. Since the current optimal joint combination values have to be continuously screened out during the inverse solution calculation, this not only increases the computational complexity but also may result in singular solutions, leading to discontinuous joint angle values before and after, which cannot be applied to practical occasions. Therefore, to achieve path planning for 6R industrial robots, the Cartesian space coordinates of the potential field function are represented in terms of joint angles. This allows the potential energy function in joint space to be integrated into the gravitational potential energy function. The modified gravitational potential energy function is expressed as follows:

$$U_{att}(\theta_r) = \frac{1}{2} k_a d\left(X(\theta_r) - X_t\right)^2 + \frac{1}{2} k_j \sum_{i=1}^{6} \left|\theta_r(i) - \theta_t(i)\right|^2 \tag{18}$$

$$d\left(X(\theta_r) - X_t\right) = \sqrt{\left(x_r(\theta) - x_t\right)^2 + \left(y_r(\theta) - y_t\right)^2 + \left(z_r(\theta) - z_t\right)^2} \tag{19}$$

where θ_r is a six-dimensional array, denoted as $\theta_r = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6,]$. k_j is the gravitational potential field gain coefficient in joint space, $\theta_r(i)$ and $\theta_t(i)$ are the current angle and the target angle of the robot $i^{th}(i = 1,2, \dots ,6)$ joint.

As for the problem of target unreachability, a safe distance factor $d(X_r - X_t)$ between the robot and the target position is incorporated into the repulsive energy function, gradually reducing the distance to limit the impact of the repulsive potential energy on the robot. Within the safe range of the robot approaching the target, the influence of repulsion can be disregarded, enabling the robot to effectively reach the destination and overcome the unreachable state of the target. The modified repulsive potential energy function is formulated as:

$$U_{rep(i)}(\theta_r) = \begin{cases} \frac{1}{2} k_{r(i)} \left(\frac{1}{d(X(\theta_r) - X_{o(i)})} - \frac{1}{d_{o(i)}}\right)^2 \times \\ d\left(X(\theta_r) - X_t\right)^n, & 0 \le d(X(\theta_r) - X_{o(i)}) \le d_{o(i)} \\ 0, & d(X(\theta_r) - X_{o(i)}) \ge d_{o(i)} \end{cases} \tag{20}$$

$$i = 1,2,\dots n$$

$$d\left(X(\theta_r) - X_{o(i)}\right) = \sqrt{\left(x(\theta_r) - x_{o(i)}\right)^2 + \left(y(\theta_r) - y_{o(i)}\right)^2 + \left(z(\theta_r) - z_{o(i)}\right)^2} \tag{21}$$

where n represents the nth power of the separation between the robot and the target, which is typically set to $n = 2$. Intuitively, as an object approaches the target, although the repulsive field continues to increase, the rate of increase is relatively slower compared to the original repulsive field. This can somewhat alleviate the problem of excessive repulsion. As the robot nears the target, the increment of repulsion decreases, aiding the robot in adjusting its position and direction more accurately to avoid collisions and reach the target faster. Therefore, selecting appropriate correction coefficients of the potential field and distance power-law exponents can effectively balance the gravitational and repulsive forces of the robot, improving their motion efficiency and stability.

Integrating modified APF and A* algorithm

To address the respective shortcomings of the APF and the A* algorithm, we propose combining the two algorithms for robot obstacle avoidance path planning. Figure 5 illustrates the implementation of this new combined algorithm. The combination algorithm addresses the issue of the robot becoming trapped in local minima and unable to escape during APF planning, while also mitigating the relatively low real-time performance of A*.

