applied sciences

MDPI

Article

A Time-Optimal Continuous Jerk Trajectory Planning Algorithm for Manipulators

Yaosheng Zhou [1], Guirong Han [1,2], Ziang Wei [3,4], Zixin Huang [3,4] and Xubing Chen [1,*]

[1] School of Mechanical and Electrical Engineering, Wuhan Institute of Technology, Wuhan 430205, China
[2] School of Industrial Design, Hubei Institute of Fine Arts, Wuhan 430205, China
[3] School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan 430205, China
[4] Hubei Key Laboratory of Digital Textile Equipment, Wuhan Textile University, Wuhan 430200, China
[*] Correspondence: bluegif@gmail.com

Abstract: In this paper, we propose a new optimal trajectory planning method for the manipulator to optimize its operating efficiency and ensure the smoothness of the motion process. The position sequences in joint space corresponding to a specified trajectory in task space are obtained by using the inverse kinematic algorithm, and the seventh-degree B-spline curve interpolation method is used to construct the joint trajectory with controllable start–stop kinematic parameters, and continuous velocity, acceleration and jerk. The kinematic constraints of the manipulator are transformed into the control point constraints of the B-spline curve, the sequential quadratic programming method is used to solve the optimal motion time node, and then the time-optimal continuous jerk trajectory satisfying the nonlinear kinematic constraints is planned. Simulation results show that the proposed trajectory planning method provides the ideal trajectory for the joint controller, ensuring the manipulator can smoothly track any specified trajectory in the task space in the shortest time.

Keywords: trajectory planning; optimal execution time; continuous jerk; sequential quadratic programming

check for updates

Citation: Zhou, Y.; Han, G.; Wei, Z.; Huang, Z.; Chen, X. A Time-Optimal Continuous Jerk Trajectory Planning Algorithm for Manipulators. *Appl. Sci.* **2023**, *13*, 11479. https://doi.org/10.3390/app132011479

Academic Editor: Luis Javier Garcia Villalba

Received: 14 September 2023
Revised: 11 October 2023
Accepted: 18 October 2023
Published: 19 October 2023

1. Introduction

With the continuous development of industrial automation, many manual operations in the field of industrial production have been replaced by industrial robots. With its accuracy, flexibility and high efficiency, industrial robots have greatly improved labor productivity, reduced the burden on workers, and can even perform many complex tasks that cannot be performed by humans [1,2]. Motion control technology guarantees the stable operation of robots, and the motion control algorithm is the core of robot control. In order to improve the working efficiency of robots, avoid the harm caused by joint vibration and prolong the service life of robots, trajectory planning for the robots has received close attention.

The trajectory planning scheme determines the motion mode and operation performance of robots. Different planning schemes are required for different applications, which greatly promotes the development of trajectory planning research of modern industrial robots [3]. In modern industrial automation applications, such as welding and spraying, the robot end effector must move according to certain operating requirements, and there are strict requirements for its displacement, speed and even acceleration, so specific trajectory planning must be carried out. On the other hand, the quality of the trajectory planning scheme directly determines the movement mode, operation accuracy and service life of the robot. A good trajectory planning scheme can not only enable the robot to complete the task accurately, but also ensure good motion stability and small mechanical wear, as well as have a good performance in time utilization and energy consumption [4–6]. Therefore, trajectory optimization has become one of the research hotspots in the field of robotics [7,8].

Traditional time-optimal motion algorithms with jump acceleration mostly focus on maximizing efficiency from the perspective of time, which will cause flutter, which may damage the actuator and transmission components, resulting in large tracking errors, making it difficult to realize these laws on the machine in reality [9]. Therefore, from the point of view of practical implementation, non-smooth trajectories should be avoided as much as possible. The non-smooth trajectory may cause sudden changes in joint torque, which the actuator cannot provide. Improved motion control can be achieved through accurate dynamic modeling of the robot [10], or by forcing the programming module to generate trajectories with sufficiently high continuity, preferably at least to a level of acceleration where the jerk is finite [11]. In addition, the planned trajectory should also meet specific constraints on position, speed and acceleration according to the mechanical characteristics of the drive device to ensure the dynamic performance of the system [12]. Gallant [13] used sequential quadratic programming to improve the payload capacity of a robotic arm by optimizing the cubic spline and Bernstein polynomial trajectories. A polynomial model combined with minimum disturbance theory was used to create a smooth human-like motion for the auxiliary robot [14]. However, motion constraints were not considered. Ref. [15] presented a constrained control scheme to steer a UAV to the desired position while ensuring constraint satisfaction at all times. The authors of [16] utilized the input shaping algorithm instead of the jerk constraint in the trajectory optimization model to achieve a smooth trajectory. In [17], an optimal trajectory planning method based on a Pythagorean curve for delta robot pick-and-place operation time was proposed. The authors of [18] expressed the equation of motion with a fifth-degree polynomial function and obtained the minimum kinematically constrained jump trajectory. No matter what kind of planning method is used, the kinematics and dynamics constraints of the robot must be considered to make the first two derivatives of the robot's motion continuous, and a continuous, smooth trajectory without large impact and reversal can be given within the range required by the performance of each robot component [19–21].

More and more practical applications show that trajectory planning can not only focus on the motion effect and task requirements, but also consider the efficiency, energy consumption, stationarity and other factors in the practical application process. In [22], an ethical trajectory planning algorithm is proposed that makes a variety of ethically significant decisions aimed at equitably distributing risk among road users. It is of more practical significance for engineering to find the optimal trajectory planning scheme under each working condition to guide the robot's operation [11,23,24]. To avoid the vibration of the manipulator, prolong the service life of the joints and improve the trajectory tracking accuracy, minimal jerk and jerk continuous trajectory planning have received close attention [25]. Time-optimal trajectory planning aims to plan the shortest execution time trajectory according to the given points while the boundary constraints are satisfied, that is, constructing a trajectory $p_m = f_m(t)$ satisfying kinematic constraints, while making $p_{mi} = f_m(t_i)$ hold and T minimized, where $m = 1, 2, ...N, i = 0, 1, ..., n$; N is the number of joints of the manipulator; and n is the number of joint positions that the joint trajectory must pass through. When $f_m(t)$ satisfies at least C^3 continuity, the jerk trajectory is continuous.

