

Simultaneous path planning and trajectory optimization for high-speed sorting system

Haojian Zhang^{1,2} , Tingting Su^{1,2}, Shaohong Wu¹ ,
Jun Zheng¹ and Yunkuan Wang¹

Abstract

This article presents a strategy for efficient sorting path planning and trajectory optimization with multiple constraints. The proposed strategy is applicable to typical high-density sorting applications. It plans the sorting path while optimizing each trajectory. The total stroke traveled by the end effector to complete the sorting task is shortened, and the efficiency of the production line is correspondingly facilitated. Thus, this article designs bow-shaped trajectories by analyzing the work process of the production line to ensure the smooth transition of the end effector and bow-shaped trajectories are based on the starting point and the end point of each sorting action. The modified trapezoidal acceleration and deceleration algorithm is used to optimize each sorting trajectory, which ensures that each sorting action is executed quickly and stably. Simultaneously, the greedy strategy is applied to plan the sorting task, which further shortens the total stroke of the end effector and improves the efficiency of the production line. A specific cost function is also designed to improve the planning strategy to prevent the occurrence of missing materials and enhance the adaptability of the sorting system. In consideration of the efficiency of the sorting system is significantly improved by this approach, the effectiveness of the proposed strategy in this article is thus verified compared with existing ones by experiments. In addition, the impact of the conveyor speed on the sorting system is also discussed.

Keywords

Path planning, trajectory optimization, real time, delta robot, simultaneous

Date received: 10 July 2017; accepted: 7 August 2018

Topic: Robot Manipulation and Control

Topic Editor: Andrey V Savkin

Associate Editor: Andrey V Savkin

Introduction

In today's industry, parallel robots have been continuously investigated because of the potential advantages such as higher stiffness, lighter structure and faster accuracy positioning with respect to serial robots, the capabilities of operating at high velocity and acceleration, and so on.^{1–3} These admirable features enable parallel robots to be successfully applied in many applications, such as assembly and disassembly processes, sorting and packaging, tooling and object handling, milling and motion simulations, among others.^{4,5} Especially in the field of sorting and

¹ Intelligent Manufacturing Technology and System Research Center, Institute of Automation Chinese Academy of Sciences, Beijing, China

² University of Chinese Academy of Sciences, Beijing, China

Corresponding author:

Shaohong Wu, Intelligent Manufacturing Technology and System Research Center, Institute of Automation Chinese Academy of Sciences, 95 Zhongguancun East Road, Beijing 100190, China.

Email: wushaohong2018@163.com



packaging, manual operations still largely exist in the sorting line with high-intensity repetitive labor. Such enterprises urgently need to carry out automated transformation so as to reduce labor cost and increase efficiency. Compared with the traditional sorting line, the robotic sorting system combined with machine vision can significantly increase the production efficiency and enhance the environmental adaptability. With the development of intelligent control technology, many intelligent control algorithms are gradually applied in the industrial field to fully explore the performance of the sorting system and further increase the production efficiency. Therefore, the robotic sorting system with machine vision and intelligent control algorithms has practical significance and broad application prospect.

The Delta robot is a three-degree-of-freedom space-translational parallel robot. The famous Delta manipulator had been presented by Clavel in 1988.⁶ As its active arms and slave arms can be made of lightweight material, it is usually developed for high-speed operation applied as a reliable system for performing light missions in the foods, electronics, and pharmaceutical. With the widespread applications, scholars have conducted extensive researches on its kinematics, dynamics, performance evaluation, control strategy, and synthesis optimization.^{7–10}

Kinematics is the primary researching task of the Delta robot, including positional relationship, velocity, and acceleration analysis.¹¹ The position relationship is the basis of motion analysis, and it is also the core of mechanism speed acceleration, work space, singularity, and so on.^{12–14}

In the applications of robots, path planning and trajectory optimization are two inseparable procedures, which have been actively investigated for decades with lots of related algorithms left today. The literatures written by Choset and LaValle provide detailed overview for available algorithms.^{15,16} In this article, we present a brief review of general path planning and trajectory optimization techniques and then focus on algorithms that are particularly relevant in the context discussed within this article.

Optimal path planning and trajectory optimization for parallel robot are important issues in robotics and industrial applications.¹⁷ The typical motion of the Delta robot is picking and placing operation (PPO) along a particular path.¹⁸ Most researchers focused on the kinematic performance or dynamics metrics for motion planning.^{19–22} However, most methods of the Delta robot movement planning mentioned above are based on the desired operation to carry out specific tasks. Although there are multiple metrics to guide the motion planning, such as the vibration of mechanical parts, preventing the collision and abrasion, avoiding the damage of mechanical structure, optimizing response time, minimizing the energy consumption,^{23–25} and so on, these metrics either focus on the structure of the Delta robot or focus on the dynamic performance, but they are not closely integrated with the application environment. As far as we know, few existing works study for path

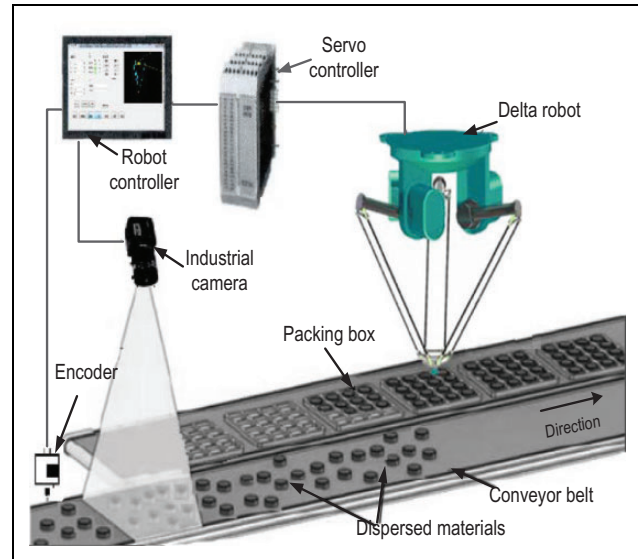


Figure 1. Structure diagram of the Delta robot sorting system.

planning and trajectory planning under the special consideration of practical production. Robots, as elements of the production process, are closely related to the production process. Thus, the study of robotic control based on specific scenario is particularly important.

This article demonstrates the implementation of online path planning with an improved greedy strategy and combines it with trajectory optimization to improve the efficiency of real-world sorting systems unresolved by existing methods. This article also contributes to solve some more specific technical problems including (1) an approach for online planning approximate optimal paths under specified conditions, (2) a method for adapting a pre-planned path for each PPO, and (3) a method for quickly generating a uniform sorting path under the consideration of specific time windows and other constraints.

The article is organized as follows. “Problem description” section shows the problem descriptions including the formulation of problem, details of the application scenario, and the equality and inequality constraints. The existing path planning strategies, the proposed path planning strategies, and trajectory optimization are described in “Path planning and trajectory optimization” section. To show the applicability and effectiveness of path planning for robotic manipulators, experiments with different planning strategies are illustrated in “Numerical examples” section. Finally, the last section gives the conclusions.

Problem description

In this article, the sorting system is mainly composed of five parts: robot controller, motion controller, robot body, conveyor belt, and image acquisition, as shown in Figure 1. The Delta robot is used as an important actuator for sorting systems, as shown in Figure 2.

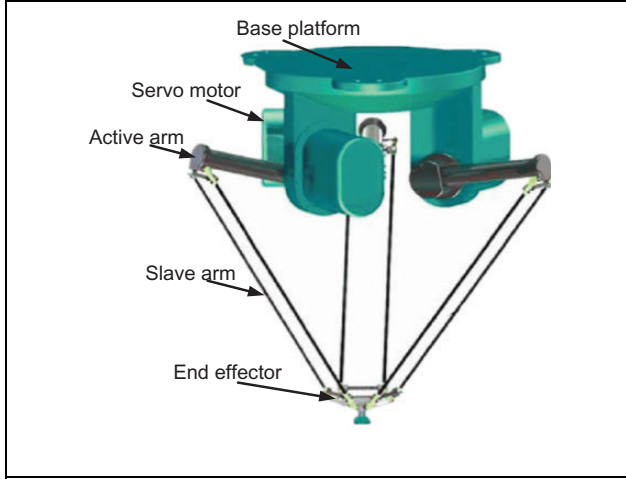


Figure 2. Illustration of Delta robot.

