

Article

# Trajectory Planning and Optimization for a Par4 Parallel Robot Based on Energy Consumption

Xiaoqing Zhang <sup>1,2</sup>  and Zhengfeng Ming <sup>1,\*</sup><sup>1</sup> School of Mechano-Electronic Engineering, Xidian University, Xi'an 710000, China<sup>2</sup> School of Physics and Electronic Engineering, Xianyang Normal University, Xianyang 712000, China

\* Correspondence: mingzf@xidian.edu.cn

Received: 7 May 2019; Accepted: 4 July 2019; Published: 9 July 2019



**Abstract:** A study on trajectory planning and optimization for a Par4 parallel robot was carried out, based on energy consumption in high-speed picking and placing. In the end-effector operating space of the Par4 parallel robot, the rectangular transition of the pick-and-place trajectory was rounded by a Lamé curve. A piecewise design method was adopted to accomplish trajectory shape planning for displacement, velocity and acceleration. To make the Par4 robot's end run more smoothly and to reduce residual vibration, asymmetric fifth-order and sixth-order polynomial motion laws were employed. With the aim of reaching the minimum mechanical energy consumption for the Par4 parallel robot, the recently proposed Grey Wolf Optimizer (GWO) algorithm was adopted to optimize the planning trajectory. The validity of the design method was verified by experiments, and it was found that the minimum mechanical energy consumption of the optimal trajectory planned under the law of fifth-order polynomial motion is lower than that of sixth-order polynomial motion. In addition, the experiments also revealed the optimal values of Parameters  $e$  and  $f$ , which were the parameters of the Lamé curve function. Parameter  $e$  can be calculated as half the pick-up span for the minimum mechanical energy consumption, unlike parameter  $f$ , whose optimal value depends on specific circumstances such as the pick-and-place coordinates and the pick-up height.

**Keywords:** parallel robot; trajectory planning; mechanical energy consumption; Grey Wolf Optimizer; polynomial motion

## 1. Introduction

In recent years, parallel robots have been widely developed for their optimal dynamic properties [1,2], especially parallel robots with lower mobility such as Delta, H4, X4, and Par4. [3,4]. At present, parallel robots are widely used in high-precision fields [5–7] such as aviation manufacturing, electronic packaging, and 3D printing.

A reasonable trajectory planning of a robot can help it not only to perform the desired tasks with satisfactory performance [8], but it is also conducive to improving efficiency [9], reducing energy consumption, and extending the operating life [10,11]. At present, there are many studies on robot trajectory planning, such as in [12] (point-to-point dynamic trajectory planning was carried out for six degree-of-freedom (six-DOF) parallel robots, attaining good effects), [13] (point-to-point dynamic trajectory planning was carried out for three degree-of-freedom (three-DOF) parallel robots, also attaining good effects), and [14] (a new  $s - \ddot{s}$  plane path method was adopted to design a dynamic feasible point-to-point path and periodic trajectories, based on the forces of geometric constraints).

In general, robot trajectory planning can be either on the end-effector or on the joint level [15]. If it is done on the joint level, the end-effector trajectory of the robot should be mapped to the joint motion trajectory based on inverse kinematics [16], then only the joint motion needs to be controlled, and the result of the motion planning is only evaluated at each joint [17]. If done on the end-effector

level, the motion planning is directly on the end-effector, and the evaluation and control are all applied on the end-effector [18]. In [19], trajectory planning for a Delta parallel robot was carried out on the end-effector and on the joint level, respectively. Through comparative analysis, it was found that the trajectory planning on the joint level could effectively reduce the maximum torque of the motor and the vibration amplitude of the robot end in the movement process. In view of this, from the lower rated torque of the motor and the lower vibration amplitude of the robot end of view, in this paper the trajectory planning of a Par4 parallel robot will be carried out on the joint level, i.e., first, we will design the shape of movement track and the motion law on the end-effector level, then, the motion feature of the end-effector will be mapped to the joint space, and the follow-up evaluation and control will all be conducted at each joint.

Trajectory planning and trajectory optimization are often carried out simultaneously, and the common optimization objectives are minimum time [20], minimum energy [21], minimum pulsation [22], and maximum load-carrying capacity [23]. The minimum time optimization, directly associated with the production efficiency of the robot, was the first of these objectives to be considered, having been widely applied. For instance, in [24], considering three different types of constraints, smooth joint trajectories were constructed with the cubic polynomial function, a genetic algorithm procedure with parallel-populations was applied for the purpose of minimizing time, and finally, the validity of the method was verified; in [25], a time-optimal trajectory planning method was proposed, based on quintic Pythagorean Hodograph (PH) curves, in order to realize a smooth and high-speed operation of a Delta parallel robot. If we blindly pursue the minimum execution time at actual operation, it will easily lead to the discontinuity of torque and acceleration. Energy consumption is also one of the most concerning indexes of parallel robots, and especially when the energy source of the robot system is limited, it is more meaningful to study the minimum energy consumption than other optimization objectives. At present, some researchers have done trajectory planning for robots with minimal energy consumption. For example, in [26], a method of trajectory planning with minimum energy damping was proposed for a flexible-base robot, considering the point-to-point motion task. The maximum residual vibration amplitude and operating energy were taken as objective functions, and the radial basis function networks (RBFNs) were optimized using non-dominated sorting genetic algorithms-II (NSGA-II). In [27], a design method based on the concept of universal redundancy was proposed to improve the dynamic performance of the mechanism. After the servo motor's motion path was divided into multiple segments, the trajectory planning of each segment was designed, respectively, applying the law of polynomial motion. Some polynomial parameters, as well as the speed and acceleration of the working point, should be selected with the minimal optimization for energy and vibration force, based on the genetic algorithm. The validity of the method was verified by experiments, but since there were many parameters to be optimized, it was easy to fall into the local optimum. An underactuated self-reconfigurable robot with active and passive joints was designed in [28], and reconfiguration planning and scheduling were carried out with the aim of minimizing energy consumption based on the genetic algorithm. In [29], three trajectories were designed based on a model of an underwater robot, based on minimum energy, minimum turbulence, and minimum torque change rate, respectively. Finally, the fourth trajectory with minimal energy based on the genetic algorithm was planned, in which a polynomial approximates the time histories of the trajectory in joint space; the running time and the coordinates of an intermediate point were taken as the object to be optimized, and the quartic polynomial was selected as its motion law. The advantage of this method was that it was not reliant on an accurate mathematical model; however, since the intermediate point was obtained by random search, the shape of the trajectory could not be well controlled. At present, to the best of our knowledge, no trajectory planning involving energy consumption has yet been elaborated for a Par4 parallel robot. In this paper, in order to study the trajectory planning and energy consumption of the Par4 parallel robot, the minimum energy consumption was taken as the optimized object.

High-speed pick-and-place operation is one of the main working modes of parallel robots [30,31], and a gate-shaped trajectory is usually adopted, as shown in Figure 1. In the figure, A and H are the

pick-and-place endpoints, B, C, E, F and G the intermediate points on the pick-and-place track,  $H_a$  is the vertical displacement, also known as pick-up height, and  $B_a$  is the horizontal displacement, also known as pick-up span. The trajectory is symmetric with respect to the dot line MN.

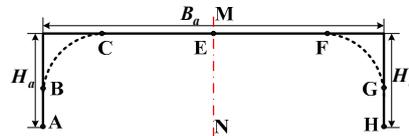
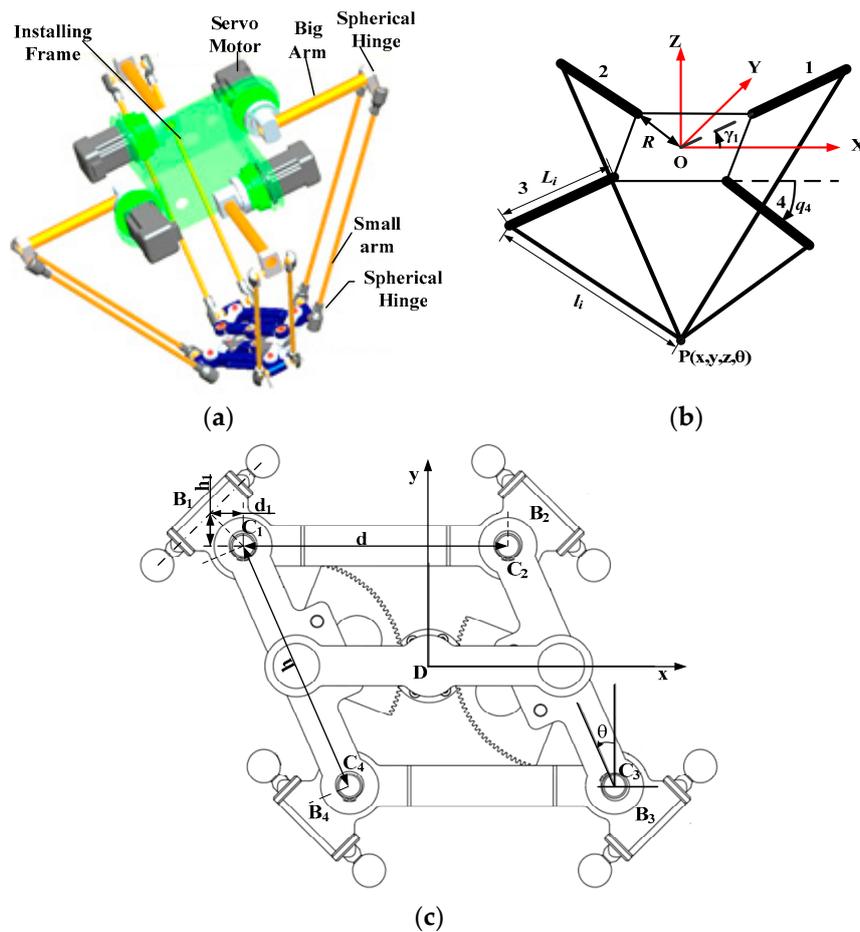


