



HAL
open science

Minimal Impact Pokes to Place Objects on Planar Surfaces

Ahmed Zermane, Léo Moussafir, Youcan A Yan, Abderrahmane Kheddar

► **To cite this version:**

Ahmed Zermane, Léo Moussafir, Youcan A Yan, Abderrahmane Kheddar. Minimal Impact Pokes to Place Objects on Planar Surfaces. *IEEE Robotics and Automation Letters*, 2024, 9 (12), pp.11393-11400. 10.1109/LRA.2024.3491412 . hal-04786864

HAL Id: hal-04786864

<https://hal.science/hal-04786864v1>

Submitted on 16 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Minimal Impact Pokes to Place Objects on Planar Surfaces

Ahmed Zermane*, *Member, IEEE*, Léo Moussafir*, Youcan Yan, and Abderrahmane Kheddar, *Fellow, IEEE*

Abstract—We present a planning and control method that computes a minimal sequence of pokes to slide a given object from an initial pose to a desired final one (or as close to it as possible) on a planar surface. Both planning and control are based on impact models to generate pokes. Our framework takes into account the object’s dynamics with a rich contact model and parameters to plan the poking sequence. The planning is conducted in the joint-space and generates trajectories tracked using an impact-aware QP control, which corrects for post-pokes errors using discrete visual feedback. We implemented our method on a Panda robot arm and assessed its versatility and robustness. The experimental results show that the proposed poking approach can bring the object to the desired position and orientation with minimal errors (0.05 m for translation and 0.2 rad for rotation), highlighting its potential application in diverse industrial scenarios such as logistics.

Index Terms—Impact-aware planning and control, Non-Prehensible Dexterous Manipulation, Optimal Control.

I. INTRODUCTION

POKING an object with a robot to make it slide toward a precise configuration can serve various applications such as in logistics. Poking is a subset of non-prehensible manipulations, it can enhance productivity by reducing time required for robotic tasks [1]. Poking objects offers an alternative to traditional pick-and-place or tossing tasks, effectively addressing issues such as dead zone problems—areas within a robot’s workspace that are difficult to reach—and gripping limitations, where robots struggle to securely grasp certain objects due to their size, shape, or surface properties. Dead zone issue is also relevant to pushing tasks, where the robot end-effector must sustain contact with the object [2]–[4].

Given a setup consisting of a robot, a planar surface, and an object to be slid from one end (initial planar configuration) to the other (target planar configuration). In imposed single-poking, pose target can be missed or even unfeasible, and no correction is then possible [5] contrary to multi-step poking strategies. Finding the minimal poking hits that achieves a target goal is a challenging problem. It requires understanding

the kinodynamics of planar motion, which is hard to predict due to the unilateral nature and the friction dynamics of the sliding contact behavior. Poking involves successive impact inputs generated by the robot’s end-effector on the object, necessitating consideration of: (i) both robot and object resilience to impact, (ii) impact-aware control [6], and (iii) impact parameters. This hybrid dynamic characteristic is what makes planning the optimal succession of pokes challenging. Yet, if feasible, the outcome can be a single hit.

Recent work on poking, e.g., [1], [5], [7], [8], see Sec. II, primarily focuses on the translational motion of the object not considering spinning. This limitation restricts their application in scenarios where precision in the object’s target pose is needed. In this work, our main contribution is in addressing this problem under a model-based formulation perspective. Indeed, we show that formulating the entire planning as a Mixed-Integer Non-Linear Optimization Problem (MINLP), effectively resolves successive pokes solutions (Sec. IV). This formulation of the problem allows integrating friction by devising a simplified model for planar motions that accounts for both sliding and spinning motions to ease online identification of the friction parameters (Sec. III). We also enabled incorporating the robot’s capabilities to generate impacts and defining poking task limits (Sec. III). In order to assess our approach, we conducted real-time experiments on a manipulator robot with a dedicated setup on different objects (Sec. V).

II. RELATED WORK

There is a large panel of research dealing with tasks in which robots manipulate objects or interact with them to achieve specific goals such as picking, tossing, pushing, stopping, or even hitting and halting, see [9]. An extensively studied form of non-prehensible manipulation is planar pushing. This quasi-static problem assumes a sustained contact force between the robot’s end-effector and the object of interest. In [2], a high-fidelity experimental dataset is created where push interactions vary across six pushing action dimensions that are: contact point, direction, speed, acceleration, surface characteristics, and shape of objects. This dataset is later employed in a data-driven pushing model on different objects [3]. In [10], a Model Predictive Controller (MPC) is formulated using a complementarity-based hard-contact model, Newton’s Impact law, and optimal control to tackle various dynamic pushing problems. In [11], combined analytical and learning models allowed to enhance the prediction during pushing manipulations. In [12], a Limit-Surface model is used to slide objects on planar surfaces to desired poses accurately. State-triggered constraints within the friction model are exploited

This work was supported by the Research Project IAM. through the European Union H2020 program (GA 871899).

* Equal Contribution. Corresponding author: Youcan Yan.

A. Zermane, L. Moussafir, Y. Yan and A. Kheddar are with the CNRS-University of Montpellier LIRMM, UMR5506, Montpellier, France, email: surname.name@lirmm.fr

A. Kheddar is also with the CNRS-AIST Joint Robotics Laboratory, IRL3218, Tsukuba, Japan.

A. Zermane is also with Complex Systems Control and Simulation Laboratory, EMP, Algiers, Algeria.

This letter has supplementary material available at <https://doi.org/10.1109/LRA.2024.XXXXXXX>, provided by the authors.

Digital Object Identifier (DOI): see top of this page.

in [13] to define contact modes of quasi-static pushing and compare the performance to a Mixed-Integer Programming (MIP) based controller. A framework in [4] built around one-finger manipulation is used to reposition and reorient 3D objects on frictional surfaces by pushing.

Yet, all the aforementioned work involves interacting with objects at low speed while maintaining contact with the object throughout the motion; which differs from poking. In [14], an intuitive physics model using interaction data is demonstrated for poking, where internal joint's forward and inverse models describe how poked objects move. Poking as a task is also addressed in [15] for objects short translational displacement by solving a contact implicit trajectory optimization; the latter is based on variable smooth contact and successive convexification yet not considering the frictional forces. This work was extended later to be tuning-free, in which physical simulations are used to solve each task [7]. Another example is [16], where an impact-aware multi-mode Trajectory Optimization (TO) method with hybrid dynamics and control encoded in a single formulation is used to halt moving objects. In a recent work [8], a data-driven framework solves single-shot poking by first aligning the object to a stable poking region, estimating its center of mass, and then applying the necessary hitting poke impulse using a tactile sensor mounted on a tool. A related approach in [5] develops a statistical model (Gaussian Process Regression) and a directional inertia controller to form "hitting flux" to poke different objects to various target positions, yielding satisfactory results. In [1], a sampling-based method is used to plan poking in the object configuration space, but it does not define the limits of the robotic poking task. Additionally, they use feed-forward physics (under the PyBullet simulation environment) to check the feasibility of solutions within the planning loop. The entire planning process is formulated as a closed loop to compensate for approach errors. However, these works consider only the sliding motion of the object, neglecting its spinning motion during poking, which is critical in real-world applications.

