Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5349426, 11 pages
https://doi.org/10.1155/2022/5349426

*Research Article*

# A Time-Optimal Intersection Search Algorithm for Robot Grasping

**Jun Wang, Kai Zhang, Mingquan Yang, Xuwei Zhang, and Fei Liang**

*School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China*

Correspondence should be addressed to Jun Wang; junwang@hbut.edu.cn

Received 7 October 2022; Revised 12 November 2022; Accepted 1 December 2022; Published 15 December 2022

Academic Editor: Zhi-Wei Liu



In industrial automation, an important task of robot system is to grasp moving objects. In order to grasp the moving target on the conveyor belt in the shortest time, this paper proposes a method of location prediction and interception grasping of the target object, and puts forward the corresponding time-optimal grasping point search algorithm. In order to reduce the motion impact of the manipulator and meet the kinematic constraints, the velocity planning curve of the normalized quintic polynomial was adopted to plan the motion of the manipulator in the joint space. In order to express the algorithm more clearly, the position-time function of the moving object and the position-time virtual function of the robot are introduced, and the time optimal grasping point appears at the intersection of the two functions. Finally, the method is verified by simulation analysis.

## 1. Introduction

With the rapid development of industrial robot technology, robots have become the core supportive equipment in intelligent manufacturing. In industrial automation, it is an important task of robot system to grasp target objects quickly, and grasping the target objects on the conveyor belt has become a routine task in automated production. In recent years, many scholars have proposed different studies on grasping moving objects with robots, and these studies can be divided into three methods.

The first method is vision-based control or visual servo tracking of moving objects. Vision-based control or vision servos are used as solutions in the robotics industry to improve the flexibility and intelligence of robotic systems, especially in unstructured environments. Visual servo control for tracking and intercepting maneuvering targets consists mainly of two different approaches, namely position-based visual servo (PBVS) and image-based visual servo (IBVS) [1, 2]. Vision servo systems are ideal for tracking maneuverable targets. Krishnan and Sankar [3] proposed an image-based visual servo which completes the tracking of two-dimensional image features. Keshmiri and Xie [4] proposed a new semioffline trajectory planning method with image-based visual servos, which expanded the operating space of the system compared with traditional visual servos. Stogl and Zumkeller [5] proposed a moving object tracking method using external depth sensors to carry out 3D reconstruction and capture of unknown objects.

The second approach is based on the prediction, planning, and execution (PPE) methodology. In the PPE method, the motion of the object is first predicted. Then, the robot motion trajectory is planned and the planned trajectory is executed to intercept the object. The PPE method is employed to intercept nonmaneuvering target objects (A target object is considered to be nonmaneuvering if it moves on a continuous path with constant velocity or acceleration). Adaptive prediction planning and execution (APPE) [6] technology is an improvement based on PPE technology, which can continuously adjust the grasping point according to changes in the target trajectory. Croft et al. [7] proposed an adaptive planning algorithm to search for the best grasping points, and the potential grasping points are not limited to a small number of candidate points. Fernandes and Lima [8] employed the APPE method to capture ping pong balls rolling on the table. Keshmiri et al. [9] proposed an online trajectory planning method based on APPE for robotic manipulators considering velocity and torque constraints. Wong and Chien [10] designed a moving object prediction and grasping system that enables a robot manipulator using a two-finger gripper to grasp moving objects on a conveyor and a circular rotating platform. Islam et al. [11] proposed a constant time motion planning algorithm applied to the belt grab so that the manipulator can complete the grab in constant time. Borangiu et al. [12] presented a correlation algorithm for visual robot tracking moving objects on a conveyor belt. Ye and Liu [13] proposed a velocity decomposition algorithm to capture objects whose transport trajectory cannot be predicted in the long time.

The third approach is based on navigation and guidance theory to intercept moving objects. Navigation-guided interception has been harnessed for nearly 50 years to track and intercept maneuvering objects, such as guided missiles that track and intercept rockets. Due to its simplicity, proportional navigation guidance (PNG) has become the most widely used guidance law [14]. In recent years, researchers have tried to involve the laws of guidance in allowing robots to intercept moving objects. Unlike missile interception, robot interception needs to be carried out smoothly. The velocities of the robot and the moving object must match at the rendezvous to achieve the smooth interception of moving objects. Mehrandezh et al. [15, 16] proposed a hybrid interception scheme that combines navigation-based interception techniques with traditional trajectory tracking methods. Su and Xi [17] also proposed a guidance-based strategy for moving target interception. Yuan and Chern [18] proposed a new guidance scheme for ideal proportional navigation guidance (IPNG). Compared to PNG-based navigation techniques, IPNG exhibits better mathematical process ability and lower sensitivity to track and circumvent the initial conditions of the problem. IPNG is a more suitable method for robots capturing and intercepting mobile objects. Keshmiri and Xie [19] proposed a strategy combining the navigation algorithm with augmented image-based visual services (AIBVS), which can quickly capture moving objects. Keshmiri [20] conducted a comparative analysis of various navigation guidance methods and proposed an improved version of the AIPNG method to solve two-dimensional problems and three-dimensional problems.

In industrial automation, it has become an important research direction to effectively shorten the movement time of robotic manipulators, improve working efficiency, and realize the grasping of moving objects on the conveyor belt in the shortest time. Almost all current algorithms have the problem of path limitation: the algorithm constantly searches for the optimal motion time of the robotic manipulator on a particular motion path. For example, Kim and Elizabeth [21] proposed an online near time-optimal trajectory planning for industrial robots. The time optimal planning of the known path is completed by combining T-shaped velocity curve with dynamic constraints. In the known research results, scholars mostly use the tracking capture method to grasp moving objects. This method has a long grasping cycle and is not suitable for rapid industrial production.

