IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

On Transient Release Dynamics in Robot Throwing: A Sliding Pivot Model

Yang Liu, *Graduate Student Member, IEEE*, and Aude Billard, *Fellow, IEEE*

***Abstract*—Humans regularly throw projectiles with high speed and accuracy; some animals, including chimpanzees and elephants, also throw objects occasionally. In comparison, robots are currently lagging behind, despite having lower communication latency and more accurate motor control. To understand this paradox and ultimately achieve ubiquitous throwing robots, one of the major obstacles is the lack of high-fidelity and tractable physical models of the transient release dynamics, where the momentum exchange between the hand and the object occurs within tens of milliseconds via the frictional interface. In this work, we try to establish a physical model for the release dynamics. We first demonstrate that the conventional model, which combines rigid-body dynamics and patch friction [limit surface (LS)], struggles to capture the release dynamics and exhibits pathological behaviors, such as Zeno-like oscillations, leading to poor accuracy in predicting throwing outcomes. To mitigate this, we formulate a viscous-smoothed variant of the limit surface model solved via implicit integration (ILS), which achieves high predictive fidelity but incurs significant computational cost. On the other hand, motivated by the dominant effect of in-hand pivoting in release dynamics, we propose a sliding pivot model that simplifies the contact dynamics by capturing the sticking–pivoting–sliding behavior emerging under vanishing normal force. This model achieves accuracy comparable to ILS, with only 10% higher error while offering over 20× faster computation. Compared to conventional LS models, our method reduces horizontal velocity prediction error by 40% and angular velocity prediction error by 63%, achieving 2.4-cm mean absolute error (MAE) for landing position and 15.4° MAE for landing orientation. These results provide a robust physically grounded foundation for future scalable robot throwing systems.**

***Index Terms*—Contact modeling, dynamic manipulation, robot throwing, transient dynamics.**

I. Introduction

OBJECT transport is a major application of robots in logistics and manufacturing automation. In the long-standing effort to optimize operations, robot throwing introduces a new paradigm to drastically increase efficiency and throughput. Unlike conventional pick-and-place, a pick-and-throw robot can set off on its next cycle immediately after releasing the current object, moving to pick the next item while the released object flies in the air toward its desired location, saving the time and energy required to carry the robot's heavy body all the way to the target along with the payload.

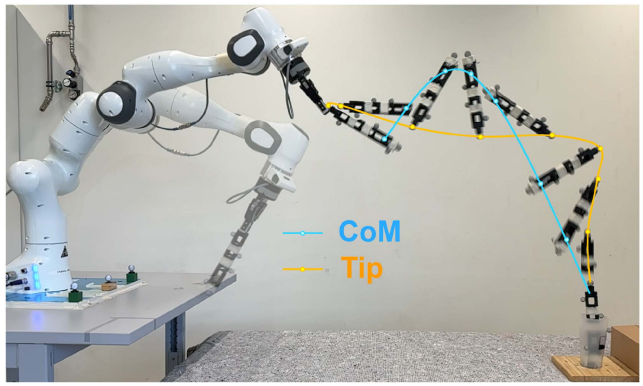


Fig. 1. Robot throws a firmly grasped bar into the target cup, completing a full flip. The bar's angular velocity during free flight far exceeds that of the robot hand. What happens during the short 50-ms window during the transient of gripper opening? In this work, we aim to understand this phenomenon through a physical model.

Although robot throwing has been demonstrated to be possible under various laboratory conditions, a complete physical model describing the momentum transfer during the throwing process is still missing. Yet, this is essential to guarantee the wide adoption and scalable deployment of robot-throwing systems. With such a model, several important questions can be addressed.

1) What is the bottleneck in the hardware system for reliable throwing? How fast and accurate should the sensing and actuation be?
2) Can one construct throwing motions that are robust in the presence of various sources of uncertainty?
3) How can we predict the object's flying trajectory and landing velocity to prevent damage upon landing?

Robot throwing unfolds into stages.

1) First, the hand firmly grasps the object to prevent unexpected slips before the release, causing deformation of the fingers and the object due to the applied force.
2) The robot executes the throwing motion and begins to accelerate. During this sticking phase, the object's linear and angular momentum are *passively* induced by the propagation of rigid-body motion along the entire kinematic chain, from the robot base to the object.
3) At a specific instant during the throwing motion, the hand receives the opening command and begins to open gradually. In response, the previously squeezed finger pad and the object expand back to their original shapes, leading

Received 22 June 2025; accepted 26 September 2025. Date of publication 18 November 2025; date of current version 11 December 2025. This work was supported by European Union project DARKO under Grant H2020 ICT-46-2020. This article was recommended for publication by Associate Editor M. Posa and Editor J. Bohg upon evaluation of the reviewers' comments. *(Corresponding author: Yang Liu.)*

The authors are with the Learning Algorithms and Systems Laboratory, Swiss Federal Institute of Technology Lausanne, 1015 Lausanne, Switzerland (e-mail: yangliudh@gmail.com; aude.billard@epfl.ch).

Digital Object Identifier 10.1109/TRO.2025.3634386

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

to *changing* normal force and *changing* patch friction limits.

4) As the normal force continues to decrease, the frictional interface can no longer maintain the object's sticking, causing it to begin sliding/spinning in hand. During this stage, the object's motion is governed by gravity and the frictional transmission through the contact patch.
5) Finally, as the normal force vanishes, the object fully detaches from the hand and begins flying and spinning toward the desired target.

In the throwing process, rich and accurate models already exist for robot dynamics [1] and the object's free-flying dynamics [2], [3]. However, the momentum exchange via the friction patch during the transient phase of release remains poorly understood. On the other hand, as demonstrated in the throwing-flipping example in Fig. 1, this frictional momentum exchange plays a crucial role in determining the object's motion and landing outcome.

In this work, we aim to build a *forward dynamics* model of the release dynamics, i.e., *given*: 1) robot's joint motion sequence in position, velocity, and acceleration during the transient phase of release; 2) object's inertia and contact properties; and 3) gripper normal force sequence during the transient phase of release, *predict*: object's state (pose and twist) upon entering free-flying when the gripper normal force vanishes.

We start with a conventional modeling strategy: combining rigid body dynamics with the common patch friction model *Limit Surface (LS)* [4], [5], [6], and we identify that the velocity-magnitude-independent frictional wrench, described by the LS models, causes the robot–object dynamical system to be discontinuous, leading to pathological behavior (similar to Zeno phenomenon [7]) and limited model accuracy due to its sensitivity to uncertainty. In response, we propose *sliding pivot* (SP), a physical surrogate model that offers smoother dynamics. Extensive real robot throwing experiments are conducted to validate and assess the models' accuracies.

Contributions: To our knowledge, this work presents the first study on the physical modeling of transient release dynamics in robot throwing. We identify the limitations of conventional patch friction models in describing the robot–object system during breaking contact. To address this, we propose an SP model that captures the sticking–pivoting–sliding behavior during the transient release. This model is physically grounded, interpretable, and computationally efficient. For comparison, we also develop a viscous-smoothed variant of the limit surface model, solved via implicit integration (denoted as ILS), which yields the highest prediction accuracy across tested models but incurs significant computational cost. The SP model achieves similar predictive accuracy—with only 10% higher error than ILS, while offering over 20× faster computation. Compared to the conventional nonsmoothed LS model, SP reduces horizontal velocity prediction error by 40%, reduces angular velocity prediction error by 63%, and improves landing prediction accuracy to a mean absolute error (MAE) of 2.4 cm for landing position and 15.4° for landing orientation, with significantly lower variability and systematic bias across a wide range of experimental conditions. The model's simplicity and efficiency make it highly suitable for integration into robot learning and planning frameworks.

II. Related Works

Over the past decades, numerous approaches have been offered to enable robots to throw objects [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31]. Different works have been developed based on various assumptions, using robots with different capabilities (e.g., speed and accuracy) to throw different sets of objects with distinct end-effectors.

A key differentiating factor across all of these works is the ability of the assumed model or hypothesis classes to handle eccentric throws. This distinction arises naturally: for center-of-mass (CoM) grasps, the object's linear motion and landing position are unaffected by its in-hand spinning, allowing simple models to capture the reality for position-targeted throws sufficiently. However, when landing poses (i.e., position and orientation) are desired, or when eccentric grasps are unavoidable (to throw a tennis racket, only the handle end fits the robot's hand), richer models become necessary. Next, we review each of these two categories in more detail.

A. Throwing With CoM Grasps

Early works in robot throwing mostly fall within the regime of nonprehensile throwing, where objects are held immobile in-hand using inertial forces during arm acceleration and are released instantaneously by drastic deceleration [8], [9], [11], [12]. In nonprehensile throwing, a "perfect release model" can be assumed, where the object's detachment state (CoM position and velocity) matches the finger's state at the release instant, facilitating throwing planning. However, nonprehensile throwing is restricted to specially shaped objects, such as balls with concave supports [8], [9] or blocks with flat surfaces [11], [12], and is limited in throwing postures.

Prehensile throwing, where the object is firmly held during acceleration and gradually released by opening the fingers, is more versatile—for instance, enabling throws with diverse postures and accommodating objects with irregular shapes. However, it also introduces uncertainties due to gradual gripper–object interactions during the release phase, particularly for deformable objects. These effects lead to both category-level uncertainties—variations in landing outcomes across different object types—and instance-level uncertainties, caused by unrepeatable gripper grasping forces and opening motions. To address such complexities, one approach is to design specialized high-speed grippers. This has been shown to enable nearly instantaneous object release, achieving high precision at the cost of high-end hardware [13].

Another line of research focuses on synthesizing robust throwing motions against the release uncertainty. Monastirsky et al. [22] *implicitly* assume the "gripper opening delay" kinematic release model and apply reinforcement learning with domain randomization on release timing. While directly deploying a model trained purely in simulation results in highly inaccurate throws, fine-tuning with a handful of real-world experiments allows the system to accurately throw various objects, including deformable ones. Liu and Billard [28] *explicitly* assume the "gripper opening delay" model and develop a convex formulation for the efficient and reliable synthesis of robust release motions against timing uncertainty. The online-generated robust throwing motion (<50 ms) achieves high accuracy in throwing a variety of complex deformable objects with dexterous postures, without training on any throwing data.

The results in these two works demonstrate the physical fidelity of the "gripper opening delay" assumption for CoM grasps. However, as noted in both works, neither the simulated

throw nor the explicit "gripper opening delay" model can accurately capture the release dynamics for eccentric throws.

B. Throwing With Eccentric Grasps

Compared to CoM grasps, eccentric throwing has received much less attention, with two notable exceptions: TossingBot [21] and TossNet [27], both of which use end-to-end data-driven approaches to model the process.

TossingBot [21] learns an end-to-end mapping from grasping to throwing, leveraging visual observations (RGB-D images) to implicitly capture the relationship between grasping offsets from the CoM and throwing outcomes. The system uses trial and error to optimize both grasping and throwing policies simultaneously, resulting in a robust throwing pipeline for rigid objects. TossingBot achieves an 85% throwing accuracy with a target box measuring 25 cm × 15 cm (25 cm in the throwing direction) after 15 000 real throws, demonstrating its capability to adapt to varying object properties such as mass distribution. However, the approach assumes that visual information alone is sufficient to infer an object's mass distribution, which limits its ability to handle more complex dynamics or deformable objects. In addition, the object's rotation is not modeled.

TossNet [27] incorporates proprioceptive sensing and force/torque (F/T) data, to encode object CoM and inertia information. Using this additional sensory input, TossNet requires fewer training samples (∼3000 throws) to achieve much more accurate results while also predicting the object's orientation during flight (MAE: position—3 cm; orientation—4.5°).

Despite their pioneering advancements, both TossingBot and TossNet face inherent limitations as data-driven end-to-end models. Their reliance on large-scale training data limits adaptability to new robot embodiments and operational conditions, whereas physical models, expressed in SI units, offer greater transferability—even across planetary environments (e.g., imagine a robot throwing on Mars). In addition, in structured settings like warehouse automation, where object properties (e.g., mass and CoM) are available, physical models can leverage this information effectively, reducing data requirements and enabling efficient zero-shot deployment.

III. Preliminaries

A. Problem Formulation

We are interested in building a forward dynamics model that predicts the initial free-flying state of the object from the prethrow grasp pose, robot motion, contact normal forces, friction parameters, and object inertia. The transient phase only lasts˜ ∼100 ms and is notoriously difficult to model due to the time-varying contact patch friction.

A unique feature of patch friction is the coupling between tangential friction and torsional friction, e.g., the torsional friction limit decreases with increasing tangential friction load [32]. Although there exist patch friction models integrated from static Coulomb friction law, i.e., standard LSs [4], and have been successfully validated in quasi-static sliding experiments [33], we will show that such a friction model, when combined with rigid body dynamics, leads to pathological behavior (Zeno-like oscillations) even when simulating the simplest eccentric throwing scenario: "object dropping." Discovering this limitation, we hypothesize that only relaxing the rigidness assumption in the friction or object dynamics models can resolve this issue.

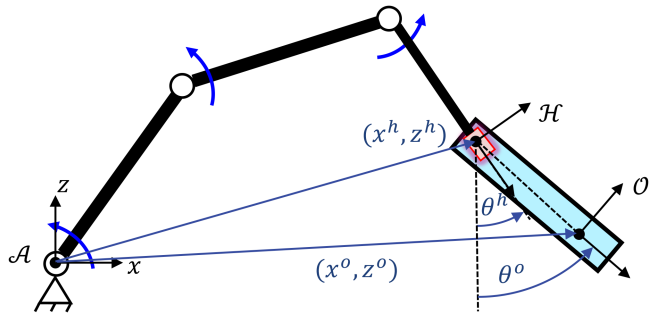


Fig. 2. Major notations for release dynamics modeling.

Instead of building a fine-grained finite-element model to capture the microscopic bristle deformation for soft friction or the continuum deformation for soft object body, we observe that the majority of the object spatial momentum is injected and vectored through the *tangential friction* during the release. Therefore, we abstract the complex transient dynamics as a simple and intuitive physical model of *SP*.

The list of assumptions is as follows:

1) the object is pinch-grasped by parallel fingers;
2) during release, the only forces acting on the object are the normal force from the fingers, friction from the fingers, and gravity, while air drag and the Magnus effect are neglected during the short period of release;
3) the frictional patches on both fingers are identical and isotropic, with the same normal force, friction coefficient, and deformation, resulting in a symmetrical contact condition between the fingers. Consequently, the net force applied to the object lies within the plane in which the object travels during free flight. Note that this assumption could be less valid for anthropomorphic hands due to asymmetric structures.

