2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Trajectory Planning for Non-Prehensile Object Transportation

Lingyun Chen[1,2], Haoyu Yu[1], Liding Zhang[1], Abdeldjallil Naceri[1], Abdalla Swikir[1,2,3] and Sami Haddadin[1,2]

***Abstract*— Non-prehensile transportation of unstable objects presents a challenging task in robotics. To ensure the success of the transportation, it is necessary to consider both the object's stability via contact dynamics and the motion constraints of the robot. We propose two novel trajectory planning methods derived from sampling and dynamic programming algorithms, tested on a 7-DoF Franka Emika robot against common strategies like Model Predictive Control (MPC) and S-curve planning, particularly under the constraint of a non-rotating tray. The results demonstrate the effectiveness of our methodologies in improving transportation speed. This research contributes to advancements in robotic manipulation techniques by tackling non-prehensile manipulation of dynamically unstable objects.**

I. Introduction

Non-prehensile manipulation refers to the manipulation of a target object without grasping it. Precisely, the robot controls the target object in at least one degree of freedom (DoF) in only one direction [1]. A typical example of non-prehensile manipulation is transporting an object on a tray. The tray exerts a normal force perpendicular to its surface, pushing the object away. Since this force can only push, not pull, the robot cannot directly control the object's vertical movement. However, it is still possible to manipulate the object by combining the force exerted on the target object with other forces, such as friction and gravity.

Compared to common prehensile manipulation, non-prehensile manipulation offers advantages such as simpler end-effector design, the ability to manipulate multiple objects simultaneously, and the expansion of the robot's manipulation range [2]. Researchers have conducted extensive studies on non-prehensile manipulation in recent years, summarized in [1]. Different forms of non-prehensile manipulations, including throwing [2], hitting [3], pushing [4], and sliding [5], have been discussed.

These studies illustrate that non-prehensile manipulation involves the integration of external environmental forces acting on an object with those exerted by the robot for object manipulation. Consequently, it becomes crucial to examine both the forces influencing the object and its dynamic behavior. Researchers have proposed various models to address various manipulated objects. In [6], the Disk-on-Disk balance model was studied, while in [7], the Ball-Beam configuration was investigated. Beyond ensuring an object's stability, studies on robotic interception of flying objects further validate that non-prehensile manipulations possess the capacity to rapidly and efficiently modify the object's motion state [8] [9] [10]. In these three studies, researchers analyzed the contact force models between the ball and the tray and between the cube and the tray. The primary focus of prior studies is on employing sensors or machine vision technologies to inform the robot about the object's motion, which then guides the development of controllers to manipulate the object. Our research, however, diverges by examining non-prehensile manipulation through the development and analysis of trajectory planning strategies.

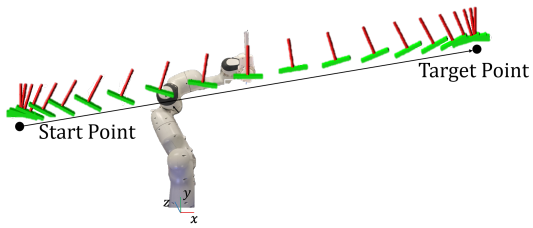


Fig. 1. Illustration of object and tray motion.

In our prior work [11], we presented a sampling-based trajectory planner for non-prehensile object transport using a tray-shaped end-effector. Building upon this foundation, we've refined our approach and introduced a novel dynamic programming method. We frame the problem as a point-to-point trajectory planning task, where the tray can execute both rotational and translational movements. The proposed trajectory's movement of the tray and the object is depicted in Fig. 1. To assess the efficacy of our methods, we implemented them on a physical robot. We compared them with existing methods, such as Model Predictive Control (MPC) and S-curve planning without tray rotation.

II. Related Work

Most research focused on designing and employing control methods, including MPC, to generate or track existing trajectories for non-prehensile object transportation. A model predictive non-sliding manipulation control method is proposed in [12]. The problem is transformed into a trajectory optimization control problem for tracking the target trajectory, using the object's friction cone as a constraint to prevent sliding between the object and the tray. Research on the transportation of unstable objects on mobile robots has been investigated in [13]. A trajectory tracking method computed

*Funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden. We also gratefully acknowledge the funding LongLeif GaPa gGmbH (Project Y) and the Lighthouse Initiative KI.FABRIK Bayern Phase 1: Aufbau Infrastruktur by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi). Please note that S. Haddadin had a potential conflict of interest as shareholder of Franka Emika GmbH.

[1] Authors are with the Munich Institute of Robotics and Machine Intelligence and the Chair of Robotics Science and Systems Intelligence, Technical University of Munich, Germany, [2] also with the Centre for Tactile Internet with Human-in-the-Loop (CeTI), [3] and with Department of Electrical and Electronic Engineering, Omar Al-Mukhtar University (OMU), Albaida, Libya. Email: {lingyun.chen@tum.de}

by solving a quadratic programming problem is designed in [14].

Several recent papers have also considered this problem from a trajectory planning perspective. In [15], researchers propose a method to calculate the inertial forces acting on the object along the target trajectory, identify points where the object might slide, and use tray rotation to prevent sliding. A non-prehensile transportation trajectory planning method based on S-curves is introduced in [16]. The researchers analyze the object's stability but do not model tray rotation. Furthermore, a method for rapidly generating non-prehensile transportation trajectories based on the admissible velocity propagation algorithm is proposed in [17]. In [18], a trajectory strategy is proposed to prevent liquid from spilling out during transportation by using tray rotation. This strategy assumes the container as a point mass suspended by a spherical pendulum and demonstrates its effectiveness in experiments but does not optimize the trajectory time.