Therefore, this paper proposes a method to predict the position of the target object and later intercept as well grasp the target object. Hence, the corresponding intersection search algorithm is put forward to greatly reduce the period of completion of the capture task. In order to avoid large motion impact during operation and satisfy motion constraints in joint space, normalized quintic polynomial velocity curve was adopted in this paper. Considering that the robot arrival function of time cannot be exhaustive through inverse kinematics, a fast convergent search algorithm is proposed.

The rest of this paper is organized as follows. Section 2 introduces the composition of the grasping system. Section 3 introduces the trajectory planning of normalized quintic polynomial. Section 4 introduces the position-time functions of the robot and the target object. Section 5 introduces two intersection search algorithms, respectively. In Section 6, the application of time optimal intersection search algorithm in grasping system is simulated and analyzed. The conclusion is given in Section 7.

## 2. Conveyor Belt Automated Production Line Grasping System

Figure 1 shows a typical conveyor belt automated production line, where objects are supplied in a continuous manner on the conveyor belt. An automated production line system that completes the automatic sorting work requires the following systems:

(1) The vision system, with a fixed CCD camera, acquires images of target objects

(2) The image processing system, which processes images of target objects, acquires of the poses of the target objects and communication with the robot control system

(3) The control system, which generates the robotic movement trajectory, controls the robot moving

(4) The execution system, a robot

(5) The conveyor belt platform

In practical applications, in order to avoid cable winding problems on constraints near the robot base, most robot manufacturers usually apply a motion limit on the axis of rotation, allowing only one direction, clockwise (CW) or counterclockwise (CCW). Therefore, object-capture systems are divided into two types, interception-based capture and tracking-based capture, which are shown in Figure 2. In order to achieve the capture of the target object with the robot in the shortest possible time, it is necessary to search for the best grasping point of the target object on the conveyor belt. This paper focuses on interception-based capture and proposed a time-optimal search algorithm for the grasping point.

## 3. Trajectory Planning

The point-to-point motion control of the robot, referred to as PTP control, only needs to perform path planning according to the pose information of the starting point and

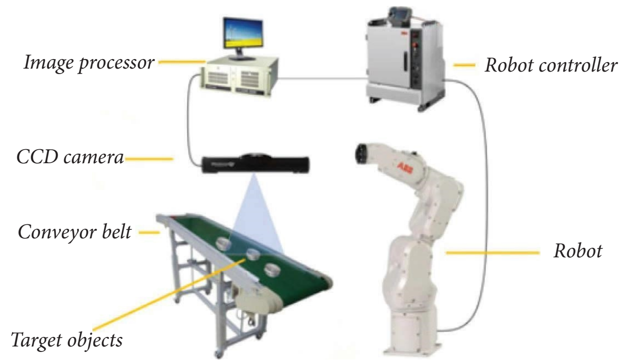



FIGURE 1: A typical conveyor belt automated production line.

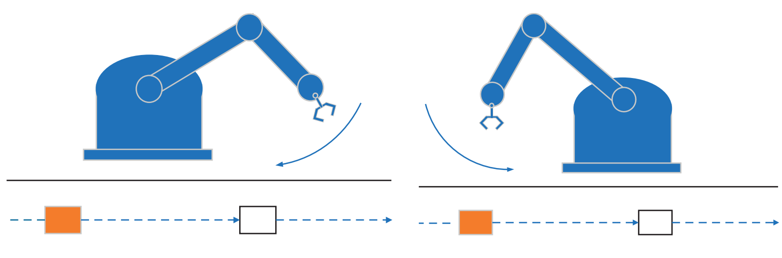



FIGURE 2: (a) Interception-based capture; (b) tracking-based capture.

the target point, and does not concern the intermediate trajectory. This control method is relatively simple and has been widely applied to handing robots. Common velocity planning includes trapezoidal profile, S-Curve/Profile, and polynomial profile.

The acceleration curve of trapezoidal profile is discontinuous so that the acceleration, uniform velocity, and deceleration process cannot achieve a smooth transition. The jump phenomenon leads to a large mechanical impact when the joint motors working. The feed drive system's vibrating not only causes damage to the motors but also wears down the robot joints and affects its service life. Therefore, trapezoidal profile is usually used in low-velocity, low-cost motion control processes.

Compared with the trapezoidal profile, the S-Profile has more stable acceleration and deceleration, and has less impact on the motor and transmission system, but at the same maximum speed, it takes longer to travel the same distance. In the S-Profile, its acceleration curve is T-shaped; in other words, when the jerk in the S-Profile is large enough, the S-Profile becomes a trapezoidal profile. And S-Profile takes more time in operation.

The mathematical expressions of trapezoidal profile and S-Profile are both piecewise functions, while polynomial profile generally refers to a curve that can be expressed by a single expression. According to different constraints, 3rd and 5th degree polynomials are commonly used. In the robot system, the simple polynomial programming has a very serious problem: there is no uniform velocity segment. As can be seen from Figure 3, simple polynomial programming cannot provide uniform speed control according to the desired speed, and in most robot applications, speed control processing is required. For multiaxis systems without coupling, since the motor capacity and load are known, a maximum achievable acceleration can be calculated so that traditional trapezoidal or S-shaped acceleration planning can be used to achieve very satisfactory results. For a multiaxis coupled system such as a robot, the load inertia of each joint is constantly changing during the movement process. Hence, traditional trapezoidal or parabolic planning cannot meet the needs well. With comprehensive consideration, the normalized quintic polynomial is adopted to carry out the time-optimized trajectory planning of the robot.

