

# Hierarchical Optimization-based Control for Whole-body Loco-manipulation of Heavy Objects

Alberto Rigo, Muqun Hu, Satyandra K. Gupta, and Quan Nguyen

**Abstract**—In recent years, the field of legged robotics has seen growing interest in enhancing the capabilities of these robots through the integration of articulated robotic arms. However, achieving successful loco-manipulation, especially involving interaction with heavy objects, is far from straightforward, as object manipulation can introduce substantial disturbances that impact the robot’s locomotion. This paper presents a novel framework for legged loco-manipulation that considers whole-body coordination through a hierarchical optimization-based control framework. First, an online manipulation planner computes the manipulation forces and manipulated object task-based reference trajectory. Then, pose optimization aligns the robot’s trajectory with kinematic constraints. The resultant robot reference trajectory is executed via a linear MPC controller incorporating the desired manipulation forces into its prediction model. Our approach has been validated in simulation and hardware experiments, highlighting the necessity of whole-body optimization compared to the baseline locomotion MPC when interacting with heavy objects. Experimental results with Unitree Aliengo, equipped with a custom-made robotic arm, showcase its ability to lift and carry an 8kg payload and manipulate doors.

## I. INTRODUCTION

Legged robots have garnered increasing attention for their potential to perform versatile locomotion tasks across challenging terrains [1], [2], [3], [4], [5]. To enhance their practical applicability, researchers have explored two approaches to perform loco-manipulation: utilizing the robot’s body or existing limbs [6], [7], [8] and integrating articulated robotic arms into legged platforms, enabling them to execute loco-manipulation tasks [9], [10], [11], [12]. The combination of locomotion and manipulation capabilities opens up exciting opportunities for legged robots in various real-world applications, from search and rescue missions to industrial automation [13], [14], [15], [16].

However, achieving successful loco-manipulation in the context of legged robotics is a formidable challenge. Unlike traditional wheeled or tracked robots [17], [18], [19], legged robots face unique dynamics and control intricacies when engaging in object manipulation. The introduction of manipulation forces and object interactions can introduce substantial disturbances that disrupt robot stability and locomotion [20], [21].

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.

<sup>1</sup>Alberto Rigo, Muqun Hu, 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, muqunhu@alumni.usc.edu, quann@usc.edu, guptask@usc.edu



Fig. 1: Snapshots of Aliengo lifting and carrying a 5kg payload. Supplemental video: <https://youtu.be/0hYDa94F78E>

While recent years have witnessed increased attention towards learning-based methods for addressing loco-manipulation problems [22], [23], [24], classic control frameworks remain prevalent. Loco-manipulation challenges entail coordinating manipulation actions with the robot’s environment and locomotion. This typically involves two core components: a manipulation planner, often computing plans offline, and a locomotion tracking controller responsible for generating torque commands to track the high-level references established by the planner [25], [26], [27], [28]. The tracking problem is typically tackled through various optimization frameworks that accommodate crucial system constraints [29], [30]. In contrast, the planning module strives to strike a balance between physical accuracy and computational complexity, frequently relying on simplified models of the problem under consideration [31], [32]. Nevertheless, exceptions exist where the full system dynamics are employed [33], [34]. However, a common limitation prevails: most of these formulations do not allow real-time solutions, rendering them unsuitable for rapid online replanning.

Recent studies have adopted a unified non-linear loco-manipulation framework, such as [35] and [36]. These works utilize a nonlinear Model Predictive Control planner, capable of computing real-time trajectories for the Center of Mass (CoM), limbs, and forces. A whole-body controller then tracks these reference trajectories. In [36], the authors introduce an offline planner that computes a sequence of locomotion and manipulation actions. Guided by a predefined library of interactions, this process facilitates the completion of user-defined tasks incorporating the environment model. However, it is important to note that while nonlinear MPC offers the advantage of considering more detailed predictive models or constraints, it typically requires significant computational power due to the complexity of the nonlinear optimization problem.