Our approach differs from control methods as we address the non-prehensile transportation problem by analyzing and designing trajectory planning strategies. Compared to existing trajectory planning methods, we propose two novel approaches to generate near time-optimal trajectories.

III. Physical Modeling

In this analysis, we model the interaction scenario with the object and tray as rigid entities. The object is characterized as a slender, uniformly shaped cylinder, and considerations of air resistance are excluded from the assessment. Through a straightforward force analysis, the contact model of the object is as shown in Fig. 2(a), where $\boldsymbol{F_i}$ is the inertial force generated by horizontal acceleration, $\boldsymbol{F_g}$ is gravity, O is the center of mass of the object, and C is the center of the pressure. There are two constraints to keep the object stable: the center of pressure must be within the contact area (external force falls into the non-tipping area) to prevent the object from tipping over, and the contact force must not exceed the friction cone. This paper focuses on slender and unstable objects prone to tipping over rather than sliding under horizontal forces. Specifically, we consider objects that satisfy the following conditions:

$$\frac{2r}{h} \leq \mu_0, \tag{1}$$

where μ_0 is the static friction coefficient between the object and the tray, r and h are the object's radius and height, respectively.

Building upon the work of [19], who introduced a novel 6D rigid body contact model, this study investigates pressure distribution for maximizing stable object transportation. The model initially assumes a fixed joint between two contacting objects. We can determine relative motion by analyzing the wrench acting on this joint. A similar approach applies to analyzing contact forces between an object and a tray. Our goal is to maintain relative stability during transportation. This means finding an optimal pressure center within the contact area on the object's bottom surface. This center

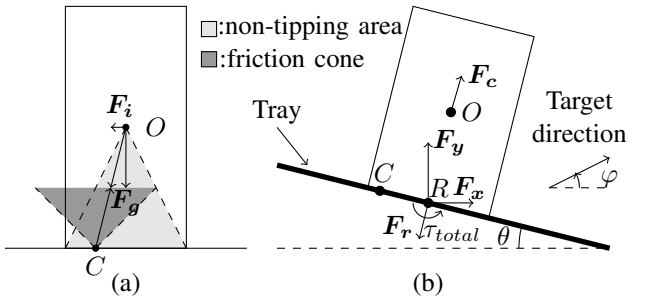


Fig. 2. Object physical modeling. (a) Stability conditions of the object. (b) Virtual wrench exerted by the tray at the bottom center R during transportation.

should maximize horizontal acceleration since the shift of the pressure center away from the object's center extends the lever arm of the normal force, thereby enabling higher acceleration. Ideally, the pressure center would be positioned at the object's edge for maximum acceleration. However, practical considerations such as control errors and object imperfections necessitate placing the pressure center within a "smaller" region. To quantify this offset, we define r_0 as the distance from the bottom surface center to the designated pressure center position.

The object is placed at the center of the tray, and the trajectory is designed to rotate around the center of its upper surface. The next step is to explore the relationship between rotational motion and the maximum horizontal acceleration while maintaining the object's stability. Let's consider the most straightforward scenario where the tray and the object move linearly in the horizontal direction. In Fig. 2(b), we illustrate the placement of a virtual fixed joint at the center of the bottom surface, designated as point R. This setup allows for calculating the linear force exerted by the wrench based on the tray's motion. The total linear force is divided into four distinct components, excluding gravitational $\boldsymbol{F_g}$ and inertial force $\boldsymbol{F_i}$: $\boldsymbol{F_x}$, the horizontal force that counterbalances inertial forces during acceleration; $\boldsymbol{F_y}$, the upward force that negates the effect of gravity; $\boldsymbol{F_r}$, the centripetal force that is crucial for maintaining circular motion; $\boldsymbol{F_c}$ represents the centrifugal force generated when the object rotates.

$$\boldsymbol{F_x} = -m\boldsymbol{a}, \boldsymbol{F_y} = -m\boldsymbol{g}, \|\boldsymbol{F_r}\| = \frac{m\omega^2 h}{2}, \|\boldsymbol{F_c}\| = -\frac{m\omega^2 h}{2} \tag{2}$$

where $\boldsymbol{g}$ is gravitational acceleration, $\boldsymbol{a}$ is the horizontal acceleration of the tray, $\boldsymbol{\omega}$ is the angular velocity of the tray, m is the object's mass, and h is the object's height. The torque required for the object to follow the rotation of the tray can be easily calculated as follows:

$$\boldsymbol{\tau_{total}} = I\boldsymbol{\alpha}, \tag{3}$$

where I is the object's scalar moment of inertia, and $\boldsymbol{\alpha}$ is the angular acceleration of the tray. This torque can be decomposed into two parts:

$$\boldsymbol{\tau_{total}} = \boldsymbol{\tau_{obj}} + \boldsymbol{\tau_{tray}} \tag{4}$$

$\boldsymbol{\tau_{obj}}$ is generated by force acting on the center of mass of

the object:

$$\boldsymbol{\tau_{obj}} = \overrightarrow{RO} \times (\boldsymbol{F_g} + \boldsymbol{F_i} + \boldsymbol{F_c}) \tag{5}$$