In view of the adverse effects of excessive impact on robots, a large number of scholars have discussed such problems as vibration, overshoot, mechanical wear and reduced life caused by the jerk [26–28]. In order to eliminate the adverse jitter effect and obtain a higher level of continuity and regularity, some works try to construct trajectories by increasing the number of polynomials. In robot picking operations and high-precision positioning systems, seventh-order polynomial interpolation is usually used to satisfy the pumping boundary conditions [29]. Ref. [30] combined cubic splines and seventh-order polynomials to ensure zero jitter at the start and end points of joint motion. The particle swarm optimization (PSO) algorithm is used to generate the time-optimal smooth trajectory. Ref. [31] described a smooth fifth-order B-spline trajectory planning method, whose objective functions were defined as movement time and integral squared acceleration. The human arm modeled by the robot arm also has a minimum jerk setting in terms of movement, which is conducive to humans smoothly grasping and lifting objects [32]. The optimal trajectory planning for the impact of industrial robots aims to find the trajectory that minimizes the impact of the robot. On the one hand, the trajectory tracking error can be reduced to a large extent by reducing the impact of the robot during movement. On the other hand, the defects such as resonance, jitter, mechanical wear and service life reduction caused by excessive impact can be greatly reduced so that the robot can run stably and smoothly.

At present, for time-optimal trajectory planning, various algorithms have their own unique problems, and so far, there is no general optimization algorithm to determine the time-optimal trajectory. Thus, in this paper, we propose a new optimal trajectory planning method for the manipulator to optimize its operating efficiency and ensure the smoothness of the motion process. First, the 7th-degree B-spline with C6 continuity is used to interpolate the joint position sequence, and the continuous jerk trajectory is obtained, whose joint start–stop velocity, acceleration and pulsation can be set separately. In this manner, the velocity, acceleration and jerk constraints of each joint are transformed into the control vertex constraints of the B-spline trajectory. Then, the shortest running time of traversing the joint position sequence is set as the objective function, and the sequential quadratic programming method is used to solve the nonlinear constraint optimization problems, to plan the time-optimal jerk continuous trajectory. Finally, the effectiveness of the proposed trajectory planning method is verified through the simulations.

The major contributions of this paper are as follows:

(1) The trajectory based on the 7th-order B-spline interpolation curve is designed, which ensures that the joint jerk is fully smooth, and the joint start–stop speed, acceleration and jerk can be set, overcoming the shortcoming that the trajectory interpolation method based on the quadratic polynomial plus cosine function needs to force the pulsation at the node to be 0.

(2) By using the convex hull property of the B-spline curve, the kinematic constraint of the manipulator is transformed into the control vertex constraint of the trajectory curve. The sequential quadratic programming method is used to solve the time-optimal trajectory planning problem, and then the time-optimal pulsating continuous trajectory is planned.

(3) The proposed strategy can also be used for online obstacle avoidance trajectory planning in complex environments and optimal trajectory planning with energy or time–energy performance as the optimization index.

The remainder of this paper is organized as follows: In Section 2, we introduce the construction idea of the jerk continuous trajectory. In Section 3, we transform the joint kinematic constraint into the control vertex constraint of the B-spline trajectory curve, to avoid the semi-infinite constraint problem of sampling and testing the trajectory curve, and then solve the time-optimal trajectory planning of the manipulator in Section 4. We present the simulation verification and result analysis in Section 5, and we summarize this paper and look forward to the future work in Section 6.

2. Jerk Continuous Trajectory Construction

According to the requirements of the trajectory tracking accuracy of a manipulator, a trajectory with a certain geometry in the task space is discretized, and the pose matrix sequence T_i and the corresponding time node sequence t_i are obtained. The joint position vector $\mathbf{p}_i$ corresponding to T_i is calculated by using the inverse kinematics, and then the joint position–time sequence is formed:

$$\mathbf{S} = (\mathbf{p}_i, t_i) \quad i = 0, 1, \cdots, n \tag{1}$$

where $\mathbf{p}_i = (p_{1i} p_{2i} \cdots p_{Ni})^{\mathrm{T}}$. The jerk continuous trajectory requires each joint trajectory $p_m = f_m(t)$ to be at least C^3 continuity, and $p_{mi} = f_m(t_i)$, that is, through the joint position sequence.

Adopting the k-th degree non-uniform rational B-spline with the C^{k-1} continuity structure, the joint trajectory for the manipulator can be uniformly described as

$$\mathbf{p}(u) = \sum_{j=0}^{n} \mathbf{d}_j N_{j,k}(u) = \sum_{j=i-k}^{i} \mathbf{d}_j N_{j,k}(u) \tag{2}$$

where $d_j \in R^{N\times 1}$ is the control top point vector, $j = 0, 1, \cdots, n$, $u \in [u_i, u_{i+1}] \subseteq [u_k, u_{n+1}] \subset [u_0, u_{n+k+1}]$, for the normalized time variable; $p(u) \in R^{N\times 1}$ is the joint position vector at the moment u; and $N_{j,k}(u)$ is the k-th degree canonical B-spline basis function.

The joint trajectory curve is related to the local support of the B-spline curve and the quality of C^{k-1} continuity. The joint velocity, acceleration and jerk at time $u \in [u_i, u_{i+1}]$ correspond to the R-order derivatives $p^r(u)$ of the B-spline trajectory curve, $r = 1, 2, 3$, respectively, which can be found by using the de Boer recurrence formula

$$\mathbf{p}^r(u) = \sum_{j=i-k+r}^{T} \mathbf{d}_j^r N_{j,k-r}(u) \tag{3}$$

$$\mathbf{d}_j^r = \begin{cases} \mathbf{d}_j, l = 0 \\ (k+1-l)\frac{\mathbf{d}_j^{l-1} - \mathbf{d}_{j-1}^{l-1}}{u_{j+k+1-l} - u_j}, l = 1, 2, \cdots, r, j = i-k+l, \cdots, i \end{cases} \tag{4}$$

Since the trajectory curve needs to strictly satisfy each position–time constraint, the control points of the B-spline interpolation trajectory curve need to be inverted according to this constraint. By normalizing the time node t_i through the cumulative duration parameterization method, we obtain the k-th degree B-spline trajectory domain node vector $u = (u_0, u_1, \cdots, u_{n+2k}) \in [0, 1]$, that is

$$u_0 = u_1 = \cdots = u_k = 0 \tag{5}$$

$$u_{n+k} = u_{n+k+1} = \cdots = u_{n+2k} = 1 \tag{6}$$

$$u_i = u_{i-1} + \frac{|\Delta t_{i-k-1}|}{\sum\limits_{j=0}^{n-1} |\Delta t_j|} \quad i = k+1, k+2, \cdots, n+k-1 \tag{7}$$

where $0 \leq u_0 \leq u_1 \leq u_2 \leq ... \leq u_{n+2k} \leq 1$. In this paper, we set u_0 and u_{n+k} to indicate multiple nodes, and the other nodes are distributed between u_k and u_{n+k}.

