

Nonprehensile Manipulation: A Trajectory-Planning Perspective

Praneel Acharya , Kim-Doang Nguyen , Hung M. La , *Senior Member, IEEE*,
 Dikai Liu , *Senior Member, IEEE*, and I-Ming Chen , *Fellow, IEEE*

Abstract—This article discusses nonprehensile manipulation of an asymmetric object using a robotic manipulator from a motion planning point of view. Four different aspects of the problem will be analyzed: object stability, motion planning, manipulator control, and experimental validation. Specifically, via an analysis of marginal stability of an object resting on a moving tray, the work establishes the critical accelerations of the manipulator's end-effector, below which the object's stability is guaranteed. These critical accelerations guide the design of the end-effector's motion for successful nonprehensile manipulation of the object. In particular, we propose two methods to formulate polynomial asymmetric s-curve trajectories such that the end-effector completes its motion in minimum time. In one method, the trajectory is divided into segments whose time intervals are then computed via a recursive algorithm. In the other method, we formulate an optimization problem and design the minimum-time trajectory by balancing the trade-off between the travel time and actuator effort. A series of experiments with a robotic arm is designed to validate and compare these motion planning methods in the context of nonprehensile manipulation. In addition, the experimental results demonstrate the advantages of the asymmetric s-curve motion profiles over the traditional symmetric s-curves.

Index Terms—Asymmetric s-curve motion profiles, minimum-time trajectories, nonprehensile manipulation, service robots.

Manuscript received August 15, 2019; revised January 20, 2020, August 18, 2020, and November 4, 2020; accepted November 7, 2020. Date of publication November 17, 2020; date of current version February 16, 2021. This work was supported by the FY21 Competitive Research Grant Program of South Dakota Board of Regents. Recommended by Technical Editor X. Jing and Senior Editor H. Qiao. (Corresponding author: Kim Doang Nguyen.)

Praneel Acharya and Kim-Doang Nguyen are with the Department of Mechanical Engineering, South Dakota State University, Brookings, SD 57007-2201 USA (e-mail: praneel.acharya@sdstate.edu; doang.nguyen@sdstate.edu).

Hung M. La is with the Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV 89557 USA (e-mail: hla@unr.edu).

Dikai Liu is with the Centre for Autonomous Systems, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: Dikai.Liu@uts.edu.au).

I-Ming Chen is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: MIChen@ntu.edu.sg).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2020.3038591

I. INTRODUCTION

OBJECT manipulation can be classified into two approaches: prehensile (grasping) and nonprehensile (without grasping). Nonprehensile manipulation is an intriguing approach because it is capable of manipulating multiple objects at a time as compared to grasping [1]. Recent developments on the topic include balancing a disk on a disk in [2] and [3], tossing and sliding soft objects in [4], and a robotic system that plays pool and snooker games [5]. In addition, a new passivity-based control framework for rolling manipulation was developed in [6]. Work in [7] demonstrates that via surface deformation, a soft robotic table can manipulate multiple objects. Though the topic has attracted a lot of attention lately, the research focus has been on the *control aspects* of nonprehensile balancing tasks. While controls may guarantee desirable manipulation given that certain stability conditions are satisfied, controllers require real-time feedback on the motion of the objects provided by vision systems or motion sensors attached to the objects.

In this article, we study nonprehensile balancing manipulation of an object from a *trajectory-planning perspective*. The work is motivated by applications where objects are exogenous entities to the robotic systems and monitoring their motions is not practical. Examples include a robot carrying a tray of food/drinks to serve at a table or a tray of medical tools to work in a human-robot team during a procedure. We are interested in objects that are not very stable and can tip over due to a small external force. Such an object is placed on top of a tray, which is attached to a manipulator's end-effector as depicted in Fig. 1. The tray on which the object is placed is constrained such that it cannot tilt. Tilting of the tray becomes important when the object needs to be thrown or rolled [8]–[10]. Moving an object without grasping from one location to another generally does not require reorienting it and tilting the tray involves extra DOF. Therefore, the scope of this article does not involve any tilting of the tray.

Four different aspects of this problem will be discussed, including object stability, motion planning, manipulator control, and experimental validation. In particular, we analyze the stability condition of an object with irregular geometry on a flat surface (a tray) in Section II. The analysis shows that for a specific object, there is a critical acceleration of the tray, below which the stability of the object on a tray is guaranteed. This leads to the selection of a particular type of motion profiles, namely the s-curve motion, for the robotic manipulator that allows easy



Fig. 1. Nonprehensile manipulation of interest: an object sitting on a tray moved by the end-effector of a robotic manipulator.

regulation of the peak accelerations that satisfy the stability condition. Furthermore, Section III presents two methods to design asymmetric s-curve motion profiles and demonstrate the advantage of asymmetric profiles over the traditional symmetric s-curve profiles in nonprehensile manipulation. Our design of the asymmetric s-curve motions is based on an optimization algorithm that guarantees minimum total travel time. The stability and motion planning framework is implemented and validated with a seven-axis robotic manipulator in Section IV.

The key scientific contributions of this article include the following.

- 1) The insights into nonprehensile manipulation from the motion planning point of view as opposed to the traditional feedback control point of view.
- 2) The analysis and quantification of the stability of an object sitting on a tray moved by a robotic manipulator.
- 3) A general algorithm that computes an asymmetric s-curve motion profile with minimum travel time.
- 4) An optimization scheme that also designs minimum-time asymmetric s-curve motions and takes into account the tradeoff between the travel time and actuator effort.
- 5) Experimental validation using a seven-DOF robotic arm. The experiments also illustrate the advantage of asymmetric s-curve motions over symmetric s-curves for nonprehensile object manipulation in minimum time.

II. OBJECT STABILITY

In this section, we use the term “stability” to represent an object’s ability to maintain its pose throughout the motion. As defined in [11], the object is considered to be in a stable state if it returns back to the initial state when external force applied is removed. As long as the line of gravity (the vertical line passing through the center of gravity) lies within the base of support, the object should remain stable. When an object is not stable, it may tip, slide, or lift in response to external forces. Tipping may occur before sliding or vice versa depending on the friction coefficient between the contact surfaces. Specifically, if the static friction is large enough, an object tends to tip over before sliding [12]. Therefore, objects that slide before tipping can be made to tip first instead of sliding by increasing the friction. In our setup, we assume that an object will tip over before sliding when the external force is large enough.

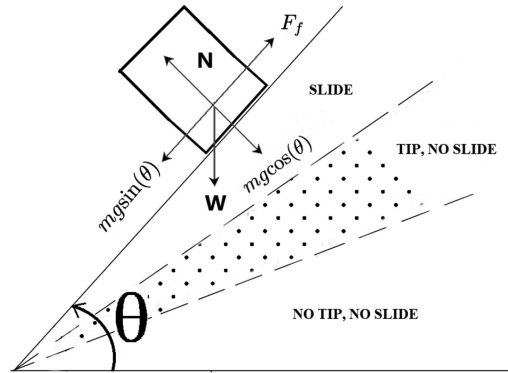


Fig. 2. Inclined-plane method used to compute the critical angles that represent the onsets of sliding and tipping. As inclination angle increases, the object responds differently depending upon which region θ is in.

A. Stability Index

In this section, we present a method to quantify object stability on a moving tray. In particular, the stability of an object in any given configuration is represented by the ratio of the contact area and the projected area of the object onto the contact surface. Let A_b be the contact area and A_p be the projected area of an object from top to the contact surface. A_p can be visualized as the area covered by an object when looking from the top view. Then, the stability index S can be defined as

$$S = \tanh(A_b/A_p). \quad (1)$$

Rigid bodies in any given configuration can have a contact surface as a point resulting in $A_b = 0$. The projected area A_p can be greater than or equal to contact area A_b , but it cannot be equal to zero for a rigid body. Besides, neither the projected area nor the contact area can be negative. Thus, the index S may have a value between 0 and 0.76159 inclusively.

For objects with $S \in (0, 0.76159]$, the line of gravity can be displaced by a certain angle θ and the objects will return to a stable pose when the external force is removed. Thus, for each of these objects, there exists a *stability-margin* angle θ^* such that if the line of gravity is further displaced, the object will tip over. In the next section, we discuss a way to experimentally determine this stability-margin angle for an arbitrary object with $S \in (0, 0.76159]$ without the need to measure its geometry and center of gravity.

