2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Contact Optimization for Non-Prehensile Loco-Manipulation via Hierarchical Model Predictive Control

Alberto Rigo, Yiyu Chen, Satyandra K. Gupta, and Quan Nguyen

***Abstract*— Recent studies on quadruped robots have focused on either locomotion or mobile manipulation using a robotic arm. However, legged robots can manipulate large objects using non-prehensile manipulation primitives, such as planar pushing, to drive the object to the desired location. This paper presents a novel hierarchical model predictive control (MPC) for contact optimization of the manipulation task. Using two cascading MPCs, we split the loco-manipulation problem into two parts: the first to optimize both contact force and contact location between the robot and the object, and the second to regulate the desired interaction force through the robot locomotion. Our method is successfully validated in both simulation and hardware experiments. While the baseline locomotion MPC fails to follow the desired trajectory of the object, our proposed approach can effectively control both object's position and orientation with minimal tracking error. This capability also allows us to perform obstacle avoidance for both the robot and the object during the loco-manipulation task.**

Fig. 1: Motion snapshots of Unitree A1 robot manipulating a 5 kg object to follow a circular trajectory. Supplemental video: `https://youtu.be/amzgE4-Yrig`

I. INTRODUCTION

Legged robots have great potential to interact with the environment and have demonstrated significant performance for locomotion, such as high-speed running and robust walking on challenging terrains [1], [2], [3], [4], [5], [6], [7], [8]. Nevertheless, with the existing control and planning algorithms, most applications for quadruped robots focus on navigation and inspection, which always try to avoid objects/obstacles even if they are movable [9], [10], [11]. In this paper, we are interested in leveraging legged robots' agile mobility and their capability to interact with the environment to manipulate a heavy object during locomotion.

In mobile manipulation, robots can exhibit different modes of interaction with the object. For example, mobile robots equipped with a robotic arm [12], [13], [14], [15], [16], [17] can enable basic manipulation tasks such as door opening, pick-and-place, and load carrying. However, such setups are limited to small and light payloads due to the force limit of the robot arm and gripper's size. For legged robots, manipulation with their feet is also an intriguing idea as quadrupedal animals can use their legs or limbs for manipulation[18], [19]. However, this setup is highly challenging for loco-manipulation tasks with quadruped robots since it requires them to move and manipulate the object simultaneously, drastically decreasing their ability to balance. Therefore, this paper tackles the problem of loco-manipulation for quadruped robots using a planar pushing motion. For large and heavy objects, non-prehensile manipulation, such as pushing, offers tremendous advantages. When the object is too large or heavy to grasp, pushing using the whole body becomes one of the few options to manipulate it effectively. In addition, this method also allows quadruped robots to manipulate objects without adding a heavy robotic arm. Pushing is a widely used motion primitive and has been thoroughly studied by the manipulation community. The mechanics of planar pushing is well-studied in [20], [21], [22]. Motion planning algorithms are introduced in [23], [24] to find open-loop trajectories to drive the object to the target pose. These approaches rely on the assumption that the manipulator always sticks with the object for the entirety of the push. To handle the complexity associated with frictional contact interactions, motion planning algorithms developed by the robotic manipulation community manage to handle different mode sequences [25], [26], [27]. Nevertheless, these approaches are computationally heavy due to the nonlinear and non-convex optimization programs. A recent work, in [28], proposes a real-time controller to reason across different contact modes, including sticking and sliding, using an online approximation for the offline mix-integer program.

The recent developments on model predictive control (MPC) for legged robot locomotion in [1], [16], [29] suggest that optimal control action can be computed online given a proper contact schedule. However, these works mainly focus on locomotion. To simultaneously achieve locomotion and manipulation tasks, we propose a novel hierarchical MPC framework including (1) high-level manipulation MPC to optimize for both contact force and contact location of

This work is supported in part by National Science Foundation Grant IIS-2133091. The opinions expressed are those of the authors and do not necessarily reflect the opinions of the sponsors.

