

RESEARCH ARTICLE

Trajectory planning of large redundant manipulator considering kinematic constraints and energy efficiency

Zhenyu Liu^{1,2}, Yu Huang^{1,2}, Daxin Liu^{1,2} , Xuxin Guo^{1,2}, Ke Wang^{1,2} and Jianrong Tan^{1,2}

¹State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China and ²Engineering Research Center for Design Engineering and Digital Twin of Zhejiang Province, Hangzhou, China

Corresponding author: Daxin Liu; E-mail: liudx@zju.edu.cn

Received: 31 December 2022; **Revised:** 5 July 2023; **Accepted:** 17 July 2023; **First published online:** 22 August 2023

Keywords: large redundant manipulator; trajectory planning; motion stability; energy saving; adaptive weighted method

Abstract

For the large redundant manipulator, due to its long working distance and large mass, the number of links (i.e., manipulator's arms) that can be driven to move simultaneously is limited. Otherwise, the control accuracy and motion stability of the manipulator will deteriorate. Focusing on that, a weighted Newton iteration (WNI) algorithm for trajectory planning of the manipulator is firstly proposed, where the motion of the manipulator joints is controlled by a weight matrix, which is constant and related to each link's energy consumption. To dynamically adjust the weight matrix according to kinematic constraints and acquire better energy efficiency, an adaptive WNI (AWNI) algorithm is further proposed. In AWNI, the weight matrix is adjusted in real-time during the planning process, with considerations of the kinematic constraints and the energy consumption of the manipulator. The switch of the links between the working state and the non-working state is made through the weight matrix to achieve flexible control of the manipulator motion. Two evaluation functions are established to validate the effectiveness of AWNI in energy saving and motion stability control. Taking a 6 degrees of freedom (DOF) manipulator as an example, simulation experiments on trajectory planning are carried out and the results show the effectiveness of the proposed AWNI algorithm.

1. Introduction

With the emergency requirement in modern industrial and construction fields, the redundant manipulator has become indispensable engineering equipment in these fields. The redundant manipulator is a kind of manipulator whose degree of freedom is greater than that of the task space. The kinematically redundant manipulator presents higher mobility than the non-redundant for the desired task. In addition, there are infinite joint angle configurations to complete the trajectory planning task with a given end effector pose. The movement attitude of the large redundant manipulator is usually controlled by the operator through a joystick, which requires high experience of the operator. Under this situation, the motion trajectory and positioning accuracy of the manipulator are often poor. In recent years, with the increasing demand for long-distance operation, large redundant manipulators with 5–7 links and a total length of more than 50 m have gradually emerged. However, due to the large structure and heavy load, the driving power is not sufficient to support all links to move at the same time. In addition, motion stability needs to be maintained since the large multi-DOF manipulator is sensitive to vibration. Given the above problems, it is urgent to propose a trajectory planning method that can maintain motion stability as well as consider energy efficiency for the large redundant manipulator.

During the modeling of the redundant manipulator, it is regarded as a kinematic chain system and each link is expressed with respect to its adjacent link [1]. As more links are used in modeling a system, the procedures required to derive the relevant motion equations become more laborious and complicated [2, 3]. To overcome this problem, it is essential to apply a recursive formulation to automatically

obtain the governing equations. The first formulation developed is Lagrange-Euler equations with computational complexity $O(n)$. However, it is found to be inefficient due to the high number of algebraic operations involved [4]. Another mainstream formulation is the Gibbs-Appell equation. However, its computational complexity is $O(n^2)$ and therefore does not meet the computational efficiency requirement. Compared with the above equations, the Newton-Euler equations, with the computational complexity $O(n)$ and rapid convergence speed, are found to be the most appropriate formulation for the trajectory planning of redundant manipulator [5].

Tremendous work has been carried out on the trajectory planning of the manipulators. Aiming at various joint ranges and working scenarios, specific trajectory planning algorithms with considerations of energy saving [6], obstacle avoidance [7], and trajectory smoothing [8] are proposed for the redundant manipulators. The trajectory planning algorithms set the corresponding constraints and optimization objectives according to the application scenarios of the manipulator to solve the optimal trajectory [9].

The manipulator trajectory planning algorithms can be classified into two categories: algorithms based on motion relationship [10, 11] and algorithms based on heuristic search [12, 13]. The trajectory planning algorithms based on motion relationship use known information, such as the initial position or historical posture, to summarize a trajectory planning pattern for the manipulator to follow. The motion of the manipulator is expressed as a nonlinear equation and the limits of joint angles, velocities or accelerations are transferred into constraints, which turns a trajectory planning problem into a nonlinear optimization problem. There are plenty of motion relationship-based trajectory planning researches aiming at energy saving. Ayten et al. [14] transferred kinematic and dynamic constraints into a cost function to reduce computational complexity. Also, they took the redundant links as a single link to make the method easier to compute and compatible with the addition of more links. However, this method neglects the energy-consuming difference between links and might not converge. Potter et al. [15] proposed an optimal trajectory planning algorithm of redundant manipulator with energy saving based on fourth-degree polynomials. They presented an optimal weighting vector that establishes the influence of each joint on the total energy consumption. However, the singularity problem will raise with the lack of consideration of joint limits. In refs. [16, 17], trajectory planning with minimum energy consumption was transferred into a nonlinear optimization problem and the multi-objective optimization method was proposed to achieve the optimal solution. However, the multi-objective function lacks consideration of jerks and constraining conditions in actual situations. Also, there are motion relationship-based trajectory planning researches aiming at motion stability. Pham et al. [18] extended the classical algorithm of time-optimal path parameterization to the case of redundantly actuated robot systems. The authors designed a numerical search for an approximate optimal acceleration to ensure the smoothness of the obtained profile and converted the polygon constraints back into the half-plane inequalities to speed up the computation. However, this method transforms the contact between human and the manipulator into torque constraints to plan the trajectory and thus is not suitable for large manipulators. Su et al. [19] proposed a Pythagorean-Hodograph curves-based trajectory planning algorithm. They classified the different operations according to the geometric relationship of the path. The trajectory planning was carried out for each situation and the curved motion segments were solved by adopting the corresponding polynomial motion law. However, this method allows joint reciprocating motion during the working process and it could cause serious consequences in the large manipulators. Shen et al. [20] investigated a new discrete-time repetitive path planning (RPP) scheme depicted in the pseudoinverse-type formulation for redundant robot manipulators. The scheme was derived from the discretization of the continuous-time RPP scheme using a special numerical difference formula, which had been previously used to discretize a minimum norm velocity scheme in other studies. It was proved that the proposed RPP scheme has the property of a cube pattern in the end effector planning accuracy. However, this method ignores the consideration of joint limits. Pei et al. [21] split the trajectory into several sub-trajectories and used an inverse kinematic-based model to obtain the angular displacements of joints corresponding to the final positions of the end effector in each sub-trajectory. To guarantee motion stability, the quintic polynomial curve was utilized to interpolate in joint space to obtain the continuous angular jerk. However,

the interpolation will reduce the end effector's position precision. In general, the trajectory planning algorithms based on motion relationship can find the solutions satisfying the constraints quickly, but they have many disadvantages, such as only finding the local optimal solutions, requiring convergence gradients, and failing for discontinuous functions.