B. Parameters and Variables

The major notations are summarized in Fig. 2. We use the robot base frame $\mathcal{A}$ as the world frame. The hand frame $\mathcal{H}$ is located on the plane of motion, at the midpoint between the two finger contact patches projected onto this plane. Given the symmetrical contact condition, this corresponds to the center of the projected contact patches. The object frame $\mathcal{O}$ is located at its CoM. The generalized coordinates of the hand are denoted by

$$\mathbf{q}^h = [x^h, z^h, \theta^h]^\top \in \mathbb{R}^3 \tag{1}$$

where x^h is the horizontal position of the hand frame $\mathcal{H}$ relative to the robot base frame $\mathcal{A}$, z^h is the vertical position of the hand frame $\mathcal{H}$ relative to the robot base frame $\mathcal{A}$, and θ^h is the hand's orientation relative to the vertical orientation. Twist, acceleration, and applied wrench at the center of the fingers are expressed in the world frame $\mathcal{A}$ and are denoted by

$$\mathbf{v}^h = [v_x^h, v_z^h, \omega^h]^\top \in \mathbb{R}^3 \tag{2}$$

$$\mathbf{a}^h = [a_x^h, a_z^h, \alpha^h]^\top \in \mathbb{R}^3 \tag{3}$$

$$\mathbf{f}^h = [f_x^h, f_z^h, \tau^h]^\top \in \mathbb{R}^3. \tag{4}$$

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

The generalized coordinates of the object are denoted by

$$\mathbf{q}^o = [x^o, z^o, \theta^o]^\top \in \mathbb{R}^3 \tag{5}$$

where x^o is the horizontal position of its CoM relative to the robot base frame $\mathcal{A}$, z^o is the vertical position of its CoM relative to the robot base frame $\mathcal{A}$, and θ^o is the object's orientation relative to the vertical orientation. Note that θ^o is unwrapped from $[0, 2\pi]$ in order to differentiate different number of flips during its free-flying. Likewise, twist, acceleration, and applied wrench at the CoM are expressed in the world frame $\mathcal{A}$ and are denoted by

$$\mathbf{v}^o = [v_x^o, v_z^o, \omega^o]^\top \in \mathbb{R}^3 \tag{6}$$

$$\mathbf{a}^o = [a_x^o, a_z^o, \alpha^o]^\top \in \mathbb{R}^3 \tag{7}$$

$$\mathbf{f}^o = [f_x^o, f_z^o, \tau^o]^\top \in \mathbb{R}^3. \tag{8}$$

Define the relative coordinates of object $\mathcal{O}$ w.r.t. hand $\mathcal{H}$

$$\mathbf{q}^r = [x^r, z^r, \theta^r]^\top := \mathbf{q}^o - \mathbf{q}^h \in \mathbb{R}^3. \tag{9}$$

For the contact point $\mathcal{C}$ on the object specified by a relative vector $\mathbf{q}^r$, the linear velocity of $\mathcal{C}$ expressed in the robot base frame $\mathcal{A}$, denoted as $\mathbf{v}_{xz}^c$, is

$$\mathbf{v}_{xz}^c = \mathbf{v}_{xz}^o + w^o \times -\mathbf{q}_{xz}^r = \begin{bmatrix} v_x^o + w^o z^r \\ v_z^o - w^o x^r \end{bmatrix} \tag{10}$$

where $\mathbf{v}_{xz}^o$ is the linear velocity of the object $\mathcal{O}$. Hence, we can define the transformation matrix $\mathbf{G}(\mathbf{q}^r)$ that relates the contact point twist $\mathbf{v}^c$ and object twist $\mathbf{v}^o$

$$\mathbf{v}^c = \begin{bmatrix} v_x^c \\ v_z^c \\ w^c \end{bmatrix} = \begin{bmatrix} v_x^o + w^o z^r \\ v_z^o - w^o x^r \\ w^o \end{bmatrix} = \mathbf{G}(\mathbf{q}_r)\mathbf{v}^o \tag{11}$$

with

$$\mathbf{G}(\mathbf{q}^r) = \begin{bmatrix} 1 & 0 & z^r \\ 0 & 1 & -x^r \\ 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

Similarly, the wrenches are related via [34]

$$\mathbf{f}^o = \mathbf{G}^\top(\mathbf{q}^r)\mathbf{f}^c. \tag{13}$$

The relative velocity $\mathbf{v}^r$ is the difference between the velocity of the *contact point* $\mathbf{v}^c$ and the velocity of the hand $\mathbf{v}^h$

$$\mathbf{v}^r := \mathbf{v}^c - \mathbf{v}^h. \tag{14}$$

The planar rigid body acceleration of the contact patch on the object $\mathbf{a}^c$ can be written as

$$\mathbf{a}^c = \begin{bmatrix} a_x^c \\ a_z^c \\ \alpha^c \end{bmatrix} = \mathbf{a}^o + \begin{bmatrix} \alpha^o z^r + \omega^{o2} x^r \\ -\alpha^o x^r + \omega^{o2} z^r \\ 0 \end{bmatrix}. \tag{15}$$

Then, we define the relative acceleration $\mathbf{a}_r$ as the *slip acceleration*

$$\mathbf{a}^r := \mathbf{a}^c - \mathbf{a}^h. \tag{16}$$

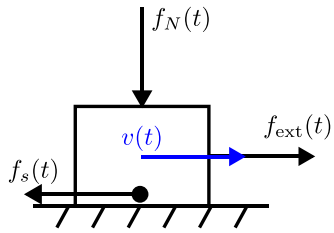


Fig. 3. 1-D frictional sliding system.

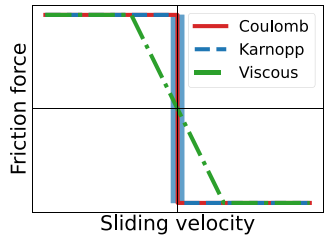


Fig. 4. Friction models: Coulomb, Karnopp, and Viscous.

IV. Transient Dynamics With Patch Friction and Rigid Body Dynamics

In this section, we first revisit the Coulomb friction law and its two variants: the Karnopp model and the viscous-smoothed Coulomb model, in the context of 1-D dynamic frictional sticking–sliding. Then, we introduce LSs, a common patch friction model based on Coulomb's friction law, and develop a release dynamics model in conjunction with rigid body dynamics. We use an "object dropping" example to demonstrate and discuss the limitations of LS-based models in capturing the release dynamics.

A. Dynamic Frictional Sticking–Sliding in 1-D

In the 1-D system illustrated in Fig. 3, a block with mass m is initially statically stuck on a rough surface with the friction coefficient μ and is subjected to a decreasing normal force $f_N(t)$ and an external force $f_{\text{ext}}(t)$. At some instant when the external force exceeds the friction limit, the sliding onsets. To describe such a system, we present three 1-D sliding friction models commonly used in robotics, shown in Fig. 4, which will also inform our discussion of patch friction modeling. For more complex 1-D friction models, we refer the reader to the survey [35].

1) Coulomb Friction Law:

$$f_s = \begin{cases} -f_{\text{ext}}, & \text{if } |v| = 0 \text{ and } |f_{\text{ext}}| \le \mu f_N \quad \text{(sticking)} \\ \text{undefined} & \text{if } |v| = 0 \text{ and } |f_{\text{ext}}| > \mu f_N \quad \text{(onset)} \\ -\text{sgn}(v)\mu f_N, & \text{if } |v| > 0 \qquad\qquad\quad \text{(sliding)}. \end{cases}$$

The standard Coulomb friction law is undefined when the external force exceeds the friction limit, i.e., at sliding onset. Therefore, it is insufficient for modeling or simulating sticking–sliding systems.

2) Karnopp Friction Model: The Karnopp friction model [36], a variant based on the Coulomb friction law, defines a δ-dead zone with $\delta > 0$ to address the underdetermined Coulomb frictional force at the onset instant and effectively

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

describes this dynamics process. According to the Karnopp model, the following ODE governs the dynamics of the block:

$$m\dot{v}(t) = f_{\text{ext}}(t) + f_s(t) \tag{17}$$

where,

$$f_s = \begin{cases} -f_{\text{ext}}, & \text{if } |v| \le \delta \text{ and } |f_{\text{ext}}| \le \mu f_N \quad \text{(sticking)} \\ -\text{sgn}(f_{\text{ext}})\mu f_N, & \text{if } |v| \le \delta \text{ and } |f_{\text{ext}}| > \mu f_N \quad \text{(onset)} \\ -\text{sgn}(v)\mu f_N, & \text{if } |v| > \delta \quad \text{(sliding).} \end{cases} \tag{18}$$

In this small sliding velocity zone, the effect of excess external load is accumulated. More specifically, if $v = 0$ and $|f_{\text{ext}}| \le \mu f_N$, the net horizontal force applied to the block is zero; hence, $\dot{v} = 0$, and the system remains sticking. If $|v| \le \delta$ and $|f_{\text{ext}}| > \mu f_N$, the *excess* external load $f_{\text{ext}} - \text{sgn}(f_{\text{ext}})\mu f_N$ becomes the net force applied to the object, pushing its velocity outside the dead zone. Physically, this dead zone represents the unmodeled presliding regime, where the sliding interface may undergo microscopic sliding while remaining macroscopically stationary. In this regime, the net force applied to the object is only a portion of the external load, $(1 - \frac{\mu f_N}{|f_{\text{ext}}|})$. The Karnopp friction model has been used in [37] and [38] to model and simulate 1-D sticking–sliding.

3) Viscous-Smoothed Coulomb Model: A simple way to regularize the Coulomb discontinuity is to let the tangential friction force grow *linearly* with the slip velocity until it reaches the static limit μf_N. Denoting a small bandwidth parameter by $d > 0[\text{m/s}]$, the law is given by

$$f_s(v) = \mu f_N \ \min\{|v|/d, 1\} \ \text{sgn}(v).$$

This introduces a transition band of width $2d$, within which the friction force ramps linearly from zero to the Coulomb limit. Viscous smoothing results in a continuous and differentiable force profile. Such regularization is widely adopted in robotic simulators [39], [40] to improve numerical stability.

B. Dynamic LS Model for Patch Contact

Friction laws of patch contact can be determined by integrating contact meshes with the local Coulomb friction law, given normal force, material friction coefficient, and pressure distribution. This results in a surface in the 3-D space of contact wrenches [4]. Such integral is generally not integrable, and real-world experimentations have shown that ellipsoidal LSs are good approximations for common robot fingers [32]. Mathematically, an ellipsoidal LS is implicitly defined by a quadratic function $H(\mathbf{f}) : \mathbb{R}^3 \to \mathbb{R}$

$$H(\mathbf{f}) = \mathbf{f}^\top \mathbf{A}\mathbf{f} = \mathbf{f}^\top \begin{bmatrix} \mu^{-2} & 0 & 0 \\ 0 & \mu^{-2} & 0 \\ 0 & 0 & (cr\mu)^{-2} \end{bmatrix} \mathbf{f} \tag{19}$$

where μ is the contact patch's friction coefficient, r is the radius of the contact patch, and the constant $c \in [0, 1]$ represents the factor accounting for the contact geometry under uniform pressure distribution. Here, we set $c = 0.6$, in line with common values in the literature [6], [41], [42]. This quadratic function describes the patch friction limit as follows.

1) The f_N-sublevel set $L^-_{f_N}(H) = \{\mathbf{f} : H(\mathbf{f}) \le f_N^2\}$ corresponds to the set $\mathcal{F}$ of all possible friction load that can resist without sliding,

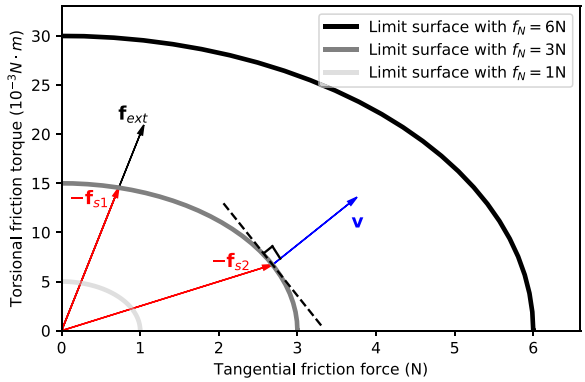


Fig. 5. Illustration of LS and Karnopp treatment for transient patch contact.

2) The f_N-level set $L_{f_N}(H) = \{\mathbf{f} : H(\mathbf{f}) = f_N^2\}$ corresponds to the set of friction wrenches during sliding, which lies on the boundary surface of $\mathcal{F}$,
3) The f_N-superlevel set $L^+_{f_N}(H) = \{\mathbf{f} : H(\mathbf{f}) > f_N^2\}$ corresponds to external loads that lie outside $\mathcal{F}$. When such a wrench is applied to the object, the onset of sliding occurs. Following the Karnopp "cutoff" scheme, the friction wrench $\mathbf{f}_s$ is a fraction of the external wrench $\mathbf{f}_{\text{ext}}$ sustained by the frictional patch, i.e., $\mathbf{f}_s = -\frac{f_N}{\sqrt{H(\mathbf{f}_{\text{ext}})}}\mathbf{f}_{\text{ext}}$.

In Fig. 5,
the color-diminishing isolines represent the family of LSs with decreasing normal force. For better intuition, the numbers are comparable in scale to those obtained from real-world experimentation by Howe et al. [32]. When $f_N = 6$ N, $\mathbf{f}_{\text{ext}}$ lies within the LS of 6-N normal force, and hence, the frictional patch is sticking. When $f_N = 3$ N, $\mathbf{f}_{\text{ext}}$ exceeds the LS of 3-N normal force, and it starts sliding. The pair $(\mathbf{f}_{\text{ext}}, -\mathbf{f}_{s1})$ illustrates the "cutoff" regime of the Karnopp model upon onset.

During sliding, the sliding friction wrench is determined by the sliding velocity direction. Given a local sliding velocity $\mathbf{v} \in \mathbb{R}^3$, the sliding friction wrench in $L_{f_N}(H)$ is the one where the surface normal $\nabla H(\mathbf{f})$ is parallel to the sliding velocity. For ellipsoidal LS $H(\mathbf{f}) = \mathbf{f}^\top \mathbf{A}\mathbf{f}$, the sliding friction wrench is given by a function $\boldsymbol{\Gamma}(\mathbf{v}) : \mathbb{R}^3 \to \mathbb{R}^3$, with

$$\boldsymbol{\Gamma}(\mathbf{v}) = -\frac{\mathbf{A}^{-1}\mathbf{v}}{\sqrt{\mathbf{v}^\top \mathbf{A}^{-1}\mathbf{v}}}.$$

In Fig. 5, the pair $(\mathbf{v}, -\mathbf{f}_{s2})$ illustrates the relationship between sliding velocity and frictional wrench during steady-state sliding. In other words, the sliding friction wrench only depends on the sliding velocity direction, irrespective of the velocity magnitude.

Mathematically, such a frictional wrench function that is independent of the sliding velocity magnitude is *positively homogeneous of degree zero* [43]. This property holds for LSs integrated from any contact pressure distribution with local Coulomb friction law, including the typical ellipsoidal approximation used here, as well as more complex-shaped LSs [44]. It leads to a codimension-3 discontinuity set in the state space of the frictional dynamical system [45], which cannot be treated using classical Filippov systems [46].

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

To address this discontinuity, we borrow the idea of viscous regularization from 1-D friction and propose a simple remedy

$$\mathbf{\Gamma}_d(\mathbf{v}) = -\frac{\mathbf{A}^{-1}\mathbf{v}}{\sqrt{\mathbf{v}^\top \mathbf{A}^{-1}\mathbf{v}}} \min\{\|v\|/d, 1\}$$

where d is the smoothing factor, with $d \to 0$ recovering the sliding velocity-magnitude-independent friction wrench.

