On-Palm Dexterity: Dynamic Reorientation of Objects via Emergent Flipping and Sliding

Zhixian Xie[1], Wen Yang[1], Wanxin Jin[1]
[1]Arizona State University, Tempe, AZ, USA
{zxxie, wenyang, wanxin.jin}@asu.edu

***Abstract*—We develop a real-time, contact-implicit model-based planning and control framework for highly dynamic, contact-rich manipulation tasks, such as object reorientation through flipping and non-prehensile sliding. At its core, our framework extends a recently proposed complementarity-free contact model to full dynamic settings, and integrate it into real-time contact-implicit model predictive control (MPC) for dynamic contact reasoning, followed by local velocity impedance control for robotic actuation. Enabled by our dynamic complementarity-free contact modeling, the contact-implicit MPC real-time optimizes the manipulation goal and generate emergent contact behavior (flipping, sliding) in highly dynamic scenarios without any motion hints such as reference trajectories or motion primitives. We validate our framework in both simulation and real-world experiments for on-palm dynamic reorientation tasks, including flipping and sliding, with various object shapes. Our framework achieves high success rates across different reorientation targets and objects, demonstrates emergent contact reasoning, and exhibits strong performance and robustness against external disturbances. Our complementarity-free contact-implicit MPC runs at 50-100Hz. The real-world demo is shown in video link.**

I. Introduction

Contact-rich manipulation, such as in-hand reorientation [1]–[3], tool use [4], and dexterous grasping [5]–[7], remains a challenge in robot autonomy. Classical model-based approaches [8]–[10], which rely on complementarity-based contact models, often face significant computational challenges in fast trajectory optimization or model predictive control (MPC), due to the discontinuous and hybrid nature of contact dynamics. Thus, most methods are limited to static or quasi-static tasks [8], [10], or scenarios where contact interactions are sparse and enumerable [9]. Recent progress in fast solvers for complementarity-constrained optimization [11] has pushed the boundaries of model-based methods toward more dynamic, non-prehensile tasks [12]. Nevertheless, these approaches remain constrained to simplified settings, such as planar (2D) manipulation or tasks involving slow object motions.

In this work, we aim to advance *dynamic* contact-rich manipulation by building on a recently proposed complementarity-free multi-contact modeling [13]. This model removes the need for complementarity constraints in contact dynamics through a simple, closed-form formulation, enabling real-time optimization while still capturing Coulomb friction and discrete contact modes. While the original formulation [13] was limited to quasi-dynamic settings, we extend it to full dynamic regimes to model highly dynamic contact interactions, such as on-palm object flipping and 3D sliding. We refer to this extension as the *dynamic complementarity-free contact model*. We incorporate this model into a MPC framework for contact dynamic contact reasoning, and then followed by a local velocity-impedance control for robot actuation. Our framework enables real-time contact-implicit planning and control for challenging dynamic manipulation tasks like on-palm object reorientation via flipping and sliding without any motion hints.

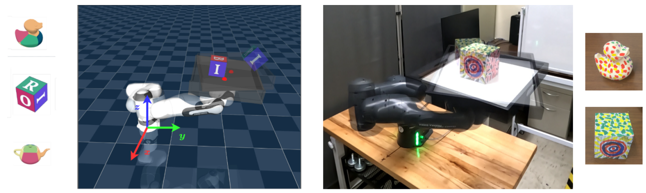


Fig. 1: On-palm dynamic reorientation via emergent flipping and sliding. Left: evaluation in MuJoCo environment. Right: evaluation in real-world hardware.

We evaluate our framework in simulation and on hardware across a range of on-palm dynamic reorientation tasks involving diverse objects (Fig. 1). The results demonstrate that our approach achieves high success rates across various reorientation targets and object geometries, exhibits emergent contact reasoning, and maintains strong performance and robustness. To our knowledge, this is the first model-based, contact-implicit method to successfully perform such dynamic 3D flipping and sliding reorientation of objects in real time.

A. Related Works

a) Contact Modeling: Contact physics is traditionally modeled as nonlinear complementarity problems (NCPs), as contact forces and inter-object distances naturally form a complementarity pair [14], [15]. To efficiently solve NCPs, some approximation techniques [16], [17] replace the friction cone with a polyhedral cone, transforming the problem into linear complementarity problems (LCPs). Alternatively, NCPs can be reformulated as Cone Complementarity Problems (CCP) by introducing a constraint that enforces the contact velocity to lie within the dual friction cone [18], [19]. This formulation allows contact force-velocity to be resolved via an optimization [16], [20], offering computational advantages. Other approaches aim to achieve differentiable contact model by employing log-barrier functions [21], [22] and carefully designed penalty terms [23], [24]. These methods solve for a residual equation, and gradients are computed using the implicit function theorem.