The trajectory planning algorithms based on heuristic search usually propose criteria to evaluate the optimization objective and then apply the heuristic search method to find the global optimal solutions. Heuristic search performs an ordered search on the planning domain and makes the generated trajectory approach the optimal trajectory. Various constraints can be adopted with strong compatibility. Many heuristic search-based trajectory planning researches aiming at energy saving have been proposed. Elshabasy et al. [22] developed a redundant manipulator trajectory planning method based on the combination of genetic algorithm and constrained function with the consideration of minimum energy consumption. However, with the motion stability not taken into consideration, the jerks of the joints might increase dramatically so it is not suitable for large manipulators. Agarwal [23] proposed a trajectory planning method based on a new fuzzy clustering model, which obtains links' posture data to establish an expert system. The optimal solution with various objectives can be achieved through the expert system. However, this method is too dependent on the richness of data. In refs. [24, 25], an improved PSO algorithm was applied to perform the trajectory planning task. The optimal solution with energy saving was achieved by stochastically generating locations in the workspace and selecting the best candidate. However, this method requires predefined hyperparameters, and it is difficult to balance computation complexity and convergence speed. Also, there are plenty of heuristic search-based trajectory planning researches aiming at motion stability. Kang et al. [26] combined intelligent hill climbing and a genetic algorithm to plan the trajectory under certain constraints, while Marcos et al. [27] proposed a new method that combines the closed-loop pseudoinverse method with the genetic algorithm to do the trajectory planning of redundant robots. In both cases, with the consideration of motion stability, good results can be achieved with various joint constraints and limits. However, neither method can distinguish distal and proximal joints and satisfy the energy saving demand. Lin [28] proposed a hierarchical genetic algorithm for manipulator path planning, which consists of a global path planner and a local motion planner. The global path planner plans a path for the manipulator end effector, and the local motion planner plans the manipulator's configurations along the path by a genetic algorithm with a non-random initial population. The planned trajectory is smoother, but the convergence speed is slower. Shrivastava et al. [29] calculated the jerk-optimized trajectory of a redundant manipulator with the gray wolf optimizer. The approach can generate a minimal difference in the joint trajectory to guarantee joint stability. However, the precision of the end effector will inevitably reduce. In brief, the trajectory planning algorithms based on heuristic search can find the optimal solutions under certain evaluation criteria, but they require long computational time, therefore not suitable for real-time trajectory planning.

Due to redundant degrees of freedom, the trajectory planning problem of the large redundant manipulator can be transformed into a multibody nonlinear problem. For the multibody nonlinear problem where analysis and computation are expensive, a robust optimization strategy with rapid solution convergence characteristics is highly desirable. To perform trajectory planning under multiple constraints, the Newton iteration (NI) method is applied in recent researches [30–32]. However, the traditional NI-based trajectory planning method cannot solve the following problems.

The motion range of the joints is constrained during the working process; therefore, the solutions solved by the trajectory planning algorithms may not be valid [33]. And if some joints reach the limits, the trajectory planning algorithms will fail to find solutions because of the singularity problem [34, 35]. Furthermore, due to the limitations of the driving power, all the joints of the manipulator cannot be activated simultaneously. Link motion switching should be considered when planning the trajectory of the manipulator.

In this paper, a trajectory planning method of redundant manipulators based on both motion relationship and heuristic search is proposed, which takes the switch of link motion and kinematic constraints into consideration, and aims at improving energy efficiency and motion stability. Firstly, a weighted

Newton iteration (WNI) algorithm is proposed by combining NI, a motion relationship-based method, with heuristic search. In WNI, the NI method is firstly used to find the solution set of trajectory planning for the link system, then the weighted method is applied to provide a weight matrix for the motion of the joints in the system to select the optimal solution. Based on that, to achieve better energy efficiency and motion stability, an adaptive weighted Newton iteration (AWNI) algorithm is further proposed. In AWNI, the weight matrix is adjusted adaptively during the planning process according to the kinematic constraints and energy consumption of the manipulator. The weight matrix takes the joint limits into consideration, which guarantees that the joints approaching the limits contribute less on the trajectory planning and gradually stop working. Also, the AWNI algorithm selects another joint to continue the trajectory planning and switches the working joint. In conclusion, the AWNI algorithm can acquire the optimal solution under the constraints above for low energy consumption and better motion stability. Correspondingly, two evaluation functions that can represent the aforementioned targets are defined to evaluate the effectiveness of the trajectory planning of the large redundant manipulator.

This paper has the following innovations:

- A trajectory planning algorithm for large redundant manipulators, AWNI, is proposed, which exploits both motion relationship and heuristic search. First, Newton iteration, which is a motion relationship-based method, is applied to solve the solution set. Then, a heuristic search-based method is applied to find the optimal solution of a loss function with the adaptively weighted joints. Link motion switching and the limits of the joints are respectively considered to achieve better energy efficiency and motion stability.
- A weight matrix is proposed to find the optimal solution for energy saving and motion stability. In terms of the energy consumption of proximal and distal joints, the proposed AWNI algorithm selects a group of working joints and adjusts the weight matrix adaptively according to the working state of each joint to achieve better energy saving. The weight value of the joint approaching the limit adaptively increases to make sure the joint contributes less on the trajectory planning and gradually stops so that motion stability can be guaranteed.
- Two evaluation functions are established to intuitively compare the differences between various algorithms in terms of energy saving and motion stability control. Simulation verifies that the AWNI algorithm outperforms NI and WNI algorithms in trajectory planning of large redundant manipulators.

The rest of this paper is organized as follows. Section 2 analyzes the kinematic model of the redundant manipulator and elaborates on the application of the Newton iteration algorithm in redundant manipulator trajectory planning. Section 3 proposes the WNI trajectory planning algorithm, aiming at improving energy efficiency and motion stability, and establishes two evaluation functions to evaluate the algorithm's performance with the aforementioned targets. Section 4 further proposes the AWNI algorithm to adjust the weight matrix adaptively during the planning process, according to the kinematic constraints and link motion switching. Section 5 carries out simulation experiments and makes a discussion about the simulation results. Section 6 summarizes the study.

2. Kinematic description and posture solving of the redundant manipulator

2.1. Kinematic description of the redundant manipulator

In most cases, a large redundant manipulator consists of several links and a rotating base. Figure 1 shows the schematic diagram of a six-link redundant manipulator. The rotating base enables the link system to rotate around the y axis, and each link makes itself move in the XOY coordinate plane by the corresponding joint.

Let the angle of the rotating base relative to the positive direction of the x axis be θ_0 . The actuated links are numbered sequentially as the link i ($i = 1 \dots 6$). For each link, l_i denote its length, and θ_i ($i = 1 \dots 6$) denotes the angle between link i and link $i-1$. In terms of the structural characteristic of the links, the manipulator's kinematic model can be established with the D-H [36] method.

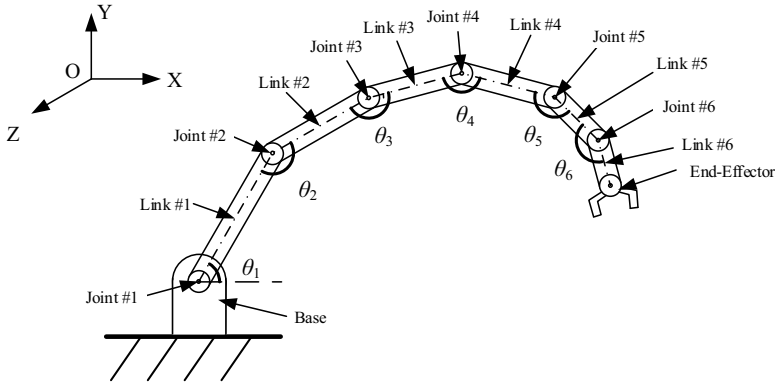


Figure 1. Schematic diagram of the manipulator.

