Mobile Pedipulation for Object Sliding via a Wheeled Bipedal Robot

Yue Qin[1] and Yanran Ding[1]

***Abstract*— This letter presents a framework that enables a wheeled bipedal robot to perform the object sliding task using its wheels, with a focus on explicitly reasoning over stick-slip transitions. The proposed approach couples phase-based offline trajectory optimization (TO) with an online model predictive controller (MPC), where the three rigid bodies with an object (TRBO) model is employed to jointly incorporate robot and object dynamics. Since stick-slip transition exhibits hybrid dynamics, the controller is designed to switch between dedicated sticking and sliding modes in an event-based manner. Preliminary MuJoCo simulations with the Tron1 robot demonstrate reliable retrieval of a thin plate over a $0.4\,\mathbf{m}$ displacement, validating the feasibility of the proposed pipeline. The framework enlarges the interaction capability of wheeled bipedal robots and offers an effective solution for the thin-plate object manipulation task which is difficult to grasp or push.**

I. Introduction

Loco-manipulation [1]–[4] has become a focal research topic because it promises robots the versatility to move and interact within complex environments. However, most existing frameworks assume that the robot can establish either a firm grasp or a stable pushing contact. Consequently, tasks involving *non-prehensile, non-pushable* objects, such as welcome mats and cardboard sheets that lie flush with the ground or objects stuck in wall corners, remain largely unsolved.

Humans often move light and thin objects using their foot when objects are on the ground, with a short acceleration of the lower limb generating horizontal force to overcome static friction and initiate motion. Figure 1 shows an example of a human sliding a welcome mat with the foot. Replicating this action on legged robots offers a possible solution for the object sliding task. Wheeled bipedal robots present a practical approach as their wheel-type end effectors [5] can exert considerable horizontal force while also enhancing the system's mobility, thereby enabling long-distance object transportation via sliding. That said, performing the object sliding task via legs remains challenging as it requires simultaneous balance control, precise modulation of reaction forces, and an explicit handling of complex stick-slip transitions

Planar sliding is the research area most closely aligned with our objectives, and researchers have explored this motion using robotic manipulators [6], [7] . The majority of these works, however, rely on quasi-static assumptions or model the robot–object interaction as an external wrench, thereby neglecting the robot's own dynamics—an approximation that is unsuitable for our scenario. Whole-body manipulation (WBM) [8], [9] does incorporate robot dynamics to enhance manipulation capabilities in pushing and grasping tasks, yet effective control under such coupled conditions remains an open problem. In our scenario, as the legs are used to perform the manipulation task, this motion can be classified as pedipulation [10], a special type of WBM, which has previously been demonstrated only on quadrupedal platforms with superior balancing capabilities. To the best of our knowledge, dynamic pedipulation on a bipedal or humanoid platform has not been demonstrated yet.

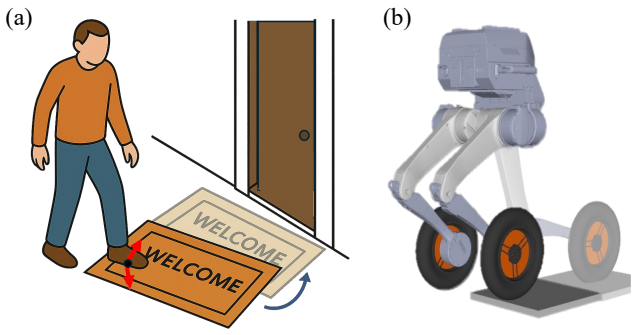

Fig. 1. (a) Human repositioning a welcome mat via sliding with one foot. (b) A wheeled bipedal robot retrieving a wooden board via sliding with one wheel keeping contact with the object.

In this letter, we propose an offline trajectory optimization and online model predictive control framework to enable a wheeled bipedal robot to perform the object sliding task through dynamic mobile pedipulation. The framework shows good performance by explicitly handling stick–slip transitions and incorporating both robot and object dynamics. The main contributions of this work are:

- Proposal and demonstration of a complex dynamic pedipulation task that effectively addresses the challenge of *non-prehensile*, *non-pushable* object sliding manipulation.
- Preliminary simulation results verifying the successful execution of an object retrieval task.