B. Stability-Margin Angle

For an object with $S \in (0, 0.76159]$, computing the stability-margin angle for an object can be done via an inclined plane experiment. In particular, an object is placed on a plane inclined at an angle θ relative to the horizontal plane as shown in Fig. 2. We assume that the friction between the object and the plane is large enough, so tipping happens before sliding. The inclination angle is then raised slowly until the object tips over. The inclination angle of the plane at the onset of tipping is the stability-margin angle θ^* . If an external force is applied to keep the object in place even when the inclination angle is greater

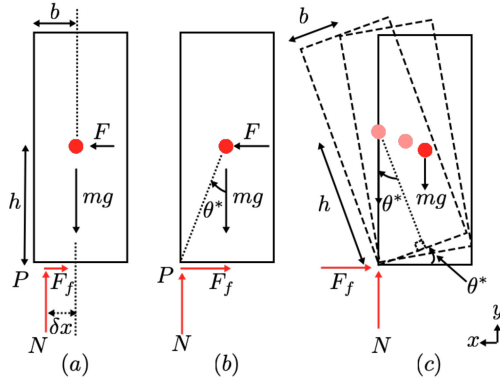


Fig. 3. Object on a surface under an external force: (a) and (b) Location of the net normal force N moves toward the tipping point P as external force F increases its magnitude. When the object is about to tip, the net normal force acts at the tipping point P as shown in (b). During tipping, the value of θ when the line of gravity passes the tipping point indicates the stability-margin angle θ^* as shown in (c).

than the stability margin, the object will start to slide as shown in Fig. 2. When the angle θ is less than the stability margin, the object is in static stability.

It is relatively straightforward to calculate the critical angle at which the object starts to slide. Specifically, simple inspection on the free body diagram in Fig. 2 leads to

$$\tan(\theta_{\text{slide}}^*) = \mu \quad (2)$$

where μ is the static friction coefficient. The stability-margin angle θ^* at which the object starts to tip over is not simple. Given the geometry of an object, careful calculations on the base of support, location of the center of mass, the projected area, etc., may result in a closed-form expression for θ^* . However, there is no general expression for θ^* that is applicable to all objects. Therefore, in our experiments, we determine θ^* for an object simply by increasing the inclination angle as depicted in Fig. 2 until the object tips over.

C. Critical Acceleration

In this section, we show that the stability-margin angle θ^* at which an object starts to tip is related to the maximum allowable acceleration of the tray below which the object remains stable throughout the motion. As mentioned earlier, for an object with $S \in (0, 0.76159]$, its geometry plays a critical role in determining if the object will tip over given a pose or an external force. For example, objects that are thin and tall tend to tip over in response to even a small external disturbance. As the applied external force F increases, the net normal force N moves toward the tipping point P as shown in Fig. 3(a) and (b). When an object is just about to tip, the normal force acts at the tipping point. Therefore, taking the moment about the tipping point P at the onset of tipping, we have

$$F = mg(b/h) \quad (3)$$

where F is the applied external force, m is the mass of the object, g is the gravitational acceleration, b is the distance from the line of gravity acting on the object to the tipping point, and h is the distance from the external force F to the tipping point. Equation

(3) results in a maximum value that if the external force is further increased, the object will tip over.

For an object sitting on a moving tray, the external force is indeed the inertial force due to the tray's acceleration a , i.e., $F = ma_{\text{tray}}$. As depicted in Fig. 3(c), object tipping occurs when the line of gravity is passing the tipping point P . Hence, we obtain $\tan(\theta^*) = b/h$. It then follows from (3) that the stability condition for an object sitting on a moving tray is

$$a_{\text{tray}} \leq g \tan(\theta^*). \quad (4)$$

If one wants to find the condition for the object not to slide, the same inequality as (4) can be used, but with θ^* replaced by θ_{slide}^* obtained in (2).

$$j = \begin{cases} J_{\text{peak}} \\ 0 \\ -J_{\text{peak}} \\ 0 \\ -J_{\text{peak}} \\ 0 \\ J_{\text{peak}} \end{cases} \quad (5)$$

$$a = \begin{cases} J_{\text{peak}} t \\ A_{1\text{peak}} \\ A_{1\text{peak}} - J_{\text{peak}} t \\ 0 \\ -J_{\text{peak}} t \\ -A_{2\text{peak}} \\ -A_{2\text{peak}} + J_{\text{peak}} t \end{cases} \quad (6)$$

$$v = \begin{cases} 0.5 \times J_{\text{peak}} t^2 & t_0 \leq t \leq t_1 \\ v_1 + A_{1\text{peak}} t & t_1 \leq t \leq t_2 \\ v_2 + A_{1\text{peak}} t - 0.5 J_{\text{peak}} t^2 & t_2 \leq t \leq t_3 \\ v_3 = V_{\text{peak}} & t_3 \leq t \leq t_4 \\ v_4 - 0.5 J_{\text{peak}} t^2 & t_4 \leq t \leq t_5 \\ v_5 - A_{2\text{peak}} t & t_5 \leq t \leq t_6 \\ v_6 - A_{2\text{peak}} t + 0.5 J_{\text{peak}} t^2 & t_6 \leq t \leq t_7 \end{cases} \quad (7)$$

The stability condition (4) is constructed with several assumptions and can only be used as an approximation for the tray's critical acceleration. In particular, as an object accelerates, drag force also comes into play and further decreases the critical acceleration, i.e., $a_{\text{tray}} \leq \tan(\theta^*)g - F_{\text{drag}}/m$. The effect of drag on an object can be prominent especially if the object is lighter. Besides, induced vibration while motion is carried out is also neglected. Due to the lack of feedback mechanism, any mathematical model that uses state information to accurately predict object tipping behavior is not applicable in our case. The advantage of the inclined plane method is that it provides a simple and convenient way to analyze the stability margin of an object and determine its critical accelerations. We will use this stability condition to approximate the critical accelerations of the tray and will validate the analysis via experiments with a robotic manipulator. The details of the experimental validation are coming up in Section IV. It is worth noting that while friction

coefficient is not modeled in this article, an intriguing way to experimentally determine friction coefficients is via iterative learning as illustrated in [13]. In the next section, we discuss how to design the motion of the tray to guarantee the stability condition (4) so that an object maintains a stable pose on the tray throughout the motion.

III. MOTION PLANNING

To this point, we have shown that for nonprehensile manipulation of an object using a flat tray, there is a stability-margin acceleration of the tray, below which the balance of the object on the tray is guaranteed. Hence, when designing the motion of the robot's end-effector, its peak acceleration throughout the motion is crucial. The selection of peak acceleration is based upon the critical acceleration discussed in Section II and the motion planning is based on s-curve motion profiles.

The majority of work on s-curve motions so far focuses on symmetric s-curves and how to implement them in different applications. Few papers investigated asymmetric s-curve motions including [14], which designed an asymmetric s-curve to replace the traditional two-step symmetric s-curve. In [15], four different shapes of asymmetric s-curve based on given constraints are formulated. In addition, the work in [16] developed an asymmetric s-curve motion command compatible with most commercial motion controllers. In general, when one plans machine motion for a task, minimum travel time is desired because this is usually corresponding to optimal productivity. However, the aforementioned work did not address the time-optimization problems. Most of the time-optimal algorithms to date have been developed for symmetric s-curve trajectories, for instance, the motion planners proposed in [17]–[19].

Motion planning was also studied through the lens of filter theories to generate smooth motion profiles. For example, research works in [20]–[22] implemented filtering algorithms to generate trajectory with a generic shape. These algorithms allow reference signals to be modified at any time during the execution. However, these studies rely on having a piecewise reference signal defined by the user as an input in addition to constraints on maximum values for kinematic features. Designing the combination of these splines for the reference trajectory can be challenging for users. In addition, these methods focus on a trajectory with a continuous position, velocity, and acceleration profiles, ignoring higher orders, such as jerks, snaps, etc. They also emphasize the theories of motion planning and filtering without considering nonprehensile manipulation.

To fill these gaps, we not only do aim to develop methods for designing asymmetric motion for nonprehensile manipulation, but also want to minimize the travel time. Additionally, we want the algorithm to be general enough such that a higher order asymmetric s-curve can be generated. Furthermore, we will design experiments to show the advantage of asymmetric s-curves over symmetric ones.