$\boldsymbol{\tau_{tray}}$ is generated by the force exerted on the object by the tray:

$$\boldsymbol{\tau_{tray}} = \overrightarrow{RC} \times (\boldsymbol{F_y} + \boldsymbol{F_x} + \boldsymbol{F_r}). \tag{6}$$

By substituting Equ.(3), (5), and (6) into Equ. (4) and solving for acceleration a, it can be concluded that the relationship between the horizontal acceleration and the rotational motion state of the tray is:

$$\|\boldsymbol{a}\| = \frac{\frac{I\|\boldsymbol{\alpha}\|}{m} + \frac{h\|\boldsymbol{g}\|}{2}\sin\theta + r_0\|\boldsymbol{g}\|\cos\theta - \frac{r_0\|\boldsymbol{\omega}\|^2 h}{2}}{\frac{h}{2}\cos\theta - r_0\sin\theta} \tag{7}$$

Where θ represents the angle between the tray and the horizontal plane. Similarly, considering that the target position and the initial position are not at the same horizontal level, assuming the angle between the target direction and the horizontal plane is φ, the horizontal and vertical accelerations during the acceleration phase are, respectively:

$$\begin{cases} a_x = \|\boldsymbol{a}\|\cos\varphi \\ a_y = \|\boldsymbol{a}\|\sin\varphi \end{cases} \tag{8}$$

By rearranging the above equations, it can be concluded that the relationship between acceleration and the tray's motion state when the object's transport direction is not horizontal should be:

$$\|\boldsymbol{a}\| = \frac{\frac{I\|\boldsymbol{\alpha}\|}{m} + \frac{h\|\boldsymbol{g}\|}{2}\sin\theta + r_0\|\boldsymbol{g}\|\cos\theta - \frac{r_0\|\boldsymbol{\omega}\|^2 h}{2}}{\frac{h}{2}\cos\varphi\cos\theta - \frac{h}{2}\sin\theta\sin\varphi - r_0\sin\theta\cos\varphi - r_0\cos\theta\sin\varphi} \tag{9}$$

IV. Trajectory Planning

In this section, we introduce two trajectory planning methods for non-prehensile object transportation: a sampling-based approach and a dynamic programming-based approach. We then conduct a comparative analysis of their strengths and weaknesses.

A. Sampling-based

Our physical model provides a connection between translational acceleration and rotational motion, allowing us to step-wise generate the complete trajectory from the initial conditions. The tray's motion is characterized by two degrees of freedom: translational and rotational. The translational and rotational motion state, $\boldsymbol{x_i^t}$ and $\boldsymbol{x_i^r}$ at time step $i \in \mathbb{N}$ are defined as follows:

$$\begin{cases} \boldsymbol{x_i^t} = \left[p_i, v_i = \frac{\partial p_i}{\partial t}, a_i = \frac{\partial^2 p_i}{\partial t^2}, j_i^t = \frac{\partial^3 p_i}{\partial t^3}\right]^{\mathrm{T}} \\ \boldsymbol{x_i^r} = \left[\theta_i, \omega_i = \frac{\partial \theta_i}{\partial t}, \alpha_i = \frac{\partial^2 \theta_i}{\partial t^2}, j_i^r = \frac{\partial^3 \theta_i}{\partial t^3}\right]^{\mathrm{T}}, \end{cases} \tag{10}$$

At time step i, p_i, v_i, a_i, and j_i^t represent the tray's position, translational velocity, translational acceleration, and translational jerk; θ_i, ω_i, α_i, and j_i^r represent the tray's rotation angle, rotational velocity, rotational angular acceleration, and rotational jerk. Through iteration, the state at the next time step can be expressed as:

$$\begin{cases} a_{i+1} = a_i + j_i^t t_s \\ v_{i+1} = v_i + a_i t_s + \frac{j_i^t t_s^2}{2} \\ p_{i+1} = p_i + v_i t_s + \frac{a_i t_s^2}{2} + \frac{j_i^t t_s^3}{6}, \\ \alpha_{i+1} = \alpha_i + j_i^r t_s \\ \omega_{i+1} = \omega_i + \alpha_i t_s + \frac{j_i^r t_s^2}{2} \\ \theta_{i+1} = \theta_i + \omega_i t_s + \frac{\alpha_i t_s^2}{2} + \frac{j_i^r t_s^3}{6}, \end{cases} \tag{11}$$

where t_s is the sampling time.

Based on the relationship between translational acceleration and rotational motion state described in Equ. (7) of the physical model, we found in our experiments that for the Franka Emika robot we used that the primary limitation for the transportation speed of the unstable object on the tray is the motion constraint of the tray rotation. To maximize the rotational speed of the tray, we choose S-curve to generate the rotational motion trajectory. The trajectory planning problem is now divided into two parts: first, generate the rotational motion trajectory of the tray based on the S-Curve principle, then calculate the translational motion trajectory based on the rotational motion state of the tray for each time step according to the physical modeling.