Figure 1. Motion trajectory.

A polynomial of third order or more, a modified trapezoid and Bang-Bang algorithms are usually applied as the law of motion [32,33]. Due to the right-angle connection of the conventional gate trajectory (as shown by the solid line in Figure 1), it is easy to cause the discontinuity of the motor’s speed and acceleration in the process of movement, which leads to severe mechanical vibration in the robot. Therefore, a smooth transition is often applied instead of the right-angle connection, as shown by the dotted line in Figure 1. The ideal transition curve should meet the criterion of continuous curvature. In [34], taking the five-bar mechanism as the object, a variety of motion trajectories were compared and analyzed, and the results show that the Lamé curve is better than the others in terms of residual vibration and the precision of the final position. However, there was no optimization research about the trajectory with the Lamé curve in [34]. Based on this, in this paper, for the design of trajectory planning and optimization of the Par4 parallel robot, the Lamé curve was calculated for a smooth transition of the pick-and-place path; some comparisons were made between the fifth- and sixth-order polynomial motion laws; and Grey Wolf Optimizer (GWO) was selected for optimizing the Lamé curve parameters with minimum mechanical energy consumption.

## 2. Par4 Parallel Robot

The structure of the 4-Dof Par4 parallel robot is shown in Figure 2a, consisting of three parts: (1) A fixed platform mainly composed of the mounting frame and the servo motor and reducer component; (2) a moving platform where the actuator is located; (3) four symmetric branched chains connecting the big arm and small arm by a spherical hinge. The simplified structure diagram for the Par4 robot is shown in Figure 2b, in which the moving platform has been simplified to Particle P. The OXYZ coordinate system is established on the fixed platform, where O is the fixed platform center. In Figure 2b,  $L_i$  and  $l_i$  are the lengths of the big arm and small arm, respectively, R is the distance from the origin O to the big arm revolute center, which is called the big arm revolute radius for short in the following article,  $q_i(i = 1, 2, 3, 4)$  is the rotation angle of each big arm (only  $q_4$  is marked in Figure 2b), and  $\gamma_i(i = 1, 2, 3, 4)$  is the angle from the axial direction from each big arm base to the X axis ( $\gamma_i = (2i - 1)\pi/4$ ). The pose of the moving platform (Particle P) is described as  $(x, y, z, \theta)$ , where  $x, y$  and  $z$  are the coordinate values in the OXYZ system, and  $\theta$  is the rotation angle of the moving platform, as shown in Figure 2c which is the top view of the end-effector. For easier description, the local reference coordinate xDy system is established and is shown in Figure 2c, where D is the moving platform center. In Figure 2c,d,h are the moving platform lengths along the x and y axes, respectively;  $B_i$  is the center of the support where the two ball hinges connecting the  $i$ th small arm with the moving platform are located; the moving platform can be considered a hinged parallelogram, as shown in Figure 2c;  $C_i$  is the pivot of its  $i$ th hinge, which can be considered as the  $i$ th vertex of the hinged parallelogram;  $d_i$  is the distance along the x axis between  $B_i$  and  $C_i$ ; and  $h_i$  is the distance along the y axis between  $B_i$  and  $C_i$ , here  $i = 1, 2, 3, 4$ .



**Figure 2.** Structure of the Par4 parallel robot. (a) Par4 parallel robot. (b) Structure sketch of the robot. (c) Top view of the end-effector.

### 2.1. Inverse Kinematics of the Par4 Parallel Robot

According to the Par4 parallel robot structure, the inverse kinematic rotation angle of each big arm can be obtained as shown in Equation (1):

$$q_i = 2\arctan\left(\frac{-I_i - \sqrt{I_i^2 - K_i^2 + J_i^2}}{K_i - J_i}\right), \quad i = 1, 2, 3, 4. \quad (1)$$

Here,  $I_i = L_i z$ ,  $K_i = (a_i^2 + b_i^2 + L_i^2 - l_i^2 + z^2)/2$ ,  $J_i = -L_i(a_i \cos \gamma_i + b_i \sin \gamma_i)$ ,  $a_i = x - \varepsilon_i h \sin \theta + \xi_i d/2 + \zeta_i d_i - R \cos \gamma_i$ ,  $b_i = y + \varepsilon_i h \cos \theta + \sigma_i h_i - R \sin \gamma_i$ ,  $\varepsilon = [\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4] = [1, 1, -1, -1]$ ,  $\xi = [\xi_1, \xi_2, \xi_3, \xi_4] = [1, -1, -1, 1]$ ,  $\zeta = [\zeta_1, \zeta_2, \zeta_3, \zeta_4] = [1, -1, -1, 1]$ , and  $\sigma = [\sigma_1, \sigma_2, \sigma_3, \sigma_4] = [1, 1, -1, -1]$ .

Given the spatial trajectory of the robot end-effector, the angular displacement of each driving mechanism can be obtained through the inverse kinematic equations, as in Equation (1). Accordingly, if the robot is driven with this obtained angular displacement law, the expected end-effector spatial trajectory will be achieved [35].

### 2.2. Dynamic Equation of the Par4 Parallel Robot

First, some simplifications of the system structure have been done, i.e., (1) the mass of the big arm ( $m_b$ ) is simplified to the geometric center of the big arm; (2) the small arm is equivalent to a bar, and the mass of the two rods of the small arm ( $m_s$  and  $m_s$ ) are equivalent to both ends of the bar. Regardless of



of IE is half the pick-up span  $B_a$ . Setting up the local plane coordinate system  $\mathbf{UO}'\mathbf{W}$ , the geometric meaning of parameters  $e$  and  $f$  in Equation (4) are shown in Figure 3.  $\varphi$  is the turned angle from the starting point B to the point  $k_i(u_i, w_i)$  on the Lamé curve, which is noted clearly in Figure 3. The turned angle from the starting point B to the point C is  $\pi/2$ .

The arc length of the curve from B to C is

$$l_{BC} = \int_0^{\frac{\pi}{4}} \frac{(1 + \tan^2 \varphi) \sqrt{e^2 \tan^4 \varphi + f^2}}{(1 + \tan^3 \varphi)^{4/3}} d\varphi + \int_0^{\frac{\pi}{4}} \frac{(1 + \tan^2 \varphi) \sqrt{f^2 \tan^4 \varphi + e^2}}{(1 + \tan^3 \varphi)^{4/3}} d\varphi. \tag{5}$$

In the local coordinate system  $\mathbf{UO}'\mathbf{W}$ , the coordinates of any point on the Lamé curve can be expressed as Equation (6), based on the turned angle  $\varphi$ :

$$\begin{cases} u(\varphi) = \frac{2e + (2e - B_a)(1 + \tan^3 \varphi)^{1/3}}{2(1 + \tan^3 \varphi)^{1/3}} \\ w(\varphi) = \frac{f \tan \varphi + (H_a - f)(1 + \tan^3 \varphi)^{1/3}}{(1 + \tan^3 \varphi)^{1/3}} \end{cases} \tag{6}$$

### 3.2. Space-Path Coordinate Transform

For the convenience of our study, it is necessary to transform the spatial gate-shaped path in the OXYZ coordinate system into the one seen in the  $\mathbf{UO}'\mathbf{W}$  plane local coordinate system. Assuming the pick-up endpoints have the same spatial Z-axis coordinate value, the transformation from the OXYZ coordinate system to the  $\mathbf{UO}'\mathbf{W}$  plane local coordinate system only needs to take the translation of the origin and the rotation transformation along the X axis into consideration. The coordinates  $(x, y, z)$  of any spatial point in the OXYZ coordinate system will be marked as  $(u, w)$  in the  $\mathbf{UO}'\mathbf{W}$  plane local coordinate system after the transition, which can be described as Equation (7).