[1]Alberto Rigo, Yiyu Chen, Satyandra K. Gupta, and Quan Nguyen, are with the Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA, 90089 `rigo@usc.edu`, `yiyuc@usc.edu`, `quann@usc.edu`, `guptask@usc.edu`

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160507

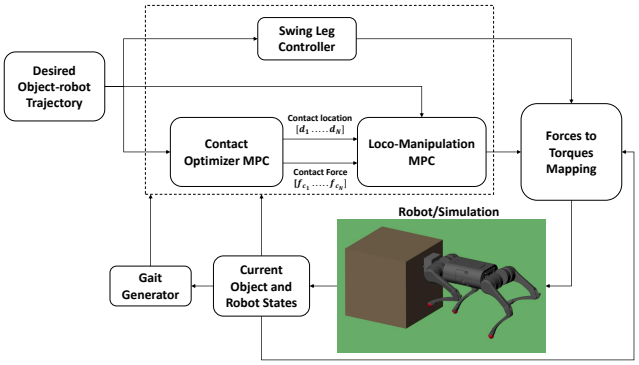

Fig. 2: Control architecture of the proposed framework for non-prehensile loco-manipulation

the manipulation task; and (2) low-level loco-manipulation MPC to regulate the interaction force between the robot and the object while maintaining the desired locomotion performance. Both MPC problems are solved effectively in real-time and allow us to consider the problem's physical constraints, such as limits on the interaction force and contact location of the robot head on the object surface. Numerical and experimental validation has shown that our approach outperforms locomotion MPC or heuristic approach for loco-manipulation. Thanks to the capability of optimizing contact location and interaction force simultaneously, our approach can allow legged robots to manipulate heavy objects effectively with highly accurate position and orientation tracking. Thus, this enables the execution of collision-free trajectory for both the robot and the object.

The rest of the paper is organized as follows. Section II introduces the pushing configuration for the robot and object system and presents the proposed control architecture and the two MPC in detail. Then, Section III shows simulation and hardware experiments results. Finally, Section IV draws conclusion remarks.

II. PROPOSED FRAMEWORK

A. System Overview

This paper investigates the pushing task of an arbitrary object along a planned trajectory in the x and y world frame positions and the yaw angle ψ. We assume that we have knowledge of all the geometric and inertial properties of the pushed object and that we can obtain feedback on its center of mass position and yaw angle. Our configuration involves the robot pushing on a predetermined object surface while being able to adjust the contact location. This enables the robot to push the object forward while changing its orientation to follow the desired trajectory. To address the complexity of the loco-manipulation problem, we propose a hierarchical control structure as illustrated in Fig 2. First, the contact optimizer MPC is used to compute the required control input, i.e., contact force and contact point on the object surface, to drive the manipulated object to the desired states. Then, the loco-manipulation MPC is used to regulate the desired interaction force and contact location to track the planned trajectory for the object-robot system based on the output of the contact optimizer MPC. The two MPCs use the same prediction horizon, so the predicted values for the contact interaction by the contact optimizer MPC are used as inputs for the loco-manipulation MPC. The integration between the two MPCs gives us a unified control framework to solve the loco-manipulation problem, allowing us to drive the manipulated object to the desired location while maintaining robot balance. The other component of the high-level control, the swing leg controller, tracks the position of the swing legs based on a heuristic policy. Moreover, based on a predefined contact schedule, the gait generator assigns a boolean variable representing either stance or swing phase to each leg. More details about the last two components can be found in [1]. Having presented the overview of our approach, we now introduce the contact optimizer and loco-manipulation MPC in detail.

B. Contact Optimizer MPC

This first controller uses a simplified model of manipulated object dynamics. In this paper, we are interested in controlling the object's position and yaw angle. Therefore we can use the following simplified 2D rigid body dynamics equations under the effect of friction forces:

$$m\ddot{\mathbf{p}}_{obj} = \mathbf{f}_{\mu} + \mathbf{f}_{c} \tag{1}$$

$$I_z\dot{\omega}_z = \mathbf{f}_{c} \times \mathbf{d} \tag{2}$$

where $\mathbf{p}_{obj}$ represents the position of the object in the world frame, $\mathbf{f}_{\mu}$ is the frictional force between the object and ground, $\mathbf{f}_{c}$ is the contact force applied to the object by the robot in world frame, I_z and ω_z are the moment of inertia and angular velocity of the object in the vertical direction with respect to its center of mass, and $\mathbf{d}$ is the vector between the contact point and object center of mass in world frame. If we consider the contact force and the contact point as control variables for the problem, the previous set of equations is nonlinear. To linearize these equations and use them as model dynamics in a linear MPC, we can make certain assumptions. The first one consists of a small enough MPC frequency update so that we can consider that the contact force will only vary by a small amount between iterations. As a result, we can use the known value of force computed at the previous controller update, f_{c_0}, as the contact force in equation (2). Moreover, we can simplify the definition of the contact point $\mathbf{d}$ by assuming that the contact force $\mathbf{f}_{c}$ always acts in the direction perpendicular to the object's surface. Under this assumption, the contact point $\mathbf{d}$ is defined as the distance from the object's center of mass to the surface's normal direction, as illustrated in Fig. 3. With these assumptions, equations (2) become linear and can be used to represent the object dynamics in the state space form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{3}$$