Our work builds on the complementarity-free contact modeling recently developed in [13]. The model computes contact forces or next velocity using simple and closed-form equations, eliminating the need for optimization-based dynamics. Thus, it enables integration into MPC for real-time contact-implicit planning and control. While simple, the complementarity-free model is capable of capturing complex contact modes [13].

b) Planning and Control with Contact Models: The discontinuity of initiating and breaking contacts poses significant challenges in contact planning. Classic methods use predefined contact profiles to guide the robot, enabling control and planning in relatively simple settings [25]–[27]. Modern methods focus on contact-implicit optimization, which often relies on relaxing the complementarity conditions [8]–[10] or incorporating mixed-integer formulations into the planning problem [11], [28], [29]. Alternatively, sampling-based approaches [30], [31] can be employed, which handle non-differentiable contact dynamics by iteratively sampling to search for optimal control sequences.

Our framework leverages the dynamic complementarity-free contact model within a contact-implicit MPC scheme. The MPC performs online contact reasoning to optimize manipulation goal and outputs the optimal robot velocity command, which is tracked by a lower-level velocity impedance controller for actuation. Enabled by the dynamic complementarity-free contact model, our contact-implicit MPC runs at 50–100 Hz, allowing the system to output emergent contact-rich behavior for dynamic contact-rich manipulation without any motion aid.

II. Method

A. Dynamic Complementarity-Free Contact Model

We model a dynamic manipulation system, including an actuated robots and unactuated object, as shown in Fig. 1, with the following discrete-time dynamic model:

$$\begin{aligned} \boldsymbol{M}_o(\boldsymbol{q}_o)(\boldsymbol{v}_o^+ - \boldsymbol{v}_o) &= h\boldsymbol{\tau}_o(\boldsymbol{q}_o, \boldsymbol{v}_o) + \sum\nolimits_{i=1}^{n_c} \boldsymbol{J}_{o,i}^{\mathsf{T}}(\boldsymbol{q}_o)\boldsymbol{\lambda}_i \\ h\boldsymbol{K}_v(\boldsymbol{v}_r^+ - \boldsymbol{u}) &= \sum\nolimits_{i=1}^{n_c} \boldsymbol{J}_{r,i}^{\mathsf{T}}(\boldsymbol{q}_r)\boldsymbol{\lambda}_i \end{aligned} \tag{1}$$

Here, $\boldsymbol{M}_o$ is the inertia matrix of the object. $\boldsymbol{q} = [\boldsymbol{q}_o; \boldsymbol{q}_r]$ is the generalized current position of the system, where $\boldsymbol{q}_o$ and $\boldsymbol{q}_r$ stand for the object pose and robot joints, respectively. $\boldsymbol{v}_o, \boldsymbol{v}_o^+$ are the object's (linear and angular) velocity at current and the next time steps, respectively. $\boldsymbol{\tau}_o$ is the non-contact force acting on object, including inertia, Coriolis, and gravity forces. For the robot arm, $\boldsymbol{v}_r, \boldsymbol{v}_r^+$ are the robot's joint velocity at current and the next time steps, respectively. We considered the robot is controlled by a lower-level joint velocity impedance controller, which leads to "damping" response with respect to the external force (here as the contact forces) as shown the second equation. $\boldsymbol{u}$ is the command velocity. $\boldsymbol{K}_v$ is the closed-loop damping parameters. $\boldsymbol{\lambda}_i$ is i-th contact impulse, while $\boldsymbol{J}_{o,i}$ and $\boldsymbol{J}_{r,i}$ are the contact Jacobians for the object and robot, respectively. Notably, we assume the external force (majorly gravity) applied on the robot is compensated and does not include it in the dynamic equations.

We distinguish our above dynamic manipulation model from the previous work [13] in the following aspects. First, the first equation exactly considers the full Newton–Euler dynamics of the object, compared to the quasi-dynamic model (ignoring the inertia force effects) in [13]. This enables us to model the highly dynamics manipulation tasks, such as flipping and sliding. Second, the robot is controlled by a joint velocity impedance controller thus has a close-loop velocity impedance dynamics. This is different from positional impedance control in [13]. This enables smoother robot motion in dynamic settings.