A. Basics of S-Curve Motion Profiles

In this section, we will use a third-order s-curve trajectory to discuss the basics of an s-curve motion profile. Such a

motion profile is composed of third-order polynomials with seven segments, each of which is defined within a time interval. In particular, $T_{i(i+1)}$ describes the time segment $[t_i, t_{i+1}]$, for $i = 0, \dots, 6$. A general third-order s-curve trajectory with seven time intervals can be divided into two different sections: $[t_0, \tilde{t}_3]$ and $[t_4, \tilde{t}_7]$. In a symmetric s-curve, the values of the peak accelerations in both sections are identical. In contrast, the peak accelerations in the two sections of an asymmetric s-curve are different. The peak acceleration in the first section will be represented by $A_{1\text{peak}}$, whereas that in the second section will be represented by $A_{2\text{peak}}$.

To fully define all kinematic features of an asymmetric s-curve, quantities such as J_{peak} , $A_{1\text{peak}}$, $A_{2\text{peak}}$, V_{peak} , S_{peak} , T_{01} , T_{12} , T_{23} , T_{34} , T_{45} , T_{56} , and T_{67} must be known. We constrain $T_{01} = T_{23}$ and $T_{45} = T_{67}$. With these constraints, the accelerations at t_3 and t_7 are zero. This allows to move an object at rest from one point to another. The selection of $A_{1\text{peak}}$ and $A_{2\text{peak}}$ is motivated by the critical acceleration discussed in Section II. In practice, critical acceleration is a best estimated acceleration such that objects maintain stability during motion when there is no feedback available. As discussed in Section II, there are uncertainties, such as aerodynamic drag, trajectory execution error, and surface nonuniform roughness, that affect the accuracy of critical acceleration computation.

In general, symmetric objects have one critical acceleration, in that case, $A_{1\text{peak}}$ will be equal to $A_{2\text{peak}}$. Asymmetric objects generally have two critical acceleration in a horizontal motion, i.e., $A_{1\text{peak}}$ and $A_{2\text{peak}}$ will be different. Once $A_{1\text{peak}}$ and $A_{2\text{peak}}$ are obtained using (4), the remaining kinematic parameters will be user's inputs as they are task-specific. For example, total distance to cover is reflected through S_{peak} , which can be limited by manipulator workspace, the maximum velocity the robot can move are reflected through V_{peak} , and so on. Thus, the user has the flexibility of choosing J_{peak} , V_{peak} , and S_{peak} . With the known peak values, the time intervals T_{01} , T_{12} , T_{34} , T_{45} , and T_{56} are to be computed based upon certain algorithms. See [23] for an example of such algorithms for a symmetric s-curve. Our focus is on an asymmetric s-curve described by (5), (6), (7), and the following equation:

$$p = \begin{cases} (1/6)J_{\text{peak}}t^3 & t_0 \leq t \leq t_1 \\ p_1 + v_1t + 0.5A_{1\text{peak}}t^2 & t_1 \leq t \leq t_2 \\ p_2 + v_2t + 0.5A_{1\text{peak}}t^2 - z & t_2 \leq t \leq t_3 \\ p_3 + v_3t & t_3 \leq t \leq t_4 \\ p_4 + v_4t - (1/6)J_{\text{peak}}t^3 & t_4 \leq t \leq t_5 \\ p_5 + v_5t - 0.5A_{2\text{peak}}t^2 & t_5 \leq t \leq t_6 \\ p_6 + v_6t - 0.5A_{2\text{peak}}t^2 + z & t_6 \leq t \leq t_7. \end{cases} \quad (8)$$

$$z = (1/6)J_{\text{peak}}t^3.$$

We will present two different methods with the goal to obtain minimum-time asymmetric s-curve: one based on a recursive algorithm and the other based on an optimization algorithm.

In most practical applications, the jerk peaks need to be restricted below a reasonably low value. Once jerk and acceleration peak values are set, in order to obtain a minimum-time motion profile, the acceleration needs to be increased as quickly

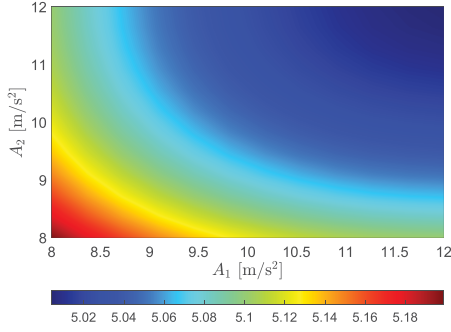


Fig. 4. Effect of A_1 and A_2 on total time: as A_1 or A_2 or both increase, the total time decreases while the peak jerk, velocity, and position are fixed.

as possible to the peak value while satisfying the jerk constraint. Fig. 4 illustrates the effect of the peak accelerations on the total travel time. The quantity A_1 represents the peak acceleration of the first section and A_2 represents the peak acceleration of the second section. For clarification, $A_{1\text{peak}}$ and $A_{2\text{peak}}$ are desired peak accelerations, which are usually provided by the users depending on the application specifications. On the other hand, A_1 and A_2 are the peak accelerations of the s-curve motion profile. Depending how the profile is designed, A_1 and A_2 may or may not reach $A_{1\text{peak}}$ and $A_{2\text{peak}}$, respectively. For a given set of A_1 and A_2 values, the total time is computed and displayed by a coloring scheme in Fig 4. We can see that an increase in the peak accelerations in both sections leads to a decrease in total time.

B. Divide and Conquer Approach

From the above observations, for given inputs of J_{peak} , $A_{1\text{peak}}$, $A_{2\text{peak}}$, V_{peak} , and S_{peak} , a minimum time trajectory is achieved when $A_1 = A_{1\text{peak}}$ and $A_2 = A_{2\text{peak}}$. This leads to five inputs (or five constraint equations) and five unknowns: T_{01} , T_{12} , T_{34} , T_{45} , and T_{56} . The unique motion profile obtained from these specifications results in a minimum-time trajectory.

One approach is to analyze each segment separately to calculate the time intervals. Due to the key property of analyzing each segment separately instead of analyzing the complete profile, this method will be referred to as the “divide and conquer” approach. In particular, as shown in (5) and (6), the interval T_{01} can be found by $T_{01} = A_{1\text{peak}}/J_{\text{peak}}$. Similarly, T_{45} can be found using relation $T_{45} = A_{2\text{peak}}/J_{\text{peak}}$. Then, it follows from (6) and (7) that

$$\int_0^{T_{01}} A_1 dt + \int_0^{T_{12}} A_1 dt - V_{\text{peak}} = 0. \quad (9)$$

With T_{01} known, we use (9) to compute T_{12} . Furthermore, also from (6) and (7), we have

$$V_{\text{peak}} - \int_0^{T_{45}} A_2 dt - \int_0^{T_{56}} A_2 dt = 0. \quad (10)$$

By substituting T_{45} in (10), T_{56} is computed. Finally, T_{34} is calculated by integrating the velocity profile to obtain

$$S_{\text{peak}} - \int_0^{t_7} v dt = 0 \quad (11)$$

with v substituted from (7).

To this point, we have shown step by step how to construct a third-order asymmetric s-curve. Next, we generalize the algorithm to design an asymmetric s-curve of the n th order.

The general algorithm for the n th-order trajectories: Since the s-curve of interest is asymmetric, if one looks at the k th layer, for $k = 2, \dots, n-1$ (e.g., acceleration in a third-order s-curve and acceleration and jerk in a fourth-order s-curve), the corresponding peak values of the two sections are different. Therefore, looking at the k th layer, for $k = 2, \dots, n-1$, we denote by $M_{1\text{peak}}^k$ the peak value of the first segment and $M_{2\text{peak}}^k$ the peak value of the second segment. In addition, let M_{peak}^0 refer to peak position value, M_{peak}^1 refer to peak velocity value, and M_{peak}^n refer to peak value of the n th layer. All these peak values are the inputs usually provided to the algorithm by a user. Refer to the beginning of this section for an example of inputs for a third-order s-curve. For a fifth-order trajectory, the inputs would be: M_{peak}^0 , $M_{1\text{peak}}^1$, $M_{2\text{peak}}^1$, $M_{1\text{peak}}^2$, $M_{2\text{peak}}^2$, $M_{1\text{peak}}^3$, $M_{2\text{peak}}^3$, $M_{1\text{peak}}^4$, $M_{2\text{peak}}^4$, and M_{peak}^5 .