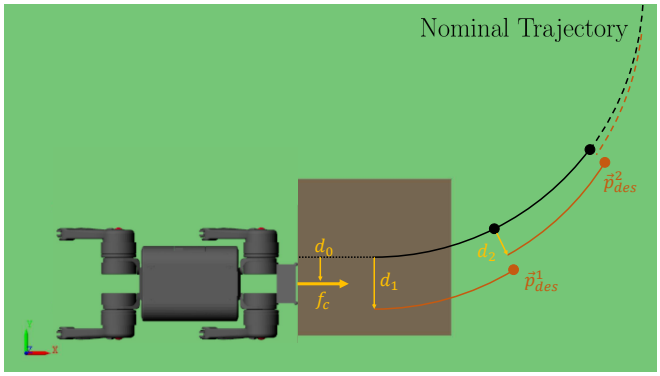

Fig. 3: Object-robot system overview and representation of the offset trajectory derived from the contact point optimization. d_0 is the contact point location at the current time instant, while $d_1, d_2, ...$ are the contact location to offset the trajectory from the contact optimizer MPC.

where $\mathbf{x} = \begin{bmatrix}\psi & x & y & \omega_z & \dot{x} & \dot{y} & g\end{bmatrix}$, $\mathbf{u} = \begin{bmatrix}f_c & d\end{bmatrix}$, and the matrices are

$$A = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \begin{bmatrix}0\\-\mu\\-\mu\end{bmatrix} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0}_{3\times2} \\ \begin{bmatrix} f_{c_0} & 0 \\ 0 & \cos\psi \\ 0 & \sin\psi \end{bmatrix} \\ \mathbf{0}_{1\times2} \end{bmatrix}. \tag{4}$$

Here we assumed that the frictional force $\mathbf{f}_\mu$ is expressed as $-\mu mg$ for both x and y directions, the body frame contact force f_c can be expressed in world frame using a rotation matrix $\mathbf{R}_\psi \in \mathbb{R}^2$, and f_{c_0} is the contact force computed at the previous controller update.

After discretizing the dynamics model equations, we use a classic quadratic programming (QP) formulation to solve the MPC problem with N horizons. The cost function

$$\min_{\mathbf{x},\mathbf{u}} \sum_{i=1}^{N} (\mathbf{x}_{i+1} - \mathbf{x}_{i+1_{ref}})^T Q (\mathbf{x}_{i+1} - \mathbf{x}_{i+1_{ref}}) + \mathbf{u}_i^T R \mathbf{u}_i, \tag{5}$$

is minimized to reduce the difference between the object's current and reference states and the control effort, with diagonal weight matrices Q and R for the states and inputs. The MPC controller determines the optimal contact force f_c and contact point d by considering inequality constraints on d and f_c, represented as:

$$d_{min} < d < d_{max} \tag{6}$$

$$0 < f_c < F_{max} \tag{7}$$

The first constraint ensures that the contact point remains within the limits of the object's dimension, while the second constraint ensures that contact is maintained and the pushing action is stable, without requiring an instantaneous force, F_{max}, that is too large to affect the locomotion's stability. The optimized result of this MPC has two impacts on the loco-manipulation MPC, which are discussed in the following section.

C. Loco-manipulation MPC

To effectively regulate the desired contact force f_c and contact location d derived from the contact optimizer MPC in Section II-B, we present a unified loco-manipulation MPC that takes into account these two variables in the control design. In comparison with the locomotion MPC [1], the following are the main developments of our framework for loco-manipulation.

- Our loco-manipulation MPC considers the interaction force f_c between the robot and the object in the robot dynamics. Therefore, it can regulate the desired manipulation force while maintaining desired performance for locomotion.
- The reference trajectory of the robot locomotion is also automatically updated based on the desired contact location d as well as real-time feedback of the object state.

The modified single rigid body dynamics (SRBD) equations used for MPC are presented as follows:

$$m\ddot{\mathbf{p}} = \sum_{i=1}^{4} \mathbf{f}_i - \mathbf{f}_g - \mathbf{f}_c, \tag{8}$$

$$\frac{d}{dt}\mathbf{I}\omega = \sum_{i=1}^{4} (\mathbf{r}_i - \mathbf{p}) \times \mathbf{f}_i. \tag{9}$$

In these equations, $\mathbf{p}$ and $\mathbf{r}_i$ represent the body position and foot position of the robot in the world frame, respectively. $\mathbf{I}$ and ω are the rotational inertia matrix of the robot body and its angular velocity expressed in the body frame. Additionally, $\mathbf{f}_i$, $\mathbf{f}_g$, and $\mathbf{f}_c$ denote the vectors for reaction forces, gravitational forces, and contact force with the object, expressed in the world frame. We assume that the contact force $\mathbf{f_c}$ always acts in the longitudinal direction of the robot body frame. Since we approximate the contact location as the center of the robot head, the contact force passes through the robot's center of mass and does not affect the rotational dynamics of the robot. This assumption is reasonable with the geometric configuration of the robot used both in simulation and experiments. Furthermore, we use the SRBD equations with the additional assumption that the contribution of the leg joints to the robot dynamics is negligible.