For the joint axis $i$, the joint displacement $\theta_i$ is obtained through the inverse kinematics of the robot. A normalized quintic polynomial is given: $s\,(u)$ ($u = t/T$, is a normalized variable); speed limit is represented by $\omega_{\lim}$; acceleration limit is represented by $a_{\lim}$; the movement time of joint $i$ represented by $T$ is obtained.
The normalized quintic polynomial is expressed in (1).

$$s(u) = b_0 + b_1 u + b_2 u^2 + b_3 u^3 + b_4 u^4 + b_5 u^5, u \in [0, 1]. \quad (1)$$

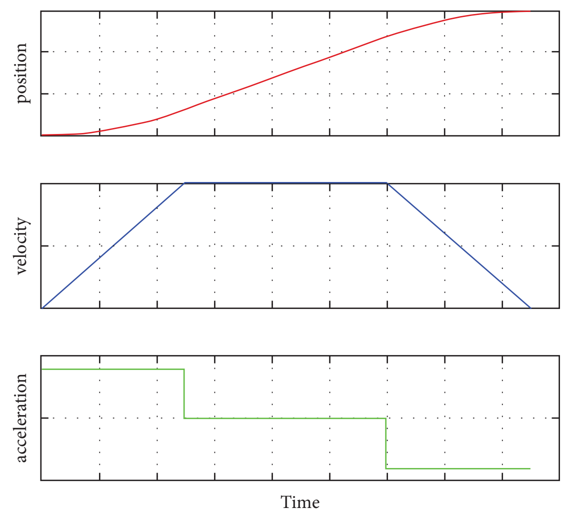
The first derivative of (1) is (2).

$$s'\,(u) = b_1 + 2b_2 u + 3b_3 u^2 + 4b_4 u^3 + 5b_5 u^4. \quad (2)$$

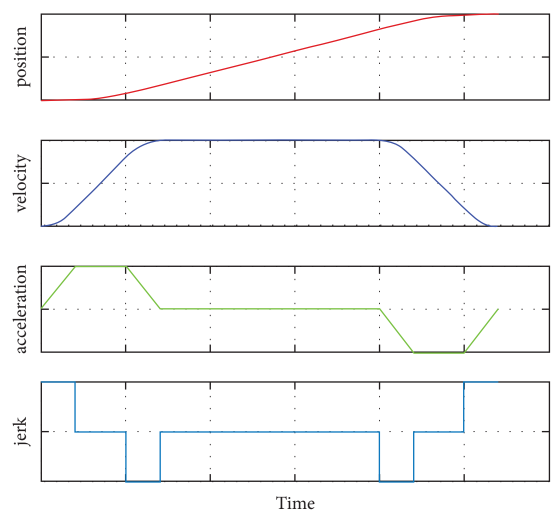
The second derivative of (1) is (3).

$$s''\,(t) = 2b_2 + 6b_3 u + 12b_4 u^2 + 20b_5 u^3. \quad (3)$$

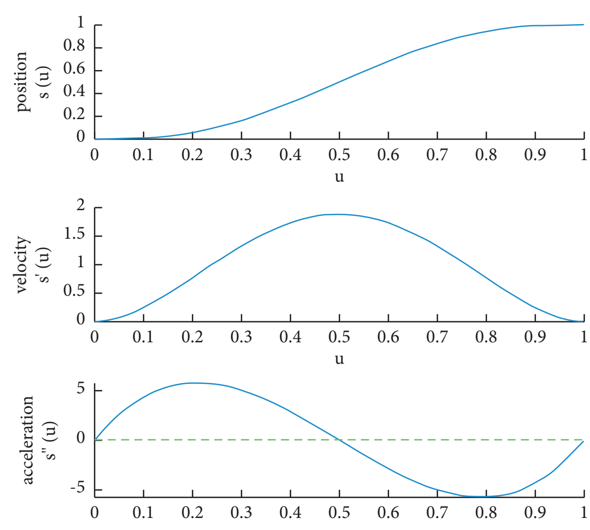
According to the constraints of the joint axis at $t = 0$ and $t = T$, (4) can be obtained.




(a)




(b)




(c)

Figure 3: Trajectory planning: (a) trapezoidal profile; (b) S-Curve/Profile; (c) polynomial profile.

$$\begin{cases} s(0) = 0, \\ s'(0) = 0, \\ s''(0) = 0, \\ s(1) = 1, \\ s'(1) = 0, \\ s''(1) = 0. \end{cases} \quad (4)$$

According to equation (4), equation (5) can be obtained.

$$\begin{cases} b_0 = 0, \\ b_1 = 0, \\ b_2 = 0, \\ b_0 + b_1 + b_2 + b_3 + b_4 + b_5 = 1, \\ b_1 + 2b_2 + 3b_3 + 4b_4 + 5b_5 = 0, \\ 2b_2 + 6b_3 + 12b_4 + 20b_5 = 0. \end{cases} \quad (5)$$

As per equation (5), $b_0 = 0$, $b_1 = 0$, $b_2 = 0$, $b_3 = 10$, $b_4 = 15$, and $b_5 = 6$; the normalized quintic polynomial is obtained.

$$\begin{aligned} s(u) &= 10u^3 - 15u^4 + 6u^5, \\ s'(u) &= 30u^2 - 60u^3 + 30u^4, \\ s''(u) &= 60u - 180u^2 + 120u^3. \end{aligned} \quad (6)$$

With the second derivative of (1) equaling one, $s'(u) = 60u - 180u^2 + 120u^3 = 0$, $u = 0,1/2,1$ is obtained. Since the velocity is 0 at the beginning and end, that is $s'(0) = s'(1) = 0$, the motor reaches maximum speed at $u = 1/2$, $s'_{\max}(u) = s'_{\max}(1/2) = 1.875$.

