Nonprehensile Manipulation: a Trajectory-Planning Perspective

Praneel Acharya, Kim-Doang Nguyen, Dikai Liu, I-Ming Chen

***Abstract*—This paper 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*—Nonprehensile manipulation, asymmetric s-curve motion profiles, minimum-time trajectories, service robots**

I. Introduction

Robots have seen ubiquitous use in a wide range of applications. As it grows in popularity, it has drawn interest from different disciplines. Robots are increasingly considered solutions to existing problems. For instance, continuum robots are being developed for minimally invasive operations such as endoscopic surgery [1], mechatronic ankle prostheses have been designed to enhance functionality for lower limb amputees [2], and robotic exoskeletons are assisting elderly individuals with age-related motor performance decay [3].

In this paper, we are particularly interested in robotic applications involving object manipulation. In general, object manipulation can be classified into two approaches: prehensile (grasping) and non-prehensile (without grasping). Grasping is a complicated process which is highly challenging to replicate using a robotic device [4]. In particular, a human hand has around 27 bones and hundreds of tactile sensors. While building a robot with such dexterity and sensitivity is technically difficult, planing the motion of each degree of freedom (DOF) to reproduce human grasping is not less a challenge [5]. Though recent advances in soft robotics [6], tactile sensors [7], and learning algorithms [8] have pushed research in robotic grasping forward, the topic is still essentially a standing problem that awaits a breakthrough.

Nonprehensile manipulation is a intriguing approach because it is capable of manipulating multiple objects at a time as compared to grasping. However, since there is no grasping involved, objects are under constrained. This makes the system as a whole underactuated and poses a tremendous challenge for robotic devices to overcome. Robotics research in nonprehensile manipulation has achieved many major accomplishments. Recent developments on the topic include balancing a disk on a disk directly driven by a DC motor [9] or a disk on a disk driven by a robotic arm [10], a robotic platform capable of tossing and sliding soft objects [11], ball positioning in a robotic system that plays pool and snooker games [12]. In addition, work in [13] demonstrates that via surface deformation, a soft robotic table can manipulate multiple objects. A new control design framework, which is based on passivity-based port-Hamiltonian formulation, for rolling manipulation, was developed in [14]. The work's efficacy is illustrated for a dynamic ball-and-beam set-up and an eccentric disk-on-disk system. See [15] for an updated extensive survey on manipulation without grasping.

Though the topic has attracted a lot of attention lately, the research focus has been on the *control aspects* of the problem, especially for nonprehensile balancing tasks. While controls may guarantee desirable manipulation given that certain stability conditions are satisfied, controllers require feedback on the motion of the objects, usually provided by vision systems or motion sensors attached to the objects.

In this paper, 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 manipulator produces horizontal motions of its end-effector such that the object does not tip over.

Four different aspects of this problem will be discussed,

Identify applicable funding agency here. If none, delete this.

P. Acharya and K.D. Nguyen are with the Department of Mechanical Engineering, South Dakota State University, USA. E-mails: Praneel.Acharya@sdstate.edu, Doang.Nguyen@sdstate.edu

D. Liu is with the Centre for Autonomous Systems, University of Technology Sydney, Australia. E-mail: Dikai.Liu@uts.edu.au

I. M. Chen is with the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore. E-mail: michen@ntu.edu.sg

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 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 six-axis robotic manipulator in Section IV.

The key scientific contributions of this work include:

- The insights into nonprehensile manipulation from the motion planning point of view as opposed to the traditional feedback control point of view,
- The analysis and quantification of the stability of an object sitting on a tray moved by a robotic manipulator,
- A general algorithm that computes an asymmetric s-curve motion profile with minimum travel time,
- An optimization scheme that also designs minimum-time asymmetric s-curve motions and takes into account the trade-off between the travel time and actuator effort.
- Experimental validation using a six-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

Consider an object sitting on a tray moved along a horizontal direction by a robotic arm as shown in Fig. 1. This setting can be modeled as an inverted pendulum on a moving cart as in [16] and [17]. Pendulum-based modeling requires measuring state information of an object throughout the motion to form a closed-loop control system. However, in most realistic situations, for example, a robot serving a beverage bottle, it is not practical to have a motion sensor attached to the object to acquire the object's state signal as feedback information. Therefore, in the absence of feedback on the object's state, the object's natural ability to maintain its pose despite the acceleration of the tray is analyzed and quantified.

In this section, we use the term "stability" to represent an object's ability to maintain its pose throughout the motion. When an object is stable, a small deviation from the stability (e.g, a gentle push) is passively corrected back to the stable pose after the upsetting force stops [18]. 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. Work in [19] demonstrated that the line of gravity moves smoothly into and out of the perimeter of the supporting base before the lift-off phase of human jumping.

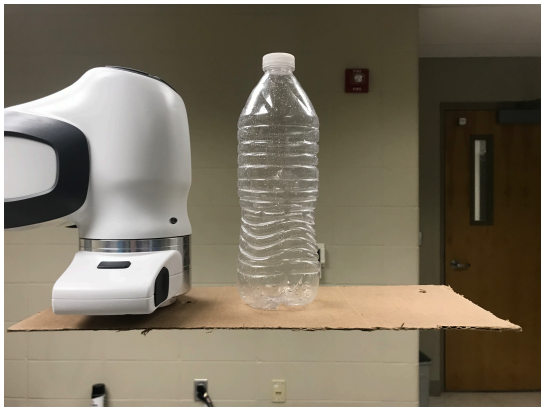


Fig. 1. Nonprehensile manipulation of interest: an object sitting on a tray moved by the end-effector of a robotic manipulator.

Similar observations were made when a human ascends from and descends to a seated position.

In our work, we are restricting our attention to objects that could not generate force themselves in contrast to human bodies as discussed in [19]. Hence, as long as the line of gravity lies within its base of support, an 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 [20]. 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 and that no lifting happens.