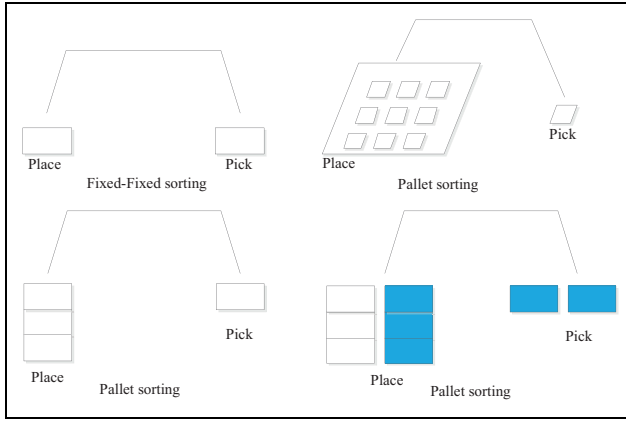


Figure 3. The existing sorting methods.

In general, different sorting systems vary with different sorting of materials, processes, and other factors. In the existing sorting and packaging applications, sorting strategies are generally simple but inflexible.²⁶ As shown in Figure 3, the sorting strategies adopted by most conventional sorting systems are fixed. Thus, the poor flexibility and inefficiency of these strategies severely limit their applications.²⁷ Also, the performance is mainly determined by the characteristics of the production line, even trajectory optimization is adopted, the performance still cannot be improved significantly.²⁸ Especially with dense distribution of high-speed sorting system, the path planning is increasingly important. The planar structure diagram of the sorting system is shown in Figure 4. Main task of the sorting system is to pick up the materials from the conveyor belt and neatly place them into the box. As the main implementing agency of the sorting system, the Delta robot completes high-intensity and fast repetitive operations to take place of human workers.

In order to achieve high-speed sorting by reducing the inertia, the active and slave arms of the Delta robot we have chosen are made of light material.² In addition, to further

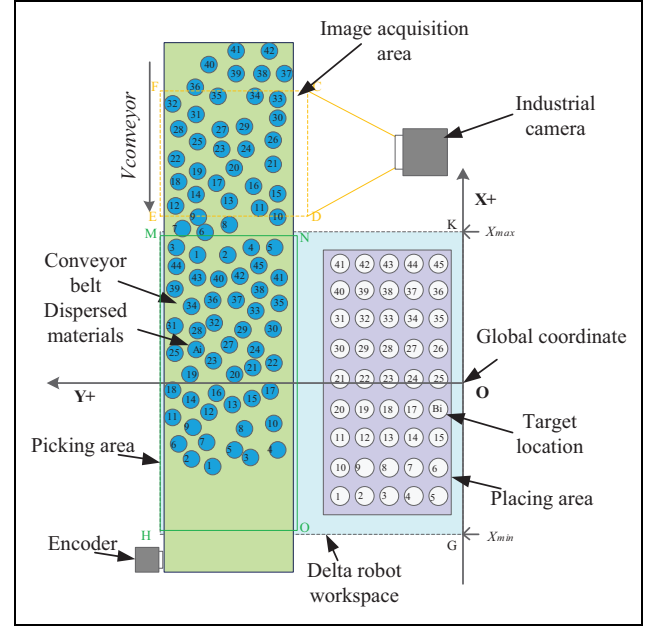


Figure 4. The planar model of the sorting system. The rectangular area CDEF is the recognition area, the area KGHM is the workspace, the area MNOH is the sort area, the area KGON is the area to be placed, $v_{conveyor}$ is the speed of the conveyor belt, the circle in MNOH identifies the material to be picked up, and the circles in KGON identify the location where the material is placed.

improve the efficiency and applicability of the sorting system, it is necessary to establish an effective location-filtering mechanism and an effective picking strategy for the materials sorted. Equipped with a machine vision, the sorting system will improve environmental adaptability and expand operational capacity compared with the existing sorting, as shown in Figure 3. Thus, it provides an efficient way for the sorting of high-density, randomly distributed materials, as shown in Figure 4.

On the contrary, it poses more challenges to the sorting strategy and the actual implementation of the robot control, like the real-time requirement of the sorting system. We know from the relevant literatures,^{18,28,29} the maximum sorting frequency of the parallel robot can reach 140 times per minute at present. Then, the execution time of each sorting operation is $60/140 = 0.4286$ s. That is, the path planning algorithm of each sorting must be performed less than 0.4286 s. To fully explore the performance of the automatic control system, this article introduces an intelligent control optimization algorithm to dynamically optimize the production process and to improve the system adaptability and operation efficiency.

As mentioned above, the working principles of the sorting system are illustrated in Figure 4. The exact location information of materials is a necessary prerequisite for the correct execution of the sorting operation. We adopt machine vision to obtain the accurate positions. The main procedure is illustrated in the following. When the sorting

system is running, robot controller outputs the periodic signal to trigger the industrial camera to take pictures according to the transmission distance information obtained by encoder. When the materials enter the visual inspection area of the industrial camera, as the conveyor belt moves forward, the pictures taken in the cycle contain the instantaneous position information of the target materials. We record the corresponding picture frame number and the encoder value, which facilitate the subsequent information processing of the visual system. The reference coordinates of the materials taken in pictures are first calibrated by visual identification, coordinate transformation, and other operations, and then sent to the host computer by Ethernet. Once the materials are transferred to the workspace, the robot controller will perform PPOs based on established models and optimization strategies. The actual path planning and trajectory optimization will be introduced in the next section.

Path planning and trajectory optimization

The relatively simple sorting system as mentioned in Figure 3 cannot satisfy the requirements of such a complex scene, as shown in Figure 4. The analysis of the work process shows that this problem looks like the traveling salesman problem (TSP), if materials and placements are used as the two city groups, A and B . In fact, there is an obvious difference between the two scenarios. Firstly, In traditional TSP problem, the location of the city remains the same, unlike this scene, in which the location of the materials city group A is changed in real time. That is, the maps of general planning algorithms for TSP are invariant. Secondly, in traditional TSP problem, the traveling salesman in the city group walks through the cities arbitrarily, but in this scene, the end effector, similar to the salesman, only walks between the two major city groups A and B back and forth. Therefore, many well-known algorithms for TSP are not suitable for this application scenario. Nevertheless, if we look at this issue from a different perspective, genetic algorithms can still be applied to this scenario by referring to the literature³⁰ and by passing the problem of map changing. At the same time, we must pay attention to the limitation of the genetic algorithm. The iterative optimization of genetic algorithms is a time-consuming process. In the optimization process, with different populations and number of iterations, the degree of optimization and execution time of the genetic algorithm are not the same, which greatly limits its application on this issue. We will conduct simulation verification in “Numerical examples” section. Next, we firstly describe the existing path planning methods for the sorting system.

The existing path planning method

To deal with this problem mentioned above, we set the sorting task of filling a box as a beat; obviously, the sorting system can be seen as composed of countless beats. The

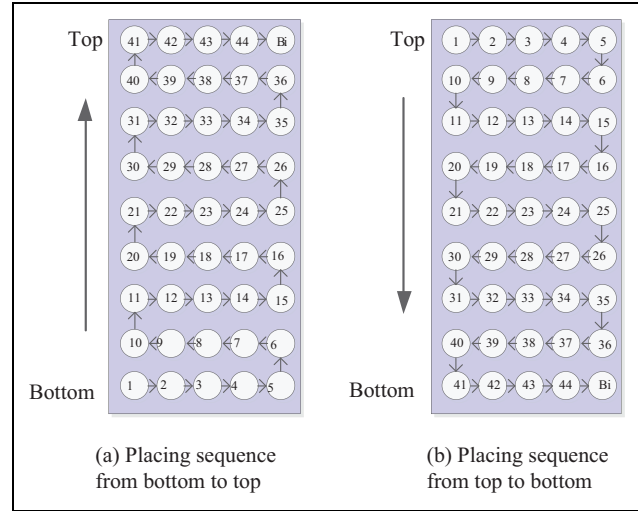


Figure 5. Scheme of two placing sequences. The sequence of the materials placed is numbered in ascending sequence, so the numbering mode can directly affect the path planning.

capacity of the box is a constant, which is assumed to be Num1. The neat arrangement of the box is numbered, and the numbering mode can be varied. The two schemes most commonly used are shown in Figure 5. The numbering mode determines the sequence in which the material is picked and placed. When the materials are delivered continuously from the conveyor belt, we also number them on the conveyor in ascending sequence from the x -coordinate. The location information of the materials is provided by the visual processing unit, but the specific details are not the subject matter herein and not described in detail herein. Although the positions of the materials on the conveyor belt change dynamically, the number corresponds to the placed positions in each beat, which facilitates subsequent operations. In field operation, PPOs are usually in a fixed sequence. In other words, the material of number A_i is placed in position B_i with the same number, as shown in Figure 4, and the sequence of each beat is

$$O \rightarrow A_1 \rightarrow B_1 \rightarrow A_2 \rightarrow B_2 \rightarrow \dots \rightarrow A_i \rightarrow B_i \\ \rightarrow \dots \rightarrow A_{\text{Num1}} \rightarrow B_{\text{Num1}}$$

where O is the starting position of the end effector. A_i is the material numbered i , and its coordinate is $A_i(x, y)$. B_i is the position numbered i in the box with coordinate $B_i(x, y)$.