By defining velocity $\boldsymbol{v} = [\boldsymbol{v}_o; \boldsymbol{v}_r]^{\mathsf{T}}$, we write (1) compactly

$$h^2\boldsymbol{Q}\boldsymbol{v}^+ = h\boldsymbol{b} + \sum\nolimits_{i=1}^{n_c} \boldsymbol{J}_i^{\mathsf{T}}\boldsymbol{\lambda}_i, \tag{2}$$

where the contact Jacobian is defined as $\boldsymbol{J}_i = \left[\boldsymbol{J}_{o,i}^{\mathsf{T}}\, \boldsymbol{J}_{r,i}^{\mathsf{T}}\right]^{\mathsf{T}}$, and matrix $\boldsymbol{Q}$ and $\boldsymbol{b}$ are defined below respectively:

$$\boldsymbol{Q}= \begin{bmatrix} \boldsymbol{M}_o(\boldsymbol{q}_o)/h^2 & 0 \\ 0 & \boldsymbol{K}_v/h \end{bmatrix}, \boldsymbol{b}= \begin{bmatrix} \boldsymbol{\tau}_o(\boldsymbol{q}_o, \boldsymbol{v}_o) + \boldsymbol{M}_o(\boldsymbol{q}_o)\boldsymbol{v}_o/h \\ \boldsymbol{K}_v\boldsymbol{u} \end{bmatrix}. \tag{3}$$

The i-th contact impulse $\boldsymbol{\lambda} = \boldsymbol{\lambda}_i^n + \boldsymbol{\lambda}_i^d$ includes the normal $\boldsymbol{\lambda}_i^n$ and friction $\boldsymbol{\lambda}_i^d$ components. To handle Coulomb friction, we linearize the friction cone as a polyhedral cone with n_d-faces. Particularly, we define a symmetric set of n_d unit vectors $\{\boldsymbol{d}_{i,j}\}_{j=1}^{n_d}$ that span the tangential contact plane. Following [13], the contact impulse $\boldsymbol{\lambda}_i$ can be parameterized by a set of n_d non-negative parameters $\boldsymbol{\beta}_i = [\beta_{i,1}, \beta_{i,2}, ..., \beta_{i,n_d}]^{\mathsf{T}}$:

$$\boldsymbol{\lambda}_i^n = \sum\nolimits_{j=1}^{n_d} \beta_{i,j}\boldsymbol{n}_i, \quad \boldsymbol{\lambda}_i^d = \mu \sum\nolimits_{j=1}^{n_d} \beta_{i,j}\boldsymbol{d}_{i,j}, \tag{4}$$

with $\beta_{i,j} \geq 0$ for all $j = 1, 2, ..., n_d$. The above parameterization of normal and friction components can automatically satisfy the Coulomb friction law, as shown in [13].

Let $\boldsymbol{J}_i^n$ be the contact Jacobian corresponding to the normal unit vector $\boldsymbol{n}_i$, and $\{\boldsymbol{J}_{i,j}^d\}_{j=1}^{n_d}$ to the respective tangential unit vector $\{\boldsymbol{d}_{i,j}\}_{j=1}^{n_d}$. With the linearize friction cone and parameterized contact impulse (4), (2) can be written as

$$\boldsymbol{v}^+ = \frac{1}{h^2}\boldsymbol{Q}^{-1}(h\boldsymbol{b} + \tilde{\boldsymbol{J}}^{\mathsf{T}}\boldsymbol{\beta}) \tag{5}$$

with the new Jacobian $\tilde{\boldsymbol{J}}$ and the contact parameter vector $\boldsymbol{\beta}$ are defined together as:

$$\tilde{\boldsymbol{J}} := \begin{bmatrix} \boldsymbol{J}_1^n - \mu_1\boldsymbol{J}_{1,1}^d \\ \cdots \\ \boldsymbol{J}_1^n - \mu_1\boldsymbol{J}_{1,n_d}^d \\ \vdots \\ \boldsymbol{J}_{n_c}^n - \mu_{n_c}\boldsymbol{J}_{n_c,1}^d \\ \cdots \\ \boldsymbol{J}_{n_c}^n - \mu_{n_c}\boldsymbol{J}_{n_c,n_d}^d \end{bmatrix}, \ \boldsymbol{\beta} := \begin{bmatrix} \beta_{1,1} \\ \cdots \\ \beta_{1,n_d} \\ \vdots \\ \beta_{n_c,1} \\ \cdots \\ \beta_{n_c,n_d} \end{bmatrix}. \tag{6}$$

