Time-Optimal Pick-and-Throw S-Curve Trajectories for Fast Parallel Robots

Ghina Hassan, Marc Gouttefarde, Ahmed Chemori, Pierre-Elie Hervé, Maher El Rafei, Clovis Francis, Damien Sallé

Time-Optimal Pick-and-Throw S-Curve Trajectories for Fast Parallel Robots

Ghina Hassan, Marc Gouttefarde, *Member, IEEE,* Ahmed Chemori, *Senior Member, IEEE,* Pierre-Elie Hervé, Maher El Rafei, Clovis Francis and Damien Sallé

***Abstract*—In suitable robotic applications, throwing an object instead of placing it has the potential of improving the cycle time. In this context, a challenge is to generate time-optimal Pick-and-Throw (P&T) trajectories in order to further increase productivity. This paper introduces a methodology to determine a minimum-time throwing motion. This methodology consists essentially in determining an optimal release configuration (i.e. position and velocity) allowing an object to be thrown towards a desired target while minimizing the travel time of the throwing motion of the robot. To validate the potential of the proposed P&T approach, a comparison with the standard Pick-and-Place (P&P) process and an existing P&T method is made using the Delta-like parallel robot T3KR under different operating conditions. The obtained experimental results demonstrate the superiority and efficiency of the proposed P&T approach over the usual P&P and the existing P&T methods in terms of picking speed and cycle time.**

***Index Terms*—pick-and-throw, pick-and-place, minimum-time trajectory, parallel robots.**

I. Introduction

In industry, SCARA robots [1] and Delta-like parallel robots [2]–[5] are commonly used for Pick-and-Place (P&P) operations. Recently, robots have been used in selective waste sorting to complement traditional pneumatic NIR-sorting machines. Indeed, these machines have great detection capabilities but a not perfect selectivity. P&P robots are also used to remove the undesired material from the output flows to increase their purity. Most of the existing sorting robots have been developed on the basis of the Delta robot owing to its high dynamic capability. One example is the ABB's Delta robot used as the basis of a sorting robot [6]. It is equipped with artificial intelligence (AI) to identify recyclables. However, adopting P&P in such applications presents many limitations in terms of picking efficiency and workspace size. Hence, provided that the placing accuracy is not critical in waste sorting and that the objects can accept additional impact, throwing an object by means of a robotic system, instead of placing it, can make recycling more efficient.

Throwing is known by its ability to increase the capability of a robotic manipulator as well as the range of its workspace by throwing objects outside its maximum kinematic range. Thanks to this potential, throwing robots have been used in several applications. For instance, in the military field, a throwing robot has been used to displace goods [7], [8]. Furthermore, a throwing robot can be useful to gather information on a disaster site for search and rescue operations, such as in [9] where a a casting device is thrown by a robotic system. Fagiolini et al. [10] dealt with casting manipulation, which consists in throwing the robot end-effector to catch objects located at a relatively large distance from the robot's base. Recently, the throwing technique has been applied in waste industry. A dual arm throwing robot has been developed as a collaborative robot (CoBot) working alongside people to sort waste [11]. In [12], a P&T approach with a Delta robot is applied for fast waste sorting. Real-time experimental results prove the improved performance of the throwing procedure, compared to conventional P&P. Several other research works dealt with throwing [13]–[17]. In particular, Zeng et al. [18] investigated the challenge of accurately throwing arbitrary objects. They proposed a framework for jointly learning grasping and throwing policies from visual observations that enable TossingBot, a picking robot using a UR5 arm, to pick and throw arbitrary objects outside of its maximum range.

As mentioned above, throwing has the potential to speed up the displacement of objects and maximize productivity. In order to take full advantage of this benefit, a time-optimal throwing motion should be generated. This can be accomplished by first determining the appropriate and feasible geometric path and then optimizing the motion time along this path. This is of great importance for waste sorting since the robot can perform more picks per minute and thereby a large amount of waste can be processed. To the best of our knowledge, none of the existing research works address the time-minimization of a P&T trajectory through the optimization of the throwing parameters. In all the above mentioned works, one or two of the initial throwing parameters are fixed, while the others are determined according to the target position without any optimization. For instance, in [12], the trajectory is designed as a usual P&P trajectory with

Ghina Hassan, Marc Gouttefarde and Ahmed Chemori are with LIRMM, Univ Montpellier, CNRS, Montpellier, France (e-mail: ghina.hassan@lirmm.fr; marc.gouttefarde@lirmm.fr; ahmed.chemori@lirmm.fr).

Ghina Hassan, Maher El Rafei and Clovis Francis are with CRSI, Lebanese University, Faculty of Engineering, Beirut, Lebanon (e-mail: maher.elrafei@ul.edu.lb; cfrancis@ul.edu.lb).

Pierre-Elie Hervé and Damien Sallé are with TECNALIA, Basque Research and Technology Alliance (BRTA), San Sebastian, Spain (e-mail: pierre-elie.herve@tecnalia.com; damien.salle@tecnalia.com).

This work is supported by Tecnalia "Pick-and-Throw" research project.

The corresponding author acknowledged the Lebanese University and the social foundation AZM and SAADE for financial support.

the difference that the object is thrown at a given position on the horizontal path between the pick and target positions, resulting in a null initial release angle.

The main objective of this paper is to determine an appropriate geometric path for a P&T motion and to find along this path the throwing configuration yielding a minimum-time motion. The P&T motion consists mainly of an acceleration and a deceleration phase. For each phase, a third-order polynomial S-curve is adopted as a motion profile to obtain smooth, continuous and fast trajectories. The main contribution of this work is a method to determine an optimal release configuration (position and velocity) allowing an object to be thrown at a desired target while minimizing robot motion time. The constraints on the robot workspace, maximum speed, acceleration and jerk are taken into account. The corresponding optimization problem is formulated in two different ways. In the first one, the time between the pick position and the release position is considered as the objective function. The optimization variables are the release position, velocity and acceleration while constraints induced by the minimum-time S-curve and by ballistic motion target are taken into account. In the second method, it is shown that this optimization problem can be simplified as it boils down to minimizing the distance between the pick position and the release position with this distance and the release angle as the only variables. The second method requires less computational time, which is useful for real-time experiments where the optimal release configuration must be calculated online based on the actual pick and target positions. A comparison of the proposed P&T approach with standard P&P and with the P&T method proposed in [12] is conducted through real-time experimental scenarios with the parallel robot T3KR, under different operating conditions, to validate its effectiveness.

The rest of the paper is organized as follows. Section II is dedicated to the minimum-time S-curve trajectories. The main contribution of the paper is presented in Section III where the methodology to determine a minimum-time throwing motion is presented. The description of T3KR parallel robot and the real-time experimental results are provided in Section IV. Section V provides conclusions and future works.

II. Minimum-Time S-Curve Trajectory

A. Polynomial S-Curve Motion Profile

A smooth enough trajectory with a limited jerk is necessary to avoid end-effector residual vibrations and thereby improve trajectory tracking accuracy. S-curve trajectories, developed for the first time by Castain et al. [19], can meet these requirements by providing high-speed motions with minimum positioning time and minimum residual vibrations, e.g. [20]–[23]. The S-curve profiles proposed in the literature can be based on polynomial, trigonometric or sigmoid equations [24], [25]. In this work, a third order polynomial S-curve is considered since it provides a good trade-off between smoothness and motion time, and it has a moderate complexity.

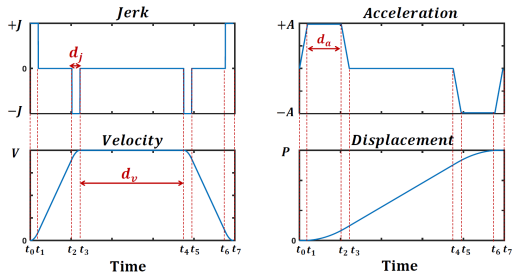


Fig. 1. Third order polynomial S-Curve model.

As illustrated in Fig. 1, the motion profile of the third order polynomial S-curve consists of seven segments, among which the first three and the last three constitute the acceleration and deceleration phases, respectively. The fourth segment constitutes the constant velocity phase. A symmetrical S-curve is considered in the present study, i.e. the acceleration and deceleration phases are symmetrical. The jerk along the S-curve trajectory is defined by the following function of time:

$$j(t) = \begin{cases} J, & t_0 \le t \le t_1, t_6 \le t \le t_7 \\ 0, & t_1 \le t \le t_2, t_3 \le t \le t_4, t_5 \le t \le t_6 \\ -J, & t_2 \le t \le t_3, t_4 \le t \le t_5 \end{cases} \quad (1)$$

where J is the jerk value. The time instants t_i are shown in Fig. 1. The time evolution of the acceleration, velocity and position can be deduced by successive integrations of (1).