With the third derivative of (2) equaling 0, $s''(u) = 6 - 360u + 360u^2 = 0$, $u = (3 \pm \sqrt{3})/6$ is obtained. The maximum acceleration value of the normalized quintic polynomial is

$$\begin{aligned} s''(u) &= \max\left(\left|s''\left(\left(\frac{3+\sqrt{3}}{6}\right)\right)\right|, \left|s''\left(\left(\frac{3-\sqrt{3}}{6}\right)\right)\right|\right) \\ &= 5.7735. \end{aligned} \quad (7)$$

The maximum velocity $\omega_{\max}$ and maximum acceleration $a_{\max}$ of the quintic polynomial trajectory $\theta(t)$ should satisfy equation (7).

$$\begin{cases} \omega_{\max} \le \omega_{\lim}, \\ a_{\max} \le a_{\lim}, \end{cases} \quad (8)$$

Because $\theta(t)/\theta_i = s(u)$, if we take the first derivative of both sides of this equation, we get $\omega(t) = \theta_i \cdot s'(u)du/dt = \theta_i/T^2 \cdot s'(u)$; the second derivative function is also obtained as $a(t) = \theta_i/T \cdot s'(u)du/dt = \theta_i/T^2 \cdot s''(u)$.

According to equation (8), for all

$$\begin{cases} \dfrac{\theta_i}{T} s'(u) \le \omega_{\lim}, & T \ge \dfrac{\theta_i s'(u)}{\omega_{\lim}}, \\ & \Rightarrow \\ \dfrac{\theta_i}{T^2} s''(u) \le a_{\lim}, & T \ge \sqrt{\dfrac{\theta_i s''(u)}{a_{\lim}}}. \end{cases} \quad (9)$$

The previous calculations only consider the case of single-axis motion. When multiaxis synchronization is required, the axis with the longest motion time (related to factors such as the maximum speed and motion displacement of each axis) should be considered, and the longest motion time will be used as the time for synchronous motion time.

The synchronized movement time is $T = \max(T_i)$.

## 4. Position-Time Functions

To describe the minimum time that the robot takes to grasp the target object, and to search for the best grasping point, the position-time function of the robot is defined as $t_r(x)$: the time required for the manipulator to reach any point on the conveyor belt. The position-time function of the target object is defined as $t_o(x)$: the time for target object takes to reach any point on the conveyor belt, as is shown in Figure 4.

Typically, the target object moves at a uniform velocity on the conveyor belt, and the conveyor's velocity is known. Therefore, the target object's position-time function can be expressed as a polynomial, which is described by equation (9).

$$t_o(x) = c_1 \cdot x + c_2 \; c_1, c_2 \in R. \quad (10)$$

The robot's time function of arrival, $t_r(x)$, is usually preunknown. Of course, if enough midpoints are obtained and interpolated to approximate the robot's position-time function, the exact form of the robot's position-time function can be identified. However, the midpoints need to be calculated via inverse kinematics, which is complex and cannot be performed in real time. In this paper, a real-time search algorithm for time-optimal capture points is proposed and an improved version is given.

To grasp the object in the shortest possible time, the robot should arrive the grasping point at the same time as the object. If the robot's time curve and the object's time curve are both monotonically increasing or decreasing, there is a unique intersection point, which is the best place for the robot to capture the object in the minimum time [22–24].

## 5. The Algorithm for Searching Grasping Point

A time optimal search algorithm is proposed for grasping point [25]. To use the time optimal search algorithm, two factors need to be fully considered.

(1) The position time curve of manipulator and the position-time curve of target have monotony to ensure the existence of intersection point; (2) the speed of the robot is generally faster than the speed of the conveyor belt guaranteeing the difference in the velocities of the robot and the target object.

*5.1. The Pyramid-Shaped Time Optimal Search Algorithm.* Firstly, the appropriate working area is selected. When the robot reaches $x_{f_1}$, the target object reaches the point $x_{i_1}$; $x_{m_1}$ is the midpoint between $[x_{i_1}, x_{f_1}]$; and $x_{m_1} = (x_{i_1} + x_{f_1})/2$. There is a unique intersection point in $[x_{i_1}, x_{f_1}]$. Since the

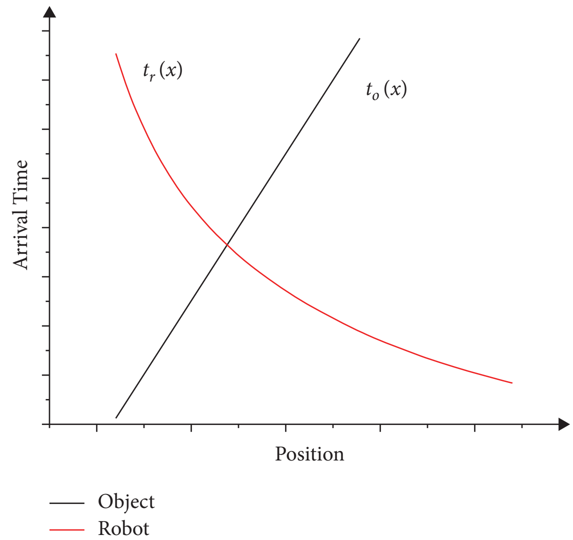



Figure 4: Robot and object position-time curves. The $x$-axis represents the position on the conveyor belt, and the $y$-axis represents the arrival time of robot and object. The position-time function curve of the robot $t_r(x)$ is represented by a red line and the position-time function curve of object $t_o(x)$ is represented by a black line.

robot's velocity is faster than the velocity of the conveyor belt, the robot reaches the midpoint $x_{m_1}$ of $[x_{i_1},x_{f_1}]$ before the target object, and when the time difference between the robot and the target object reaches a certain point is less than or equal to the specified threshold, we identify this point as the best grasping point $x^*$, as is shown in Figure 5. The operation steps of the search algorithm are as follows. (Algorithm 1)

In the process of searching intersections, the accuracy of the discriminant, $|t_o(x_{m_k}) - t_r(x_{m_k})| < \delta$, affects the search process of the algorithm. If the value of $\delta$ is too small, then a more accurate intersection can be obtained, but at the same time, the search time is increased. If $\delta$ is too large, the accuracy of the best grasping point obtained is reduced.