In summary, the dynamic LS model is described as follows:

$$\mathbf{f}_s = \begin{cases} -\mathbf{f}_{\text{ext}}, & \text{if } \|\mathbf{v}\| \le \delta \text{ and } H(\mathbf{f}_{\text{ext}}) \le f_N^2 \quad \text{(sticking)} \\ -\frac{f_N}{\sqrt{H(\mathbf{f}_{\text{ext}})}}\mathbf{f}_{\text{ext}}, & \text{if } \|\mathbf{v}\| \le \delta \text{ and } H(\mathbf{f}_{\text{ext}}) > f_N^2 \quad \text{(onset)} \\ f_N \mathbf{\Gamma}_d(\mathbf{v}), & \text{if } \|\mathbf{v}\| > \delta \quad \text{(sliding)} \end{cases} \tag{20}$$

where $\mathbf{f}_{\text{ext}} \in \mathbb{R}^3$ is the external force that the friction patch must resist to remain sticking. In robot throwing, the external force comes from the gravity acting on the object and the inertial forces of the rotating and accelerating object.

C. Release Dynamics With Rigid Body Dynamics and LS Friction

The rigid body dynamics expressed in the object frame is

$$\mathbf{M}\mathbf{a}^o = \mathbf{G}^\top(\mathbf{q}^r)\mathbf{f}^c + \mathbf{g} \tag{21}$$

where $\mathbf{M} = \text{diag}(m, m, I)$ denotes the mass matrix of the object, $\mathbf{g} = [0, -g, 0]^T$ represents the gravitational wrench acting on the object in the vertical plane, and $\mathbf{f}^c$ is the frictional wrench applied on the object.

1) Sticking Dynamics: During sticking, finger acceleration $\mathbf{a}^h$ equals the acceleration of the contact patch on the object $\mathbf{a}^c$, i.e., relative acceleration $\mathbf{a}^r$ is zero. Thus, we obtain the object acceleration $\mathbf{a}^o$ by setting $\mathbf{a}^c = \mathbf{a}^h$ in (15), as follows:

$$\mathbf{a}^o = \begin{bmatrix} a_x^o \\ a_z^o \\ \alpha^o \end{bmatrix} = \mathbf{a}^h - \begin{bmatrix} \alpha^o z^r + \omega^{o2} x^r \\ -\alpha^o x^r + \omega^{o2} z^r \\ 0 \end{bmatrix} \tag{22}$$

and the *required* friction wrench $\mathbf{f}^p$ to achieve this acceleration to remain sticking

$$\mathbf{f}^p = \mathbf{G}^\top(-\mathbf{q}^r)\left(\mathbf{M}\mathbf{a}^o - \mathbf{g}\right). \tag{23}$$

If $H(\mathbf{f}^p) \le F_N^2$, then the required friction load lies within the LS, and hence, friction wrench $\mathbf{f}^c = \mathbf{f}^p$.

2) Sliding Onset: Continue integrating the sticking dynamics and as normal force F_N gradually decreases until $H(\mathbf{f}^p)$ just bypass F_N^2, sliding onsets. Following the Karnopp routine in (20), we obtain

$$\mathbf{f}^c = \frac{f_N}{\sqrt{H(\mathbf{f}^p)}}\mathbf{f}^p \tag{24}$$

where $\mathbf{f}^p$ is determined from (22) and (23). The object acceleration is then obtained from the rigid body dynamics as follows:

$$\mathbf{a}^o = \mathbf{M}^{-1}\left(\mathbf{G}^\top(\mathbf{q}^r)\mathbf{f}^c + \mathbf{g}\right). \tag{25}$$

3) Sliding Dynamics: During sliding, the LS determines the sliding wrench via $\mathbf{f}^c = f_N \mathbf{\Gamma}(\mathbf{v}^r)$, and object acceleration is given by (25).

TABLE I
PARAMETERS IN THE DROPPING SIMULATION

Parameter	Value
object mass m	0.2 kg
object CoM offset h	0.1 m
object radius of gyration ρ	0.05 m
gravity constant g	9.81 m/s^2
friction coefficient μ	0.8
radius of the contact patch r	0.0075 m
integration step size dt	10^{-4} s
Karnopp deadzone δ	10^{-8} m/s

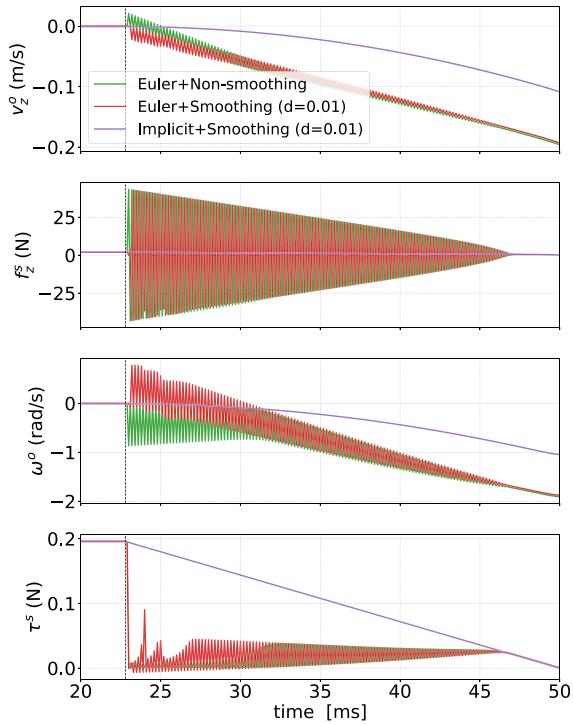


Fig. 6. Simulation result for object dropping. The vertical dashed lines indicate the instant of sliding onset.

D. Case Study on Object Dropping

Consider a bar with mass m, horizontally pinch-grasped by parallel fingers with a CoM offset by h. As the grasp normal force gradually decreases during release, the bar will first rotate around the finger pad and then slide downward. When the gripper's normal force vanishes, the bar will free-fall while rotating around its CoM. We examine whether the release dynamics model, based on LS friction and rigid body dynamics, can predict this behavior.

Table I summarizes the simulation parameters. The normal force is set to start at 50 N, decrease linearly, and reach zero at 50 ms. By running the aforementioned release dynamics model with different solving configurations, we obtain the trajectories of object twist and friction wrench in the vertical and rotation directions, as shown in Fig. 6. For models with viscous smoothing, the half-band d is set to 0.01 m/s. The LSODA solver [47] is used for implicit integration. Notably, all the terminal states of the object are plausible (falling downward with negative rotation).

1) Euler Integration: As shown in Fig. 6, both the nonsmoothed and the smoothed models with Euler integration exhibit

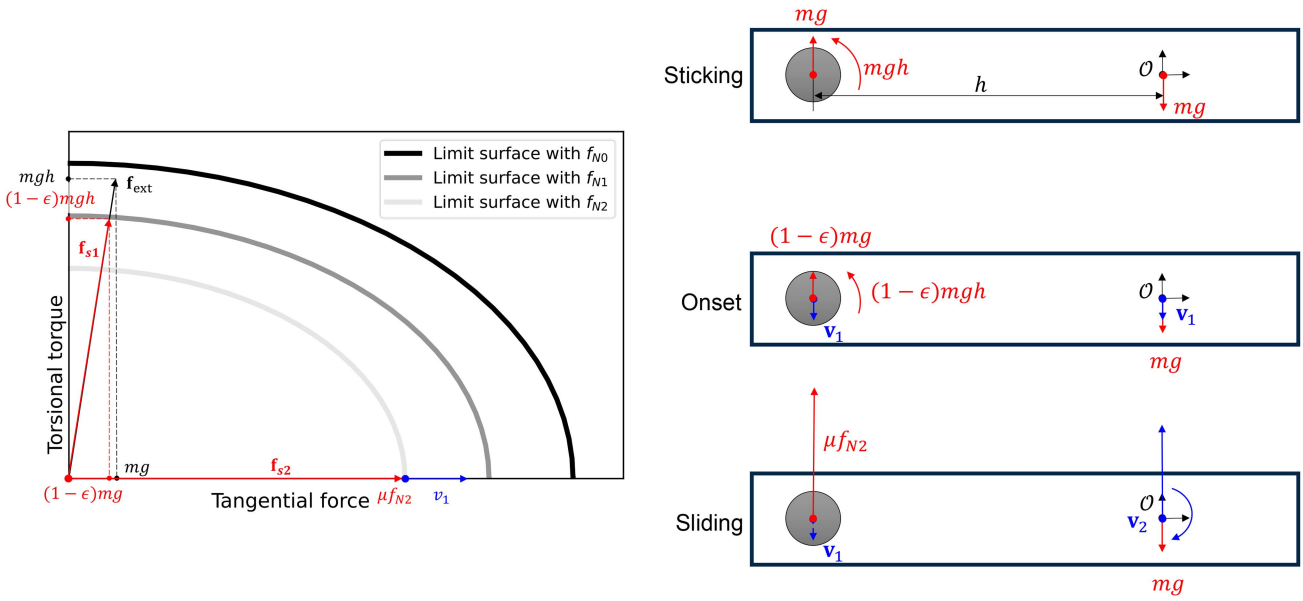


Fig. 7. Illustration of dropping simulation upon onset with LS friction and rigid body dynamics.

frequent oscillations in velocities and patch forces and result in identical object detach motion. To understand this oscillatory behavior, we next closely examine the instant of onset in the nonsmoothed model. As shown in Fig. 7, during sticking, to keep the bar remaining static, the contact patch must provide tangential friction mg and torsional friction torque mgh, i.e., $\mathbf{f}_{\text{ext}} = [0, mg, mgh]$. As the normal force decreases to f_{N1}, the friction patch cannot provide enough wrench to resist the external load due to gravity, leading to the onset of sliding. At this instant, according to the Karnopp routine, the friction wrench $\mathbf{f}_{s1}$ is $[0, (1-\epsilon)mg, (1-\epsilon)mgh]$, where $\epsilon = 1 - \frac{f_{N1}}{\sqrt{H(\mathbf{f}_{\text{ext}})}}$. Then, net wrench applied on the object is then $[0, -\epsilon mg, 0]$, resulting in a small downward twist $\mathbf{v}_1$ at this time step. Notably, the pure downward net wrench and acceleration persist throughout the onset phase, regardless of the value of δ.

At a future time step, with a normal force of f_{N2} after bypassing the δ dead zone, the system switches to the sliding mode, where the friction wrench is determined *only* by normal force and sliding twist direction, irrespective of the sliding twist magnitude. In this case, the sliding twist $\mathbf{v}_1$ is a pure translation, causing the sliding friction wrench $\mathbf{f}_{s2}$ to be a purely upward force, i.e., $\mathbf{f}_{s2} = [0, \mu f_{N2}, 0]$, which remains large due to the significant f_{N2}, compared to the object's weight. This explains the sudden jump of vertical patch force upon sliding onset shown in Fig. 7. At this instant, the net wrench is $[0, \mu f_{N2} - mg, -\mu f_{N2} h]$, leading to a large angular acceleration that causes the contact patch to move upwards. In the next time step, however, this upward contact velocity results in a large downward friction force, causing the contact patch to move downward. Consequently, oscillation occurs.

Although the model "Euler+Nonsmoothing" can qualitatively render a plausible detach motion for pen-dropping, the Zeno-like behavior raises concerns when applying this model to estimation, planning, and control for robot throwing.

1) *Estimation:* For a typical robotic system with irregular sampling time, the nonsmooth dynamics may be sensitive to communication delays, leading to poor model accuracy.
2) *Planning:* Numerical optimal control for nonsmooth dynamics requires sophisticated treatments for discontinuities, such as softmax smoothing. However, these treatments often resolve numerical difficulties at the expense of physical fidelity, as discussed in [48].
3) *Control:* For a torque-controlled robot, it is impossible to compensate for the oscillating frictional wrench, predicted by the release dynamics model with LS and rigid body dynamics.

Indeed, the observed rapid oscillations are numerical artifacts arising from the time discretization of ill-posed or nonsmooth friction models. These oscillations persist even with smaller integration time steps and should not be interpreted as approximations of the continuous-time dynamics. While running the Karnopp routine throughout the release window without switching to the sliding mode can suppress this Zeno-like behavior, the object would exhibit no rotation upon entering free fall, rendering the motion physically inaccurate.

2) Implicit Integration: The implicit integrator for the nonsmoothed model failed to converge and is, therefore, not shown in Fig. 6. On the other hand, the recipe of a viscous-smoothed LS with implicit integration yields a smooth simulation result. Interestingly, the smoothed model solved using the implicit integrator (purple) does not merely average out the oscillations observed with the explicit integrator (green and red); instead, it yields a drastically different transient motion. The black box nature of the implicit integrator obscures our understanding of this behavior, motivating the alternative analysis presented in the next section.

3) LS With Different Shapes: To illustrate the fundamental limitation of velocity-magnitude-independent friction wrenches, we also apply a richer LSs described by a rotated ellipsoid (nondiagonal $\mathbf{A}$ in (19) [33] and higher order homogeneous polynomial functions [44], extending beyond the diagonal quadratic shape. The investigated LSs are shown in Fig. 8(a). Note that, unlike the quadratic (elliptic) case, where the friction wrench admits a closed-form solution, higher order polynomial LSs do not yield an analytical expression for the

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

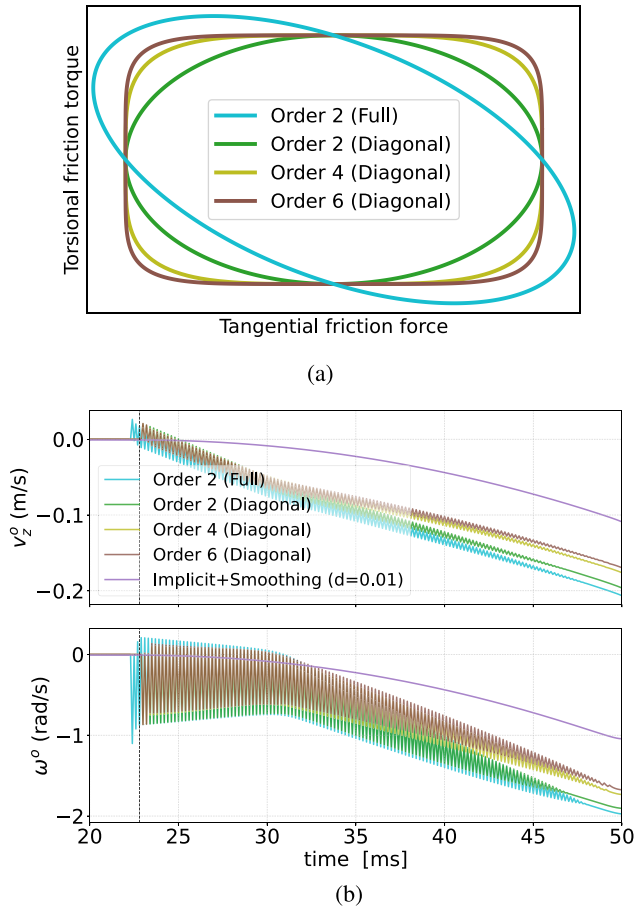


Fig. 8. Simulation result for object dropping with different shaped LSs. (a) Illustration of LS shapes. (b) Simulation result.

friction wrench given sliding velocity. Instead, we adopt the iterative method proposed in [44], which finds the friction wrench at which the surface normal aligns with the sliding velocity direction.