To overcome this limitation, we propose a simple friction model based on energy loss during planar motion. This model accounts for both sliding and spinning of the object and is computationally efficient compared to other friction models [17]–[19], such as those based on LuGre Dynamics and Limit-Surface [19]. We have also investigated impact analysis in mechanical interactions [20], estimating the robot's dynamics regarding impact, torque, and velocity jumps (using methods described in [6], [21]) and integrating them within our QP control framework over an entire trajectory horizon. Experimental results show that our proposed poking approach can move the object to the desired position and orientation with minimal errors (0.05 m for translation and 0.2 rad for rotation), highlighting its potential applications in diverse industrial scenarios such as logistics.

III. PHYSICS MODELING

Given an object with fixed shape and inertia, having a mass m and an inertia matrix I , center of mass (CoM) \mathbf{c} , initial location \mathbf{O} on a flat support, and a static-balanced

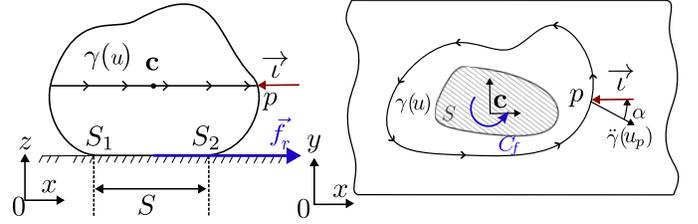


Figure 1: Schematic illustration of the object and poking parameters.

pose $P_{\mathbf{O}}$ such that its current contacting known surface with the support is not nil (i.e., planar contact). Let \mathbf{B} and $P_{\mathbf{B}}$ be, respectively, this object's target location and pose to be achieved by means of a sequence of impact-pokes generated by a robot. Pokes can be made on a point anywhere on the lateral object's surfaces, noted p . Both p and \mathbf{O} are assumed within the robot's reachable space \mathcal{W} , where as $P_{\mathbf{B}}$ can be inside or outside \mathcal{W} . Starting from \mathbf{O} , this problem can be formulated as finding a sequence of CoM planar velocities $\dot{\mathbf{c}}_k$, with $\min k \in \mathbb{N}^* \ll +\infty$, that achieves \mathbf{c} reaching \mathbf{B} . The pokes shall not induce a change in the contact topology, and shall not induce a non-reversible deformation on the object. Only sliding and spinning around the planar support's normal is allowed. Therefore, impulses \mathbf{u}_k must apply in the horizontal plane at $z = z_c$. Static and dynamic translational and rotation frictions are considered. Figure 1 shows these parameters for any object. The main challenges in solving this problem are:

- find a minimal bounded k (number of pokes);
- find the best first pokes, especially if $P_{\mathbf{B}} \notin \mathcal{W}$, to bring the object as close to $P_{\mathbf{B}}$ yet in \mathcal{W} such that the last poke warrant to highest possible precision to both $P_{\mathbf{B}}$ and \mathbf{B} .

A. Planar Motion Modeling with Energy Loss

1) *Sliding*: we express the linear kinetic energy E_c (J) of the object and the work W_f (J) due to frictional forces \mathbf{f}_r (N) between the object and the support as follows:

$$E_c = \frac{1}{2}m\|\dot{\mathbf{c}}\|^2 \quad (1)$$

$$W_f = \|\mathbf{f}_r\|\|\Delta\mathbf{c}\| = m\mu_d g\|\Delta\mathbf{c}\| \quad (2)$$

where μ_d is the coefficient of dynamic friction; g is the gravity (m/s^2); and $\Delta\mathbf{c}$ is the CoM planar displacement (m).

At impact time ($t = 0$), the kinetic energy E_c is at its maximum subsequent to post-impact velocity. The work W_f done by friction forces dissipates this energy until the object comes to rest. At the final position, the kinetic energy is zero, and the total mechanical energy is given by $E_m = E_c - W_f$.

The initial velocity can then be derived from (1) and (2):

$$\dot{\mathbf{c}}^i = \sqrt{2g\mu_d}\|\mathbf{c}^{\mathbf{B}}\| \quad (3)$$

2) *Spinning*: since the object orientation is also controlled, we express the rotational kinetic energy $E_{\theta k}$ (J) as:

$$E_{\theta k} = \frac{1}{2}\boldsymbol{\omega}^T I \boldsymbol{\omega} \quad (4)$$

with $\boldsymbol{\omega}$ the angular velocity (rad/s) of the object. The energy lost by friction can be interpreted as a result of a *brake*

action on a surface S between the object and the support. Each surface element dS located on the point r of the object's contact surface produces a torque C_f at the CoM.

$$C_f = \int_S \|\vec{c}\vec{r}\| \frac{gm}{S} \mu_d dS \quad (5)$$

The corresponding work W_f (J) is $W_f = C_f \Delta\theta$, where $\Delta\theta$ is the resultant rotation around the surface's normal. Once k -th initial rotational energy E_{θ_k} , due to initial angular velocity ω_i , is fully dissipated, the final rotation θ_{final} is:

$$\theta_{\text{final}} = \frac{\omega_i^T \mathbf{I} \omega_i}{2C_f} \quad (6)$$

B. Impact Dynamics Modeling

Impact dynamics derives from Newton's second law applied to the object's CoM \mathbf{c} at impact. Let \mathbf{f} be the impact force applied to the object at point p , we have:

$$\mathbf{f} = m\ddot{\mathbf{c}}; \quad \int_0^{\delta t} \mathbf{f} dt = m \int_0^{\delta t} \ddot{\mathbf{c}} dt \quad (7)$$

$$\dot{\mathbf{c}}^{\delta t} = m^{-1} \boldsymbol{\iota}$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \vec{c}\vec{p} \times \mathbf{f}; \quad \mathbf{I} \int_0^{\delta t} \dot{\boldsymbol{\omega}} dt = \vec{c}\vec{p} \times \int_0^{\delta t} \mathbf{f} dt \quad (8)$$

$$\boldsymbol{\omega}^{\delta t} = \mathbf{I}^{-1}(\vec{c}\vec{p} \times \boldsymbol{\iota})$$

These equations establish the relation between velocity ($\dot{\mathbf{c}}, \boldsymbol{\omega}$) and impulse $\boldsymbol{\iota}$, linking kinetic energies to the poking impulse $\boldsymbol{\iota}$ and, eventually, to robot control. In our work, we assumed that the object is still before each poke. Hence, its initial velocity is equal to the post-impact velocities: $\dot{\mathbf{c}}^{\delta t} = \dot{\mathbf{c}}_i$, $\boldsymbol{\omega}^{\delta t} = \boldsymbol{\omega}_i$. In other words, we do not apply poking to moving objects. This model depicts how an object responds to a specific impulse.