*5.2. The Improved Time Optimal Search Algorithm.* The improved time optimal search algorithm introduces an approximate robot position-time curve $\widetilde{t}_{r_k}(x)$ of the robot, as is shown in Figure 6, which is a straight line connecting the point $t_r(x_{i_k})$ and $t_r(x_{m_k})$. The expression of $\widetilde{t}_{r_k}(x)$ is shown in equation (11).

$$\widetilde{t}_r(x) = \frac{t_r(x_{i_k}) - t_r(x_{m_k})}{x_{i_k} - x_{m_k}} x + \frac{x_{i_k} \cdot t_r(x_{m_k}) - x_{m_k} \cdot t_r(x_{i_k})}{x_{i_k} - x_{m_k}} \text{if} \quad x \in [x_{i_k}, x_{m_k}]. \tag{11}$$

The intersection point of approximate position-time curve $\widetilde{t}_{r_k}(x)$ and object position-time curve $t_0(x)$ is denoted as $\widetilde{x}_k^*$, and it is assumed to be the approximate solution of the $k_{\text{th}}$ iteration, so it is expressed as follows:

$$\widetilde{x}_k^* = \frac{c_2(x_{i_k} - x_{m_k}) - (x_{i_k} \cdot t_r(x_{m_k}) - x_{m_k} \cdot t_r(x_{i_k}))}{(t_r(x_{ik}) - t_r(x_{mk})) - c_1(x_{ik} - x_{mk})} \text{if } \widetilde{x}_k^* \in [x_{i_k}, x_{m_k}]. \tag{12}$$

Hence, the improved algorithm is as follows. (Algorithm 2)

Similarly, the algorithm terminates when the difference of the robot arrival time and object arrival time is less than $\delta$, a predefined value. Threshold $\delta$ will affect the search time of the search algorithm.

## 6. Simulation

To verify the performance of the above-mentioned algorithm, a simulation experiment was conducted. In the experiment, a 6-DOF robot was utilized (The model of the robot is TA6-R10. The parameters of the robot are as shown in Table 1). Since the first three axes of the 6-DOF robot

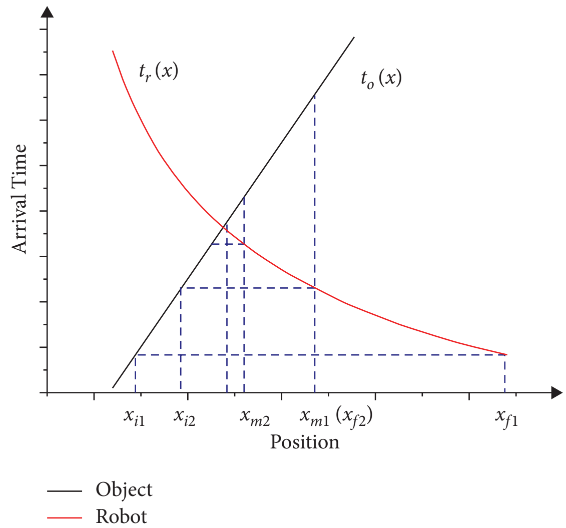



Figure 5: The pyramid-shaped time optimal search algorithm.

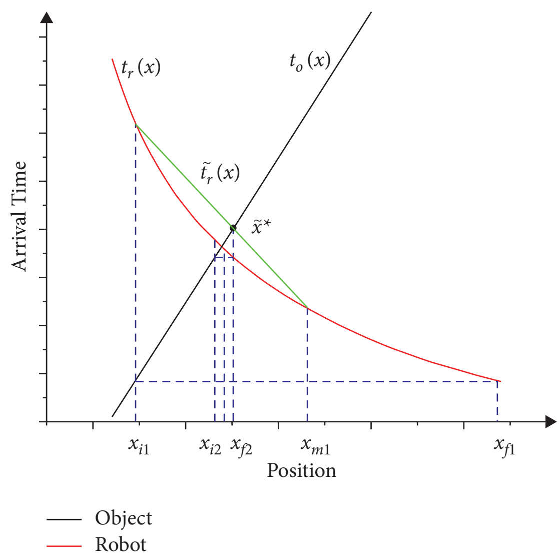



Figure 6: The improved time optimal search algorithm.

mainly affect the arrival time during which the end effector of the 6-DOF robot reaches the target point, and the last three axes mainly affect the pose of end effector and does not affect the arrival time, in the experiment, only the first three axes of the 6-DOF robot were considered. In the simulation, parameters of the 6-DOF robotic manipulator were set. The maximum angular velocity of the robotic joint is 150 degrees/s, and the velocity of the conveyor belt was set to 200 mm/s, which moved along the $x$-axis direction. The standby position of the end effector is [1000, 0, 500].

When the 6-DOF robotic end effector is ready for operation, the target object can be at any positions in the grasping area. The starting and final positions of the target object's position-time curve will change, but the gradient will not change. As shown in Figures 7 and 8, the position-time curves under different scenarios are described.