The dropping simulation results for these LSs with different shapes (without viscous smoothing), solved via Euler integration, are shown in Fig. 8(b), where oscillatory behavior is also observed.

E. On Dynamic Patch Friction Models

In the future, a higher fidelity dynamic friction model can be used for better accuracy. For example, Waltersson and Karayiannidis [49] propose a dynamic LS model integrated from the local LuGre friction model [50], [51], instead of the local Coulomb friction model. Unlike the Coulomb friction model, which was originally designed to characterize steady-state sliding, the LuGre friction model describes friction as a consequence of microscopic bristle deformation, making it potentially more accurate and versatile than the Karnopp approach. However, since no parameter identification procedure for LuGre friction is provided in [49], we leave its application to release dynamics as future work. The following points are worth noting.

1) Although the LuGre friction model may improve the continuity of sliding dynamics due to its differentiable nature, the release dynamics may still be *stiff*, stemming from the high stiffness (on the order of 10^6 in the simulation experiments of [49]) of the bristle. This stiffness could result in a similarly high sensitivity of the frictional wrench with respect to the sliding twist during the release. In [49], an implicit integrator is used to ensure numerical stability.
2) The viscous-smoothed LS model resembles the "damping" term of the microscopic bristle dynamics in the LuGre model.

V. Transient Dynamics With Sliding Pivot

In this section, we propose a surrogate physical model of *SP* to describe the release dynamics. As the name suggests, we model the release process as the object pivoting around the finger pad, with the pivoting point first sticking and then sliding on the contact surface. This model can be regarded as a physical regularization of the nonsmooth dynamics of planar sliding.

A. Physical Grounding: Stress Analysis via Overhanging Beam

In the previous section, we observed that the classical patch friction model, the LS, is highly sensitive in the transient release regime. Specifically, when the sliding twist is small, the sliding direction of the patch can change drastically due to the large friction wrench at the beginning of sliding, which, in turn, leads to a drastic change in the friction wrench. The velocity-magnitude-independent friction wrench induces oscillatory behavior in the transient dynamical system. Although such oscillations can be suppressed by viscous smoothing and the use of implicit integration, this is not sufficient to explain the empirical phenomenon of "temporal hinge" during release.

In typical LSs, the frictional interface is modeled as a continuous patch, and the entity of interest is the $SE(2)$ sliding motion and friction wrench at the center of pressure. Such a holistic view does not capture the microscopic force–motion relationship during the transient phase. In this regard, we attempt to analyze and better understand the transient behavior using an overhanging beam model, where two point contacts sustain both the tangential and torsional friction during sticking. As shown in Fig. 9, the bar is initially held horizontally and supported at two contact points C_1 and C_2, with friction forces $f_1, f_2 \in \mathbb{R}^2$. The positive directions of all force components are defined w.r.t. the global XZ frame, as illustrated in Fig. 2. Assuming a uniform pressure distribution, we have $f_{N1} = f_{N2} = f_N/2$. Let $f^C_{\max}(t) = \mu f_N(t)/2$ denote the time-varying friction limit at each contact.

1) Sticking: Static analysis gives

$$\begin{cases} f_{1x} + f_{2x} = 0 \\ f_{1z} + f_{2z} = mg \\ -f_{1z}r + f_{2z}r = mgh \end{cases} \Rightarrow \begin{cases} f_{1x} = f_{2x} = 0 \\ f_{1z} = \frac{1}{2}mg\left(1 - \frac{h}{r}\right) \\ f_{2z} = \frac{1}{2}mg\left(1 + \frac{h}{r}\right). \end{cases}$$

Remarks:

1) The system is hyperstatic, as the horizontal force is underdetermined. We apply the common minimum norm principle [40], which yields zero horizontal friction force at the two contact points.
2) In eccentric dropping/throwing, $h \gg r$, and therefore, the contact stress at C_2 is much larger than the weight of the object. This explains why the contact patch begins sliding before the normal force f_N decreases to mg/μ, the limit required to sustain the object's weight.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

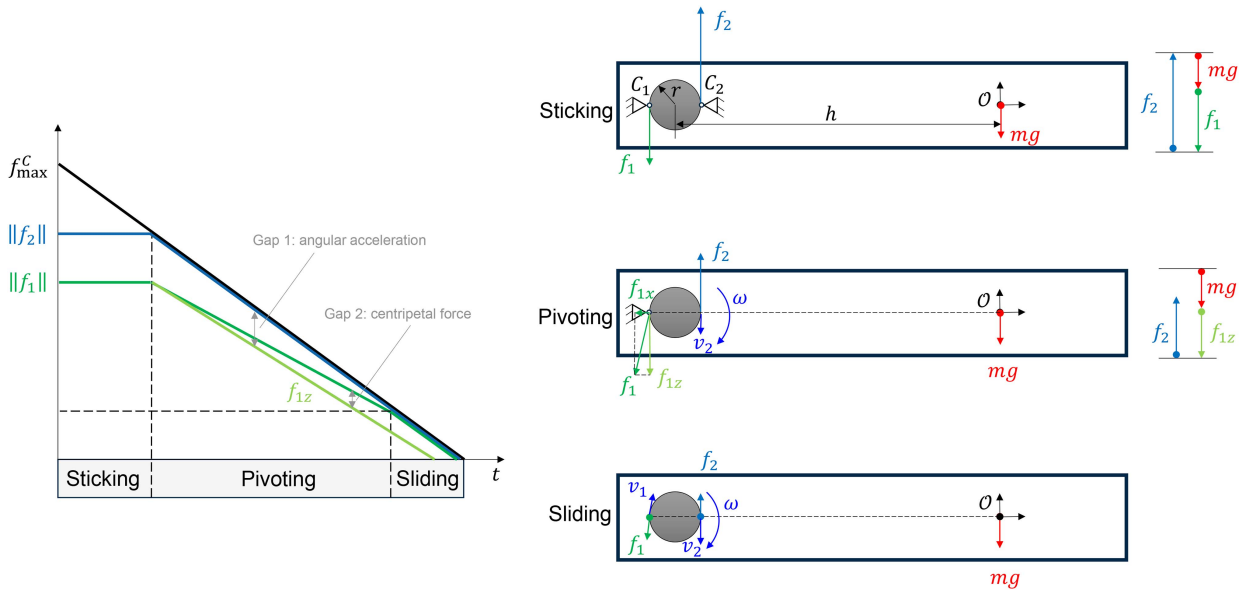


Fig. 9. Illustration of dropping simulation with two-point contact and rigid body dynamics.

3) Since $\|f_1\| < \|f_2\|$, C_2 will always start sliding, while C_1 remains sticking. Effectively, a temporal hinge around C_1 is established when $f^C_{\max}(t) = \frac{1}{2}mg(1 + \frac{h}{r})$.

2) Pivoting: In the short window of transient (∼50 ms), we assume that the sliding displacement is negligible to ease our discussion. After the pivot around C_1 is established, as the normal force keeps decreasing, according to the Coulomb friction law, $\|f_2\| = f^C_{\max}(t) < \frac{1}{2}mg(1 + \frac{h}{r})$. Then, the net torque applied at C_1 is

$$\begin{aligned}\tau^{C_1} &= -mg(h + r) + 2rf_{2z} \\ &< -mg(h + r) + 2r\frac{1}{2}mg\left(1 + \frac{h}{r}\right) = 0.\end{aligned}$$

In other words, the angular acceleration is negative (clockwise).

The net torque applied to the object's CoM is

$$\tau^o = f_{1z}(h + r) + f_{2z}(h - r).$$

Since the angular acceleration is negative, we have

$$\tau^o \le 0 \Rightarrow |f_{1z}| < \frac{h - r}{h + r}\|f_2\| < f^C_{\max}(t).$$

This implies that the vertical hinge force at C_1 remains below the friction limit $f^C_{\max}$. Therefore, the temporal hinge around C_1 can be sustained for an extended period once it is established, until the bar's angular velocity becomes large enough such that $\|f_1\| > f^C_{\max}(t)$, due to the contribution of centripetal force.

3) Sliding: Once the hinge force at C_1 also exceeds the friction limit, the frictional contact patch transitions to spin-sliding, with each contact point following the Coulomb friction law.

Discussion:

1) The analysis via the two-point overhanging beam model gives a qualitative explanation to the "temporal hinge" phenomenon, without requiring numerical simulation. The behavior also resembles the "detachment wave" observed in real-world rubber–glass sliding onset experiments [52].
2) The LS constructed from finite contact points admits facets [4], leading to nonunique frictional wrenches for sliding twists that are normal to the facets. The uniqueness of the solution to the sliding dynamical system with such LSs is discussed in [53, Sect. 3], which is analogous to an implicit integration scheme. In contrast, our per-contact-point analysis renders the transient release behavior more transparent.
3) The pivoting dominance of the release motion resembles sliding eigendirections, which have been studied extensively in contact mechanics [43], [45], [53], [54]. However, prior discussions have focused on scenarios with a *constant normal force*. In comparison, we observe that the sustained pivoting in release dynamics arises from the interplay between the *vanishing normal force*, the nonsmooth Coulomb friction law at the two contact points, and rigid-body dynamics.

The justified "temporal hinge" phenomenon from the overhanging beam analysis motivates the development of an SP model. Unlike the LS-based model, which computes planar sliding motion in $SE(2)$ from the instantaneous sliding wrench and rigid-body dynamics, the SP model constrains the motion to progress through *sticking–pivoting–sliding* phases. This sequence regularizes the relative motion between the finger and the object, while capturing the dominant transient phenomena.

B. Decoupled Friction Limit

The SP model differs from the LS-based model in its *decoupled* treatment of sliding and spinning on the contact patch, where the tangential friction limit $f_{\max}$ and torsional friction limit $\tau_{\max}$ are represented as separate functions of normal force f_N, with $f_{\max}(t) = \mu f_N(t)$ and $\tau_{\max}(t) = cr\mu f_N(t)$.

C. Obtaining External Load From Pivot Dynamics

Compared to the model in the previous section, where the external load (to remain sticking) is determined by matching the

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

contact patch's linear and angular acceleration with the hand's and then solving the rigid body dynamics in (21), here for the pivoting dynamics model, only the patch's linear acceleration needs to match that of the hand to satisfy the pivoting constraint, leading to a *constraint dynamical system* with one-degree-of-freedom (DoF) pivoting as the admissible relative motion.

Mathematically, the object's force–motion relationship during sticking–pivoting satisfies the following linear equations:

$$\mathbf{M}\mathbf{a}^o = \mathbf{M}\begin{bmatrix} a_x^h - \alpha_{\text{fic}}^o z^r - \omega^{o2} x^r \\ a_z^h + \alpha_{\text{fic}}^o x^r - \omega^{o2} z^r \\ \alpha_{\text{fic}}^o \end{bmatrix} = \mathbf{G}^\top(\mathbf{q}^r)\begin{bmatrix} f_x^p \\ f_z^p \\ \tau^p \end{bmatrix} + \mathbf{g} \tag{26}$$

with four unknowns: hinge reaction force $[f_x^p, f_z^p]$, torsional friction τ^p, and a fictitious object angular acceleration at pivoting α_{fic}^o. Equation (26) can be written as follows:

$$f_x^p = m(a_x^h - \alpha_{\text{fic}}^o z^r - \omega^{o2}) \tag{27}$$

$$f_x^p = m(a_z^h + \alpha_{\text{fic}}^o x^r - \omega^{o2} + g) \tag{28}$$

$$\alpha_{\text{fic}}^o = \frac{m(a_x^h z^r - a_z^h x^r) - mgx^r + \tau^p}{(mh^2 + I)} \tag{29}$$

where I is the rotational inertia in the throwing plane and $h = \|[x^r, z^r]\| = \|[x^o - x^h, z^o - z^h]\|$ is the CoM offset. Effectively, $(mh^2 + I)$ is the reflected rotational inertia at the pivot, according to the parallel axis theorem. Note that the calculation of hinge reaction forces is decoupled from the rest two variables τ^p and α_{fic}^o. Let $\alpha_{\text{free}}^o = \frac{m(a_x^h z^r - a_z^h x^r) - mgx^r}{(mh^2+I)}$, denoting the object's free pivoting acceleration in the absence of torsional friction; then

$$\tau^p = (mh^2 + I)(\alpha_{\text{fic}}^o - \alpha_{\text{free}}^o). \tag{30}$$

For the pivot to remain sticking without relative rotation to the hand, the required torsional friction can be obtained by setting $\alpha_{\text{fic}}^o = \alpha^h$

$$\tau^p = (mh^2 + I)(\alpha^h - \alpha_{\text{free}}^o) \tag{31}$$

$$f_x^p = m(a_x^h - \alpha^h z^r - \omega^{o2}) \tag{32}$$

$$f_x^p = m(a_z^h + \alpha^h x^r - \omega^{o2}). \tag{33}$$

During sticking, the solution is identical to the sticking dynamics in (22). During pivoting, real-world measurement noise in hand acceleration is amplified by the denominator in (29)—the reflected inertia, which is small in scale—leading to spurious hinge reaction forces via (27) and (28). Therefore, we choose to calculate the external torsional load at the frictional interface using (31) throughout the release window, which regularizes the hinge reaction forces and thus mitigates oscillatory mode switching between pivoting and sliding.

D. Release Dynamics With Sliding Pivot

1) Sticking Dynamics: If $|\tau^p| \le \tau_{\max}$ and $\|[f_x^p, f_z^p]\| \le f_{\max}$, the object is sticking to the hand and its motion can be determined by (22), i.e., propagating hand motion with the object be the extended body.

2) Pivoting Dynamics: If $|\tau^p| > \tau_{\max}$ and $\|[f_x^p, f_z^p]\| \le f_{\max}$, then the friction patch cannot generate enough torsional friction to resist spinning but can generate enough linear friction to prevent the contact patch from sliding. In this case, the system becomes a one-DoF pivot around the finger pad. This pivoting can be conveniently described by the state of relative rotation $(\theta^r, \omega^r) = (\theta^o - \theta^h, \omega^o - \omega^h)$, with pivot acceleration α^r being

$$\alpha^r := \dot{\omega}^r = \frac{\text{sgn}(|\tau^p|)(|\tau^p| - \tau_{\max}(f_N))}{mh^2 + I}. \tag{34}$$

Recall that the external torsional load is determined via (31), implying the Karnopp cutoff routine is applied throughout the sticking–pivoting phase. With this, (32) and (33) capture the enlarged tangential friction load with enlarged angular velocity, while ignoring the variations in α_{fic}^o, as shown in (27) and (28).