Assuming the center point of joint #1 is the initial point and its coordinates are $[0 \ 0 \ 0]^T$, the relationship between the initial point and the manipulator’s terminal center point can be described as follows

$$\mathbf{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} c_0(l_1c_1 - l_2c_{12} + l_3c_{123} - l_4c_{1234} + l_5c_{12345} - l_6c_{123456}) \\ l_1s_1 - l_2s_{12} + l_3s_{123} - l_4s_{1234} + l_5s_{12345} - l_6s_{123456} \\ s_0(l_1c_1 - l_2c_{12} + l_3c_{123} - l_4c_{1234} + l_5c_{12345} - l_6c_{123456}) \end{bmatrix}. \tag{1}$$

where $c_0 = \cos(\theta_0)$, $s_0 = \sin(\theta_0)$, $c_1 = \cos(\theta_1)$, $s_1 = \sin(\theta_1)$, $c_{12} = \cos(\theta_1 + \theta_2)$, $s_{12} = \sin(\theta_1 + \theta_2)$ and so on, \mathbf{P} is the 3×1 coordinate vector of the manipulator’s terminal center point.

Therefore, when θ_0 is determined, Eq. (1) can be rewritten as

$$\mathbf{P} = F(\Theta). \tag{2}$$

where Θ is the joint angle vector composed of $\theta_i (i = 1 \dots 6)$, which directly determines the posture of the manipulator’s links.

2.2. Posture solving based on Newton iteration

Toward the trajectory planning of the redundant manipulator under the given motion path, the key task is to divide the whole path into small steps and solve the optimal posture of manipulator links (i.e., the joint angle vector Θ) in each step.

Obviously, from Eq. (2) we can find that the partial derivative can be expressed as

$$\dot{\mathbf{P}} = J(\Theta)\dot{\Theta}. \tag{3}$$

where $\dot{\mathbf{P}}$ is the velocity matrix of the manipulator’s terminal center point, $\dot{\Theta}$ is the angular velocity matrix of the manipulator’s joints, $J(\Theta)$ is the 3×6 Jacobian matrix which describes the transformation relationship from Θ to $\dot{\mathbf{P}}$ under current joint angles Θ .

Since $J(\Theta)$ is not a positive definite matrix, the inverse matrix of $J(\Theta)$ cannot be solved directly. Define $J^+(\Theta)$ as the pseudoinverse matrix of $J(\Theta)$.

$$J^+(\Theta) = J^T(\Theta) (J(\Theta)J^T(\Theta))^{-1}. \tag{4}$$

From the work of Cheah et al. [37], it can be seen that the progressive stability and boundedness of $J^+(\Theta)$ can be guaranteed with the continuous recalculation of $J^+(\Theta)$ in the iteration process.

From Eqs. (3) and (4), the angular velocity matrix of the manipulator’s joints can be given as

$$\dot{\Theta} = J^+(\Theta)\dot{\mathbf{P}}. \tag{5}$$

The Newton iterative equation for solving Θ can be expressed as

$$\Theta_i = \Theta_{i-1} + J^+ (\Theta_{i-1}) \text{error}_{i-1}, \tag{6}$$

$$\text{error}_i = P_{\text{goal}} - P_i. \tag{7}$$

where error_i is the error function in the iterative calculation and is used to measure the gap between the target point P_{goal} and the current terminal center point P_i in the i th iteration. Compare $\|\text{error}_i\|_2$ with the threshold ξ to determine whether the iterative calculation result satisfies the required accuracy.

3. Trajectory planning based on WNI algorithm

3.1. Solution set for trajectory planning

Since the manipulator has redundant degrees of freedom, there are infinite number of posture solutions in its trajectory planning. After obtaining the solution set to the link posture with the given target coordinates, it is necessary to find the optimal solution satisfying the constraints and optimization objectives. The posture solution set can be obtained by

$$\Delta\Theta = J^+ (\Theta_{\text{NI}}) \Delta P + (I - J^+ (\Theta_{\text{NI}}) J (\Theta_{\text{NI}})) z, \tag{8}$$

$$\Theta_{\text{op}} = \Theta_{\text{origin}} + \Delta\Theta. \tag{9}$$

In Eq. (8), $\Delta\Theta$ is the variation of the manipulator’s joint angles, ΔP is the variation of the manipulator’s end effector coordinates, Θ_{NI} is the posture solution with Newton iteration, which is also the minimum norm solution of Eq. (2). $(I - J^+ (\Theta_{\text{NI}}) J (\Theta_{\text{NI}}))$ is link’s null-space projection matrix under the angles of Θ_{NI} . Given the characteristic of J^+ , $(I - J^+ (\Theta_{\text{NI}}) J (\Theta_{\text{NI}})) J^+ (\Theta_{\text{NI}}) = 0$, which means the two terms multiplied are orthogonal. In Eq. (9), Θ_{origin} is the original joint angle vector of the manipulator, Θ_{op} is the joint angle vector corresponding to the target coordinates of the end effector. The calculation result of Θ_{op} varies with the 6×1 arbitrary vector z , and the posture solution set of the target point for trajectory planning can be formed with different Θ_{op} .

Although the solution set can be obtained with the NI method, the optimal solution with best energy saving and motion stability control remains unknown. The optimal solution of Θ_{op} can be obtained by an optimized z during the whole process of trajectory planning. Therefore, a weighted Newton iteration (WNI) algorithm is proposed in this paper, which applies the NI algorithm to find the solution set and the heuristic method with a weight matrix for the joint motion to find the optimal posture solution.

3.2. Principle of WNI algorithm

In the process of calculating $\Delta\Theta$ by changing z , Θ_{NI} cannot remain the same when $\Delta\Theta$ changes, which badly influences the position accuracy of the end effector. It is necessary to propose a trajectory planning strategy, which combines Newton iteration and z -optimization to simultaneously satisfy the constraints and accuracy limits.

The energy consumed by the manipulator in a trajectory is used for driving the angular movements of the joints. Therefore, the energy consumption is closely related to the changes of the joint angles. A loss function is proposed and expressed as

$$Q = \sum_{i=1}^6 W_i \Delta\theta_i^2. \tag{10}$$

where W_i is the weight assigned to the squared angle variation $\Delta\theta_i$ of the i th joint to control the link motion, Q is the computed loss. To obtain the optimal energy saving solution, the weight value assigned to each joint should be determined by its energy consumption. Since the proximal joints consume more energy than the distal joints performing the same amount of angular movement, larger weights are

assigned to the joints closer to the base, and vice versa. Therefore, the angular changes of the proximal joints will be set smaller during the optimization of the loss function. Furthermore, an adjustment strategy for the weights is needed to guarantee stability when the joints approach the limits. When the working links and the corresponding joint weights are determined, the $\Delta \Theta = [\Delta \theta_1 \cdots \Delta \theta_6]^T$ that minimizes the loss function can be found, which is the optimal solution for calculating the posture Θ_{op} by Eq. (9).

Under the actual operation conditions of the 6-DOF large redundant manipulator, to ensure the operation accuracy and stability, the number of links moving simultaneously must not exceed four. The four distal links are often used as the main working links, due to their lighter weight and better controllability. In addition, the direction of joint motion should remain unchanged to prevent link vibrations caused by reciprocating motion. Under the constraints of the direction of joint motion and the number of working joints/links, there may be no posture that can reach the specific target position. In that case, the working links should be switched so that the trajectory planning can be continued. During the optimization of $\Delta \Theta$, the switch of the working links should also be performed when a working link reaches the motion boundary, as a result of the corresponding joint reaching its limit.

As mentioned above, it is possible that the state of the links needs to be switched between working and non-working. However, the states of all links in Eq. (5) are set to be the working state by default. Therefore, in the solving process for trajectory planning, Θ and its corresponding Jacobian matrix $J(\Theta)$ need to be constantly modified according to the current link states.