As it takes some time for the robotic manipulator to arrive the grasping interval, the target object may still be captured in the grasping interval although the target object's initial position is outside the grasping interval such as position 1. When the initial position of the target object is close to the final edge of the grasping interval, the robotic manipulator cannot complete the grasping of the target object which is inside the grasping interval, such as position 4. Therefore, we proposed a new concept: candidate interval. The candidate interval varies according to the speed difference between the conveyor belt and the robot.

To demonstrate the performance of the proposed improved search algorithm, we use the two algorithms to simulate several times, and the simulation results are compared. The simulation results of the pyramid search algorithm are shown in Table 2, and the simulation results of the improved time optimal search algorithm are shown in Table 3. Under different error thresholds, the operation time of the two algorithms is shown in Figure 9.

In step 1, set $k = 1$, and select $x_{f_1}$. The arrival time $t_1$ is obtained according to inverse kinematics, and $x_{i_1}$ is calculated via the equation, $t_{o_k} = c_1 \bullet x_{i_k} + c_2$ ($c_1, c_2 \in R$).
In step 2, if $t_o(x_{f_1}) - t_r(x_{f_1}) > 0$, the algorithm is stopped and the intersection point does not exist; if $t_o(x_{f_1}) - t_r(x_{f_1}) < 0$, go to step 3.
In step 3, calculate $x_{m_k} = (x_{i_k} + x_{f_k})/2$.
In step 4, if $|t_o(x_{m_k}) - t_r(x_{m_k})| < \delta$, set $x^* = x_{m_k}$, and stop the algorithm; otherwise skip to step 5.
In step 5, set $x_{f_{k+1}} = x_{m_k}$, and $x_{i_{k+1}}$ is calculated; then go to step 3.

Algorithm 1: The pyramid-shaped time optimal search algorithm.

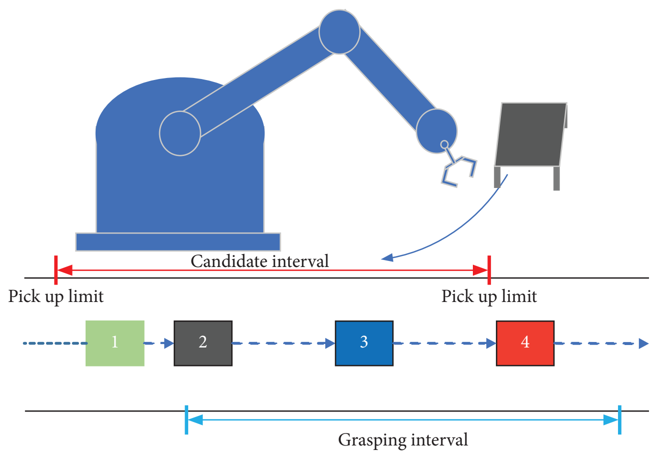



Figure 7: The initial positions of object.

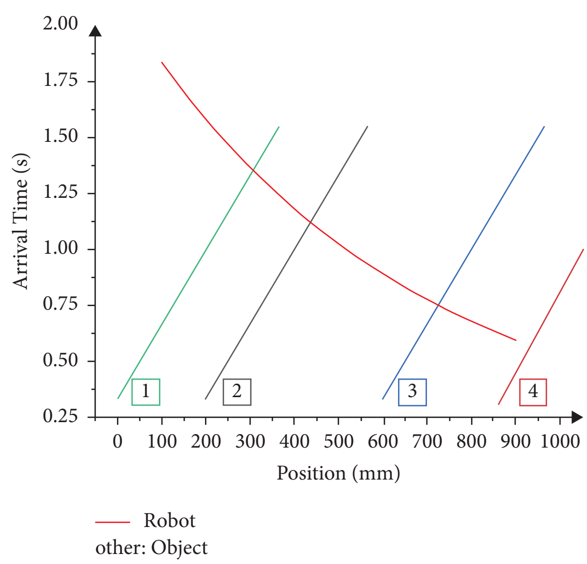



Figure 8: The arrival time curves of robot and objects.

By comparing the data in Tables 2 and 3, three inferences can be made.

(1) When the initial location of the target object is near the final edge of the grasping interval, the search algorithm can locate the pickup point faster

(2) When the error requirement is not strict or errors are high, the running time of pyramid optimization algorithm is shorter than the improved search algorithm

(3) The running time of pyramid optimization algorithm decreases with the increase of error, but the error values of the improved algorithm are always in the state of high precision, and the running time is relatively fixed.

Therefore, in order to study the advantages of the optimized algorithm, when the initial position of the target object (200 mm) and the speed of the conveyor belt (200 mm/s) are fixed, the two algorithms are respectively

In step 1, set $k = 1$, and an appropriate $x_{f_1}$ is selected. The arrival time $t_1$, is obtained according to inverse kinematics, and $x_{i_1}$ is calculated via the equation $t_{o_k} = c_1 \cdot x_{i_k} + c_2 \ (c_1, c_2 \in R)$.
In step 2, if $t_o(x_{f_1}) - t_r(x_{f_1}) > 0$, stop the algorithm and the intersection point does not exist; if $t_o(x_{f_1}) - t_r(x_{f_1}) < 0$, go to step 3.
In step 3, calculate $x_{m_k} = (x_{i_k} + x_{f_k})/2$ and calculate the intersection point $\widetilde{x}_k^*$ between the robot's approximate position-time curve $\widetilde{t}_{r_k}(x)$, and the target object's position-time curve $t_0(x)$.
In step 4, $|t_r(\widetilde{x}_k^*) - t_0(\widetilde{x}_k^*)| < \delta$, set $x^* = \widetilde{x}_k^*$ and stop the algorithm. If $|t_r(\widetilde{x}_k^*) - t_0(\widetilde{x}_k^*)| \geq \delta$, go to step 5.
In step 5, set $x_{f_{k+1}} = \widetilde{x}_k^*$, and calculate the position $x_{i_{k+1}}$ of the target object at the same time; go to step 3.