Trajectory planning for rotational motion involves selecting the jerk at each time step. Following the principle of the S-Curve, the goal is to maximize the jerk during the acceleration phase until the trajectory reaches the motion constraints or certain stop conditions, which are explained in detail below. For a third-order S-Curve, there are three constraints to consider. First, the jerk itself cannot exceed the jerk limit:

$$j_i^r \leq j_{rm} \tag{12}$$

Secondly, the acceleration has a limit:

$$\alpha_{i+1} \leq \alpha_{rm} \tag{13}$$

$$j_i^r \leq \frac{\alpha_{rm} - \alpha_i}{t_s} \tag{14}$$

Finally, since it takes some time for the rotational acceleration to decrease to zero, to ensure that the rotational velocity of the trajectory does not exceed the robot's motion limits, it is necessary to calculate the angular velocity at the moment when the rotational acceleration decreases to zero under the current state $\boldsymbol{x_i^r}$. This angular velocity is given by:

$$\omega_{peak}^{min} = \frac{\alpha_i^2}{2j_{rm}} + \omega_i \tag{15}$$

For each time step, to ensure that the rotational trajectory does not exceed the velocity limit, it needs to satisfy:

$$\omega_{peak}^{min} \leq \omega_{rm}. \tag{16}$$

Therefore, when selecting the jerk for the current time step, it is necessary to ensure that Equ. (16) is still satisfied for

the next time step, so that

$$\frac{(\alpha_i + j_i^r)^2}{2j_{rm}} \leq \omega_{rm} - (\omega_i + \alpha_i t_s + \frac{j_i^r t_s^2}{2}). \tag{17}$$

Algorithm 1 Sampling method

$m = 0, n = 0$
while True **do**
  Find maximum rotational jerk: $j_n^r \leftarrow \theta_n, \omega_n, \alpha_n$
  Calculate the next rotational motion state: $\boldsymbol{x_{n+1}^r} \leftarrow \boldsymbol{x_n^r}$
  Calculate the translational acceleration using Equ.(7): $a_{n+1} = \boldsymbol{x_{n+1}^r}$
  Calculate the translational motion state: $\boldsymbol{x_{n+1}^t} \leftarrow \boldsymbol{x_n^t}$
  $m = n + 1$
  while $\omega_j \geq 0$ **do**
    Find minimum rotational jerk: $j_j^r \leftarrow \theta_m, \omega_m, \alpha_m$
    Calculate the next rotational motion state: $\boldsymbol{x_{m+1}^r} \leftarrow \boldsymbol{x_m^r}$
    Calculate the translational acceleration using Equ.(7): $a_{m+1} = \boldsymbol{x_{m+1}^r}$
    Calculate the translational motion state: $\boldsymbol{x_{m+1}^t} \leftarrow \boldsymbol{x_m^t}$
    $m = m + 1$
  end while
  if $4mt_s v_m \geq p_t$ **then**
    Stop the iteration and return Case A.
  end if
  if $a_m \geq a_{max}$ **then**
    if $(mt_s + \frac{v_{max} - 2v_m}{2a_{max}})v_{max} \leq p_t$ **then**
      Stop the iteration and return Case D.
    else
      Stop the iteration and return Case B.
    end if
  end if
  if $v_m \geq \frac{v_{max}}{2}$ **then**
    Stop the iteration and return Case C.
  end if
  $n = n + 1$
end while

The acceleration phase of the tray's rotational motion trajectory is generated step by step, and the translational acceleration is calculated based on the rotational motion state and Equ. (7). At each time step, a similar iterative method as the rotational acceleration phase is used to solve for the deceleration phase of rotational motion. Translational acceleration is then solved step by step until the tray's rotational velocity decelerates to zero. The specific time step is expressed as t_a in Fig. 3. This process is illustrated in Alg. 1. Considering the translational motion constraints, based on different objects and motion constraints, the translational motion trajectory is divided into four different cases, as shown in Fig. 3.

Case A: The trajectory is depicted in Fig. 3 (a). Since the target distance is relatively close, to ensure that the tray motion does not overshoot the target point, the translational trajectory of the tray does not reach the maximum acceleration and maximum velocity limits. At step t_a, the rotational velocity decelerates to zero, and the tray translational motion velocity matches the trajectory's average velocity. Therefore, the conditions for the trajectory to fall into Case A is:

$$4t_a v_{t_a} \leq p_t \tag{18}$$

Case B: The trajectory is depicted in Fig. 3 (b). For case B, at time step t_a, the translational acceleration reaches the limits, but the maximum velocity does not. Therefore, the conditions for the trajectory to fall into Case B is:

$$\begin{cases} a_{t_a} \leq a_{max} \\ (t_a + \frac{v_{max} - 2v_{t_a}}{2a_{max}})v_{max} \geq p_t \end{cases} \tag{19}$$

Apart from the trajectories generated during the iteration process, it is also necessary to calculate the duration of the constant acceleration phase t_{ca}. Since there is no constant velocity phase, the average velocity of the entire trajectory is exactly the speed at $\frac{1}{2}t_{acc}$. Hence, it can be concluded that

$$\frac{1}{2}v_{top}(2t_{ca} + 4t_a) = p_t \tag{20}$$

$$v_{top} = 2v_{t_a} + a_{max}t_{ca} \tag{21}$$

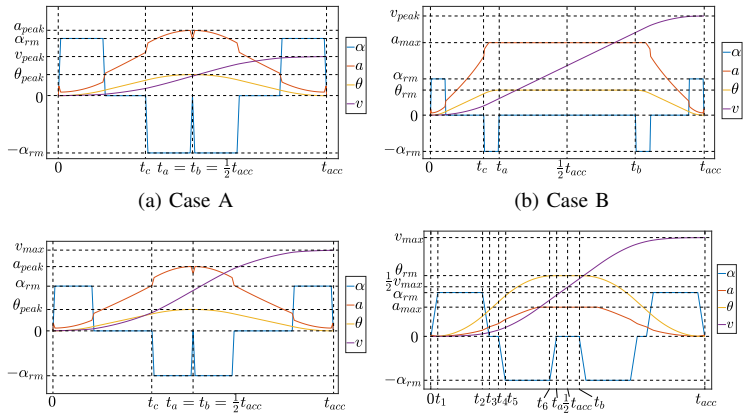


Fig. 3. Example trajectory of sampling-based method.