II. Offline Trajectory Generation

We employ trajectory optimization to generate dynamically feasible state and input trajectories that guide the robot to accomplish the object sliding task. These trajectories serve as references for online tracking via MPC. Owing to the complexity involved in fully three-dimensional interactions,

[1]Yue Qin and Yanran Ding are with the Department of Robotics, University of Michigan, Ann Arbor, MI - 48109, USA. {yueqin, yanrand}@umich.edu

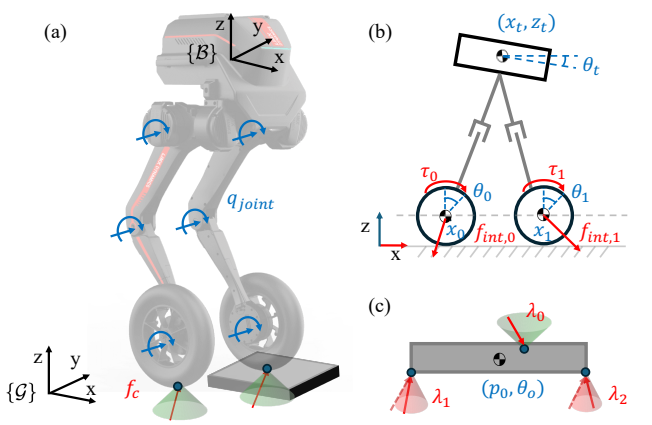

Fig. 2. (a) Configuration for a wheeled bipedal robot within the sagittal plane; (b) Reduced-order model for a wheeled bipedal robot; (c) Object model as a single rigid body with three contact forces, where $\boldsymbol{\lambda_0}$ represents the force from the robot and $\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}$ are the ground reaction forces, which are restricted to lie on the edge of the friction cone when sliding.

this work first addresses the motion planning and control problem within the sagittal plane. In this section, we detail the robot-object dynamics, sliding mode schedule, and trajectory optimization formulation.

A. Dynamics

In this letter, we focus on the planning and control of the object sliding manipulation task for a wheeled bipedal robot, which can be modeled as a multi-link rigid-body system with a floating base. We denote the robot configuration by $\boldsymbol{q} = [\boldsymbol{q}_{\text{joint}}^\top, \boldsymbol{q}_{\text{float}}^\top]^\top$, which includes both the joint angles and floating base coordinates. The standard equations of motion are given by

$$H(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}}) = S_a^\top \boldsymbol{\tau} + J_c^\top(\boldsymbol{q}) \boldsymbol{f}_c, \tag{1}$$

where $H(\boldsymbol{q})\ddot{\boldsymbol{q}}$ is the mass matrix; $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ accounts for the centripetal, Coriolis and gravitational terms; $\boldsymbol{f}_c$ is the reaction force from the contact point, and $J_c(\boldsymbol{q})$ is the corresponding contact Jacobian matrix. The matrix S_a is the selection matrix for the actuated joint torque vector $\boldsymbol{\tau}$.

The manipulated object is modeled as a single rigid body. Assuming planar motion, we consider the contact between the object and ground as two point contacts. Here we use a friction cone-based approach to analyze the interaction force between the object, the robot, and the ground. The dynamic equation for the object is

$$\dot{\boldsymbol{x}}_o = \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \boldsymbol{p}_o \\ \theta_o \\ \dot{\boldsymbol{p}}_o \\ \dot{\theta}_o \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_o \\ \dot{\theta}_o \\ \sum_i \boldsymbol{\lambda}_i / m_o + \boldsymbol{a}_g \\ \sum_i (\boldsymbol{r}_i \wedge \boldsymbol{\lambda}_i) / I_o \end{bmatrix}, \tag{2}$$

where $\boldsymbol{p}_o$ and θ_o are the object center of mass (CoM) location and pitch angle, respectively; m_o and I_o are the object mass and rotational moment of inertia; $\boldsymbol{\lambda}_i$ is the contact force at the i^{th} contact point. Here $i \in \{0, 1, 2\}$ represents one contact point with the robot and two contact points with the ground. $\boldsymbol{a}_g = [0, -g]^\top$ is the gravitational acceleration vector. Notably, $\boldsymbol{\lambda}_0$ is the reaction force from the robot contact on the leg at the manipulation side, which is denoted as $\boldsymbol{f}_{c,\text{man}}$. The robot and object models are shown in Figure 2.