Algorithm 2: The improved time optimal search algorithm.

Table 1: Kinematic parameters of the robot.

| $i$ | $a_i$ | $\alpha_i$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0 | −90° | 130 | $\theta_1$ (0°) |
| 2 | 613 | 0 | 0 | $\theta_2$ (−90°) |
| 3 | 572 | 0 | 0 | $\theta_3$ (0°) |
| 4 | 0 | 90° | 163 | $\theta_4$ (90°) |
| 5 | 0 | −90° | 123 | $\theta_5$ (0°) |
| 6 | 0 | 0 | 103 | $\theta_6$ (0°) |

Table 2: Operation result of the pyramid-shaped search algorithm.

| Target initial position | $\delta$ (ms) | Operation time (ms) | Robot arrival time (ms) | Target arrival time (ms) | Error (ms) |
|---|---|---|---|---|---|
| | 5 | 504 | 761.3 | 763.3 | 2.0 |
| 200 | 10 | 440 | 760.7 | 765.9 | 5.2 |
| | 50 | 180 | 789.8 | 755.3 | 34.5 |
| | 5 | 422 | 526.6 | 631.3 | 4.7 |
| 500 | 10 | 422 | 526.6 | 531.3 | 4.7 |
| | 50 | 307 | 523.9 | 549.6 | 25.7 |
| | 5 | 306 | 371.2 | 373.9 | 2.7 |
| 800 | 10 | 251 | 370.9 | 376.9 | 6 |
| | 50 | 143 | 369.0 | 398.1 | 29.1 |

Table 3: Operation results of the improved time optimal search algorithm.

| Target initial position | $\delta$ (ms) | Operation time (ms) | Robot arrival time (ms) | Target arrival time (ms) | Error (ms) |
|---|---|---|---|---|---|
| | 5 | 232 | 761.0 | 764.8 | 3.8 |
| 200 | 10 | 234 | 761.0 | 764.8 | 3.8 |
| | 50 | 236 | 761.0 | 764.8 | 3.8 |
| | 5 | 227 | 527.1 | 527.7 | 0.6 |
| 500 | 10 | 226 | 527.1 | 527.7 | 0.6 |
| | 50 | 227 | 527.1 | 527.7 | 0.6 |
| | 5 | 227 | 371.4 | 371.2 | 0.2 |
| 800 | 10 | 225 | 371.4 | 371.2 | 0.2 |
| | 50 | 225 | 371.4 | 371.2 | 0.2 |

used to intercept and capture the target, and the errors after each iteration of the two search algorithms are recorded and compared.

As shown in Figure 10, although the improved time optimal search algorithm has a complex calculation formula and each iteration takes longer time, the improved algorithm has a larger search step distance and higher iteration efficiency during each iteration; the improved algorithm can complete the search for the grasping points in a shorter time within the high-precision error range, and is more suitable for the application in high-precision grasping tasks.

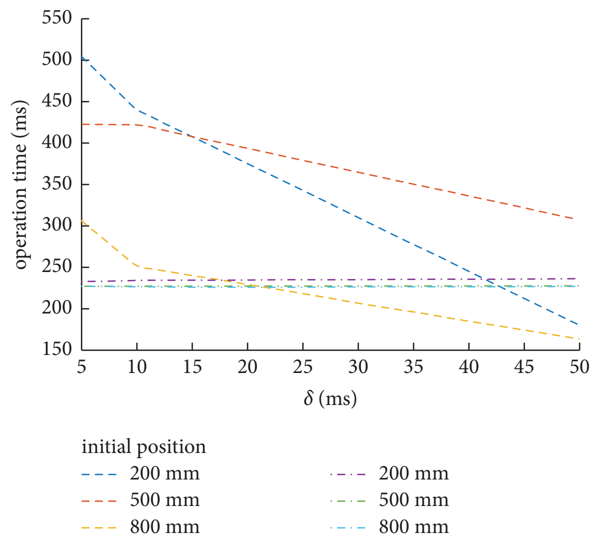



FIGURE 9: The operation time of the program under different error thresholds.

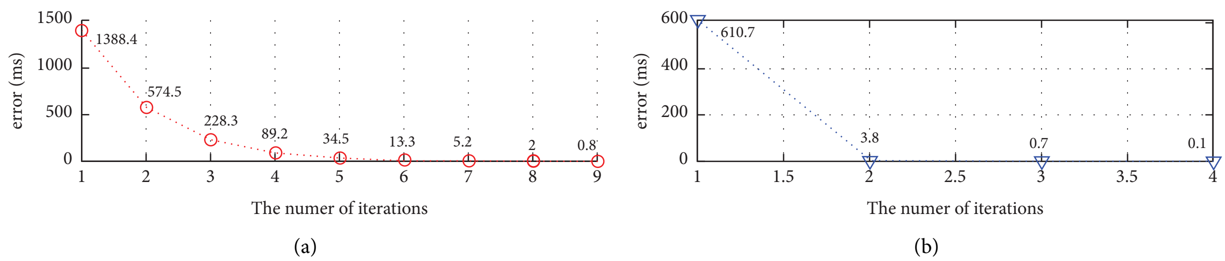



FIGURE 10: The number of iterations error. (a) The pyramid-shaped time optimal search algorithm. (b) The improved time optimal search algorithm.

## 7. Conclusions

To complete the grasping of the objects on the conveyor belt in the shortest possible time, a real-time search algorithm for the best grasping point is proposed. The robot's arrival function of time and the target object's arrival function of time were introduced. Simulation experiments were carried out on the pyramid-shaped search algorithm and the improved time optimal search algorithm; the optimal grasping point, which is the intersection where the robot's position-time curve and the object's position-time curve meet, was successfully found. The experiment results show that the improved search algorithm is more efficient.