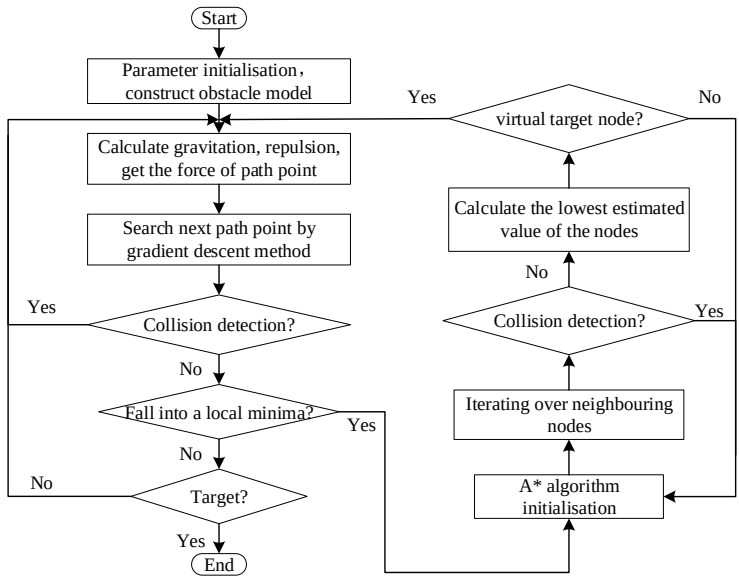

Figure 5: Flow chart of the modified algorithm

In the modified algorithm, the gradient descent method is employed to implement the modified APF method for comprehensive robot path planning. This method enables the robot to move against the gradient, starting from the initial point and continuing until the potential field's gradient vector reaches zero. To minimize the impact of inverse kinematics computations on the algorithm's efficiency, this study opts to perform path planning in joint space, planning the path point of each joint of the robot separately, and representing the path point with the joint angle. Through the constraint relationship between the adjacent links, the Cartesian space coordinates X_r of the robot endpoint can be computed based on the joint angle of the 6R robot.

Firstly, a modified APF algorithm is used for path planning. The search step is λ, the next path point is $\theta_{i+1} = \theta_i - \lambda F(\theta_i)$. The variable θ_i represents a six-dimensional array, with each element corresponding to one of the six joint angles of the 6R industrial robot, i denotes the step i. When it is judged

to enter the local minima, to set a virtual target point and control the robot to move towards it, which is denoted as the variable θ_{tar}. When the path planning method is changed into the A* algorithm, the robot's present joint angles θ_{int} (the local minimum point) is the initial searching node of A*; the neighbouring joint space $(\theta_i - \mu, \theta_i, \theta_i + \mu)$, where μ is the joint angle searching step; and the valuation function in step i is defined as follows:

$$g(\theta_i) = \sum_{n=1}^{i} \left\| \theta_n - \theta_{n-1} \right\| \tag{22}$$

$$h(\theta_i) = \max_{m=1,2,\ldots,6} \left| \theta_i[m] - \theta_{tar}[m] \right| \tag{23}$$

The succeeding node at step $(i+1)$ is generated based on the specified step length, and the node with the lowest value of $f(\theta_{i+1})$ is then selected as the optimal joint angle θ_{best}. This iterative process ensures that the optimal joint angle is determined at each step, allowing the robot to seek the ideal route within the joint space based on heuristic information provided by this estimation function.

Results and Discussion

This section discusses the results of analyzing the workspace of the IRB120 industrial robot to delineate its range of motion and reachable points. Based on this, the performance of an enhanced robot obstacle avoidance algorithm is also evaluated to optimize path planning to improve the robot's work efficiency and safety.

Results analysis of robot workspace

The workspace analysis of the ABB IRB120 robot was performed using the MATLAB Robotics Toolbox. Based on the angular range and kinematic equations of each joint, the Monte Carlo method was used to generate random values using the rand function to cover the predefined intervals of joint variables. The final results of the robot's workspace are presented in Figure 6. Table 2 lists the assignment of the random variable and the range of motion of the robot along each axis. The results show that the overall workspace of the IRB120 industrial robot is compactly distributed in a spherical shape, but the robot cannot reach its tail space due to joint restrictions. In addition, the Monte Carlo method achieved a very smooth spatial variation without any significant abrupt changes. Importantly, the workspace does not exceed the rotational range of each joint variable, which is very much in line with the requirements of practical applications.