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{Q} \begin{bmatrix} u \\ 0 \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} & \lambda_{14} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} & \lambda_{24} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} & \lambda_{34} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & \lambda_{44} \end{bmatrix} \begin{bmatrix} u \\ 0 \\ w \\ 1 \end{bmatrix} \tag{7}$$

where  $\mathbf{Q}$  is the transformation matrix from the  $\mathbf{UO}'\mathbf{W}$  coordinates to the OXYZ coordinates, and  $\mathbf{Q}^{-1}$  is the transformation matrix from the OXYZ coordinates to the  $\mathbf{UO}'\mathbf{W}$  coordinates.

### 3.3. Motion Law Planning

In this section, the motion law planning of the Par4 robot in the operating space is studied, which focuses on the displacement, velocity, and acceleration of the robot's end-effector. In order to ensure the smooth running of the robot end and to reduce residual vibration, a polynomial of fifth order or more is employed for the displacement planning. The higher the order is, the more complex the calculation will be. Thus, for simplicity, fifth- and sixth-order polynomials are selected here for displacement planning.

#### 3.3.1. Polynomial Displacement Planning

- Quintic Polynomial Displacement Planning

For the stable operation of the robot end, the maximum acceleration is set at T/4 time, where T is the pick-and-place period. The expressions of the quintic polynomial displacement and maximum acceleration are obtained as Equations (8) and (9), by analyzing the initial and final displacement boundary conditions, velocity boundary conditions and acceleration boundary conditions:

$$s(t) = a_{\max} T^2 (6\tau^5 - 15\tau^4 + 10\tau^3) / 5.625 \tag{8}$$

$$a_{\max} = 5.625S_a/T^2 \tag{9}$$

where  $\tau = t/T$ ,  $t$  is the current time;  $S_a$  is the total pick-up displacement, and here  $S_a = 2(H_a - f - e + l_{BC}) + B_a$ .

- Sextic Polynomial Displacement Planning

To reduce residual vibration at the falling point of the robot end, the proportion of the acceleration time and deceleration time is set as 4:6 in the sextic polynomial motion law, meaning that the maximum velocity is at  $2T/5$  time when the acceleration is zero,  $a(2T/5) = 0$ . Then, the expression of displacement can be described as in Equation (10).

$$s(t) = S_a(-10\tau^6 + 36\tau^5 - 45\tau^4 + 20\tau^3) \tag{10}$$

It should be remembered that both the quintic and the sextic polynomial displacement planning expressions are asymmetric.

### 3.3.2. Piecewise Motion Planning Based on the Lamé Curve

Based on Figures 2 and 3, there are five segments in the gate-shaped pick-and-place movement: The AB segment rising vertically, the BC segment rising along the Lamé curve, the CF segment moving horizontally, the FG segment falling along the Lamé curve and the GH segment falling vertically. Due to the symmetry of the motion trajectory, the following section will mainly study the motion planning of the AB segment, BC segment, and CF segment in the  $UO'W$  plane local coordinate system, and finally convert the planning to the space coordinate OXYZ.

#### 1. AB Segment Motion Planning

The displacement range is  $0 \leq s(t) < (H_a - f)$ , with vertical motion only. The spatial coordinates of point A are set as  $(x_A, y_A, z_A)$ , and the current time is denoted as  $t_i$ , so, the displacement, velocity and acceleration laws in the space coordinate OXYZ can be shown in Equations (11)–(13):

$$\begin{bmatrix} x_i & y_i & z_i \end{bmatrix} = \begin{bmatrix} x_A & y_A & z_A + s(t_i) \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} v_{xi} & v_{yi} & v_{zi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & s'(t_i) \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} a_{xi} & a_{yi} & a_{zi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & s''(t_i) \end{bmatrix}. \tag{13}$$

#### 2. BC Segment Motion Planning

- Displacement Planning

When the displacement is taken as  $(H_a - f) \leq s(t) < (H_a - f) + l_{BC}$ , it moves along the Lamé curve. Taking the previous running point to be  $k_{i-1}$ , the current running point is  $k_i$ , as indicated in Figure 3. The key is to calculate the interpolation values  $\Delta u_i$  and  $\Delta w_i$ . In Figure 3, there is the increment of arc length  $\Delta l_i = s(t_i) - s(t_{i-1})$ , in which  $t_i$  represents the present time, and  $t_{i-1}$  the previous time. According to Equation (4), Equation (14) is deduced:

$$u'_i(w) = -(w_i^2 e^3)/(u_i^2 f^3) \tag{14}$$

When the interpolation time is  $\Delta t_i \rightarrow 0$ , then  $\Delta l_i \rightarrow 0$ . The approximate arc differential calculation formula can be obtained as in Equation (15):

$$\Delta w_i \approx \Delta l_i / \sqrt{1 + [u'_i(w)]^2}. \tag{15}$$

Thus,

$$w_i = w_{i-1} + \Delta w_i. \tag{16}$$

The following can be obtained from Equations (4) and (16):

$$u_i = e[1 - (w_i/f)^3]^{1/3} \tag{17}$$

$$\Delta u_i = u_i - u_{i-1}. \tag{18}$$

According to the analysis above, to avoid  $u'_i(w) \rightarrow \infty$  when  $k_i$  is approaching the W axis,  $w'_i(u) = -(u_i^2 f^3)/(w_i^2 e^3)$  can be calculated first with Equation (4), and in the same way,  $\Delta u_i$  can be calculated first, before calculating  $u_i$ ,  $w_i$ , and  $\Delta w_i$  in turn. There is a strong duality between the two calculated methods. In order to reduce calculation errors, when  $(H_a - f) \leq s(t) < (H_a - f) + l_{BC}/2$ , the former method is adopted; when  $(H_a - f) + l_{BC}/2 \leq s(t) < (H_a - f) + l_{BC}$ , the latter method is adopted.

The coordinate transformation can be attained as

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \lambda_{13} \\ \lambda_{21} & \lambda_{23} \\ \lambda_{31} & \lambda_{33} \end{bmatrix} \begin{bmatrix} u_i \\ w_i \end{bmatrix} + \begin{bmatrix} \lambda_{14} \\ \lambda_{24} \\ \lambda_{34} \end{bmatrix} \tag{19}$$

If  $X_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$ ,  $Q_1 = \begin{bmatrix} \lambda_{11} & \lambda_{13} \\ \lambda_{21} & \lambda_{23} \\ \lambda_{31} & \lambda_{33} \end{bmatrix}$ ,  $Q_2 = [\lambda_{14} \ \lambda_{24} \ \lambda_{34}]^T$ ,  $U_i = [u_i \ w_i]^T$ , Equation (19) can be denoted as  $X_i = Q_1 U_i + Q_2$ .

- Speed Planning

When  $\Delta t_i \rightarrow 0$ ,  $\Delta l_i \rightarrow 0$ ,  $\Delta l_i \approx \sqrt{\Delta u_i^2 + \Delta w_i^2}$ , then

$$v_{ui} \approx s'(t_i) \frac{\Delta u_i}{\sqrt{\Delta u_i^2 + \Delta w_i^2}} = \frac{s'(t_i) \Delta u_i}{\Delta l_i} \tag{20}$$

$$v_{wi} \approx s'(t_i) \frac{\Delta w_i}{\sqrt{\Delta u_i^2 + \Delta w_i^2}} = \frac{s'(t_i) \Delta w_i}{\Delta l_i}. \tag{21}$$

The velocities converted to the robot's end space coordinate OXYZ are

$$\begin{bmatrix} v_{xi} & v_{yi} & v_{zi} \end{bmatrix}^T = Q_1 \begin{bmatrix} v_{ui} & v_{wi} \end{bmatrix}^T. \tag{22}$$

- Acceleration Planning

The acceleration along the U axis can be calculated as follows:

$$u''(t) = \frac{d}{dt} u'(t) = \frac{d}{dt} (u'(\varphi) \varphi'(t)) = u''(\varphi) [\varphi'(t)]^2 + u'(\varphi) \varphi''(t). \tag{23}$$

Similarly, the acceleration calculation formula along the W axis can be obtained as

$$w''(t) = w''(\varphi) [\varphi'(t)]^2 + w'(\varphi) \varphi''(t). \tag{24}$$

Here, all of  $u'(\varphi)$ ,  $u''(\varphi)$ ,  $w'(\varphi)$  and  $w''(\varphi)$  can be calculated by Equation (6), and the calculation of  $\varphi'(t)$  and  $\varphi''(t)$  will be derived as follows:

Since  $s'(t) = s'(\varphi)\varphi'(t)$ , and the motion of the Lamé curve shown in Figure 3 is in the rising state, i.e.,  $s'(\varphi) > 0$ , then  $\varphi'(t)$  can be calculated as Equation (25):

$$\varphi'(t) = \frac{s'(t)}{s'(\varphi)} = \frac{s'(t)}{\sqrt{[u'(\varphi)]^2 + [w'(\varphi)]^2}}. \tag{25}$$

Thus,  $\varphi''(t)$  can be calculated by performing the derivation of Equation (25) with respect to the variable  $t$ .