These equations are discretized and used in an MPC formulation with N horizons and a prediction horizon representing an entire gait cycle. The MPC is formulated as a quadratic program (QP) that can efficiently be solved in real-time. The cost function for the MPC problem is similar to (5). However, in this case, x represents the states of the robot body, and u represents the reaction forces on the four feet.

The second deviation from a conventional MPC for legged robots is related to the definition of the reference states. In our approach, we start with the nominal reference trajectory for the object-robot system and modify it by incorporating the contact point d computed by the contact optimizer MPC. The changes to the reference trajectory occur only in the $x-y$ plane trajectory, as depicted in Fig 3. We use the same prediction horizon for both MPCs and adjust the trajectory by the optimal distance d_i for each horizon. Additionally, we

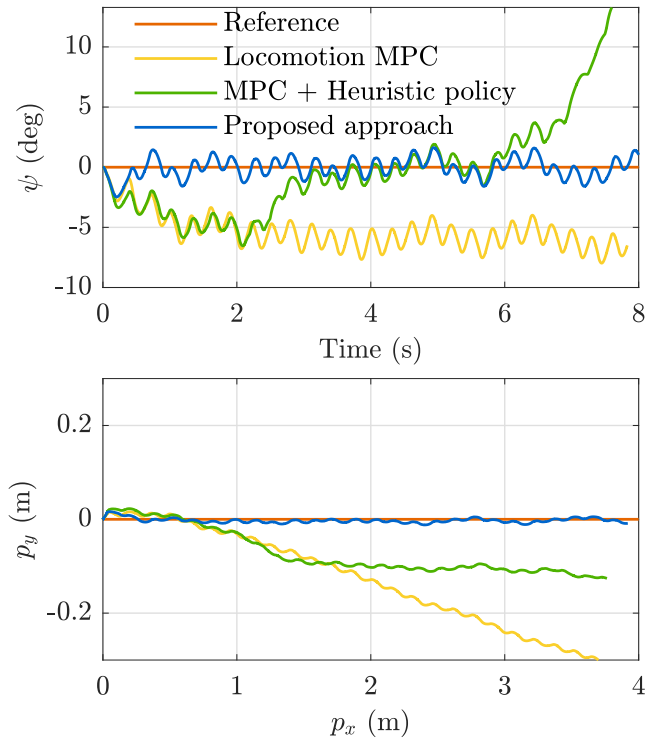

Fig. 4: Comparison of box's $x-y$ plane position and yaw angle following a straight line trajectory for the three controllers. Results obtained in simulation pushing a box of mass $m = 5\ kg$ and coefficient of friction $\mu = 0.5$.

set the desired yaw angle of the robot to be equal to the yaw angle of the box. As the position gains in the cost function's $\vec{Q}$ matrix plays a crucial role in tracking the contact point's location, we set them to a relatively high value.

III. RESULTS

This section presents the validation and results of simulation and hardware experiments using a Unitree A1 robot. For simplification purposes, we show results using a box of mass m and known dimensions, but as we showed in the previous section, the proposed framework is generalizable to an arbitrary object.

A. Simulation

The Matlab Simscape Multibody package is utilized in the simulation, which enables the representation of the interactions among the robot, object, and ground plane. Both MPCs have a prediction horizon of 30 ms and a sampling frequency of 3 ms, resulting in ten horizons consistent with the baseline locomotion controllers. The loco-manipulation MPC takes the predicted optimal contact force and contact point distance for all ten horizons as inputs. The mass of the box is 5 kg, and the coefficient of friction between the box and the ground is 0.5.