Table 2: Results of IRB120 industrial robot workspace

Parameters	Value
Random value	20000
x-axis moving range	[-0.65, 0.65]
y-axis moving range	[-0.65, 0.65]
z-axis moving range	[-0.18, 0.94]

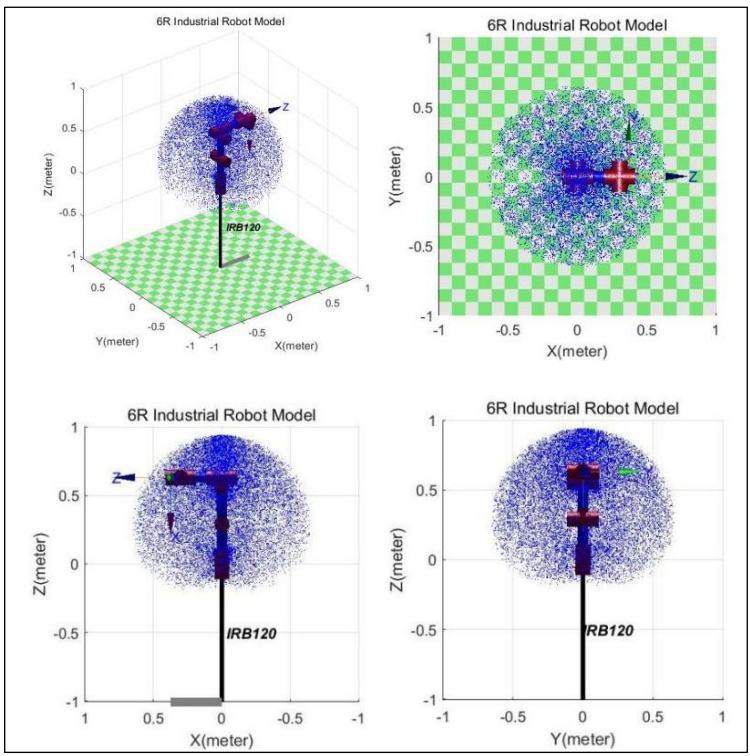

Figure 6: Workspace of IRB120 industrial robot

Results analysis of enhanced robot path planning

To evaluate the obstacle avoidance and path optimization performance of the modified algorithm, the following experiments were conducted in the Software MATLAB 2020b. The computer system is Windows 10 professional edition, the CPU parameter is Intel Core I5-8250U CPU @ 1.60 GHz 1.80 GHz, the graphics card parameter is Inter(R) UHD Graphics 620, and the memory is 16 GB. Set the relevant parameters of the algorithm, the gravitational potential field correction coefficients are set as $k_a = k_j = 20$, while the repulsive

potential field correction coefficient for each obstacle is defined as $k_{r(i)} = 25$, the influence range of repulsion is set at d_o = 0.15 m, the search step size is $\lambda = 2°$ and $\mu = 0.5°$. The motion control parameters of the 6R industrial robot are detailed in Table 3.

Table 3: Parameters of 6R industrial robot and obstacles (unit: m)

Parameters	**Experiment I**	**Experiment II**
The start position	(0, 0.346, 0.46)	(0,0.346, 0.46)
The target position	(0.09, 0.335, 0.297)	(0.09, -0.335, 0.297)
Obstacles position	(0.35, 0.3, 0.35) (0.45, -0.3, 0.25)	(0.38, 0.2, 4) (0.4, -0.2, 4)
Obstacle radius	0.1, 0.13	0.1, 0.13

The results of the experiment in MATLAB are depicted in Figures 7, 8, and 9, where the spheres represent the obstacle that encloses the bounding sphere, and the solid red line denotes the path from the start to the target. Figure 7 illustrates the outcomes of classical APF path planning. It is observed that when two obstacles are far apart, the classical APF algorithm is effective in planning a path. However, when the obstacles are close in distance, Figure 8 demonstrates that the robot becomes trapped in a local minima during classical APF planning, rendering it unable to progress towards the target, leading to path planning failure. In contrast, Figure 9 illustrates that the modified APF method successfully navigates out of the local minima and reaches the target. The performance outcomes of the algorithms are presented in Table 3.