The acceleration converted to the robot end space coordinate OXYZ is

$$\begin{bmatrix} a_{xi} & a_{yi} & a_{zi} \end{bmatrix}^T = Q_1 \begin{bmatrix} a_{ui} & a_{wi} \end{bmatrix}^T = Q_1 \begin{bmatrix} u''(t_i) & w''(t_i) \end{bmatrix}^T. \tag{26}$$

### 3. CF Segment Motion Planning

When the displacement is taken as  $(H_a - f + l_{BC}) \leq s(t) < (H_a - f + l_{BC} + B_a - 2e)$ , the motion of the CF segment is a plane horizontal movement, and in the  $UO'W$  coordinate system, there are the following equations:  $\Delta u_i = \Delta l_i = s(t_i) - s(t_{i-1})$ ,  $\Delta w_i = 0$ ,  $v_{ui} = s'(t_i)$ ,  $v_{wi} = 0$ ,  $a_{ui} = s''(t_i)$ , and  $a_{wi} = 0$ .

Taking the coordinate of point C in the  $UO'W$  coordinate system as  $(u_C, w_C)$ , then,  $u_i = u_{i-1} + \Delta u_i = u_{i-1} + \Delta l_i$ , and  $w_i = w_{i-1} = w_C$ .

The planning converted to the robot end space OXYZ is

$$\begin{cases} \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T = Q_1 \begin{bmatrix} u_i & w_C \end{bmatrix}^T + Q_2 \\ \begin{bmatrix} v_{xi} & v_{yi} & v_{zi} \end{bmatrix}^T = Q_1 \begin{bmatrix} s'(t_i) & 0 \end{bmatrix}^T \\ \begin{bmatrix} a_{xi} & a_{yi} & a_{zi} \end{bmatrix}^T = Q_1 \begin{bmatrix} s''(t_i) & 0 \end{bmatrix}^T \end{cases}. \tag{27}$$

### 4. Other Segment Motion Planning

Due to the left-right symmetry of the gate-shaped trajectory, the FG segment's motion is identical to the BC segment's, and the GH segment's motion to the AB segment's, so they will be not elaborated here.

According to the motion planning described above, the interpolation displacement, velocity, and acceleration of the gate-shaped trajectory at each moment can be obtained based on the Lamé curve, and then the rotation angle of each driving unit can be obtained through the inverse kinematic equation analysis, so that the desired trajectory planning can be achieved.

## 4. Trajectory Optimization Based on GWO

Although the trajectory shape and motion law of the Par4 parallel robot have been planned in the previous sections, the problem of parameter selection for the Lamé curve remains unresolved. Previously, the parameters of the Lamé curve have mainly been determined in the planning process through prior knowledge, and certainly there are some optimization designs done already, such as the method for a Delta parallel robot in [37], which is a brute force optimization method where the parameters of the Lamé curve were traversed at a step length of 1 mm, thus causing not only a heavy workload but also a large degree of optimization error. Therefore, in this paper, we adopt the recently proposed Grey Wolf Optimizer (GWO) to achieve the trajectory optimization with the target of minimal energy consumption for the Par4 parallel robot. In this section, the GWO algorithm is first introduced, and then the trajectory optimization method is elaborated.

### 4.1. Grey Wolf Optimizer

The GWO algorithm is a new intelligent algorithm, proposed in 2014 [38]. Because of its features of simple operational structure, high search efficiency, and ease of understanding and programming, it has been widely employed in all walks of life since it was proposed [39–41].

The GWO algorithm aims to simulate the group hunting behavior of grey wolves [38]. In GWO, the prey represents the optimal solution, and the process of optimization can be understood as catching the prey. Generally, there are three grey wolves in leadership roles, named Wolf  $\alpha$ , Wolf  $\beta$  and Wolf  $\delta$ , and there are many searching wolves, all called Wolf  $\omega$ , whose number is often in dozens. During the process of catching the prey, Wolf  $\alpha$  is the wolf closest to the prey, Wolf  $\beta$  the further away, and Wolf  $\delta$  further away again. The positions of the three leader grey wolves are always taken as a reference for Wolf  $\omega$  to carry out the hunting search behavior. Until all Wolves  $\omega$  complete the hunting search behavior with their own positions updated, the position of Wolf  $\alpha$ , Wolf  $\beta$  or Wolf  $\delta$  can be updated as long as the solution of Wolf  $\omega$  is better than theirs. In each iteration, Wolf  $\alpha$  always represents the optimal solution, meaning it is closest to the prey. In the hunting search progress, the updated position of Wolf  $\omega$  is very important, which can be calculated as follows:

$$x_1^{j+1} = x_\alpha - a_1 |c_1 x_\alpha - x_r^j| \tag{28}$$

$$x_2^{j+1} = x_\beta - a_2 |c_2 x_\beta - x_r^j| \tag{29}$$

$$x_3^{j+1} = x_\delta - a_3 |c_3 x_\delta - x_r^j| \tag{30}$$

$$x_r^{j+1} = \frac{x_1^{j+1} + x_2^{j+1} + x_3^{j+1}}{3} \tag{31}$$

Here,  $x_r^j$  is the position vector of the  $r$ th wolf in the current iteration,  $x_r^{j+1}$  the position vector of the  $r$ th wolf in the next iteration,  $x_\alpha$ ,  $x_\beta$  and  $x_\delta$  the current position vectors of Wolf  $\alpha$ , Wolf  $\beta$  and Wolf  $\delta$ , respectively, and the calculation formulas of the parameters  $a$  and  $c$  are as follows:

$$a = 2 \cdot rand1 \cdot \chi - \chi \tag{32}$$

$$c = 2 \cdot rand2 \tag{33}$$

In Equations (32) and (33), all elements in vectors  $rand1$  and  $rand2$  are random variables between 0 and 1, the value of the element in vector  $\chi$  gradually decreases from 2 to 0 with the increasing of iterations, and the  $i$ th-dimensional element of vector  $\chi$  can be calculated as follows:

$$\chi_i = 2 - 2 \cdot i / Max\_iter \tag{34}$$

where  $i$  is the number of the current iteration and  $Max\_iter$  the total number of iterations.

The result obtained by substituting the position vector of the wolf into the objective function is called the fitness. Then, the steps of GWO can be described as follows:

**Step 1:** Randomly initialize the position vectors of all wolves, including Wolf  $\alpha$ , Wolf  $\beta$ , Wolf  $\delta$  and all Wolves  $\omega$  within the allowable range, and obtain the corresponding fitness values.

**Step 2:** Compare the fitness value of Wolf  $\omega$  with the fitness values of Wolf  $\alpha$ , Wolf  $\beta$  and Wolf  $\delta$ , respectively, and if the fitness value of Wolf  $\omega$  is superior, then use the position vector of Wolf  $\omega$  to replace the position vector of Wolf  $\alpha$ , Wolf  $\beta$  or Wolf  $\delta$ , accordingly.

**Step 3:** Update the position vector of Wolf  $\omega$  in accordance with Equations (28)–(31).

**Step 4:** Determine whether the iteration end condition is met. If the condition is satisfied, output the position vector and the fitness value of Wolf  $\alpha$ , otherwise return to Step 2, and continue to perform the searching behavior.

#### 4.2. Trajectory Optimization

From the geometric meaning of parameters  $e$  and  $f$  as shown in Figure 3, it can be established that the shape of the Lamé curve will be vary with different parameters of  $e$  and  $f$ , which will result in different pick-and-place trajectories for the Par4 parallel robot. Then, the question arises of which set

of parameters  $e$  and  $f$  is more suitable for the Par4 parallel robot. Here, the GWO algorithm will be introduced to find the optimum values of parameters  $e$  and  $f$ , and the optimization will be measured by the energy consumption of the Par4 parallel robot.