The $n+1$ equations corresponding to the position–time constraints can be listed as

$$\mathbf{p}(u_{i+k}) = \sum_{j=i}^{i+k} \mathbf{d}_j N_{j,k}(u_{i+k}) = \mathbf{p}_i \tag{8}$$

where $i = 0, 1, \cdots, n$, d_j is the $n+k$ control vertex vector of the B-spline trajectory, $j = 0, 1, \cdots, n+k-1$. The ideal joint trajectory requires that the joint start–stop velocity, acceleration and jerk can be set. Providing six boundary conditions (taking $k = 7$) results in $n+7$ equations:

$$\begin{cases} \mathbf{p}(u_{i+7}) = \sum\limits_{j=i}^{i+7} \mathbf{d}_j N_{j,7}(u_{i+7}) = \mathbf{p}_i \quad i = 0, 1, \cdots, n \\ \mathbf{p}'(u)|_{u=u_7} = \mathbf{v}_\mathrm{s}, \mathbf{p}'(u)|_{u=u_{n+7}} = \mathbf{v}_\mathrm{e} \\ \mathbf{p}''(u)|_{u=u_7} = \mathbf{a}_\mathrm{s}, \mathbf{p}''(u)|_{u=u_{n+7}} = \mathbf{a}_\mathrm{e} \\ \mathbf{p}'''(u)|_{u=u_7} = \mathbf{j}_\mathrm{s}, \mathbf{p}'''(u)|_{u=u_{n+7}} = \mathbf{j}_\mathrm{e} \end{cases} \tag{9}$$

where $\mathbf{v}_\mathrm{s}$, $\mathbf{a}_\mathrm{s}$ and $\mathbf{j}_\mathrm{s}$ are the starting velocity, acceleration and jerk vector, respectively; $\mathbf{v}_\mathrm{e}$, $\mathbf{a}_\mathrm{e}$ and $\mathbf{j}_\mathrm{e}$ are the stop velocity, acceleration and jerk vector, respectively; and $\mathbf{p}'(u)$, $\mathbf{p}''(u)$ and $\mathbf{p}'''(u)$, respectively, represent the joint velocity, acceleration and jerk trajectory curve.

The inverse equation of the trajectory curve control point of the MTH joint is described in the form of the following matrix equation:

$$\mathbf{A}_m \mathbf{d}_m = \mathbf{p}_m \tag{10}$$

where $\mathbf{A}_m \in R^{(n+7)\times(n+7)}$ is the coefficient matrix, and

$$\begin{cases} \mathbf{d}_m = \left(d_{m0}, d_{m1}, \cdots, d_{m(n+6)}\right)^{\mathrm{T}} \\ \mathbf{p}_m = \begin{pmatrix} p_{m0}, & p_{m1}, & \cdots, & p_{mn}, & v_{m\mathrm{s}}, & v_{m\mathrm{e}}, & a_{m\mathrm{s}}, & a_{m\mathrm{e}}, & j_{m\mathrm{s}}, & j_{m\mathrm{e}} \end{pmatrix}^{\mathrm{T}} \end{cases} \tag{11}$$

Define $A_m(n_r, n_c)$ as representing the n_c-column and n_r-row element in the matrix A_m. Then, there is

$$\mathbf{A}_m(n_{\mathrm{r}} n_{\mathrm{c}}) = N_{n_{\mathrm{r}}+n_{\mathrm{c}}-1,7}(u_{n_{\mathrm{r}}+7}) \tag{12}$$

where $n_r = 1, 2, \cdots, n-1, n_c = 1, 2, \cdots, 7$.

Other elements in A_m can be determined based on joint start–stop position, velocity, acceleration and jerk, and $d_0 = p_0, d_{n+6} = p_n, d_1^1 = v_{\mathrm{s}}, d_{n+6}^1 = v_{\mathrm{e}}, d_2^2 = a_{\mathrm{s}}, d_{n+6}^2 = a_{\mathrm{e}}, d_3^3 = j_{\mathrm{s}}$ and $d_{n+6}^3 = j_{\mathrm{e}}$, where $\mathbf{d_1^1}$, $\mathbf{d_2^2}$,$\mathbf{d_3^3}$,$\mathbf{d_{n+6}^1}$,$\mathbf{d_{n+6}^2}$ and $\mathbf{d}_{n+6}^3$ can be solved using Equation (4).

$$\mathbf{A}_m(n_{\mathrm{r}} n_{\mathrm{c}}) = N_{n_{\mathrm{r}}+n_{\mathrm{c}}-1,7}(u_{n_{\mathrm{r}}+7}) \tag{13}$$

According to (10), the control point vector of the seventh-degree B-spline trajectory curve can be obtained:

$$\mathbf{d}_m = \mathbf{A}_m^{-1} \mathbf{p}_m \tag{14}$$

According to the control vertex vector $\mathbf{d_m}$ and normalized time node vector $\mathbf{u}$, the joint trajectory with C^6 continuity for joint m passing through position $\mathbf{p}_i$ at time t_i can be obtained, and each joint trajectory has a uniform description form and is not limited by the number of joints N.