Following the complementarity-free contact resolution in [13], the vector $\boldsymbol{\beta}$ can be computed in closed form using:

$$\boldsymbol{\beta} = \max\big(- h\boldsymbol{K}(\tilde{\boldsymbol{J}}\boldsymbol{Q}^{-1}\boldsymbol{b} + \tilde{\boldsymbol{\phi}}), 0\big), \tag{7}$$

where $\tilde{\boldsymbol{\phi}} = [\phi_1, \ldots, \phi_1, \ldots, \phi_{n_c}, \ldots, \phi_{n_c}]^{\mathsf{T}}$ is the (extended) vector of collision distances, and $\boldsymbol{K}$ is a stiffness matrix which is a hyperparameter. As analyzed in [13], the above contact impulse resolution has an intuitive physical interpretation: $\boldsymbol{\beta}$ is the "spring-like" forces due to compressing (penetrating) the dual friction cone by the depth of $\max\big(- (\tilde{\boldsymbol{J}}\boldsymbol{Q}^{-1}\boldsymbol{b} + \tilde{\boldsymbol{\phi}}), 0\big)$ and $\boldsymbol{K}$ is the stiffness of such spring force effect.

In dynamic settings, only spring-like contact forces may lead to the oscillation of the contact dynamics (5). Thus, we add an additional dampening term $-\boldsymbol{D}\tilde{\boldsymbol{J}}\boldsymbol{Q}^{-1}\boldsymbol{b}/h$. Here, $\tilde{\boldsymbol{J}}\boldsymbol{Q}^{-1}\boldsymbol{b}/h$ is the velocity of compressing the dual cone, and $\boldsymbol{D}$ is the damping coefficient matrix, which is another hyper parameter. Therefore, the contact parameters becomes

$$\boldsymbol{\beta}^{\text{dyn}} = \max(-h\boldsymbol{K}(\tilde{\boldsymbol{J}}\boldsymbol{Q}^{-1}\boldsymbol{b} + \tilde{\boldsymbol{\phi}}) - \boldsymbol{D}\tilde{\boldsymbol{J}}(\boldsymbol{Q}^{-1}\boldsymbol{b}/h), 0), \tag{8}$$

The final dynamic complementarity-free contact model becomes

$$\boldsymbol{v}^+ = \frac{1}{h}\boldsymbol{Q}^{-1}\boldsymbol{b} + \frac{1}{h^2}\boldsymbol{Q}^{-1}\tilde{\boldsymbol{J}}^{\mathsf{T}}\boldsymbol{\beta}^{\text{dyn}}. \tag{9}$$

For differentiablity, we replace the $\max$ in (8) with $\mathrm{softmax}$. The next system pose will be an backward Euler integration of the above resolved next velocity, i.e., $\boldsymbol{q}_{t+1} = \boldsymbol{q}_t \oplus h\boldsymbol{v}_t^+$.

B. Complementarity-Free Model Predictive Control

We define the full system state as $\boldsymbol{x} = [\boldsymbol{q}, \boldsymbol{v}]$. With access to a differentiable dynamics model and predefined cost functions $c(\boldsymbol{x}, \boldsymbol{u})$ for each step and $V_T(\boldsymbol{x})$ at the terminal of horizon, we formulate a receding-horizon optimization problem solved at each control step:

$$\begin{aligned} \min_{\boldsymbol{u}_{0:H-1}} \quad & \sum\nolimits_{t=0}^{H-1} c(\boldsymbol{x}_t, \boldsymbol{u}_t) + V_T(\boldsymbol{x}_H) \\ \text{s.t.} \quad & \boldsymbol{q}_{t+1} = \boldsymbol{q}_t \oplus h\boldsymbol{v}_t^+ \end{aligned} \tag{10}$$

where $\oplus$ denotes velocity integration. The post-impact velocities $\boldsymbol{v}_t^+$ are computed using the formulation in (2). The contact information (contact location, Jacobian, contact distance) is supposed to be re-calculated via collision detection pipeline at every time step. However, since the collision detection itself is not differentiable, we instead fix the contact Jacobian $\tilde{\boldsymbol{J}}$ and penetration vector $\tilde{\boldsymbol{\phi}}$ to their values at the initial state $\boldsymbol{x}_0$ throughout each MPC horizon.