Integrating the equation of motion over the impact time δt , see [22], one can express the joint space velocity jump in terms of the task space impulse:

$$\mathbf{I}_{(q)} \Delta \dot{\mathbf{q}} = \mathbf{J}^T \boldsymbol{\iota} \quad (9)$$

$\mathbf{I}_{(q)}$ is the robot's inertia matrix, \mathbf{J} is the jacobian, and \mathbf{q} is the joint configuration. By left-multiplying by $\mathbf{I}_{(q)}^{-1}$, we have:

$$\Delta \dot{\mathbf{q}} = \mathbf{I}_{(q)}^{-1} \mathbf{J}^T \boldsymbol{\iota} \quad (10)$$

Then the end-effector (EE) cartesian velocity \mathbf{V} is:

$$\mathbf{J} \Delta \dot{\mathbf{q}} = \Delta \mathbf{V} = \mathbf{J} \mathbf{I}_{(q)}^{-1} \mathbf{J}^T \boldsymbol{\iota} \quad (11)$$

where $\mathbf{J} \mathbf{I}_{(q)}^{-1} \mathbf{J}^T$ is the inverse of the effective mass matrix $\Lambda_{(q)}$ at EE. Using the translation part of this matrix, defined as $\Lambda_{(q)}^t$, one can express the impulse $\boldsymbol{\iota}$ in terms of momentum \mathbf{h} as:

$$\boldsymbol{\iota} = \Delta \mathbf{h}_{\text{box}} = \Delta \mathbf{h}_{\text{robot}} \quad (12)$$

$$m\dot{\mathbf{c}}^+ - m\dot{\mathbf{c}}^- = \Lambda_{(q)}^t \mathbf{V}^+ - \Lambda_{(q)}^t \mathbf{V}^-$$

and the coefficient of restitution e_r as:

$$e_r = \frac{\|\dot{\mathbf{c}}^+ - \mathbf{V}^+\|}{\|\mathbf{V}^- - \dot{\mathbf{c}}^-\|} \quad (13)$$

where the post and pre-impact velocities are denoted with the superscript + and -, respectively. Let $\hat{\boldsymbol{\iota}}$ be the directional unit

vector of $\boldsymbol{\iota}$, we express the object velocity as $\dot{\mathbf{c}}^+ = \|\dot{\mathbf{c}}^+\| \hat{\boldsymbol{\iota}} = \dot{c}^+ \hat{\boldsymbol{\iota}} = (\iota/m) \hat{\boldsymbol{\iota}}$, and EE velocities as $\mathbf{V}^\square = \|\mathbf{V}^\square\| \hat{\boldsymbol{\iota}} = V^\square \hat{\boldsymbol{\iota}}$, equations (12) and (13) can be written as:

$$m\dot{c}^+ \hat{\boldsymbol{\iota}} = (V^+ - V^-) \Lambda_{(q)}^t \hat{\boldsymbol{\iota}} \quad (14a)$$

$$e_r = \frac{\|(\dot{c}^+ - V^+) \hat{\boldsymbol{\iota}}\|}{\|V^- \hat{\boldsymbol{\iota}}\|} = \frac{|\dot{c}^+ - V^+|}{V^-} \quad (14b)$$

By substituting V^+ from (14b) into (14a) and left-multiply it by $\hat{\boldsymbol{\iota}}^T$, the robot's required pre-impact velocity that satisfies the impulse $\boldsymbol{\iota}$ is:

$$V^- = \frac{\dot{c}^+(m + \lambda)}{(1 + e_r)\lambda} = \frac{\iota}{(1 + e_r)} \frac{m + \lambda}{\lambda m} \quad (15)$$

where $\lambda = \hat{\boldsymbol{\iota}}^T \Lambda_{(q)}^t \hat{\boldsymbol{\iota}}$ is the directional mass.

C. Constraints and Limits Definition for Poking Tasks

Now we define the limits for the poking task. The impulse space $\mathcal{X}_{\text{impulse}}$ of the robot can be define as:

$$\{\boldsymbol{\iota} \in \mathbb{R}^3 \mid \boldsymbol{\iota} = \Lambda_{(q)}^t \Delta \mathbf{V}, \quad \forall \Lambda_{(q)}^t \in \mathcal{X}_{\text{mass}} \wedge \forall \mathbf{V}^+, \mathbf{V}^- \in \mathcal{X}_{\text{velocity}}\} \quad (16)$$

where $\mathcal{X}_{\text{velocity}}$ is the set of admissible velocities, and $\mathcal{X}_{\text{mass}}$ is the set of allowed masses. In order to effectively define these limits, we consider the following key features:

1) *Poking Reachable Workspace*: This defines the spatial limits of the robot's poking ability. It can be estimated off-line by generating position and velocity samples within the robot's reachable workspace \mathcal{W} . The resulting reachable-by-poking workspace depends on the coefficient of restitution, friction, and the object's inertial parameters.

2) *Poked Object Maximum Mass*: estimating the maximum weight that can be poked involves considering the maximum allowed joint state jumps of the used robot, also the static friction coefficient μ_s . The robot must be able to exert sufficient impulsive force to move the object. However, if the static friction or the object's mass m are too high, and the allowed torque jumps $\Delta \boldsymbol{\tau}_q$ are limited, the robot is unable to generate poking that lead to effective motions of the object. Based on the impact model and the joint-space projections reported in [6], the relationship between the impulsive force and the torque jumps can be expressed as:

$$\Delta \boldsymbol{\tau}_q = \mathbf{J}^t \mathbf{f} = \mathbf{J}^t \frac{d\boldsymbol{\iota}}{dt} \quad (17)$$

Here, $\boldsymbol{\tau}_q$ is the joint torques; from Fig. 1 (left side) we have:

$$\left\| \frac{d\boldsymbol{\iota}}{dt} \right\| > f_s \quad (18)$$

where $f_s = mg\mu_s$ is the static friction force. Consequently, the torque jumps in the robot are determined as follows:

$$\Delta \boldsymbol{\tau}_q < mg\mu_s \mathbf{J}^t \frac{\boldsymbol{\iota}}{\|\boldsymbol{\iota}\|} = mg\mu_s \mathbf{J}^t \mathbf{u}_\boldsymbol{\iota} \quad (19)$$

where $\mathbf{u}_\boldsymbol{\iota}$ is the impact directional vector. The problem is constrained by the robot's limits: $\Delta \boldsymbol{\tau}_q < (\Delta \boldsymbol{\tau}_q)^{\text{limits}}$. With

these constraints, the maximal pokable mass constraint writes:

$$m_{\max} < \min \left((\Delta \tau_q)^{\text{limits}} \odot \frac{1}{g \mu_s J^t \mathbf{u}_l} \right) \quad (20)$$

where \odot denotes the Hadamard product (i.e., element-wise product). This quantity is configuration-dependent, observed during trials on a real robot, as it includes the Jacobian term.