After organizing, it can be concluded that

$$0 = 4v_{t_a}t_a - p_t + 2(a_{max}t_a + v_{t_a})t_{ca} + a_{max}t_{ca}^2 \quad (22)$$

Case C: The trajectory is depicted in Fig. 3 (c). The maximum velocity of the translational motion reaches the limit, but the maximum acceleration does not. For case C, at time step t_{acc}, the translational velocity reaches the limits, and at time step t_a, the translational velocity is precisely half of the velocity constraint. Therefore, the conditions for the trajectory to fall into Case C is:

$$v_{t_a} \leq \frac{v_{max}}{2} \quad (23)$$

Apart from the trajectories generated during the iteration process, it is also necessary to calculate the duration of the constant velocity phase t_{cv}. Since the maximum speed can be achieved, the average velocity during the acceleration and deceleration phases of the translation motion is half of the maximum velocity limit. Therefore, it can be obtained:

$$\frac{1}{2}v_{max}4t_a + v_{max}t_{cv} = p_t \quad (24)$$

$$t_{cv} = \frac{p_t - 2v_{max}t_a}{v_{max}} \quad (25)$$

Case D: The trajectory is depicted in Fig. 3 (d). Both the maximum velocity and the maximum acceleration of the translational motion reach the limits. Therefore, the conditions for the trajectory to fall into Case D is:

$$\begin{cases} a_{t_a} \leq a_{max} \\ (t_a + \frac{v_{max} - 2v_{t_a}}{2a_{max}})v_{max} \geq p_t \end{cases} \quad (26)$$

The duration of the constant acceleration phase should be set so that the trajectory reaches the maximum velocity limit just after the acceleration phase. Therefore, it can be derived that:

$$2v_{t_a} + a_{max}t_{ca} = v_{max} \quad (27)$$

$$t_{ca} = \frac{v_{max} - 2v_{t_a}}{a_{max}} \quad (28)$$

Considering the symmetric acceleration changes during the acceleration phase, the average velocity during this phase should be exactly half of the maximum velocity limit. Therefore, it can be obtained:

$$v_{max}t_{cv} + \frac{1}{2}v_{max}(4t_a + 2t_{ca}) = p_t \quad (29)$$

$$t_{cv} = \frac{p_t - v_{max}(2t_a + t_{ca})}{v_{max}} \quad (30)$$

The motion state of the trajectory is shown in Tab. I.

1) Dynamic programming method: From another perspective, the time-optimal trajectory planning problem for non-prehensile object transportation can be regarded as an optimal control problem, where the state of the robot is defined as:

$$\boldsymbol{x}(t) = [p(t), v(t), a(t), \theta(t), \alpha(t), \omega(t)] \quad (31)$$

The initial state is defined as:

$$\boldsymbol{x}(t_0) = [\mathbf{0}_{1\times 6}] \quad (32)$$

The final state can be defined as:

$$\boldsymbol{x}(t_f) = [p(t_f) = p_t, else = 0] \quad (33)$$

Since the goal is to find the time-optimal trajectory, the objective function can be defined as:

$$J(x) = \int_{t_0}^{t_f} dt \quad (34)$$

The constraints of this optimal control problem can be divided into two parts. Firstly, the motion constraints of the tray:

$$\begin{cases} -v_{max} \leq v(t) \leq v_{max} \\ -a_{max} \leq a(t) \leq a_{max} \\ -\alpha_{max} \leq \alpha(t) \leq \alpha_{max} \\ -\omega_{max} \leq \omega(t) \leq \omega_{max} \end{cases} \quad (35)$$

Another part is ensuring that the tray object does not tip over. This can be expressed through Equ. (7), which gives the limits on acceleration:

$$\begin{cases} a(t) \leq \frac{\frac{I\alpha(t)}{m} + \frac{hg}{2}\sin\theta(t) + rg\cos\theta(t) - \frac{r\omega(t)^2 h}{2}}{\frac{h}{2}\cos\theta(t) - r\sin\theta(t)} \\ a(t) \geq \frac{\frac{I\alpha(t)}{m} + \frac{hg}{2}\sin\theta(t) - rg\cos\theta(t) + \frac{r\omega(t)^2 h}{2}}{\frac{h}{2}\cos\theta(t) + r\sin\theta(t)} \end{cases} \quad (36)$$

Due to the non-linearity and non-convexity of the physical model constraints, solving the optimal control problem directly is challenging. Therefore, we employ a dynamic programming method to solve this problem. By incorporating the robot's feasible region into the state space, we obtain a state space with four dimensions:

$$\boldsymbol{x}(t) = [p(t), v(t), \theta(t), \omega(t)] \quad (37)$$

	$t < t_1$	$t_1 \leq t < t_2$	$t_2 \leq t < t_3$	$t_3 \leq t < t_4$	$t_4 \leq t < t_5$	$t_5 \leq t < t_6$	$t_6 \leq t < t_a$	$t_a \leq t < \frac{1}{2}t_{acc}$
j_r	j_{rm}	0	$-j_{rm}$	0	$-j_{rm}$	0	j_{rm}	0
α	$\uparrow$	α_{rm}	$\downarrow$	0	$\downarrow$	$-\alpha_{rm}$	$\uparrow$	0
ω	$\uparrow$	$\uparrow$	$\uparrow$	ω_{rm}	$\downarrow$	$\downarrow$	$\downarrow$	0
φ	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	φ_{rm}
a	Calculated based on Equ. (9)							a_{max}
v	Increasing and reaching $\frac{1}{2}v_{max}$ at time $\frac{1}{2}t_{acc}$.							