III. Simulated Experiment

We use MuJoCo [16] as the simulation environment for evaluating the proposed method. It serves two purposes: first, as a simulation environment; second, as a real-time component for collision detection. We focus on dynamic on-palm object reorientation tasks [12]. As shown in Fig. 1, the manipulation system includes a 7-DoF Franka Research 3 robot arm with a rigidly attached flat tray; a cube, duck and teapot are selected from contactDB dataset [32] as test objects of various geometry.

In our dynamic complementarity-free contact model, we set the parameters $\boldsymbol{K}_v = 0.5\mathbf{I}$, $\boldsymbol{K} = 0.55\mathbf{I}$, $\boldsymbol{D} = 0.15\mathbf{I}$, $\mu = 0.6$. The cube and the teapot have mass 0.1, and the duck has mass 0.2. The inertia matrix $\boldsymbol{M}_o$ is estimated by Mujoco for each object. The MPC horizon H is set to be 6. The step cost and the terminal cost of the MPC are set:

$$\begin{aligned} c(\boldsymbol{x}, \boldsymbol{u}) = {} & w_1 c_{\text{pos}}(\mathbf{p}_o) + w_2 c_{\text{quat}}(\mathbf{q}_o) + w_3 c_{\text{joint}}(\boldsymbol{q}_r) \\ & + w_4 \|\mathbf{v}_o\|^2 + w_5 \|\mathbf{w}_o\|^2 + w_6 \|\boldsymbol{u}\|^2 \end{aligned} \tag{11}$$

$$\begin{aligned} V_T(\boldsymbol{x}) = {} & 25 * \big(w_1 c_{\text{pos}}(\mathbf{p}_o) + w_2 c_{\text{quat}}(\mathbf{q}_o) \\ & + w_4 \|\mathbf{v}_o\|^2 + w_5 \|\mathbf{w}_o\|^2 + w_6 \|\boldsymbol{u}\|^2 \big) \end{aligned} \tag{12}$$

where $c_{\text{pos}}(\mathbf{p}_o)$ is the distance of object position $\mathbf{p}_o$ to a reference position $\mathbf{p}_{\text{ref}}$. Similarily, $c_{\text{quat}}(\mathbf{q}_o)$ is the quaternion distance between the object and reference quaternion $\mathbf{q}_{\text{ref}}$, and $c_{\text{joint}}(\boldsymbol{q}_r)$ is the distance between the robot joint and reference robot joint $\boldsymbol{q}_{\text{ref}}$. The term $c_{\text{joint}}(\boldsymbol{q}_r)$ is designed to stabilize the robot around its initial configuration and to penalize large joint movements. In all tasks, $\mathbf{p}_{\text{ref}} = [0.1, 0.55, 0.623]^\mathsf{T}$, and $\boldsymbol{q}_{\text{ref}} = [0.11, 1.37, 1.69, -2.02, 1.82, 1.57, 0.24]^\mathsf{T}$ is the robot starting position. $\mathbf{q}_{\text{ref}}$ depends on the varying target quaternions of each tasks. With a weight $\boldsymbol{w} = [w_1, w_2, w_3, w_4, w_5, w_6]^\mathsf{T}$.

Two tasks are evaluated in the simulated environment to demonstrate the effectiveness of our method:

A. Dynamic Reorientation via Controlled Sliding

We evaluate our method on three different objects: a cube, a rubber duck, and a teapot. For each object, three target orientations are specified by applying $\pm 90°$ and $180°$ rotations about the z-axis in robot base frame as shown in the left panel of Fig. 1, forming the desired reference quaternion $\mathbf{q}_{\text{ref}}$. A small random noise is added to the initial position and orientation of the object. For each target orientation, we conduct 20 independent trials, each lasting 500 MPC control steps. A trial is considered successful if the object maintains the desired pose—defined by $c_{\text{pos}}(\mathbf{p}_o) \leq 0.1$ and $c_{\text{quat}}(\mathbf{q}_o) \leq 0.05$ —for at least 15 consecutive steps. Success rates are computed for each object, aggregated across all target orientations.

The success rates for reorienting the three objects are: 98.3% for cube, 90.0% for duck and 78.3% for the teapot. Both the cube and the duck achieve high success rates, while the teapot exhibits a lower success rate. This is primarily due to inconsistent multiple contact detections between the teapot and the tray, which impacts control stability.