First, we present the comparison in performing two tasks between 3 types of controllers:

- Baseline locomotion MPC with a fixed contact location.
- Locomotion MPC + a heuristic policy to adjust the contact location

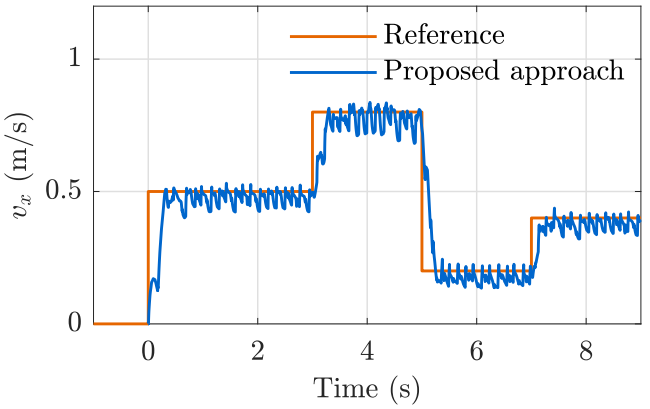

Fig. 5: Tracking of the box velocity with step changes every 2 seconds to the desired velocity. Results obtained in simulation pushing a box of mass $m = 5\ kg$ and coefficient of friction $\mu = 0.5$.

- Our proposed controller using hierarchical MPC to optimize for both contact force and contact location.

To better emphasize the advantage of our proposed approach with contact optimization, we also investigate a heuristic policy to adjust the contact location to allow the robot to control the yaw motion of the object. The heuristic policy commands a positive or negative lateral velocity, in the robot frame, based on the yaw angle direction, to properly change the contact point and adjust the box direction of motion, as follows:

$$v_y = v_y^{des}\text{sign}(\psi_{box} - \psi_{target}) \tag{10}$$

where ψ_{box} is the yaw angle of the box, ψ_{target} is the yaw angle from the current box position to the target, and v_y^{des} is a constant desired lateral velocity in the robot's body frame. With this policy, the robot tries to align the box orientation to face the target. Still, unlike our proposed approach, the robot could fail the task by losing contact with the box since there is no constraint to keep the contact within limits.

The first task is reaching the desired target location following a straight line. The results are presented in Fig. 4. Here, we can see that the baseline locomotion MPC fails, while the other two approaches can reach the target. On the contrary, only the proposed approach can keep the robot in a straight line toward the target, thanks to the optimized contact point location updating in real-time based on the dynamics of the box. Moreover, in Fig. 5, we can see that the proposed controller can effectively track sudden changes in the desired velocity for the object-robot system, thanks to the optimal values of contact force computed in the contact optimizer MPC.

The next task consists in reaching the desired target while following a curved trajectory, in this case, one-quarter of a circle. The baseline locomotion MPC immediately fails, losing contact with the box since there is no policy to adjust the contact point location. With this task, we can highlight the difference between a controller that considers the manipulated object dynamics and adjusts its action in real time and a simple policy for changing the pushing contact

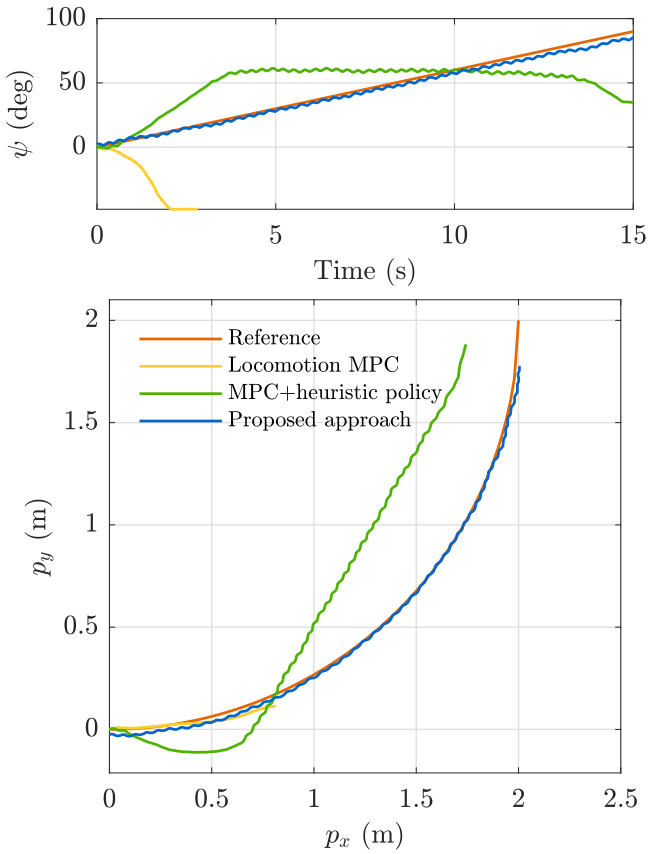

Fig. 6: Comparison of box's $x-y$ plane position and yaw angle following a quarter circle trajectory for the three controllers. Results obtained in simulation pushing a box of mass $m = 5\ kg$ and coefficient of friction $\mu = 0.5$.

point. Fig. 6 shows the results for this task, and we can see that, while both controllers can accomplish the task, only our proposed approach can successfully follow the desired trajectory. This occurs because the heuristic controller is limited to tracking only the box's yaw angle while the robot pushes it forward to reach the target. Instead, our proposed approach shows that it can dynamically change contact points to adjust the box position and orientation, always maintaining the contact location within limits.