B. Sliding Mode Schedule

In our targeted task, the robot is controlled to exert a force on the object via the wheel contact to overcome static friction and make it slide. Consequently, the interaction dynamics between the object and the ground are hybrid as it changes between the sticking and sliding states. Here, we use a phase-based method to formulate it as a time-switched hybrid system:

$$\begin{cases} |\lambda_{i,x}| \le \mu_o \lambda_{i,z}, & v_i^t = 0, \quad t \notin \mathcal{S}, \\ \lambda_{i,x} = \operatorname{sign}(v_i^t)\, \mu_o\, \lambda_{i,z}, & v_i^t \neq 0, \quad t \in \mathcal{S}, \end{cases} \tag{3}$$

where $i \in \{1, 2\}$ denotes two contact points between the object and ground; μ_o is the friction coefficient between the object and ground; v_i^t is the tangential velocity at the i^{th} contact point; $\mathcal{S}$ means the slide mode of the system. In our scenario, phase sequence is pre-determined as stick-slide-stick.

C. Constraints

While controlled slippage between the object and the ground is desirable, we would like to prevent slippage between the wheel and the object by imposing the friction cone constraints as follows.

$$\begin{gathered} \boldsymbol{f}_c \in \mathcal{C} \\ \mathcal{C} = \{\boldsymbol{f}_c \mid 0 \le f_{c,z} \le f_{c,\max},\ -\mu f_{c,z} \le f_{c,x} \le \mu f_{c,z}\}. \end{gathered} \tag{4}$$

Here, μ is the friction coefficient between the robot wheels and the environment, including both the object and the ground; $f_{c,\max}$ is the maximum vertical component of the reaction force. In addition, we enforce kinematic feasibility and torque limits as box constraints.

$$\boldsymbol{q} \in [\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}], \boldsymbol{\tau} \in [-\boldsymbol{\tau}_{\max}, \boldsymbol{\tau}_{\max}], \tag{5}$$

Specifically, the manipulation side wheel should always maintain contact with the object, and we enforce this condition by constraining the contact point position computed by forward kinematics (FK):

$$\begin{gathered} x_{c,\text{man}} = \text{FK}(q), \\ |x_{c,\text{man}} - x_o| <= l/2, \end{gathered} \tag{6}$$

where l is the length of the object; $x_{c,\text{man}}, x_o$ are the horizontal positions of the manipulation side wheel contact point and the object CoM, respectively.

D. TO Formulation

Trajectory optimization is formulated via the direct collocation method, which is transcribed as a nonlinear programming (NLP) problem.

$$\min_{\boldsymbol{x}[\cdot],\boldsymbol{u}[\cdot]} \quad \ell^f(\boldsymbol{x}[N]) + \sum_{n=0}^{N-1} \ell(\boldsymbol{x}[n], \boldsymbol{u}[n])$$

s.t. robot dynamics: (1)
object dynamics: (2)
time switched hybrid system dynamics: (3)
friction cone constraints: (4)
joint angle and torque limit: (5)
contact point position constraint: (6)
$\boldsymbol{x}[0] = \boldsymbol{x}_{\text{initial}}$,

where N is the planned horizon; $\boldsymbol{x} = [\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top, \boldsymbol{x_o}^\top]^\top, u = [\boldsymbol{\tau}^\top, \boldsymbol{f}_c^\top, \lambda_i^\top]^\top$ represent our state and input variables in the TO. $\ell(\boldsymbol{x}[n], \boldsymbol{u}[n])$ is the cost function defined as the weighted sum of state tracking errors and joint torques. For the terminal cost, we approximate the system's value function:

$$\ell^f(\boldsymbol{x}[N]) = x[N]^\top S\, x[N] \tag{7}$$

where S is the solution of the associated algebraic Riccati equation, which is obtained using the `dlqr` function in MATLAB. This weighted quadratic cost function aims to estimate the total cost over the next infinitely long period, with a simplified linear model and all additional constraints relaxed. This design is similar to Model Hierarchy Predictive Control [11] which can encode more information of the system's future behavior with low computational cost to improve the motion planning performance. With this formulation, the planner outputs the reference trajectory of robot-object system states and inputs, which serve as the desired trajectory in the following controller.

III. Model Predictive Control

After generating the reference trajectory, we design a unified MPC controller that utilizes state feedback from both the robot and the object to synthesize motor commands online. In this section, we detail the reduced-order model employed by the MPC, as well as the controller framework.