Obviously, the difference between the two placement patterns will cause the difference in sorting efficiency, which can be proved from the subsequent test analysis. The materials move forward with the conveyor. They are considered missed when they are out of the workspace of the robot. Therefore, to avoid the risk and to ensure the stability of the sorting system, the existing picking strategies always give priority for materials at the forefront of the conveyor belt. As can be seen from the coordinate system in Figure 4, they maintain the priority for materials with the smallest x -coordinate values. The picking or placing trajectory of

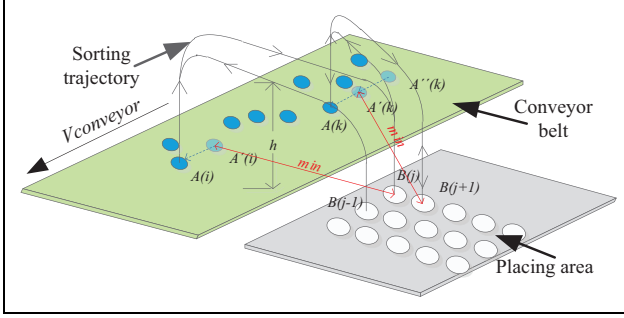


Figure 6. Sorting operation diagram. The system plans the sorting path of the next material and performs the trajectory optimization while executing the sorting operation.

each sorting is a typical bow-shaped trajectory with the height h , as shown in Figure 6; the total stroke of the end effector in a beat is

$$\begin{aligned} S &= (S_{01} + S_{11}) + (S_{12} + S_{22}) \\ &\quad + (S_{23} + S_{33}) + \dots + (S_{(i-1)i} + S_{ii}) + \dots \\ &\quad + (S_{(\text{Num1}-1)\text{Num1}} + S_{(\text{Num1})(\text{Num1})}) \\ &= \sum_{i=1}^{\text{Num1}} (S_{(i-1)i} + S_{ii}) \end{aligned} \quad (1)$$

$$\begin{aligned} S_{(i-1)i} &= \sqrt{(B_{i-1}.x - A_i.x)^2 + (B_{i-1}.y - A_i.y)^2} \\ &\quad - 4R + 2 \times \frac{\pi R}{2} + 2h \end{aligned} \quad (2)$$

$$\begin{aligned} S_{ii} &= \sqrt{(B_i.x - A_i.x)^2 + (B_i.y - A_i.y)^2} \\ &\quad - 4R + 2 \times \frac{\pi R}{2} + 2h \end{aligned} \quad (3)$$

where $S_{(i-1)i}$ is the distance between the position where end effector picks up the material A_i and the position where material $A_{(i-1)}$ is placed. S_{ii} is the distance that the end effector places the material from A_i to position B_i .

The position of the material on the conveyor belt changes in real time; thus, $\sum_{i=1}^{\text{Num1}} (S_{(i-1)i} + S_{ii})$ is a function of the argument t . The above formula can be rewritten as

$$\begin{aligned} S(t) &= S_{01}(t) + S_{11}(t) + S_{12}(t) + S_{22}(t) \\ &\quad + S_{23}(t) + S_{33}(t) + \dots + S_{(i-1)i}(t) + S_{ii}(t) + \dots \\ &\quad + S_{(\text{Num1}-1)\text{Num1}}(t) + S_{(\text{Num1})(\text{Num1})}(t) \\ &= \sum_{i=1}^{\text{Num1}} (S_{(i-1)i}(t) + S_{ii}(t)) \end{aligned} \quad (4)$$

$$\begin{aligned} S_{(i-1)i}(t) &= \sqrt{(B_{(i-1)}.x - A_i.x(t))^2 + (B_{(i-1)}.y - A_i.y(t))^2} \\ &\quad - 4R + 2 \times \frac{\pi R}{2} + 2h \end{aligned} \quad (5)$$

$$\begin{aligned} S_{ii}(t) &= \sqrt{(B_i.x - A_i.x(t))^2 + (B_i.y - A_i.y(t))^2} \\ &\quad - 4R + 2 \times \frac{\pi R}{2} + 2h \end{aligned} \quad (6)$$

The optimization of the bow-shaped trajectory will be covered in subsequent sections. The specific expression will be described in detail later.

The improved path planning method

In “The existing path planning method” section, the existing sorting path planning method is presented. The main disadvantage of that method is that it only considered preventing materials from being missed without extra performance optimization. Therefore, there should be a large room to improve the efficiency using path optimization for this scene. However, this point is largely overlooked by the past literature. In this section, we carefully analyze the sorting process and propose a dynamic path planning method based on an improved greedy strategy, which can optimize the sorting path in real time. Meanwhile, taking into account the risk of material being missed, we integrate the cost function to further improve the planning model and better ensure the continuous and stable operation of the sorting system. The implementation of this strategy is as follows. The sequence of placement is consistent with the previous section. Then, the optimization focuses on the sequence of picking up materials. By optimizing the sorting sequence, the total stroke of the end effector can be shortened and the efficiency of the sorting system can be improved. We can get the basic form of sorting sequence

$$\begin{aligned} O &\rightarrow A_i \rightarrow B_1 \rightarrow A_j \rightarrow B_2 \rightarrow \dots \rightarrow A_m \rightarrow B_i \\ &\rightarrow \dots \rightarrow A_{\text{Num1}} \rightarrow B_{\text{Num1}} \end{aligned}$$

It can be seen that the sequence of placement is ascending, which is consistent with the existing method mentioned in the previous section. But the sequence of picking is no longer fixed. It is derived from real-time path planning instead. It is assumed that the speed of the end effector is certain; then, the shorter the total stroke to complete the sorting task, the shorter the time spent and the higher the efficiency. The goal of the greedy algorithm is to shorten the stroke by optimizing the picking sequence. Then, in conjunction with the conclusion of the previous section, the objective function is the total stroke $S(t)$ of the sorting task of completing a beat, which is a function of the argument t and can be decomposed into the following equation

$$\min(S(t)) = \min \sum_{(i,j=1,2,3 \dots \text{Num1})} (S_{(i-1)i}(t) + S_{ji}(t)), \quad (7)$$

The above equation can be further decomposed into

$$\begin{aligned} \min \sum (S_{(i-1)j}(t) + S_{ji}(t)) = \min \sum & \\ \left(\sqrt{(B_{i-1}.x(t) - A_j.x(t))^2 + (B_{i-1}.y - A_j.y)^2} \right. & \\ \left. - (4R + 2 \times \frac{\pi R}{2} + 2h) \right) & \\ + \left(\sqrt{(B_i.x(t) - A_j.x(t))^2 + (B_i.y - A_j.y)^2} \right. & \\ \left. - (4R + 2 \times \frac{\pi R}{2} + 2h) \right) & \end{aligned} \quad (8)$$

where $S_{(i-1)j}(t)$ is the distance that the end effector picks up the material A_j after placing the material $A_{(i-1)}$. $S_{ji}(t)$ is the distance that the end effector places the material from A_j to position B_i .