Referring to Fig. 1, d_j is the time needed to raise the acceleration from zero to its maximum value A or to decrease the acceleration from its maximum value A to zero (i.e. the time during which the jerk remains constant at its maximum value J), d_a corresponds to the time during which the acceleration profile remains constant and equal to the maximum acceleration A, and d_v corresponds to the time during which the velocity profile remains constant. The above three time intervals can be expressed as follows:

$$d_j = \frac{A}{J}, \quad d_a = \frac{V}{A} - \frac{A}{J}, \quad d_v = \frac{P}{V} - \frac{V}{A} - \frac{A}{J} \quad (2)$$

where A and V are the velocity and acceleration achieved for a given displacement P. Note that d_j, d_a and d_v should all be greater than or equal to zero. The total time to travel a distance P is then expressed as follows:

$$T = 4d_j + 2d_a + d_v \quad (3)$$

B. Minimum-Time S-Curve Trajectory

Several methods have been proposed in the literature to generate time-optimal S-curve trajectories [26]–[28]. In this section, an original algorithm formulation to compute the maximum velocity V and acceleration A of a minimum-time 3^{rd} order S-curve trajectory is introduced. This formulation is equivalent to the one presented in [29]. However, it stems from a proof of the time optimality of this algorithm.

Algorithm 1 Minimum-Time S-Curve Trajectory

Input: P, J, V_{max}, A_{max}
Output: A and V yielding the minimum time

1: **if** $(J^2P \leq 2A_{max}^3)$ **then**
2: **if** $(JP^2 \leq 4V_{max}^3)$ **then**
3: $V = \sqrt[3]{\frac{JP^2}{4}}$ and $A = \sqrt[3]{\frac{J^2P}{2}}$
4: **else**
5: $V = V_{max}$ and $A = \sqrt{JV_{max}}$
6: **end if**
7: **else**
8: **if** $(\sqrt{JV_{max}} \leq A_{max})$ **then**
9: $V = V_{max}$ and $A = \sqrt{JV_{max}}$
10: **else**
11: $A = A_{max}$
12: $V = \frac{-A_{max}^2 + \sqrt{A_{max}^4 + 4J^2PA_{max}}}{2J}$
13: **if** $(V > V_{max})$ **then**
14: $V = V_{max}$
15: **end if**
16: **end if**
17: **end if**

Moreover, it will be used in Section II-C to gain further insight into the properties of the minimum-time 3^{rd} order S-curve trajectory.

Taking into account constraints on the maximum acceleration and velocity ($A \leq A_{max}$ and $V \leq V_{max}$), the goal is to minimize the total time T needed to travel the distance P. This can be achieved by determining the optimal acceleration A and the optimal velocity V being given J, P, A_{max} and V_{max}. By substituting the three time intervals (2) in the total displacement time (3), the function T to be minimized can be expressed as:

$$T = \frac{P}{V} + \frac{V}{A} + \frac{A}{J} \tag{4}$$

The optimization problem of finding the minimum-time S-curve trajectory is then formulated as minimizing T under the constraints $d_a \geq 0$, $d_v \geq 0$, $0 < V \leq V_{max}$ and $0 < A \leq A_{max}$. As proved in [30], using the KKT optimality conditions, the optimal solution of this problem can be computed by Algorithm 1 which determines the values of V and A yielding the minimum-time polynomial S-curve trajectory.

C. Time T and velocity V as functions of displacement P

With acceleration A and velocity V computed by Algorithm 1, this section points out that the total displacement time T given in (4) is a continuous increasing function of the displacement P and also that the maximum velocity V is a continuous nondecreasing function of P. These two properties of the minimum-time S-curve trajectory will be used in Section III to devise a method to generate minimum-time throw motions.

According to its expression in (4), T depends on P, V and A. Referring to Algorithm 1, either V and A are functions of P, e.g. at line 3, or else they are constant, i.e., independent of P (e.g. at line 5). When V and A are independent of P, from (4), T is directly seen to be an increasing function of P. On the contrary, when V or A is a function of P, it is not obvious from (4) that T is an increasing function of P. Indeed, V and A appear both at the numerator and denominator of one of the terms of the sum on the right-hand side of (4).

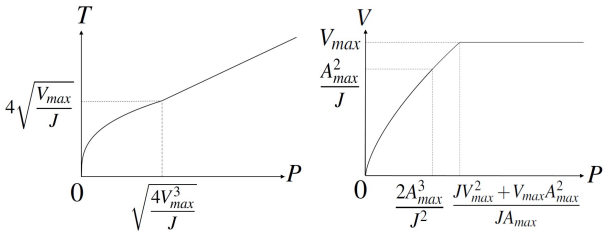


Fig. 2. A typical evolution of the increasing function $T(P)$ when $\sqrt{JV_{max}} < A_{max}$ (left) and the nondecreasing function $V(P)$ when $\sqrt{JV_{max}} \geq A_{max}$ (right).

In fact, when P is sufficiently small, V and A are the functions of P given at line 3 of Algorithm 1 and, when P increases, the conditions on P at lines 1, 2 and 13 will not be satisfied anymore for large values of P, so that V and A become constant. Let us look at the relationship between the conditions on P at lines 1 and 2 of Algorithm 1, namely

$$P \leq \frac{2A_{max}^3}{J^2} \quad \text{and} \quad P \leq \sqrt{\frac{4V_{max}^3}{J}} \tag{5}$$

respectively. In particular, when P increases from zero, line 5 will be executed if and only if

$$\sqrt{\frac{4V_{max}^3}{J}} < \frac{2A_{max}^3}{J^2}. \tag{6}$$

which, after some elementary calculations, can be shown to be equivalent to

$$\sqrt{JV_{max}} < A_{max}. \tag{7}$$

Hence, to analyze the dependency of T and V on P, two cases are distinguished: (7) satisfied and (7) not satisfied. In both cases, as detailed in [30], the function $T(P)$, obtained with Algorithm 1, can be proved to be a continuous increasing function over $0 \leq P \leq +\infty$ as illustrated in Fig. 2 for the case where (7) is satisfied.

Moreover, as also detailed in [30], the function $V(P)$, obtained with Algorithm 1, is a continuous nondecreasing function of P. Since they will be used in Section III-E, the different expressions of $V(P)$ are presented below.

1) Case 1 – $\sqrt{JV_{max}} < A_{max}$: The continuous nondecreasing function $V(P)$ is composed of two segments: Namely, for $0 \leq P \leq \sqrt{(4V_{max}^3)/J}$, $V(P) = \sqrt[3]{JP^2/4}$, and for $\sqrt{(4V_{max}^3)/J} < P \leq +\infty$, $V(P) = V_{max}$.

2) Case 2 – $\sqrt{JV_{max}} \geq A_{max}$: As illustrated in Fig. 2, the continuous nondecreasing function $V(P)$ is composed of three segments:

• For $0 \leq P \leq 2A_{max}^3/J^2$, $V(P) = \sqrt[3]{JP^2/4}$.

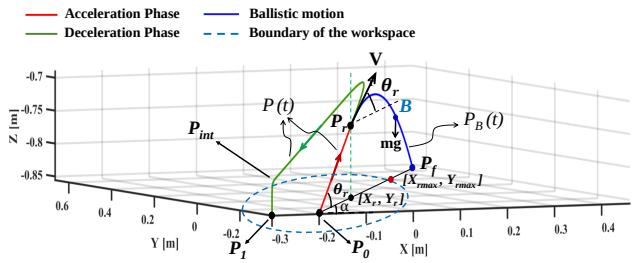


Fig. 3. Illustration of the pick and throw motion consisting of robot motion acceleration and deceleration phases together with the ballistic motion of the thrown object.

- For $2A_{max}^3/J^2 \leq P \leq \left(JV_{max}^2 + V_{max}A_{max}^2\right)/\left(JA_{max}\right)$, $V(P)$ is given by the expression at line 12 of Algorithm 1.
- For $\left(JV_{max}^2 + V_{max}A_{max}^2\right)/\left(JA_{max}\right) \leq P \leq +\infty$, $V(P) = V_{max}$.