#### 4.2.1. Mechanical Energy Consumption of the Par4 Parallel Robot

For the Par4 parallel robot, the total power of its motors is the sum of the circuit thermal power, the magnetic field energy storage power and the mechanical power, among which the circuit thermal power and magnetic field energy power are relatively small, and consequently, the total energy consumption of the robot can be considered equal to the mechanical energy consumption. The instantaneous mechanical power of the Par4 robot is calculated as follows:

$$p(t) = p_1(t) + p_2(t) + p_3(t) + p_4(t) \quad (35)$$

where  $p_i(t)$  ( $i = 1, 2, 3, 4$ ) is the instantaneous mechanical power of each drive unit, and  $t$  is the time, in seconds.

Assuming that only energy consumption braking is applied, the instantaneous mechanical power of each motor in the Par4 parallel robot is:

$$p_i(t) = \begin{cases} \tau_i(t)\dot{q}_i(t), & \tau_i(t)\dot{q}_i(t) > 0 \\ 0, & \tau_i(t)\dot{q}_i(t) \leq 0 \end{cases} \quad (36)$$

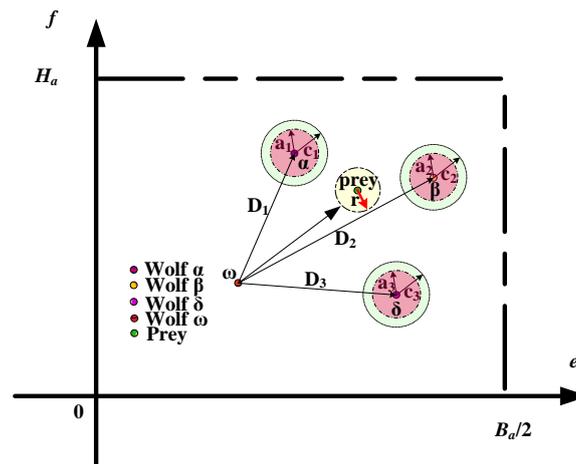
Then, the total mechanical energy consumption of the Par4 parallel robot can be calculated as follows:

$$E_m = \int_0^T p(t)dt = \int_0^T \sum_{i=1}^4 p_i(t)dt \quad (37)$$

where  $E_m$  is the total mechanical energy consumption, and  $T$  is the running period in seconds.

#### 4.2.2. Trajectory Optimization Based on the GWO Algorithm

Next, we arrive at the main question of this research work—the minimum mechanical energy consumption and the optimization of the parameters  $e$  and  $f$ . For the trajectory optimization of the Par4 parallel robot, the mechanical energy consumption is taken as the fitness value and the parameters  $e$  and  $f$  are selected as the elements of the wolf's position vector. The specific design scheme of the trajectory planning optimization is described as follows: First, the value ranges of parameters  $e$  and  $f$  are determined by the pick-up span and the pick-up height respectively, i.e., the value range of  $e$  is  $[0, B_a/2]$ , and the value range of  $f$  is  $[0, H_a]$ . The pick-up span can be computed from the pick-and-place endpoint coordinates, and the pick-up height can be set at an appropriate value by the user. Then, according to the current distances from Wolf  $\omega$  to Wolf  $\alpha$ , Wolf  $\beta$  and Wolf  $\delta$ , namely  $D_1$ ,  $D_2$ , and  $D_3$ , respectively, as shown in Figure 3, we can calculate the likely location of the prey by Equations (28)–(31), which falls within the range of Radius  $r$  in Figure 4, representing the possible optimal value range of parameters  $e$  and  $f$  in the trajectory planning of the Par4 parallel robot. Then, we substitute the obtained optimal  $e$  and  $f$  values into the gate-shaped path formula which has been planned above, and the corresponding minimum mechanical energy consumption is obtained by following the fifth-order or sixth-order polynomial motion law. Finally, we determine whether the mechanical energy consumption after several iterations is at the minimum. If it is at the minimum, the trajectory optimization is ended; otherwise, we continue to find the optimal parameters  $e$  and  $f$  by using the GWO algorithm. In Figure 4,  $c_1$ ,  $c_2$ ,  $c_3$ ,  $a_1$ ,  $a_2$ , and  $a_3$  correspond to the parameters in Equations (28)–(30). Among them,  $c_1$ ,  $c_2$ , and  $c_3$  represent the weights of Wolf  $\alpha$ , Wolf  $\beta$ , and Wolf  $\delta$ , respectively, in the searching calculation, and the numerical calculations all adopt the formula of Equation (33);  $a_1$ ,  $a_2$ , and  $a_3$  represent the distance factors from Wolf  $\alpha$ , Wolf  $\beta$ , and Wolf  $\delta$ , respectively, to the current Wolf  $\omega$ , whose calculations follow Equation (32).



**Figure 4.** The work principle sketch map of the GWO algorithm applied in the trajectory planning.

The optimization of the planned trajectory with the GWO algorithm has the following advantages:

1. The GWO algorithm takes a simple structure and legible logic, making the programming easily accessible and the program statement more concise. Therefore, the GWO to some extent achieves a complementary effect for the complex programmed control system of the Par4 parallel robot itself, acting as a time balancer in the total running time.

2. The parameters of the GWO algorithm need little setting, making the operation of the Par4 parallel robot more convenient and easier when changing the parameters, such as the pick-and-place endpoints and the pick-up height.

3. In the GWO algorithm, the searching method around the location of the three leader wolves is adopted, which helps towards a fast convergence.

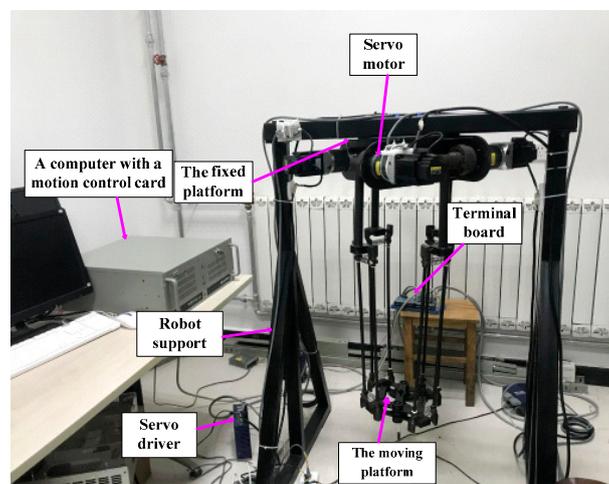
Although the GWO algorithm has the disadvantage of easily falling into the local optimum when all the three leader wolves are in the same local optimal region, it has little impact on the practical application with limited parameters in a fixed range under certain conditions. Therefore, the possibility of the algorithm falling into the local optimum is greatly reduced when applied in the above trajectory optimization.

## 5. Experimental Verification of Trajectory Optimization

In this section, the planning and optimization methods designed above are verified by some experiments. To compare and analyze the influence of the fifth- and sixth-order polynomial movement rules on mechanical energy consumption, the following scheme was adopted: the basic experiment with a group of data was done first, then the comparison experiments with multiple data were performed. In the comparison experiments, three parameters, i.e., pick-up span, pick-up height and pick-and-place endpoints coordinates, were taken into consideration. The overall comparative experiments were designed by fixing the values of two parameters and changing the value of the third parameter. In all experiments, the number of wolves was set at 50, and the number of iterations was set at 40. The parameters of the Par4 robot are shown in Table 1, and the experimental platform is shown in Figure 5.

**Table 1.** Par4 parallel robot-related parameters.

Parameter Name	Value
Length of big arm $L_i$	273 mm
Length of small arm $l_i$	600 mm
Radius of big arm revolute $R$	190 mm
Length of moving platform along x axis $d$	160 mm
Length of moving platform along y axis $h$	160 mm
Distance between $B_i$ and $C_i$ along y axis $h_i$	20 mm
Distance between $B_i$ and $C_i$ along x axis $d_i$	20 mm
Load mass of moving platform $m_c$	0 kg
Mass of big arm $m_b$	0.08 kg
Mass of small arm $m_s$	0.07 kg
Mass of moving platform $m_d$	2.6 kg
Mass of big arm end connections $m_{end}$	0.59 kg

**Figure 5.** The Par4 parallel robot platform.

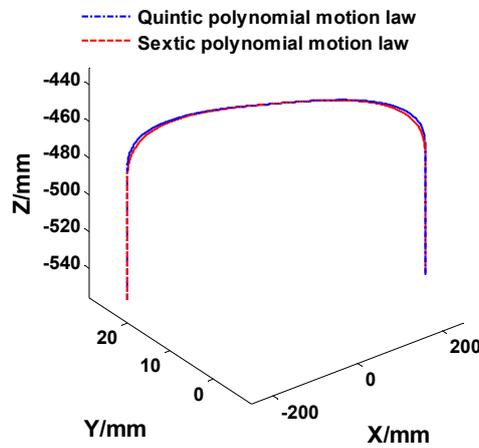
### 5.1. Basic Experiment

With a group of pick-and-place endpoints taken and the pick-up height given, the trajectory planning above was adopted, i.e., the sleek gate-shaped path based on the Lamé curve was selected, and the fifth- and sixth-order polynomial motion laws were respectively applied to carry out the optimization experiment based on the GWO algorithm for the lowest mechanical energy consumption.