The sorting task of each beat is broken down into multiple PPOs. If the current picking or placing operation has the shortest trajectory, the total stroke will be shortened.

There is a simple example of the picking and placing process. It is assumed that the second method is applied as the sequence in which the material is placed, as shown in Figure 5 (a). When the system starts, the end effector of the Delta robot is in the starting position O . The bow-shaped-type trajectory is designed for the sorting operation. As shown in Figure 6, the current system status works in a way that the end effector first places the material into the position $B_{(j-1)}$. The next step is to continue picking up the material on the conveyor belt to positions B_j and $B_{(j+1)}$, until this beat is completed. Therefore, it is necessary to find the next material to be picked up, plan the picking and placing paths, and optimize the trajectories before controlling the execution of the end effector. Specifically speaking, first we traverse all the distances between the position B_j and all the materials on the conveyor belt within this beat. Then, we need to find the nearest material A_i from the position B_j , as shown in Figure 6, and record its current location. It can be seen from the above analysis that the material A_i is moving forward with the conveyor belt during the execution of being picked by the end effector. Because the trajectory of the picking operation is not only always in a plane, but the form remains unified and the trajectory optimization is easy to implement. The ideal sorting operation is when the end effector and the selected material arrive at the same location at the same time. Therefore, it is necessary to calculate the exact coordinate of material A_i . If the end effector is started from the position $B_{(j-1)}$, the i th material now is in the coordinate A'_i ; the picking position is shown in Figure 6. We can compute the time spent as

$$t_{A_i} = \frac{A'_i.x - A_i.x}{v_{\text{conveyor}}} = \frac{A'_i.x(t) - A_i.x(t)}{v_{\text{conveyor}}} \quad (9)$$

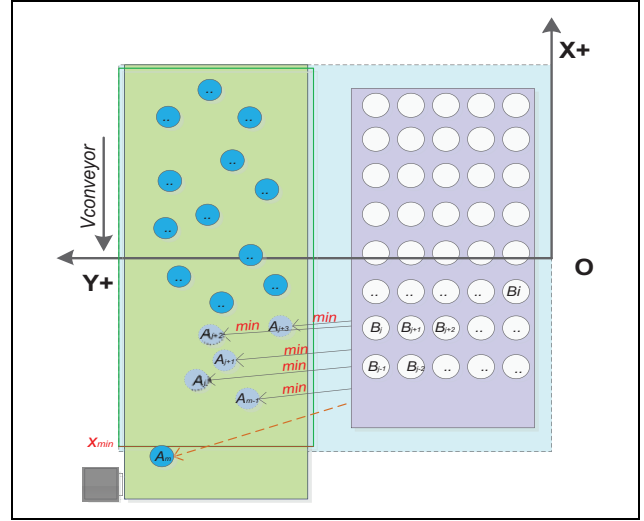


Figure 7. Schematic diagram of material being missed.

where v_{conveyor} is the transmission speed of the conveyor belt. t_{A_i} is equal to the running time of the picking operation.

A concrete expression will be given in the next section. By solving the equation, we get the specific coordinate A_i . Combining with the starting coordinate of the picking trajectory, we plan the picking path and provide an optimized trajectory. Since the PPOs are performed alternately, the ending point of the picking path is the starting point of the placing trajectory. If the location of the placement is known, we plan the path of the placing operation and optimize trajectory. Thus, when the process of picking up the A_i material is performed, the control system calculates the picking and placing paths and trajectories of the next material A_k in parallel, as shown in Figure 6. Consequently, one cycle of the sorting task is completed.

Moreover, since the novel path planning no longer requires picking up the material with smallest x -coordinate with priority, there will be a risk of missing materials. For example, as shown in Figure 7, when the system is running, although materials A_j , from $A_{(j+1)}$ to $A_{(j+3)}$, or even $A_{(m-1)}$ have been picked up to the box with shortest distance of placements, material A_m is not picked up in time. The worse thing is that it still moves forward with the conveyor belt. If it is not sorted before crossing the lower limit x_{\min} of the workspace of the Delta robot, the material will be missed, which is an issue that is avoided as much as possible in the work process.

In order to avoid this risk, we have to improve the greedy strategy and integrate the strategy with an evaluation function, which can be used to assess the risk of missing material and further optimize the path. The closer the distance between material and the lower limit x_{\min} of the workspace is, the greater the risk of being missed. The

probability of priority picking increases correspondingly. The evaluation function is expressed as

$$\lambda \times f(A_i.x(t)) \quad (10)$$

Strategy is integrated into the cost function $L_i(t)$

$$L_i(t) = (S_{(i-1)j}(t) + S_{ji}(t) - \lambda \times f(A_i.x(t))) \quad (11)$$

$$f(A_i.x(t)) = \begin{cases} 0, & (A_i.x(t) \geq 0) \\ \frac{1}{A_i.x(t) - x_{\min} + 0.2}, & (x_{\min} < A_i.x(t) < 0) \\ +\infty, & (A_i.x(t) \leq x_{\min}) \end{cases} \quad (12)$$

where λ is the coefficient. $A_i.x$ is the x -coordinate of the i th material on the conveyor belt, and x_{\max} and x_{\min} are the upper and lower limits of the working range of the robot, respectively.

The expression (11) improves the evaluation criteria of the principle greedy algorithm through a comprehensive assessment of the location of the material, which shortened the sorting of the total stroke, improved operating efficiency, and ensured no materials missed as much as possible.

As in the previous example, before the evaluation function is integrated, the material A_m has lower evaluation priority than other materials

$$L_{A_m}(t) > L_{A_n}(t) \quad (13)$$

$$(S_{B_{(i-1)A_m}}(t) + S_{A_mB_i}(t)) > (S_{B_{(i-1)A_n}}(t) + S_{A_nB_i}(t)) \quad (14)$$

$(m \neq n)$

The material A_m cannot be picked up until the last one is picked up or even missed. This will inevitably cause the sorting stroke to be extended and slow down the efficiency. The priority of sorting evaluation after integrating the cost function is improved as follows

$$L_{A_m}(t) < L_{A_n}(t) \quad (15)$$

$$\begin{aligned} (S_{B_{(i-1)A_m}}(t) + S_{A_mB_i}(t)) - \lambda f(A_m.x(t)) \\ < (S_{B_{(i-1)A_n}}(t) + S_{A_nB_i}(t)) - \lambda f(A_n.x(t)) \end{aligned} \quad (16)$$

$(m \neq n)$

Then, the material A_m was picked preferentially before it exceeded the workspace.

Trajectory optimization

Once the path planning strategy is determined, the next work is to consider the optimization of the picking and placing trajectory. The predecessors have done a lot of work in terms of trajectory optimization. However, there are shortcomings. For example, the trajectory optimization algorithms mentioned in a variety of literatures are time-consuming.^{31,32} Optimization may only be done offline or

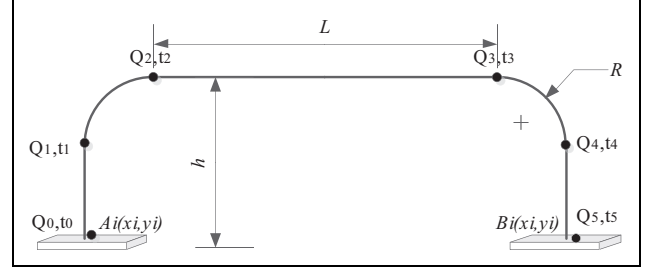


Figure 8. The bow-shaped trajectory of the sorting operation.

optimized for fixed paths like the four cases, as shown in Figure 3. In this article, we will show a real-time online optimization for each random picking and placing trajectory.

Due to the random distribution of the materials on the conveyor belt, the picking and placing trajectories are instantly generated with the changes of the i th material position. This poses new challenges to the trajectory optimization such as more constraints, limited optimization time, and so on. To deal with this issue, we combine a large number of previous works with the characteristics of the process. Then, we decide to use a series of bow-shaped trajectories derived from a basic one,³³ as shown in Figure 8. The advantages are as follows. Firstly, these trajectories have a uniform form which includes the same height, except that the horizontal width varies with the starting point and the ending point. Secondly, these trajectories are simple and easy to implement. The above advantages reduce the complexity and computational burden of trajectory optimization, which provides a guarantee for online optimization in real time.