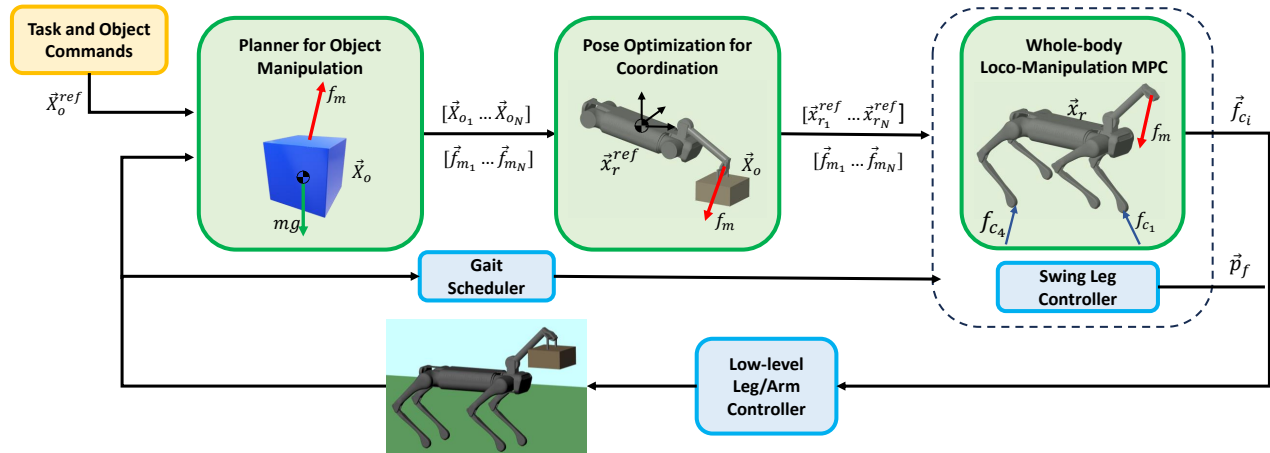


Fig. 2: **Block diagram for the proposed framework.** Highlighted in green are the novel components that, together with the swing leg controller, form the high-level controller for the quadruped

This paper presents a novel framework for legged agile loco-manipulation that leverages whole-body coordination to tackle the inherent complexities of the task. Our approach integrates elements from Model Predictive Control (MPC) and pose optimization to synthesize a control strategy that coordinates locomotion and manipulation, as illustrated in Fig. 1. We have devised a control structure capable of executing loco-manipulation tasks by decomposing the nonlinear problem into elementary components that interact hierarchically. Utilizing online pose optimization enables full coordination between manipulation and locomotion, demonstrated by the complex tasks we can perform with the custom-made 1-DOF robot arm introduced in this paper.

The remainder of the paper is organized as follows. Section II introduces the proposed control architecture explaining briefly the hierarchical interactions between the components. Each of these components is explained in detail in Sections III, IV, V. Then, Section VI shows the results of simulation and hardware experiments.

## II. CONTROL SYSTEM OVERVIEW

In this section, we present the control system architecture, illustrated in Fig. 2, that underlies our proposed whole-body coordination framework. Loco-manipulation problems are usually described with nonlinear models due to the mutual interactions between robot and object. Nevertheless, solving nonlinear optimization problems is a computational burden and requires powerful on-board capabilities for the robot. We split the loco-manipulation problems into three elementary sub-problems that work hierarchically to circumvent this issue. Our approach begins by defining a user-specified task and the corresponding commands for the manipulated object. The initial phase of our method treats the object as an isolated entity subjected to manipulation forces. Using an MPC structure with a linear prediction model for object dynamics, we calculate the optimal object states and manipulation forces, aligning them with the desired manipulation task commands. Subsequently, we introduce pose optimization, a critical step to coordinate the loco-manipulation.

Pose optimization takes as input the desired object states, which we aim to track, and the manipulation force derived from the manipulation planner. It then generates a sequence of poses that define the robot's reference trajectory. Pose optimization also offers the advantage of enabling flexible loco-manipulation, accommodating various object parameter configurations, such as dimensions and gripping points. With the robot's reference trajectory now defined through pose optimization, the whole-body loco-manipulation MPC efficiently tracks this trajectory while considering the impact of manipulation forces on the robot's dynamic stability. The sections III, IV, V provide a detailed breakdown of each component of our proposed approach.