Algorithm 1:

```

segments = 2^n - 1;
mid = 2^n / 2;
for i = 1 : (n - 1) do
    s = i - 1;
    for j = i : mid : segments do
        if j < mid then
            l = sum_{k=0}^{i-1} 2^k;
            Solve for T_{2^{s-1}, 2^s} as defined below:
                \int_{t_0}^{t_l} m_{0,l}^{n-s}(t) dt - M_{1\text{peak}}^{n-i} = 0 \quad (12)
        else
            l = mid + sum_{k=0}^{i-1} 2^k;
            Solve for T_{mid-1+2^s, mid+2^s} as defined below:
                \int_{t_{\text{mid}}}^{t_l} m_{\text{mid},l}^{n-s}(t) dt + M_{2\text{peak}}^{n-i} = 0 \quad (13)
        where \int_{t_b}^{t_c} m_{b,c}^{n-s}(t) dt is computed using equation
        (19) and M_{2\text{peak}}^1 = 0.
    end
end

```

Solve for $T_{\text{mid}-1, \text{mid}}$ using following equation:

$$\int_{t_0}^{t_{\text{segments}}} m_{0, \text{segments}}^1(t) dt - M_{\text{peak}}^0 = 0 \quad (14)$$

The above third-order calculations are generalized as shown in Algorithm 1 for an n th order asymmetric s-curve. Here, the naming scheme is updated as follows: M_{peak}^0 refers to peak position value, M_{peak}^1 refers to peak velocity value, M_{peak}^2 refers to peak acceleration value, and so on up to M_{peak}^n . The sets of equations describing different segments of the n th layer

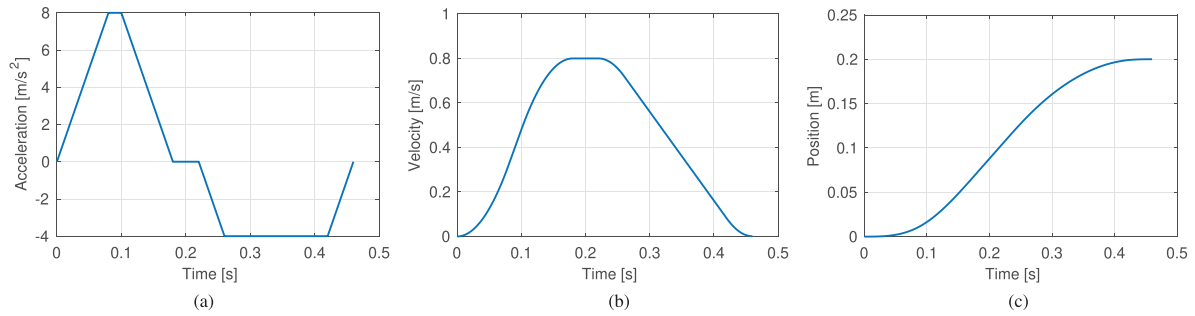


Fig. 5. Asymmetric s-curve trajectory with its velocity and acceleration profiles generated by the *divide and conquer method*. (a) Acceleration over time. (b) Velocity over time. (c) Position over time.

$(m_{0,1}^n, m_{1,2}^n, \dots, m_{2^{n-2}, 2^{n-1}}^n)$ can be obtained using similar approach as shown in (5). Note here that the first layer (m_*^0) refers to position, the second layer (m_*^1) refers to velocity, the third layer (m_*^2) refers to acceleration, and so on. As an example, for a third-order s-curve, $m_{0,1}^3 = M_{\text{peak}}^n$, $m_{1,2}^3 = 0$, $m_{2,3}^3 = -M_{\text{peak}}^n$, and so on. Once $m_{0,1}^n, m_{1,2}^n, \dots, m_{2^{n-2}, 2^{n-1}}^n$ are fully defined, $m_{0,1}^{n-1}$ and $m_{0,1}^{n-2}$ can be obtained as follows:

$$m_{0,1}^{n-1}(t) = \int m_{0,1}^n(t) dt, \quad t_0 \leq t \leq t_1 \quad (15)$$

$$m_{0,1}^{n-2}(t) = \int m_{0,1}^{n-1}(t) dt, \quad t_0 \leq t \leq t_1. \quad (16)$$

In general

$$m_{0,1}^{n-k}(t) = \int m_{0,1}^{n-k+1}(t) dt, \quad t_0 \leq t \leq t_1 \quad (17)$$

where $k = 0, \dots, n$ and $m_{0,1}^{n-k}$ describes only the first segment of the $(n-k)$ th layer. For a general segment of the $(n-k)$ th layer, we use the following recursive formula:

$$m_{b,b+1}^{n-k}(t) = m_{b-1,b}^{n-k}(t_b) + \int m_{b,b+1}^{n-k+1}(t) dt, \quad t_b \leq t \leq t_{b+1}. \quad (18)$$

Equation (18) is the recursive expression used to establish any polynomial segments in any layer of a general asymmetric s-curve. In addition, since every segment is properly defined, one may calculate the area under the curve of any time period within the trajectory by simply summing individual areas within the period, e.g., for the period from t_c to t_b

$$\int_{t_b}^{t_c} m_{b,c}^{n-s} dt = \sum_{p=b}^{c-1} \int_{t_p}^{t_{p+1}} m_{p,p+1}^{n-s}(t) dt. \quad (19)$$

Hence, by summing up the area under the curve in a relevant period, one may obtain the peak value of a layer (e.g., the peak velocity) and then set up the corresponding constraint equation as shown in (12), (13), and (14). Algorithm 1 summarizes the pseudocodes that enable the computation of a general n th order s-curve trajectory with minimal travel time.

Though this approach is a simple and quick way to plan an s-curve trajectory, it requires reasonable inputs be provided. For example, with $J_{\text{peak}} = 100 \text{ m/s}^3$, $A_{1\text{peak}} = 8 \text{ m/s}^2$, $A_{2\text{peak}} = 4 \text{ m/s}^2$, $V_{\text{peak}} = 0.8 \text{ m/s}$, and $S_{\text{peak}} = 0.2 \text{ m}$, the algorithm is able to calculate minimum time trajectory as shown in Fig. 5. As

the method works by analyzing each segment separately, the influence of a segment is not seen in the another segment, e.g., T_{45} and T_{56} are obtained without considering the effect of T_{34} . As a result, the algorithm may return a *negative* time segment and fail to calculate the trajectory. For instance, consider another set of inputs: $J_{\text{peak}} = 100 \text{ m/s}^3$, $A_{1\text{peak}} = 10 \text{ m/s}^2$, $A_{2\text{peak}} = 15 \text{ m/s}^2$, $V_{\text{peak}} = 0.8 \text{ m/s}$, and $S_{\text{peak}} = 0.2 \text{ m}$. Solving for these constrains would lead to an error since all given constraints could not be satisfied with positive values of time intervals.

To overcome this limitation, in the next section, we will formulate the trajectory planning as an optimization problem. We will then use an optimization algorithm that allows for selecting suitable values for $A_1 \leq A_{1\text{peak}}$ and $A_2 \leq A_{2\text{peak}}$ that satisfy all the constraints. We will also demonstrate that the optimization approach is able to plan a minimum-time trajectory where the divide and conquer approach fails.