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. Then the stability index k can be defined as

$$k = \tanh(A_b/A_p). \tag{1}$$

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 k may have a value between 0 and 0.76159 inclusively.

For objects with $k \in (0\ 0.76159]$, 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 without the need to measure its geometry and center of gravity.

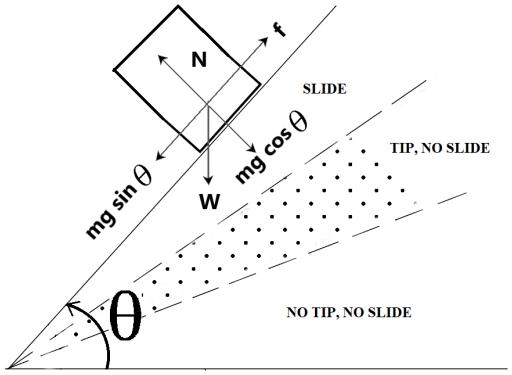


Fig. 2. The inclined-plane method used to compute the critical angles that represent the onsets of sliding and tipping. As inclination angle increases, the object responds differently depending upon which region θ is in.

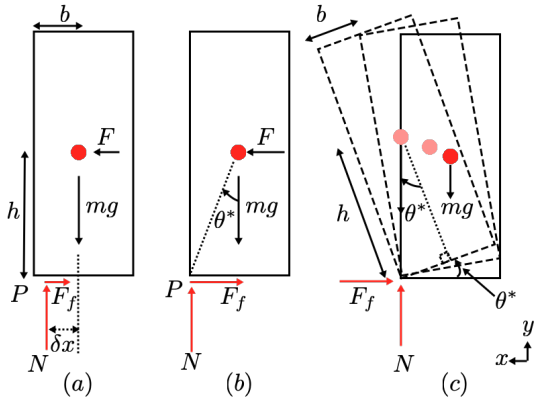


Fig. 3. An object on a surface under an external force: (a) and (b) illustrate that the location of the net normal force N moves toward the tipping point P as external force F increases its magnitude. When the object is about to tip, the net normal force acts at the tipping point P as shown in (b). During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

B. Stability-Margin Angle

For an object with $k \in (0\ 0.76159]$, 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 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

$$\tan(\theta^*_{\text{slide}}) = \mu, \tag{2}$$

where μ is the static friction coefficient. The stability-margin angle θ^* at which the object starts to tip over is not as simple. Given the geometry of an object, careful calculations on the base of support, location of the center of mass, the projected area, and so on, 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 $k \in (0\ 0.76159]$, its geometry plays a critical role in determining if the object will tip over or not 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 towards the tipping point P as shown in Fig. 3a 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

$$F = mg(b/h), \tag{3}$$

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 eternal 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.

$$F = ma_{\text{tray}}. \tag{4}$$

The acceleration of the tray can be then related to the inclination angle described in the experiment in Fig. 2. Specifically, as depicted in Fig. 3c, object tipping occurs when the line of gravity is passing the tipping point P. Hence, we obtain

$$\tan(\theta^*) = b/h \tag{5}$$

Taking (3), (4), and (5) into account, we conclude that the stability condition for an object sitting on a moving tray is

$$a_{\text{tray}} \leq g\tan(\theta^*). \tag{6}$$

If one wants to find the condition for the object not to slide, the same inequality as (6) can be used, but with θ^* replaced by θ^*_{slide} obtained in (2).

The stability condition (6) 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. $a_{\text{tray}} \leq \tan(\theta^*)g - F_{\text{drag}}/m$. Besides, induced vibration while motion is carried out is also neglected. In addition, 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 roughness and unevenness of the contact area between

the object and the tray are also not a part of (6). Nonetheless, we will use this stability condition to approximate the critical acceleration 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. In the next section, we discuss how to design the motion of the tray to guarantee the stability condition (6) 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 robotic arm's end-effector, its maximum acceleration throughout the motion is crucial. While the literature on motion planning of multi-axis manipulators is rich, s-curve motion planning is a method that provides a motion designer the full control of the trajectory's peak velocities, accelerations, jerks, and so on. Thus, in this project, we employ this motion planning method for our manipulator and demonstrate that s-curve profiles may be an excellent trajectory planning tool for nonprehensile manipulation.

The majority of work on s-curve motions so far focuses on symmetric s-curves and how to implement them in different applications. A few papers investigate asymmetric s-curve motions include [21], which suggests the use of an asymmetric s-curve to replace the traditional two-step symmetric s-curve. In [22], four different shapes of asymmetric s-curve based upon given constraints are formulated. In addition, work in [23] develops an asymmetric s-curve motion command that is compatible with most commercial motion controllers.

In general, when one plans machine motion for a task, minimum travel time is desired because this means the task is completed within the least amount of time, which is usually corresponding to optimal productivity. However, the aforementioned work did not formulate the s-curve motion planning as time-optimization problems. All time-optimal algorithms to date have been developed for symmetric s-curve trajectories, for instance, the motion planners proposed in [24]–[26]. To fill these gaps, not only are we interested in developing methods for designing asymmetric motion for nonprehensile manipulation, but we also want to minimize the travel time. Furthermore, we will design experiments to show the advantage of asymmetric s-curves over symmetric ones, which has never been shown before, to the best of our knowledge.

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_{01} describes the first time segment $[t_0,\ t_1]$, T_{12} describes the second time segment $[t_1,\ t_2]$, T_{23} describes third time segment $[t_2,\ t_3]$, T_{34} describes the fourth time segment $[t_3,\ t_4]$, T_{45} describes the fifth time segment $[t_4$ to $t_5]$, T_{56} describes the sixth time segment $[t_5,\ t_6]$, and T_{67} segment describes the last segment $[t_6,\ t_7]$.