This article chooses more flexible acceleration and deceleration processing method and computes the simpler modified trapezoidal acceleration and deceleration algorithm.³⁴ This is the acceleration and deceleration curve algorithm with the cosine function as the transition section. The expression for planning acceleration is

$$a(t) = \begin{cases} a_{\max} \sin\left(\frac{4\pi t}{T}\right), & \left(0 < t \leq \frac{T}{8}\right) \\ a_{\max}, & \left(\frac{T}{8} < t \leq \frac{3T}{8}\right) \\ a_{\max} \cos\left(\frac{4\pi}{T}\left(t - \frac{3T}{8}\right)\right), & \left(\frac{3T}{8} < t \leq \frac{5T}{8}\right) \\ -a_{\max}, & \left(\frac{5T}{8} < t \leq \frac{7T}{8}\right) \\ -a_{\max} \cos\left(\frac{4\pi}{8}\left(t - \frac{7T}{8}\right)\right), & \left(\frac{7T}{8} < t \leq T\right) \end{cases} \quad (17)$$

The acceleration curve of this mode is smooth, and jerk does not mutate. For the two integrations of equation (17), the displacement expression can be obtained

$$S_{ij}(t) = \begin{cases} \frac{Tt}{4\pi} a_{\max} - \left(\frac{T}{4\pi}\right)^2 a_{\max} \sin\left(\frac{4\pi t}{T}\right), & \left(0 < t \leq \frac{T}{8}\right) \\ \frac{1}{2} a_{\max} t^2 + \left(\frac{1}{4\pi} - \frac{1}{8}\right) a_{\max} Tt \\ + \left(\frac{1}{128} - \frac{1}{16\pi^2}\right) a_{\max} T^2, & \left(\frac{T}{8} < t \leq \frac{3T}{8}\right) \\ - \left(\frac{T}{4\pi}\right)^2 a_{\max} \cos\left(\frac{4\pi}{T}\left(t - \frac{3T}{8}\right)\right) \\ + \left(\frac{1}{4\pi} + \frac{1}{4}\right) a_{\max} Tt - \frac{1}{16} a_{\max} T^2, & \left(\frac{3T}{8} < t \leq \frac{5T}{8}\right) \\ - \frac{1}{2} a_{\max} t^2 + \left(\frac{1}{4\pi} + \frac{7}{8}\right) a_{\max} Tt \\ + \left(\frac{1}{16\pi^2} - \frac{33}{128}\right) a_{\max} T^2, & \left(\frac{5T}{8} < t \leq \frac{7T}{8}\right) \\ \left(\frac{T}{4\pi}\right)^2 a_{\max} \cos\left(\frac{4\pi}{T}\left(t - \frac{3T}{8}\right)\right) \\ + \frac{1}{4\pi} a_{\max} Tt + \frac{1}{8} a_{\max} T^2, & \left(\frac{7T}{8} < t \leq T\right) \end{cases} \quad (18)$$

The equation only shows the acceleration or deceleration phase, while the Delta robot is running at high speed during the sorting process. The acceleration or deceleration time is very short and the rest of the stages are running at the maximum speed v_{\max} . Through the analysis of the above, combined with the robot's performance parameters (maximum speed v_{\max} , maximum acceleration a_{\max}), we can get the robot running time T in the acceleration and deceleration phase. The expression of the time T is

$$T = \frac{2v_{\max}}{\sqrt{\left(\frac{1}{\pi} + \frac{1}{2}\right) a_{\max}}} \quad (19)$$

And, the total running time $T_{S_{ij}}$ is

$$T_{S_{ij}}(t) = \frac{2v_{\max}}{\sqrt{\left(\frac{1}{\pi} + \frac{1}{2}\right) a_{\max}}} + \frac{S_{S_1} - \frac{v_{\max}^2}{\left(\frac{1}{\pi} + \frac{1}{2}\right) a_{\max}}}{v_{\max}} \quad (20)$$

The pseudocode of the strategy

From the previous work process, we can draw the whole strategy. The sorting task consists of countless beats; in the process of completing each beat task, the picked materials are placed in ascending sequence of $B_j (j = 1, 2, 3 \dots \text{Num1})$. While the Delta robot sorts the material from A_i to B_j , the proposed algorithm plans the next trajectory of sorting material from A_m to B_{j+1} . When the Delta robot sorts the next material from A_m to B_{j+1} , the proposed algorithm continue to plan the next trajectory of sorting material from A_n to B_{j+2} and so forth. The pseudocode of the entire algorithm proposed is shown in algorithm 1.

Algorithm 1. Improved greedy-based motion planning.

Input : the goal positions $B_i(x, y)$;
the materials $A_i(x, y)$;
the speed of the conveyor belt v_{conveyor} ;
the configuration space of the Delta robot

Output : Optimized trajectory ($B \leftrightarrow A$)

```

1: InitSys();
2: Repeat Beat
3:   NumberGoalB();
4:   NumberBeatA();
5:   For  $i = 1, 2 \sim \text{Num1}$  do
6:     While sorting  $A_j \rightarrow B_i$  do
7:       For  $j = 1, 2 \sim \text{Num1}$  do
8:         Cost( $B_{i+1}, A_j$ ) ( $j = 1, 2 \sim \text{Num1}$ );
9:       End For
10:      Target( $A_m$ ) = Greedy( $B_{i+1}, A_j$ );
11:      OptimizedTrajectory( $B_{i+1}, A_m$ );
12:      InverseTrajectory( $B_{i+1}, A_m$ );
13:      OutputTrajectory( $B_{i+1} \leftrightarrow A_m$ );
14:    End While
15:  End For

```

Numerical examples

In this section, real-time path planning and trajectory optimization of high-speed sorting system of complex scene are implemented. All the tests have been performed with 1.7–2.4 GHz Intel Core 4 single-cores and 4 GB RAM. The detailed parameters of the working scene of the Delta robot are shown in Table 1. In order to demonstrate the effectiveness of the proposed planning strategy, we implement six different simulations using existing algorithms, genetic algorithms, and the algorithms presented in this article. Meanwhile, we adjust the genetic algorithm and the algorithm proposed in this article to achieve approximately the same efficiency and compare their real-time performance. The genetic algorithm with an initial population of 100 and a number of iterations of 300 is used after repeated adjustments. The six simulations are divided into two groups according to the placement sequence (from bottom to top and from top to bottom). The specific strategies are as follows.

Table 1. The parameters of the work scene.

Parameter	Value
Workspace	800×800 mm
x_{\min}	-400 mm
x_{\max}	400 mm
The width of the conveyor belt	350 mm
h —The height of trajectory	150 mm
The radius of plate	150 mm
The length of active arm	325 mm
The length of slave arm	800 mm
The radius of traveling plate	51 mm
The max velocity	5000 mm/s
The capacity of the box	60

The first strategy is the existing sorting strategy with a fixed picking sequence abbreviated as FS1. The fixed picking sequence is arranged according to the x -coordinate of the material from small to large, as shown in Figure 4, and the sequence of placement is arranged from bottom to top, as shown in Figure 5(a).

The second is the sorting strategy with the genetic algorithm abbreviated as GES1. The sequence of placement is the same as for FS1 and the picking sequence is optimized with the genetic algorithm.

The third is the sorting strategy with an improved greedy strategy abbreviated as GRS1. The sequence of placement is the same as for FS1 and the picking sequence is optimized with the improved greedy strategy, as shown in Figure 7.

The fourth is the existing sorting strategy with a fixed picking sequence abbreviated as FS2. The picking sequence is arranged according to the x -coordinate of the material from small to large, and the sequence of placement is arranged from top to bottom, as shown in Figure 5(b).

The fifth is the sorting strategy with the genetic algorithm abbreviated as GES2. The sequence of placement is the same as for FS2 and the picking sequence is optimized with the genetic algorithm.

The sixth is the sorting strategy with an improved greedy strategy abbreviated as GRS2. The sequence of placement is the same as for FS2 and the picking sequence is optimized with the improved greedy strategy, as shown in Figure 6.