The interception grasping method and intersection searching algorithm proposed in this paper can be applied to the tasks such as the classification of known parts or the sorting of defective parts. Compared with tracking grab, interception grasping method is more suitable for industrial automation production line. Due to the lack of tracking process of the object to be captured in the interception grasping method, there is a velocity difference between the robot's claw for capturing the object and the object. When the speed of the conveyor belt is too high, the capturing process may failure. In the actual application process, the time of the image processing and the time of the claw's closure should be fully considered.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Kai Zhang, Mingquan Yang, Xuwei Zhang, and Fei Liang contributed equally to this work.

## References


[1] N. Houshangi, "Control of a robotic manipulator to grasp a moving target using vision," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 604–609, Cincinnati, OH, USA, May 1990.

[2] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2267–2271, Sacramento, CA, USA, April 1991.

[3] M. G. Krishnan and A. Sankar, "Image space trajectory tracking of 6-DOF robot manipulator in assisting visual servoing," *Automatika*, vol. 63, no. 2, pp. 199–215, 2022.

[4] M. Keshmiri and W.-F. Xie, "Image-based visual servoing using an optimized trajectory planning technique," *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 359–370, 2017.
[5] D. Stogl and D. Zumkeller, "Tracking, Reconstruction and Grasping of Unknown Rotationally Symmetrical Objects from a Conveyor Belt," in *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, Limassol, Cyprus, September 2017.
[6] D. Hujic, G. Zak, E. Croft, R. G. Fenton, J. K. Mills, and B. Benhabib, "An active prediction, planning and execution system for interception of moving objects," in *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, pp. 347–352, Pittsburgh, PA, USA, August 1995.
[7] E. A. Croft, R. G. Fenton, and B. Benhabib, "Time-optimal interception of objects moving along predictable paths," in *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, pp. 419–425, Pittsburgh, PA, USA, August 1995.
[8] D. G. Fernandes and P. U. Lima, "A testbed for robotic visual servoing and catching of moving objects," in *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems. Surfing the Waves of Science and Technology*, pp. 475–478, Lisboa, Portugal, September 1998.
[9] M. Keshmiri, M. Keshmiri, and A. Mohebbi, "Augmented online point to point trajectory planning, a new approach in catching a moving object by a manipulator," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA)*, Xiamen, China, June 2010.
[10] C. C. Wong and M. Y. Chien, "Moving Object Prediction and Grasping System of Robot Manipulator," *IEEE ACCESS*, vol. 10, pp. 20159–20172, 2022.
[11] F. Islam, O. Salzman, A. Agarwal, and M. Likhachev, "Provably constant-time planning and replanning for real-time grasping objects off a conveyor belt," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1370–1384, 2021.
[12] Th. Borangiu, F. D. Anton, and A. Dogar, "Visual robot guidance in conveyor tracking with belt variables," in *Proceedings of the 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pp. 28–30, Cluj-Napoca, Romania, May 2010.
[13] X. Ye and S. Liu, "Velocity decomposition based planning algorithm for grasping moving object," in *Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 644–949, Enshi, China, May 2018.
[14] C.-D. Yang and C.-C. Yang, "A unified approach to proportional navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 557–567, 1997.
[15] M. Mehrandezh, N. M. Sela, R. G. Fenton, and B. Benhabib, "Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 30, no. 3, pp. 238–250, 2000.
[16] M. Mehrandezh, M. N. Sela, R. G. Fenton, and B. Benhabib, "Proportional navigation guidance in robot trajectory planning for intercepting moving objects," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 145–150, Detroit, MI, USA, May 1999.
[17] J. Su and Y. Xi, "Path planning for robotic hand/eye system to intercept moving objects," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2963–2968, Phoenix, AZ, USA, December 1999.
[18] P. J. Yuan and J. S. Chern, "Ideal proportional navigation," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, pp. 1161–1165, 1992.
[19] M. Keshmiri and W. F. Xie, "Catching Moving Objects Using a Navigation Guidance Technique in a Robotic Visual Servoing System," in *Proceedings of the American Control Conference*, pp. 6302–6307, Washington, DC, USA, June 2013.
[20] M. Keshmiri, "Performance comparison of various navigation guidance methods in interception of a moving object by a serial manipulator considering its kinematic and dynamic limits," in *Proceedings of the 2010 15th International Conference on Methods and Models in Automation and Robotics*, pp. 212–217, Miedzyzdroje, Poland, August 2010.
[21] J. Kim and A. Elizabeth, "Online near time-optimal trajectory planning for industrial robots," *Robotics And Computer-Integrated Manufacturing*, vol. 58, pp. 158–171, 2019.
[22] E. A. Croft, R. G. Fenton, and B. Benhabib, "Optimal rendezvous-point selection for robotic interception of moving objects," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 2, pp. 192–204, 1998.
[23] I. S. Shin, S. Nam, R. G. Roberts, and S. B. Moon, "Minimum-time algorithm for intercepting an object on the conveyor belt by robot," in *Proceedings of the 2007 International Symposium on Computational Intelligence in Robotics and Automation*, pp. 362–367, Jacksonville, FL, USA, June 2007.
[24] J. Xu, S. S. Akinola and P. K. Allen, "Dynamic grasping with reachability and motion awareness," in *Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9422–9429, Prague, Czech, October 2021.
[25] Z. Chaoyang, Z. Huixing, C. Rongmin, and W. Xiaoyan, "Research on dynamic conveyor grasping with pyramid search algorithm," *Manufacturing Automation*, vol. 37, no. 9, pp. 92–95, 2015.