III. Minimum-Time Throw Motion

A. Problem Formulation

A robot moving in 3D environment has to throw an object towards a desired target point P_f, located inside or outside of its workspace. The main objective is then to search for an optimal throwing configuration (i.e. position and velocity) in order to increase as much as possible the number of picks per minute. This configuration should thus allow to throw the object into the desired target while ensuring a minimum-time robot movement. Referring to Fig. 3, the throw motion of the robot end-effector reference point $P(t)$ and the ballistic motion $P_B(t)$ of the object B are defined as follows.

Throw motion: The robot trajectory consists of successive acceleration and deceleration phases. At $t = 0$, the manipulator is at the pick position with zero velocity, i.e. $P(0) = P_0$ with $\dot{P}(0) = V_0 = 0$. Once the robot picks the object B, it accelerates to the release position at time t_r. The release configuration is defined by the position P_r and velocity $\dot{P}(t_r) = V_r \neq 0$. After tossing B instantaneously, the robot has to decelerate back to the next pick position at time t_1 to pick another object at point P_1 with velocity $\dot{P}(t_1) = V_1 = 0$.

Ballistic motion: Once released, the object B follows a free-flight motion from the release point P_r with the velocity V_r to the desired target $P_B(t_f) = P_f$ reached at time t_f.

During the throw motion, the robot must satisfy two sets of constraints. The first set of constraints Σ includes the limits of the workspace as well as the maximum velocity, acceleration and jerk of the robot's end-effector. The second set of constraints Ω ensures that the ballistic trajectory interacts with the desired target position set Γ_{target}. This last one includes the target position point P_f and possibly a tolerance for reaching the target. The target position set can have different forms, if it is a hyper-rectangle, the tolerance can be expressed as follows: $P_{target}^{min} < P_B(t_f) < P_{target}^{max}$.

For given pick positions $\mathbf{P}_0$ and $\mathbf{P}_1$, and target position $\mathbf{P}_f$, the problem addressed in this paper is the determination of the release configuration corresponding to a minimum-time throw motion, this configuration being characterized by the release position $\mathbf{P}_r = [X_r, Y_r, Z_r]$ and the corresponding release velocity $\mathbf{V}_r = [V_{xr}, V_{yr}, V_{zr}]$. On the one hand, this configuration shall ensure that the thrown object will reach the desired target and, on the other hand, it should guarantee a minimum-time movement for the robot, while satisfying the sets of constraints Σ and Ω. The geometric path from $\mathbf{P}_0$ to $\mathbf{P}_r$ is a straight line lying in the vertical plane containing $\mathbf{P}_0$ and $\mathbf{P}_f$, θ_r denotes the angle between this straight line and the straight line $P_0 - P_f$. Hence, θ_r is an unknown to be determined along with the position of $\mathbf{P}_r$ along the straight line and the velocity $\mathbf{V}_r$. The description of the throw motion path will be detailed in Section III-C.

B. Recall of Mathematical Modeling of Ballistic Motion

Considering the ballistic motion of an arbitrary object B in 3D space, we assume that there is no obstacles and that air resistance is negligible so that the ballistic trajectory is only affected by gravity. Moreover, the object is considered as a pointwise mass. The ballistic motion of the object B is illustrated in Fig. 3. Applying Newton's second law to B gives simply $\mathbf{g} = \mathbf{a}$, where $\mathbf{g} \in \mathbb{R}^3$ is the gravity acceleration vector ($\mathbf{g} = [0, 0, -g]^T$, $g = 9.81\ m/s^2$, and $\mathbf{a} \in \mathbb{R}^3$ is the acceleration of B. By integration of the equation $\mathbf{g} = \mathbf{a}$, the trajectory of B along the x, y and z axes can be expressed as follows:

$$x_B(t) = V_{xr}t + X_r, \quad y_B(t) = V_{yr}t + Y_r \tag{8a}$$

$$z_B(t) = -\frac{1}{2}gt^2 + V_{zr}t + Z_r \tag{8b}$$

As shown in Fig. 3, $\mathbf{P}_f = [X_f, Y_f, Z_f]^T$ is the target position of B, reached at the final time t_f (i.e. $P_B(t_f) = P_f$). With $z_B(t_f) = Z_f$, t_f can be calculated from (8b) as follows:

$$t_f = \frac{V_{zr}}{g} + \sqrt{\left(\frac{V_{zr}}{g}\right)^2 + 2\frac{Z_r - Z_f}{g}} \tag{9}$$

C. Throw Motion Geometric Path and S-curve Motion Profile

The geometric path of the throw motion includes four points: The pick position of the object, $\mathbf{P}_0 = [X_0, Y_0, Z_0]$, the release point, $\mathbf{P}_r = [X_r, Y_r, Z_r]$, the pick position of the second object, $\mathbf{P}_1 = [X_1, Y_1, Z_1]$, and the intermediate point, $\mathbf{P}_{int} = [X_1, Y_1, Z_1 + Z_{off}]$. The overall throw motion can then be divided into three phases: (i) An acceleration phase along the straight line from $\mathbf{P}_0$ to $\mathbf{P}_r$, (ii) a deceleration phase from $\mathbf{P}_r$ to $\mathbf{P}_{int}$, and (iii) finally, a vertical movement from $\mathbf{P}_{int}$ to $\mathbf{P}_1$. Once the robot end-effector reaches the release position $\mathbf{P}_r$ with the release velocity V_r, it throws the object towards the target position and then moves forwards until the velocity becomes zero. After that, it starts decelerating towards $\mathbf{P}_{int}$ in a continuous motion. The parallel robot

T3KR, used in the experiments (cf. Section IV), cannot decelerate directly towards $\mathbf{P}_1$ because the end-effector must move down vertically in order to successfully grasp the object. The z coordinate of $\mathbf{P}_{int}$ is then chosen as $Z_1 + Z_{off}$, with $Z_{off} > 0$ being a small vertical offset.

The three phases of the robot throw motion are defined as point-to-point movements. Besides, the S-curve velocity profile described in Section II is adopted to generate each movement, where a minimum-time point-to-point movement is obtained by using Algorithm 1. As mentioned in Section II-A, the motion profile of a S-curve-based point-to-point movement consists of seven segments (cf. Fig. 1). For each segment of motion, the evolution of the variation of the motion profile with time, $S(t)$, is governed by the following equation:

$$S(t) = S_i + V_i(t-t_i) + \frac{1}{2}A_i(t-t_i)^2 + \frac{1}{6}J(t-t_i)^3 \qquad (10)$$

where J, A_i, V_i, S_i, t_i are the initial values of jerk, acceleration, velocity, position and time corresponding to each segment. The desired Cartesian position of the robot end-effector is then generated as follows:

$$X_d(t) = X_i + \frac{S(t)}{P}(X_f - X_i) \qquad (11)$$

where $X_d(t) = [x_d(t), y_d(t), z_d(t)]$ is the desired Cartesian trajectory, $X_i = [x_i, y_i, z_i]$ and $X_f = [x_f, y_f, z_f]$ are the initial and final positions of a point-to-point movement, respectively. $P = \|X_f - X_i\|$ is the distance to be traveled. By means of equations (10) and (11) the synchronization between axes is guaranteed.

Regarding the connection of two consecutive phases (i.e. corner blending), we exploit an overlap strategy (e.g. trajectory generation with via points [31]) to smooth the trajectory and eliminate discontinuities. In brief, each trajectory phase is designed as a point-to-point trajectory. The duration of the deceleration phase of the first trajectory is compared to the duration of the acceleration phase of the second trajectory, and a minimum overlap is determined. This ensures that the first trajectory will smoothly disappear during the second trajectory acceleration phase yielding a continuous switching between the two consecutive trajectories.

D. Determination of the Optimal Release Configuration

The optimal release configuration is defined as the one leading to a minimum-time throw motion of the robot. This motion consists of the three phases described in Section III-C. Since the third phase is a small vertical movement required to pick the next object, which is assumed to be located close to P_0, the optimal release configuration is the one that minimizes the time of the first two phases. By design of the deceleration phase, the smaller the duration of the acceleration phase, the smaller is the duration of the deceleration phase. Hence, the minimum-time throw motion is obtained by minimizing the time of the acceleration phase. The release configuration, including $\mathbf{P}_r$ and $\mathbf{V}_r$, allowing to minimize this time, while allowing the object to reach the target position $\mathbf{P}_f$, is the optimal one.