Note that sticking dynamics and pivoting dynamics can be written in the following compact form:

$$\alpha^r = \dot{\omega}^r = \frac{\text{sgn}(|\tau^p|)(\max\{|\tau^p|, \tau_{\max}\} - \tau_{\max}(f_N))}{mh^2 + I}. \tag{35}$$

3) Sliding Dynamics: If $\|[f_x^p, f_z^p]\| > f_{\max}$, the pivot starts sliding on the contact surface. The sliding dynamics is given by

$$\begin{bmatrix} a_x^r \\ a_z^r \end{bmatrix} = \frac{1}{m}\left(\begin{bmatrix} f_x^p \frac{f_{\max}}{\|[f_x^p, f_z^p]\|} \\ f_z^p \frac{f_{\max}}{\|[f_x^p, f_z^p]\|} \end{bmatrix} + \mathbf{g}_{xz}\right) \tag{36}$$

where $[a_x^r, a_z^r]^\top$ is the sliding acceleration, $\mathbf{g}_{xz} = [0, -g]^\top$. Pivot acceleration is also influenced by the decreased lever torque due to the insufficient friction force to maintain pivoting. Mathematically, α^s is the amount of decreased pivot acceleration, given by

$$\alpha_s = \frac{\begin{bmatrix} -x^r \\ -z^r \end{bmatrix} \times \begin{bmatrix} f_x^p(1 - \frac{f_{\max}}{\|[f_x^p, f_z^p]\|}) \\ f_z^p(1 - \frac{f_{\max}}{\|[f_x^p, f_z^p]\|}) \end{bmatrix}}{mh^2 + I}. \tag{37}$$

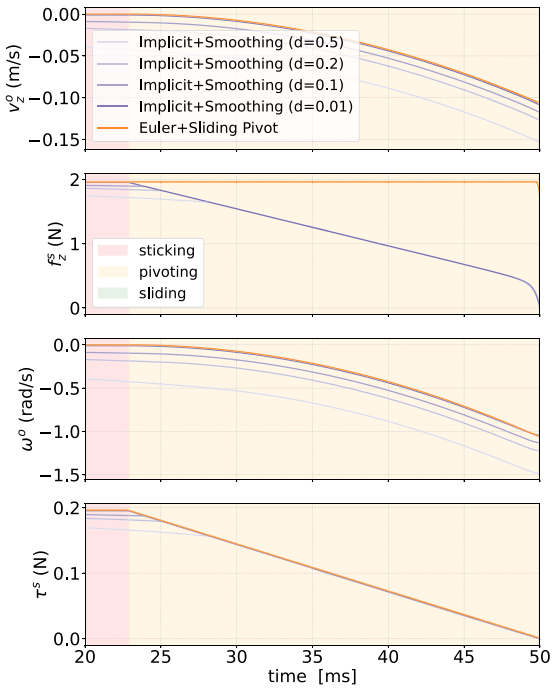
As a result, the pivot acceleration during sliding is

$$\alpha^r = \dot{\omega}^r = \frac{\text{sgn}(|\tau^p|)(\max\{|\tau^p|, \tau_{\max}\} - \tau_{\max}(f_N))}{mh^2 + I} - \alpha^s. \tag{38}$$

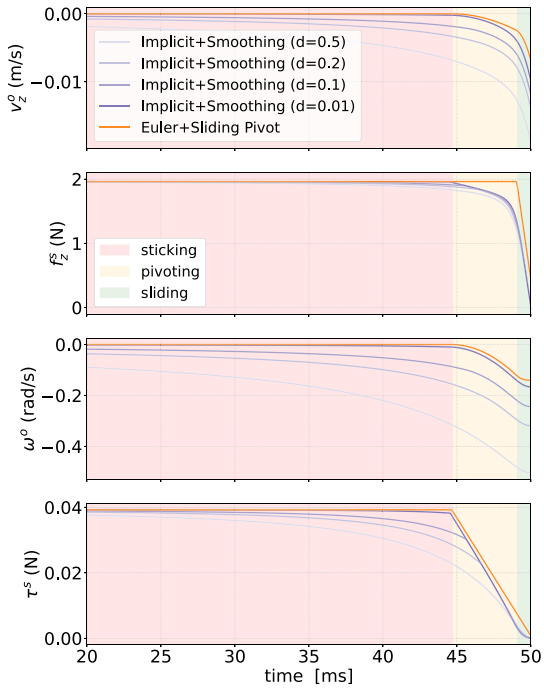
E. Numerical Comparison With LS

To justify the SP surrogate model, we rerun the bar-dropping experiment using the SP model solved via Euler integration (denoted as "Euler+Sliding Pivot"), alongside a family of smoothed LS models with different smoothing parameters d, solved via implicit integration (denoted as "Implicit+Smoothing"). The results are shown in Fig. 10(a). Recall that the parameter d (unit: m/s) defines the width of the velocity "creep band" over which the transition from sticking to sliding is smoothed. Smaller values more closely approximate the original nonsmoothed model. We demonstrate a range from 0.05 to 0.01 m/s based on convergence behavior—below 0.01 m/s (e.g., at 0.001 m/s), results show no discernible difference. As the smoothing parameter decreases, we observe the following.

1) For the bar detachment motion (v_Z^o and ω^o) and the torsional friction torque (τ^s), the results of "Implicit+Smoothing" is converging toward those of "Euler+Sliding Pivot."
2) The tangential friction force exhibits a larger discrepancy, which is expected, as the SP model neglects the coupling between torsional friction and tangential friction. However, in this case, the bar's vertical CoM velocity



(a)



(b)

Fig. 10. Comparison of simulation results: SP versus viscous-smoothed LS. The colored area indicates the modes of the SP model. (a) 0.1-m CoM offset (same as in Fig. 6). (b) 0.02-m CoM offset.

v_z^o is dominated by the contribution from pivoting—i.e., the propagation of angular velocity from the pivot to the CoM—while the effect of sliding is minimal. As a result, this simplification has only a minor impact on the object's motion.

We also run the dropping simulation with a much smaller CoM offset (2 cm). From Fig. 10(b), we obtain the following results.

1) The onset angular velocity (-0.15 rad/s) is much smaller than for a 10-cm CoM offset (-1.0 rad/s), which is consistent with physical intuition.
2) There is a similar convergence behavior of the smoothed LS family; however, the detachment twist exhibits a small bias compared to the SP model in the small CoM offset regime. The bias arises from two factors: a) the longer "Sliding" phase in the small CoM case causes the difference between the coupled friction wrench in ILS and the uncoupled friction wrench in SP to accumulate and b) unlike the large CoM offset case, here, the shorter lever arm reduces the pivoting-induced velocity at the CoM, making the differences on the object CoM motion between the two models more pronounced.

In summary, recall that "Implicit+Smoothing" demonstrates different results compared to "Euler+Smoothing/Non-smoothing" (as shown in Fig. 6). The consensus demonstrated here in Fig. 10 makes the two models ("Euler+Sliding Pivot" and "Implicit+Smoothing") strong candidates for capturing the reality.

VI. Model Validation Experiments

In this section, we conduct real robot throwing experiments to systematically assess the accuracy of three candidate release models: 1) nonsmoothed axis-aligned quadratic limit surface with Euler integration ("LS"); 2) sliding pivot with Euler integration ("SP"); and 3) viscous-smoothed axis-aligned quadratic limit surface with implicit integration and a smoothing factor of $d = 0.01$ m/s ("ILS"). We design two sets of experiments.

1) *Batch Experiments:* To assess the overall capability of the models in capturing the transient dynamics, we select a mesh of throwing conditions that covers a large space of landing poses. 36 experiment conditions (two CoM positions, three throwing pitch angles, and six release motion) with each condition repeated five times, resulting in 180 throws.
2) *Factor Analysis:* Vary one throwing condition at a time, and examine if the model can capture the distribution shift of the throwing outcomes due to the "axis-wise" change of throwing conditions. 6 throwing conditions are varied and studied, including three conditions that are not considered in the batch experiments for completeness [surface friction, mass, and moment of inertia (MOI)].

A. Experimental Setup

1) Hardware: The throwing experiments are conducted by a seven-DoF fully actuated manipulator (Franka Emika Panda) mounted with Robotiq 2F-85 parallel gripper (60A black silicone fingertip). To measure the vanishing normal force during gripper opening, an ATI Nano 17 F/T sensor is attached behind one of the gripper's finger pads. The experimental hardware setup is illustrated in Fig. 11. The thrown object is a 3-D printed bar (material: ABS) with a known and configurable mass distribution attached with markers, shown in Fig. 12. The physical parameters of the configured bar are summarized in Table II.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

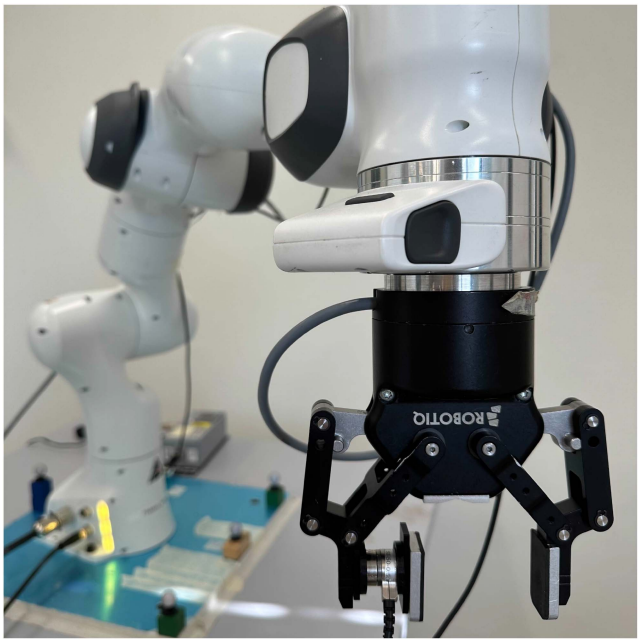


Fig. 11. Hardware setup: Franka Emika Panda manipulator with Robotiq 2F-85 gripper. An ATI Nano 17 F/T sensor is mounted behind one finger pad.

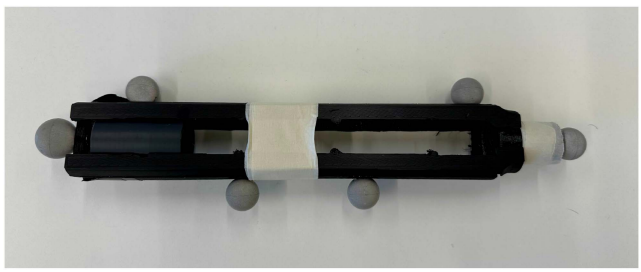


Fig. 12. Thrown object: a 3-D printed bar with configurable mass distribution attached with markers.

TABLE II
PARAMETERS OF THE THROWN BAR CONFIGURED IN CASE STUDY AND BATCH EXPERIMENTS

Parameter	Value
mass m	0.24 kg
bar length l	0.26 m
CoM offset from tip h	0.10/0.16 m
moment of inertia I	535/802 kg · mm^2
width of the finger patch	0.015 m

2) Throwing Motion Generation: The robot is controlled using a joint impedance controller running at 500 Hz. As illustrated in Fig. 13, the throwing motion comprises two stages: LAUNCH and RELEASE. The LAUNCH stage brings the robot from stationary to a high-energy high-velocity state. In this stage, the desired joint trajectory is generated from PolyMPC [55], a trajectory optimizer with dynamic feasibility constraints and time-optimal objective, given the initial robot state and nominal throwing state. The RELEASE stage is a short segment of constant-acceleration trajectory, explicitly designed to modulate

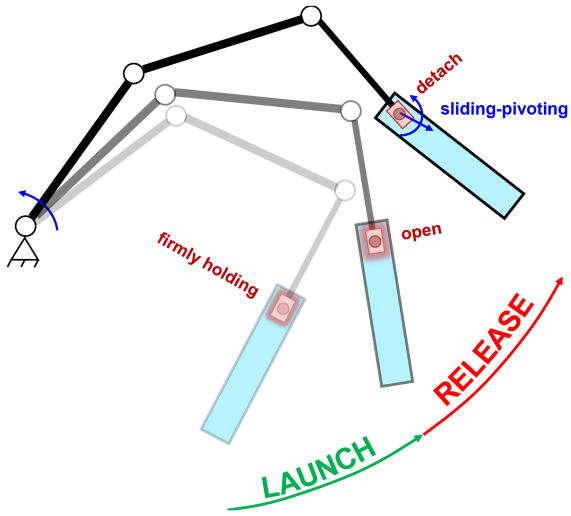


Fig. 13. Schematic of robot throwing motion in the experiments.

the robot's motion during release. The gripper opening command is given 50 ms before entering the RELEASE stage to account for the gripper's dwell time.

3) Data Recording and Processing: We use ROS [56] to broadcast and record all the data from different hardware. The robot joint position trajectory is recorded at 500 Hz and filtered through a Butterworth filter with 35-Hz cutoff frequency. Robot joint velocity and acceleration are obtained from cubic spline interpolation of the filtered position data. The F/T sensor measurement is recorded at 1000 Hz. The object's pose is tracked by an OptiTrack motion capture system at 250 Hz with a spatial accuracy of 0.2 mm. The object velocity is obtained by differentiating a Savitzky–Golay filter with a window length of 21.

4) Collected Batch Experimental Data: For the batch of robot throwing data, the throwing conditions are configured as follows.

1) *Two CoM offsets:* A metal cylinder payload (mass: 154 g) is configured at two different slots in the 3-D printed bar (mass: 92 g), yielding two distinct CoM offsets from the tip ("CoM-1"—10 cm; "CoM-2"—16 cm).
2) *Three throwing pitch angles:* There are three different nominal throwing postures with distinct pitch angles (θ^h upon release).
3) *Six release accelerations:* There are six different release motions ranging from acceleration to deceleration.

Each (*CoM*, *pitch angle*, and *release acceleration*) tuple is thrown five times, resulting in 180 throws. The collection of landing poses is shown in Fig. 14, demonstrating that the experiment design covers a wide range of landing poses. Considering that landing pose error amplifies linearly with flight time in projectile flight dynamics, we report the flight time statistics of our dataset to facilitate future study comparisons: the mean flight time is 0.6 s, with a standard deviation (STD) of 0.06 s.

B. Parameter Identification

In all three candidate models ("LS," "SP," and "ILS"), the friction coefficient μ and the radius of contact r are to be estimated. Following the pioneering work on validating impact models of robotics simulators [57], we identify model-specific contact

TABLE III
PARAMETER IDENTIFICATION RESULTS

Model	Min Mean Error [m]	Max Mean Error [m]	Optimal (μ^*, r^* [mm])	$\mu^* \cdot r^*$	Solving Time Per Throw [ms] (mean±std)
LS	0.159	6.639	(0.6, 12.0)	7.20	48.37±3.77
SP	0.060	0.091	(1.0, 2.5)	2.50	43.42±3.01
ILS	0.051	0.086	(0.3, 8.5)	2.55	966.82±2260.47

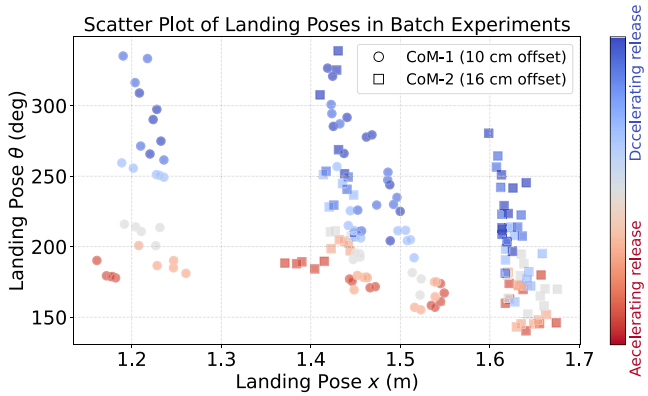


Fig. 14. Scatter plot of landing poses in batch experiments. The marker shape indicates the two bar CoM configurations tested. The color bar represents the family of release motions, ranging from acceleration to deceleration. To avoid clutter, the three pitch angles are not explicitly labeled in the scatter plot but can be inferred from the three vertical clusters of landing poses.

parameters that best fit the experimental data, to examine each model's capacity to describe the reality. To mitigate concerns about parameter "overfitting," we conduct a sensitivity analysis to understand the error landscape around the optimal contact parameters.