The coordinates of the starting pick-and-place endpoint were set at  $k_0 = [-250, 20, -543.5]$ , the ending coordinates at  $k_f = [250, 0, -543.5]$ , and the pick-up height at  $H_a = 100$  mm, so then the pick-up span could be obtained as  $B_a = 500.4$  mm. The optimal values of parameter  $e$ , parameter  $f$ , and the minimum mechanical energy consumption (noted as  $E_{\min}$  for convenience) based on the GWO algorithm are shown in Table 2. The unit of parameters  $e$  and  $f$  is millimeters (mm), the unit of  $E_{\min}$  is joules (J). The units of these parameters in subsequent tables are the same. The numerical results in Table 2 reflect the fact that the mechanical energy consumption is relatively smaller under the law of quintic polynomial motion. It is worth mentioning that the optimization result of parameter  $e$  is the same (250.2 mm,  $B_a/2$ ) regardless of the motion law of the quintic polynomial or the sextic polynomial. However, the optimization result of Parameter  $f$  is not the same. The optimal values in Table 2 were taken, and the trajectory planning method above was implemented. Then, the trajectories were obtained under the fifth-order and sixth-order polynomial motion laws, as shown in Figure 6.

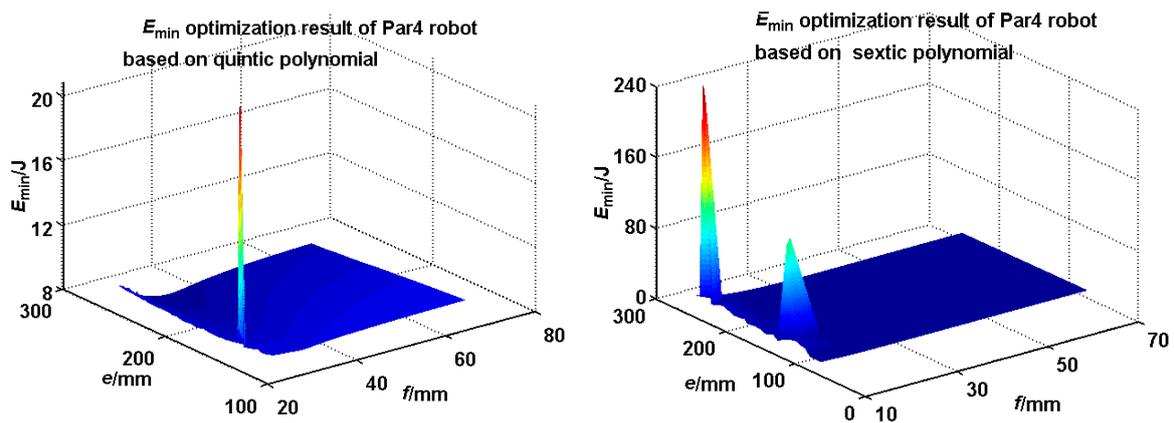
**Table 2.** Results of parameter optimization based on GWO.

Motion Law	$e/mm$	$f/mm$	$E_{min}/J$
Quintic polynomial	250.2	32.3	8.209
Sextic polynomial	250.2	38.2	8.500



**Figure 6.** Motion trajectory.

The iteration trajectory of mechanical energy consumption  $E_{min}$  is shown in Figure 7. From the results in Figure 7, the overall  $E_{min}$  value of the quintic polynomial motion law (left figure) is smaller than that of the sextic polynomial motion law (right figure). When the  $E_{min}$  reading is at a higher value, the left graph displays a thin thread-like peak in Figure 7, which shows that only a few wolves in the GWO algorithm failed to find the best prey in time. In other words, in the iterative process, GWO has higher efficiency and faster convergence under the law of quintic polynomial motion. The rotation angles, angular velocities, and corresponding drive torques of each joint of the Par4 parallel robot were obtained based on the optimum parameters  $e$  and  $f$ , which are shown in Figure 8.



**Figure 7.** Iterative results of the minimum mechanical energy consumption based on GWO.

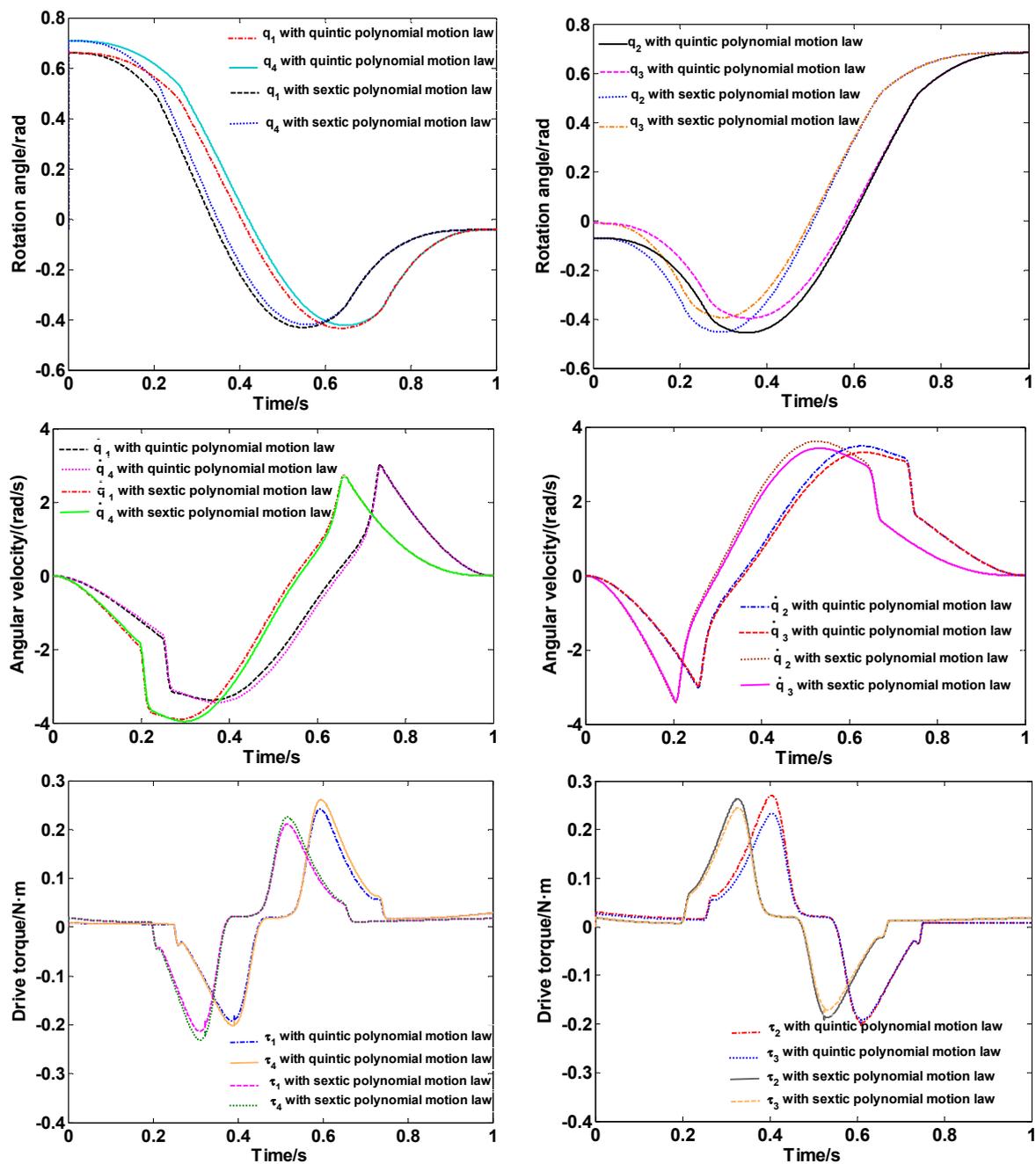


Figure 8. Curves of rotation angle, angular velocity, and drive torque.