As mentioned above, the adopted motion profile along the straight line $P_0 - P_r$ is a 3^{rd} order polynomial S-curve. For the object to be thrown at a certain distance (to the target position $\mathbf{P}_f$), the release velocity $V_r = \|\mathbf{V}_r\|$ should be relatively high and, according to Section II, the velocity along the S-curve is maximal (equal to V) at the half of the traveling distance, i.e. at $P/2$. Hence, the traveled distance P of the S-curve is defined as being equal to twice P_0P_r, $P = 2\|\mathbf{P}_r - \mathbf{P}_0\|$. Accordingly, the time t_r required to travel the distance from $\mathbf{P}_0$ to $\mathbf{P}_r$ is then equal to the half of the time calculated in (4). In this way, the release velocity is equal to the S-curve maximum velocity, i.e. $V_r = V$. From (4), the time t_r can then be expressed as a function of V_r and of the coordinates of $\mathbf{P}_r$ as follows:

$$t_r = \frac{1}{2}\left(\frac{2\|\mathbf{P}_r - \mathbf{P}_0\|}{V_r} + \frac{V_r}{A} + \frac{A}{J}\right) \qquad (12)$$

The optimal release configuration $(\mathbf{P}_r, V_r)$ is then the one that minimizes the traveling time t_r in (12) while satisfying the two sets of constraints Σ and Ω defined in Section II.

Let us now define $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$, where $x_1 = V = V_r$, $x_2 = A$, $x_3 = X_r$, $x_4 = Y_r$ and $x_5 = Z_r$ as well as $x_{1max} = V_{max}$, $x_{2max} = A_{max}$, $x_{3max} = X_{r_{max}}$, $x_{4max} = Y_{r_{max}}$ and $x_{5max} = Z_{r_{max}}$. As illustrated in Fig. 3, $X_{r_{max}}$, $Y_{r_{max}}$ and $Z_{r_{max}}$ are respectively the x, y and z coordinates of the intersection point between the straight line $P_0 - P_f$ and the boundary of the robot workspace. With these notations and being given that the goal is to minimize t_r given in (12), the objective function $f(\mathbf{x})$ to be minimized over $\mathbf{x}$ can be defined as:

$$f(\mathbf{x}) = \frac{2\sqrt{(x_3 - X_0)^2 + (x_4 - Y_0)^2 + (x_5 - Z_0)^2}}{x_1} + \frac{x_1}{x_2} + \frac{x_2}{J} \qquad (13)$$

The optimization problem includes also the following constraints.

1) Bound constraints: $0 < x_1 \leq x_{1max}$, $0 < x_2 \leq x_{2max}$, $X_0 \leq x_3 \leq x_{3max}$, $Y_0 \leq x_4 \leq x_{4max}$, and $Z_0 \leq x_5 \leq x_{5max}$.

2) Nonlinear inequality constraints:

$$d_a = \frac{x_1}{x_2} - \frac{x_2}{J} \geq 0 \qquad (14)$$

$$d_v = \frac{2\sqrt{(x_3 - X_0)^2 + (x_4 - Y_0)^2 + (x_5 - Z_0)^2}}{x_1} - \frac{x_1}{x_2} - \frac{x_2}{J} \geq 0 \qquad (15)$$

$$X_f - \delta \leq x_1 \cos\theta_r \cos(\alpha_{xy}) t_f + x_3 \leq X_f + \delta \qquad (16)$$

$$Y_f - \delta \leq x_1 \cos\theta_r \sin(\alpha_{xy}) t_f + x_4 \leq Y_f + \delta \qquad (17)$$

where $\mathbf{P}_f = [X_f, Y_f, Z_f]$, $\alpha_{xy} = \arctan(x_4 - Y_0, x_3 - X_0)$ is the angle between the x-axis and the vertical plane containing P_0 and P_r, and $\theta_r = \arctan\left(x_5 - Z_0, \sqrt{(x_3 - X_0)^2 + (x_4 - Y_0)^2}\right)$ is the angle between the horizontal plane and $P_0 - P_r$ as shown in Fig. 3. The constraints (14) and (15) are related to the S-curve ensuring that d_a and d_v are nonnegative, while (16) and (17) ensure that the object B reaches the target horizontal position $[X_f, Y_f]$ with a tolerance of δ (chosen as

$\delta = 0.005$ m in Section IV). Note that the final time t_f in (9) is a function of x_1 and x_5 as follows:

$$t_f = \frac{x_1 \sin\theta_r}{g} + \sqrt{\left(\frac{x_1 \sin\theta_r}{g}\right)^2 + 2\frac{x_5 - Z_f}{g}} \qquad (18)$$

since the release velocity vector is given by $\mathbf{V}_r = x_1[\cos\theta_r \cos\alpha_{xy}, \cos\theta_r \sin\alpha_{xy}, \sin\theta_r]$.

3) Nonlinear equality constraint: $\alpha_{xy} - \alpha = 0$, where $\alpha = \arctan(Y_f - Y_0, X_f - X_0)$. This constraint guarantees that the release point lies in the vertical plane containing P_0 and P_f.

To sum up, the problem of minimizing $f(\mathbf{x})$ over $\mathbf{x}$ subject to the constraints defined above is a nonlinear constrained optimization problem where the S-curve time minimization is coupled with the determination of the ballistic motion release configuration. This problem can be solved with standard nonlinear programming solvers.

E. Optimization problem simplification

The nonlinear constrained optimization problem introduced in Section III-D can be substantially simplified. First, as shown in Fig. 3, since both the robot throw motion acceleration phase and the ballistic motion lie in the vertical plane containing P_0, P_r and P_f, the three-dimensional problem can easily be converted into a planar one (setting the tolerance zone defined in (16) and (17) aside). Moreover, the pick position P_0 is known. Hence, without loss of generality, we can choose the reference frame origin as being P_0 and its x-axis oriented along the projection of the straight line $P_0 - P_r$ on the (x,y) plane. Then, the vertical plane containing the acceleration phase and the ballistic motion is the (x,z) plane. Accordingly, referring to (8a) and 8b, $y_B(t) = 0, \forall t$ and after substituting the time t calculated from the equation of $x_B(t)$ into $z_B(t)$, and rearranging the terms, the following expression is obtained:

$$z_B = -\frac{g}{2}\frac{(x_B - P_r\cos\theta_r)^2}{(V_r\cos\theta_r)^2} + x_B\tan\theta_r \qquad (19)$$

where, with the chosen reference frame, $\alpha_{xy} = 0$ and $\mathbf{P}_0 = \mathbf{0}$ so that $\mathbf{P}_r = [X_r, Y_r, Z_r] = P_r[\cos\theta_r, 0, \sin\theta_r]$ and $\mathbf{V}_r = V_r[\cos\theta_r, 0, \sin\theta_r]$. By substituting the coordinates of the target position P_f into (19) ($x_B = X_f$ and $z_B = Z_f$), the release velocity V_r can be expressed as a function of the distance P_r and the release angle θ_r as follows:

$$V_r = \frac{-P_r + \left(X_f/\cos\theta_r\right)}{\sqrt{\frac{2}{g}(X_f\tan\theta_r - Z_f)}} \qquad (20)$$

with $\arctan\left(Z_f/X_f\right) \le \theta_r \le \pi/2$. Equation (20) is the ballistic motion target constraint that shall be satisfied for the object to reach the target P_f.

Besides, as explained in Section III-D, we have $P = 2P_r$ and $V = V_r$ where P and V are the displacement and maximum velocity of the S-curve motion profile used to generate the throw motion acceleration phase. Referring to Section II-C, the S-curve total time $T = 2t_r$ is an increasing function of the distance P and thus of $P_r = P/2$. Consequently, minimizing t_r, which is $f(\mathbf{x})/2$ in Section III-D, is equivalent to minimizing P_r, i.e., the minimum-time throw motion is obtained by minimizing P_r.