Let the serial numbers of working links be j, k, m and n , where $1 \leq j < k < m < n \leq 6$, then Θ and $J(\Theta)$ can be modified as

$$\Theta = [\theta_j \quad \theta_k \quad \theta_m \quad \theta_n]^T, \tag{11}$$

$$J(\Theta) = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_j} & \frac{\partial p_x}{\partial \theta_k} & \frac{\partial p_x}{\partial \theta_m} & \frac{\partial p_x}{\partial \theta_n} \\ \frac{\partial p_y}{\partial \theta_j} & \frac{\partial p_y}{\partial \theta_k} & \frac{\partial p_y}{\partial \theta_m} & \frac{\partial p_y}{\partial \theta_n} \\ \frac{\partial p_z}{\partial \theta_j} & \frac{\partial p_z}{\partial \theta_k} & \frac{\partial p_z}{\partial \theta_m} & \frac{\partial p_z}{\partial \theta_n} \end{bmatrix}. \tag{12}$$

In combination with Eqs. (8)–(12), the loss function Q can be described as a function of vector z . So, finding the optimal solution is equivalent to solving the following equation

$$z_{op} = \arg \min Q \tag{13}$$

Equation (13) can be transformed to

$$\frac{\partial Q}{\partial z} = 0. \tag{14}$$

It can be obtained from Eq. (14) that

$$z_{op} = A^{-1}B. \tag{15}$$

where $A = (I - J^+J)^T W (I - J^+J)$, $B = -(I - J^+J)^T W J^+ \Delta P$, $W = \begin{bmatrix} w_j & 0 & 0 & 0 \\ 0 & w_k & 0 & 0 \\ 0 & 0 & w_m & 0 \\ 0 & 0 & 0 & w_n \end{bmatrix}$ is a 4×4 weight matrix associated with the current working joints to calculate Q . The optimal solution of Θ_{op} can be obtained by substituting the optimized z_{op} in Eqs. (8) and (9).

The manipulator trajectory planning algorithm based on WNI is shown in Algorithm 1 and Fig. 2.

Algorithm 1: Manipulator trajectory planning algorithm based on WNI.

Θ_{origin} : joint angles before path planning
 Θ_{NI} : calculated joint angles after Newton iteration
 Θ_{op} : optimal solution to path planning
 W : weight matrix associated with the working joints
 $\|error\|_2$: $L2$ norm between the target point and the current terminal center point
 ξ : preset threshold
 Divide the working path into n_steps
 Input Θ_{origin}
 Determine the working direction of the joint
 Initialize weight matrix W
 While (step< n_steps) do
 Newton iteration to solve Θ_{NI}
 Apply the adaptive weighted method to obtain Θ_{op}
 If joint i exists reciprocating motion
 Keep the angle of joint i unchanged
 Continue
 End if
 If $\|error\|_2 \leq \xi$ is not satisfied
 Continue
 End if
 If the boundary constraints are not satisfied
 Switch working joints
 Regenerate W
 Continue
 End if
 Save Θ_{op}
 step += 1
 End While

3.3. Establishment of evaluation functions

Two evaluation functions E_1 and E_2 are proposed to intuitively compare the differences between various algorithms in terms of energy saving and motion stability control.

On the one hand, due to the long working distance and large mass of the redundant manipulator, it is desirable that the planning algorithm can complete the task with minimum energy consumption. In the working process, the motion of each link is controlled by the angular movement of a corresponding joint. The energy consumption of the manipulator between two different postures can be evaluated as the summed absolute change of the kinetic energy of all links. Therefore, the evaluation function for energy saving is expressed as

$$E_1 = \frac{1}{Z} \sum_{t=1}^N \sum_{i=1}^6 \left| \frac{1}{2} J_i \omega_{i,t}^2 - \frac{1}{2} J_i \omega_{i,t-1}^2 \right|, \tag{16}$$

where Z represents the length of the working path, $t(t = 1 \dots N)$ represents the t th step of the whole working path, $i(i = 1 \dots 6)$ represents the i th joint and link of the manipulator, J_i is the moment of inertia of link i , $\omega_{i,t}$ and $\omega_{i,t-1}$ are the angular velocities of joint i in different steps.

On the other hand, the manipulator’s motion stability is closely related to the jerks of the joints [38]. When the joint jerk is high, the motion of the joint would change abruptly, making the manipulator vibrate or even run out of control [39, 40]. The evaluation function for motion stability can be defined

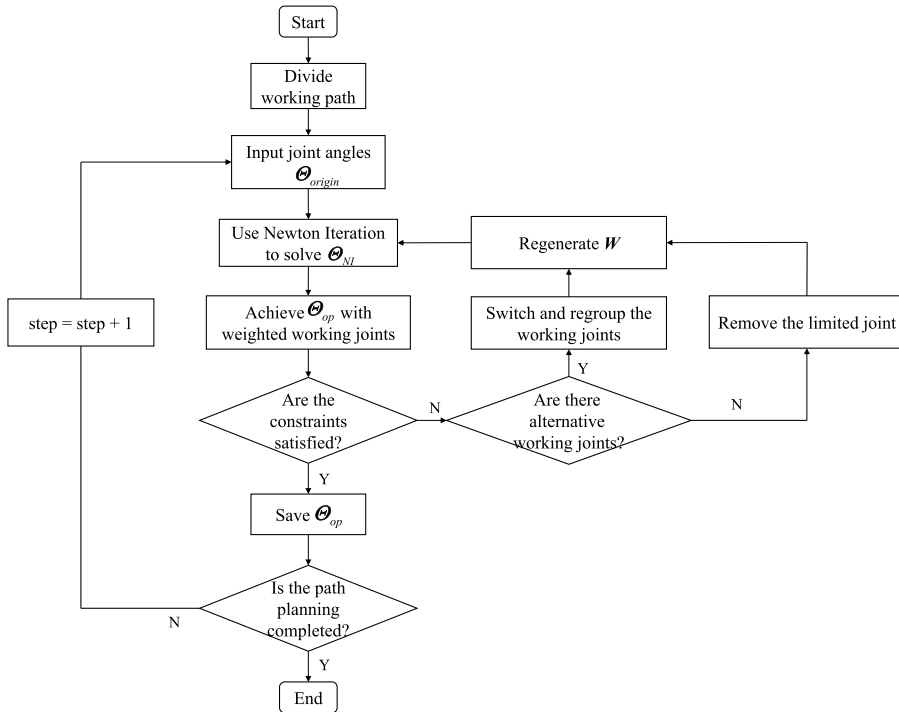


Figure 2. Flowchart of manipulator trajectory planning based on WNI algorithm.

by the maximum jerk the joints generate during the working process. Therefore, the evaluation function for motion stability control is expressed as

$$E_2 = \max p_i. \tag{17}$$

where p_i is the average angular jerk of joint i .

E_1 calculates the absolute changes of the manipulator’s kinetic energy in each step and then adds them up to evaluate the energy consumption during the working process, a higher E_1 indicates higher energy consumption. E_2 calculates the maximum angular jerk of all joints. In the working process, the most severe vibration occurs when the joint jerk is at its maximum, therefore a smaller E_2 indicates better motion stability.

When calculating E_1 and E_2 , what we know are the time spent on the joint motion in each step and the length of each step. So, the average angular velocity of the joint in each step is used to replace the instantaneous angular velocity at the step dividing point. And then, the joint angular acceleration and jerk can be obtained by the difference method.

4. Adaptive adjustment of the weight matrix

4.1. Configuration of weight matrix

The weight matrix W is determined by each joint’s position, motion state and distance from its limit. A constant weight matrix is difficult to adapt to different working conditions; therefore, it is necessary to recalculate W constantly and adaptively during the trajectory planning. The elements of W can be adjusted based on the current link posture as follows

$$W_i = \frac{T_i + S_i}{\frac{\theta_{boundary_j} - \theta_{origin_j}}{\theta_{boundary_j} - \theta_{threshold_j}}} = \frac{(T_i + S_i) (\theta_{boundary_j} - \theta_{threshold_j})}{(\theta_{boundary_j} - \theta_{origin_j})}, \quad i = 1 \dots 6. \tag{18}$$

where $\mathbf{T} = [T_1 \cdots T_6]$ is a 1×6 constant weight parameter vector, $\mathbf{S} = [S_1 \cdots S_6]$ is a 1×6 constant state parameter vector, $\theta_{\text{boundary}_i}$ is the limit to the angle of joint i , $\theta_{\text{threshold}_i}$ is the preset angle threshold of joint i (a certain distance from the angle limit).