5.2. Experiments with the Same Pick-and-Place Endpoints and Different Pick-Up Heights

Table 3 shows the optimization results with the minimum mechanical energy consumption when the pick-and-place endpoint coordinates were set at  $k_0 = [-250, 20, -543.5]$  and  $k_f = [250, 0, -543.5]$ , thus the pick-up span was fixed at  $B_a = 500.4$  mm. The pick-up heights were set at 100, 110, 120, 130, 140 and 150 mm, respectively. According to the results in Table 3, the mechanical energy consumption of the sextic polynomial was slightly higher than that of the quintic polynomial. In addition, at different pick-up heights, the optimal value of parameter  $e$  was 250.2 mm (this value is  $B_a/2$ ) regardless of the movement law of the quintic or sextic polynomial, and the optimal value of parameter  $f$  increases with the increase in pick-up height.

**Table 3.** Optimized results with different  $H_a$ , the same  $B_a$  and the same endpoints.

Motion Law/ $H_a$	Quintic Polynomial			Sextic Polynomial		
	$e/mm$	$f/mm$	$E_{min}/J$	$e/mm$	$f/mm$	$E_{min}/J$
100 mm	250.2	32.3	8.209	250.2	38.2	8.500
110 mm	250.2	33.7	8.757	250.2	40.2	9.054
120 mm	250.2	35.4	9.323	250.2	42.3	9.628
130 mm	250.2	37.1	9.912	250.2	44.6	10.224
140 mm	250.2	38.7	10.528	250.2	47.1	10.847
150 mm	250.2	40.5	11.178	250.2	50.0	11.503

5.3. Experiments with Different Pick-and-Place Endpoints and the Same Pick-Up Height and Span

For the sake of comparison and analysis, here, five groups of different pick-and-place endpoints were taken for planning and comparison in the Par4 robot’s spatial feasible region. The X and Y axis coordinates of the former two groups of pick-and-place endpoints were both small and far away from the boundary of the robot’s feasible region, while the X or Y axis coordinates of the last three groups of pick-and-place endpoints were relatively larger and closer to the boundary of the robot’s feasible region. In the experiments, the pick-up height was the same, set at 120 mm; the pick-up spans were all set at 500.4 mm. Table 4 shows the optimization results based on GWO, and the minimum mechanical energy consumptions are given accordingly. Since all the Z-axis coordinates (−543.5) were the same, for simplicity, only X-axis and Y-axis coordinates are listed in Table 4 for the pick-and-place endpoints.

**Table 4.** Optimized results with the same  $H_a$ , the same  $B_a$  and different endpoints.

Motion Law/Pick-and-Place Points		Quintic Polynomial			Sextic Polynomial		
$k_0$	$k_f$	$e/mm$	$f/mm$	$E_{min}/J$	$e/mm$	$f/mm$	$E_{min}/J$
[−200,237.6]	[150,−120]	250.2	39.2	9.445	250.2	50.3	9.566
[−250,20]	[250,0]	250.2	35.4	9.323	250.2	42.3	9.628
[20,20]	[360,387.2]	250.2	120.0	11.266	250.2	120	12.515
[−10,−20]	[−260,−453.5]	250.2	120.0	15.142	250.0	120	15.167
[0,200]	[−400,−100.7]	250.2	120.0	10.954	250.2	120	10.963

From Table 4, the mechanical energy consumptions under the former two sets of coordinates were relatively lower than those under the last three sets of coordinates, and in particular, the  $E_{min}$  values for the fourth set of coordinates were the highest. Moreover, in the experiments, the mechanical energy consumption based on the quintic polynomial was less than that based on the sextic polynomial under the same conditions. Besides, it is worth mentioning that almost all of the optimal values of parameter  $e$  were 250.2 mm (this value is  $B_a/2$ ), while the optimal values of parameter  $f$  were relatively smaller in the former two sets of coordinates than those in the last three sets of coordinates, which were close to the pick-up height  $H_a$  (120 mm). Hence, the optimal value of parameter  $f$  is related to the spatial coordinates of the pick-and-place endpoints.

5.4. Experiments with Different Pick-Up Spans, Different Pick-and-Place Endpoints and the Same Pick-Up Height

The pick-up height was taken as 120 mm, and six groups of different pick-and-place endpoints were selected, of which the corresponding pick-up spans are shown in Table 5 as  $B_a/2$ . Since all the Z axis coordinates are −543.5, only X axis and Y axis coordinates of the pick-and-place endpoints are listed in Table 5, for simplicity. In Table 5, the difference between the two motion laws of quintic and sextic polynomials is not obvious when the pick-up height and the pick-and-place coordinates are the same. If the average mechanical energy consumption (denoted as average  $E_{min}$ ) in the six different groups of coordinates is taken for comparison, the value under the quintic polynomial motion law is

lower than the one under the sextic polynomial motion law. In addition, regardless of the movement law of the quintic or sextic polynomial, almost all the optimization values of parameter  $e$  are equal to  $B_a/2$ , but the optimization feature of parameter  $f$  is not so obvious, and its value is not only related to the movement law but also related to the pick-and-place endpoints.

**Table 5.** Optimized results with the same  $H_a$ , different  $B_a$  and different endpoints.

Motion Law/Pick-and-Place Points		Quintic Polynomial			Sextic Polynomial			
$k_0$	$k_f$	Ba/2/mme/mm	f/mm	$E_{min}/J$	e/mm	f/mm	$E_{min}/J$	
[80,30]	[30,-20]	35.4	35.4	81.5	7.283	35.3	76.8	7.287
[75,55]	[-35,150]	72.7	72.7	120	7.746	72.7	120	7.744
[150,50]	[-50,20]	101.1	101.1	120	7.883	101.1	120	7.878
[300,0]	[-10,75]	159.5	159.5	120.0	7.721	159.5	120	7.711
[-210,50]	[150,235]	202.4	202.4	37.0	8.852	202.4	40.8	8.890
[0,-305]	[0,396.7]	350.9	350.9	63.3	11.048	350.9	76.5	11.124
average $E_{min}$					8.422			8.439

### 6. Conclusions

The main work of this paper is as follows:

1. The Lamé curve was employed in the trajectory planning of the Par4 parallel robot, and the piecewise design method was adopted. Detailed analysis and precise planning were elaborated, focusing on displacement, velocity, and acceleration in the Par4 parallel robot end operating space.

2. To minimize the total mechanical energy consumption of the driving joints, asymmetric displacement planning was introduced based on the quintic and sextic polynomial motion laws. A series of experiments were performed to compare and analyze the mechanical energy consumption of the Par4 parallel robot under the quintic and sextic polynomial motion laws. The results reflect that the mechanical energy consumption is lower when the motion law of the quintic polynomial is adopted.

3. To find the optimal trajectory, the recently proposed GWO algorithm was used. After a series of experiments to minimize the total mechanical energy consumption, it turned out that the optimal value of parameter  $e$  of the Lamé curve can be set as half of the pick-up span in the Par4 parallel robot’s trajectory planning, while the optimal value of parameter  $f$  depends on different pick-up heights and different pick-and-place endpoints.

In addition, the mechanical energy consumption is related not only to the pick-up height and pick-up span of the trajectory, but also to the coordinate position of the pick-and-place endpoints. The closer the pick-and-place endpoints to the boundary of the Par4 parallel robot’s feasible region, the higher the mechanical energy consumption will be.

The trajectory planning method proposed in this paper is also suitable for other parallel robots, such as Delta parallel robots. Although the trajectory is based on the Lamé curve in this paper, the design method is well suited to two-dimensional curves whose coordinate variables can both be expressed as functions of an intermediate variable that changes over time, such as the circle, the transitional parametric spline curve, and so on.

**Author Contributions:** Z.M., as the corresponding author, takes primary responsibility for this article. X.Z. drafted the manuscript. All co-authors reviewed the manuscript.