C. Optimization Approach

We first set up an optimization problem to plan a minimum-time third-order s-curve trajectory with inputs: J_{peak} , $A_{1\text{peak}}$, $A_{2\text{peak}}$, V_{peak} , and S_{peak} . In the previous section, we directly set $A_1 = A_{1\text{peak}}$ and $A_2 = A_{2\text{peak}}$, then calculate the time intervals. In contrast, in the optimization approach here, A_1 and A_2 are variable and parts of what the algorithm solves for. However, they have to be no greater than the user provided peak accelerations $A_{1\text{peak}}$ and $A_{2\text{peak}}$. In summary, for a third-order trajectory, we have seven variables, five equality constraints, and two inequality constraints

$$\begin{cases} J_{\text{peak}} - A_1/T_{01} = 0 \\ J_{\text{peak}} - A_2/T_{45} = 0 \\ A_1 \leq A_{1\text{peak}} \\ A_2 \leq A_{2\text{peak}} \\ V_{\text{peak}} - 2 \int_0^{T_{01}} A_1 dt - \int_0^{T_{12}} A_1 dt = 0 \\ V_{\text{peak}} - 2 \int_0^{T_{45}} A_1 dt - \int_0^{T_{56}} A_2 dt = 0 \\ S_{\text{peak}} - p_7 = 0. \end{cases} \quad (20)$$

Though the constraints in (20) are designed with a single peak value for the n th profile for simplicity, the formulation facilitates multiple different peak values for the highest order template. For example, if we are designing third-order s-curve, we can have $J_{1\text{peak}}$ and $J_{2\text{peak}}$ instead of single J_{peak} . Constraints can also be accordingly updated. As example, given two peak values of

jerk, top equations in (22) of our manuscript can be updated as follows: $J_{1\text{peak}} - A_1/T_{01} = 0$ and $J_{2\text{peak}} - A_2/T_{45} = 0$. Once we have defined all constraints as in (20), to deal with the inequality constraints, slack variables are used to transform them into equality constraints. Hence, the optimization includes nine variables, namely the two slack variables, A_1 , A_2 , T_{01} , T_{12} , T_{34} , T_{45} , and T_{56} , and seven equality constraints described in (20). The two DOFs allow the algorithm to search for a solution that optimizes the following objective function:

$$\min f = |T_{01}| + |T_{12}| + |T_{34}| + |T_{45}| + |T_{56}|. \quad (21)$$

Once the objective function and constraints are set up, one may develop a simple solver or has a choice among many off-the-shelf solvers to solve the optimization problem. In our case, we employ the solver ‘‘FMINCON’’ available in the MATLAB optimization toolbox to solve the optimization problem. The initial condition plays an important role in whether the optimization algorithm will converge or not. As discussed earlier, a minimum-time trajectory occurs at the desired peak accelerations. Hence, the initial conditions are set to $x_0 = [A_{1\text{peak}}, A_{2\text{peak}}, 0.1, 0.1, 0.1, 0.1, 0.1]$, in which the first two elements are the desired peak accelerations of the trajectory, and the other elements are the initial values for the time intervals. All trajectories tested with the optimization approach use this initial condition. For every tested trajectory, above-mentioned initial conditions x_0 lead to a time minimum solutions. Furthermore, we use the results demonstrated in [18] for comparison to further investigate the claim that the trajectory obtained by our proposed method yields a minimum time trajectory. Work in [18] found optimal time trajectory for symmetric s-curve for following inputs: $J_{\text{peak}} = 1000 \text{ m/s}^3$, $A_{1\text{peak}} = A_{2\text{peak}} \leq 6 \text{ m/s}^2$, $V_{\text{peak}} = 20 \text{ mm/s}$, and $S_{\text{peak}} = 0.4 \text{ mm}$. Total time is not directly stated in [18] but based upon the given graphical results (see [18, Fig. 2]), the total time of trajectory is between 0.028 and 0.030 s. Applying our algorithm, we were able to obtain a total time of 0.028 s.

It is observed from Fig. 4 that it takes 5.004 s to complete the motion while accelerating at 12 m/s^2 as compared to 5.027 s while accelerating at 10.55 m/s^2 . In other words, the acceleration needs to be increased by 1.45 m/s^2 (about 12%) just to shorten the total time by 0.023 s (about 0.46%). In most applications, 0.023 s faster is not worth the extra energy consumed to increase the acceleration. Therefore, a tradeoff between actuator effort and total time may be desired. To achieve the tradeoff, we define a new objective function as follows:

$$\min f = T_{01}^2 + T_{12}^2 + T_{34}^2 + T_{45}^2 + T_{56}^2 + (A_1^2 + A_2^2)R \quad (22)$$

where the quantity R is a weight factor, which can be tuned to achieve the balance between the travel time and the actuator effort. Specifically, higher values of R result in lower values of peak accelerations (hence, actuator effort) and longer travel time, and vice versa. Therefore, R can be thought of as a penalty term that generally assigns more cost value for using higher values of acceleration.

To demonstrate the tradeoff, the same inputs as in Fig. 4 are used, whereas the objective functions in (21) and (22) are implemented with different values of R . When $R = 0.001$, the

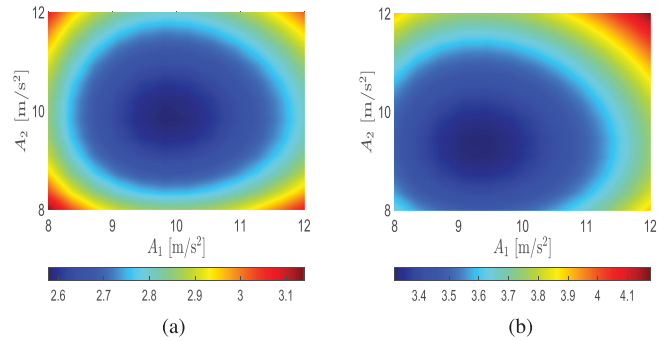


Fig. 6. Effect of R on acceleration A_1 and A_2 : As value of R increases, values of A_1 and A_2 , computed by the optimization algorithm, decrease. (a) Contour plot of objective function with $R=0.001$. (b) Contour plot of objective function with $R=0.005$.

objective function in (22) results in an acceleration profile with a peak value of 10 m/s^2 and a travel time of 5.048 s as compared to 12 m/s^2 in peak acceleration and 5.004 s of travel time when the objective function in (21) is used. The difference in the required peak acceleration is 20%, whereas the difference in time is only 0.87%. When $R = 0.005$, an acceleration profile with a peak value of 9.4 m/s^2 is obtained and the total time to complete the trajectory is 5.075 s. Again, the difference in the required peak acceleration is 27.66%, whereas the difference in time is only 1.40%. The dependence of the objective function in (22) on A_1 and A_2 in these two examples is depicted in Fig. 6. Thus, we can see that the objective function in (22) generates a trajectory with similar travel time and significantly less actuator effort indicated by the required peak acceleration.

The objective function (22) can also be written in quadratic form as $x^T Q x$, where Q is the identity matrix of dimension 6 by 6 with last entry of the matrix replaced by R and $x = [T_{01}, T_{12}, T_{34}, T_{45}, T_{56}, \sqrt{(A_1^2 + A_2^2)}]^T$. We can choose R such that Q is a positive definite matrix, and hence the objective function is strictly convex. Thus, according to the convex function theory presented in [24], there exists a unique solution to the objective function defined in (22).

As a result, this optimization approach is more flexible as compared to the divide and conquer approach in Section III-B. To illustrate it, we consider the case where the divide and conquer approach fails to yield positive value for all time intervals as discussed in Section III-B. Particularly, the inputs are $J_{\text{peak}} = 100 \text{ m/s}^3$, $A_{1\text{peak}} = 10 \text{ m/s}^2$, $A_{2\text{peak}} = 15 \text{ m/s}^2$, $V_{\text{peak}} = 0.8 \text{ m/s}$, and $S_{\text{peak}} = 0.2 \text{ m}$. The optimization scheme succeeds to plan the motion and the results obtained using objective function in (22) are shown in Fig. 7. It is interesting to note that the optimal solution indicates that $A_1 < A_{1\text{peak}}$ and $A_2 < A_{2\text{peak}}$, which explains why the divide and conquer approach fails.

The algorithm for the n th order: The same naming scheme described earlier in Section III-B is used here. A major task of the optimization approach is to develop the constraints and objective function. After these are fully defined, a solver can be employed to solve the optimization problem given an initial condition. The pseudocode discussed in Algorithm 2 describes the scheme to set up an objective function and constraints for

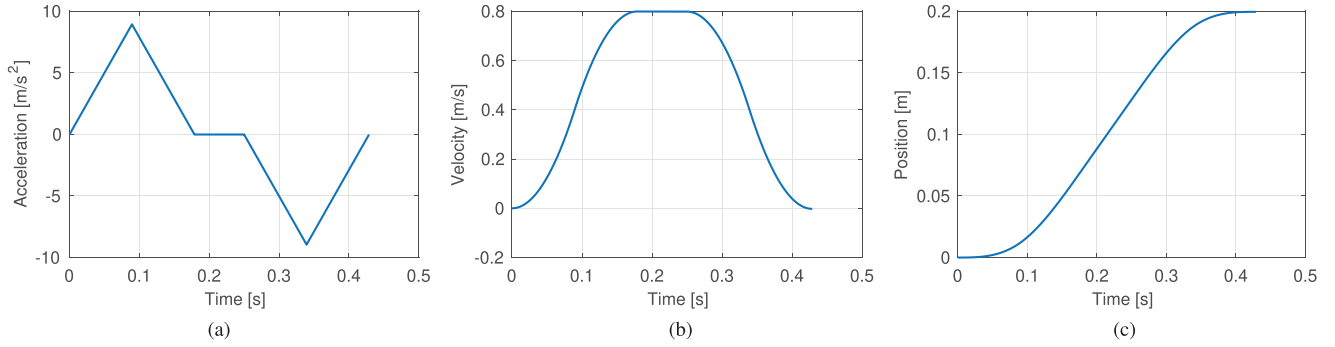


Fig. 7. Asymmetric s-curve trajectory with its velocity, acceleration, and jerk profiles obtained from the *optimization approach*. In this particular case, the *divide and conquer approach* was unable to produce an acceptable result with positive time intervals. (a) Acceleration profile over time. (b) Velocity profile over time. (c) Position profile over time.

the optimization problem of a general minimum-time s-curve trajectory.