As mentioned above, the large redundant manipulator cannot move all its links at the same time, and the four distal links are often used as the main working links. The closer a joint is to the base, the more difficult it is to drive it, and a larger weight should be assigned to it in the loss function. Therefore, in Eq. (18), the weight parameters $T_i (i = 1 \dots 6)$ are set to constant values according to the driving difficulty of the links. The state parameters $S_i (i = 1 \dots 6)$ are set according to the motion state of the link. The links should move slower during acceleration and deceleration than in the normal working state. So, S_i is set to a predefined positive number during acceleration and deceleration, and zero in the normal working state. According to Eq. (18), during the motion of the manipulator, the variation of \mathbf{W} is relatively small when the original joint angles Θ_{origin} differ greatly from the preset angle thresholds. After reaching the threshold, due to the property of the inverse proportional function in Eq. (18), the weight will increase rapidly, reducing the angular change of the joint until it stops (in the non-working state). By taking the current link posture into account, we can prevent the solution results from exceeding the joint limits. Meanwhile, the vibration caused by a sudden change in the links' motion state during acceleration and deceleration can be reduced by adaptively adjusting the weight matrix \mathbf{W} under different link postures.

4.2. Adaptive weighted Newton iteration (AWNI) algorithm

In trajectory planning, if the joint reciprocates, the corresponding link will violently vibrate and affect the control accuracy. Therefore, it is necessary to guarantee that the joint will not reciprocate during the entire motion. The direction of the joint motion should be determined as the working path is set. If the result in Eq. (8) is opposite to the default direction, reset the result as 0 and continue the iterative calculation until a valid solution is found.

As described above, set the angle threshold $\Theta_{\text{threshold}} = [\theta_{\text{threshold}_1} \cdots \theta_{\text{threshold}_6}]$ for all joints. When the angle of a joint is close to the corresponding angle threshold in $\Theta_{\text{threshold}}$, the actuated links should be switched, which means that the corresponding link should be set to non-working state and an originally non-working link should be actuated instead. Then \mathbf{W} should be reconstructed with the new set of working links. The trajectory planning algorithm continues with the new weight matrix \mathbf{W} .

In conclusion, by integrating the WNI algorithm in Section 3 and the weight matrix's adaptive adjusting method described above, the AWNI algorithm is proposed. The whole trajectory is divided into some sub-trajectories and planned step by step. At a certain sub-trajectory, assuming the original joint angle vector is Θ_{origin} , the corresponding weight matrix \mathbf{W} is constructed and the joint angle vector Θ_{op} corresponding to the target coordinates can be achieved by

$$\Theta_{\text{op}} = \Theta_{\text{origin}} + \arg \min \left(\sum W_i \Delta \theta_i^2 \right) \tag{19}$$

Algorithm 2 shows the adaptive adjustment of weight matrix \mathbf{W} and the joint/link motion switching in the AWNI algorithm.

5. Simulation and Validation

5.1. Configuration of parameters

As shown in Fig. 3, take a 6-DOF truck-mounted concrete pump manipulator as an example. The torsion angle α and base tangent length a in the D-H method are all set as 0. Table 1 shows the kinematic parameters of the manipulator.

Algorithm 2: Adaptive adjustment of weight matrix and joint/link motion switching in AWNI algorithm.

θ_i : angle of joint i
 $\theta_{\text{threshold}_i}$: preset angle threshold of joint i
 Θ : joint angle vector
 T_j : constant weight parameter of joint j
 S_j : constant state parameter of joint j
 W : weight matrix
 Working path planning begins
 For each $\theta_i, i = 1, 2 \dots 6$
 If $\theta_i > \theta_{\text{threshold}_i}$
 If alternate joint j exists
 Replace θ_i with θ_j in Θ
 Locate the corresponding T_j and S_j
 Calculate the weight value W_j of joint j with Eq. (18)
 Replace weight value W_i of joint i in W with W_j
 Else
 Remove θ_i from Θ
 Remove weight value W_i of joint i from W
 Continue working with the rest joints
 End if
 Else
 Recalculate the weight value W_i of joint i in W with Eq. (18)
 End if
 End for each
 Regenerate W

Table 1. Kinematic parameters of each link.

Link i	1	2	3	4	5	6
Length (l/mm)	11,400	9241	8761	11,085	8087	3457
Range ($\theta/^\circ$)	[0,90]	[0,180]	[0,180]	[0,240]	[0,210]	[0,110]

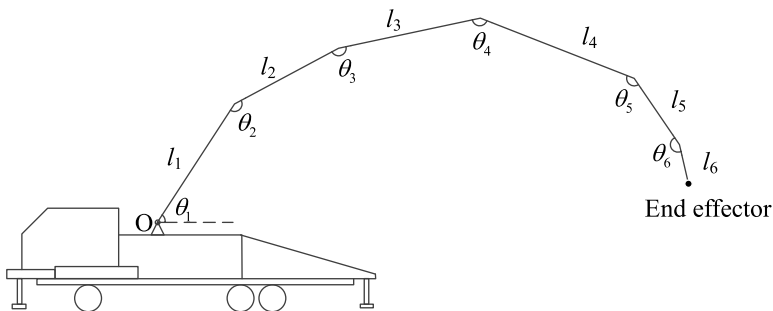


Figure 3. Diagram of the large redundant manipulator.

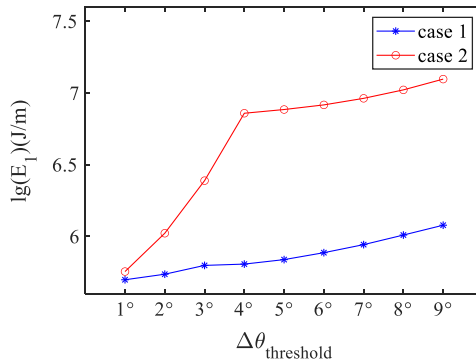


Figure 4. Generalized E_1 in nine groups.

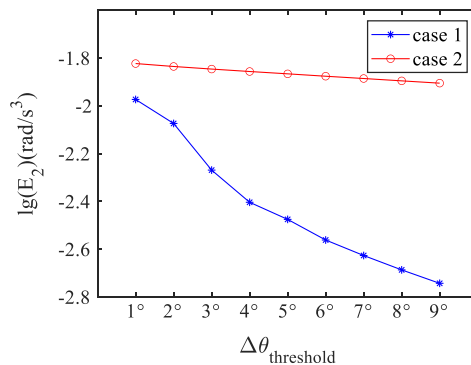


Figure 5. Generalized E_2 in nine groups.