**Funding:** This research was funded by the National High-Tech Research and Development Program (863 Torch Program) (grant no. 2015AA7041003), the Educational Reform Project of Xianyang Normal University (grant no. 2017Y004), and the Scientific Research Plan Projects of Shaanxi Education Department (grant no. 15JK1784).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Song, X.G.; Zhao, Y.J.; Jin, L.; Zhang, P.; Chen, C.W. Dynamic feedforward control in decoupling space for a four-degree-of-freedom parallel robot. *Int. J. Adv. Robot. Syst.* **2019**, *2*, 1–10. [[CrossRef](#)]
2. Sohn, J.W.; Kim, G.W.; Choi, S.B. A State-of-the-Art Review on Robots and Medical Devices Using Smart Fluids and Shape Memory Alloys. *Appl. Sci.* **2018**, *8*, 1928. [[CrossRef](#)]
3. Nabat, V.; François, P.; Poignet, P. Dynamic Modeling and Identification of Par4, A Very High Speed Parallel Manipulator. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, Beijing, China, 9–15 October 2006.
4. Staicu, S.; Shao, Z.F.; Zhang, Z.K.; Tang, X.Q.; Wang, L.P. Kinematic analysis of the X4 translational–rotational parallel robot. *Int. J. Adv. Robot Syst.* **2018**, *10*, 1–12. [[CrossRef](#)]
5. Sun, T.; Liang, D.; Song, Y.M. Singular-Perturbation-Based Nonlinear Hybrid Control of Redundant Parallel Robot. *IEEE T. Ind. Electron.* **2018**, *65*, 3326–3336. [[CrossRef](#)]
6. Francisco, G.S.; Jorge, O.S.; Raymundo, J.T.; Ricardo, P.P. A stable proportional–proportional integral tracking controller with self-organizing fuzzy-tuned gains for parallel robots. *Int. J. Adv. Robot. Syst.* **2019**, *2*, 1–16.
7. Mohan, S.; Corves, B. Inverse dynamics and trajectory tracking control of a new six degrees of freedom spatial 3-RPRS parallel manipulator. *Mech. Sci.* **2017**, *8*, 235–248. [[CrossRef](#)]
8. Vincent, R.; Mohammed, T.; Gilles, L. Fast genetic algorithm path planner for fixed-wing military UAV using GPU. *IEEE Trans. Aero. Electron. Syst.* **2018**, *54*, 2105–2117.
9. Farajtabar, M.; Daniali, H.M.; Varedi, S.M. Pick and place trajectory planning of planar 3-RRR parallel manipulator in the presence of joint clearance. *Robotica* **2017**, *35*, 241–253. [[CrossRef](#)]
10. Lin, C.J.; Chen, C.T. Reconfiguration for the Maximum Dynamic Wrench Capability of a Parallel Robot. *Appl. Sci.* **2016**, *6*, 80. [[CrossRef](#)]
11. Kucuk, S. Optimal trajectory generation algorithm for serial and parallel manipulators. *Robot. Cim. Int. Manuf.* **2017**, *48*, 219–232. [[CrossRef](#)]
12. Jiang, X.L.; Barnett, E.; Gosselin, C. Dynamic point-to-point trajectory planning beyond the static workspace for six-dof cable-suspended parallel robots. *IEEE Trans. Robot.* **2018**, *34*, 781–793. [[CrossRef](#)]
13. Jiang, X.L.; Gosselin, C. Dynamic point-to-point trajectory planning of a three-dof cable-suspended parallel robot. *IEEE Trans. Robot.* **2016**, *32*, 1550–1557. [[CrossRef](#)]
14. Zhang, N.; Shang, W.W.; Cong, S. Geometry-based trajectory planning of a 3-3 cable-suspended parallel robot. *IEEE Trans. Robot.* **2017**, *33*, 484–491. [[CrossRef](#)]
15. Chen, Y.D.; Li, L. Predictable trajectory planning of industrial robots with constraints. *Appl. Sci.* **2018**, *8*, 2648. [[CrossRef](#)]
16. Jahanpour, J.; Motallebi, M.; Porghoveh, M. A novel trajectory planning scheme for parallel machining Robots enhanced with NURBS curves. *J. Intell. Robot. Syst.* **2016**, *82*, 257–275. [[CrossRef](#)]
17. Nan, W.H.; Tang, X.Q.; Song, B. A new automatic motion planning algorithm for a 4-degree-of-freedom parallel kinematic manipulator based on the centre sphere method. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2015**, *229*, 64–77. [[CrossRef](#)]
18. Chen, Z.H.; Wang, Y.; Ouyang, P.; Huang, J.; Zhang, W.J. A novel iteration-based controller for hybrid machine systems for trajectory tracking at the end-effector level. *Robotica* **2011**, *29*, 317–324. [[CrossRef](#)]
19. Mei, J.P.; Zang, J.W.; Qiao, Z.Y.; Liu, S.T.; Song, T. Trajectory planning of 3-DOF Delta parallel manipulator. *J. Mech. Eng.* **2016**, *52*, 9–17. [[CrossRef](#)]
20. Bourbonnais, F.; Bigras, P.; Bonev, I.A. Minimum-time trajectory planning and control of a pick-and-place five-bar parallel robot. *IEEE/ASME Trans. Mech.* **2015**, *20*, 740–749. [[CrossRef](#)]
21. Bonami, P.; Olivares, A.; Staffetti, E. Energy-optimal multi-goal motion planning for planar robot manipulators. *J. Optimiz. Theory App.* **2014**, *163*, 80–104. [[CrossRef](#)]
22. Petrinc, K.; Kovačić, Z. Trajectory planning algorithm based on the continuity of jerk. In Proceedings of the 15th Mediterranean Conference on Control and Automation, Athens, Greece, 27–29 June 2007.
23. Chen, C.T.; Liao, T.T. Trajectory planning of parallel kinematic manipulators for the maximum dynamic load-carrying capacity. *Meccanica* **2016**, *51*, 1653–1674. [[CrossRef](#)]
24. Fares, J.A.; Iyad, F.A.; Rasha, M.A.; Mohamed, A. Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 389–406.

25. Su, T.T.; Cheng, L.; Wang, Y.K.; Liang, X.; Zheng, J.; Zhang, H.J. Time-optimal trajectory planning for Delta robot based on quintic Pythagorean-hodograph curves. *IEEE Access*. **2018**, *6*, 28530–28539. [[CrossRef](#)]
26. Abe, A. Minimum energy trajectory planning method for robot manipulator mounted on flexible base. In Proceedings of the 9th Asian Control Conference (ASCC), Istanbul, Turkey, 23–26 June 2013.
27. Sun, Z.H.; Zhang, B.; Cheng, L.; Zhang, W.J. Application of the redundant servomotor approach to design of path generator with dynamic performance improvement. *Mech. Mach. Theory* **2011**, *46*, 1784–1795. [[CrossRef](#)]
28. Zhang, T.; Zhang, W.J.; Gupta, M.M. An underactuated self-reconfigurable robot and the reconfiguration evolution. *Mech. Mach. Theory* **2018**, *124*, 248–258. [[CrossRef](#)]
29. Shintaku, E. Minimum energy trajectory for an underwater manipulator and its simple planning method by using a genetic algorithm. *Adv. Robot.* **1999**, *13*, 115–138. [[CrossRef](#)]
30. Mo, J.; Shao, Z.F.; Guan, L.W.; Xie, F.G. Dynamic performance analysis of the X4 high-speed pick-and-place parallel robot. *Robot. Cim. Int. Manuf.* **2017**, *46*, 48–57. [[CrossRef](#)]
31. Li, Y.H.; Huang, T.; Chetwynd, D.G. An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using quintic B-splines. *Mech. Mach. Theory* **2018**, *126*, 479–490. [[CrossRef](#)]
32. Zhang, N.; Shang, W.W.; Cong, S. Dynamic trajectory planning for a spatial 3-DoF cable-suspended parallel robot. *Mech. Mach. Theory* **2018**, *122*, 177–196. [[CrossRef](#)]
33. Machmudah, A.; Parman, S.; Zainuddin, A.; Chacko, S. Polynomial joint angle arm robot motion planning in complex geometrical obstacles. *Appl. Soft. Comput.* **2013**, *13*, 1099–1109. [[CrossRef](#)]
34. Barnard, C.J.; Briot, S.; Caro, S. Trajectory generation for high speed pick and place robots. In Proceedings of the ASME 2012 11th Biennial Conference on Engineering Systems Design, Nantes, France, 2–4 July 2012.
35. Six, D.; Briot, S.; Chriette, A.; Martinet, P. The kinematics, dynamics and control of a flying parallel robot with three quadrotors. *IEEE Robot. Autom. Lett.* **2018**, *3*, 559–566. [[CrossRef](#)]
36. Gauthier, J.F.; Angeles, J.; Nokleby, S. *Optimization of a Test Trajectory for Scara Systems*; Springer: Berlin/Heidelberg, Germany, 2008.
37. Xie, Z.X.; Li, B.; Ren, P. Energy-Based optimization and experimental validation of the pick-and-place trajectories for the delta parallel robot. *Comput. Integr. Manuf.* **2018**, *24*, 3073–3081.
38. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
39. Zhang, X.Q.; Ming, Z.F. An optimized grey wolf optimizer based on a mutation operator and eliminating-reconstructing mechanism and its application. *Front. Inform. Technol. Electron. Eng.* **2017**, *18*, 1705–1719. [[CrossRef](#)]
40. Ali, M.E.; Hassan, M.H. Dynamic global maximum power point tracking of the PV systems under variant partial shading using hybrid GWO-FLC. *Sol. Energy* **2019**, *177*, 306–316.
41. Rao, A.M.; Ramji, K.; Kumar, T.N. Intelligent navigation of mobile robot using grey wolf colony optimization. *Mater. Today Proc.* **2018**, *5*, 19116–19125.