IV. PROBLEM FORMULATION

A. Cost Function

After establishing the planar motion model and the related constraints for the robot, we formulate the poking hits problem as a multi-objective Mixed-Integer Nonlinear Programming (MINLP) with the following cost functions:

$$g_1 = |\overrightarrow{OT} - \mathbf{c}_{k+1}|, \quad g_2 = |\theta_T - \theta_{\mathbf{c}_{k+1}}|, \quad g_3 = k \quad (21)$$

For k pokes, we can predict the position and orientation of the object with equations (3) and (6) after the k^{th} poke as:

$$\mathbf{c}_{k+1} = \frac{\dot{\mathbf{c}}_k^2 \odot \text{sign}(\dot{\mathbf{c}}_k)}{2g\mu_d} + \mathbf{c}_k \quad (22)$$

$$\theta_{k+1} = \frac{\mathbf{I}_{zz} \omega_k^z \odot \text{sign}(\omega_k^z)}{2C_f} + \theta_k \quad (23)$$

where \star_k^z , and \mathbf{I}_{zz} are the components around the support normal direction z for $(\overrightarrow{cp} \times \mathbf{l})$, ω and \mathbf{I} respectively. We replace velocities with the impulse from eqs. (7) and (8) as:

$$\mathbf{c}_{k+1} = \frac{\mathbf{l}_k^2 \odot \text{sign}(\mathbf{l}_k)}{2m^2 g \mu_d} + \mathbf{c}_k \quad (24)$$

$$\theta_{k+1} = \frac{(\overrightarrow{cp} \times \mathbf{l})_k^z \odot \text{sign}(\overrightarrow{cp} \times \mathbf{l})_k^z}{2\mathbf{I}_{zz} C_f} + \theta_k \quad (25)$$

The decision variables are impulses \mathbf{l}_k and application points p . We can conveniently express point p in the object's frame \mathcal{F}_{obj} with \mathbf{c} as the origin. At each step, k , the object's new orientation, and position will be obtained by a straightforward transform.

B. Constraints

The optimization problem embeds several constraints related to the object's shape, the poking tool used, the robot's dynamics, and the global strategy. The shape contour is described in the object's frame \mathcal{F}_{obj} as a parametric function $\gamma(u)$ (see Figure 1). Depending on the real object's shape, this function might not be differentiable $\forall u$; to address this, we approximate it by a quadratic function (other parametrized curves such as Nurbs can be used), leading to a closed-form tangent space $\dot{\gamma}(u)$ and normal space $\ddot{\gamma}(u)$. To hit the object efficiently, the impulse vector must lie within the static friction cone with coefficient μ_s on point p [6]. We constraint the angle α between the normal $\ddot{\gamma}(u)$ and the impulse \mathbf{l} as follows:

$$-\arctan(\mu_s) < \alpha < \arctan(\mu_s) \quad (26)$$

Additionally, the impact point must be within the set $\mathcal{X}_{\text{workspace}}$ of \mathcal{W} , and the impulse vector must be within the impulse space $\mathcal{X}_{\text{impulse}}$ (16).

$$p \in \mathcal{X}_{\text{workspace}} \quad (27)$$

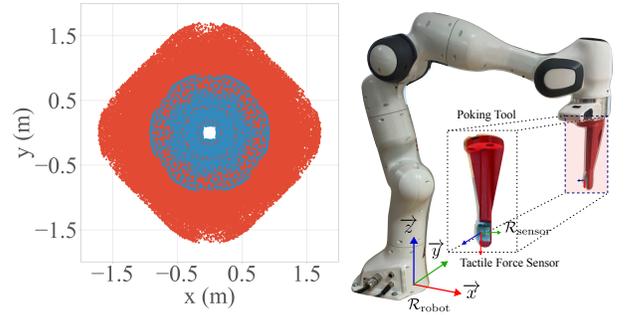


Figure 2: Left: Panda's reachable workspace \mathcal{W} (upper view, in blue) and the extended reachable-by-poking workspace (in red), considering hardware limits specified by the manufacturer with an object's mass of 0.9 kg, a coefficient of friction $\mu_d = 0.28$ and a plan $z = 0.1$ m. Right: The experimental setup with a poking tool (equipped with a tactile sensor) mounted at the end of the robot.

$$\mathbf{l} \in \mathcal{X}_{\text{impulse}} \quad (28)$$

To consider the friction cone at the poking point, \mathbf{l}_k is expressed by its angle of impact α_k and norm l for every poke. The point p_k has two components for our 2D problem. Therefore, the decision variables vector can be detailed as:

$$\mathcal{X}_{\text{variables}} = \{k, k \cdot [l], k \cdot [\alpha], k \cdot [p_x], k \cdot [p_y]\} \quad (29)$$

Using (21) and corresponding weights w_1, w_2, w_3 , the optimization problem is defined as multi-objective MINLP:

$$\begin{aligned} & \underset{\mathcal{X}_{\text{variables}}}{\text{Minimize}} && w_1 g_1 + w_2 g_2 + w_3 g_3 \\ & \text{Subject to} && \text{Impulse angular limitation (26)} \\ & && \text{Workspace (27)} \\ & && \text{Impulse space (28)} \end{aligned} \quad (30)$$

that can be solved in an efficient way using either derivative-free optimization methods (e.g., Genetic Algorithms, Particle Swarm...) or gradient-based solvers under CasADi [23], e.g., multiple NLPs, such as Bonmin or Knitro. We have been able to solve it at a 10 Hz frequency using C++ genetic algorithm implementation.