One real-world scenario where our proposed controller can be applied is pushing an object through a path while avoiding environmental obstacles. The obstacle avoidance policy is not the focus of this paper, and we can assume the desired trajectory is already collision-free. With the following plots, we want to show the capability of following sharp changes in the direction of motion, typically occurring in obstacle avoidance. Fig 7 shows our controller's ability to change the contact point to react quickly to sharp changes in desired yaw angle, unlike the other two controllers that fail by colliding with the obstacles. One limitation of our approach is that the desired forward velocity must be tuned down during the turns to facilitate the motion. While during straight pushing, we can track velocity larger than 0.5 m/s, during a sharp turn, we have to limit it to 0.1 m/s.

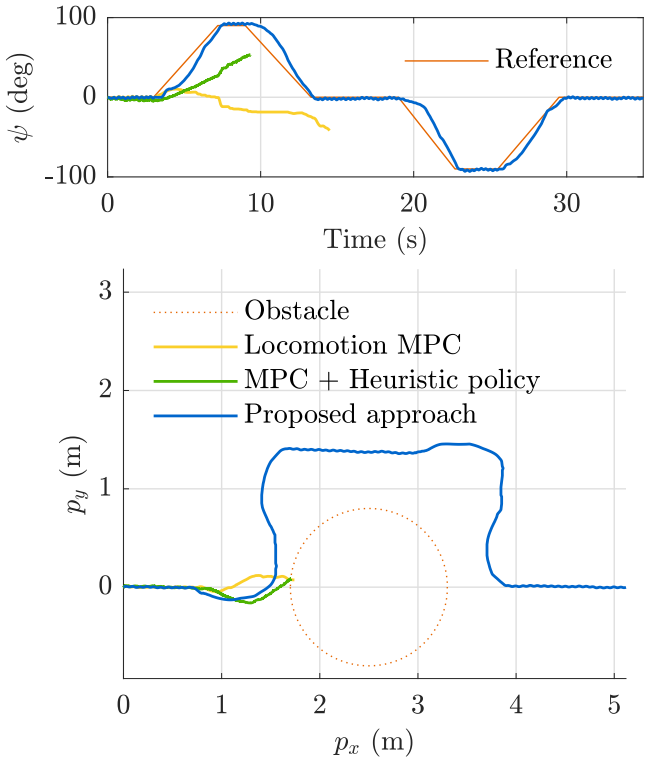

Fig. 7: Following a collision-free trajectory in an obstacle avoidance scenario. Box's $x-y$ plane position and yaw angle are shown.

B. Hardware Experiments

We validate our proposed approach for hardware experiments using a Unitree A1 robot with a low-friction head to avoid sticking between the head and the box surface. The box mass we use in experiments is 5 kg, and we assume a coefficient of friction of 0.2 between the box surface and the ground. The width of the surface where the robot is pushing is 0.25 m, which translated into a constraint on the contact location of the robot head, such that its position is within the range of -0.08 m and 0.08 m from the center of the box. The yaw angle and position feedback for both robot and box are obtained using an Optitrack motion capture system (MoCap). Due to the high precision required to follow the optimal contact point on the box, we could not rely consistently on the robot's internal state estimation, which demonstrated drift in position estimation. The MoCap system comprises 6 Optitrack Prime^X 13W, for improved tracking of ground objects, with a tracking frequency of 100Hz. The trackable surface area by the MoCap for the hardware experiments is 2 meters by 2 meters.

The first task is to follow a straight line and maintain a constant yaw angle, similar to the simulation results presented in Fig. 4. In the experimental results presented in Fig. 8, the controller effectively corrects the box's direction to keep it on the desired trajectory, even in the presence of model uncertainties such as values of the friction coefficient and box inertia. From (c), we can see that the optimized contact location remains within the boundaries imposed by