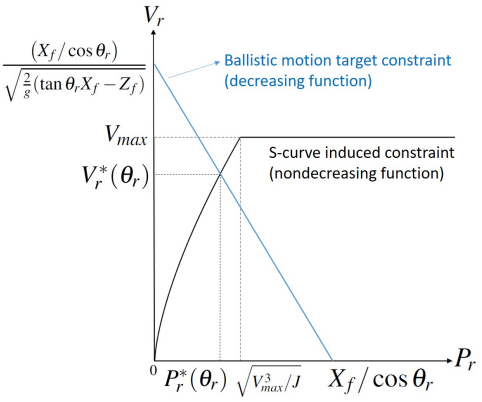


Fig. 4. The ballistic motion constraint and the minimum-time S-curve induced constraint as functions $V_r(P_r)$. For a given θ_r, there exists a unique couple $(P_r^*(\theta_r), V_r^*(\theta_r))$ satisfying both constraints.

The minimization of P_r is subjected to the ballistic motion target constraint in (20) and to a constraint due to the minimum-time S-curve motion profile. Indeed, the latter imposes a relationship between P and V and thus also between P_r and V_r since $P = 2P_r$ and $V = V_r$. As pointed out in Section II-C, this relationship takes the form of V_r being a continuous nondecreasing function of P_r composed of either two or three segments. For each of these segments, the expression of $V_r(P_r)$ is known as detailed in Section II-C for $V(P)$. It is worth noting that this function $V_r(P_r)$ is nondecreasing while, in (20), V_r is a decreasing linear function of P_r for a given θ_r. Hence, as illustrated in Fig. 4, for any value of θ_r, $\arctan\left(Z_f/X_f\right) \le \theta_r \le \pi/2$, there exists a unique couple $(P_r^*(\theta_r), V_r^*(\theta_r))$ satisfying both the function $V_r(P_r)$ from the constraint induced by the minimum time S-curve and the function $V_r(P_r)$ from the ballistic motion target constraint in (20). These values of P_r and V_r depend on θ_r, and this is the rationale of the notations $P_r^*(\theta_r)$ and $V_r^*(\theta_r)$. Hence, minimizing P_r boils down to finding the value of θ_r that yields the smallest $P_r^*(\theta_r)$.

Based on the above analysis, an efficient method to determine $P_r^*(\theta_r)$ can be devised as follows. First, note that obtaining a closed-form expression of P_r as a function of θ_r, i.e. eliminating V_r in (20) by means of the expression $V_r(P_r)$ obtained from the S-curve induced constraint (Section II-C), is difficult except in the case where $V_r(P_r) = V_{max}$. Indeed, in this case, (20) implies

$$P_r = \left(X_f/\cos\theta_r\right) - V_{max}\sqrt{2(X_f\tan\theta_r - Z_f)/g}. \qquad (21)$$

An analysis of this function shows that it possesses a unique minimum $P_r^m = P_r(\theta_r^m)$ for $\arctan\left(Z_f/X_f\right) \le \theta_r \le \pi/2$. Moreover, this minimum can be straightforwardly calculated by solving a univariate nonlinear equation in θ_r obtained from $dP_r/d\theta_r = 0$. Then, the two cases defined above in Section II-C have to be distinguished.

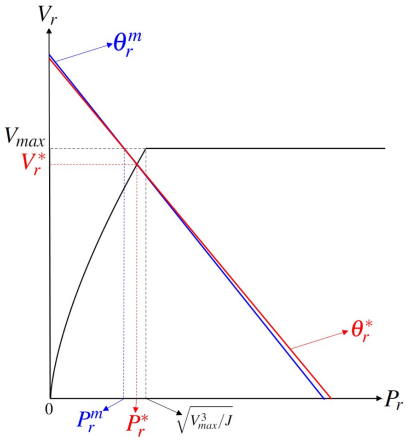


Fig. 5. In the case $\sqrt{JV_{max}} < A_{max}$, this figure shows the minimum $P_r^m = P_r(\theta_r^m)$ of the function in (21) (case where $P_r^m < \sqrt{V_{max}^3/J}$) and the optimal release configuration P_r^*, V_r^* and θ_r^*.

Case 1 – $\sqrt{JV_{max}} < A_{max}$. Referring to Fig. 4, if $P_r^m \geq \sqrt{V_{max}^3/J}$, the problem is solved since the optimal release configuration is $P_r^* = P_r^m$ and $V_r^* = V_{max}$. Otherwise, as illustrated in Fig. 5, the S-curve induced constraint is not satisfied at P_r^m and the optimal release configuration (P_r^*, V_r^*) lies on the first segment $V_r = \sqrt[3]{JP_r^2}$ of this constraint. The optimal release configuration is obtained by solving the following nonlinear optimization problem where the nonlinear equation $g(P_r, \theta_r) = 0$ is obtained by replacing V_r in (20) by $\sqrt[3]{JP_r^2}$.

$$\min_{P_r, \theta_r} P_r \quad \text{s.t.} \quad g(P_r, \theta_r) = 0 \qquad (22)$$
$$0 \leq P_r \leq \sqrt{V_{max}^3/J}$$
$$\arctan\left(Z_f/X_f\right) \leq \theta_r \leq \pi/2$$

The solution of this optimization problem is P_r^* and θ_r^* where P_r^* is the smallest value of P_r allowing to reach the ballistic motion target with the minimum-time S-curve motion profile for the throw motion acceleration phase. In other words, P_r^*, θ_r^* and $V_r^* = \sqrt[3]{JP_r^{*2}}$ constitute the optimal release configuration. As shown in Fig. 5, it is worth noting that the ballistic motion target constraints for θ_r^m and θ_r^* are close to each other (this comes from the fact that $X_f/\cos\theta_r$ is an increasing function of θ_r). Hence, using P_r^m and θ_r^m as an initial estimate of the solution of (22) leads to a fast solving of this optimization problem.

Case 2 – $\sqrt{JV_{max}} \geq A_{max}$. Referring to Fig. 2 and Section II-C, if the point (P_r^m, V_{max}) lies on the third segment of the S-curve induced constraint, i.e. $P_r \geq \left(JV_{max}^2 + V_{max}A_{max}^2\right)/\left(2JA_{max}\right)$, then the optimal release configuration is $P_r^* = P_r^m$ and $V_r^* = V_{max}$. Otherwise, a closed-form expression of P_r as a function of θ_r is obtained by replacing V_r in (20) by A_{max}^2/J. Similarly to the case of (21) discussed above (with A_{max}^2/J in place of V_{max}), this function possesses a unique minimum $P_r^M = P_r(\theta_r^M)$ which can be easily found by solving a univariate nonlinear equation. If $P_r^M \leq A_{max}^3/J^2$, then the optimal release configuration (P_r^*, V_r^*) lies on the first segment of the S-curve induced constraint and can be efficiently calculated by solving the optimization problem (22) with the initial estimate (P_r^M, θ_r^M). Otherwise, the optimal release configuration (P_r^*, V_r^*) lies on the second segment of the S-curve induced constraint. It can then be obtained by solving an optimization problem similar to (22) but with $g(P_r, \theta_r) = 0$ obtained from the expression of $V(P)$ at line 12 of Algorithm 1, and with the initial estimate (P_r^M, θ_r^M).

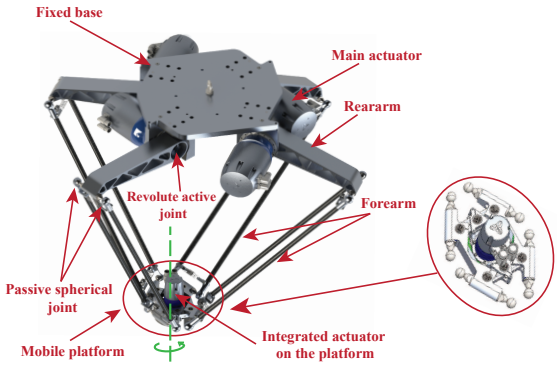


Fig. 6. CAD view of the T3KR parallel robot and its main components.

In summary, the optimal release configuration (P_r^*, V_r^*) yielding a minimum-time throw motion can be efficiently obtained by solving one or two univariate nonlinear equations and one optimization problem (22) having two variables with bound constraints and one equality constraint. Solving this rather simple optimization problem takes a short time especially since an initial estimate close to the optimal solution is known. Hence, compared to the nonlinear constrained optimization problem introduced in Section III-D, the solving method presented in this section is more efficient. Furthermore, it provides insight into the nature of the problem of determining the optimal release configuration. It has notably been pointed out that the corresponding optimization problem possesses a unique minimum, i.e., as defined in this paper, the optimal release configuration is unique.