1) Evaluation Metric: In the context of robot throwing, we are primarily interested in the landing position and orientation. Therefore, we define the error metric as the following *scaled error:* For a ground-truth landing pose $\mathbf{q}^{\text{land}} = [x^{\text{land}}, z^{\text{land}}, \theta^{\text{land}}]^\top$ and a model-predicted landing pose $\hat{\mathbf{q}}^{\text{land}} = [\hat{x}^{\text{land}}, \hat{z}^{\text{land}}, \hat{\theta}^{\text{land}}]^\top$

$$e(\mathbf{q}^{\text{land}}, \hat{\mathbf{q}}^{\text{land}}) = \left|x^{\text{land}} - \hat{x}^{\text{land}}\right| + \left|\theta^{\text{land}} - \hat{\theta}^{\text{land}}\right| \frac{l}{2} \tag{39}$$

where l is the length of the bar. The scaled rotation error provides a comparable scale to the position error, measured in meters, which yields an interpretable error value.

2) Sensitivity Analysis: We perform a sensitivity analysis by calculating the mean scaled error (over 180 throws) of a 20×20 grid of μ and r: $\mu \in [0.25, 1.2]$ with 0.5 intervals, $r \in [2.5, 12]$ mm with 5-mm intervals. The heatmap of mean scaled landing error is shown in Fig. 15, and the parameter identification results are presented in Table III. The computation is conducted on a desktop PC with an Intel i9-12900K CPU.

Discussion:

1) *Mean scaled error:* For "SP" and "ILS," the errors are of similar scale, with "ILS" achieving 5–15% smaller error than "SP." In contrast, "LS" yields much larger errors (its minimum is even larger than the maximum of the other two models across the parameter space) and is highly sensitive (the white squares in the left panel of Fig. 15 indicate contact parameters that produce spurious predictions compared to neighboring parameter values).
2) *Error landscape:* "ILS" exhibits a well-organized landscape with a single isolated minimum region (hatched area with deep purple color). However, the error is only sensitive along the $\mu \cdot r$ direction, which means that the error remains flat for families of contact parameters with the same torsional friction limit. "SP" captures a similar phenomenon, with the minimizer shifted to a larger μ. At the optimal parameter, $\mu^* \cdot r^*$ attains a similar value for "SP" and "ILS" (2.50 versus 2.55). Although "SP" exhibits a local maximal region at medium $\mu \cdot r$ values (the bright stripe in the middle of the SP heatmap in Fig. 15), its influence on parameter identification is minor, and the lowest 10% error region remains consistent with "ILS." In comparison, the "LS" model exhibits a strong local minimum in the upper-middle region, which shifts the optimal parameter far from the minimizer region of "ILS."
3) *Solving time:* The lowest error of "ILS" among the three models is achieved at the cost of approximately 20× higher computation time due to its use of an implicit integrator.[1] Notably, Fig. 15 was generated in 22 h for 3 models × 180 throwing samples × 400 contact parameters = 216 000 predictions, with the majority of the computation time spent on "ILS." Moreover, the computation time of implicit integrators is also less predictable, as shown by the large STD in solving time, which may pose challenges for scheduling downstream tasks such as optimal control or reinforcement learning for throwing motion generation.

With the optimal parameters identified for each model, we next examine their predictions.

C. Case Study

We compare the predictions of the three release dynamics models through a case study, summarized in the collage shown in Fig. 16. Fig. 16(a) illustrates the model predictions alongside the real trajectory of the object within a 700-ms window after giving the gripper opening command. This window includes in-hand sticking, spinning-sliding, free-flying, and finally, touchdown. In the "Linear Vel." and "Angular Vel." plots, the constant v_x^o and ω^o during free-flying verify that in our indoor low-speed throwing experiments, the effect of air-drag on our 3-D printed bar can be ignored.

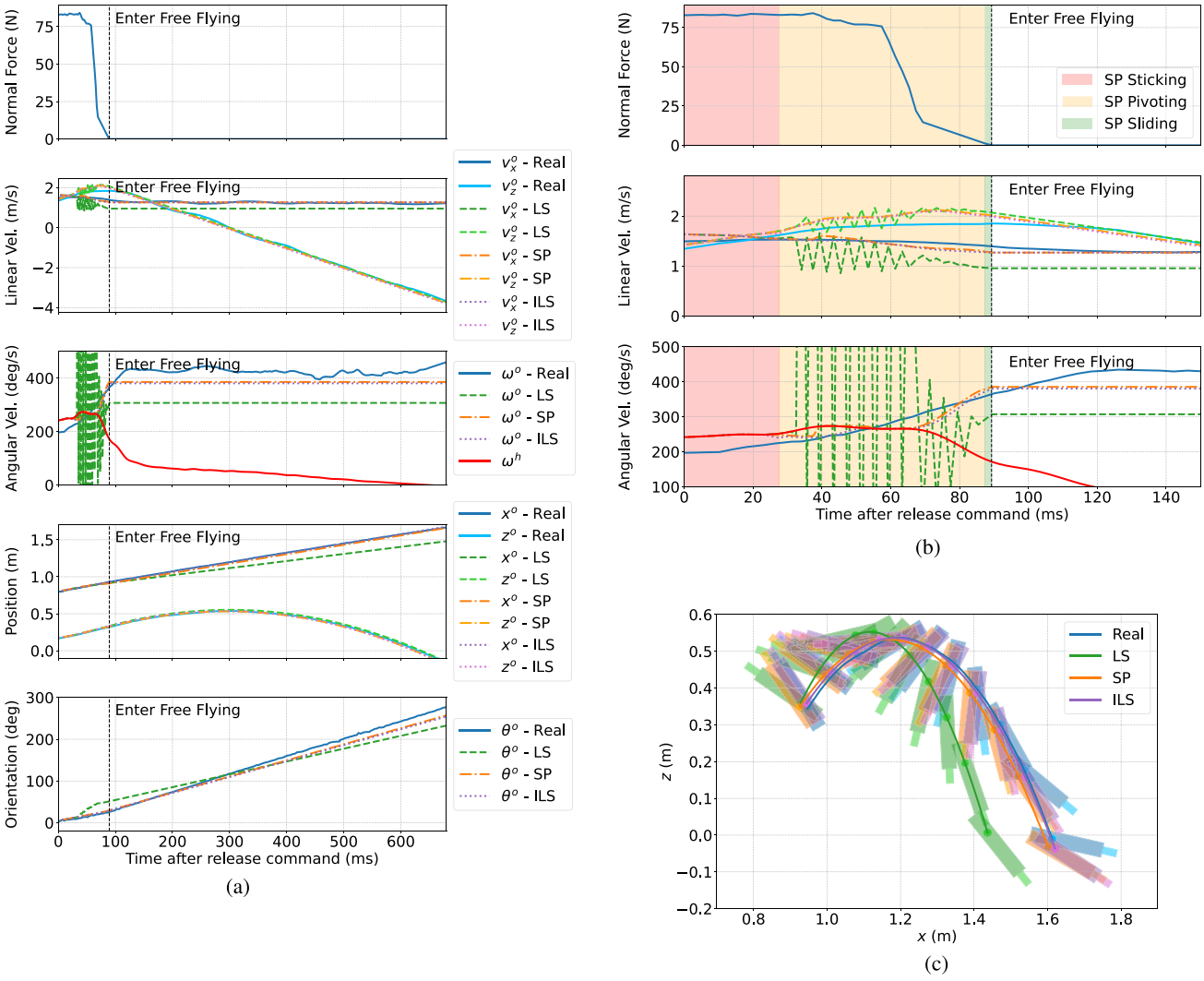
For the model predictions, first, we notice that the "SP" predictions are very close to those of "ILS." In the "Positions" plot, the predictions of "SP" and "ILS" (x^o—SP/ILS; z^o—SP/ILS) almost perfectly overlaps with the bar's real flying trajectory (x^o—Real; z^o—Real), while the "LS" model's prediction exhibits a significant error. In the "Orientations" plot, although the "SP"/"ILS" model exhibits some prediction error, it is significantly smaller than the "LS" model.

To closely examine the transient phase of the release, Fig. 16(b) provides a zoomed-in view of the 150 ms after

[1] We used the "LSODA" solver, which is the fastest implicit integrator available in SciPy—faster than alternatives such as "Radau" and "BDF" [58].

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

Fig. 15. Landing pose error heatmaps for the three models: "LS," "SP," and "ILS." The heatmaps show the mean scaled landing error (39) as a function of contact friction coefficient μ and contact radius r. Darker colors indicate lower errors. The hatched region highlights the lowest 10% of landing errors across the parameter space. The red marker indicates the optimal contact parameter for each model.



(a) (b) (c)

Fig. 16. Case study comparing models for throwing dynamics. The left panel shows the entire throwing process, from the gripper opening command to the bar's free flight and touchdown. The top-right panel zooms in on the release transient phase, and the bottom-right panel illustrates the bar's free-flying motion. The slight discrepancy between the ω^h and ω^o-real during sticking arises from edge effects caused by different filters applied across sensing modalities. These effects have negligible influence on the measurements during the free-flight and touchdown phases that we focus on and, consequently, on the model-comparison results. (a) Summary of throwing models. (b) Zoomed-in view of the 150-ms window after gripper opening command. (a) Summary of throwing models. (b) Zoomed-in view of the 150 ms window after gripper opening command. (c) Free-flying trajectories.

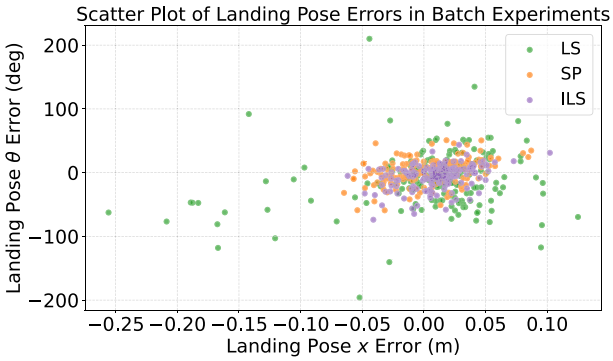


Fig. 17. Scatter plot comparing the model predictions on the 180 throws. The "SP"/"ILS" error population is much more condensed around zero compared to "LS."

giving the gripper opening command. It is important to note that the object's velocity/momentum changes drastically during the transient phase: during the release phase (0–85 ms), the object's angular velocity ω^o *accelerates* from approximately 240 to 420°/s, while the end-effector's angular velocity ω^h *decelerates* from approximately 240 to 170°/s. In the "Angular Vel." plot, the "SP"/"ILS" models effectively capture the separation of ω^o and ω^h, while the "LS" model's prediction upon detachment exhibits a much larger error. Consequently, the "SP"/"ILS" model yields an almost perfect prediction of the bar's horizontal velocity upon entering free flight. In addition, the "LS" model's prediction of the object twists during the transient phase has significant jittering during the release phase, whereas the "SP"/"ILS" models' predictions are much smoother.

The animated trajectories in Fig. 16(c) further show that the "SP"/"ILS" models predict the bar's flying motion and landing pose more accurately than the "LS" model. Specifically, the "SP" model reduces the landing horizontal position error by over 99% (<1 mm versus 16 cm), and the landing orientation error is reduced by 63% (23° versus 62°).

In this case study, the "SP" and "ILS" models demonstrate superior accuracy and stability compared to the "LS" model.

D. Batch Experiment Validation Results

In this section, we examine the predictions for the entire batch of throwing experiments with model-specific optimal contact parameters.

The landing pose prediction errors, defined as the difference between the predicted and actual landing poses, are shown in Fig. 17. The "LS" predictions exhibit a systematic bias, tending toward larger horizontal displacements (positive error in landing x) and smaller rotations (negative error in landing θ). In contrast, the "SP"/"ILS" predictions show no discernible trend or bias, indicating the absence of systematic error. Moreover, "SP"/"ILS" prediction errors are significantly more concentrated around zero compared to "LS," highlighting the superior accuracy and consistency of "SP" and "ILS" in predicting landing poses.

1) Quantitative Metrics: The quantitative results of the landing pose prediction errors are summarized in Table IV, which compares the MAE and STD of the absolute error of the predictions. The metrics include free-flying twist variables, i.e., horizontal velocity (v_o^x) and angular velocity (ω^o), and landing pose variables, i.e., horizontal landing position (x^{land}) and landing orientation (θ^{land}).

TABLE IV
MAE AND STD OF LS, SP, AND ILS PREDICTIONS FOR FREE-FLYING AND LANDING METRICS. BOLDFACE INDICATES THE LOWEST MAE FOR EACH METRIC.

Model	Free-Flying Twist		Landing Pose	
	v_x^o (m/s)	ω^o (deg/s)	x^{land} (m)	θ^{land} (deg)
LS	0.073±0.062	61.5±52.9	0.040±0.044	34.9±32.1
SP	0.044±0.028	**22.8±20.5**	0.024±0.017	15.4±12.7
ILS	**0.034±0.024**	23.7±23.2	**0.021±0.016**	**13.7±13.8**

Errors are reported as MAE ± STD.
Boldface indicates the lowest MAE for each metric.

The table highlights a significant improvement in prediction accuracy and consistency achieved by the "SP" model and the "ILS" model. Specifically, for free-flying twist predictions, SP demonstrates over 40% reduction in MAE for horizontal velocity (v_x^o) and angular velocity (ω^o), with MAEs of 0.044 m/s and 22.8°/s, respectively, compared to 0.073 m/s and 61.5°/s for "LS." Similarly, "SP" outperforms LS in landing pose prediction, achieving MAEs of 0.024 m and 15.4° for horizontal landing position (x^{land}) and orientation (θ^{land}), respectively, compared to the larger errors of "LS" (0.04 m and 34.9°).

In addition to achieving lower MAE, "SP" and "ILS" predictions also exhibit significantly smaller STD across all metrics, indicating a higher level of consistency in their predictions. This is particularly evident in the free-flying angular velocity and landing orientation, where "LS" predictions show high variability (STD of 52.9°/s and 34.9°, respectively), whereas "SP" predictions are much more concentrated (STD of 20.5°/s and 12.7°). These results align with the visual analysis in Fig. 17, further demonstrating that the "SP" model not only eliminates the systematic biases observed in "LS" but also delivers robust and reliable predictions across diverse experimental conditions.

2) Comparison to Learning-Based Models: The quantitative metrics of MAE and STD used above are consistent with those applied in TossNet [27], a data-driven deep learning model designed to predict throwing outcomes based on proprioceptive measurements and F/T sensor data mounted between the manipulator flange and gripper. Therefore, here, we highlight the comparison of our results with TossNet.

Although TossNet achieves a comparable landing position error [MAE ± STD: 0.03 ± 0.015 m] and smaller landing orientation error [MAE ± STD: 4.5 ± 2.3°], it operates under much simpler experimental conditions. Specifically, TossNet's dataset includes limited flying ranges and little to no rotation, whereas our experiments encompass a significantly broader dynamic range, including complete bar-flipping motions.