V. EXPERIMENTAL SETUP

The overall framework of the proposed poking method, see Fig. 3, includes planning, tracking, and perception. First, the frictional parameters of the contact surface and the box's center of mass \mathbf{c} are estimated. Then, a target pose is defined in the task frame. By solving the problem described in Sec III, the outputs generated by the optimization solver, $\mathcal{X}_d = \{k, k \cdot [l], k \cdot [\alpha], k \cdot [p_x], k \cdot [p_y]\}_d$ are fed into Poking-Aware-IK (Sec V-A) to determine the set of equivalent joint space $\mathcal{Q}_i = \{\mathbf{q}_{\text{ipoke}}, \dot{\mathbf{q}}_{\text{ipoke}}\}$. Subsequently, the reference trajectories are planned by an online trajectory generator (OTG) for each of these configurations from different launching poses defined by the transition motion policy (Sec V-C). Finally, our QP-based tracking executes the trajectory under the embedded safety constraints and collision avoidance. All the experiments are conducted using a 7 Degrees of Freedom (DoF) Panda robot to poke containers of different shapes (mainly boxes). The poking reachable workspace (Figure. 2) is estimated using

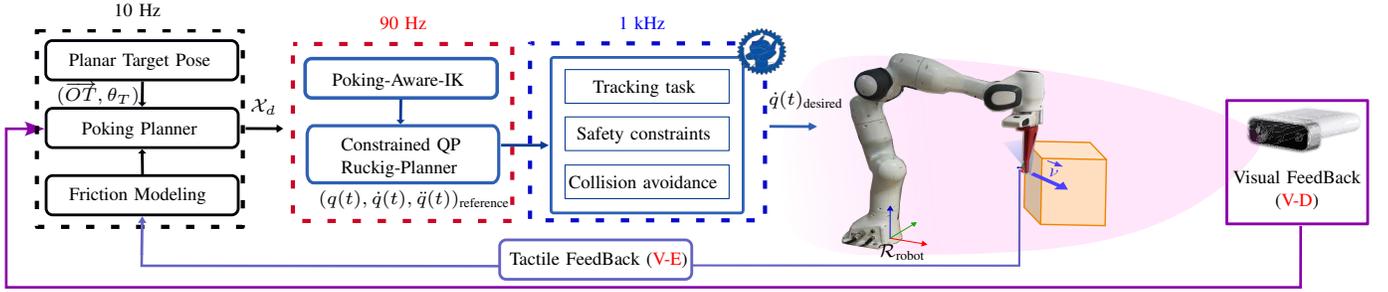


Figure 3: The proposed model-based poking framework, starting from modeling and poke planning to task execution (tracking) through a QP formulation combining an IK-solver and Jerk-Limited planning. Sensory inputs include tactile feedback (from the tactile sensor mounted on the poking tool) and visual feedback (from the Kinect Azure camera).

the robotic toolbox [24], with a fixed rotation of the poking tool. The entire implementation is done using the open-source C++ framework `mc_rtc`¹ along with the corresponding robot module and control interface `mc_panda` and `mc_franka`.

A. Poking-aware Inverse Kinematics

The optimization solver returns poking points and impact vectors in the task-space. To express this information in the configuration space, we first perform inverse kinematics using `IKFast` [25]. For each given optimization solution $(p, \nu)_k$, we compute a batch of joint position solutions and corresponding mass matrices. Then, to control ν_k , we compute the effective mass $\Lambda(q_k) = (JI^{-1}J^T)^{-1}$ allowing us to find a feasible joint velocity by estimating the corresponding task-space velocity as expressed by V^- in eq (15).

B. Poking Trajectories

Poking task trajectories are generated in the configuration space by the `OTG Ruckig` [26]. The latter considers third-order constraints and calculates valid extremal profiles for all DoFs, taking into account the blocked intervals for each. The result is a joint-space reference trajectory $\{\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t)\}_{\text{ref}}$. This planner is combined with a QP posture problem with well-defined static environment obstacles to ensure collision-free trajectories. Up to this point, which corresponds to the red dotted square in Figure 3, the process runs at around 90 Hz whenever it is called.

The planned trajectories are then tracked using a QP-controller [27] that operates at realtime frequency of 1 kHz (blue dotted square in Figure 3), in which the main task is defined as follows:

$$\min_{\ddot{\mathbf{q}} \in \mathbb{R}^n} \|\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{\text{desired}}\| \quad (31a)$$

$$\text{s.t: } \mathcal{C}(\dot{\mathbf{q}}, \mathbf{q})\ddot{\mathbf{q}} + \mathbf{D} \leq \mathbf{0} \quad (31b)$$

$$\text{Collisions Constraints} \quad (31c)$$

where the desired task acceleration is: $\ddot{\mathbf{q}}_{\text{desired}} = \ddot{\mathbf{q}}_{\text{ref}} - K_s(\mathbf{q} - \mathbf{q}_{\text{ref}}) - K_d(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{ref}})$, here, K_s and K_d are strictly positive gains for stiffness and damping of the task respectively, $\{\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}\}_t \in \mathbb{R}^{3n}$ represent the position, velocity, acceleration in the configuration space of dimension n , $\mathcal{C}(\dot{\mathbf{q}}, \mathbf{q})$ is the constraints matrix related to the robot's limits (position,

velocity, acceleration, torque, torque derivatives, and jerk) and \mathcal{D} is the bounds values vector of all these constraints. The collision constraints represent the set of defined collisions with dynamic obstacles while tracking, such as boxes to be poked in undesired locations and other robots.

In order to find suitable solutions, we can use all sets of joint configurations \mathbf{q} for each pose P , $(JI^{-1}J^T)^{-1}$ and V^- previously computed and avoid non-feasible trajectories.

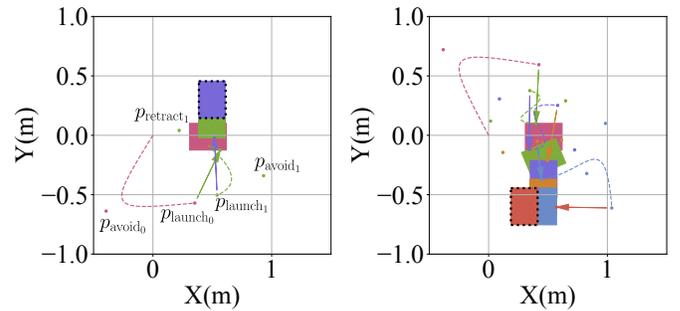


Figure 4: Transition motion policy for two poking planning examples.

C. Transition Motion Policy

Between each two consecutive pokes, it is paramount to ensure that the robot moves close to the next poking position while avoiding the box (see Figure 4). To achieve this, we established a simple ad-hoc policy in 2D. Based on the next poking point p_k , we first identify a nearby point p_{launch} , placed at a sufficient distance to generate the impulse and positioned in the scaled opposite direction of the impulse vector. To ensure the robot does not collide with the box during this displacement, we create a second point p_{avoid} equidistant from p_{launch} and the initial point, which can be either the previous poking point or the resting pose. There will be two potential solutions for p_{avoid} , and the one closest to p_{launch} is chosen. Finally, we add a third point p_{retract} at a distance equal to the longest dimension of the box (here, the diagonal) from the initial point, in a direction normal to the impulse vector and closer to p_{avoid} . The resulting trajectory is planned and executed using a QP-based position task, with continuous collision constraints enforced to avoid undesired contacts.