IV. Real-Time Experimental Results

A. Description of the Experimental Testbed

The T3KR robot (3-Translation Kinematically Redundant robot) is a rigid-link parallel robot with an optimized workspace to footprint ratio, designed by Tecnalia, LIRMM and SATT AxLR. The CAD view of T3KR is shown in Fig. 6. It is a kinematically redundant Delta-like parallel robot having four kinematic chains providing three translations at the mobile platform. An actuator located on the mobile platform allows the rotation of the end-effector around the z-axis but this rotation is not needed in the present work. Due to its asymmetrical mechanical structure, T3KR has an elliptical workspace in top view. The actuator of each kinematic chain can generate a maximum torque of 530 Nm. The robot is controlled by an industrial PC equipped with the B&R Automation studio with a sampling frequency

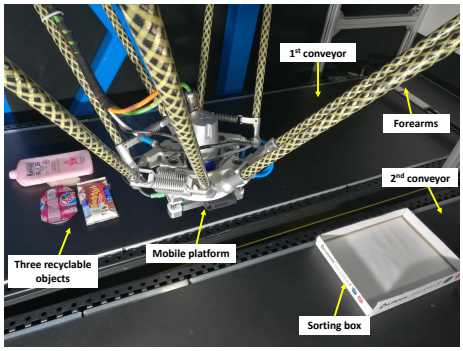


Fig. 7. The experimental testbed: T3KR robot, conveyors, recyclable objects and the sorting box.

of 2500 Hz. The motors motion control is performed by a PID controller integrated in ACOPOSmulti B&R drive. The mobile platform can reach a maximum speed of 6 m/s (i.e. $V_{max} = 6\,m/s$), a maximum acceleration of 12 G (i.e. $A_{max} = 12G = 120\,m/s^2$) and carry a maximum payload of 5 kg. The experimental setup is displayed in Fig. 7. It includes the T3KR robot, two conveyors, three objects (recyclables) of different masses, sizes and materials, and a sorting box to deposit the objects.

B. Description of P&P and P&T Reference Trajectories

To validate the efficiency of the proposed P&T technique, i.e., the minimum-time throw motion introduced in Section III, comparisons with a standard P&P method as the one used in [3] and with the P&T approach proposed in [12] are conducted in real-time experiments. The three associated reference trajectories are described hereafter. All the length units are in meters.

1) Pick-and-Place Reference Trajectory: The 3D view of the adopted P&P trajectory in Cartesian space is depicted in Fig. 8. A standard P&P trajectory consists of a vertical movement, followed by a horizontal movement and a final vertical movement. The robot moves from the initial position $\mathbf{P}_{initial} = [0, -0.77, 0]$ to the first pick position $\mathbf{P}_{pick1} = [-0.2, -0.25, -0.85]$ (the conveyor height is -0.85). After picking the object, it follows the first P&P trajectory (shown in the red) to place the object at $\mathbf{P}_{place} = [0.3, 0.4, -0.85]$. Then, the robot follows the second P&P trajectory (shown in green) to pick the second object at $\mathbf{P}_{pick2} = [-0.2, -0.35, -0.85]$ and moves back to $\mathbf{P}_{place}$. The same movement is repeated for the third object located at $\mathbf{P}_{pick3} = [-0.3, -0.35, -0.85]$. After placing the third object, the robot goes back to the initial position $\mathbf{P}_{initial}$.

2) Existing Pick-and-Throw Reference Trajectory: The P&T trajectory used in [12] is shown in Fig. 9. This trajectory is similar to a P&P trajectory and can be described as follows: After moving the end effector from its initial position $\mathbf{P}_{initial}$ to the first pick position $\mathbf{P}_{pick1}$, the robot performs a vertical movement followed by a horizontal movement towards the release point. It throws the object to the target position $\mathbf{P}_f = [0.3, 0.4, -0.85]$ along the horizontal movement and then makes a U-turn to decelerate

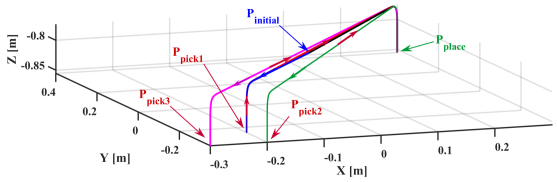


Fig. 8. 3D-view of P&P reference trajectory in Cartesian space.

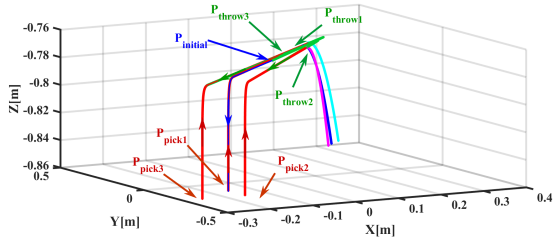


Fig. 9. 3D-view of existing P&T reference trajectory [12].

towards the next pick position $\mathbf{P}_{pick2}$. The same movement is repeated for the third object located at $\mathbf{P}_{pick3}$. After throwing the third object, the robot returns back to its initial position. In [12], the path of the P&T trajectory is presented without describing the release point computation. In the present experiments, the release points are computed with the methodology proposed in Section III where only the horizontal distance to the release position has to be optimized because, in the P&T trajectory of [12], Z_r is fixed, $V_{zr} = 0$ m/s and $\theta_r = 0$ deg.

3) Proposed Pick-and-Throw Reference Trajectory: The P&T trajectory proposed in this work is depicted in Fig. 10. The robot follows the first picking motion from $\mathbf{P}_{initial}$ to $\mathbf{P}_{pick_1}$. After picking the object, the optimal release configuration, including $\mathbf{P}_{throw_1} = \mathbf{P}_{r1}$ and $\mathbf{V}_{r1}$, is calculated as described in Section III. The robot accelerates while moving along a straight line towards the calculated release point $\mathbf{P}_{throw_1}$ at which it throws the object towards the target $\mathbf{P}_f$. Once released, the object follows a ballistic trajectory to $\mathbf{P}_f$ while the robot decelerates back to pick the second object. The same cyclic movement is repeated for the second and the third objects, located at $\mathbf{P}_{pick_2}$ and $\mathbf{P}_{pick_3}$, respectively. After throwing the third object, the robot moves back to $\mathbf{P}_{initial}$. As a numerical example, in a case study with 30% of the maximum dynamic performances of the T3KR robot (i.e. 30% of $V_{max} = 6\,m/s$, $A_{max} = 12\,m/s^2$ and $J = 3000\,m/s^3$) and for a target position inside its workspace, $P_f = [0.3, 0.4, -0.85]$, the following optimal release configurations are calculated with the method proposed in Section III: $\mathbf{P}_{throw_1} = [0.09, 0.13, -0.75]$ and $V_{r1} = 1.8\,m/s$, $\mathbf{P}_{throw_2} = [0.15, 0.17, -0.75]$ and $V_{r2} = 1.8\,m/s$, $\mathbf{P}_{throw_3} = [0.12, 0.17, -0.75]$ and $V_{r3} = 1.8\,m/s$.

C. Obtained Experimental Results

The performance of the proposed P&T approach is evaluated through three experimental scenarios. These scenarios

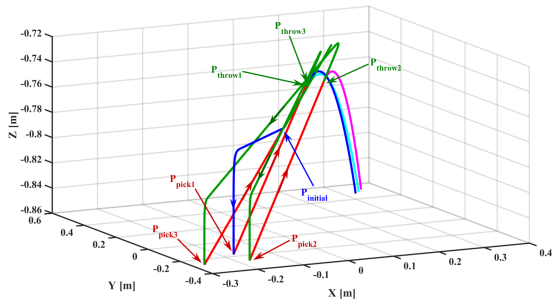


Fig. 10. 3D-view of the proposed P&T reference trajectory.

are performed under different operating conditions (acceleration, speed, and different types of objects of different sizes, materials and masses, etc.). The three types of objects considered are: A paper object of 7 g of mass, a metallic can with a mass of 16 g ($\Delta_{mass} = +128\%$ w.r.t the first object) and a plastic package of 49 g of mass ($\Delta_{mass} = +600\%$ w.r.t the first object). The demonstration video of the three tested scenarios is available at: https://youtu.be/4bRvqKurMJU. These three scenarios are detailed in the sequel.