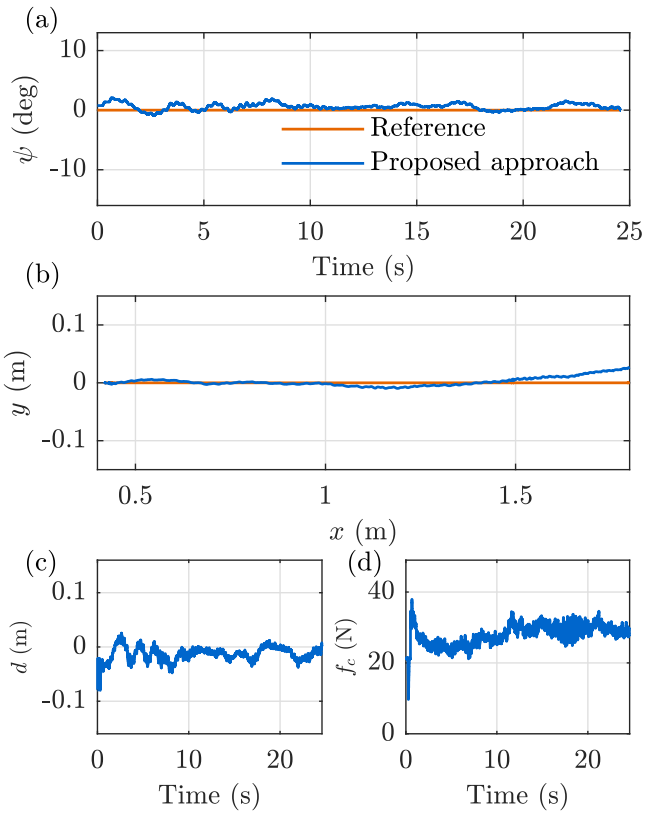

Fig. 8: Straight line experiment results. a) Pushed box yaw angle; b) Pushed box trajectory; c) Optimized contact location. Origin at the center of the box's surface; d) Optimized contact force.

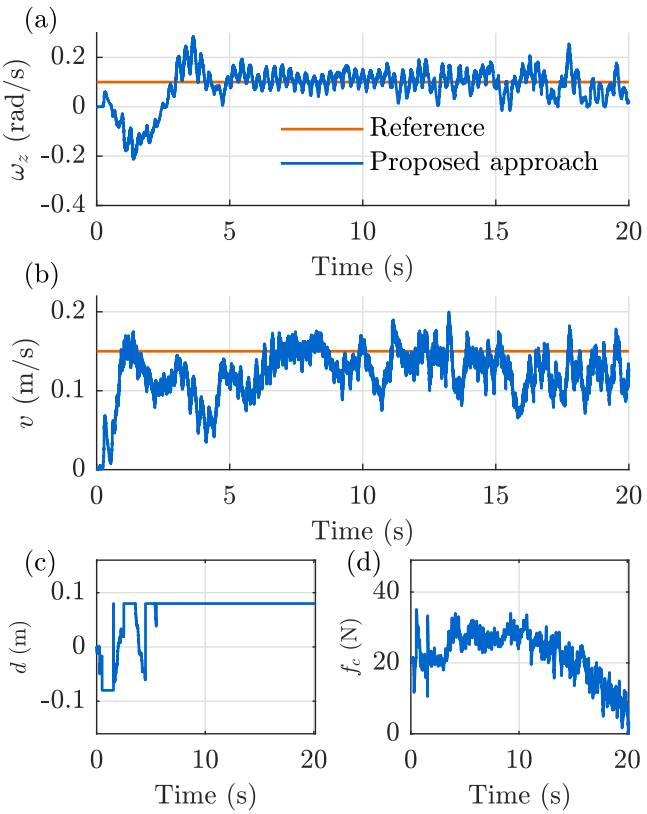

Fig. 9: Constant yaw rate experiment results. a) Pushed box yaw rate; b) Pushed box velocity; c) Optimized contact location. Origin at the center of the box's surface; d) Optimized contact force.

the dimensions of the box, thanks to the constraints enforced by the contact optimizer MPC. Furthermore, the optimized contact location varies appropriately to track the desired yaw angle. The MPC formulation ensures that the optimized contact force (in (d)) is higher during the initial phase when the box needs to accelerate to the desired velocity and then settles to a steady-state value required to maintain the desired speed against the friction force acting on the box.

The second task is to maintain a constant yaw rate and velocity for the box. The results are shown in Fig. 9. We can see that the controller is able to track both the yaw rate and the velocity of the box after an initial settling period where the box needs to be accelerated. To follow the desired yaw rate, the contact optimizer MPC saturates the contact location to the maximum value imposed by the box dimensions. This means we are tracking the maximum yaw rate and that, in case we would command a higher yaw rate, the robot would still keep the contact location within limits thanks to the constraints in the MPC formulation. Considering the results from both tasks, we have shown that thanks to the hierarchical structure of our approach, we can effectively track the yaw angle, yaw rate, and velocity of the box while maintaining stable locomotion under the effect of the interaction forces between the robot and the box.