Table 4: The performance outcomes of the algorithms

Parameters	**Classical APF in Experiment I**	**Classical APF in Experiment II**	**Modified APF & A* in Experiment II**
Obstacle position	Two obstacles far away	Two obstacles nearby	Two obstacles nearby
Path length	1.363 m	-	1.346 m
Run time	1.563 s	-	0.895 s
Iterations	356	500 (max)	323
Simulation results	Not trapped in local minima (Figure 7)	Trapped in the local minima (Figure 8)	Jump out of the local minima (Figure 9)

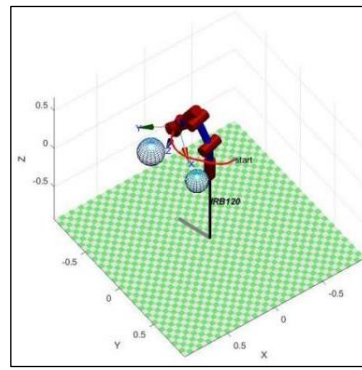

Figure 7: Not trapped in local minima in classical APF method

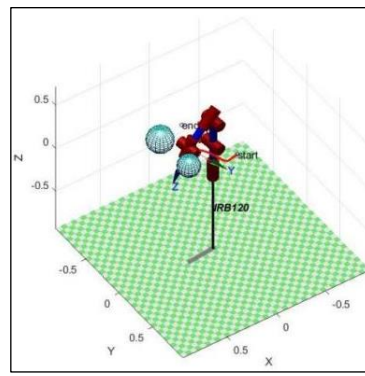

Figure 8: Trapped in local minima in classical APF method

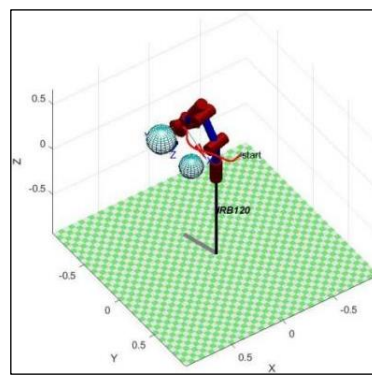

Figure 9: Jump out of local minima in modified APF and A* algorithm

Additionally, ten experiments were conducted in MATLAB utilizing both the classical APF method and the modified algorithm to investigate the issue of inaccessible targets when obstacles are located close to the target. The experimental parameters were set up to ensure that the robot would steer clear of specified obstacles within the workspace. Under normal circumstances, all ten experiments were successfully planned. The error between the planned target and the given target obtained by the function is:

$$\varepsilon = d(X_{ti} - X_t) \tag{24}$$

where X_{ti} is the target planned by the algorithm, X_t is the given target. $\mathrm{d}(X_{ti} - X_t)$ is the distance between the planned target and the given target. The errors obtained by the two algorithms are shown in Table 5.

Using the errors collected from the ten experiments, the average error of the algorithm can be computed as follows:

$$\varepsilon_a = \frac{1}{n}\sum_{n=1}^{\infty}\varepsilon_n \tag{25}$$

Table 5: Error of the target obtained based on the two algorithms (unit: *m*)

Algorithms	**Error** ε										
	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7	ε_8	ε_9	ε_{10}	ε_a
Classical APF	0.025	0.023	0.025	0.025	0.026	0.024	0.025	0.023	0.026	0.025	0.0247
Modified APF and A*	0.013	0.012	0.013	0.011	0.011	0.012	0.010	0.012	0.014	0.013	0.0121

The calculations indicate that the mean error of the enhanced algorithm is 0.0121 m, while that of the classical algorithm is 0.0247 m. This demonstrates that the improved algorithm achieves greater accuracy in reaching the target compared to the classical APF method.