More importantly, TossNet is an end-to-end black-box model trained on 3000 real-world throwing samples, while "SP" and "ILS" are purely physics-based models. This distinction allows them to avoid the extensive data collection requirements of TossNet and offers the advantages of interpretability and generalization to "unseen" conditions.

3) Error Distributions: In addition to the quantitative metrics, the distribution of landing pose prediction errors for both models is further analyzed using boxplots for the absolute (unsigned) error and violin plots for signed error, as shown in Figs. 18 and 19, respectively. The boxplots provide a concise

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

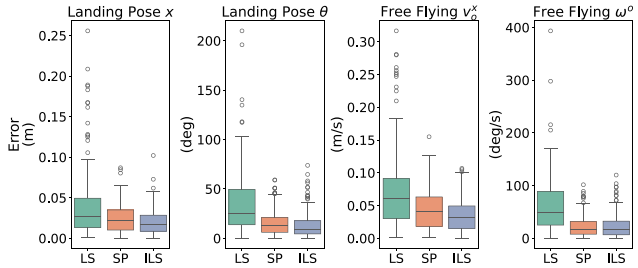


Fig. 18. Boxplot comparison of absolute errors of transient release dynamics models. "SP" and "ILS" achieve consistently smaller median errors and IQR.

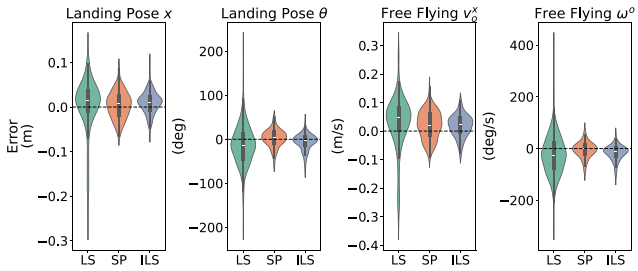


Fig. 19. Violin plot comparison of landing pose prediction errors. The "SP" and "ILS" models exhibit concentrated error distributions with a sharp density around zero, while "LS" shows broader and biased distributions.

representation of the median, interquartile range (IQR), and potential outliers for each error metric, while the violin plots offer a more detailed visualization of the error distribution's density.

Fig. 18 illustrates that the "SP" and "ILS" models consistently achieve smaller median errors and a narrower IQR across all metrics compared to the "LS" model. The large spread and presence of significant outliers in the "LS" model, particularly for free-flying angular velocity and landing orientation, further highlight the inconsistent performance of "LS."

The violin plots in Fig. 19 corroborate these observations, showing concentrated and symmetric error distributions for "SP"/"ILS," with the density centered around zero. In contrast, "LS" exhibits wider and less symmetric error distributions, reflecting its high variability and systematic biases.

4) Summary: In summary, the results of the batch experiments demonstrate the superior performance of the "SP" model compared to the conventional "LS" model. Across all evaluation metrics, "SP" and "ILS" achieve significantly lower mean errors and exhibit far less variability and systematic biases. These findings, validated through various analysis methods, provide a comprehensive assessment of the robustness and accuracy of "SP" and "ILS." Furthermore, the performance of "SP" and "ILS" is competitive with state-of-the-art data-driven end-to-end models. Compared to "ILS," "SP" offers the added benefits of interpretability and fast solving speed, reinforcing its potential for robot-throwing tasks.

E. Factor Analysis

In this section, we systematically evaluate the performance of the three release dynamics models under various variations in throwing conditions. The goal is to analyze the models' robustness and generalizability by isolating individual factors that influence landing outcomes. We conduct throwing experiments with six variations: CoM offset, pitch angle, release condition, friction, payload mass, and MOI. Notably, the first three correspond to the 3-D mesh of batch experiments in the previous section. The results are summarized in Fig. 20. For each variation, we are interested in the following questions.

1) *Real domain shift:* What is the influence of this variation on the landing outcome? How do the landing poses of the throwing condition A (shallow blue circles) *shift* compare to the landing poses of condition B (deep blue squares)?
2) *Predicted domain shift*: Can the models capture the observed shift by adjusting the corresponding physical parameters of the variation?
3) *Model consistency:* Do the models behave consistently across the two throwing conditions? This is quantified by conducting a Mann–Whitney U test [59] on each model's prediction errors for the two throwing conditions. If the error statistics are significantly different ($p < 0.05$), a "$*$" symbol is added above the corresponding pair of bar plots in Fig. 20.

1) CoM Offset: Small Offset → Large Offset: Two distinct CoM configurations are tested, as described in the batch experiments ("CoM-1": 10 cm and "CoM-2": 16 cm).

1) *Real domain shift:* As shown in Fig. 20(a), increasing the CoM offset results in a larger horizontal landing position and a smaller landing angle.
2) *Predicted domain shift:* The "SP" model, with its accurate landing pose predictions, successfully captures the real domain shift (shallow yellow circles → deep yellow squares).

2) Pitch Angle: Large Pitch → Small Pitch: Two nominal pitch angles are tested, corresponding to the largest and smallest pitch angles in the batch experiment. By decreasing the pitch angle, the robot's throwing motion changes from an "upward throw" to a "forward throw."

1) *Real domain shift:* As shown in Fig. 20(b), decreasing the pitch angle leads to an increase in the horizontal landing position and a decrease in the landing angle.
2) *Predicted domain shift:* The "SP" model accurately predicts the landing outcomes, correctly reflecting the observed shift in reality. However, in this case, similar to the other two models, "SP" exhibits performance inconsistency across the two scenarios.

3) Release Motion: Acceleration → Deceleration: After bypassing the nominal throwing state and giving the gripper opening command, it remains a design choice how to generate the robot motion during the transient phase. Here, we compare the shift from an accelerating release motion to a decelerating release motion.

1) *Real domain shift:* As shown in Fig. 20(c), transitioning from acceleration to deceleration during the release window increases the landing angle while slightly reducing the horizontal landing position. Intuitively, for a decelerating release, part of the angular momentum of the robot body is transmitted to the thrown object.
2) *Predicted domain shift:* "SP" predictions accurately reflect the shift caused by the release motion. Although "ILS" achieves the smallest error for accelerating release, its prediction of landing orientation degrades significantly for decelerating release. In comparison, in the decelerating release regime, where the "temporal hinge" is more

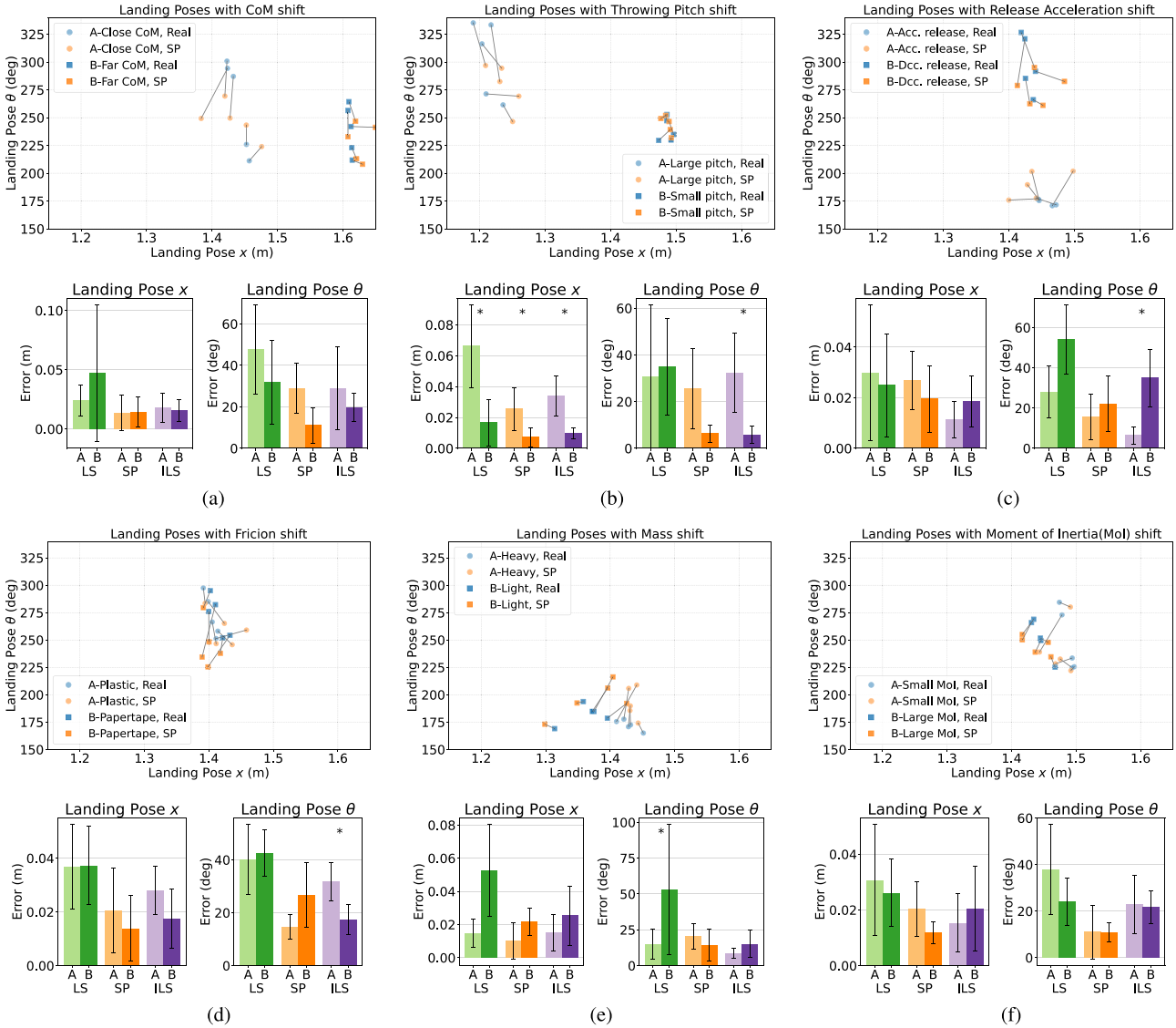


Fig. 20. Combined factor analysis showing the effects of various factors on LS versus SP models' predicted landing poses. The scatter plots show the collection of real and predicted landing poses of SP only for clarity. Solid lines connect ("Real" and "SP") pairs. To demonstrate the domain shift relative to feasible throwing ranges, all six plots are kept with the same axis limits. The bar plots summarize the MAE and STD of errors. $*$: statistically significant different performance across two throwing conditions under the Mann–Whitney U test with $p < 0.05$. (a) CoM offset shift. (b) Pitch angle shift. (c) Release motion shift. (d) Friction shift. (e) Mass shift. (f) MOI shift.

prominent (as indicated by the large landing orientation), "SP" yields the smallest landing orientation error.

4) Friction: Plastic → Papertape: By attaching papertape to the tip of the bar, we decrease the friction coefficient and examine its influence on landing outcomes. A set of model-specific contact parameters in this case is identified via the same procedure as described in Section VI-B. The identified parameters are (μ, r [mm], $\mu \cdot r$): "LS——(0.3, 8, 2.4), "SP"—(0.4, 5, 2.0), and "ILS"—(0.45, 2.5, 1.125).

1) *Real domain shift:* As shown in Fig. 20(d), the two different contact materials produce no discernible changes in the landing populations.
2) *Predicted domain shift:*"SP" correctly captures the blending of landing poses in the two scenarios. Although both "SP" and "ILS" have smaller errors than "LS," the "SP" prediction is more consistent than that of "ILS."

5) Mass: Metal Payload → Plastic Payload: The payload is varied from a heavy metal payload (154 g) to a light plastic payload (30 g) to examine the models' ability to adapt to changes in object weight.

1) *Real domain shift*: As shown in Fig. 20(e), increasing the payload mass leads to a decrease in the horizontal landing position and no distinguishable change in landing angle.
2) *Predicted domain shift*: "SP" predictions align closely with the real domain shift, effectively capturing the effects of mass variation.

6) MOI: Small → Large: For our 3-D printed bar with four payload slots, the MOI is varied by moving two metal payloads from the middle slots to the outermost slots. This effectively increases the MOI from 714 to 1961 kg $\cdot$ mm^2, while keeping the mass and CoM unchanged.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

1) *Real domain shift:* As shown in Fig. 20(f), increasing the MOI decreases the horizontal landing position, while, interestingly, causing no distinguishable change in the landing angle.
2) *Predicted domain shift:* "SP" predictions successfully reflect the observed trends, capturing the reduced horizontal landing position and the consistent landing angle.

Model consistency: In addition to the large prediction errors, "LS" is inconsistent when the throwing conditions are subject to shifts in throwing pitch angle and mass shift, indicating the model's sensitivity to multiple factors. In comparison, "SP" is only inconsistent during pitch shifts (small pitch $\rightarrow$ small error), suggesting that future improvements to "SP" should focus on the large pitch regime.

Summary: The factor analysis demonstrates that the proposed "SP" model is highly robust and generalizable across a wide range of factors, including CoM offset, pitch angle, release condition, contact friction, payload mass, and MOI. These findings validate the "SP" model as a reliable tool for predicting landing poses under diverse robot-throwing conditions.

VII. Discussion on Impact on Robot Throwing

The proposed physical model on transient release dynamics opens up new possibilities for scalable and industrial-grade robot throwing.

A. Fast and Accurate Sorting

As shown in Fig. 17, the throwing displacement error is bounded within ± 11 cm across a wide range of throwing configurations and landing poses. This provides a direct comparison with TossingBot [21], an end-to-end learning model that achieves an 85% success rate in throwing various objects with a 25-cm tolerance, but at the cost of tens of thousands of real throwing trials. In contrast, our physical model provides a more interpretable and generalizable framework for throwing planning, with the potential to reduce sample complexity and improve accuracy across diverse object types. While reliable throwing of deformable or complex objects remains an open challenge, this work advances the modeling foundations necessary for future high-technology readiness level (TRL) robot throwing systems .

B. Throwing With Desired Landing Pose

Fig. 17 demonstrates that the landing orientation error is bounded within $\pm 50^\circ$, indicating that it is possible to plan throwing motions to achieve desired landing poses (position and orientation) with high precision. This capability can facilitate barcode scanning, optimize space utilization, and improve ergonomic access for human operators or other robots in industrial automation.

C. Knowledge Transfer Among Fleets of Throwing Robots

Compared to end-to-end learning models, a key advantage of physical models is their excellent transferability. Object stock-keeping unit (SKU) information, sensor measurements, and robot motion are all described in the common language of SI units, allowing knowledge to be easily transferred across different embodiments and operating conditions. In particular, object inertial parameters (mass, CoM, and MOI) can be estimated onboard using standard payload identification techniques [60], [61], [62], [63], [64], [65] and subsequently used for model-based planning or learning in throwing tasks. On the other hand, it is unclear how information can be transferred between end-to-end learning models trained for individual robots under varying conditions.

D. Physics-Guided Active Learning

Finally, recent studies in operations research [66], [67] have shown that algorithms leveraging structural information achieve better exploration–exploitation tradeoffs than classical structure-free methods in online decision making. Therefore, for robot throwing in unstructured environments where object information is unavailable, the algebraic structure encoded in the physical model has the strong potential to guide strategic active learning and reduce sample complexity.