B. Dynamic Reorientation via Flipping

In the flipping task, the goal is to reorient a cube placed on the robot's palm to one of six distinct target orientations. These targets are generated by applying ±90° and 180° rotations around the x and y axes of the robot base frame, see Fig. 1. Each of these target orientations is associated with a distinct set of hand-tuned cost weights to account for the differing contact modes involved. As in the re-orientation task, a trial is marked as successful if the object satisfies the same condition in controlled sliding tasks for at least 15 consecutive steps. For successful trials, we also report the position difference $\|\mathbf{p}_o - \mathbf{p}_{\text{ref}}\|$ and quaternion distance $1 - (\mathbf{q}_o^\mathsf{T}\mathbf{q}_{\text{ref}})^2$ over those steps to quantify precision.

The success rate, along with the mean position and quaternion errors for each target rotation, are reported in the Table I. The success rate varies due to the differing levels of difficulty among target poses when starting from the same robot configuration. Despite the inherent difficulty of the flipping task—which involves highly dynamic contact interactions—our proposed control framework achieves a strong success rate while maintaining good control precision.

TABLE I: Results of flipping tasks

Rotation	**Success Rate**	$\|\mathbf{p}_o - \mathbf{p}_{\text{ref}}\|$	$1 - (\mathbf{q}_o^\mathsf{T}\mathbf{q}_{\text{ref}})^2$
$x, +90°$	90%	0.029 ± 0.010	0.018 ± 0.013
$x, -90°$	65%	0.033 ± 0.011	0.016 ± 0.013
$x, 180°$	75%	0.062 ± 0.019	0.016 ± 0.013
$y, +90°$	85%	0.073 ± 0.018	0.019 ± 0.014
$y, -90°$	75%	0.061 ± 0.017	0.020 ± 0.015
$y, 180°$	70%	0.078 ± 0.018	0.019 ± 0.014

IV. Real-World Experiment

We evaluate our dynamic complementarity-free contact-implicit MPC in hardware experiments on on-palm object reorientation tasks. The setup features a Franka Emika Research 3 robot arm with a flat tray rigidly mounted to its end effector. Due to the limited torque capacity of the robot, the hardware experiments focus specifically on sliding-based reorientation. The objective is to control the robot to execute precise sliding motions that move the object from an arbitrary, perturbed pose to a target pose centered on the tray. Two 3D-printed objects are used in the experiments: a cube and a duck. The workspace of the arm is equipped with a vision-based pose estimation system to perform real-time perception. A calibrated Intel D435i RGB-D camera is positioned to provide external tracking of object pose (40Hz) during execution, using our

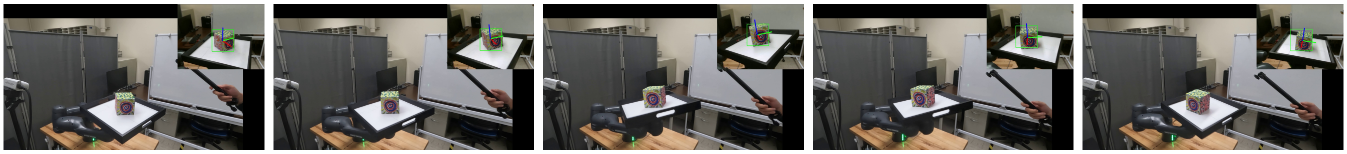


Fig. 2: Screenshots of real-world on-palm reorientation via sliding. The object tracking result is shown in the top-right corner of each frame. Following a human perturbation, the robot rotates the tray and exploits torsional friction to re-orient the object toward the target pose.

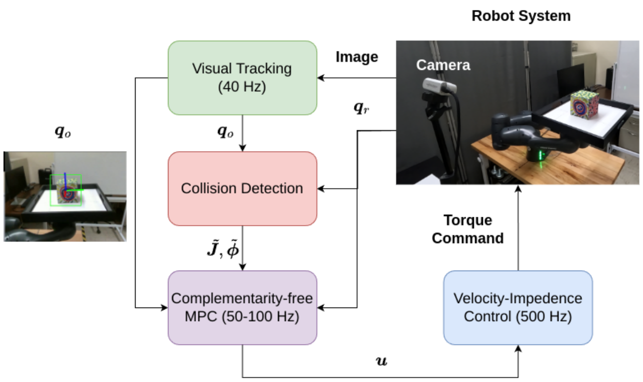


Fig. 3: Overview of the real-world on-palm manipulation system.

proposed tracking framework (which will be published soon), while a nearby Dell workstation with Intel i9-13900K CPU handles control computation, communication and data logging. Fig. 3 shows an overview of the real-world control system. During execution, the estimated object pose and the robot state are continuously provided to the complementarity-free MPC controller. The system state are also streamed into a collision detection components (realized by MuJoCo) to compute contact Jacobians and distance.