1) Scenario 1: It consists in comparing the usual P&P (cf. Section IV-B1) and the proposed P&T (cf. Section IV-B3) trajectories inside the workspace, i.e., both the pick and the target positions are located inside the robot workspace. For comparison purposes, we use different percentages of the maximum speed ($V_{max} = 6\,m/s$) and maximum acceleration ($A_{max} = 12\,G$) of T3KR end-effector. These percentages are 15%, 30% and 40%, which correspond to 0.9 m/s, 1.8 m/s and 2.4 m/s of maximum velocity, respectively and to 1.8 G, 3.6 G and 4.8 G of maximum accelerations, respectively. The desired Cartesian positions, generated using the minimum-time S-curve motion profile, are shown on the left side of Figs. 11 for the P&P tasks and on the right side of Figs. 11 for the proposed P&T tasks inside the workspace. Figure 12 depicts the time evolution of the desired Cartesian velocity V_s and acceleration A_s of the robot end-effector obtained with the proposed P&T method for the case study that corresponds to 1.8 m/s of maximum velocity and 3.6 G of maximum acceleration. As clearly shown in this figure, V_s and A_s are continuous. Moreover, they reach their maximum constraints (i.e. the maximum of V_s is 1.8 m/s while the maximum of A_s is 36 m/s^2, equivalent to 3.6 G) which is a consequence of the time optimality of the proposed P&T method. Similarly, the velocities and accelerations for the other case studies and those for the next scenarios also attain their maximum values. The corresponding curves are not shown due to space limitations.

2) Scenario 2: This scenario consists in comparing, inside the robot workspace, the proposed P&T method with the existing P&T strategy of [12] (cf. Section IV-B2). As in the previous scenario, this test is performed under 15%, 30% and 40% of maximum acceleration and maximum velocity. The desired Cartesian positions of the existing P&T strategy are depicted on the left side of Figs. 13.

3) Scenario 3: In this scenario, the proposed P&T method is compared with the existing P&T method of [12], but with a target position outside the robot workspace. The generated P&T trajectories are executed with 35%, 40% and 45% of maximum velocity and maximum acceleration, which corresponds to 2.1 m/s, 2.4 m/s and 2.7 m/s of maximum velocity, respectively and to 4.2 G, 4.8 G and 5.4 G of maximum acceleration, respectively. We start with a percentage of 35% in this scenario because, with a lower percentage, the speed is not sufficient to throw the object towards the desired target located outside the workspace, $P_f = [0.45, 0.65, -0.85]$. The evolution of desired positions is depicted on the left side of Figs. 14 for the existing P&T method [12] and on the right side of Figs. 14 for the proposed P&T approach. As it can be seen, with the increase in the operating acceleration, the calculated release point is closer to the picking position. Therefore, the duration of the whole trajectory is reduced and the robot can perform more picks per minute.

D. Results Discussion

The obtained experimental results demonstrate the superiority of the proposed P&T approach over the standard P&P method and the existing P&T method of [12]. The number of picks per minute obtained by each method in each case study are summarized in the Table I. Inside the workspace, using the P&P method, the robot can perform 24 to 51 picks per minute when the acceleration increases from 1.8 G to 4.8 G. While for the existing P&T method, it can perform 29 to 75 picks/min. However, with the proposed P&T approach, the obtained number of picks per minute goes from 30 to 120. Therefore, the proposed P&T method outperforms the two other strategies. At 4.8 G of maximum acceleration, the existing P&T method increases the number of picks per minute by up to 32% compared to the P&P approach. The proposed P&T strategy improves the performance by 57.5% over the standard P&P approach and by 37% over the existing P&T method, which are significant performance improvements for applications requiring reduced processing time and high productivity. Moreover, the improvement brought by the proposed P&T method over the existing approaches illustrate the relevance of determining an optimal release configuration. Regarding scenario 3, the P&T is the only candidate to perform such a task since the P&P method cannot place an object outside the workspace of the robot. At an acceleration of 5.4 G, the robot can reach 65 picks per minute by adopting the existing P&T method and 81 picks per minute by using the proposed P&T method. In conclusion, the proposed P&T method largely outperforms the two other methods in all case studies.

V. Conclusions and Future Work

In this paper, a new approach to generate a time-optimal throwing trajectory has been proposed. This approach consists in determining an optimal release configuration resulting in a minimum-time throw motion while ensuring that the released object will reach the desired target. The

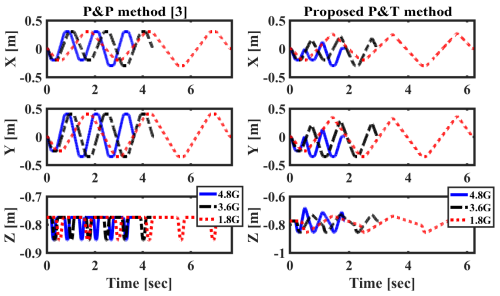


Fig. 11. Scenario 1: Evolution of the desired Cartesian positions versus time for the P&P task [3] and the proposed P&T task for different values of the maximum acceleration.

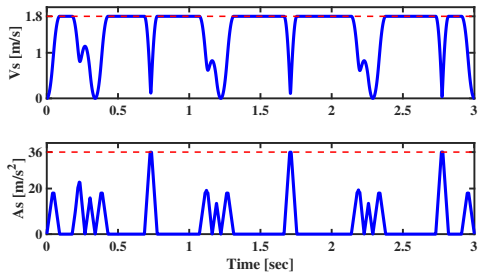


Fig. 12. Scenario 1: Evolution of the desired Cartesian velocity and acceleration of the robot end-effector versus time for the proposed P&T method at 1.8 m/s of maximum velocity and 3.6 G of maximum acceleration.

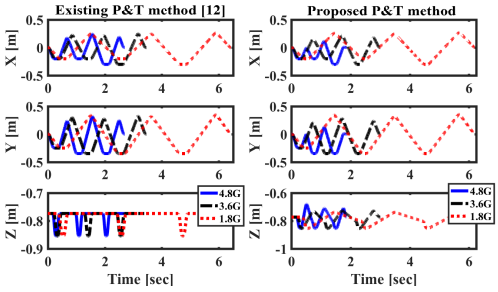


Fig. 13. Scenario 2: Evolution of the desired Cartesian positions versus time for the existing P&T method [12] and the proposed P&T one for different values of the maximum acceleration.

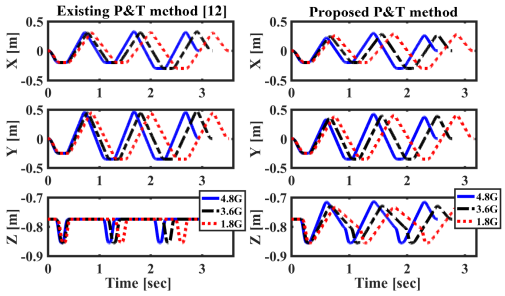


Fig. 14. Scenario 3: Evolution of the desired Cartesian positions versus time for the existing P&T method [12] and the proposed P&T one for different values of the maximum acceleration.

corresponding optimization problem has been formulated in two different ways where the second one allows the optimal release configuration to be efficiently computed. A comparison between the proposed P&T procedure, an existing P&T method and the conventional P&P has been conducted through real-time experiments on the T3KR parallel robot. The obtained experimental results validate the efficiency of the proposed P&T method, over the two other methods, in terms of processing time minimization and, thereby, of productivity maximization. The optimized Pick-and-Throw trajectory, combined with an AI technique to identify recyclables, can be applied in real waste recycling industry. Furthermore, this work may be extended to consider not only the position of the object into the desired target but also its orientation in the target landing position.

REFERENCES

[1] W.-B. Li, G.-Z. Cao, X.-Q. Guo, and S.-D. Huang, "Development of a 4-dof scara robot with 3R1P for pick-and-place tasks," in *2015 6th International Conference on Power Electronics Systems and Applications (PESA)*, pp. 1–5, IEEE, 2015.

[2] R. Clavel, "Device for the movement and positioning of an element in space," Switzerland, URL https://patents.google.com/patent/US4976582A, US Patent 4976582A, 1990.

[3] V. Nabat, M. O Rodriquez, O. Company, S. Krut, and F. Pierrot, "Par4: very high speed parallel robot for pick-and-place," in *2005 IEEE/RSJ International conference on intelligent robots and systems*, pp. 553–558, IEEE, 2005.