3. Transformation of Joint Kinematic Constraints

The time-optimal trajectory of the manipulator requires both the shortest execution time of the trajectory and the motion of each joint to satisfy the kinematic constraints. The velocity, acceleration and jerk constraints of each joint in the manipulator are c_{vm}, c_{am} and c_{jm}, respectively. From Equation (4), the velocity, acceleration and jerk curve equations of each joint can be obtained:

$$\begin{cases} \mathbf{v}(t) = \mathbf{p}'(u) = \sum\limits_{j=i-k+1}^{i} \mathbf{d}_j^1 N_{j,k-1}(u) \\ \mathbf{a}(t) = \mathbf{p}''(u) = \sum\limits_{j=i-k+2}^{i} \mathbf{d}_j^2 N_{j,k-2}(u) \\ \mathbf{j}(t) = \mathbf{p}'''(u) = \sum\limits_{j=i-k+3}^{i} \mathbf{d}_j^3 N_{j,k-3}(u) \end{cases} \tag{15}$$

where $\mathbf{d}_j^r = \begin{pmatrix} d_{1j}^r & d_{2j}^r & \cdots & d_{Nj}^r \end{pmatrix}^{\mathrm{T}}, \quad r = 1, 2, 3.$

The B-spline curve has the nature of the convex hull, and as a result, the trajectory curve of joint m satisfies the kinematic constraints:

$$|v_m(t)| \le c_{vm} \quad |a_m(t)| \le c_{am} \quad |j_m(t)| \le c_{jm} \tag{16}$$

Only the control points of the 7-th degree B-spline curve need to be satisfied:

$$\begin{cases} \max\left(\left|d_{mj}^{1}\right|\right) \le c_{vm} & j = 1,2,\cdots,n+6 \\ \max\left(\left|d_{mj}^{2}\right|\right) \le c_{am} & j = 2,3,\cdots,n+6 \\ \max\left(\left|d_{mj}^{3}\right|\right) \le c_{jm} & j = 3,4,\cdots,n+6 \end{cases} \tag{17}$$

where d_{mj}^1, d_{mj}^2 and d_{mj}^3 are the j-th control points of the B-spline velocity, acceleration and jerk curves of the m-th joint, respectively, which can be obtained from (4).

$$\begin{cases} \mathbf{d}_j^1 = 7\frac{\mathbf{d}_j - \mathbf{d}_{j-1}}{\mathbf{u}_{j+7} - \mathbf{u}_j} \\ \mathbf{d}_j^2 = \frac{42}{u_{j+6}-u_j}\left(\frac{\mathbf{d}_j - \mathbf{d}_{j-1}}{u_{j+7}-u_j} - \frac{\mathbf{d}_{j-1} - \mathbf{d}_{j-2}}{u_{j+6}-u_{j-1}}\right) \\ \mathbf{d}_j^3 = \frac{210}{\left(u_{j+5}-u_j\right)\left(u_{j+6}-u_j\right)}\left(\frac{\mathbf{d}_j - \mathbf{d}_{j-1}}{u_{j+7}-u_j} - \frac{\mathbf{d}_{j-1} - \mathbf{d}_{j-2}}{u_{j+6}-u_{j-1}} - \frac{\mathbf{d}_{j-1} - \mathbf{d}_{j-2}}{u_{j+6}-u_{j-1}} + \frac{\mathbf{d}_{j-2} - \mathbf{d}_{j-3}}{u_{j+5}-u_{j-2}}\right) \end{cases} \tag{18}$$

By transforming the joint kinematic constraints into the control vertex constraints of the B-spline trajectory curve, the semi-infinite constraint problem of sampling and checking the trajectory curve is avoided.

4. Solving the Time Optimal Problem

The time-optimal trajectory planning problem for the manipulator is to solve and normalize the sequence of time nodes corresponding to the time node vector u that minimizes the normalized total time, which is a nonlinear constrained optimization problem

$$\begin{aligned} & f(\mathbf{x}) = \min \sum_{i=0}^{n-1} x_i \\ & \text{s.t.} c_m(\mathbf{x}) = \max_{j=1,2,\cdots,n+6}\left\{\left|d_{mj}^1(\mathbf{x})\right|\right\} - c_{vm} \le 0 \\ & c_{N+m}(\mathbf{x}) = \max_{j=2,3,\cdots,n+6}\left\{\left|d_{mj}^2(\mathbf{x})\right|\right\} - c_{am} \le 0 \\ & c_{2N+m}(\mathbf{x}) = \max_{j=3,4,\cdots,n+6}\left\{\left|d_{mj}^3(\mathbf{x})\right|\right\} - c_{jm} \le 0 \end{aligned} \tag{19}$$

where $x = (x_0, x_1, \cdots, x_{n-1})^{\mathrm{T}}$ and $x_i = \triangle t_i = t_{i+1} - t_i$, where $\triangle t_i$ has the lower bound, that is, each element of $T_L = (\triangle t_{L0}, \triangle t_{L1}, \cdots, \triangle t_{L(n-1)})$ satisfies

$$\Delta t_{\mathrm{L}i} \ge \max_{m=1,2,\cdots,N}\left(\frac{p_{m(i+1)} - p_{mi}}{c_{vm}}\right) \tag{20}$$

Let

$$\begin{cases} k_1 = \max\left(\frac{d_{mj}^1}{c_{vm}}\right) & j = 1,2,\cdots,n+6 \quad m = 1,2,\cdots,N \\ k_2 = \max\left(\frac{d_{mj}^2}{c_{am}}\right) & j = 2,3,\cdots,n+6 \quad m = 1,2,\cdots,N \\ k_3 = \max\left(\frac{d_{mj}^3}{c_{jm}}\right) & j = 3,4,\cdots,n+6 \quad m = 1,2,\cdots,N \end{cases} \tag{21}$$

Determining the initial value of the time node vector according to k_1, k_2 and k_3, which can improve the search efficiency of the optimization algorithm, the initial value is

$$\mathbf{x}_0 = \max\begin{pmatrix} 1 & k_1 & \sqrt{k_2} & \sqrt[3]{k_3} \end{pmatrix} \times \mathbf{T}_{\mathrm{L}} \tag{22}$$

The sequential quadratic programming method with superlinear convergence performance is used to solve the nonlinear constrained optimization problem described in Equation (19). The Lagrangian function is constructed to linearize the nonlinear constraints. The Lagrangian function is given by

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda^{\mathrm{T}} C(\mathbf{x}) \tag{23}$$

where $\lambda = (\lambda_1 \lambda_2 \cdots \lambda_{3N})^{\mathrm{T}}$ is the Lagrange multiplier, $C(\mathbf{x}) = (c_1(\mathbf{x}) c_2(\mathbf{x}) \cdots c_{3N}(\mathbf{x}))^{\mathrm{T}}$. When the gradient of the Lagrange function is $\nabla L(\mathbf{x}^*, \lambda) = \nabla f(\mathbf{x}^*) - \lambda^{\mathrm{T}} \nabla C(\mathbf{x}^*) = 0$, $\mathbf{x}^*$ is the K-T point of the nonlinear optimization problem, that is, the solution of the time-optimal problem.