5.2. Determining joint threshold

From Eq. (18), it can be seen that the threshold $\Theta_{\text{threshold}}$ must be settled before dynamic weight calculation. $\Theta_{\text{threshold}}$ reflects the working range of the links and has a great influence on W . Empirically, $\Theta_{\text{threshold}}$ should be close to Θ_{boundary} to prevent wasting too much workspace. To verify the effectiveness of $\Theta_{\text{threshold}}$ in the AWNI algorithm, define $\Theta_{\text{threshold}} = \Theta_{\text{boundary}} - \Delta\Theta_{\text{threshold}}$. A threshold descriptor $\Delta\theta_{\text{threshold}}$ is introduced to formulate $\Delta\Theta_{\text{threshold}}$ and nine groups of $\Delta\Theta_{\text{threshold}}$ are set, where $\Delta\Theta_{\text{threshold}} = V(\Delta\theta_{\text{threshold}})$, $\Delta\theta_{\text{threshold}} = 1^\circ, 2^\circ, \dots, 9^\circ$. $V(x)$ generates a 6×1 vector and each of its elements equals x . Generally, the distal links are more likely to reach the threshold, also, the range of the 4th and 5th joints both exceed 180° , therefore the 3rd and 6th joints are the appropriate experimental target joints. Respectively take the 3rd joint and 6th joint as tests, set the links' initial pose as $\Theta_{\text{initial}} = [70^\circ \ 168^\circ \ 168^\circ \ 208^\circ \ 178^\circ \ 98^\circ]^T$, then plan the trajectory towards positive Y direction. During each test, to verify the effectiveness of $\Theta_{\text{threshold}}$ in trajectory planning and avoid the influence of the other joints, calculate the weights of the target joints with Eq. (18) and the weights of the other joints remain constant. In case 1, the 6th joint is chosen as target joint, while in case 2, the 3rd joint is chosen as target joint. The results of E_1 and E_2 are shown in Figs. 4 and 5. It can be seen that the increase of $\Delta\theta_{\text{threshold}}$ causes the increase of E_1 and the decrease of E_2 , which means higher energy consumption and better motion stability. In the nine groups, a relatively smaller $\Delta\theta_{\text{threshold}}$ means that the movable range of the joint beyond the angle threshold is smaller, so the proximal joint will start earlier, which is detrimental to the energy saving of the manipulator. Meanwhile, the joints' deceleration generates in the range of $\Delta\theta_{\text{threshold}}$; therefore, a larger $\Delta\theta_{\text{threshold}}$ is more suitable considering the motion smoothness.

Consequently, the energy consumption of the manipulator increases when $\Delta\theta_{\text{threshold}}$ becomes larger, which will lead to higher work costs. And the motion stability of the joint gets poor when $\Delta\theta_{\text{threshold}}$

Table 2. Joints' angle variety results in the simulation experiment.

Case	Algorithm	Joint #1 angle variation/°	Joint #2 angle variation/°	Joint #3 angle variation/°	Joint #4 angle variation/°	Joint #5 angle variation/°	Joint #6 angle variation/°
3	NI	16.4	2.6	12.6	16.7	14.9	4.8
	WNI	11.1	0.5	8.4	16.4	22.9	15.8
	AWNI	0	18.9	7.9	22.2	38.6	9.1
4	NI	12.0	2.6	12.4	16.2	11.8	1.9
	WNI	7.5	1.5	10.4	18.3	18.9	2.3
	AWNI	0	0	7.2	21.6	28.9	5.8
5	NI	9.3	7.3	4.5	1.8	1.0	1.1
	WNI	7.9	7.3	5.6	2.8	2.8	5.6
	AWNI	0	0	22.7	4.6	27.3	25.8
6	NI	14.2	1.7	3.0	4.4	2.2	0.9
	WNI	8.0	7.1	5.2	2.3	2.7	5.5
	AWNI	0	0	25.4	14.9	52.0	6.9

become smaller, which will affect the working accuracy. Under normal conditions, the working process of the manipulator should consider both energy consumption and motion stability. To sum up, the experiment results show that the appropriate value of $\Delta\theta_{\text{threshold}}$ is between 4° and 6° .

5.3. Verification of AWNI algorithm

With the joints' initial angle $\Theta_{\text{initial}} = [75^\circ \ 140^\circ \ 150^\circ \ 150^\circ \ 130^\circ \ 90^\circ]^T$, the initial end point's coordinate can be calculated to be (28,048, 3685, 0). Assume that the working path is as follows: (1) 10,000 mm along the X axis; (2) -10,000 mm along the X axis; (3) 10,000 mm along the Y axis, and (4) -10,000 mm along the Y axis. Divide the working path into sub-trajectories with a step size of 100 mm, and the starting position of each sub-trajectory is the current pose and the ending point is the desired target point of the end effector. To prove the advantages of dynamically adjusting the weight matrix, we set an experimental group with a constant weight matrix. Separately apply (a) NI, (b) WNI with the weight matrix, with $W_1 = 6$, $W_2 = 5$, $W_3 = 4$, $W_4 = 3$, $W_5 = 2$, $W_6 = 1$, and (c) AWNI to plan the links' posture on the path. Set Newton iteration threshold $\xi = 0.1$ mm. The links' posture varieties are compared in Fig. 6.

As shown in Fig. 6 and Table 2, six links/joints work at the same time following the NI algorithm planning result. After adding the weighted optimization of the loss function in the trajectory planning, the WNI algorithm still drives six links/joints to complete the trajectory planning tasks, but it relies more on the distal joints. When applying the AWNI algorithm, distal links/joints #3-#6 work firstly, if a working link approaches its joint angle limit, it can be replaced by another alternative link/joint (in case 3, link/joint #6 is replaced by link/joint #2).

As shown in Figs. 7, 8 and Table 3, the trajectory planning result of NI algorithm leads to high energy consumption and poor motion stability, so it is not recommended in engineering practice. The WNI algorithm manages to plan the trajectory with a relatively smaller energy consumption. But the constant weight matrix does not consider the joints' pose, the jerk increases sharply when a joint approaches the limit. Compared to WNI, the weight matrix of AWNI algorithm is adaptively adjusted during the planning process according to the kinematic constraints and energy consumption of the manipulator. Evaluation functions E_1 and E_2 validate that the AWNI algorithm has less energy consumption and better motion stability than NI and WNI algorithms. All the results above have demonstrated that the performance of AWNI is superior to that of NI and WNI algorithms.