The MPC problem (10) is re-optimized at each time step with CasADi library [33]. The physical parameters of the dynamics model and the cost function weights used in the controller are detailed in Table II, where $\mathbf{I}$ stands for the identity matrix.

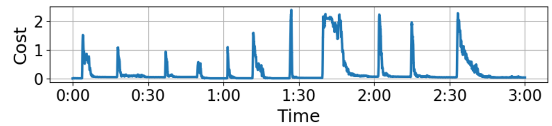
TABLE II: Control Parameters in RealWorld Reorientation Tasks

Parameter	Value (Cube)	Value (Duck)
h	0.015 s	0.02s
$\boldsymbol{M}_o$	$\text{diag}(0.1, 0.1, 0.1,$ $2\text{e}{-4}, 2\text{e}{-4}, 2\text{e}{-4})$	$\text{diag}(0.2, 0.2, 0.2,$ $4\text{e}{-4}, 4\text{e}{-4}, 3\text{e}{-4})$
$\boldsymbol{K}_v$	$0.5\mathbf{I}$	$0.5\mathbf{I}$
$\boldsymbol{K}$	$0.5\mathbf{I}$	$0.45\mathbf{I}$
$\boldsymbol{D}$	$0.1\mathbf{I}$	$0.1\mathbf{I}$
μ	0.7	0.45
H	6 steps	6 steps
$\boldsymbol{w}$	$[6.0, 2.0, 0.8,$ $0.0, 1\text{e}-5, 9\text{e}-3]^{\mathsf{T}}$	$[12.0, 4.0, 6.0,$ $1\text{e}-4, 3\text{e}-3, 1.2\text{e}-2]^{\mathsf{T}}$
$\mathbf{p}_{\text{ref}}$	$[0.073, 0.480, 0.586]^{\mathsf{T}}$	
$\boldsymbol{q}_{\text{ref}}$	$[-0.07, 1.37, 1.72, -2.31, 1.81, 1.55, 0.07]^{\mathsf{T}}$	
$\mathbf{q}_{\text{ref}}$	$[1, 0, 0, 0]^{\mathsf{T}}$	

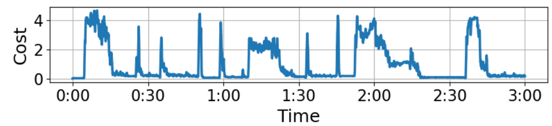
As shown in Fig. 3, we fix the target pose of the object across all trials, which is at the center of the tray and with a specific surface facing the camera. We let a human to perturb the object into different initial poses (Fig. 2). The objective is that robot needs to reorient the objects back into the target pose through controlled sliding.

A. Result

We evaluate the performance of our proposed method using to-goal cost as the primary metrics. The to-goal cost quantifies



(a) Cube reorientation



(b) Duck reorientation

Fig. 4: Cost trajectories for the real-world on-palm reorientation tasks

how closely the object pose at every point on the trajectory matches the desired target pose, based on the same step cost function used by the MPC controller.

Experiments are conducted for both objects—the 3D-printed cube and the rubber duck, with the setting described in the previous section. Total running time is 3 minutes for both tasks. The graphs of the to-goal cost of both experiment are shown in Fig. 4a and Fig. 4b. In both plots, each spike of the cost means a human perturbs the object into a random pose. For each perturbed pose, the robot performs dynamic controlled sliding to move the object back to the targe pose (see video link for video demo). Therefore, we see the rapid drop of the cost-to-go values. Throughout the experiments (around 10 total human perturbation), the on-palm manipulation system consistently demonstrates the robustness and reactivity of the control strategy. A sequence of real-world images from a successful reorientation trial is shown in Fig. 2. The MPC solving time per step is 14.9 ± 5ms for the cube reorientation and 16.1 ± 6ms for the duck reorientation, showing great potential in real-time application. Qualitatively, the algorithm maintains object orientation more accurately than position. This is primarily due to two factors: camera calibration errors on translation and the choice of MPC weighting parameters. Failures are mainly caused from the tracking side. When object tracking is stable, re-orientation is consistently successful.

V. Conclusion

We present a real-time contact-implicit MPC framework for dynamic, contact-rich manipulation using a complementarity-free contact model. Our approach enables efficient optimization of contact interactions and is validated through both simulated and real-world on-palm reorientation tasks. Results demonstrate its effectiveness and highlight the potential of complementarity-free models for dynamic dexterous manipulation.