IV. CONCLUSIONS

To solve the challenging problem of body loco-manipulation, we have presented a practical approach with a hierarchical structure comprising two MPCs. The nonlinear nature of the problem is simplified through this hierarchical structure. Our approach's effectiveness is demonstrated using numerical and experimental validations. Our proposed method outperforms other controllers in simulation by improving the loco-manipulation problem due to contact point location and contact force optimization. This feature enables the robot to follow diverse trajectories with changes in velocity and sharp turns, making it applicable in many scenarios. Because of the imposed constraints on the manipulation problem, we can ensure that the robot is always in contact with the box and never exceeds the object's dimensions. In experiments, we have replicated the results obtained in simulation, showing the successful implementation of contact optimization. Furthermore, we showed our approach can effectively push a box of 5 kg, accounting for 50% of the robot mass. In the future, we plan to extend our framework to consider online obstacle avoidance, enabling the robot to navigate an environment with obstacles while pushing an object.

REFERENCES

[1] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[2] M. Sombolestan, Y. Chen, and Q. Nguyen, "Adaptive force-based control for legged robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7440–7447.

[3] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, "Perceptive locomotion in rough terrain–online foothold optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5370–5376, 2020.

[4] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[5] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[6] Q. Nguyen and K. Sreenath, "L 1 adaptive control for bipedal robots with control lyapunov function based quadratic programs," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 862–867.

[7] Q. Nguyen, X. Da, J. Grizzle, and K. Sreenath, "Dynamic walking on stepping stones with gait library and control barrier functions," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 384–399.

[8] Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath, "Dynamic bipedal locomotion over stochastic discrete terrain," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1537–1553, 2018.

[9] M. X. Grey, A. D. Ames, and C. K. Liu, "Footstep and motion planning in semi-unstructured environments using randomized possibility graphs," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4747–4753.

[10] O. Cebe, C. Tiseo, G. Xin, H.-c. Lin, J. Smith, and M. Mistry, "Online dynamic trajectory optimization and control for a quadruped robot," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 773–12 779.

[11] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 295–302.

[12] G. Xin, F. Zeng, and K. Qin, "Loco-manipulation control for arm-mounted quadruped robots: Dynamic and kinematic strategies," *Machines*, vol. 10, no. 8, p. 719, 2022.

[13] H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, and S. Vijayakumar, "Roloma: Robust loco-manipulation for quadruped robots with arms," *arXiv preprint arXiv:2203.01446*, 2022.

[14] B. U. Rehman, M. Focchi, J. Lee, H. Dallali, D. G. Caldwell, and C. Semini, "Towards a multi-legged mobile manipulator," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3618–3624.

[15] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, "Alma-articulated locomotion and manipulation for a torque-controllable robot," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8477–8483.

[16] 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, 2021.

[17] V. Morlando, M. Selvaggio, and F. Ruggiero, "Nonprehensile object transportation with a legged manipulator," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6628–6634.

[18] F. Shi, T. Homberger, J. Lee, T. Miki, M. Zhao, F. Farshidian, K. Okada, M. Inaba, and M. Hutter, "Circus anymal: A quadruped learning dexterous manipulation with its limbs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2316–2323.

[19] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. Sreenath, "Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot," *arXiv preprint arXiv:2208.01160*, 2022.

[20] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986.

[21] S. Akella and M. T. Mason, "Posing polygonal objects in the plane by pushing," *The International Journal of Robotics Research*, vol. 17, no. 1, pp. 70–88, 1998.

[22] J. Zhou, R. Paolini, J. A. Bagnell, and M. T. Mason, "A convex polynomial force-motion model for planar sliding: Identification and application," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 372–377.

[23] K. M. Lynch and M. T. Mason, "Stable pushing: Mechanics, controllability, and planning," *The international journal of robotics research*, vol. 15, no. 6, pp. 533–556, 1996.

[24] J. Zhou, J. A. Bagnell, and M. T. Mason, "A fast stochastic contact model for planar pushing and grasping: Theory and experimental validation," *arXiv preprint arXiv:1705.10664*, 2017.

[25] Y. Hou, Z. Jia, and M. T. Mason, "Fast planning for 3d any-pose-reorienting using pivoting," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1631–1638.

[26] J. Z. Woodruff and K. M. Lynch, "Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4066–4073.

[27] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," 2018.

[28] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 755–773, 2020.

[29] J. Li and Q. Nguyen, "Force-and-moment-based model predictive control for achieving highly dynamic locomotion on bipedal robots," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 1024–1030.