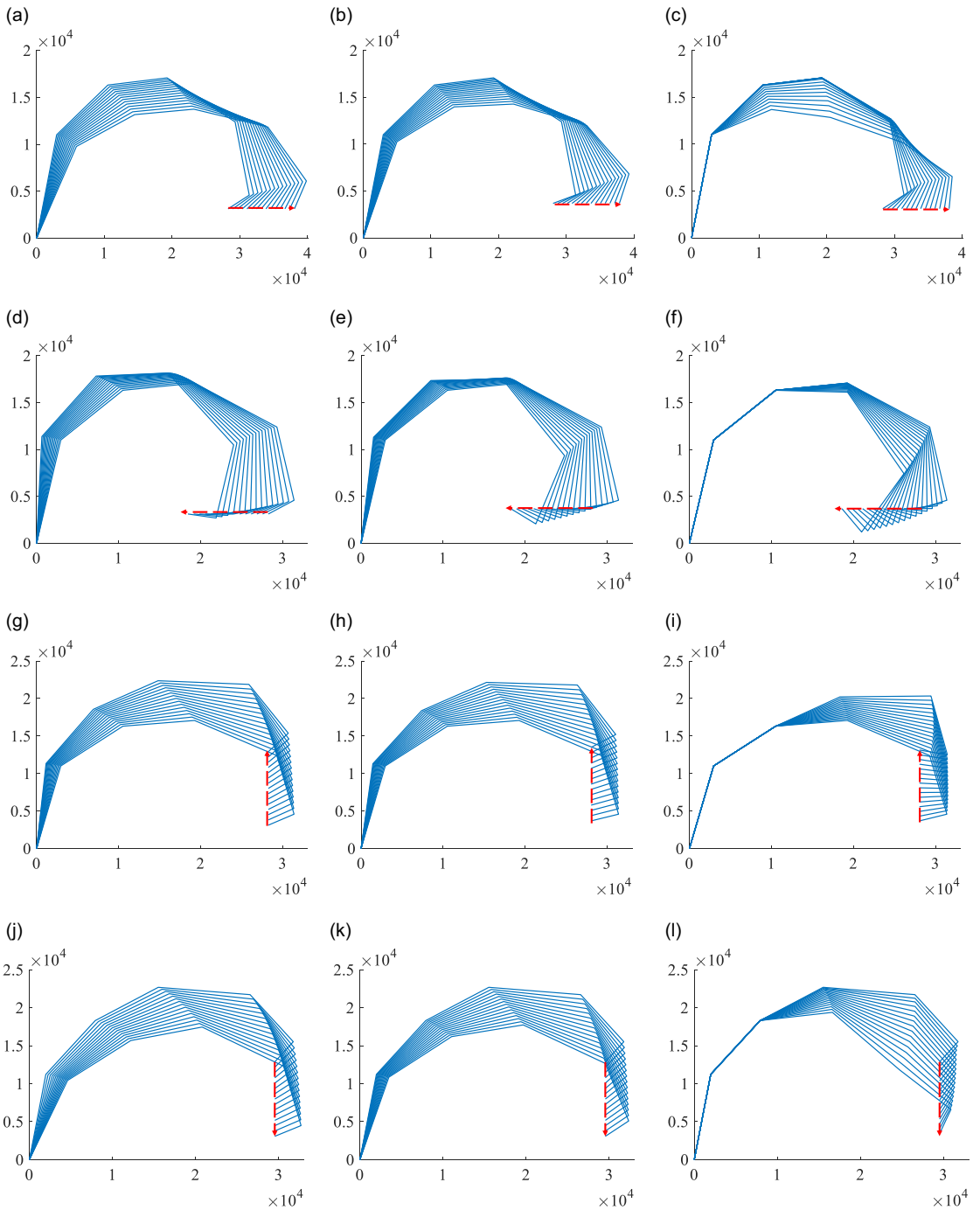


Figure 6. Planned trajectory in case algorithm: (a) NI in case 3, (b) WNI in case 3, (c) AWNI in case 3, (d) NI in case 4, (e) WNI in case 4, (f) AWNI in case 4, (g) NI in case 5, (h) WNI in case 5, (i) AWNI in case 5, (j) NI in case 6, (k) WNI in case 6 and (l) AWNI in case 6.

Table 3. E_1, E_2 in cases 3, 4, 5 and 6.

Case	Evaluation function	NI	WNI	AWNI
3	E_1 /(J/m)	1.03×10^6	7.98×10^5	1.48×10^5
	E_2 /(rad/s ³)	2.61×10^{-2}	1.31×10^{-2}	7.85×10^{-3}
4	E_1 /(J/m)	3.93×10^4	6.11×10^3	1.59×10^3
	E_2 /(rad/s ³)	8.71×10^{-3}	8.62×10^{-4}	7.34×10^{-4}
5	E_1 /(J/m)	1.93×10^5	1.09×10^5	5.94×10^3
	E_2 /(rad/s ³)	3.85×10^{-2}	1.65×10^{-2}	4.24×10^{-3}
6	E_1 /(J/m)	1.83×10^5	9.67×10^4	7.72×10^3
	E_2 /(rad/s ³)	7.13×10^{-2}	2.62×10^{-2}	5.79×10^{-3}

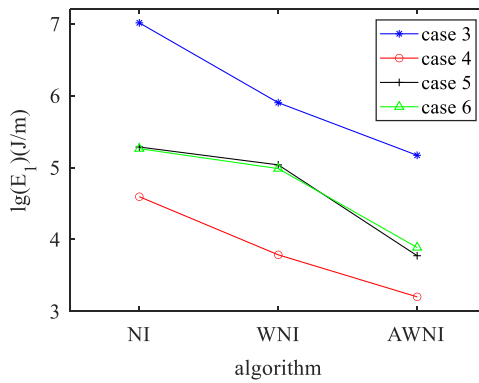


Figure 7. Generalized E_1 trend in cases 3, 4, 5 and 6.

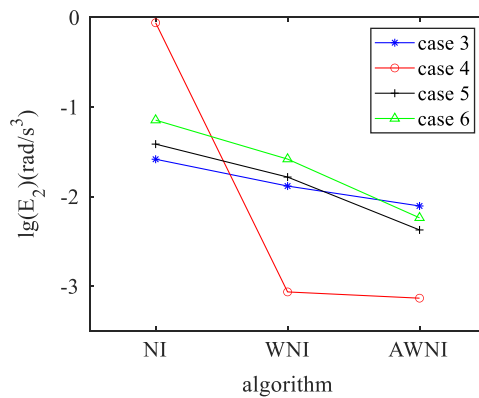


Figure 8. Generalized E_2 trend in cases 3, 4, 5 and 6.

6. Conclusions

This study combines Newton iteration with an adaptive weighted method and proposes a trajectory planning algorithm AWNI with consideration of energy saving and motion stability control for large redundant manipulators. Such objectives are achieved by optimizing a loss function where different weights are assigned to the working joints according to their energy consumption. The weight matrix of the AWNI algorithm is adjusted adaptively during the trajectory planning process for three benefits: (1) to prevent reciprocating motion; (2) to ensure the motion of the links is within the boundaries, and;

(3) to improve performance in terms of energy consumption and motion stability. The working joints are replaced by alternative joints when they reach the limits. Two evaluation functions E_1 and E_2 are designed to compare the energy consumption and motion stability of different algorithms. The effect of threshold descriptor $\Delta\theta_{\text{threshold}}$ is studied, and the results show that the appropriate range of $\Delta\theta_{\text{threshold}}$ is between 4° and 6° . Trajectory planning in four directions is carried out with NI, WNI and AWNI algorithms, respectively. The results show that the AWNI algorithm outperforms NI and WNI algorithms as far as E_1 and E_2 are concerned.

The future works include improving the convergence performance of the AWNI algorithm and proposing a better generating method of the weight matrix W .

Author contribution. Daxin Liu conceived and designed the study and its methodology. Yu Huang conducted simulations, interpreted the results and prepared this article manuscript under supervision by Zhenyu Liu and Daxin Liu. Zhenyu Liu, Daxin Liu and Jianrong Tan reviewed and edited the manuscript. Xuxin Guo and Ke Wang helped with the simulation and validation.

Financial support. This work was supported by the National Key Research and Development Program of China (Grant No. 2019YFB1312600), the National Natural Science Foundation of China (Grant No. 52075480) and the Key Research and Development Program of Zhejiang Province (Grant No. 2021C01008).

Competing interests. The authors declare no competing interests exist.

Ethical approval. Not applicable.