TABLE I

MOTION STATE OF THE EXAMPLE TRAJECTORY.

To ensure trajectory accuracy, it's necessary to discretize each dimension with high precision. However, discretizing four dimensions leads to state space explosion, resulting in a state space that is too large to compute or store directly. Therefore, we simplified the problem. When testing the sampling-based method mentioned earlier, we found that the main factor affecting the trajectory speed was the time taken for the acceleration process. Here, we try to find the fastest trajectory to accelerate to the specified speed, as a near-time optimal solution. The advantage of doing this is that it removes the dimension of position, significantly reducing the size of the state space:

$$\boldsymbol{x'}(t) = [v(t), \theta(t), \omega(t)] \tag{38}$$

We compared the results with the sampling method to validate the time efficiency of the sampling method. We only need to start from the initial state and find the next reachable state at each time step. Once all the states reachable at the current time step are recorded, we proceed to find the next reachable states from these recorded states. We continue this process until the target state becomes reachable. The algorithm can be summarized in Alg. 2.

B. Method analysis

We compared the two proposed trajectory planning methods by attempting to solve the same transportation task to verify the time efficiency of the proposed methods. The trajectories generated by the two methods are shown in Fig. 4. It shows that when accelerating to the same target velocity ($v_{target} = 0.1\ m/s$) with the same constraints, the trajectories generated by both methods are almost identical, verifying the time efficiency of using an S-curve as the rotational motion trajectory in our sampling-based method.

In terms of time complexity analysis of the two methods, in dynamic programming, as each step requires searching through all reachable states, the worst-case complexity for each step is the square of the state space size $\mathcal{O}(n^2)$. However, the sampling method only needs to iteratively solve until the rotational axis decelerates to zero. Each iteration only involves constant-level mathematical operations thus, the time complexity is $\mathcal{O}(d)$. d is the time step required to achieve maximum velocity. In our experiment, for the acceleration limit is a_{max}, the velocity limit is v_{max}, and the time interval between each step is t_i, the approximate value of d can be calculated as follows:

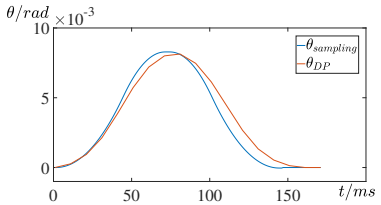
$$d \approx \frac{2v_{max}}{a_{max}t_i} \tag{39}$$

This makes it possible to implement online trajectory planning based on the sampling method in the future.

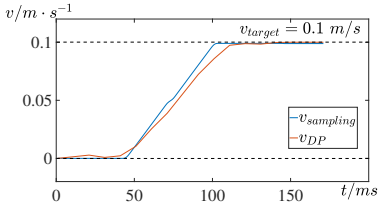
Interestingly, near-time-optimal trajectories from a stationary state to all valid states in the discrete state space are computed by calculating acceleration trajectories using the dynamic programming method. By recording these trajectories, we obtain a dataset of near-time-optimal acceleration trajectories starting from a stationary state. We could potentially enable rapid planning of transport trajectories by training a neural network.

Algorithm 2 Point-to-point trajectory planning

```
Initialization: State(x = x_{t_0}) ← x_{t_0}
Setting other state to unreachable: State(x ≠ x_{t_0}) ← 0_{1×3}
Put initial state into the queue: Q_c.push(x_{t_0})
while t ≤ t_max do
  while Q_c not empty do
    Get state need to be checked from queue: x_c ← Q_c.front()
    Search for the state reachable in the next time step: X ← f(x_c)
    while x ∈ X do
      if x meets the stability constraint then
        if x = x_{t_f} then
          Finish computation.
        else
          Put this state into the queue: Q_n.push(x)
          Record the trajectory: State(x) ← x_c
        end if
      end if
    end while
    Remove checked state from the queue: Q_c.pop()
  end while
  t = t + 1
  Move to next time step: Q_c ← Q_n
end while
if State(x_{t_f}) = 0 then
  Return computation failure.
else
  Return trajectory from target state: x_l ← x_{t_f}
  while x_l ≠ x_{t_0} do
    Output current state: x_l
    Go to find the last state: x_l ← State(x_l)
  end while
end if
```



(a) rotation angle



(b) translation velocity

Fig. 4. The acceleration phase trajectory comparison.