## III. PLANNER FOR OBJECT MANIPULATION

The first component of our framework focuses on computing the necessary manipulation actions to achieve pre-defined tasks. We employ a linear Model Predictive Controller (MPC) structure that shares the same control horizon as the final robot controller, ensuring seamless coordination between manipulation and locomotion. Each manipulation task specifies the object states to be planned and the manipulation forces to be optimized. For instance, if the task involves opening a hinged door, we plan the opening angle and the force required to manipulate the handle. The chosen task also dictates the dynamics of the object under consideration. We can apply the same MPC planner to accommodate various tasks while adjusting the object states and manipulation forces accordingly.

The general dynamic formulation for the manipulated object in our linear MPC is expressed as follows:

$$\mathbf{A}_m \dot{\mathbf{X}}_o = \mathbf{f}_\mu + \mathbf{f}_m \quad (1)$$

Here,  $\mathbf{X}_o$  encompasses the object states we optimize,  $\mathbf{A}_m$  represents the diagonal matrix for the system linear dynamics,  $\mathbf{f}_\mu$  denotes external forces acting on the object (e.g., frictional forces), and  $\mathbf{f}_m$  stands for the manipulation force exerted on the object.

Different tasks involve different object dynamics and the corresponding commands. For example, lifting an object may require a command in terms of lifting velocity or lifting time. The desired command for each task produces reference values for the states of the objects, denoted as  $\bar{X}_o^{ref}$ , which are incorporated into the cost function of the linear MPC problem to minimize deviations from these references. We also include equality and inequality constraints tailored to the specific task at hand, such as guaranteeing that the manipulation force maintains its perpendicular orientation to the door surface while opening the door.

The desired manipulation force for the entire time horizon is passed to the quadruped loco-manipulation MPC and the pose optimization for whole-body coordination. In the former, the force is treated as a known quantity representing the interaction between the robot and the object over the prediction horizon. In the latter, the force is used to compute the arm joint torques needed to manipulate the object and minimize them. More details about how to use these forces are presented in the respective sections. The optimized object trajectory from the MPC planner plays a crucial role in the subsequent pose optimization phase, where it coordinates the robot's locomotion to execute the desired manipulation task effectively.

#### IV. POSE OPTIMIZATION FOR COORDINATED LOCO-MANIPULATION

In this section, we present the details of pose optimization and its role in bridging the gap between the object manipulation planner and the whole-body loco-manipulation controller. The fundamental idea is to translate the computed manipulation forces and optimal object states into robot-centric states and dynamics. To achieve this, we perform optimized pose computations for the robot at each MPC horizon, accounting for the manipulation force and system kinematic constraints. Our approach is inspired by the work in [37], with the necessary modifications to adapt it to real-time execution while maintaining meaningful constraints. The pose optimization problem is formulated as a Non-Linear Programming (NLP) problem, with the optimization variable  $\mathbf{X}$  encompassing the robot's Center of Mass (CoM) location  $\mathbf{p}_r$ , body Euler angles  $\Theta$ , arm joint angles  $\mathbf{q}_{arm}$ , and the manipulation force  $\mathbf{f}_m$  acting at the end-effector location. However, we exclude the leg joint angles and ground reaction forces from the optimization variables, which are computed by the swing leg controller and loco-manipulation MPC, allowing the solution of the NLP problem to be computed in real-time. The problem is defined as follows:

$$\min_{\mathbf{X}} Q_p(p_{r,z} - p_{r,z}^{ref})^2 + \|\Theta\|_{Q_\Theta}^2 + \|\tau\|_{Q_\tau}^2 \quad (2)$$

$$\text{s.t. } p_{z,min} \leq p_{hip_z}^i \leq p_{z,max} \quad (3)$$

$$\Theta_{min} \leq \Theta < \Theta_{max} \quad (4)$$

$$\mathbf{q}_{arm,min} \leq \mathbf{q}_{arm} \leq \mathbf{q}_{arm,max} \quad (5)$$