Algorithm 2:

$segments = 2^n - 1;$
 $mid = 2^{n/2};$
 $trade = (A_1^2 + A_2^2) * R;$
 Objective function:

$$trade + \sum_{k=0}^{n-1} T_{2^k-1,2^k}^2 + \sum_{k=0}^{n-2} T_{mid+2^k-1,mid+2^k}^2 \quad (23)$$

Defining inequality constrains *cinq*:

$$cinq(a) = \begin{cases} A_1 \leq M_{1peak}^2 & a = 1 \\ A_2 \leq M_{2peak}^2 & a = 2 \end{cases} \quad (24)$$

```

z = 1;
for i = 1 : (n - 1) do
  s = i - 1;
  for j = i : segments do
    if j < mid then
      l = sum_{k=0}^{i-1} 2^k;
      ceq(z) = integral_{t_0}^{t_l} m_{0,l}^{n-s}(t) dt - M_{1peak}^{n-i};
      z = z + 1;
    else
      l = mid + sum_{k=0}^{i-1} 2^k;
      ceq(z) = integral_{t_mid}^{t_l} m_{mid,l}^{n-s}(t) dt + M_{2peak}^{n-i};
      z = z + 1;
    end
  end
end

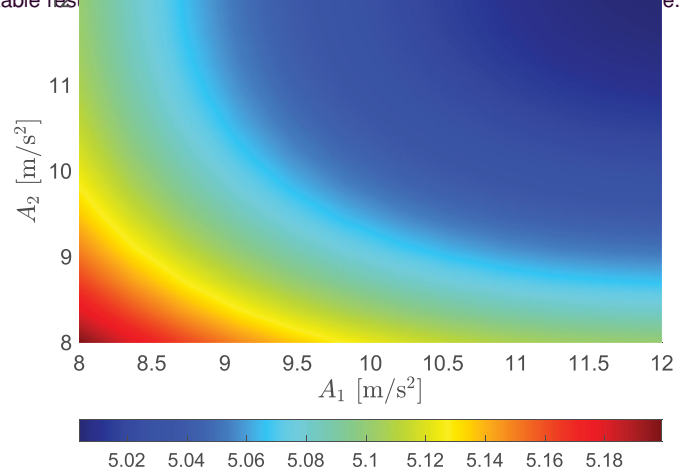
```

The position constraint can be expressed as:

$$ceq(z) = \int_{t_0}^{t_{segments}} m_{0,segments}^1(t) dt - M_{peak}^0 \quad (25)$$

D. Simulation Results

In this section, we discuss simulation data to validate that the proposed divide and conquer approach in Section III-B and the optimization in Section III-C both generate minimum-time trajectories. The methods are implemented for a wide range of input values (J_{peak} , $A_1 \leq A_{1peak}$, $A_2 \leq A_{2peak}$, V_{peak} , S_{peak}).



It is worth noting that the input values are selected such that the divide and conquer approach converges to the optimal solution. The results are shown in Table I, in which the first five columns show the values of the peak jerk, accelerations, velocity, and position, and the final two columns show the resultant total travel time obtained using the divide and conquer approach (Time-1) and the optimization approach (Time-2), respectively. We can see that both methods result in mostly identical total time in all cases with minor differences caused by rounding errors. This implies that they both converge to the only optimal solution.

In the next section, we implement the proposed motion-planning algorithm to demonstrate their efficacy in moving the end-effector of a robotic manipulator for nonprehensile manipulation as discussed in Section II.

IV. EXPERIMENTATION AND VALIDATION

A. Experimental Design and Setup

We use the experimental setup shown in Fig. 8 to validate the use of minimum-time s-curve trajectories for a robot arm's end-effector to maintain the stability of an object with an irregular shape, sitting on a tray mounted to the end-effector. The robot arm used in the experiments is a Franka Emika's Panda manipulator, which has seven rotational DOFs. The object to be manipulated is a rectangular plywood with properties shown in Table II and the object sits on a cardboard tray, which is screwed onto the manipulator's end-effector. A small wooden



Fig. 8. Experimental setup with an object to manipulated placed on top of a tray, which is mounted to the end effector of a robotic manipulator.

TABLE II
PARAMETERS OF THE RECTANGULAR BLOCK

Parameter	Value	Unit
Mass	43	g
Length	1.2	cm
Width	5.1	cm
Height	9.6	cm
Average critical angle when object tilts forward	0.0570	rad
Average critical acceleration when object tilts forward	0.5603	m/s ²
Average critical angle when object tilts backward	0.0874	rad
Average critical acceleration when object tilts backward	0.8598	m/s ²

piece is taped onto the rectangular plywood to make the whole object's shape irregular and asymmetric, which adds challenges to the nonprehensile manipulation task. Due to this asymmetrical shape of the object, the center of gravity is close to one edge facing forward in the direction of the motion as seen in Fig. 8. As a result, the object is very susceptible to forces applied along the direction of motion. This will be experimentally illustrated in Section IV-B.

In the experiments, the algorithms in Sections III-B and III-C generate s-curve trajectories for the end-effector. Their velocity profiles are used to compute the joint velocity profiles via inverse kinematics calculations. The joint velocities are passed to the robot program. The manipulator's prebuilt, fine-tuned controller then executes the motion. We have tried other control schemes, such as the resolved motion rate controller [25] and computed torque controller [26]. Nonetheless, the accuracy obtained from these methods is very similar. Therefore, we decided to use the prebuilt controller of the manipulator for convenience. Additionally, since the actuators used to drive the robot arm are velocity-controlled motors, passing velocity profiles to the joints is a reasonable choice.

For the plywood object shown in Fig. 8 and described in Table II, since it is not symmetric, there are two stability margin angles: 0.057 and 0.0874 rad, with the corresponding critical accelerations of 0.8598 and 0.5603 m/s², respectively. These

TABLE III
EXPERIMENT

Experiment	Critical angle [rad]		Critical acceleration [m/s ²]	
	back	front	back	front
1	0.0930	0.0596	0.9149	0.5850
2	0.0834	0.0548	0.8204	0.5380
3	0.0858	0.0584	0.8440	0.5732
4	0.0882	0.0596	0.8676	0.5850
5	0.0911	0.0584	0.8960	0.5732
6	0.0906	0.0560	0.8912	0.5498
7	0.0834	0.0524	0.8204	0.5146
8	0.0858	0.0548	0.8440	0.5380
9	0.0870	0.0584	0.8558	0.5732
10	0.0858	0.0584	0.8440	0.5732
Average	0.0874	0.0570	0.8598	0.5603

TABLE IV
SYMMETRIC S-CURVE MOTION WITH TOTAL TIME

$A_1 = A_2$	T_{01}	T_{12}	T_{34}	Total time	Remark
0.3	0.033	0.967	0.3	2.36667	Stable
0.35	0.039	0.818	0.437	2.22937	Stable
0.4	0.044	0.706	0.539	2.12778	Stable
0.42	0.047	0.668	0.572	2.09429	Wobble
0.44	0.049	0.633	0.603	2.06404	Wobble
0.46	0.051	0.601	0.63	2.03662	Tip over
0.48	0.053	0.572	0.655	2.01167	Tip over
0.5	0.056	0.544	0.678	1.98889	Tip over
0.51	0.057	0.532	0.688	1.97824	Tip over
0.52	0.058	0.519	0.699	1.96803	Tip over

acceleration values are the average values from 10 inclined plane experiments described by Fig. 2 and (4). The results from these experiments are shown in Table III. These critical accelerations are used to guide the selection of the peak accelerations that the motion planning algorithms use as inputs. The outcomes will be the desired motions of the end-effector that maintain the stability of the object. Furthermore, we also designed experiments that illustrate the advantage of an asymmetric s-curve trajectory over a symmetric one. Hence, the next section discusses the experimental results with symmetric s-curve trajectories followed by those with asymmetric motions.