V. Experiment and Results

A. Experimental Setup

The following experiments are conducted to assess the reliability and time efficiency of the sampling-based method we proposed. The experiment setup is shown in Fig. 5. Experiments are conducted with a 7-DoF Franka Emika robot [20]. The robot is controlled by a computer (Intel Core i5-12600K CPU @ 4.50GHz) using the Franka Control Interface (FCI) at 1 kHz. In the experiments, the robot's motion trajectories planned in Cartesian space are transformed into joint-space trajectories through analytical inverse kinematics calculations [21]. We employ the internal joint impedance controller. The stiffness $\boldsymbol{K}_\theta$ and damping $\boldsymbol{D}_\theta$ parameters of the impedance control are set to $diag[3000, 3000, 3000, 2500, 2500, 2000, 2000]N \cdot m/rad$ and $diag[75, 75, 75, 70, 70, 60, 60]N \cdot m \cdot s/rad$ respectively. The following conditions are selected as motion constraints in the experiment:$j_{max} = 6500m/s^3$, $a_{max} = 13m/s^2$, $v_{max} = 0.6m/s$, $j_{rm} = 4500rad/s^3$, $\alpha_{rm} = 9rad/s^2$, $\omega_{rm} = 2.5rad/s$. For the dynamic programming method, we divided the translational velocity range from zero to the maximum velocity limit into 4000 equal parts, and we used the tray rotation angle reference from the sampling method trajectory. The maximum rotation angle was set to $\theta_{max} = 0.2\ rad$ and divided into 750 equal parts. Additionally, we divided the tray rotation angular velocity range from $-\omega_{max}$ to ω_{max} into 150 equal parts. The target displacement distance is $p_t = 0.4m$.

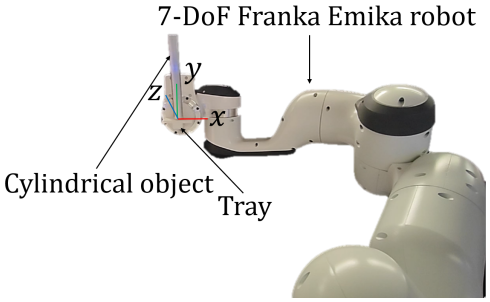


Fig. 5. Experiment setup.

To assess the actual stability of target objects, we conducted an experiment where each object was placed on a tilted plane. The tilt angle was gradually increased until the object toppled. This method provides a practical measure of stability, which is then used to inform pressure center positioning during trajectory planning. Three uniformly sized aluminum cylinders with varying dimensions were tested. Their parameters are summarized in Table II, where θ_m denotes the measured maximum stable angle and r_0^t represents the theoretically calculated pressure center position based on these experimental results.

To account for practical limitations during object transportation, the pressure center position in trajectory planning adopts a more conservative approach compared to the theoretical value. This adjustment considers potential control errors in robot motion, external factors like air resistance, and the non-ideal rigidity of the object (a simplification made in the physical model). This ensures a higher probability of maintaining object stability throughout the process.

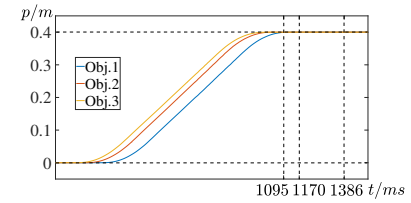
	Obj.1	Obj.2	Obj.3
h	$20cm$	$18cm$	$16cm$
r	$0.75cm$	$1cm$	$1cm$
θ_m	$0.060rad$	$0.095rad$	$0.108rad$
r_0^t	$0.601cm$	$0.858cm$	$0.867cm$

TABLE II

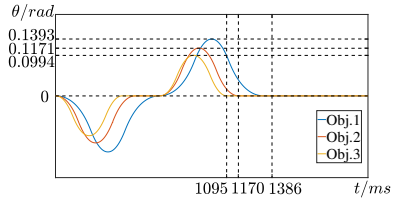
The parameters of the tested objects.

B. Experimental Result

The effectiveness of our method was validated experimentally using different objects. Fig. 6 showcases the successfully transported trajectories of three distinct objects. We explored various pressure center positions during experimentation, ultimately identifying settings that ensured stable transportation for all three targets. Interestingly, the results demonstrate that more aggressive pressure center placements can be attempted for inherently more stable objects. Conversely, various error sources not accounted for in the physical model become more influential for less stable objects.



(a) translation trajectory



(b) rotation trajectory

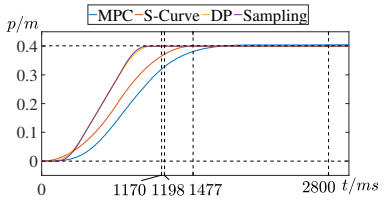
Fig. 6. The trajectories obtained by testing three target objects using our proposed sampling method, with the pressure centre set to $Obj.1 : r_0 = 3mm$, $Obj.2 : r_0 = 6mm$, $Obj.3 : r_0 = 7.5mm$

To assess the performance of our proposed method, we compared it with two baseline methods: the MPC method introduced by Heins et al. [13] and the S-curve-based method proposed by Acharya et al. [16] which lacks a rotating axis. Following parameter tuning focused on achieving rapid transportation, both baseline methods were tasked with executing the transportation scenario identical to our proposed method. It is important to acknowledge that the baseline trajectories might not represent the absolute optimal solution. While we optimized both baseline methods for speed, achieving perfect optimization through parameter tuning is computationally expensive and time-consuming, specifically for the MPC method. In practical scenarios, this becomes a limiting factor.

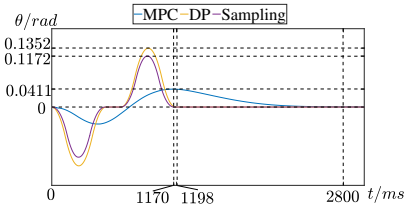
In the experiment, we measured the trajectory generation time for different methods. As mentioned in the method

analysis, the sampling-based method has the lowest time complexity for trajectory planning, with a generation time of $1.018s$. In contrast, planning the same trajectory using the dynamic programming method took 12 hours and 32 minutes.