A general third-order s-curve trajectory with seven time intervals can be divided into two different sections. The first section refers to the trajectory defined between t_0 to t_3 , while the second section refers to the trajectory defined between t_4 to t_7. In a symmetric s-curve, the jerk profile in the second section can be obtained by mirroring jerk profile of the first section. This leads to symmetric s-curves having the same value for the peak accelerations in both sections. In contrast, the jerk profile of an asymmetric s-curve in the second section cannot be obtained by mirroring the first section. Thus, the peak accelerations in the two sections are different. The peak acceleration in first section will be represented by $A_{1\text{peak}}$, while 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 $T_{45} = T_{67}$. The peak values J_{peak}, $A_{1\text{peak}}$, $A_{2\text{peak}}$, V_{peak}, and S_{peak} are inputs from users as they are task-related. 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 [27] for an example of such algorithms for a symmetric s-curve. Our focus is on an asymmetric s-curve described by equations (7), (8), (9), and (10):

$$p = \begin{cases} (1/6) * J_{\text{peak}} * t^3 & t_0 \le t \le t_1 \\ p_1 + v_1 * t + 0.5 * A_{1\text{peak}} * t^2 & t_1 \le t \le t_2 \\ p_2 + v_2 * t + 0.5 * A_{1\text{peak}} * t^2 - z & t_2 \le t \le t_3 \\ p_3 + v_3 * t & t_3 \le t \le t_4 \\ p_4 + v_4 * t - (1/6) * J_{\text{peak}} * t^3 & t_4 \le t \le t_5 \\ p_5 + v_5 * t - 0.5 * A_{2\text{peak}} * t^2 & t_5 \le t \le t_6 \\ p_6 + v_6 * t - 0.5 * A_{2\text{peak}} * t^2 + z & t_6 \le t \le t_7. \end{cases} \tag{10}$$

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

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 as possible to the peak value while satisfying the jerk constraint. Figure 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.

$$j = \begin{cases} J_{\text{peak}} \\ 0 \\ -J_{\text{peak}} \\ 0 \\ -J_{\text{peak}} \\ 0 \\ J_{\text{peak}} \end{cases} \quad (7) \quad a = \begin{cases} J_{\text{peak}} * t \\ A_{1\text{peak}} \\ A_{1\text{peak}} - J_{\text{peak}} * t \\ 0 \\ -J_{\text{peak}} * t \\ -A_{2\text{peak}} \\ -A_{2\text{peak}} + J_{\text{peak}} * t \end{cases} \quad (8) \quad v = \begin{cases} 0.5 * J_{\text{peak}} * t^2 & t_0 \le t \le t_1 \\ v_1 + A_{1\text{peak}} * t & t_1 \le t \le t_2 \\ v_2 + A_{1\text{peak}} * t - 0.5 * J_{\text{peak}} * t^2 & t_2 \le t \le t_3 \\ v_3 = V_{\text{peak}} & t_3 \le t \le t_4 \\ v_4 - 0.5 * J_{\text{peak}} * t^2 & t_4 \le t \le t_5 \\ v_5 - A_{2\text{peak}} * t & t_5 \le t \le t_6 \\ v_6 - A_{2\text{peak}} * t + 0.5 * J_{\text{peak}} * t^2 & t_6 \le t \le t_7 \end{cases} \quad (9)$$

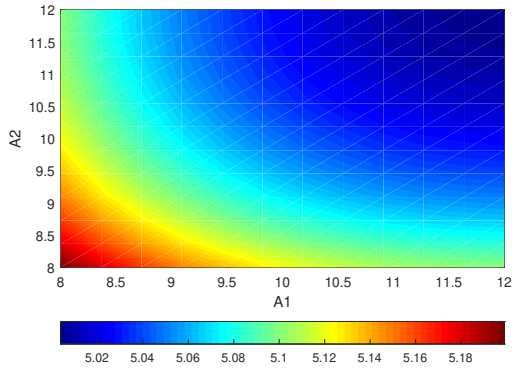


Fig. 4. The effect of A_1 and A_2 on total time: as A_1 or A_2 or both increase, the total time decreases while the peak jerk, velocity and position are fixed.

B. Divide and Concur 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 $A_1 = A_{1\text{peak}}$, $A_2 = A_{2\text{peak}}$. This leads to 5 inputs (or 5 constraint equations) and 5 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. This will be refereed to as the "divide and concur approach". In particular, as seen in (7) and (8), 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 (8) and (9) that

$$\int_0^{T_{01}} A_1 dt + \int_0^{T_{12}} A_1 dt - V_{\text{peak}} = 0. \quad (11)$$

With T_{01} known, we use (11) to compute T_{12}. Further, also from (8) and (9), we have

$$V_{\text{peak}} - \int_0^{T_{45}} A_2 dt - \int_0^{T_{56}} A_2 dt = 0. \quad (12)$$

By substituting T_{45} in (12), T_{56} is computed. Finally, T_{34} is calculated by integrating the velocity profile to obtain

$$S_{\text{peak}} - \int_0^{t_7} v dt = 0. \quad (13)$$

with v substituted from (9).

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 n^{th} order.

The general algorithm for n^{th} order trajectories: Since the s-curve of interest is asymmetric, if one looks at the k^{th} layer, for $k = 2, ..., n-1$, (e.g. acceleration in a third-order s-curve, acceleration and jerk in a fourth-order s-curve, etc), the corresponding peak values of the two sections are different. Therefore, looking at the k^{th} layer, for $k = 2, ..., n-1$, we denote by $M^k_{1\text{peak}}$ the peak value of the first segment and $M^k_{2\text{peak}}$ the peak value of the second segment. In addition, let M^0_{peak} refer to peak position value, M^1_{peak} refer to peak velocity value, and M^n_{peak} 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^0_{peak}, M^1_{peak}, $M^2_{1\text{peak}}$, $M^2_{2\text{peak}}$, $M^3_{1\text{peak}}$, $M^3_{2\text{peak}}$, $M^4_{1\text{peak}}$, $M^4_{2\text{peak}}$, and M^5_{peak}.