B. Comparative Experiments

1) *Existing Methods—Symmetric Trajectory*: In order to highlight the advantages of the proposed time-optimal asymmetric s-curve method in this article, we compare its performance with the state-of-the-art method used in [17]–[19], [23], and [27]. This comparison in the context of nonprehensile manipulation, to the best of our knowledge, has never been done before.

To achieve these comparisons, following inputs are used with the optimization algorithm in Section III-C to generate symmetric s-curves: $J_{\text{peak}} = 9 \text{ m/s}^3$, $V_{\text{peak}} = 0.3 \text{ m/s}$, and $S_{\text{peak}} = 0.4 \text{ m}$. A wide range of values for $A_1 = A_2 = A_{\text{peak}}$ is employed and the results obtained are shown in Table IV. From these results, when the peak acceleration is below 0.42 m/s², the manipulator is able to maintain the stability of the object on the cardboard tray. As the tray's acceleration is increased, the object starts to wobble at 0.42 m/s². Here, wobbling means the object moves back and forth for some subseconds even after motion

TABLE V
ASYMMETRIC S-CURVE MOTION WITH TOTAL TIME

A_1	T_{01}	T_{12}	T_{34}	T_{45}	T_{56}	Total time	Remark
0.55	0.061	0.484	0.633	0.044	0.706	2.03384	Stable
0.62	0.069	0.415	0.660	0.044	0.706	2.00694	Stable
0.64	0.071	0.398	0.666	0.044	0.706	2.00049	Stable
0.68	0.076	0.366	0.678	0.044	0.706	1.98892	Stable
0.70	0.078	0.351	0.683	0.044	0.706	1.98373	Stable
0.72	0.080	0.337	0.688	0.044	0.706	1.97889	Wobble
0.74	0.082	0.323	0.692	0.044	0.706	1.97437	Wobble
0.75	0.083	0.317	0.694	0.044	0.706	1.97222	Tip over
0.78	0.087	0.298	0.700	0.044	0.706	1.96620	Tip over
0.80	0.089	0.286	0.704	0.044	0.706	1.96250	Tip over

has come to an end, and then regains a stable pose. When the peak acceleration is further increased to 0.46 m/s^2 and beyond, the wobbling amplitude increases and object tips over at the end of the trajectory. For higher accelerations, i.e., $\geq 0.5 \text{ m/s}^2$, no wobbling occurs and the object tips over during the motion. Now, we will implement asymmetric s-curve for the same task.

2) Proposed Method—Minimum-Time Asymmetric Trajectory: As discussed in Section III, our proposed asymmetric s-curve has the flexibility to design two peak values of accelerations: $A_{1\text{peak}}$ and $A_{2\text{peak}}$, both of which are less than or equal to the maximum possible acceleration input.

In the comparative experiments, we design an asymmetric s-curve trajectory with the same inputs as the existing method described in Section IV-B.1: $J_{\text{peak}} = 9 \text{ m/s}^3$, $V_{\text{peak}} = 0.3 \text{ m/s}$, and $S_{\text{peak}} = 0.4$. To reduce the wobbling effect, we leverage the advantage of our method and set $A_{2\text{peak}} = 0.4 \text{ m/s}^2$, whereas $A_{1\text{peak}}$ receives a wide range of values. The optimization algorithm described in Section III-C is employed to generate minimum-time asymmetric s-curve trajectories. The results obtained are shown in Table V. In summary, when $A_{1\text{peak}}$ is less than 0.70 m/s^2 , the object is stable and when $A_{1\text{peak}}$ is greater than or equal to 0.75 m/s^2 , the object loses stability.

3) Comparison Between two Methods: With an s-curve trajectory of the end-effector in general, an object tends to tilt backward during the first section of the trajectory due to being accelerated (the inertial force is acting backward). Similarly, the object tends to tilt forward during the second section of the trajectory due to being decelerated (the inertial force is acting forward). The stability-margin angle when the object tilts backward is larger as compared to when it tilts forward due to the asymmetric geometry of the object.

To further demonstrate this observation, we implemented an asymmetric s-curve with $A_{1\text{peak}} = 0.7 \text{ m/s}^2$ and $A_{2\text{peak}} = 0.46 \text{ m/s}^2$. Fig. 9 shows a sequence of snapshots of the object's pose during the end-effector's motion. During the first section of the trajectory, the object tilts backward as shown in Fig. 9(a) indicated by a positive angular displacement of its center of gravity. On the other hand, during the second section of the trajectory, the object tilts forward as shown in Fig. 9(b)–(d) indicated by a negative angular displacement of its center of gravity. The magnitude of the displacement is getting larger and the object eventually tips over under the action of the inertial force. This demonstration is consistent with the results obtained in Table IV with a peak acceleration of 0.46 m/s^2 . Thus, the experiment

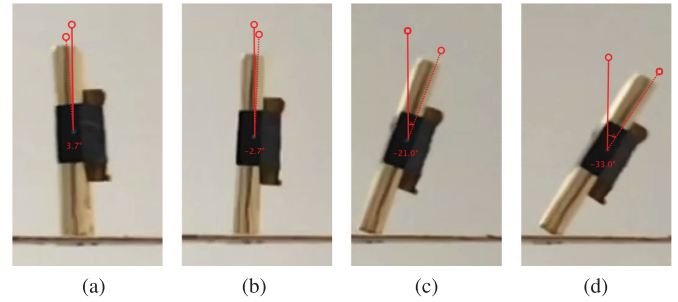


Fig. 9. Object response to an asymmetric s-curve motion with $A_{1\text{peak}} = 0.7 \text{ m/s}^2$ and $A_{2\text{peak}} = 0.46 \text{ m/s}^2$. Panel (a) showing a snapshot of the object leaning backward while accelerating. All of other panels show snapshots of the object leaning forward while decelerating. (a) $\theta \approx 3.7^\circ$. (b) $\theta \approx -2.7^\circ$. (c) $\theta \approx -21^\circ$. (d) $\theta \approx -33^\circ$.

clearly indicates two different critical peak accelerations for an asymmetric object during an s-curve motion. As a result, we were able to leverage this observation to manipulate an object in less time as compared to the existing symmetric s-curve method.

In the case of the existing method: as indicated in Table IV, with the end-effector's peak acceleration of 0.4 m/s^2 , the total time it takes to complete the trajectory is 2.12778 s . The total time to move the object can be reduced by increasing acceleration up to 0.46 m/s^2 , but at the cost of inducing wobbling that leads to falling. If wobbling is to be avoided, the end-effector can be accelerated with a peak acceleration up to only 0.4 m/s^2 .

In the case of our proposed method: the results indicate that the object starts to wobble while accelerating with $A_{1\text{peak}} = 0.72 \text{ m/s}^2$. When the peak acceleration of the end-effector is $A_{1\text{peak}} = 0.7 \text{ m/s}^2$ and below, the object stability is maintained throughout the motion. In particular, when $A_{1\text{peak}} = 0.7 \text{ m/s}^2$, the total time to complete the trajectory is 1.98373 s . In comparison, the existing method's best performance results in a total time of 2.12778 s . Thus, the flexibility in picking different peak accelerations in the two sections of an asymmetric s-curve enables the robot to complete the trajectory in a shorter time.

V. CONCLUSION

This article illustrates the effectiveness of using s-curve motions for nonprehensile manipulation. In particular, we analyze the stability margins of an object sitting on a tray moved by a robot arm. The stability analysis establishes a method to determine critical peak accelerations of the manipulator's end-effector, below which the object would not tip over. These critical accelerations lead us to s-curve motion profiles, for which motion designers may conveniently control the peak accelerations, among other kinematic specifications. From this perspective, we improve upon our previous work in [23] and [27] on this type of motion profiles by developing two general algorithms for planning an asymmetric s-curve with minimum travel time. The framework on the object's stability margin and the manipulator's motion planning for nonprehensile manipulation is validated via extensive experiments with a seven-DOF robotic arm. Furthermore, the experiments illustrate cases in which asymmetric s-curve motions are more effective than symmetric counterparts:

achieving stable nonprehensile manipulation of irregular objects in shorter time.