The KTH quadratic programming subproblem of the sequential quadratic programming method is obtained by simulating the Newton–Raphson method:

$$\begin{array}{c} \min_{d \in \mathbf{R}^n} \left({\mathbf{g}_k}^{\mathrm{T}} \mathbf{d} + \frac{1}{2} \mathbf{d}^{\mathrm{T}} \mathbf{B}_k \mathbf{d} \right) \\ \text{s.t.} \quad a_i(\mathbf{x}_k)^{\mathrm{T}} \mathbf{d} + c_i(\mathbf{x}_k) \le 0 \quad i = 1, 2, \cdots, 3N \end{array} \tag{24}$$

where $A(\mathbf{x}_k) = (a_1(\mathbf{x}_k), a_2(\mathbf{x}_k), \cdots, a_{3N}(\mathbf{x}_k))^{\mathrm{T}} = \nabla C(\mathbf{x}_k)^{\mathrm{T}}$, $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)$ and $\mathbf{B}_k \in \mathbf{R}^{n \times n}$ is an approximation of the Hessian matrix of the Lagrangian function.

Through successive solving of the quadratic programming problems, the optimal trajectory of manipulator time node vector x converges to the optimal value x^*.

5. Simulation Results and Analysis

In order to verify the effectiveness of the proposed trajectory planning method, a 6-DOF serial manipulator is used as the test object to plan the time-optimal jerk continuous trajectory over the specified joint position sequence. The sequence of joint positions passed by the manipulator is shown in Table 1, and the kinematic constraints of each joint are shown in Table 2.

Table 1. Joint position sequence.

Node	Position of Each Joint p					
	1	2	3	4	5	6
p_0	43.35	7.37	130.57	0	39.06	−46.66
p_1	43.33	−18.00	152.09	0	45.90	48.02
p_2	50.04	−41.85	170.66	0	51.19	−32.35
p_3	62.67	−53.76	179.41	0	53.59	−4.99
p_4	78.04	−57.32	182.70	0	54.62	33.04
p_5	94.40	−52.73	178.45	0	53.38	75.94
p_6	104.13	−42.46	173.54	0	50.33	94.76
p_7	111.91	6.79	132.80	0	40.41	112.16

Table 2. Kinematical constraint.

Constraint Term	Constraint Value of Each Joint					
	1	2	3	4	5	6
Velocity $v/\left((^\circ) \cdot \mathrm{s}^{-1}\right)$	100	95	100	150	130	110
Acceleration $a/\left((^\circ) \cdot \mathrm{s}^{-2}\right)$	45	40	75	70	90	80
Jerk $j/\left(((^\circ) \cdot \mathrm{s}^{-3}\right)$	60	60	55	70	75	70

Set the joint start–stop velocity, acceleration and jerk to 0, and adopt the sequential quadratic programming method to obtain the optimal time node vector as t = (0, 4.71, 8.11, 9.03, 9.73, 10.67, 11.24, 15.18) and the normalized time node vector as u = (0, 0, 0, 0, 0, 0, 0, 0, 0.31, 0.53, 0.60, 0.64, 0.70, 0.74, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00). We can see that the execution time of the trajectory is only 15.18 s. The control vertices of the time-optimal trajectory based on the 7-th degree B-spline curve are shown in Table 3.

Table 3. B-spline trajectory control vertices.

Vertex	Angle Value of Vertex *d*					
d_0	43.35	7.37	130.57	0	39.06	−46.66
d_1	43.35	7.37	130.57	0	39.06	−46.66
d_2	43.35	7.37	130.57	0	39.06	−46.66
d_3	43.35	7.37	130.57	0	39.06	−46.66
d_4	2.92	−65.47	164.59	0	53.91	168.89
d_5	113.54	31.87	156.25	0	40.42	51.19
d_6	−2.45	−51.77	157.32	0	51.42	−135.90
d_7	103.17	−60.24	199.65	0	55.77	71.38
d_8	87.62	−58.12	165.17	0	55.42	89.92
d_9	140.43	−11.57	178.72	0	39.28	135.04
d_{10}	111.91	6.79	132.80	0	40.41	112.16
d_{11}	111.91	6.79	132.80	0	40.41	112.16
d_{12}	111.91	6.79	132.80	0	40.41	112.16
d_{13}	111.91	6.79	132.80	0	40.41	112.16

The joint position, velocity, acceleration and jerk trajectory curves of the time-optimal trajectory are shown in Figures 1–4, where the sequence of joint positions that each joint must pass through is marked by dots. We can see that the planned time-optimal trajectory passes through the specified joint position sequence, and the starting and stopping velocity, acceleration and jerk of each joint are equal to the set value. By comparing the proposed trajectory design method with the trajectory interpolation method combining the quadratic polynomial and cosine function, we can set the start–stop velocity, acceleration and jerk of the trajectory, without forcing the jerk at the node to be 0 like the latter. The joint position, velocity, acceleration and jerk are continuous and satisfy the kinematic constraints, and only the jerk of the sixth joint is close to the constraint value at the maximum value. Compared with the performance in [33] that the trajectory execution time is about 18.91 s, under the conditions of the same joint position sequence and kinematic constraints, the velocity, acceleration and jerk of the time-optimal trajectory planned in this paper at multiple joints meet the kinematic constraints, and the trajectory execution time is shorter.

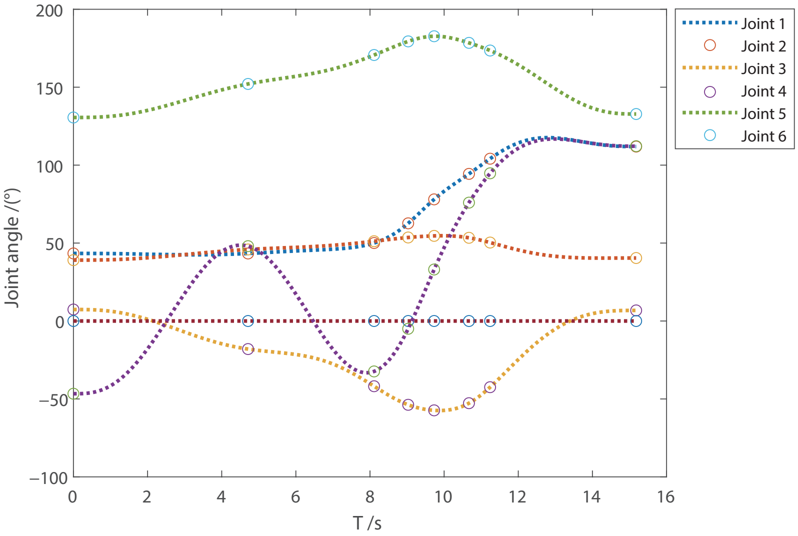

Figure 1. Joint angle.