A. Reduced Order Model

We use a planar three rigid bodies [12] with an object (TRBO) model as our reduced-order model. In the TRBO, each wheel is connected to the torso using a massless and extendable rod due to the negligible dynamic effect from the legs, and the object is modeled as a single rigid body with friction cones, the same as in the TO. We denote $\boldsymbol{q} = [x, z, \theta]^\top$ as the state for the torso and wheels and take $\boldsymbol{\mathcal{F}} = [f_{\text{int},x}, f_{\text{int},z}, \tau_{\text{int}}]^\top$ to represent internal planar wrench that the torso exerts on the wheel CoM. $\boldsymbol{f}_c, \boldsymbol{\lambda}$ mean the contact forces on the robot and the object. The dynamic equations of the robot-object system could be formulated as:

$$\begin{aligned} &H_t\ddot{\boldsymbol{q}}_t + \boldsymbol{g} = -\sum_j B_j \boldsymbol{\mathcal{F}_j} \\ &H_w\ddot{\boldsymbol{q}}_j + \boldsymbol{g} = \boldsymbol{\mathcal{F}_j} + J_w^\top \boldsymbol{f}_{c,j} \\ &\text{object dynamic: (2)} \end{aligned} \tag{8}$$

where

$$B_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ r_{z,j} & -r_{x,j} & 1 \end{bmatrix} \quad J_w = \begin{bmatrix} 1 & 0 & -r_w \\ 0 & 1 & 0 \end{bmatrix}$$

Here B_j is the input matrix for internal reaction wrench from the j^{th} wheel to torso, and J_w is the Jacobian matrix of the wheel contact point in the wheel frame, respectively. $r_{z,j}$ and r_{x_j} are the distances from the torso to the j^{th} wheel where $j \in \{0, 1\}$ represents the left and right legs of the robot. $H_t = \text{diag}(m_t, m_t, I_t)$ is the mass matrix for the torso. The same structure applies to the wheel. $\boldsymbol{g} = [0, -g, 0]^T$ accounts for gravity.

As for the contact constraint in the vertical direction, we assume that the wheeled bipedal robot maintains continuous contact with the environment through its wheels and that the object also keeps contact with the ground. The contact constraints for the system are given by

$$z_{\text{sup}} \equiv r_w, \;\; z_{\text{man}} \equiv r_w + w_o, \; z_o \equiv w_o$$

where z_{sup}, z_{man}, z_o are the CoM heights of our supporting wheel, manipulating wheel, and the object which are all considered to remain constant in our problem. To decrease the model dimension, we integrate the contact constraint equations directly into the model dynamics [13] and select a minimal coordinate set, $\boldsymbol{q}_{rom} = [\boldsymbol{q}_t^\top, x_0, \theta_0, x_1, \theta_1, \boldsymbol{p}_0^\top, \theta_0]^\top$, as the model state, which accounts for potential slip. Also, the input is $\boldsymbol{u}_{rom} = [\boldsymbol{\mathcal{F}}_{\boldsymbol{0}}^\top, \boldsymbol{\mathcal{F}}_{\boldsymbol{1}}^\top, \boldsymbol{f}_c^\top, \boldsymbol{\lambda}^\top]^\top$. This TRBO model captures the essential properties of the system and maintains good computational efficiency, which is beneficial in our controller design.

B. Unified MPC Formulation

Due to the hybrid nature of the sliding task, we propose two separate MPC controllers that separately handle the sticking and sliding phases, with transitions governed by an event-based switching mechanism. Here, we take the retrieving motion as an example for analysis while pushing motion is similar. During the sliding phase, the MPC formulation is given by

$$\min_{\boldsymbol{x}_k,\boldsymbol{u}_k} \quad \ell^f_{\text{mpc}}(\boldsymbol{x}_N) + \sum_{k=1}^{N-1} \ell_{\text{mpc}}(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{9a}$$

s.t. reduced-order model dynamics:(8) (9b)
friction cone:(4) (9c)
joint angle and input limit (9d)
$|x_{\text{man}} - x_o| < l/2$ (9e)
$\lambda_x = \mu_o \lambda_z$ (9f)
$\dot{x}_o <= -v_{\text{min}}$ (9g)
$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{initial}}$, (9h)