D. Visual Feedback

Visual feedback is used for high-level control of the poking task using a Kinect Azure RGB-D camera and a tracking

¹https://jrl.cnrs.fr/mc_rtc/index.html

software developed in [28]. The latter operates on a multi-threading configuration at a frequency of 30 Hz.

E. Tactile Feedback

In order to control all the contact parameters introduced, we use a tactile sensor, described in [29], see Figure 2, to measure the force during the poking task in the early stages to estimate and monitor impacts. This sensor is mounted on a dedicated 3D-printed tool designed for poking and housing the sensor. This latter operates at around 100 Hz.

VI. EXPERIMENTS RESULTS

In our poking experiments, we used a set of cardboard boxes with varying masses and dimensions (see Table I). Each box is poked on a wooden surface (table), and we selected arbitrarily initial and final desired poses and locations for each trial. We discuss further the estimation of the friction parameters, and the results obtained from poking experiments. We also ran our poking algorithms on non box-shaped objects (see Figure 5); including a cylindrical object (0.39 Kg) and triangular prism (0.41 Kg), to validate its effectiveness (see Figure 10). Videos of our experiments are available in a supplementary material². The overall planning time for the poking in our trails was at 100 ms, whereas the QP-Jerk-limited planning block checks the trajectory feasibility in 11 ms. The QP-based controller tracks these trajectories at 1 ms.

Table I. Parameters of the set of boxes used.

	m (kg)	L (m)	w (m)	h (m)
Box 1	0.21	0.237	0.199	0.102
Box 2	0.32	0.240	0.145	0.1
Box 3	0.91	0.31	0.225	0.11
Box 4	1.12	0.349	0.206	0.099

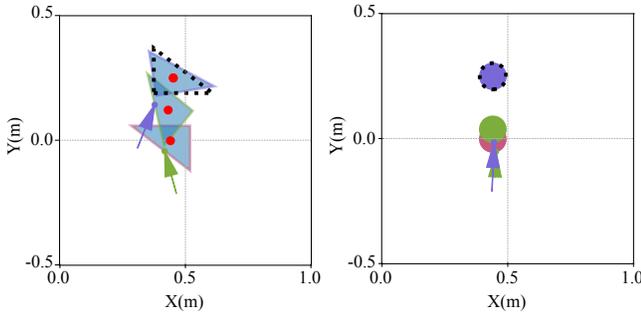


Figure 5: Examples on minimal poking applied to different shapes, triangular prism and cylinder.

A. Estimation of μ_d and e_r

We estimate boxes' planar velocity and displacement using vision and eq. (3). The dynamical coefficient of friction μ_d is estimated from:

$$\mu_d = \frac{\|\dot{\mathbf{c}}^0\|^2}{2g\|\mathbf{c}_i\|} \quad (32)$$

where $\|\mathbf{c}_i\|$ is distance travelled after the initial poke. For the static coefficient of friction μ_s , an offline estimation is done

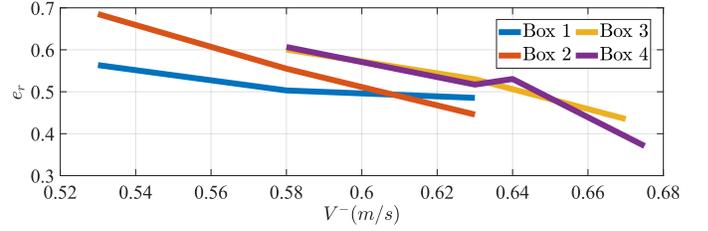


Figure 6: Experimental coefficient of restitution estimation using measured velocities from encoder and visual feedback.

for objects with known-mass by relying on force sensor measurements when start sliding under uniform constant velocity pushing, $\mu_s = \|\mathbf{f}_{\text{measured}}\|/mg$ [8].

Theoretical models used to estimate the coefficient of restitution, e.g., [30], suffer from significant number of unknown or uncertain parameters (e.g., stress, strain... of the colliding pair of objects) when applied to real-world use-cases. Alternatively, we can experimentally estimate the e_r parameter from (15):

$$e_r = \frac{m + \lambda}{\lambda} \frac{\dot{c}^+}{V^-} - 1 \quad (33)$$

We conducted tests on the set of boxes in Table I by poking along the y -direction with different velocities and a fixed λ (around 3.5 kg). The results are reported in Figure 6. For boxes 3 and 4, the velocity that produces noticeable motion of the boxes is higher than that of the boxes 2 and 1. This is due to the difference of weights. It can be noticed that the variation of e_r is inversely proportional to the pre-impact robot velocity V^- for all boxes. Our results from equation (33) align with those in [31, Eq. 20] presenting a method to estimate e_r .

B. Experimental Results for Poking Tasks

1) *Positional and Orientational Accuracy*: From open-loop trials on experiments' video, our approach is efficient regarding translational motions. It achieved limited translational error $\|e_t\| \leq 0.05$ m. Whereas rotation motions are less precise. In combined motions, this would affect both the sliding and spinning results. This is due to the sensitivity regarding the poke point p . This is also because of the variability of the frictional rotational torque C_f due to either anisotropic coefficient of friction, non-planar contact, or non-homogeneous mass distribution on the surface S . Figures (7-10) show that these limitations can be compensated for using visual feedback. We achieved a reasonable rotational error $\|e_{\text{rot}}\| \leq 0.2$ rad.

2) *Impact Reconstruction*: In Figure 11, we show the reconstructed impact value by integrating the force measurements from the tactile sensor, assuming an impact duration of $T = 20$ ms. The estimated impact is consistent with the desired one with negligible energy dissipated due to the deformation of the silicone material of the tactile sensor. This indicates the validity of the required post-impact velocity defined in eq (15).

3) *Trajectory Tracking*: Figure 12 depicts the task-space planar errors for the position and velocity targets, alongside the actual measured position and velocity of the end-effector. The targets at the moment of impact are represented by dashed straight lines. These targets are met only at the desired poking

²<https://seafile.lirmm.fr/f/d5b7c9b3da054e599bf0/>

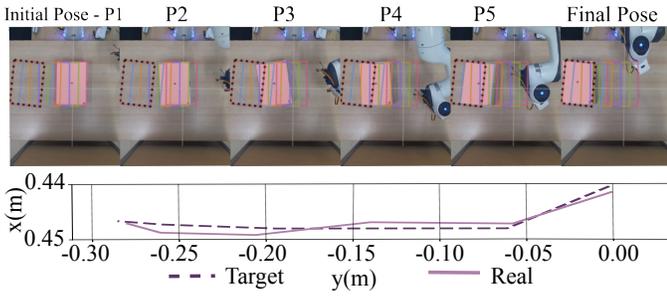


Figure 7: Closed-loop poking of a box to a goal pose by successive sliding, demonstrating the convergence of the real planar motion to the planned trajectory.