In order to generalize the above third-order calculations for an n^{th} order asymmetric s-curve, the naming scheme is updated as follows: M^0_{peak} refers to peak position value, M^1_{peak} refers to peak velocity value, M^2_{peak} refers to peak acceleration value, and so on up to M^n_{peak}. The sets of equations describing different segments of n^{th} layer $(m^n_{0,1}, m^n_{1,2}, ..., m^n_{2^n-2,2^n-1})$ can be obtained using similar approach as shown in equation (7). 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, $m^3_{0,1} = M^n_{\text{peak}}$, $m^3_{1,2} = 0$, $m^3_{2,3} = -M^n_{\text{peak}}$, and so on. Once $m^n_{0,1}, m^n_{1,2}, ..., m^n_{2^n-2,2^n-1}$ are fully defined, $m^{n-1}_{0,1}$, $m^{n-2}_{0,1}$ can be obtained as follows

$$m^{n-1}_{0,1}(t) = \int m^n_{0,1}(t)\text{d}t, \quad t_0 \le t \le t_1 \quad (14)$$

$$m^{n-2}_{0,1}(t) = \int m^{n-1}_{0,1}(t)\text{d}t, \quad t_0 \le t \le t_1 \quad (15)$$

In general,

$$m^{n-k}_{0,1}(t) = \int m^{n-k+1}_{0,1}(t)\text{d}t, \quad t_0 \le t \le t_1 \quad (16)$$

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

$$m^{n-k}_{b,b+1}(t) = m^{n-k}_{b-1,b}(t_b) + \int m^{n-k+1}_{b,b+1}(t)\text{d}t, \quad t_b \le t \le t_{b+1}. \quad (17)$$

Equation (17) 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:

$$\int_{t_b}^{t_c} m_{b,c}^{n-s} = \sum_{p=b}^{c-1} \int_{t_p}^{t_{p+1}} m_{p,p+1}^{n-s}(t). \tag{18}$$

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 equations (19), (20), and (21) below. Algorithm 1 summarizes the pseudo-codes that enable the computation of a general n^{th} order s-curve trajectory with minimal travel time.

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:

$$\int_{t_0}^{t_l} m_{0,l}^{n-s}(t)\,\mathrm{d}t - M_{1\text{peak}}^{n-i} = 0 \tag{19}$$

    else
      $l = mid + \sum_{k=0}^{i-1} 2^k$;
      Solve for $T_{mid-1+2^s,mid+2^s}$ as defined below:

$$\int_{t_{mid}}^{t_l} m_{mid,l}^{n-s}(t)\,\mathrm{d}t + M_{2\text{peak}}^{n-i} = 0 \tag{20}$$

    Where $\int_{t_b}^{t_c} m_{b,c}^{n-s}(t)$ is computed using equation (18) and $M_{2\text{peak}}^{1} = 0$.

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

$$\int_{t0}^{t_{segments}} m_{0,segments}^{1}(t)\,\mathrm{d}t - M_{\text{peak}}^{0} = 0 \tag{21}$$

Though this approach is a simple and quick way to plan an s-curve trajectory, it requires reasonable inputs be provided. For example, with $J_{\text{peak}} = 100$, $A_{1\text{peak}} = 8$, $A_{2\text{peak}} = 4$, $V_{\text{peak}} = 0.8$, $S_{\text{peak}} = 0.2$, the algorithm is able to calculate minimum time trajectory as seen 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: $J_{\text{peak}} = 100$, $A_{1\text{peak}} = 10$, $A_{2\text{peak}} = 15$, $V_{\text{peak}} = 0.8$, $S_{\text{peak}} = 0.2$. Solving for these constrains would lead to error since all given constrains 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 propose 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 concur approach fails.

C. Optimization Approach

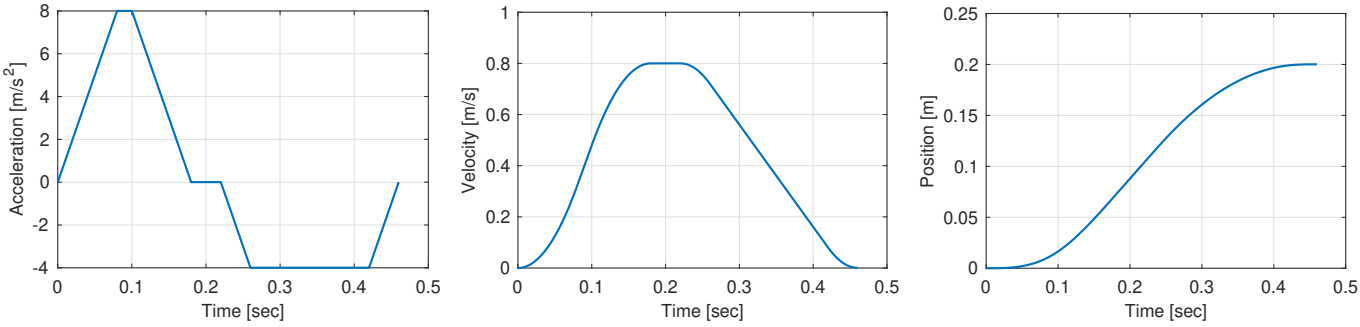
We first set up an optimization problem to plan a minimum-time third-order s-curve trajectory with inputs: J_{peak}, $A_{1\text{peak}}$, $A_{2\text{peak}}$, V_{peak}, S_{peak}. In the previous section, we directly set $A_1 = A_{1\text{peak}}$ and $A_2 = A_{2\text{peak}}$, then calculate the time intervals. 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 $A_{2\text{peak}}$, respectively. In summary, for a third-order trajectory, we have seven variables, five equality constraints and two inequality-constraints:

$$\begin{cases} J_{\text{peak}} - A_1/T_{01} = 0 \\ J_{\text{peak}} - A_2/T_{45} = 0 \\ A_1 \leq A_{1\text{peak}} \\ A_2 \leq A_{2\text{peak}} \\ V_{\text{peak}} - 2 * \int_0^{T_{01}} A_1 dt - \int_0^{T_{12}} A_1 dt = 0 \\ V_{\text{peak}} - 2 * \int_0^{T_{45}} A_1 dt - \int_0^{T_{56}} A_2 dt = 0 \\ S_{\text{peak}} - p_7 = 0 \end{cases} \tag{22}$$