From the previous analysis, we can see how the efficiency of the sorting system is reflected in the speed of the conveyor, while the Delta robot remains running at full speed. The faster the conveyor belt, the higher the sorting system efficiency, on the contrary, the lower the efficiency. Since one of our objectives is to increase the efficiency of production line, experiments and validations are carried out with different conveyor speeds. Moreover, the conveyor speed and the distribution of the materials on the conveyor belt are closely linked; the distribution density is also an important factor that can affect the conveyor speed, so the distribution density is an indispensable important

parameter of the sorting system. Fortunately, the distribution density of each sorting system is nearly constant and can be obtained by collecting data from the actual field. From the work process of the sorting system, we can see that the materials are continuously distributed randomly on the conveyor belt. We introduce the thought of the beat presented above to deal with the continuous material in batches. It is assumed that the number of materials for each beat is 60 and the distribution density of the materials is 300 per meter. Then, we test the six strategies mentioned above for 10 beats. Limited to the length of the article, the detailed data for each experiment are not completely listed. However, the detailed data for completing the 10-beat sorting task with the conveyor speed $v_{\text{conveyor}} = 7$ mm/s are given separately, as shown in Table 2. The mean time consumed to complete the sorting task of one beat and the distance traveled by the end effector are also given in Table 3.

Compared to the results in Table 2, we can conclude that using two different existing strategies, there is no significant difference in the running time to complete a beat with $v_{\text{conveyor}} = 7$ mm/s. The mean times are 28.03 and 28.10 s, respectively. Obviously, we can also learn from Table 3 that the moving distances of the conveyor belt are 196.21 and 196.7 mm. The effect of the corresponding optimization strategy is obvious. The GRS1 strategy is only 5.18% more efficient than the FS1 strategy. The GRS2 strategy increased efficiency by 10.32% over the FS2 strategy. Meanwhile, by comparing the standard deviations of various strategies, the standard deviation of GRS1 is less than that of FS1, and the standard deviation of GRS2 is less than that of FS2. Meanwhile, we can see that in the case of almost the same efficiency, the path planning algorithms proposed in this article have average execution times of $ST_{\text{GRS1}}/60 = 0.1026/60 = 0.00171$ and $ST_{\text{GRS2}}/60 = 0.0996/60 = 0.00136$ s when dynamically planning each step of the sorting path, as shown in Table 2. These execution times are much less than 0.4286 s and fully meet real-time requirements. The path planning algorithm based on the genetic algorithm is to plan the sorting path of an entire beat each time, and it cannot dynamically plan each sorting path. The average execution times of GES1 and GES2 for one beat are $ST_{\text{GES1}} = 21.26$ s and $ST_{\text{GES2}} = 21.58$ s. The disadvantages of the planning algorithm GES are obvious. The algorithm GES has a long execution time and real-time performance is poor, and the implementation of the planning algorithm is not flexible enough. So we conclude that the algorithm proposed in this article is superior to the genetic algorithm in dealing with this issue in many aspects.

In addition, if the sorting system is to be stable, the implementing agencies must coordinate with the application scenarios. This means that the Delta robot sorts the materials on the conveyor belt continually in accordance with the planned paths and trajectories. As the workspace of the Delta robot is

Table 2. Results of the sorting simulation using different algorithms at conveyor speed $v_{\text{conveyor}} = 7 \text{ mm/s}$.

Beat No.	BT _{FS1} (s)	BT _{GES1} (s)	P _{GES1} (%)	ST _{GES1} (s)	BT _{GRS1} (s)	P _{GRS1} (%)	ST _{GRS1} (s)	BT _{FS2} (s)	BT _{GES2} (s)	P _{GES2} (%)	ST _{GES2} (s)	BT _{GRS2} (s)	P _{GRS2} (%)	ST _{GRS2} (s)
1	27.48	25.12	8.60	20.03	26.74	2.69	0.0980	27.58	25.22	8.53	25.10	25.27	8.38	0.1037
2	28.92	25.35	12.34	22.20	27.00	6.65	0.1059	28.93	25.45	12.01	19.69	25.73	11.06	0.0918
3	28.58	25.66	10.21	19.72	26.40	7.64	0.0976	28.68	25.58	10.82	20.40	25.22	12.08	0.1062
4	27.81	25.43	8.57	20.28	26.55	4.53	0.1006	27.93	25.59	8.35	22.30	25.40	9.06	0.0991
5	28.71	25.28	11.93	23.26	26.54	7.56	0.1106	28.70	25.22	12.11	24.18	25.32	11.76	0.0919
6	26.99	25.04	7.21	23.45	26.48	1.90	0.1035	27.12	25.60	12.99	22.67	25.07	7.59	0.1005
7	28.83	25.08	13.01	21.71	26.68	7.44	0.1014	28.86	25.23	12.57	22.02	24.94	13.59	0.1080
8	28.46	25.24	11.32	19.72	26.34	7.45	0.1058	28.47	24.61	12.13	20.14	24.93	12.45	0.0983
9	27.06	25.91	7.95	22.13	26.34	2.69	0.1022	26.99	25.62	8.83	19.74	24.86	7.88	0.0996
10	27.50	25.41	7.59	20.18	26.62	3.22	0.1005	27.78	25.11	7.78	22.30	25.20	9.30	0.0973
Mean	28.03	25.25	9.87	21.26	26.57	5.18	0.1026	28.11	25.19	10.61	21.85	25.19	10.32	0.0996
SD	0.75	0.58			0.20			0.72	0.26			0.26		

BT_{FS1}: the time it takes to complete the sorting of one beat using existing strategies; BT_{GES1}: the time it takes to complete the sorting of one beat using the genetic algorithm; BT_{GRS1}: the time it takes to complete the sorting of one beat using the improved greedy algorithm; BT_{FS2}: the time it takes to complete the sorting of one beat using existing strategies; BT_{GES2}: the time it takes to complete the sorting of one beat using the genetic algorithm; BT_{GRS2}: the time it takes to complete the sorting of one beat using the improved greedy algorithm. P_{GES1} : an increase in the efficiency of using GES1 over using FS1; P_{GRS1} : an increase in the efficiency of using GRS1 over using FS1; P_{GES2} : an increase in the efficiency of using GES2 over using FS2; P_{GRS2} : an increase in the efficiency of using GRS2 over using FS2. ST_{GES1}: the time it takes to execute GES1; ST_{GRS1}: the time it takes to execute GRS1; ST_{GES2}: the time it takes to execute GES2; ST_{GRS2}: the time it takes to execute GRS2; SD: standard deviation.

Note: The bold values in Table 2 mainly highlight the comparison effect between the proposed algorithm and the genetic algorithm.

Table 3. Results of the sorting simulation using different algorithms at various conveyor speeds.

v_{conveyor}		6 mm/s	6.5 mm/s	7 mm/s	7.5 mm/s	8 mm/s	8.5 mm/s
FS1	Mean time of one beat (s)	27.97	28.00	28.03	28.08	28.11	28.16
	SD	0.75	0.75	0.75	0.74	0.75	0.75
	Distance of conveyor (mm)	167.80	182.00	196.24	210.57	224.91	239.34
GRS1	Mean time of one beat (s)	26.49	26.52	26.57	26.60	26.66	26.70
	SD	0.20	0.21	0.20	0.20	0.21	0.21
	Distance of conveyor (mm)	158.93	172.40	185.97	199.49	213.30	226.99
FS2	Mean time of one beat (s)	28.19	28.15	28.11	28.09	28.03	28.00
	SD	0.72	0.72	0.72	0.71	0.73	0.73
	Distance of conveyor (mm)	169.14	182.95	196.74	210.66	224.25	237.98
GRS2	Mean time of one beat (s)	25.28	25.24	25.19	25.15	25.12	25.09
	SD	0.26	0.26	0.26	0.26	0.26	0.27
	Distance of conveyor (mm)	151.66	164.03	176.35	188.63	200.94	213.25

SD: standard deviation.