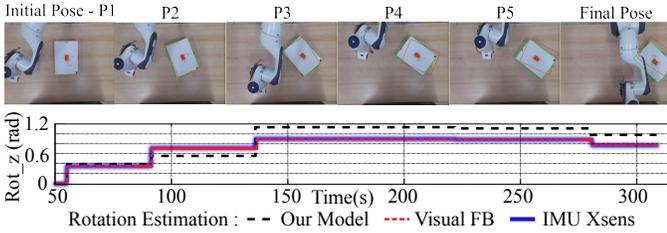


Figure 8: Closed-loop poking of a box to a goal pose, showing the planner's spinning angle differences between the model and two ground truth sources (Xsens and visual feedback).

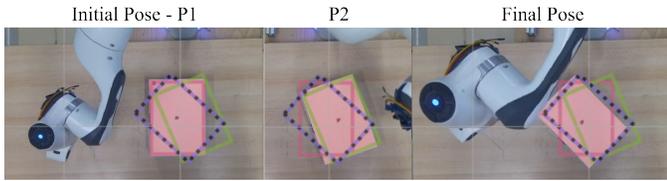


Figure 9: Closed-loop poking performing stationary rotation.

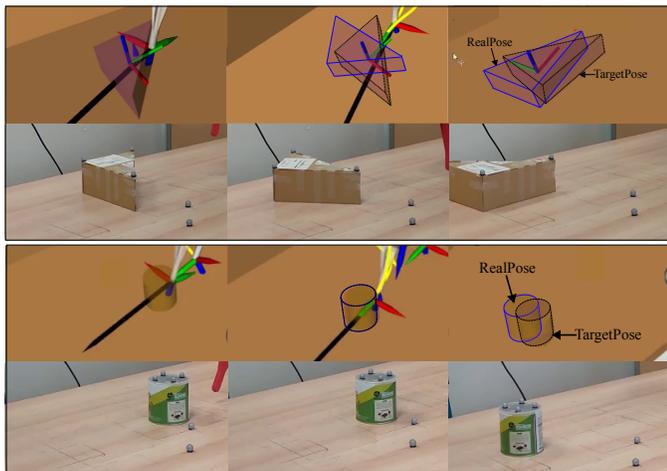


Figure 10: Examples of poking on triangular prism and cylindrical object. Targets are indicated by a dark dashed outline, whereas the real measurements, obtained from a motion capture system, are shown in blue outline.

point on the object's surface. The results show that the real position and velocity of the robot align closely with the desired values, exhibiting only a small error during the impact moment represented by the `Impact Instant` long-dashed line within the light-gray area. This indicates a high accuracy in trajectory tracking performance.

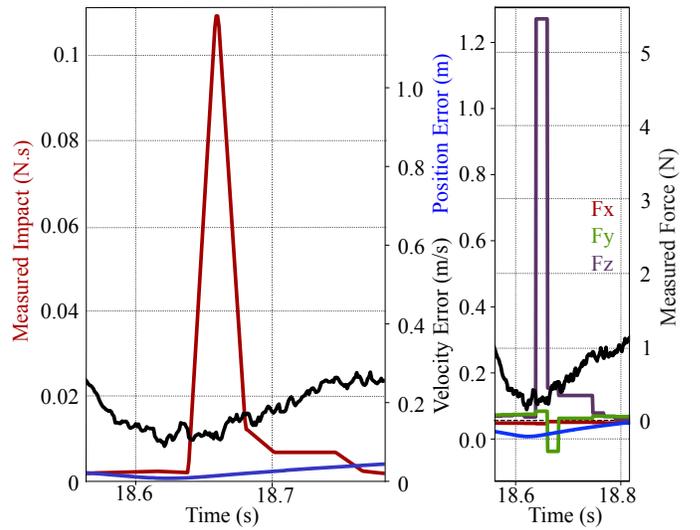


Figure 11: Reconstructed impulse (left) from tactile force measurements (right) at the moment of poking, along with task-space errors: position (blue) and velocity (black) relative to the poking point.

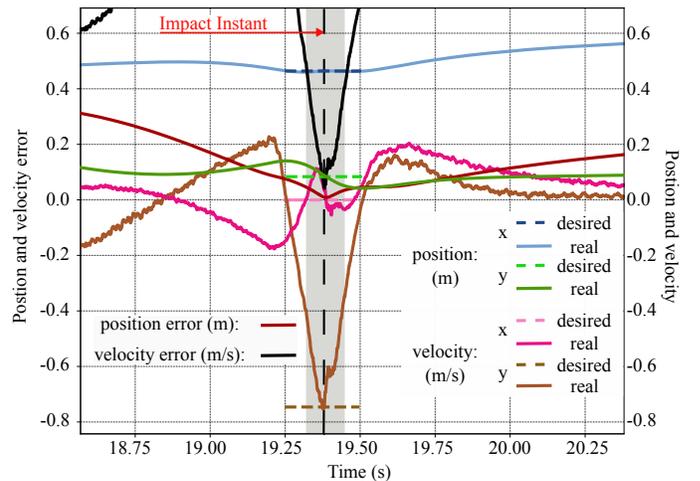


Figure 12: Task-space position and velocity errors, showing the real and target poking point planar positions (x , y) and the corresponding velocities.

C. Discussions and limitations

In contrast to [8] and [5], our model-based poking approach not only plans for sliding motions but also addresses spinning motions that have not been previously investigated. The experimental results show that the actual position and rotation of the object are close to the target values for the poking task, suggesting the efficacy of our method. During the experiments, we observed that the assumptions regarding the nature of contact were highly significant. We are aware that the coefficient of restitution variability w.r.t changes in impact velocity during the poking task. While the sliding model is relatively insensitive to the nature of the contact surface, the rotational model suffers from inaccuracies. In the point-contact mode, for example, the object tends to rotate more than predicted by the model [32]. Additionally, the contact force distribution related to inertial parameters could affect both the sliding and spinning models. Our method relies on the friction model, and therefore, we can integrate a more complex approach to

address problems with nonuniform μ_d on the support surface or the object. This model is also highly dependent on the physics attribute of the object; hence, we could improve our method by incorporating initial measurements of μ , e_r , C_f or I through a few preliminary pokes. Another issue is the robot's impact-aware control, where the torque derivatives limits can sometimes be violated during impact. Implementing an efficient constraint on the trajectory for this measure will lead to greater robustness. Finally, we demonstrated a scenario where poking can be combined with another robotic tasks (tossing). This shows that the purpose of poking is not only to increase the robot's workspace as it also allows collaborative process in a multi-robot setting to accelerate tasks, especially in logistics [33].