To deal with the inequality constraints, slack variables are used to transform them into equality constraints. Hence, the optimization includes 9 variables, namely the 2 slack variables, A_1, A_2, T_{01}, T_{12}, T_{34}, T_{45}, and T_{56}, and 7 equality constraints described in (22). The two DOF allow the algorithm to search for a solution that optimizes the objective function:

$$\min f = |T_{01}| + |T_{12}| + |T_{34}| + |T_{45}| + |T_{56}|. \tag{23}$$

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. Besides, it is assumed that the user inputs are given such that the problem is feasible.



(a) Acceleration over time (b) Velocity over time (c) Position over time

Fig. 5. An asymmetric s-curve trajectory with its velocity and acceleration profiles generated by the *divide and concur method*.

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

$$\min f = T_{01}^2 + T_{12}^2 + T_{34}^2 + T_{45}^2 + T_{56}^2 + (A_1^2 + A_2^2)R, \quad (24)$$

where the quantity R is a weight factor, which determines the trade-off between the travel time and the actuator effort. Specifically, lower values of R will result in solutions that require higher actuator effort and vice-versa.

To demonstrate the trade-off, the same inputs as in Fig. 4 are used, while the objective functions in (23) and (24) are implemented with different values of R. When $R = 0.001$, the objective function in (24) results in an acceleration profile with a peak value of 10 m/s^2 and a travel time of 5.048 seconds as compared to 12 m/s^2 in peak acceleration and 5.004 seconds of travel time when the objective function in (23) is used. The difference in the required peak acceleration is 20%, while 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 seconds. Again, the difference in the required peak acceleration is 27.66%, while the difference in time is only 1.40%. The dependence of the objective function in (24) on A_1 and A_2 in these two examples is depicted in Fig. 6. Thus, we can see that the objective function in (24) generates a trajectory with similar travel time and significantly less actuator effort indicated by the required peak acceleration.

This optimization approach is more flexible as compared to the divide and concur approach in Section III-B. To illustrate it, we consider the case where the divide and concur approach fails to yield positive value for all time intervals as discussed in Section III-B. Particularly, the inputs are $J_{\text{peak}} = 100$, $A_{1\text{peak}} = 10$, $A_{2\text{peak}} = 15$, $V_{\text{peak}} = 0.8$, and $S_{\text{peak}} = 0.2$. The optimization scheme succeeds to plan the motion and the results obtained using objective function in (24) 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 concur approach fails.

The algorithm for 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 pseudo-code discussed in Algorithm 2 describes the scheme to set up an objective function and constraints for the optimization problem of a general minimum-time s-curve trajectory.

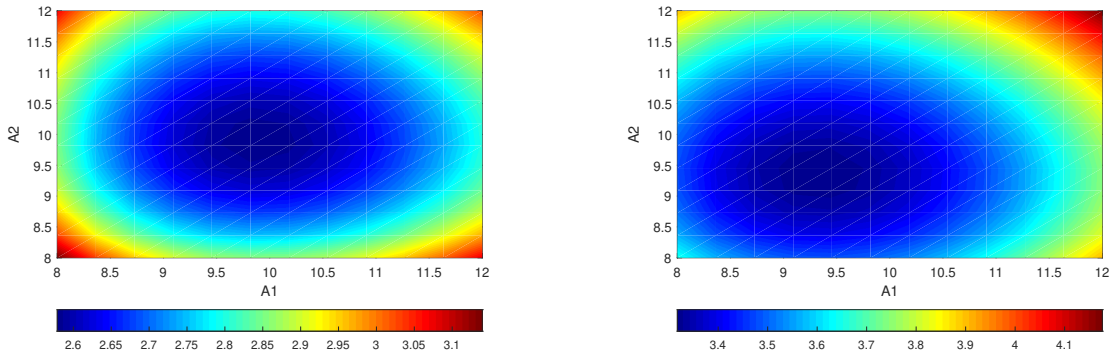
D. Simulation Results

In this section, we discuss simulation data to validate that the proposed divide and concur 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}, $A_1 <= A_{1\text{peak}}$, $A_2 <= A_{2\text{peak}}$, V_{peak}, S_{peak}). It is worth noting that the input values are selected such that the divide and concur approach succeeds. 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 concur 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 bar numerical errors. Total time obtained from both methods has been rounded to the third decimal place.