Nonprehensile manipulation is often formulated as feedback control problems, which require measuring the motions of the objects throughout the operation. Since we do not rely on feedback control, one of the main limitations of this work is that it is challenging to compute an exact critical acceleration such that the object does not tip over. Extending our proposed minimum time s-curve algorithm to a system capable of three-dimensional motion where object stability can be obtained by a feedback mechanism (like tracking with computer vision) is a possible direction of future work. Also, we are interested in exploring the effect of disturbances caused by the material inside the object (e.g., water motion in a bottle) on the critical accelerations.

From a totally different perspective, this article is one of very few that investigates nonprehensile manipulation as a motion planning problem without the need to continuously measure the state of the object to be manipulated. The work provides insights into how a robotic manipulator should move to guarantee the stability of an object sitting on top of a tray driven by its end-effector. Furthermore, the comparative experiments illustrate the advantages of asymmetric s-curve motion profiles over symmetric profiles. In addition, one of the novel elements of this article is an optimization scheme that considers the tradeoff between the total time of a trajectory and the actuator effort.

REFERENCES

- [1] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1711–1718, Jul. 2018.
- [2] J.-C. Ryu, F. Ruggiero, and K. M. Lynch, "Control of nonprehensile rolling manipulation: Balancing a disk on a disk," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1152–1161, Oct. 2013.
- [3] S. R. Erumalla, S. Pasupuleti, and J.-C. Ryu, "Throwing, catching, and balancing of a disk with a disk-shaped end effector on a two-link manipulator," *J. Mechanisms Robot.*, vol. 10, no. 5, 2018, Art. no. 054501.
- [4] F. Ruggiero *et al.*, "Nonprehensile manipulation of deformable objects: Achievements and perspectives from the RobDyman project," *IEEE Robot. Autom. Mag.*, vol. 25, no. 3, pp. 83–92, Sep. 2018.
- [5] S. Mathavan, M. R. Jackson, and R. M. Parkin, "Ball positioning in robotic billiards: A nonprehensile manipulation-based solution," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 1, pp. 184–195, Feb. 2016.
- [6] D. Serra, F. Ruggiero, A. Donaire, L. R. Buonocore, V. Lippiello, and B. Siciliano, "Control of nonprehensile planar rolling manipulation: A passivity-based approach," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 317–329, Apr. 2019.
- [7] Z. Deng, M. Stommel, and W. Xu, "Mechatronics design, modeling, and characterization of a soft robotic table for object manipulation on surface," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 6, pp. 2715–2725, Dec. 2018.
- [8] M. A. Erdmann and M. T. Mason, "An exploration of sensorless manipulation," *IEEE J. Robot. Autom.*, vol. 4, no. 4, pp. 369–379, Aug. 1988.
- [9] K. M. Lynch and M. T. Mason, "Dynamic nonprehensile manipulation: Controllability, planning, and experiments," *Int. J. Robot. Res.*, vol. 18, no. 1, pp. 64–92, 1999.
- [10] P. Mannam, A. V. Volkov, R. Paolini, G. Chirikjian, and M. T. Mason, "Sensorless pose determination using randomized action sequences," *Entropy*, vol. 21, no. 2, 2019, Art. no. 154.
- [11] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Cambridge, MA, USA: MIT Press, 2004.
- [12] R. G. Brown, *Introductory Physics I*, Durham, NC, USA: Duke Univ. Press, 1993.
- [13] J. Luo and K. Hauser, "Robust trajectory optimization under frictional contact with iterative learning," *Auton. Robots*, vol. 41, no. 6, pp. 1447–1461, 2017.
- [14] K.-H. Rew and K.-S. Kim, "Using asymmetric S-curve profile for fast and vibrationless motion," *Proc. Int. Conf. Control, Autom. Syst.*, 2007, pp. 500–504.
- [15] F. Zou, D. Qu, and F. Xu, "Asymmetric S-curve trajectory planning for robot point-to-point motion," in *Proc. IEEE Int. Conf. Robot. Biomimetics.*, 2009, pp. 2172–2176.
- [16] Y. Bai, X. Chen, and Z. Yang, "A generic method to generate AS-curve profile in commercial motion controller," in *Proc. ASME Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2017, Art. no. V009T07A047.
- [17] 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 Mach. Theory*, vol. 137, pp. 127–153, 2019.
- [18] H. Mu, Y. Zhou, S. Yan, and A. Han, "Third-order trajectory planning for high accuracy point-to-point motion," *Front. Elect. Electron. Eng. China*, vol. 4, no. 1, pp. 83–87, 2009.
- [19] B. Tondu and S. A. Bazaz, "The three-cubic method: An optimal online robot joint trajectory generator under velocity, acceleration, and wandering constraints," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 893–901, 1999.
- [20] R. Zanasi, C. G. L. Bianco, and A. Tonielli, "Nonlinear filters for the generation of smooth trajectories," *Automatica*, vol. 36, no. 3, pp. 439–448, 2000.
- [21] M. Yuan, Z. Chen, B. Yao, and X. Liu, "Fast and accurate motion tracking of a linear motor system under kinematic and dynamic constraints: An integrated planning and control approach," *IEEE Trans. Control Syst. Technol.*, to be published, doi: 10.1109/TCST.2019.2955658.
- [22] C. G. L. Bianco and F. Ghilardelli, "A discrete-time filter for the generation of signals with asymmetric and variable bounds on velocity, acceleration, and jerk," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4115–4125, Aug. 2014.
- [23] K. D. Nguyen, T.-C. Ng, and I.-M. Chen, "On algorithms for planning S-curve motion profiles," *Int. J. Adv. Robot. Syst.*, vol. 5, no. 1, pp. 99–106 2008.
- [24] J. Soriano, "Global minimum point of a convex function," *Appl. Math. Comput.*, vol. 55, no. 2–3, pp. 213–218, 1993.
- [25] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, vol. 10, no. 2, pp. 47–53, Jun. 1969.
- [26] D. M. Dawson, C. T. Abdallah, and F. L. Lewis, *Robot Manipulator Control: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2003.
- [27] K. D. Nguyen, I.-M. Chen, and T.-C. Ng, "Planning algorithms for s-curve trajectories," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics.*, 2007, pp. 1–6.



Praneel Acharya received the B.E. degree in aeronautical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2017. He is currently working toward the Ph.D. degree in mechanical engineering with the Mechanical Engineering Department, South Dakota State University, Brookings, SD, USA.

His research interests include manipulation for robotic systems, motion planning, control, and computer vision.



Kim-Doang Nguyen received the Ph.D. degree in mechanical engineering from the University of Illinois, Urbana-Champaign, Champaign, IL, USA, in 2015.

He is currently an Assistant Professor in mechanical engineering with South Dakota State University, Brookings, SD, USA. His research interests include robotics, mechatronics, adaptive control, applied machine learning, and time-delay systems.

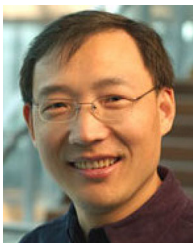


Hung M. La (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Oklahoma State University, Stillwater, OK, USA, in 2011.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, University of Nevada at Reno, Reno, NV, USA.

Dr. La is an Associate Editor for the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, and a Guest Editor for the *International Journal*

of *Robust and Nonlinear Control*.



Dikai Liu (Senior Member, IEEE) received the Ph.D. degree in dynamics and control from the Wuhan University of Technology, Wuhan, Hubei, China, in 1997.

He is currently a Professor of mechanical and mechatronic engineering with the Centre for Autonomous Systems, University of Technology Sydney, Sydney, NSW, Australia. His research interests include robot perception, planning and control for mobile manipulators operating in complex environments, such as

steel bridges, physical human–robot interaction, multirobot coordination, and bioinspired robotics.



I-Ming Chen (Fellow, IEEE) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, in 1986, and the M.S. and Ph.D. degrees from California Institute of Technology, Pasadena, CA, USA, in 1989 and 1994, respectively, all in mechanical engineering.

He is currently a Full Professor with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. He works on several different topics in robotics, such as mechanism, actuator, human–robot interaction, and industrial automation.

Dr. Chen is a fellow of ASME, the General Chairman of 2017 *IEEE International Conference on Robotics and Automation*, and a Senior Editor of the IEEE TRANSACTION ON ROBOTICS.