where v_{min} is the threshold for the sliding motion. The sticking phase MPC controller shares the same formulation except for the weight matrix and friction constraints–(9f), (9g). These two constraints are modified as follows

$$|\lambda_x| <= \mu_o \lambda_z, \quad \dot{x}_o = 0$$

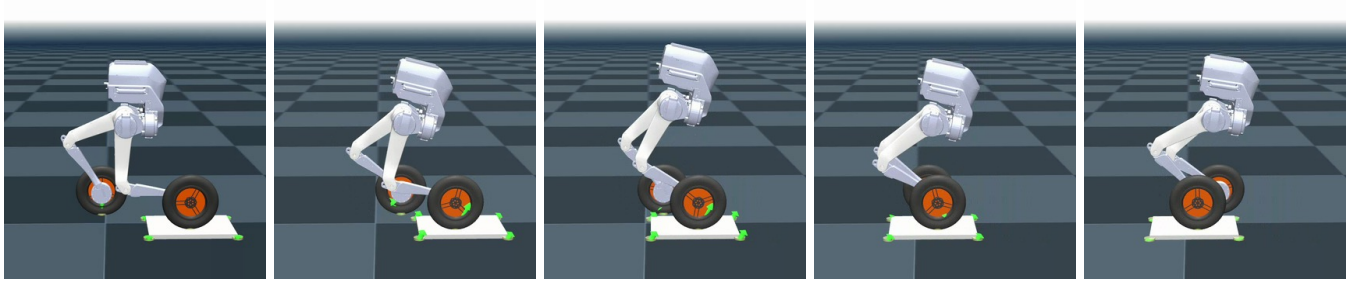

Fig. 3. Motion snapshots from one test for the retrieving task: the green arrows represent the contact forces on the object and robot.

The formulation is implemented in CasADi [14] and run on a desktop computer with a 13th Gen Intel Core i7-13700 CPU and communicates to the simulator via LCM [15]. The MPC control frequency can achieve 500 Hz thanks to the fast convergence speed of the `fatrop` solver.

IV. Experiment

The experimental platform used in this work is the Tron1 wheeled bipedal robot developed by LimX Dynamics. The robot weighs approximately 20 kg and has 14 degrees of freedom (DoF), including 8 actuated DoFs and 6 DoFs associated with the floating base. Under our assumption, the robot was restricted to the sagittal plane, which has 9 DoFs, 6 of which are actuated. This section presents experimental results obtained through simulation using the Tron1 robot in MuJoCo [16].

Figure 3 illustrates the process of the robot retrieving a thin plate. The object used in this test measures $0.3\,\mathrm{m} \times 0.3\,\mathrm{m} \times 0.02\,\mathrm{m}$ and weighs $1\,\mathrm{kg}$. The friction coefficient between the robot wheels and both the object's top surface and the ground is $\mu = 0.8$, while the friction coefficient between the object's bottom surface and the ground is $\mu_o = 0.6$. The desired transportation distance is set to $0.4\,\mathrm{m}$ and the trajectory optimization spans a period of $4\,\mathrm{s}$, with sliding motion specified to begin at $t_1 = 1\,\mathrm{s}$ and required to conclude by $t_2 = 2\,\mathrm{s}$.

As shown in Fig. 4, the robot attains the desired final sliding distance with acceptable accuracy. The intermediate object trajectory, however, deviates from the reference generated by the trajectory optimizer, indicating that the friction model employed during offline planning is insufficiently accurate. Despite this discrepancy, the MPC is able to replan control inputs in real time and still accomplish the task, underscoring the critical role of online motion synthesis in the presence of modeling errors.