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

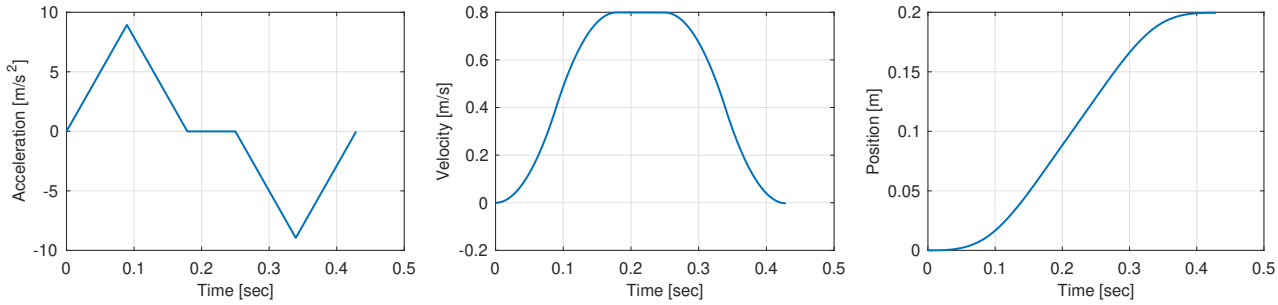
We use the experimental set-up shown in Fig. 8 to validate the use of minimum-time s-curve trajectories for a robot arm's end-effector to maintain the stability of an object with an irregular shape, sitting on a tray mounted to the end-effector. The robot arm used in the experiments is a Franka Emika's Pada manipulator, which has six rotational DOF. The object to be manipulated is rectangular plywood with properties shown



(a) Contour plot of objective function with R=0.001

(b) Contour plot of objective function with R=0.005

Fig. 6. The effect of R on acceleration A_1 and A_2: As value of R increases, values of $A1$ and $A2$, computed by the optimization algorithm, decreases.



(a) Acceleration profile over time

(b) Velocity profile over time

(c) Position profile over time

Fig. 7. An asymmetric s-curve trajectory with its velocity, acceleration, and jerk profiles obtained from the *optimization approach*. In this particular case, the *divide and concur approach* was unable to produce an acceptable result with positive time intervals.

in table II. A small wooden 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. The object sits on a cardboard tray which is screwed onto the manipulator's end-effector.

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 joints 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 [28] and computed torque controller [29]. Nonetheless, the accuracy obtained from these methods is very similar. Therefore, we decided to used 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 rad and 0.0874 rad, with the corresponding critical accelerations of 0.8598 m/s^2 and 0.5603 m/s^2, respectively. These acceleration values are the average values from 10 inclined plane experiments described by Fig. 2 and equation (6). 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

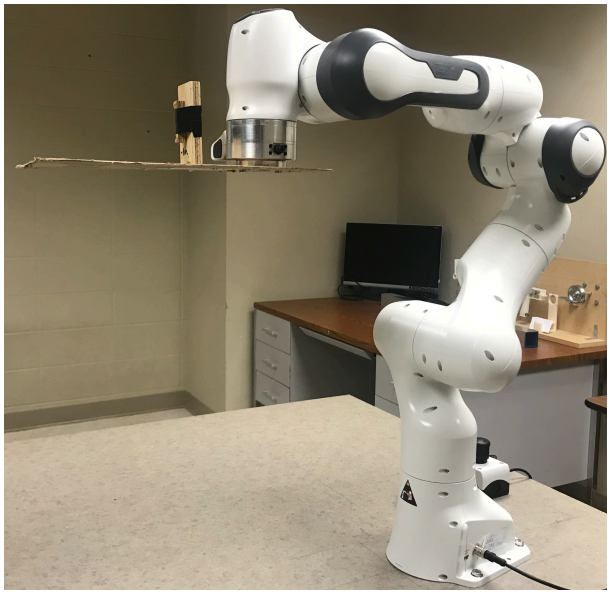


Fig. 8. Experimental setup with an object to manipulated placed on top of a tray, which is mounted to the end effector of a robotic manipulator.

Algorithm 2:

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

$$trade + \sum_{k=0}^{n-1} T^2_{2^k-1,2^k} + \sum_{k=0}^{n-2} T^2_{mid+2^k-1,mid+2^k} \quad (25)$$

Defining inequality constrains $cinq$:

$$cinq(a) = \begin{cases} A_1 \leq M^2_{\text{1peak}} & a = 1 \\ A_2 \leq M^2_{\text{2peak}} & a = 2 \end{cases} \quad (26)$$

$z = 1$;
for $i = 1 : (n-1)$ **do**
  $s = i - 1$;
  for $j = i : mid : segments$ **do**
    if $j < mid$ **then**
      $l = \sum_{k=0}^{i-1} 2^k$;
      $ceq(z) = \int_{t_0}^{t_l} m^{n-s}_{0,l}(t)\,\mathrm{d}t - M^{n-i}_{1_{peak}}$
      $z = z + 1$;
    else
      $l = mid + \sum_{k=0}^{i-1} 2^k$;
      $ceq(z) = \int_{t_m id}^{t_l} m^{n-s}_{mid,l}(t)\,\mathrm{d}t + M^{n-i}_{2_{peak}}$
      $z = z + 1$;

The position constraint can be expressed as:

$$ceq(z) = \int_{t_0}^{t_{segments}} m^1_{0,segments}(t)\,\mathrm{d}t - M^0_{peak} \quad (27)$$

TABLE I
TOTAL TIME FROM BOTH METHODS

J_{peak}	A_{1peak}	A_{2peak}	V_{peak}	S_{peak}	Time-1	Time-2
13.993	29.524	39.512	111.573	931.040	14.113	14.113
17.379	18.734	30.036	51.912	298.339	9.400	9.400
15.612	55.095	73.073	342.028	3384.488	19.445	19.444
14.283	27.565	30.816	66.488	471.026	11.413	11.413
19.827	57.953	70.299	249.257	2524.689	17.287	17.286
18.797	61.521	70.647	265.525	2137.765	15.604	15.604
13.196	28.042	41.902	133.057	1077.500	14.709	14.708
19.687	41.873	53.159	143.541	837.230	11.311	11.310
15.271	27.977	47.733	149.203	1284.563	15.318	15.318
10.680	10.904	14.949	20.925	114.303	8.334	8.332

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.