$$\mathbf{X}_e = \mathbf{X}_{obj} \quad (6)$$

$$\mathbf{f}_m = \mathbf{f}_m^{plan} \quad (7)$$

Starting from eq. 2, the objectives of pose optimization are minimizing the difference between robot CoM height and reference value, minimizing the body rotations, and minimizing the arm torques  $\tau$  needed for manipulation, calculated using the contact Jacobian of the arm  $\mathbf{J}_{arm}(\mathbf{X})$ . These objectives are weighted by respective scalar or diagonal matrices  $Q_p$ ,  $Q_\Theta$ ,  $Q_\tau$  of appropriate dimensions, allowing for tailored control of the robot's behavior during manipulation. While the CoM height is directly incorporated into the cost function, the  $x - y$  position is determined by the optimization process based on problem constraints. We avoid explicitly including leg joint angles in the pose, relying instead on Equation 3 to ensure that each hip location's height remains within feasible bounds for effective stepping. Equation 4 restricts robot orientation to physically feasible values, particularly concerning pitch and roll, while Equation 5 enforces constraints on the arm joint angles. The last two constraints (Equations 6 and 7) establish the crucial connections between pose optimization and the MPC manipulation planner. Equation 6 ensures that the end effector's pose  $\mathbf{X}_e = \mathbf{X}_e(\mathbf{X})$  matches the optimized object states computed by the manipulation planner. The specifics of this constraint depend on the particular task at hand; for instance, in the case of lifting an object, it constrains the end effector's position, whereas for turning a door handle, it also includes the end effector's orientation. Equation 7 requires that the manipulation force in the optimization variable aligns with the force computed by the manipulation planner  $\mathbf{f}_m^{plan}$ .

The pose optimization is executed at every MPC horizon, producing a reference trajectory for the robot's states for the whole-body loco-manipulation MPC.

#### V. WHOLE-BODY LOCO-MANIPULATION MPC

In this section, we present the formulation of our proposed Loco-Manipulation Model Predictive Controller (MPC). We have developed this formulation to address a crucial aspect of our work: the manipulation of heavy objects, which significantly affects the dynamics of the robot. While the baseline locomotion MPC for quadruped robots, as introduced in [1], is designed to handle minor external disturbances to the robot's dynamics, it falls short when dealing with heavy objects. These disturbances need explicit consideration in the robot's dynamics. Our model considers a single rigid body with contact forces applied at the feet locations and a manipulation force at the arm gripper, as illustrated in Fig. 2. We express the dynamic model in terms of the robot's position  $\mathbf{p}_r \in \mathbb{R}^3$  and angular velocity  $\boldsymbol{\omega}_r \in \mathbb{R}^3$ , both in the world frame:

$$m_r(\ddot{\mathbf{p}}_r + \mathbf{g}) = \sum_{i=1}^4 \mathbf{f}_{c_i} + \mathbf{f}_m, \quad (8)$$

$$\mathbf{I}_r \dot{\boldsymbol{\omega}}_r = \sum_{i=1}^4 \mathbf{r}_{c_i} \times \mathbf{f}_{c_i} + \mathbf{r}_m \times \mathbf{f}_m \quad (9)$$

In Equation 8,  $m_r$  represents the combined mass of the robot and arm,  $\mathbf{g} = [0 \ 0 \ -g]^T$  is the gravity vector,  $\mathbf{f}_{c_i}$  denotes the ground reaction force acting on foot  $i$ , and  $\mathbf{f}_m$  is the

external manipulation force. In Equation 9, the derivative of the angular momentum is simplified retaining only the term  $\mathbf{I}_r \dot{\boldsymbol{\omega}}_r$ , and  $\mathbf{I}_r$  is the moment of inertia of the robot in the world frame,  $\mathbf{r}_{c_i}$  is the position vector from the robot's Center of Mass (CoM) to the  $i^{\text{th}}$  foot location, and  $\mathbf{r}_m$  is the position vector from the robot's CoM to the gripper location.