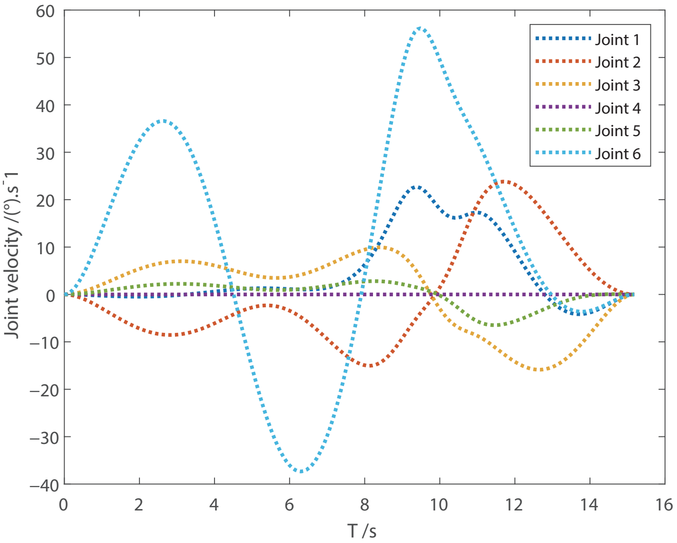

Figure 2. Joint velocity.

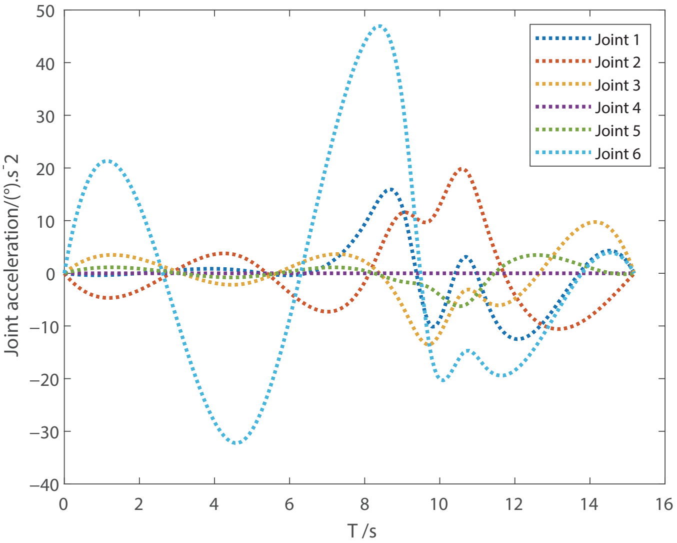

Figure 3. Joint acceleration.

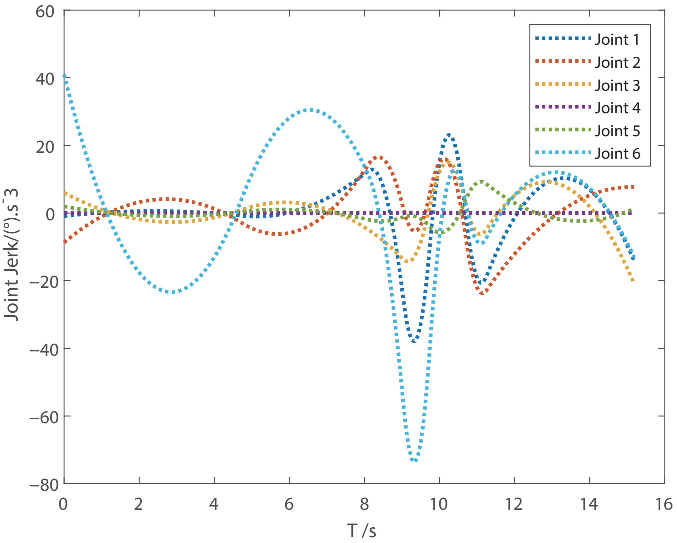

Figure 4. Joint Jerk.

The multi-objective particle swarm optimization (MOPSO) algorithm is widely used in multi-objective optimization because of its fast convergence speed. Thus, in order to compare with the method proposed in this paper, we choose the MOPSO algorithm to solve the above optimal problem. Under the same conditions, we use the algorithm to obtain the optimal trajectory, and the results are shown in Figures 5–8. Compared with Figures 1–4, the execution time of the optimal trajectory obtained using the sequential quadratic programming method is significantly shorter than that obtained using MOPSO.

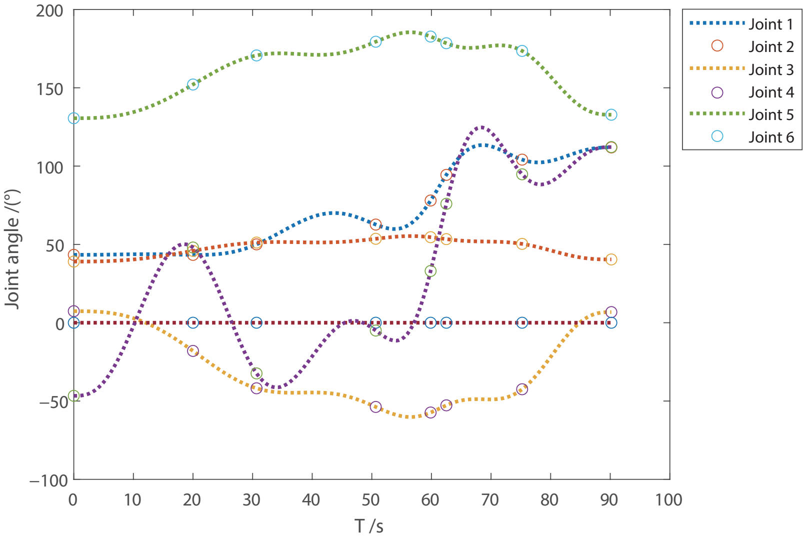

Figure 5. Joint angle with MOPSO.