A. Symmetric minimum time trajectory

The following inputs $J_{\text{peak}} = 9$, $V_{\text{peak}} = 0.3$, $S_{\text{peak}} = 0.4$ are used with the optimization algorithm in Section III-C to generate symmetric s-curves. A wide range of values for

TABLE II
PARAMETERS OF RECTANGULAR BLOCK

Parameters	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

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

$A_1 = A_2 = A_{peak}$ is employed and the results obtained are shown in Table IV. From these results, when the peak acceleration is below 0.42 m/s^2, 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/s^2. Here, wobbling means the object moves back and forth for some sub-seconds even after motion 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. ≥ 0.5 m/s^2, no wobbling occurs and the object tips over during the motion.

In addition, 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 seconds. 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. If wobbling is to be avoided, the end-effector can be accelerated with peak acceleration up to 0.4 m/s^2. This total time will be compared to the total time obtained while executing an asymmetric s-curve trajectory as discussed in the following section.

B. Asymmetric minimum time trajectory

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 object tilts backward is larger as compared to when it tilts forward due to the asymmetric geometry of the object.

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

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

To further demonstrate this observation, we implemented an asymmetric s-curve with $A_{1\text{peak}} = 0.7$ m/s^2 and $A_{2\text{peak}} = 0.46$ m/s^2. Figure 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 Figure 9a 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 Figure 9b, c, and 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 clearly indicates two different critical peak accelerations for an asymmetric object during an s-curve motion. As a result, we will show that asymmetric s-curves can be used to leverage this observation to manipulate an object in less time as compared to a symmetric s-curve.

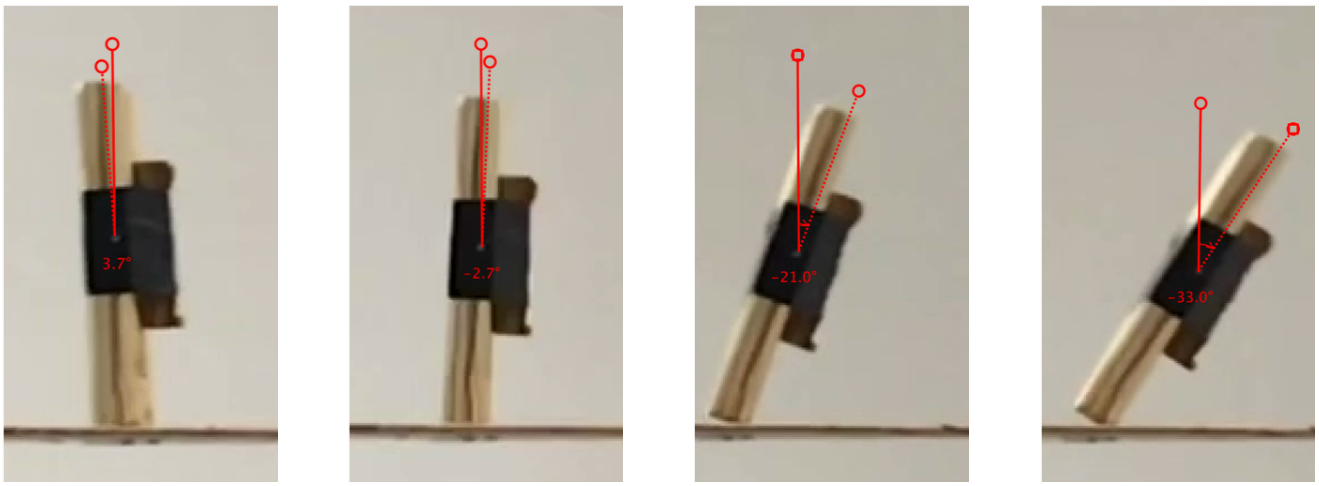
Since the object can be accelerated with a higher peak value during the first section as compared to the second section of an s-curve trajectory while maintaining its stability, an asymmetric s-curve with a higher peak acceleration during the first section of the trajectory is generated, i.e. $A_{1\text{peak}} > A_{2\text{peak}}$. We use the same inputs as in the symmetric case in the last section except for the peak accelerations: $J_{\text{peak}} = 9$, $V_{\text{peak}} = 0.3$, $S_{\text{peak}} = 0.4$. To reduce wobbling effect, $A_{2\text{peak}}$ is picked to have the value of 0.4 m/s^2 based upon the results obtained when symmetric s-curves are applied in the last section. With $A_{2\text{peak}}$ value of 0.4 m/s^2 fixed, $A_{1\text{peak}}$ value of a wide range is implemented. The optimization approach described in Section III-C is again employed to generate minimum-time asymmetric s-curve trajectories. The results obtained are shown in Table V.

The results indicate that the object starts to wobble while accelerating with $A_{1\text{peak}} = 0.72$ m/s^2. When the peak acceleration of the end-effector is $A_{1\text{peak}} = 0.7$ m/s^2 and below, the object stability is maintained throughout the motion. In particular, when $A_{1\text{peak}} = 0.7$ m/s^2, the total time to complete the trajectory is 1.98373 s. As a reminder of the result from the last section, the symmetric s-curve 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 paper 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 [30] and [27] on this type of motion profiles by developing two general algorithms for planning an asymmetric s-curve with minimum travel time. One method is based on dividing the profile into segments and then computing the time intervals, while the other method is formulated as an optimization problem and then solves for a trade-off balance between the optimal time and the actuator effort. The framework on the object's stability margin and the manipulator's motion planning for nonprehensile manipulation is validated via extensive experiments with a six-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 challenging to achieve since objects to be manipulated are under-constrained. This is often formulated as feedback control problems, which require measuring the motions of the objects throughout the operation. This is usually impractical if nonprehensile manipulation is applied in situations such as a mobile manipulator serving food or beverages in restaurants or carrying medical tools to work alongside doctors and nurses in hospitals. From a totally different perspective, this paper is one of very few that investigate 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 of a tray driven by its end-effector. Furthermore, this is the only work so far, to the best of our knowledge, that illustrates the advantages of asymmetric s-curve motion profiles over symmetric profiles. In addition, one of the novel elements of this paper is an optimization scheme that considers the trade-off between the total time of a trajectory and the actuator effort.