Conclusion

In summary, we have tailored an enhanced obstacle avoidance path planning algorithm for the 6R industrial robot. By integrating APF and A* algorithms, our approach optimizes global and local path planning, thus saving robot motion time, reducing mechanical wear, and ensuring safe robot operation in obstacle-intensive environments. With accurate Denavit-Hartenberg modelling and simplified collision detection methods, our algorithm achieves

precise motion control and efficient obstacle avoidance. Building on this foundation, the integration of digital twin technology can be further pursued. By establishing a digital twin system, virtual testing and optimization studies on the robot's motion performance can be conducted to enhance planning capabilities and improve path quality, thereby laying the groundwork for the future development of practical applications. In addition, overcoming the limitations of the collision detection model and exploring the applicability of the algorithm in complex working environments require further research.

Contributions of Authors

The conception and design of the study were contributed to by all authors. Fan Rui was responsible for material preparation, data collection, and analysis. The initial draft of the manuscript was authored by Fan Rui, and all authors provided feedback on prior versions. The final version of this work was reviewed and approved by all authors.

Funding

This research was financially assisted by the Hebei Institute of Mechanical and Electrical Technology, China, through scholarships dedicated to facilitating this study.

Conflict of Interests

One of the authors, Mohd Nor Azmi Ab Patar is an assistant managing editor of the Journal of Mechanical Engineering (JMechE). The author has no other conflict of interest to note.

Acknowledgement

The support provided by the UiTM Collaborative Research Laboratory is gratefully acknowledged by the authors. The authors would also like to express their gratitude for the support of experimental equipment provided by the Hebei Institute of Mechanical and Electrical Technology, Xingtai, China.

References

[1] M. Zafar, E. F. Hamza, Langås, and F. Sanfilippo, "Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review", *Robotics and Computer-Integrated Manufacturing*, vol. 89, p. 102769, 2024. https://doi.org/10.1016/j.rcim.2024.102769

[2] M. Zhu, C. Liang, A. C. L. Yeung, et al, "The impact of intelligent manufacturing on labor productivity: An empirical analysis of Chinese listed manufacturing companies", *International Journal of Production Economics*, vol. 267, no. 6, p. 109070, 2024. https://doi.org/10.1016/j.ijpe.2023.109070

[3] S. Liu, and P. Liu, "A review of motion planning algorithms for robotic arm systems", *Singapore: Springer Singapore*. InRiTA 2020: Proceedings of the 8th International Conference on Robot Intelligence Technology and Applications, pp. 56-66, 2021. https://doi.org/10.1007/978-981-16-4803-8_7

[4] T. Zhang, B. Wu, F. Q. Zhou, "Research on improved ant colony algorithm for robot global path planning", *Computer Engineering and Applications*, vol. 58, no. 1, pp. 282-291, 2022. http://doi.org/10.3778/j.issn.1002-8331.2107-0369

[5] F. G. Liu and S. Qiu, "Path Planning of Indoor Mobile Robot Based on Improved A* Algorithm", *2nd International Conference on Artificial Intelligence and Information Systems*, vol. 106, pp. 1-4, 2021. https://doi.org/10.1145/3469213.3470309

[6] Y. S. Li, Y. Wan, Y. Zhang, et al, "Path Planning for Warehouse Robots Based on Artificial Bee Colony-Adaptive Genetic Algorithm", *Chinese Journal of Scientific Instrumentation*, vol. 43, no. 4, pp. 282-290, 2022. http://doi.org/10.19650/j.cnki.cjsi.J2108514

[7] K. Hao, J. L. Zhao, B. B. Wang, et al, "The Application of an Adaptive Genetic Algorithm Based on Collision Detection in Path Planning of Mobile Robots", *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 5536574, 2021. https://doi.org/10.1155/2021/5536574

[8] P. Xin, Y. H. Wang, X. L. Liu, et al, "Optimization and Improvement of path planning Algorithm of RRT and Artificial Potential Field Method", *Computer Integrated Manufacturing Systems*, vol. 29, no. 9, pp. 2899-2907, 2023. http://doi.org/10.13196/j.cims.2023.09.003