Note: The bold values in Table 3 correspond to the analysis below. The main purpose is to find the matching speed of the belt when using different strategies.

limited, as shown in Figure 4, it must be ensured that the materials can be effectively delivered to the sorting area. Tables 2 and 3 show the data on the sorting tasks for the Delta robot completing one beat on average. The speed of the conveyor belt is set to $v_{\text{conveyor}} = 7 \text{ mm/s}$. When the sorting system adopts the FS1 strategy, the moving distance of the conveyor belt is 196.21 mm, and when the FS2 strategy is adopted, the moving distance is 196.7 mm. The two distances are relatively close to the length of 200 mm. This length is derived from the distribution density of the material mentioned above and is the length of the conveyor belt occupied by the materials within one beat. In other words, when the Delta robot finishes the sorting task for the previous beat, the materials of this beat enter the workspace and

coincide approximately with the distribution of the previous beat. It provides a guarantee for the uninterrupted sorting operation of the Delta robot. Therefore, for the sorting system adopting the two existing planning strategies, the conveyor speed $v_{\text{conveyor}} = 7 \text{ mm/s}$ is substantially matched with the current material density. Based on the above analysis and the data in Table 3, when the GRS1 strategy is applied, the Delta robot completes a beating sorting task with the conveyor speed $v_{\text{conveyor}} = 7.5 \text{ mm/s}$ and the conveyor belt moves a distance of 199.49 mm. Therefore, we conclude that the sorting system can run stably with the conveyor speed $v_{\text{conveyor}} = 7.5 \text{ mm/s}$. Similarly, when applying the GRS2 strategy, the sorting system can run stably with the conveyor speed $v_{\text{conveyor}} = 8.0 \text{ mm/s}$.

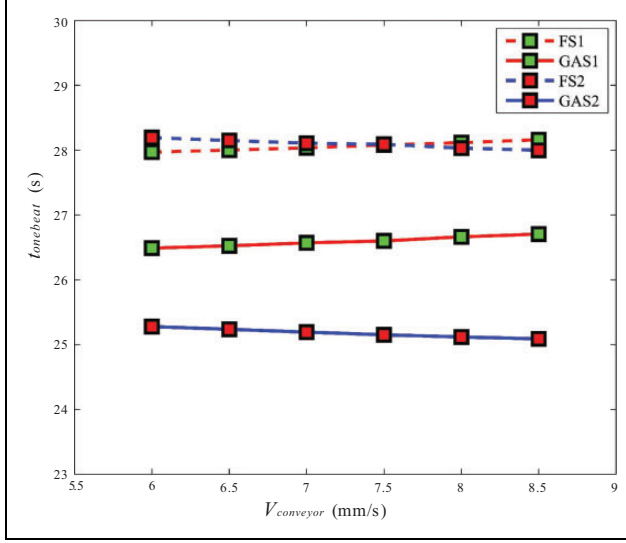


Figure 9. Result of path planning and trajectory optimization simulation with different conveyor speeds.

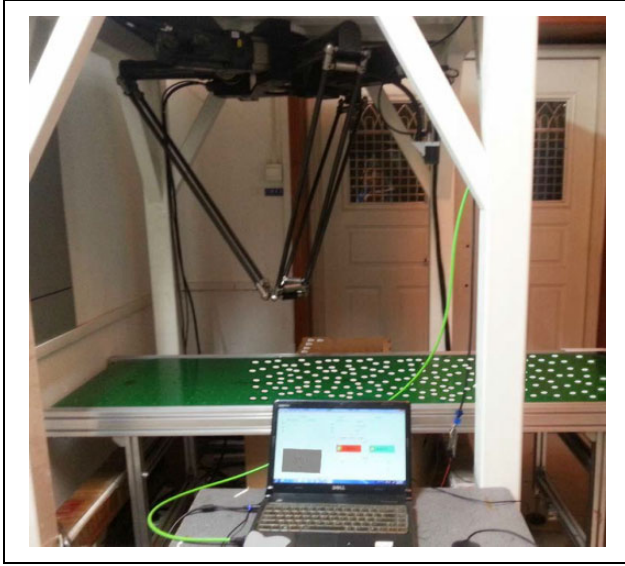


Figure 10. Scenario of continuous sorting experiments.

We plot Figure 9 for further illustration based on the data in Table 3. For the sorting system with the FS1 and GRS1 strategies, as the conveyor speed increases, the time it takes to complete the sorting task one beat increases. The distance traveled by the conveyor belt during this period also increases. That is, when the conveyor speed increases, within a certain period of time, the materials are delivered to the sorting area increases. Especially when the conveyor speed is greater than the matching speed, the Delta robot cannot complete the sorting task in time. From the analysis of the previous paragraph, the matching speeds of sorting systems adopting different strategies FS1 and GRS1 are approximately 7.0 and 7.5 mm/s, respectively. For example, if the two sorting systems run with the conveyor speed

$v_{conveyor} = 8.0$ mm/s, the area covered by each beat inevitably exceeds the area of the previous beat on the conveyor. Eventually, it will cause the materials to cover the entire sorting area and exceed the workspace. In the contrary, if the conveyor speed is less than the matching speed, the Delta robot takes less time to complete a beat sorting task and can finish the task faster. However, the efficiency of the sorting system is reduced. For example, if these two systems run with the conveyor speed $v_{conveyor} = 6.5$ mm/s, the area covered by the materials of each beat lags behind the area of the last beat on the conveyor. The materials delivered to the sorting area cannot meet the sorting demands of the Delta robot, and the Delta robot has to pause and waits for new materials to be delivered, so the efficiency of the sorting systems is reduced. For the sorting systems with the FS2 and GRS2 strategies, the faster the conveyor speed is, the shorter the times they takes for the Delta robot to finish a beat sorting task, and the longer the conveyor belt moves.

However, the speed of the change in the distance traveled by the conveyor belt is faster than that of the change in the time consumed for one beat. Similar to the above analysis, the sorting systems using the FS2 and GRS2 strategies have matching speeds of $v_{conveyor} = 7.0$ and $v_{conveyor} = 8.0$ mm/s, respectively. Therefore, from the matching speed, the performance of the sorting system with GRS1 is 7.14% higher than that of the sorting system with FS1, and the performance of the sorting system with GRS2 is 14.28% higher than that of the sorting system with FS2. This is consistent with the previous conclusion in general.

In order to fully demonstrate the laws of the sorting systems, to improve the cost function, and to further verify the effectiveness of the proposed algorithms, we conduct sorting experiments of continuous 50 beats with different conveyor speeds. The experimental scenario is shown in Figure 10 and Table 4 shows the times spent by different systems to complete a 50-beat sorting task. The slashes in the table indicate that the sorting system failed to complete the sorting task because of missing materials during the sorting process. We can see that when the sorting system adopts the FS1 strategy, the consumption time of sorting task is reduced with the increase of the conveyor speed, and when the conveyor speed is greater than 7.14 mm/s, the sorting system cannot complete the sorting task successfully. When the other strategies are adopted, the analysis process is similar to the sorting system with the FS1 strategy. From the previous conclusion, we know that the matching speed reflects the efficiency of the sorting system. We also note from Table 4 that the conveyor speed $v_{conveyor} = 7.14$ mm/s is the matching speed of the FS1 and FS2 strategies, while $v_{conveyor} = 7.41$ mm/s is the matching speed of the GRS1NC (GRS1 with no cost function) strategy, $v_{conveyor} = 7.5$ mm/s is the matching speed of the GRS1 strategy, $v_{conveyor} = 7.72$ mm/s is the matching speed of the GRS2NC (GRS2 with no cost function) strategy, and $v_{conveyor} = 8.0$ mm/s is the matching speed of the GRS2 strategy. Obviously, it is assumed that the sorting system can

Table 4. Results of continuous 50-beat sorting experiments using different algorithms at various conveyor speeds.

V_{conveyor}	6.5 mm/s	7 mm/s	7.14 mm/s	7.41 mm/s	7.5 mm/s	7.72 mm/s	8 mm/s	8.5 mm/s
$BT_{FS1} \times 50$ (s)	1494.3	1401.7	1402.2					
$BT_{GRSINC} \times 50$ (s)	1492.1	1389.7	1376.1	1331.3				
$BT_{GRS1} \times 50$ (s)	1491.3	1388.5	1372.3	1340.9	1330.5			
$BT_{FS2} \times 50$ (s)	1496.5	1405.4	1402.4					
$BT_{GRS2NC} \times 50$ (s)	1493.7	1391.1	1373.4	1337.5	1321.3	1297.5		
$BT_{GRS2} \times 50$ (s)	1492.5	1386.3	1367.9	1332.3	1295.4	1273.1	1254.7	

BT_{FS1} : the time it takes to complete the sorting of one beat using existing strategies; BT_{GRSINC} : the time it takes to complete the sorting of one beat using the greedy algorithm without cost function; BT_{GRS1} : the time it takes to complete the sorting of one beat using the improved greedy algorithm with cost function; BT_{FS2} : the time it takes to complete the sorting of one beat using existing strategies; BT_{GRS2NC} : the time it takes to complete the sorting of one beat using the greedy algorithm without cost function; BT_{GRS2} : the time it takes to complete the sorting of one beat using the improved greedy algorithm with cost function.