(a) $\theta \approx 0.064$ rad $(3.7\,\text{deg})$ (b) $\theta \approx -0.047$ rad $(-2.7\,\text{deg})$ (c) $\theta \approx -0.366$ rad $(-21\,\text{deg})$ (d) $\theta \approx -0.575$ rad $(-33\,\text{deg})$

Fig. 9. Object response to an asymmetric s-curve motion with $A_{1\text{peak}} = 0.7$ m/s^2 and $A_{2\text{peak}} = 0.46$ 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.

REFERENCES

[1] M. Hwang and D. Kwon, "Strong continuum manipulator for flexible endoscopic surgery," *IEEE/ASME Transactions on Mechatronics*, 2019.

[2] T. Yu, A. R. Plummer, P. Iravani, J. B. S. Bhatti, S. Zahedi, and D. Moser, "The design, control and testing of an integrated electrohydrostatic powered ankle prosthesis," *IEEE/ASME Transactions on Mechatronics*, 2019.

[3] T. Zhang and H. Huang, "Design and control of a series elastic actuator with clutch for hip exoskeleton for precise assistive magnitude and timing control and improved mechanical safety," *IEEE/ASME Transactions on Mechatronics*, 2019.

[4] Y. C. Nakamura, D. M. Troniak, A. Rodriguez, M. T. Mason, and N. S. Pollard, "The complexities of grasping in the wild," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 233–240.

[5] W. S. Howard and V. Kumar, "On the stability of grasped objects," *IEEE transactions on robotics and automation*, vol. 12, no. 6, pp. 904–917, 1996.

[6] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Sci. Robot*, vol. 1, no. 1, p. eaah3690, 2016.

[7] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, 2017.

[8] Z. Li, T. Zhao, F. Chen, Y. Hu, C.-Y. Su, and T. Fukuda, "Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 121–131, 2017.

[9] J.-C. Ryu, F. Ruggiero, and K. M. Lynch, "Control of nonprehensile rolling manipulation: Balancing a disk on a disk," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1152–1161, 2013.

[10] 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," *Journal of Mechanisms and Robotics*, vol. 10, no. 5, p. 054501, 2018.

[11] F. Ruggiero, A. Petit, D. Serra, A. Satici, J. Cacace, A. Donaire, F. Ficuciello, L. Buonocore, G. Fontanelli, V. Lippiello *et al.*, "Nonprehensile manipulation of deformable objects: Achievements and perspectives from the robdyman project," *IEEE Robotics and Automation Magazine*, vol. 25, no. 3, pp. 83–92, 2018.

[12] S. Mathavan, M. R. Jackson, and R. M. Parkin, "Ball positioning in robotic billiards: a nonprehensile manipulation-based solution," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 1, pp. 184–195, 2015.

[14] 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 Transactions on Robotics*, 2019.

[13] Z. Deng, M. Stommel, and W. Xu, "Mechatronics design, modeling, and characterization of a soft robotic table for object manipulation on surface," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2715–2725, 2018.

[15] 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.

[16] R. Lozano and I. Fantoni, "Passivity based control of the inverted pendulum," in *Perspectives in control*. Springer, 1998, pp. 83–95.

[17] A. K. Stimac, "Standup and stabilization of the inverted pendulum," Ph.D. dissertation, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 1999.

[18] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2004.

[19] M. Murray, A. Seireg, and R. Scholz, "Center of gravity, center of pressure, and supportive forces during human activities." *Journal of applied physiology*, vol. 23, no. 6, pp. 831–838, 1967.

[20] R. G. Brown, *Introductory Physics I*, 1993.

[21] K.-H. Rew and K.-S. Kim, "Using asymmetric s-curve profile for fast and vibrationless motion," in *2007 International Conference on Control, Automation and Systems*. IEEE, 2007, pp. 500–504.

[22] F. Zou, D. Qu, and F. Xu, "Asymmetric s-curve trajectory planning for robot point-to-point motion," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2009, pp. 2172–2176.

[23] Y. Bai, X. Chen, and Z. Yang, "A generic method to generate as-curve profile in commercial motion controller," in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2017, pp. V009T07A047–V009T07A047.

[24] Y. Fang, J. Hu, W. Liu, Q. Shao, J. Qi, and Y. Peng, "Smooth and time-optimal s-curve trajectory planning for automated robots and machines," *Mechanism and Machine Theory*, vol. 137, pp. 127–153, 2019.

[25] H. Mu, Y. Zhou, S. Yan, and A. Han, "Third-order trajectory planning for high accuracy point-to-point motion," *Frontiers of Electrical and Electronic Engineering in China*, vol. 4, no. 1, pp. 83–87, 2009.

[26] B. Tondu and S. A. Bazaz, "The three-cubic method: an optimal online robot joint trajectory generator under velocity, acceleration, and wandering constraints," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 893–901, 1999.

[27] K. D. Nguyen, T.-C. Ng, and I.-M. Chen, "On algorithms for planning s-curve motion profiles," *International Journal of Advanced Robotic Systems*, vol. 5, no. 1, p. 11, 2008.

[28] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on man-machine systems*, vol. 10, no. 2, pp. 47–53, 1969.

[29] D. M. Dawson, C. T. Abdallah, and F. L. Lewis, *Robot manipulator control: theory and practice*. CRC Press, 2003.

[30] K. D. Nguyen, I.-M. Chen, and T.-C. Ng, "Planning algorithms for s-curve trajectories," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*. IEEE, 2007, pp. 1–6.