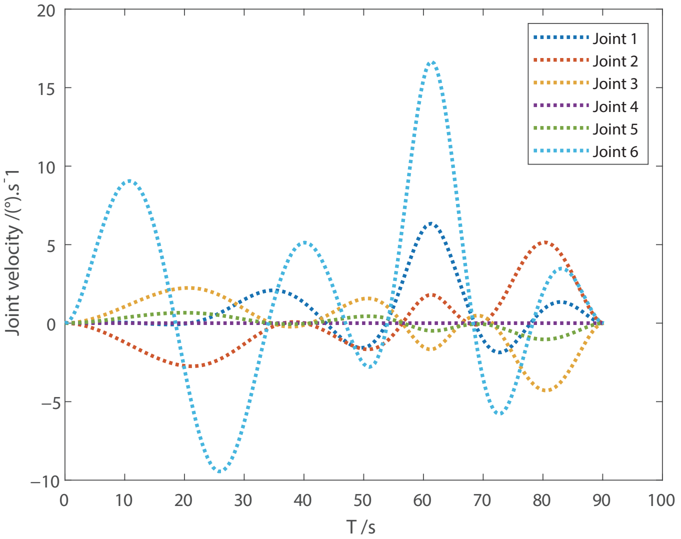

Figure 6. Joint velocity with MOPSO.

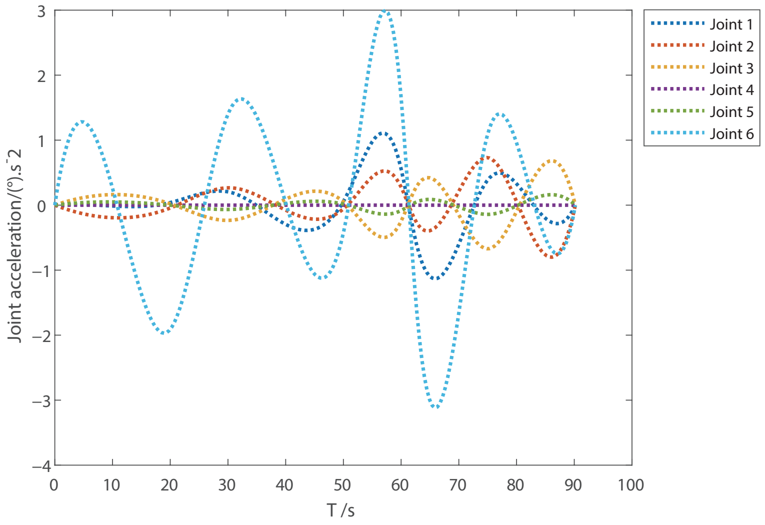

Figure 7. Joint acceleration with MOPSO.

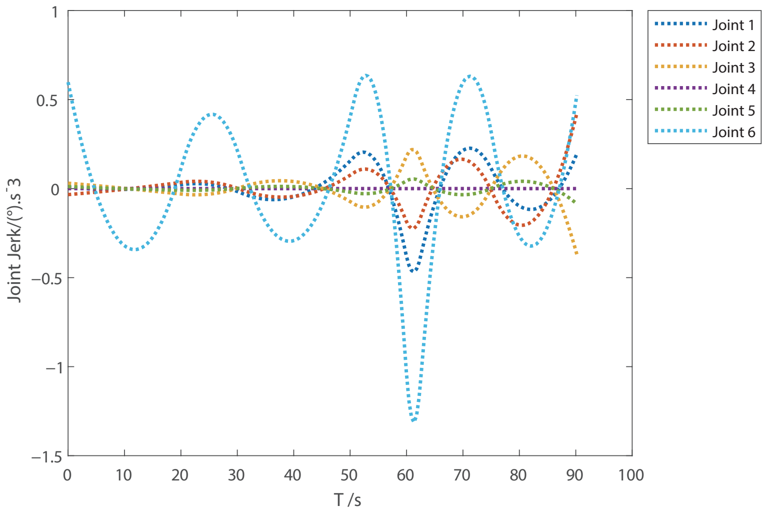

Figure 8. Joint jerk with MOPSO.

6. Conclusions

In this paper, we proposed a manipulator trajectory planning method based on the seventh-order B-spline interpolation curve, which ensures that the joint velocity, acceleration and jerk are continuous. We transformed the motion constraint of the manipulator into the control vertex constraint of the trajectory curve, and used the sequential quadratic programming method to solve the time-optimal trajectory planning meeting the motion reduction bundle, obtaining the time-optimal jerk continuous trajectory. The simulation results verify the effectiveness of the proposed trajectory planning algorithm.