[4] F. Pierrot, V. Nabat, O. Company, S. Krut, and P. Poignet, "Optimal design of a 4-dof parallel manipulator: From academia to industry," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 213–224, 2009.

[5] G. S. Natal, A. Chemori, and F. Pierrot, "Dual-space control of extremely fast parallel manipulators: Payload changes and the 100g experiment," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1520–1535, 2014.

[6] BHS, "Max-AI AQC robotic sorter," https://www.youtube.com/watch?v=2gjUpDnJrZA, 2018.

[7] H. Frank, "Throwing of axial-symmetric objects in production systems," in *2010 12th International Conference on Computer Modelling and Simulation*, pp. 1–1, IEEE, 2010.

[8] H. Frank, "Throwing of objects: A new technology for factory automation," in *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 4–5, IEEE, 2009.

[9] H. Tsukagoshi, E. Watari, K. Fuchigami, and A. Kitagawa, "Casting device for search and rescue aiming higher and faster access in disaster site," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4348–4353, IEEE, 2012.

[10] A. Fagiolini, H. Arisumi, and A. Bicchi, "Casting robotic end-effectors to reach faraway moving objects," *Computing Research Repository (CoRR)*, vol. abs/1101.2268, 2011.

[11] BHS, "Max-AI AQC-C Recycling CoBot," https://www.youtube.com/watch?v=tEAr1w1Jxww, 2019.

[12] F. Raptopoulos, M. Koskinopoulou, and M. Maniadakis, "Robotic pick-and-toss facilitates urban waste sorting," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pp. 1149–1154, IEEE, 2020.

TABLE I
NUMBER OF PICKS PER MINUTES VERSUS OPERATING ACCELERATION FOR THE THREE CASE STUDIES.

		1.8 G	3.6 G	4.2 G	4.8 G	5.4 G
Inside Workspace	P&P	24	42	-	51	-
	Existing P&T	29	56	-	75	-
	Proposed P&T	30	68	-	120	-
Outside Workspace	Existing P&T	-	-	54	60	65
	Proposed P&T	-	-	62	71	81

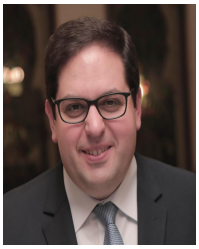
[13] K. M. Lynch and M. T. Mason, "Dynamic manipulation with a one joint robot," in *Proceedings of International Conference on Robotics and Automation*, vol. 1, pp. 359–366, IEEE, 1997.
[14] N. Uzzaman, S. Hossain, and A. Hossain, "Geometrical approach for determining the throwing destination of an automatic throwing robot," in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pp. 1–5, IEEE, 2016.
[15] M. Okada, A. Pekarovskiy, and M. Buss, "Robust trajectory design for object throwing based on sensitivity for model uncertainties," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3089–3094, IEEE, 2015.
[16] A. Sintov and A. Shapiro, "A stochastic dynamic motion planning algorithm for object-throwing," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2475–2480, IEEE, 2015.
[17] F. Hu, Q. Liu, J. Cheng, and Y. He, "Flight trajectory simulation of robotic throwing shuttlecock," in *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pp. 77–82, IEEE, 2018.
[18] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.
[19] R. H. Castain and R. P. Paul, "An on-line dynamic trajectory generator," *The International Journal of Robotics Research*, vol. 3, no. 1, pp. 68–72, 1984.
[20] A. Piazzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE transactions on industrial electronics*, vol. 47, no. 1, pp. 140–149, 2000.
[21] S. Macfarlane and E. A. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Transactions on robotics and automation*, vol. 19, no. 1, pp. 42–52, 2003.
[22] K.-H. Rew and K.-S. Kim, "A closed-form solution to asymmetric motion profile allowing acceleration manipulation," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2499–2506, 2009.
[23] H. Li, "A jerk-constrained asymmetric motion profile for high-speed motion stages to reduce residual vibration," *International Journal of Computer Applications in Technology*, vol. 53, no. 2, pp. 149–156, 2016.
[24] K. D. Nguyen, T.-C. Ng, and I.-M. Chen, "On algorithms for planning s-curve motion profiles," *International Journal of Advanced Robotic Systems*, vol. 5, no. 1, p. 11, 2008.
[25] Y. Fang, J. Hu, W. Liu, Q. Shao, J. Qi, and Y. Peng, "Smooth and time-optimal s-curve trajectory planning for automated robots and machines," *Mechanism and Machine Theory*, vol. 137, pp. 127–153, 2019.
[26] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 415–426, 2008.
[27] Y. Bai, X. Chen, H. Sun, and Z. Yang, "Time-optimal freeform s-curve profile under positioning error and robustness constraints," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1993–2003, 2018.
[28] L. Biagiotti and C. Melchiorri, "Optimization of generalized s-curve trajectories for residual vibration suppression and compliance with kinematic bounds," *IEEE/ASME Transactions on Mechatronics*, 2020.
[29] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer-Verlag, 2008.
[30] M. Gouttefarde, G. Hassan, and A. Chemori, "An alternative algorithm to compute minimum-time third order polynomial s-curve trajectories," *Research report*, https://seafile.lirmm.fr/f/34c0937a73ed4e9da8d3/?dl=1, 2021.
[31] W. Khalil and E. Dombre, *Modeling identification and control of robots*. CRC Press, 2002.

Ghina HASSAN received the B.Eng. degree from the Lebanese University, Tripoli, Lebanon, in 2017 and the M.Eng. form the Lebanese University, Tripoli, Lebanon and the University Paris-Est Créteil, Paris, France, in 2018. She is currently a PhD Student at the University of Montpellier Laboratory of Informatics, Robotics, and Microelectronics. Her research interests include optimized trajectory generation and advanced control of parallel kinematic manipulators.

Marc GOUTTEGARDE received the B.Eng. degree in mechatronics from INSA, Strasbourg, France, in 2001, and the M.Sc. and Ph.D. degrees in mechanical engineering from Laval University, Québec, Canada, in 2002 and 2005, respectively. From 2005 to 2007, he was a Postdoctoral Fellow with INRIA, Sophia-Antipolis, France. He is currently a CNRS Senior Researcher with LIRMM, Montpellier, France, where he is mainly working on cable-driven parallel robots.

Ahmed CHEMORI received the M.Sc. and Ph.D. degrees both in automatic control from the Grenoble Institute of Technology, Grenoble, France, in 2001 and 2005, respectively. He has been a Postdoctoral Fellow with the Automatic Control Laboratory, Grenoble, France, in 2006. He is currently a tenured Research Scientist in automatic control and robotics with LIRMM. His research interests include nonlinear, adaptive, and predictive control and their applications in robotics.

Pierre-Elie HERVE Graduated in Mechanical engineer and Mechatronics at the University of Technology of Belfort Montbéliard (UTBM – Belfort, France 2006 - 2011) and Cranfield University (UK, 2011). Research engineer at LIRMM (2012 – 2016) and R&D engineer for Tecnalia since 2016. He has been involved in several national and European projects on cable robotics. He takes part to the development of advanced control laws, advanced modeling and industrial demonstrations.



Maher EL RAFEI received his M. Eng. degree in Industrial Control from the Lebanese University (2004). He received his M.Sc. and Ph.D. degrees, both in Automatic Control from The University of Montpellier II (2005) and Grenoble Institute of Technology (2008), France. He is currently an associate professor at the Lebanese University and a researcher in Automatic Control and Robotics at CRSI. His research interests include nonlinear control with special emphasis on their real-time application in robotics.

Clovis FRANCIS received his Engineering degree from the Lebanese University, Faculty of Engineering in 1990, the Master degree from the University Paul Sabatier – Toulouse in 1991, the Ph.D degree from the University paris XI – France in 1994 and the Accredidation to Supervise Research (HDR) in 2009. Currently he is a Professor at the Lebanese University in Automatic Control, Fault Tolerant Control and system diagnosis.

Damien SALLE received his Ph.D. in Robotics from University Paris-Sorbonne (Paris) for his research on surgical robotics. He then created and managed the R&D department of Robosoft, developing service robotics applications and products (autonomous vehicles, AGV, off-road mobile robots and companion robots for elderlies). Since 2010, he leads the Robotics activities in Tecnalia, developing and transferring technologies such as Mobile and Dual-arm manipulation, AI for robotics, Collaborative robotics, 3D vision-guided industrial robots etc.