VII. CONCLUSION AND FUTURE WORK

In this letter, we present a model-based method for determining an optimal sequence of pokes to achieve the desired planar motion of an object using a robotic manipulator. The experimental results demonstrate that our approach can always provide feasible solutions for the trajectory planning with high robustness for robot control, and it can move the object to the desired position and orientation with minimal error and using a limited number of pokes. Given this accuracy and efficiency, the method has significant potential for industrial applications, particularly in addressing logistic challenges and simulating object motion. We delve into contact mechanics and robot control under impacts, providing all the necessary information and guidance to implement this process on any robot at a low cost, utilizing open-source tools. While visual feedback control is not mandatory for poking tasks, it could significantly improve the precision of the results depending on the requirements of specific applications. We have explored the use of impact-aware QP control for this task. Still, it may be beneficial to find an optimal solution for the effective mass $\Lambda_{(q)}$ by introducing it to the mechanical constraints of the robot's structure. In future work, we plan to integrate this approach with various model-based modules, (e.g., tossing, grabbing, picking) to create a dedicated workflow for handling logistics packages in diverse scenarios.

REFERENCES

- [1] A. Pasricha, Y.-S. Tung, B. Hayes, and A. Roncone, "Pokerrt: Poking as a skill and failure recovery tactic for planar non-prehensile manipulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4480–4487, 2022.
- [2] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez, "More than a million ways to be pushed: a high-fidelity experimental dataset of planar pushing," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 30–37.
- [3] M. Bauza and A. Rodriguez, "A probabilistic data-driven model for planar pushing," in *IEEE Int. Conf. Robot. Automat.*, 2017, pp. 3008–3015.
- [4] M. Xiao, Y. Ding, and S. Fan, "One-finger manipulation of 3d objects by planning start-to-push points and pushing forces," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2694–2701, 2024.
- [5] H. Khurana and A. Billard, "Motion planning and inertia-based control for impact aware manipulation," *IEEE Trans. on Robotics*, vol. 40, pp. 2201–2216, 2024.
- [6] Y. Wang, N. Dehio, A. Tanguy, and A. Kheddar, "Impact-aware task-space quadratic-programming control," *Int. J. Robot. Res.*, vol. 42, no. 14, pp. 1265–1282, 2023.
- [7] A. Ö. Önel, R. Corcodel, P. Long, and T. Padr, "Tuning-free contact-implicit trajectory optimization," in *IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1183–1189.
- [8] B. Liang, W. Liang, and Y. Wu, "Tactile-guided dynamic object planar manipulation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 3203–3209.
- [9] M. Toussaint, J.-S. Ha, and D. Driess, "Describing physics for physical reasoning: Force-based sequential manipulation planning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6209–6216, 2020.
- [10] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-implicit trajectory optimization for dynamic object manipulation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6814–6821.
- [11] A. Kloss, S. Schaal, and J. Bohg, "Combining learned and analytical models for predicting action effects from sensory data," *Int. J. Robot. Res.*, vol. 41, no. 8, pp. 778–797, 2022.
- [12] X. Yi and N. Fazeli, "Precise object sliding with top contact via asymmetric dual limit surfaces," in *Robotics: Science and Systems*, 2023.
- [13] M. Wang, A. Ö. Önel, P. Long, and T. Padr, "Contact-implicit planning and control for non-prehensile manipulation using state-triggered constraints," in *Robotics Research*. Springer Nature Switzerland, 2023, pp. 189–204.
- [14] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," *Advances in neural information processing systems*, vol. 29, 2016.
- [15] A. Ö. Önel, P. Long, and T. Padr, "Contact-implicit trajectory optimization based on a variable smooth contact model and successive convexification," in *IEEE Int. Conf. Robot. Automat.*, 2019, pp. 2447–2453.
- [16] T. Stouraitis, L. Yan, J. Moura, M. Gienger, and S. Vijayakumar, "Multi-mode trajectory optimization for impact-aware manipulation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9425–9432.
- [17] 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.
- [18] Z. Farkas, G. Bartels, T. Unger, and D. E. Wolf, "Frictional Coupling between Sliding and Spinning Motion," *Physical Review Letters*, vol. 90, no. 24, 2003.
- [19] G. A. Waltersson and Y. Karayiannidis, "Planar friction modeling with LuGre dynamics and limit surfaces," *IEEE Trans. on Robotics*, vol. 40, pp. 3166–3180, 2024.
- [20] K. L. Johnson, *Contact mechanics*. Cambridge university press, 1987.
- [21] Y. Wang, N. Dehio, and A. Kheddar, "Predicting Impact-Induced Joint Velocity Jumps on Kinematic-Controlled Manipulator," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6226–6233, Jul. 2022.
- [22] I. Walker, "Impact configurations and measures for kinematically redundant and multiple armed robot systems," *IEEE Trans. on Robotics and Automation*, vol. 10, no. 5, pp. 670–683, 1994.
- [23] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [24] P. Corke and J. Haviland, "Not your grandmother's toolbox—the robotics toolbox reinvented for python," in *IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11 357–11 363.
- [25] R. Diankov and J. Kuffner, "Openrave: A planning architecture for autonomous robotics," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, vol. 79, 2008.
- [26] L. Berscheid and T. Kröger, "Jerk-limited real-time trajectory generation with arbitrary target states," in *Robotics: Science and Systems*, 2021.
- [27] K. Bouyarmane and A. Kheddar, "On weight-prioritized multitask control of humanoid robots," *IEEE Trans. on Automatic Control*, vol. 63, no. 6, pp. 1632–1647, 2017.
- [28] M. Stoiber, "Closing the loop: 3d object tracking for advanced robotic manipulation," Ph.D. dissertation, Technische Universität München, 2023.
- [29] Y. Yan, Z. Hu, Z. Yang, W. Yuan, C. Song, J. Pan, and Y. Shen, "Soft magnetic skin for super-resolution tactile sensing with force self-decoupling," *Science Robotics*, vol. 6, no. 51, 2021.
- [30] R. L. Jackson, I. Green, and D. B. Marghitu, "Predicting the coefficient of restitution of impacting elastic-perfectly plastic spheres," *Nonlinear Dynamics*, vol. 60, pp. 217–229, 2010.
- [31] I. Green, "The prediction of the coefficient of restitution between impacting spheres and finite thickness plates undergoing elastoplastic deformations and wave propagation," *Nonlinear Dynamics*, vol. 109, no. 4, pp. 2443–2458, 2022.
- [32] A. Zermane, "Impact driven programmed robots," Ph.D. dissertation, University of Montpellier, 2024.
- [33] A. Zermane, N. Dehio, and A. Kheddar, "Planning impact-driven logistic tasks," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2184–2191, 2024.