References

- [1] M. Ahmadzadeh, A. M. Shafei and M. Fooladi, "Dynamic analysis of multiple inclined and frictional impact-contacts in multi-branch robotics systems," *Appl. Math. Model.* **91**, 24–42 (2021).
- [2] A. Agarwal, S. V. Shah, S. Bandyopadhyay and S. K. Saha, "Dynamics of serial kinematic chains with large number of degrees-of-freedom," *Multibody Syst. Dyn.* **32**(3), 273–298 (2014).
- [3] V. Mata, S. Provenzano, F. Valero and J. I. Cuadrado, "Serial-robot dynamics algorithms for moderately large numbers of joints," *Mech. Mach. Theory* **37**(8), 739–755 (2002).
- [4] J. M. Hollerbach, "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Trans. Syst. Man Cybern.* **10**(11), 730–736 (1980).
- [5] K. S. Anderson and J. H. Critchley, "Improved 'Order-N' performance algorithm for the simulation of constrained multi-rigid-body dynamic systems," *Multibody Syst. Dyn.* **9**(2), 185–212 (2003).
- [6] P. Boscarol and D. Richiedei, "Energy-efficient design of multipoint trajectories for Cartesian robots," *Int. J. Adv. Manuf. Technol.* **102**(5-8), 1853–1870 (2019).
- [7] S. Katiyar and A. Dutta, "Dynamic path planning over CG-Space of 10DOF Rover with static and randomly moving obstacles using RRT* rewiring," *Robotica* **40**(8), 2610–2629 (2022).
- [8] H. Wang, H. Wang, J. H. Huang, B. Zhao and L. Quan, "Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve," *Mech. Mach. Theory* **139**, 284–293 (2019).
- [9] B. H. Cai and Y. N. Zhang, "Equivalence of velocity-level and acceleration-level redundancy-resolution of manipulators," *Phys. Lett. A* **373**(38), 3450–3453 (2009).
- [10] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access* **2**, 56–77 (2014).
- [11] C. J. Lin, C. K. Chen and C. L. Chen, "Path planning and dynamic control of a redundant robot manipulator for conveyor tracking," *Int. J. Syst. Sci.* **30**(5), 491–503 (1999).
- [12] M. D. Marcos, J. A. T. Machado and T. P. Azevedo-Perdicoulis, "An evolutionary approach for the motion planning of redundant and hyper-redundant manipulators," *Nonlinear Dyn.* **60**(1-2), 115–129 (2010).
- [13] P. Savsani, R. L. Jhala and V. J. Savsani, "Comparative study of different metaheuristics for the trajectory planning of a robotic arm," *IEEE Syst. J.* **10**(2), 697–708 (2016).
- [14] K. K. Aytan, M. N. Sahinkaya and A. Dumlu, "Real time optimum trajectory generation for redundant/hyper-redundant serial industrial manipulators," *Int. J. Adv. Robot. Syst.* **14**(6), 1729881417737241 (2017).
- [15] H. Potter, J. Kern, G. Gonzalez and C. Urrea, "Energetically optimal trajectory for a redundant planar robot by means of a nested loop algorithm," *Elektron. Elektrotech.* **28**(2), 4–17 (2022).
- [16] Y. Li, Z. Wang, H. Yang, H. Zhang and Y. Wei, "Energy-optimal planning of robot trajectory based on dynamics," *Arab. J. Sci. Eng.* **48**(3), 3523–3536 (2023).
- [17] B. Zhang, P. Xiao, H. Liu, M. Ma, Q. Wang and Z. Mao, "Quadrotor convergence trajectory optimization model based on minimum energy consumption," *Adv. Appl. Math. Mech.* **10**(3), 673–689 (2018).
- [18] Q. C. Pham and O. Stasse, "Time-optimal path parameterization for redundantly actuated robots: A numerical integration approach," *IEEE-ASME Trans. Mechatron.* **20**(6), 3257–3263 (2015).

- [19] T. Su, X. Liang, X. Zeng and S. Liu, "Pythagorean-Hodograph curves-based trajectory planning for pick-and-place operation of Delta robot with prescribed pick and place heights," *Robotica* **41**(6), 1651–1672 (2023).
- [20] L. M. Shen and Y. M. Wen, "Investigation on the discretization of a repetitive path planning scheme for redundant robot manipulators," *IEEE Access* **8**, 23895–23903 (2020).
- [21] Y. H. Pei, Z. F. Liu, J. J. Xu and C. B. Yang, "Minimum-Time Trajectory Planning for a 4-DOF Manipulator Considering Motion Stability and Obstacle avoidance," *In: 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)* (2020) pp. 203–207.
- [22] M. Elshabasy, K. T. Mohamed and A. A. Ata, "Power optimization of planar redundant manipulator moving along constrained-end trajectory using hybrid techniques," *Alex. Eng. J.* **56**(4), 439–447 (2017).
- [23] V. Agarwal, "Trajectory planning of redundant manipulator using fuzzy clustering method," *Int. J. Adv. Manuf. Technol.* **61**(5-8), 727–744 (2012).
- [24] H. Cao, S. Sun, K. Zhang and Z. Tang, "Visualized trajectory planning of flexible redundant robotic arm using a novel hybrid algorithm," *Optik* **127**(20), 9974–9983 (2016).
- [25] M. Jin and D. Wu, "Collision-free and energy-saving trajectory planning for large-scale redundant manipulator using improved PSO," *Math. Probl. Eng.* **2013**, 1–8 (2013).
- [26] X. D. Kang, G. Huang, X. L. Cao and X. Zhou, "Trajectory planning for concrete pump truck based on intelligent hill climbing and genetic algorithm," *Appl. Mech. Mater.* **127**, 360–367 (2012).
- [27] M. D. Marcos, J. A. T. Machado and T. P. Azevedo-Perdicoulis, "Trajectory planning of redundant manipulators using genetic algorithms," *Commun. Nonlinear Sci. Numer. Simul.* **14**(7), 2858–2869 (2009).
- [28] C. C. Lin, "Hierarchical path planning for manipulators based on genetic algorithm with non-random initial population," *Int. J. Innov. Comput. Inf. Control* **8**(3A), 1773–1786 (2012).
- [29] A. Shrivastave and V. K. Dalla, "Jerk optimized motion planning of redundant space robot based on grey-wolf optimization approach," *Arab. J. Sci. Eng.* **48**, 2687–2699 (2022).
- [30] M. Safeea, R. Bearee and P. Neto, "Collision avoidance of redundant robotic manipulators using newton's method," *J. Intell. Robot. Syst.* **99**(9), 673–681 (2020).
- [31] G. Song, S. Su, Y. Li, X. Zhao, H. Du, J. Han and Y. Zhao, "A closed-loop framework for the inverse kinematics of the 7 degrees of freedom manipulator," *Robotica* **39**(4), 572–581 (2021).
- [32] A. Reiter, A. Muller and H. Gattringer, "On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators," *IEEE Trans. Ind. Inform.* **14**(4), 1680–1690 (2018).
- [33] T. Nammoto and K. Kosuge, "An analytical solution for a redundant manipulator with seven degrees of freedom," *Int. J. Autom. Smart Technol.* **2**(4), 339–346 (2012).
- [34] D. Zhou, L. Ji, Q. Zhang and X. Wei, "Practical analytical inverse kinematic approach for 7-DOF space manipulators with joint and attitude limits," *Intell. Serv. Robot.* **8**(4), 215–224 (2015).
- [35] W. H. Liu, D. S. Chen and J. C. Steil, "Analytical inverse kinematics solver for anthropomorphic 7-DOF redundant manipulators with human-like configuration constraints," *J. Intell. Robot. Syst.* **86**(1), 63–79 (2017).
- [36] V. F. Tarasov, "Multipole matrix-elements for DH-systems and their asymptotics," *Int. J. Mod. Phys. B* **9**(20), 2699–2718 (1995).
- [37] C. C. Cheah, S. Kawamura and S. Arimoto, "Feedback control for robotic manipulator with an uncertain Jacobian matrix," *J. Robot. Syst.* **16**(2), 119–134 (1999).
- [38] S. Kang, K. Komoriya, K. Yokoi, T. Koutoku and K. Tanie, "Reduced Inertial Effect in Damping-Based Posture Control of Mobile Manipulator," *In: IEEE/RSJ International Conference on Intelligent Robots & Systems* (2002) pp. 488–493.
- [39] G. Lee, S. Park, D. Lee, F. C. Park, J. I. Jeong and J. Kim, "Minimizing energy consumption of parallel mechanisms via redundant actuation," *IEEE/ASME Trans. Mechatron.* **20**(6), 2805–2812 (2015).
- [40] R. Bearee and A. Olabi, "Dissociated jerk-limited trajectory applied to time-varying vibration reduction," *Robot. Comput. Integr. Manuf.* **29**(2), 444–453 (2013).