The manipulation force  $\mathbf{f}_m$  is a known quantity derived from the MPC manipulation planner. It represents the force the robot requires to follow the desired commands for various tasks. Furthermore, during the discretization of the dynamics for prediction in the MPC, we have access to the desired manipulation forces for all prediction horizons. Thus, we treat it as part of the input vector in the dynamics and enforce equality constraints at each horizon to match the force to the desired manipulation force. With the prediction model defined, we represent it in a state-space form  $\dot{\mathbf{x}}_r = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ , where:

$$\mathbf{x}_r = [\boldsymbol{\Theta} \quad \mathbf{p}_r \quad \boldsymbol{\omega}_r \quad \dot{\mathbf{p}}_r \quad g] \quad (10)$$

$$\mathbf{u} = [\mathbf{f}_{c_1} \quad \mathbf{f}_{c_2} \quad \mathbf{f}_{c_3} \quad \mathbf{f}_{c_4} \quad \mathbf{f}_m] \quad (11)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{R}_z(\psi) & \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{g}{\|\mathbf{g}\|} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 \end{bmatrix} \quad (12)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_r^{-1} \mathbf{r}_{c_1} \times & \cdots & \mathbf{I}_r^{-1} \mathbf{r}_{c_4} \times & \mathbf{I}_r^{-1} \mathbf{r}_m \times \\ \frac{\mathbf{I}_3}{m_r} & \cdots & \frac{\mathbf{I}_3}{m_r} & \frac{\mathbf{I}_3}{m_r} \\ \mathbf{0}_{1 \times 3} & \cdots & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (13)$$

In equation 12, matrix  $\mathbf{R}_z(\psi)$  is the rotation matrix corresponding to the yaw angle  $\psi$ , while, in equation 13,  $\mathbf{r}_i \times$  represents the skew-symmetric transformation matrix of position vector  $\mathbf{r}_i \in \mathbb{R}^3$ . We can discretize the state-space formulation to use it as a prediction model in a linear MPC formulation, with  $N$  horizons, defined as follows:

$$\min_{\mathbf{x}_r, \mathbf{u}} \sum_{i=1}^N \|\mathbf{x}_{r_{i+1}} - \mathbf{x}_{r_{i+1}}^{\text{ref}}\|_{\mathbf{Q}_r}^2 + \|\mathbf{u}_i\|_{\mathbf{R}_r}^2 \quad (14)$$

$$\text{s.t. } \mathbf{x}_{r_{i+1}} = \mathbf{A}_d \mathbf{x}_{r_i} + \mathbf{B}_d \mathbf{u}_i \quad (15)$$

$$\underline{\mathbf{c}}_f \leq \mathbf{C} \mathbf{u}_i \leq \bar{\mathbf{c}}_f \quad (16)$$

$$\mathbf{f}_{c_i} = 0 \quad \text{if swing leg} \quad (17)$$

$$\mathbf{f}_m = \mathbf{f}_{m_{\text{des}}} \quad (18)$$

In the cost function of the problem, equation 14,  $\mathbf{x}_r^{\text{ref}}$  denotes the reference values for the robot states, which are computed from the pose optimization presented in section IV. These reference values are crucial for coordinating the entire-body motion to execute the loco-manipulation task accurately. Equation 15 represents the dynamic constraints for each prediction horizon, and  $\mathbf{A}_d$  and  $\mathbf{B}_d$  are the discrete-time equivalents of the matrices presented in equations 12 and 13. Equation 16 represents the frictional pyramid constraints for each leg, equation 17 enforces the vanishing of the

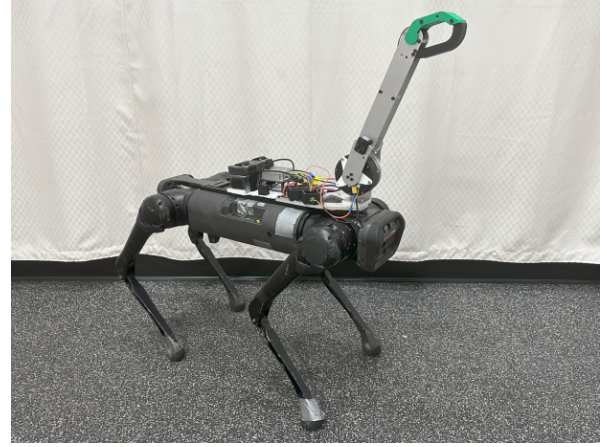


Fig. 3: Unitree Aliengo with custom-made arm used for experimental validation of the proposed approach