Fig. 7 illustrates the transportation trajectories generated by our proposed method alongside those of the two baseline methods. Compared to the MPC and S-curve methods, our proposed sampling method increased the trajectory speed by 59.0% and 20.7%, respectively. The proposed dynamical programming method achieved similar time efficiency. While our proposed methods and the S-curve method leverage trajectory planning, our method incorporates tray rotation to achieve faster transportation. In contrast, the MPC method, like our proposed methods, utilizes tray rotation to improve efficiency. However, the inherent complexity (non-convexity) of the optimal control problem and the conservative design employed by the MPC controller for robust robot control lead to significantly slower trajectories compared to ours. Our proposed trajectory planning method offers a key advantage: it allows complete control over the pressure center position during the planning stage. This enables us to generate trajectories closer to the theoretical time-optimal solution.



(a) translation trajectory



(b) rotation trajectory

Fig. 7. The trajectories of the three methods, with Obj.2 as the target object. In the sampling method, the pressure centre is set to $r_0 = 6mm$. The figure does not depict the rotation trajectory for the S-curve method, as the tray remains horizontal.

VI. Conclusion

In this paper, we propose two novel trajectory planning methods for non-prehensile unstable object transportation. By comparing the accelerated trajectory of the sampling method with that of the dynamic programming method, we demonstrate that the sampling method significantly simplifies the trajectory planning process while ensuring that the trajectories are near time-optimal. We also validate our proposed methods through experiments and compare them with recent similar studies, demonstrating that our trajectory planning methods enhance the transportation speed of unstable objects compared to MPC-based control and S-curve planning methods. Currently, the method we proposed is only applicable to transporting objects along a straight line. In future work, we will explore more complex transportation trajectories. Additionally, by further refining the physical model, we aim to achieve rapid and stable non-prehensile transportation of objects with more complex shapes.

References

[1] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1711–1718, 2018.

[2] M. T. Mason and K. M. Lynch, "Dynamic manipulation," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 1. IEEE, 1993, pp. 152–159.

[3] T. Senoo, A. Namiki, and M. Ishikawa, "Ball control in high-speed batting motion using hybrid trajectory generator," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1762–1767.

[4] G. Zhang, S. Ma, and Y. Li, "Nonprehensile pushing manipulation strategies for a multi-limb robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[5] Y. Aiyama, M. Inaba, and H. Inoue, "Pivoting: A new method of graspless manipulation of object by robot fingers," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 1. IEEE, 1993, pp. 136–143.

[6] A. Donaire, F. Ruggiero, L. R. Buonocore, V. Lippiello, and B. Siciliano, "Passivity-based control for a rolling-balancing system: The nonprehensile disk-on-disk," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2135–2142, 2016.

[7] K. Ryu and Y. Oh, "Balance control of ball-beam system using redundant manipulator," in *2011 IEEE International Conference on Mechatronics*. IEEE, 2011, pp. 403–408.

[8] G. Bätz, A. Yaqub, H. Wu, K. Kühnlenz, D. Wollherr, and M. Buss, "Dynamic manipulation: Nonprehensile ball catching," in *18th Mediterranean Conference on Control and Automation, MED'10*. IEEE, 2010, pp. 365–370.

[9] A. Pekarovskiy, F. Stockmann, M. Okada, and M. Buss, "Hierarchical robustness approach for nonprehensile catching of rigid objects," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3649–3654.

[10] C. Zhou, Y. Long, Y. Cao, L. Zhao, B. Huang, and Y. Zheng, "Optimal nonprehensile interception strategy for objects in flight," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3130–3136.

[11] L. Chen, H. Yu, A. Naceri, A. Swikir, and S. Haddadin, "Time-optimized trajectory planning for non-prehensile object transportation in 3d," 2024. [Online]. Available: https://arxiv.org/abs/2408.16420

[12] M. Selvaggio, A. Garg, F. Ruggiero, G. Oriolo, and B. Siciliano, "Non-prehensile object transportation via model predictive non-sliding manipulation control," *IEEE Transactions on Control Systems Technology*, 2023.

[13] A. Heins and A. P. Schoellig, "Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator," *IEEE Robotics and Automation Letters*, 2023.

[14] R. Subburaman, M. Selvaggio, and F. Ruggiero, "A non-prehensile object transportation framework with adaptive tilting based on quadratic programming," *IEEE Robotics and Automation Letters*, 2023.

[15] G. Martucci, J. Bimbo, D. Prattichizzo, and M. Malvezzi, "Maintaining stable grasps during highly dynamic robot trajectories," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9198–9204.

[16] P. Acharya, K.-D. Nguyen, H. M. La, D. Liu, and I.-M. Chen, "Nonprehensile manipulation: a trajectory-planning perspective," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 1, pp. 527–538, 2020.

[17] P. Lertkultanon and Q.-C. Pham, "Dynamic non-prehensile object transportation," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2014, pp. 1392–1397.

[18] R. I. C. Muchacho, R. Laha, L. F. Figueredo, and S. Haddadin, "A solution to slosh-free robot trajectory optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 223–230.

[19] C. Bouchard, M. Nesme, M. Tournier, B. Wang, F. Faure, and P. Kry, "6d frictional contact for rigid bodies," in *Graphics Interface*, 2015.

[20] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jaehne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, 2022.

[21] Y. He and S. Liu, "Analytical inverse kinematics for Franka Emika Panda – a geometrical solver for 7-DOF manipulators with unconventional design," in *2021 9th International Conference on Control, Mechatronics and Automation (ICCMA2021)*. IEEE, Nov. 2021.