At present, we are building a 6-DOF robot experiment platform. In the future, we will apply the method proposed in this paper to the experimental platform, further improve the control strategy proposed in this paper to provide a practical and reliable control algorithm for 6-DOF robots, and greatly improve the operational efficiency of the robot arm.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z. and G.H.; software, Z.H. and X.C.; validation, Y.Z., G.H., Z.W., Z.H. and X.C.; formal analysis, Y.Z. and G.H.; investigation, Z.H. and X.C.; resources, Y.Z.; writing—original draft preparation, Y.Z., G.H. and Z.W.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by (1) Hubei Province Technology Innovation Key Research and Development Project (No. 2023BAB071); (2) National Natural Science Foundation of China (No. 52205536); (3) Hubei Province Central Government Guide Local Science and Technology Development Project (No. 2022BGE035); and (4) Hubei Province Nature Science Foundation (No. 2023AFB380).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Siliciano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*; Springer: New York, NY, USA, 2010; pp. 415–418.
2. Kohrt, C.; Stamp, R.; Pipe, A.; Kiely, J.; Schiedermeier, G. An online robot trajectory planning and programming support system for industrial use. *Robot. Comput.-Integr. Manuf.* **2013**, *29*, 71–79. [CrossRef]
3. Madridano, A.; Al-Kaff, A.; Martín, D.; De La Escalera, A. Trajectory planning for multi-robot systems: Methods and applications. *Expert Syst. Appl.* **2021**, *173*, 114660. [CrossRef]
4. Zhang, Y.; Sun, Z.; Sun, Q.; Wang, Y.; Li, X.; Yang, J. Time-jerk optimal trajectory planning of hydraulic robotic excavator. *Adv. Mech. Eng.* **2021**, *13*, 16878140211034611. [CrossRef]
5. Wu, G.; Zhao, W.; Zhang, X. Optimum time-energy-jerk trajectory planning for serial robotic manipulators by reparameterized quintic NURBS curves. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *235*, 4382–4393. [CrossRef]
6. Zhou, J.; Cao, H.; Jiang, P.; Li, C.; Yi, H.; Liu, M. Energy-saving trajectory planning for robotic high-speed milling of sculptured surfaces. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 2278–2294. [CrossRef]
7. Rubio, F.; Valero, F.; Sunẽr, J.L.S.; Mata, V. A comparison of algorithms for path planning of industrial robots. In Proceedings of the EUCOMES 08: The Second European Conference on Mechanism Science, Cassino, Italy, 17–20 September 2008 ; Springer: Berlin/Heidelberg, Germany, 2009; pp. 247–254.
8. Valero, F.; Mata, V.; Besa, A. Trajectory planning in workspaces with obstacles taking into account the dynamic robot behaviour. *Mech. Mach. Theory* **2006**, *41*, 525–536. [CrossRef]
9. Heo, H.J.; Son, Y.; Kim, J.M. A trapezoidal velocity profile generator for position control using a feedback strategy. *Energies* **2019**, *12*, 1222. [CrossRef]
10. Wu, J.; Wang, D.; Wang, L. A control strategy of a two degrees-of-freedom heavy duty parallel manipulator. *J. Dyn. Syst. Meas. Control* **2015**, *137*, 061007. [CrossRef]
11. Gasparetto, A.; Zanotto, V. Optimal trajectory planning for industrial robots. *Adv. Eng. Softw.* **2010**, *41*, 548–556. [CrossRef]
12. Zanotto, V.; Gasparetto, A.; Lanzutti, A.; Boscariol, P.; Vidoni, R. Experimental validation of minimum time-jerk algorithms for industrial robots. *J. Intell. Robot. Syst.* **2011**, *64*, 197–219. [CrossRef]
13. Gallant, A.; Gosselin, C. Extending the capabilities of robotic manipulators using trajectory optimization. *Mech. Mach. Theory* **2018**, *121*, 502–514. [CrossRef]
14. Seki, H.; Tadakuma, S. Minimum jerk control of power assisting robot on human arm behavior characteristic. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands, 10–13 October 2004; Volume 1, pp. 722–727.
15. Hermand, E.; Nguyen, T.W.; Hosseinzadeh, M.; Garone, E. Constrained control of UAVs in geofencing applications. In Proceedings of the 2018 26th Mediterranean Conference on Control and Automation (MED), Zadar, Croatia, 19–22 June 2018; pp. 217–222.
16. Zhang, T.; Zhang, M.; Zou, Y. Time-optimal and smooth trajectory planning for robot manipulators. *Int. J. Control. Autom. Syst.* **2021**, *19*, 521–531. [CrossRef]
17. Su, T.; Cheng, L.; Wang, Y.; Liang, X.; Zheng, J.; Zhang, H. Time-optimal trajectory planning for delta robot based on quintic pythagorean-hodograph curves. *IEEE Access* **2018**, *6*, 28530–28539. [CrossRef]
18. Bureerat, S.; Pholdee, N.; Radpukdee, T.; Jaroenapibal, P. Self-adaptive MRPBIL-DE for 6D robot multiobjective trajectory planning. *Expert Syst. Appl.* **2019**, *136*, 133–144. [CrossRef]
19. Raheem, F.A.; Sadiq, A.T.; Abbas, N.A.F. Robot arm free Cartesian space analysis for heuristic path planning enhancement. *Int. J. Mech. Mechatron. Eng* **2019**, *19*, 29–42.
20. Gasparetto, A.; Boscariol, P.; Lanzutti, A.; Vidoni, R. Path planning and trajectory planning algorithms: A general overview. In *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 3–27.
21. Liu, L.; Yao, W.; Guo, Y. Prescribed performance tracking control of a free-flying flexible-joint space robot with disturbances under input saturation. *J. Frankl. Inst.* **2021**, *358*, 4571–4601. [CrossRef]
22. Geisslinger, M.; Poszler, F.; Lienkamp, M. An ethical trajectory planning algorithm for autonomous vehicles. *Nat. Mach. Intell.* **2023**, *5*, 137–144. [CrossRef]

23. He, S.; Hu, C.; Lin, S.; Zhu, Y. An online time-optimal trajectory planning method for constrained multi-axis trajectory with guaranteed feasibility. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7375–7382. [CrossRef]
24. Yu, X.; Dong, M.; Yin, W. Time-optimal trajectory planning of manipulator with simultaneously searching the optimal path. *Comput. Commun.* **2022**, *181*, 446–453. [CrossRef]
25. Huang, P.; Xu, Y.; Liang, B. Global minimum-jerk trajectory planning of space manipulator. *Int. J. Control. Autom. Syst.* **2006**, *4*, 405–413.
26. Lin, C.; Chang, P.; Luh, J. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Trans. Autom. Control* **1983**, *28*, 1066–1074. [CrossRef]
27. Bazaz, S.A.; Tondu, B. Minimum time on-line joint trajectory generator based on low order spline method for industrial manipulators. *Robot. Auton. Syst.* **1999**, *29*, 257–268. [CrossRef]
28. Gasparetto, A.; Lanzutti, A.; Vidoni, R.; Zanotto, V. Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning. *Robot. Comput.-Integr. Manuf.* **2012**, *28*, 164–181. [CrossRef]
29. Fang, S.; Ma, X.; Zhao, Y.; Zhang, Q.; Li, Y. Trajectory planning for seven-DOF robotic arm based on quintic polynormial. In Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 24–25 August 2019; Volume 2, pp. 198–201.
30. Kucuk, S. Optimal trajectory generation algorithm for serial and parallel manipulators. *Robot. Comput.-Integr. Manuf.* **2017**, *48*, 219–232. [CrossRef]
31. Gasparetto, A.; Zanotto, V. A new method for smooth trajectory planning of robot manipulators. *Mech. Mach. Theory* **2007**, *42*, 455–471. [CrossRef]
32. Freeman, P. *Minimum Jerk Trajectory Planning for Trajectory Constrained Redundant Robots*; Washington University in St. Louis: St. Louis, MO, USA, 2012.
33. Tan, G.; Wang, Y. Theoretical and experimental research on time-optimal trajectory planning and control of industrial robots. *Control Theory Appl.* **2003**, *20*, 185–192.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.