^aSlashes indicate that the materials are missed when the conveyor runs at current speed.

Note: The main purpose of bold values is to highlight the differences in the average sorting path when the system is running stably with different strategies.

only start the sorting of the next beat after finishing the sorting of the previous beat. The experiment results show that the efficiency of the sorting system using the improved algorithm that incorporates the cost function is higher than that of the sorting system only using the greedy strategy. In addition, the sorting system with the GRS2 strategy has achieved a 10.7% improvement in the efficiency compared to that with the FS2 strategy, and the sorting system with the GRS1 strategy has improved efficiency by about 5.1% over that with the FS1 strategy. The experimental results are basically consistent with the previous conclusions.

Conclusion

The applications for the work presented in this article are high-speed sorting system which employs the lightweight Delta robot to sort distributed materials intensively. The specific problem addressed can be able to provide approximate optimal paths with immediate response and provides optimized trajectories feasible for an existing picking and placing trajectory. The method utilizes the improved greedy algorithm, ensuring that the path optimization is more effective than the existing one. It gives the trajectory optimization to be performed in the next operation in advance while considering the risk of missing materials. Experiments show that applying the planner results in a significant reduction of running time required for executing the same tasks. Furthermore, the algorithm has been integrated into real application, where it has not only been used to plan paths between PPOs, but also provides real-time optimization of the trajectory. The algorithm is relatively simple to calculate, easy to program, and with small computing burden, which fully meets the real-time requirements of the sorting system. With high accuracy and stability, it is suitable for practical applications.


Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Major Science and Technology Project of Henan Province under grant number 161100210300.

ORCID iD

Haojian Zhang  <http://orcid.org/0000-0001-8447-0269>

Shaohong Wu  <http://orcid.org/0000-0001-9552-9000>

References

- Pierrot F, Nabat V, Company O, et al. Optimal design of a 4-DOF parallel manipulator: from academia to industry. *IEEE Trans Robot* 2009; 25(2): 213–224.
- Wu J, Li T, Wang J, et al. Stiffness and natural frequency of a 3-DOF parallel manipulator with consideration of additional leg candidates. *Robot Auto Syst* 2013; 61(8): 868–875.
- Gasparetto A, Alessandro A, Boscariol P, et al. Experimental validation of a dynamic model for lightweight robots. *Int J Adv Robot Syst* 2013; 10(3): 182–188.
- Gosselin C and Angeles J. The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator. *J Mech Trans Auto Des* 1988; 110(1): 35–41.
- Kosinska A, Galicki M, and Kedzior K. Design of parameters of parallel manipulators for a specified workspace. *Robotica* 2003; 21(5): 575–579.
- Clavel R. Delta, a fast robot with parallel geometry. In: *Proceeding of 18th international symposium on industrial robots*, 1988, pp. 91–100.
- Sira-Ramírez H, Núñez C, and Visairo N. Robust sigma-delta generalised proportional integral observer based control of a ‘buck’ converter with uncertain loads. *Int J Control* 2010; 83(8): 1631–1640.
- Zhao Y. Dynamic performance evaluation of a three translational degrees of freedom parallel robot. *Int J Robot Auto* 2012; 27(1): 31–40.
- Balmaceda-Santamaría A, Castillo-Castaneda E, and Gallardo-Alvarado J. A novel reconfiguration strategy of a

- Delta-type parallel manipulator. *Int J Adv Robot Syst* 2016; 13(1): 15–26.
10. Wu J, Wang L, and You Z. A new method for optimum design of parallel manipulator based on kinematics and dynamics. *Nonli Dynam* 2010; 61(4): 717–727.
11. Tho T and Thinh N. *Analysis of kinematics and dynamics of 4-DOF Delta parallel robot*. Switzerland: Springer International Publishing, 2014.
12. Stock M and Miller K. Optimal kinematic design of spatial parallel manipulators: application to linear delta robot. *J Mech Des* 2003; 125(2): 292–301.
13. Cha H, Woo J, Yi B, et al. Workspace analysis of the delta robot according to robot parameters and ball joints. In: *2013 10th International conference on ubiquitous robots and ambient intelligence (URAI)*, Jeju, South Korea, 30 October–2 November 2013, pp. 587–588. IEEE. DOI: 10.1109/URAI.2013.6677396.
14. Di Gregorio R. Determination of singularities in delta-like manipulators. *Int J Robot Res* 2004; 23(1): 89–96.
15. Choset H, Hutchinson S, Lynch K, et al. *Principles of robot motion: theory, algorithms, and implementation*. Cambridge, Massachusetts, USA: MIT press, 2005.
16. LaValle S. *Planning algorithms*. Cambridge, UK: Cambridge University Press, 2006.
17. Merlet J. *Parallel robots*, Vol. 128. New York, USA: Springer Science & Business Media, 2006.
18. Poppeova V, Uricek J, Bulej V, et al. Delta robots - robots for high speed manipulation. *Tehnicki Vjesnik* 2011; 18(3): 435–445.
19. Sahar G and Hollerbach J. Planning of minimum-time trajectories for robot arms. *Int J Rob Res* 1986; 5(3): 90–100.
20. Field G and Stepanenko Y. Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators. In: *IEEE international conference on robotics and automation, 1996. proceedings*, vol. 3. Minneapolis, MN, USA, 22–28 April 1996, pp. 2755–2760. IEEE. DOI: 10.1109/ROBOT.1996.506579.
21. Costantinescu D and Croft E. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *J Robot Syst* 2000; 17(5): 233–249.
22. Piazzzi A and Visioli A. Global minimum-jerk trajectory planning of robot manipulators. *IEEE Trans Ind Elect* 2000; 47(1): 140–149.
23. Ata A. Optimal trajectory planning of manipulators: a review. *J Eng Sci Technol* 2007; 2(1): 32–54.
24. Gasparetto A and Zanotto V. A new method for smooth trajectory planning of robot manipulators. *Mech Mach Theory* 2007; 42(4): 455–471.
25. Azizi M and Khani R. An algorithm for smooth trajectory planning optimization of isotropic translational parallel manipulators. *Proc Instit Mech Eng C J Mech Eng Sci* 2016; 230(12): 1987–2002.
26. Chen G, Zhai L, Li L, et al. Trajectory planning of delta robot for dynamic tracking, pick and placement. *Adv Mater Res* 2013; 680: 473–478.
27. Zhang Y, Huang R, Lou Y, et al. Dynamics based time-optimal smooth motion planning for the delta robot. In: *IEEE international conference on robotics and biomimetics*, Guangzhou, China, 11–14 December 2012, pp. 1789–1794. IEEE. DOI: 10.1109/ROBIO.2012.6491227.
28. Behzadipour S and Khajepour A. Time-optimal trajectory planning in cable-based manipulators. *IEEE Trans Robot* 2006; 22(3): 559–563.
29. Krebs D, Borchert G, and Raatz A. Simulation and design of an orientation mechanism for assembly systems. *Proc CIRP* 2016; 44(3): 245–250.
30. Tong J, Jiang H, and Zhou M. Optimization of seedling transplanting path based on genetic algorithm. *Trans Chin Soc Agric Mach* 2013; 4: 9.
31. Kelaiaia R, Company O, and Zaatri A. Multiobjective optimization of a linear delta parallel robot. *Mech Machine Theory* 2012; 50(2): 159–178.
32. Castañeda LA, Luviano-Juárez A, and Chairez I. Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control. *IEEE Trans Control Syst Technol* 2015; 23(4): 1387–1398.
33. Perumaal S and Jawahar N. Automated trajectory planner of industrial robot for pick and place task. *Int J Adv Robot Syst* 2013; 10(10): 1–17.
34. Siyyam HI and Syam MI. The modified trapezoidal rule for line integrals. *J Comput Appl Math* 1997; 84(1): 1–14.