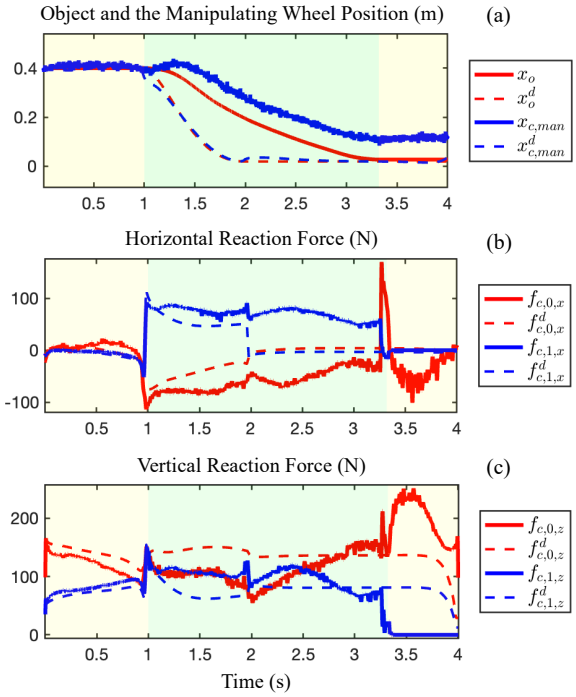

Fig. 4. Data plot from a test of the retrieving task, where green shading indicates sliding phase, while yellow shading indicates the sticking phase. (a) Object position and the manipulating wheel's CoM position; (b) Horizontal reaction force values of the two legs, in this test $j = 1$ is the manipulating side; (c) Vertical reaction force values of the two legs

V. Conclusion and Future Work

In this letter, we present a trajectory optimization and online MPC framework that enables a wheeled bipedal robot to perform object sliding tasks. Simulation results present a preliminary validation of the method, demonstrating reliable and accurate sliding manipulation behavior.

In addition to improving performance in the presented scenario, our future work will extend the current approach to three-dimensional dynamics and perform validation through hardware experiments. During the simulation, we observed limitations in the friction-cone-based model. Integrating advanced learning-based techniques to enhance the friction model and to improve the robustness of our framework constitutes an important research direction. In terms of controller architecture, the current event-based switching between two MPC controllers shows acceptable performance at this stage. However, we anticipate significant challenges when handling more complex manipulation tasks with frequent mode transitions. A promising solution would be to formulate the stick–slip transition as a linear complementarity problem within a contact-implicit MPC framework, enabling online planning of these complex transitions.

REFERENCES

[1] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, A. Kheddar, X. B. Peng, Y. Zhu, G. Shi, Q. Nguyen, G. Cheng, H. Gao, and Y. Zhao, "Humanoid Locomotion and Manipulation: Current Progress and Challenges in Control, Planning, and Learning," Jan. 2025.

[2] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Versatile multicontact planning and control for legged loco-manipulation," *Science Robotics*, vol. 8, no. 81, p. eadg5014, Aug. 2023.

[3] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A Unified MPC Framework for Whole-Body Dynamic Locomotion and Manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, Jul. 2021.

[4] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song, "UMI on Legs: Making Manipulation Policies Mobile with Manipulation-Centric Whole-body Controllers," Jul. 2024.

[5] M. Mason, D. Pai, D. Rus, L. Taylor, and M. Erdmann, "A mobile manipulator," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 3, May 1999, pp. 2322–2327 vol.3.

[6] X. Yi and N. Fazeli, "Precise Object Sliding with Top Contact via Asymmetric Dual Limit Surfaces," May 2023.

[7] W. Yang and M. Posa, "Dynamic On-Palm Manipulation via Controlled Sliding," May 2024.

[8] A. Purushottam, C. Xu, Y. Jung, and J. Ramos, "Dynamic Mobile Manipulation via Whole-Body Bilateral Teleoperation of a Wheeled Humanoid," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1214–1221, Feb. 2024.

[9] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, "Learning Whole-Body Manipulation for Quadrupedal Robot," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 699–706, Jan. 2024.

[10] P. Arm, M. Mittal, H. Kolvenbach, and M. Hutter, "Pedipulate: Enabling Manipulation Skills using a Quadruped Robot's Leg," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5717–5723.

[11] H. Li, R. J. Frei, and P. M. Wensing, "Model hierarchy predictive control of robotic systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3373–3380, 2021.

[12] J. Yu, Z. Zhu, J. Lu, S. Yin, and Y. Zhang, "Modeling and mpc-based pose tracking for wheeled bipedal robot," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 7881–7888, 2023.

[13] V. Klemm, Y. de Viragh, D. Rohr, R. Siegwart, and M. Tognon, "Nonsmooth Trajectory Optimization for Wheeled Balancing Robots With Contact Switches and Impacts," *IEEE Transactions on Robotics*, vol. 41, pp. 497–517, 2025.

[14] J. A. 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, pp. 1–36, 2019.

[15] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4057–4062.

[16] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 5026–5033.