VIII. Conclusion

This work introduces SP, a novel physical model for capturing the transient release dynamics in robot throwing. To evaluate its performance, we develop a viscous-smoothed limit surface model with implicit integration (ILS), which achieves the highest prediction accuracy among the models considered but at substantially higher computational cost. The SP model offers a favorable tradeoff: it attains comparable accuracy, with only a 10% increase in prediction error, while providing over $20\times$ faster computation. This combination of physical interpretability, accuracy, and efficiency makes the SP model well suited for integration into model-based planning and learning frameworks. Compared to conventional patch friction models (LS), the SP model achieves over 40% reduction in horizontal velocity prediction error and 63% reduction in angular velocity prediction error, with significantly lower variability and systematic bias. This results in high accuracy for predicting the landing poses of thrown objects, with MAE of 2.4 cm for position and 15.4° for orientation. Unlike data-driven approaches, the SP model is a purely physics-based framework, offering both interpretability and generalization across diverse robot embodiments and throwing conditions. As a result, this work lays a foundation for scalable robot throwing.

Despite these advancements, the proposed model is still limited by inconsistent prediction accuracies when throwing at different pitch angles, indicating that the complex effects of gravity and frictional wrenches are not sufficiently explained by the SP model. To address this, soft contact models based on microscopic bristle deformation, such as the LuGre friction model, may be necessary. On the application side, it would be interesting to investigate data assimilation in learning to throw—specifically, how to effectively leverage this physics model to improve throwing performance as robots start to gather throwing data.

Acknowledgment

The authors would like to thank Bruno Da Costa, Elise Jeandupeux, and Tristan Bonato for assistance with the experimental setups. The authors are also grateful to the anonymous reviewers for their invaluable and constructive feedback.

REFERENCES

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ, USA: Wiley, 2020.

[2] S. Kim and A. Billard, "Estimating the non-linear dynamics of free-flying objects," *Robot. Auton. Syst.*, vol. 60, no. 9, pp. 1108–1122, 2012.

[3] M. V. Lubarda and V. A. Lubarda, "A review of the analysis of wind-influenced projectile motion in the presence of linear and nonlinear drag force," *Arch. Appl. Mech.*, vol. 92, no. 7, pp. 1997–2017, 2022.

[4] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction Part 1. Limit surface and moment function," *Wear*, vol. 143, no. 2, pp. 307–330, 1991.

[5] R. D. Howe and M. R. Cutkosky, "Practical force-motion models for sliding manipulation," *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 557–572, 1996.

[6] N. Xydas and I. Kao, "Modeling of contact mechanics and friction limit surfaces for soft fingers in robotic, with experimental results," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 941–950, 1999.

[7] A. R. Champneys and P. L. Várkonyi, "The Painlevé paradox in contact mechanics," *IMA J. Appl. Math.*, vol. 81, no. 3, pp. 538–588, 2016.

[8] E. W. Aboaf, C. G. Atkeson, and D. J. Reinkensmeyer, *Task-Level Robot Learning: Ball Throwing*. Cambridge, MA, USA: Massachusetts Inst. Technol., 1987.

[9] S. Schaal and C. G. Atkeson, "Open loop stable control strategies for robot juggling," in *Proc. 1IEEE Int. Conf. Robot. Autom.*, 1993, pp. 913–918.

[10] S. Schaal and C. G. Atkeson, "Robot juggling: Implementation of memory-based learning," *IEEE Control Syst. Mag.*, vol. 14, no. 1, pp. 57–71, Feb. 1994.

[11] M. T. Mason and K. M. Lynch, "Dynamic manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1993, pp. 152–159.

[12] 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.

[13] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed throwing motion based on kinetic chain approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3206–3211.

[14] W. Mori, J. Ueda, and T. Ogasawara, "1-DOF dynamic pitching robot that independently controls velocity, angular velocity, and direction of a ball: Contact models and motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 1655–1661.

[15] H. Miyashita, T. Yamawaki, and M. Yashima, "Control for throwing manipulation by one joint robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 1273–1278.

[16] J. Kober, K. Muelling, and J. Peters, "Learning throwing and catching skills," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5167–5168.

[17] Y. Zhang, J. Luo, and K. Hauser, "Sampling-based motion planning with dynamic intermediate state objectives: Application to throwing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 2551–2556.

[18] Y. Gai, Y. Kobayashi, Y. Hoshino, and T. Emaru, "Motion control of a ball throwing robot with a flexible robotic arm," *Int. J. Comput. Inf. Eng.*, vol. 7, no. 7, pp. 937–945, 2013.

[19] A. Pekarovskiy and M. Buss, "Optimal control goal manifolds for planar nonprehensile throwing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4518–4524.

[20] O. Taylor and A. Rodriguez, "Optimal shape and motion planning for dynamic planar manipulation," *Auton. Robots*, vol. 43, no. 2, pp. 327–344, 2019.

[21] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "TossingBot: Learning to throw arbitrary objects with residual physics," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1307–1319, Aug. 2020.

[22] M. Monastirsky, O. Azulay, and A. Sintov, "Learning to throw with a handful of samples using decision transformers," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 576–583, Feb. 2023.

[23] Y. Liu, A. Nayak, and A. Billard, "A solution to adaptive mobile manipulator throwing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 1625–1632.

[24] M. Bombile and A. Billard, "Bimanual dynamic grabbing and tossing of objects onto a moving target," *Robot. Auton. Syst.*, vol. 167, 2023, Art. no. 104481.

[25] H. Kasaei and M. Kasaei, "Throwing objects into a moving basket while avoiding obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 3051–3057.

[26] H. V. S. Pasala, N. Govindan, and S. Brahmbhatt, "Identification and learning-based control of an end-effector for targeted throwing," *IEEE Robot. Autom. Lett.*, vol. 9, no. 11, pp. 9558–9564, Nov. 2024.

[27] L. Chen, W. Lu, K. Zhang, Y. Zhang, L. Zhao, and Y. Zheng, "TossNet: Learning to accurately measure and predict robot throwing of arbitrary objects in real time with proprioceptive sensing," *IEEE Trans. Robot.*, vol. 40, pp. 3232–3251, 2024.

[28] Y. Liu and A. Billard, "Tube acceleration: Robust dexterous throwing against release uncertainty," *IEEE Trans. Robot.*, vol. 40, pp. 2831–2849, 2024.

[29] A. Gallant and C. Gosselin, "Maximizing the throwing distance of robotic manipulators: An optimization approach," *J. Mech. Robot.*, vol. 16, no. 9, 2024, Art. no. 091008.

[30] A. Zermane, N. Dehio, and A. Kheddar, "Planning impact-driven logistic tasks," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2184–2191, 2024.

[31] L. Werner, F. Nan, P. Eyschen, F. A. Spinelli, H. Yang, and M. Hutter, "Dynamic throwing with robotic material handling machines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 98–104.

[32] R. D. Howe, I. Kao, and M. R. Cutkosky, "The sliding of robot fingers under combined torsion and shear loading," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1988, pp. 103–105.

[33] M. M. Ghazaei Ardakani, J. Bimbo, and D. Prattichizzo, "Quasi-static analysis of planar sliding using friction patches," *Int. J. Robot. Res.*, vol. 39, no. 14, pp. 1775–1795, 2020.

[34] R. Featherstone, *Rigid Body Dynamics Algorithms*. New York, NY, USA: Springer, 2014.

[35] F. Marques, P. Flores, J. Pimenta Claro, and H. M. Lankarani, "A survey and comparison of several friction force models for dynamic analysis of multibody mechanical systems," *Nonlinear Dyn.*, vol. 86, pp. 1407–1443, 2016.

[36] D. Karnopp, "Computer simulation of stick-slip friction in mechanical dynamic systems," *J. Dyn. Syst., Meas., Control*, vol. 107, no. 1, pp. 100–103, 1985.

[37] A. Sintov, O. Tslil, and A. Shapiro, "Robotic swing-up regrasping manipulation based on the impulse–momentum approach and cLQR control," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1079–1090, Oct. 2016.

[38] S. Cruciani and C. Smith, "In-hand manipulation using three-stages open loop pivoting," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1244–1251.

[39] A. M. Castro, F. N. Permenter, and X. Han, "An unconstrained convex formulation of compliant contact," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1301–1320, Apr. 2023.

[40] Q. Le Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, "Contact models in robotics: A comparative analysis," *IEEE Trans. Robot.*, vol. 40, pp. 3716–3733, 2024.

[41] J. Shi, J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch, "Dynamic in-hand sliding manipulation," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 778–795, Aug. 2017.

[42] X. Yi and N. Fazeli, "Precise object sliding with top contact via asymmetric dual limit surfaces," in *Proc. Robot. Sci. Syst. Conf.*, Daegu, Republic of Korea, pp. 1–10, Jul. 2023.

[43] M. Antali, "Harmonic expansion and nonsmooth dynamics in a circular contact region with combined slip-spin motion," *Nonlinear Dyn.*, vol. 112, no. 9, pp. 6785–6811, 2024.

[44] J. Zhou, M. T. Mason, R. Paolini, and D. Bagnell, "A convex polynomial model for planar sliding mechanics: Theory, application, and experimental validation," *Int. J. Robot. Res.*, vol. 37, nos. 2/3, pp. 249–265, 2018.

[45] M. Antali and P. L. Varkonyi, "The nonsmooth dynamics of combined slip and spin motion under dry friction," *J. Nonlinear Sci.*, vol. 32, no. 4, 2022, Art. no. 58.

[46] M. Antali and G. Stepan, "Sliding and crossing dynamics in extended Filippov systems," *SIAM J. Appl. Dyn. Syst.*, vol. 17, no. 1, pp. 823–858, 2018.

[47] A. C. Hindmarsh, "ODEPACK, a systemized collection of ODE solvers," in *Scientific Computing*. Amsterdam, The Netherlands: North Holland, 1983.

[48] A. Nurkanović, M. Sperl, S. Albrecht, and M. Diehl, "Finite elements with switch detection for direct optimal control of nonsmooth systems," *Numer. Math.*, vol. 156, pp. 1115–1162, 2024.

[49] G. A. Waltersson and Y. Karayiannidis, "Planar friction modelling with LuGre dynamics and limit surfaces," *IEEE Trans. Robot.*, vol. 40, pp. 3166–3180, 2024.

[50] C. Canudas de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 419–425, Mar. 1995.

[51] K. Johanastrom and C. Canudas-De-Wit, "Revisiting the LuGre friction model," *IEEE Control Syst. Mag.*, vol. 28, no. 6, pp. 101–114, Dec. 2008.

[52] A. J. Tuononen, "Onset of frictional sliding of rubber–glass contact under dry and lubricated conditions," *Sci. Rep.*, vol. 6, no. 1, 2016, Art. no. 27951.
[53] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction Part 2. Dynamics of motion," *Wear*, vol. 143, no. 2, pp. 331–352, 1991.
[54] P. D. Weidman and C. P. Malhotra, "On the terminal motion of sliding spinning disks with uniform Coulomb friction," *Phys. D: Nonlinear Phenomena*, vol. 233, no. 1, pp. 1–13, 2007.
[55] P. Listov and C. Jones, "PolyMPC: An efficient and extensible tool for real-time nonlinear model predictive tracking and path following for fast mechatronic systems," *Optimal Control Appl. Methods*, vol. 41, no. 2, pp. 709–727, 2020.
[56] M. Quigley et al., "ROS: An open-source robot operating system," in *Proc. IEEE Int. Conf. Robot. Autom. Workshop Open Source Softw.*, Kobe, Japan, 2009, p. 5.
[57] B. Acosta, W. Yang, and M. Posa, "Validating robotics simulators on real-world impacts," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6471–6478, Jul. 2022.
[58] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, no. 3, pp. 261–272, 2020.
[59] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, 1947.
[60] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 101–119, 1986.
[61] D. Kubus, T. Kroger, and F. M. Wahl, "On-line estimation of inertial parameters using a recursive total least-squares approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3845–3852.
[62] C. Gaz and A. De Luca, "Payload estimation based on identified coefficients of robot dynamics—With an application to collision detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3033–3040.
[63] A. Kurdas, M. Hamad, J. Vorndamme, N. Mansfeld, S. Abdolshah, and S. Haddadin, "Online payload identification for tactile robots using the momentum observer," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 5953–5959.
[64] P. Nadeau, M. Giamou, and J. Kelly, "Fast object inertial parameter identification for collaborative robots," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 3560–3566.
[65] S. Shan and Q.-C. Pham, "Fast payload calibration for sensorless contact estimation using model pre-training," *IEEE Robot. Autom. Lett.*, vol. 9, no. 10, pp. 9007–9014, Oct. 2024.
[66] D. Russo and B. Van Roy, "Learning to optimize via information-directed sampling," *Oper. Res.*, vol. 66, no. 1, pp. 230–252, 2018.
[67] B. Van Parys and N. Golrezaei, "Optimal learning for structured bandits," *Manage. Sci.*, vol. 70, no. 6, pp. 3951–3998, 2024.

Yang Liu (Graduate Student Member, IEEE) received the B.Sc. degree in mechanical engineering from Jilin University, Changchun, China, in 2017, and the M.Sc. degree in robotics, systems, and control from ETH Zurich, Zurich, Switzerland, in 2020. He is currently working toward the Ph.D. degree with the Learning Algorithms and Systems Laboratory, Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland.

His research interests include dexterous dynamic manipulation and reactive motion planning for robots.

Aude Billard (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in physics from the Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland in 1994 and 1995, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K. in 1998.

She is currently a Full Professor, the Head of the Learning Algorithms and Systems Laboratory, and the Associate Dean for Education with the School of Engineering, EPFL. Her research interests include the fields of machine learning and robotics with a particular emphasis on fast and reactive control and on safe human–robot interaction.

Dr. Billard is the recipient of the Intel Corporation Teaching Award, the Swiss National Science Foundation Career Award, the Outstanding Young Person in Science and Innovation from the Swiss Chamber of Commerce, the IEEE RAS Distinguished Award, and the IEEE-RAS Best Reviewer Award. She has received numerous best conference paper awards, as well as the prestigious King-Sun Fu Memorial Award for the Best IEEE Transactions in Robotics Paper, and is regularly featured in premier venues (BBC, IEEE Spectrum, and Wired). She is the President of the IEEE Robotics and Automation Society, the Director of the ELLIS Robot Learning Program, and a co-Director of the Robot Learning Foundation, a nonprofit corporation that serves as the governing body behind the Conference on Robot Learning and leads the Innovation Booster Robotics, a program funding technology transfer in robotics and powered by the Swiss Innovation Agency, Innosuisse, Bern, Switzerland. She was a Plenary Speaker at major robotics, AI, and control conferences (International Conference on Robotics and Automation, AAAI Conference on Artificial Intelligence, Conference on Robot Learning, ACM/IEEE conference on Human–Robot Interaction, IEEE International Conference on Automation Science and Engineering, International Conference on Development and Learning, European Conference on Machine Learning, Annual Learning for Dynamics and Control Conference, IFAC Symposium, ROMAN, Humanoids, and many others) and acted on various positions on the organization committee of numerous International Conferences in Robotics.