Future work includes merging multisensory data such as tactile sensor for more precise contact perception, transferring to environments with more contact modes, as well as extending our method to the torque control for better agility.

References

[1] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023.

[2] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning*, pages 1722–1732. PMLR, 2023.

[3] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023.

[4] Meiying Qin, Jake Brawer, and Brian Scassellati. Robot tool use: A survey. *Frontiers in Robotics and AI*, 9:1009488, 2023.

[5] Haonan Duan, Peng Wang, Yayu Huang, Guangyun Xu, Wei Wei, and Xiaofei Shen. Robotics dexterous grasping: The methods based on point cloud and deep learning. *Frontiers in Neurorobotics*, 15:658280, 2021.

[6] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023.

[7] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

[8] Tao Pang, HJ Terry Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on robotics*, 39(6):4691–4711, 2023.

[9] Simon Le Cleac'h, Taylor A Howell, Shuo Yang, Chi-Yen Lee, John Zhang, Arun Bishop, Mac Schwager, and Zachary Manchester. Fast contact-implicit model predictive control. *IEEE Transactions on Robotics*, 40:1617–1629, 2024.

[10] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.

[11] Alp Aydinoglu, Adam Wei, Wei-Cheng Huang, and Michael Posa. Consensus complementarity control for multi-contact mpc. *IEEE Transactions on Robotics*, 2024.

[12] William Yang and Michael Posa. Dynamic on-palm manipulation via controlled sliding. *arXiv preprint arXiv:2405.08731*, 2024.

[13] Wanxin Jin. Complementarity-free multi-contact modeling and optimization for dexterous manipulation. *Robotics: Science and Systems*, 2025.

[14] David Stewart and Jeffrey C Trinkle. An implicit time-stepping scheme for rigid body dynamics with coulomb friction. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 162–169. IEEE, 2000.

[15] David E Stewart. Rigid-body dynamics with friction and impact. *SIAM review*, 42(1):3–39, 2000.

[16] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[17] David E Stewart and Jeffrey C Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[18] Mihai Anitescu and Alessandro Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications*, 47:207–235, 2010.

[19] Alessandro Tasora and Mihai Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):439–453, 2011.

[20] Tao Pang and Russ Tedrake. A convex quasistatic time-stepping scheme for rigid multibody systems with contact and friction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6614–6620. IEEE, 2021.

[21] Emanuel Todorov. Implicit nonlinear complementarity: A new approach to contact dynamics. In *2010 IEEE international conference on robotics and automation*, pages 2322–2329. IEEE, 2010.

[22] Wanxin Jin, Shaoshuai Mou, and George J Pappas. Safe pontryagin differentiable programming. *Advances in Neural Information Processing Systems*, 34:16034–16050, 2021.

[23] Xueyi Liu, Kangbo Lyu, Jieqiong Zhang, Tao Du, and Li Yi. Parameterized quasi-physical simulators for dexterous manipulations transfer. In *European Conference on Computer Vision*, pages 164–182. Springer, 2024.

[24] Katsu Yamane and Yoshihiko Nakamura. Stable penalty-based model of frictional contacts. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1904–1909. IEEE, 2006.

[25] Jean-Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021.

[26] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar, and Nicolas Mansard. Crocoddyl: An efficient and versatile framework for multi-contact optimal control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2536–2542. IEEE, 2020.

[27] Alexander W Winkler, C Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018.

[28] Bernardo Aceituno-Cabezas and Alberto Rodriguez. A global quasi-dynamic model for contact-trajectory optimization in manipulation. 2020.

[29] Wei-Cheng Huang, Alp Aydinoglu, Wanxin Jin, and Michael Posa. Adaptive contact-implicit model predictive control with online residual learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5822–5828. IEEE, 2024.

[30] Taylor Howell, Nimrod Gileadi, Saran Tunyasuvunakool, Kevin Zakka, Tom Erez, and Yuval Tassa. Predictive sampling: Real-time behaviour synthesis with mujoco. *arXiv preprint arXiv:2212.00541*, 2022.

[31] Vince Kurtz Albert H. Li, Preston Culbertson and Aaron D. Ames. Drop: Dexterous reorientation via online planning. *arXiv preprint arXiv:2409.14562*, 2024.

[32] Samarth Brahmbhatt, Cusuh Ham, Charles C Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8709–8719, 2019.

[33] Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11:1–36, 2019.