[9] K. Hao, C. S. Deng, L. Zhao, "Robot path planning based on region search particle swarm optimization", Journal *of Electronic Measurement and Instrument*, vol. 36, no. 12, pp. 126-135, 2022

[10] L. J. Chen, H. F. Li, T. Zhang, "Research on Obstacle avoidance control of Multi-Robot Formation based on Artificial Potential Field Method", *Journal of Beijing Institute of Printing and Technology*, vol. 32, no. 3, pp. 15-21, 2024

[11] W. Ji, F. Cheng, D. Zhao, Y. Tao, et al, "Obstacle Avoidance Method of Apple Harvesting Robot Manipulator", *Transactions of the Chinese Society for Agricultural Machinery*, vol. 44, no. 11, pp. 253-259, 2013

[12] T. Y. Abdalla, A. A. Abed, A. A. Ahmed, "Mobile Robot Navigation Using Pso-Optimized Fuzzy Artificial Potential Field with Fuzzy Control", *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 6, pp. 3893-3908, 2017. http://dx.doi.org/10.3233/IFS-162205

[13] F. Ding, Y. X. Shi, G. P. Zhu, "Real-Time Estimation for The Parameters of Gaussian Filtering Via Deep Learning", *Journal of Real-Time Image Processing*, vol. 17, no 2, pp. 17-27, 2020. https://doi.org/10.1007/s11554-019-00907-5

[14] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, 2nd ed. Wiley, 2020.

[15] Gaeid, S. Khalaf, et al. "Robot control and kinematic analysis with 6DoF manipulator using direct kinematic method", *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 70-78, 2021. https://doi.org/10.11591/eei.v10i1.2482

[16] Fomin, Alexey, et al, "Inverse and forward kinematic analysis of a 6-DOF parallel manipulator utilizing a circular guide", *Robotics*, vol. 10, no. 1, p. 31, 2021. https://doi.org/10.3390/robotics10010031

[17] Z. Yılmaz, Y. Orkun, B. Zafer, "Design, analysis and simulation of a 6-DOF serial manipulator", *Kocaeli Journal of Science and Engineering*, vol. 3, no. 1, pp. 9-15, 2020. https://doi.org/10.34088/kojose.677184

[18] R. Guamán Rivera, R. García Alvarado, A. Martínez-Rocamora, and F. Auat Cheein, "Workspace analysis of a mobile manipulator with obstacle avoidance in 3D printing tasks", *Applied Sciences*, vol. 11, no. 17, p. 7923, 2021. https://doi.org/10.3390/app11177923

[19] J. W. He, X. L. Ping, Z. Y. Li, et al, "Research and application of the solving method for robotic workspace", *Journal of Mechanical Transmission*, vol. 39, no. 10, pp. 68-71, 2015

[20] Y. A. Lu, K. Tang, and C. Y. Wang, "Collision-free and smooth joint motion planning for six-axis industrial robots by redundancy optimization", *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102091, 2021. https://doi.org/10.1016/j.rcim.2020.102091

[21] B. Gan and Q. Dong, "An improved optimal algorithm for collision detection of hybrid hierarchical bounding box", *Evolutionary Intelligence*, vol. 15, no. 4, pp. 2515-2527, 2022. https://doi.org/10.1007/s12065-020-00559-6

[22] J. T. Klosowski, M. Held, J. S. Mitchell, et al, "Efficient collision detection using bounding volume hierarchies of k-DOPs", *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 1, pp. 21-36, 1998. http://dx.doi.org/10.1109/2945.675649

[23] O. Khatib, et al, "Real-time obstacle avoidance for manipulators and mobile robots", *The International Journal of Robotics Research*. vol. 5, no. 1, pp. 1-98, 1985. https://doi.org/10.1177/027836498600500106

[24] P. E. Hart, N. J. Nilsson, B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968. https://doi.org/10.1109/TSSC.1968.300136