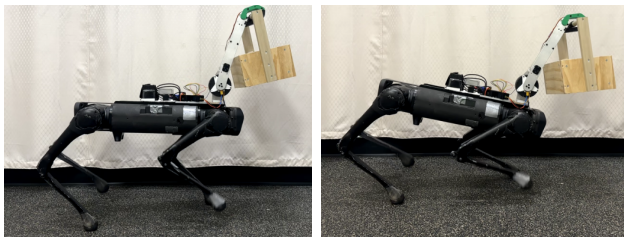
reaction forces on the swing legs, and equation 18 is the equality constraint that sets the manipulation force in the dynamics equal to the desired manipulation force from the MPC manipulation planner for each prediction horizon. The controller determines optimal ground reaction forces while accounting for the presence of the manipulation force at each prediction horizon, enabling precise tracking of desired state trajectories generated by the pose optimization.

## VI. RESULTS

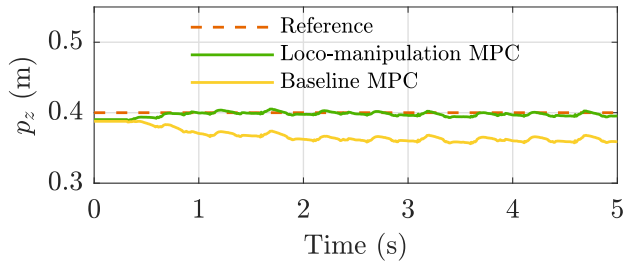
In this section, we present the results we obtained in both simulation and real hardware to prove the effectiveness of our approach. Simulations were performed using the Simscape multibody package in Matlab Simulink, a high-fidelity environment that accurately simulates contact-rich scenarios. To test and validate our framework, we conducted experiments using a Unitree Aliengo robot equipped with a custom 1-DOF robotic arm, shown in Fig.3. We designed this arm with three key considerations:

- By limiting it to 1-DOF, we reduced the arm's weight, enabling a higher maximum payload capacity.
- By installing a single powerful actuator at the base of the arm, we can perform highly dynamic tasks with heavy objects.
- With proper whole-body coordination, we can still achieve many manipulation tasks even with the reduced arm DOFs.

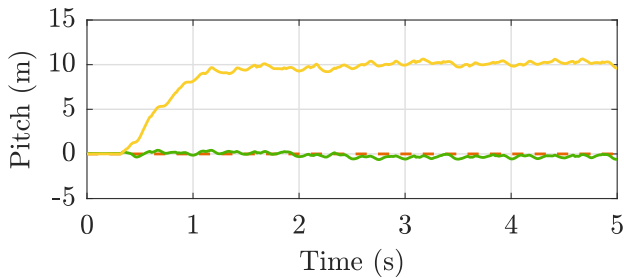
Our control architecture, detailed in Section II, efficiently operates on the robot's onboard computer. The low-level controller operates at a frequency of 1 kHz. At the same time, the object manipulation planner and loco-manipulation Model Predictive Control (MPC) run at 30 Hz with a time horizon of  $T = 0.5$  s and  $N = 10$  horizons. We utilize CasADi and the Ipopt solver to solve the pose optimization problem. Despite the hierarchical structure of the framework leading to a higher number of hyperparameters, the linear formulations facilitate straightforward tuning. Through a series of experiments, we demonstrate the impact of each component of our approach compared to baseline controllers.



(a) Snapshot of lifting 3Kg object using loco-manipulation MPC (b) Snapshot of lifting 3Kg object using baseline MPC



(c) Robot COM height

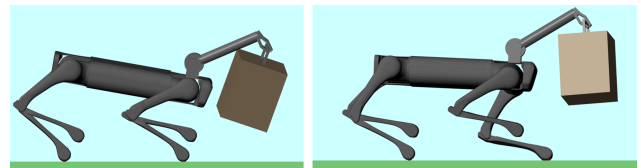


(d) Robot Pitch

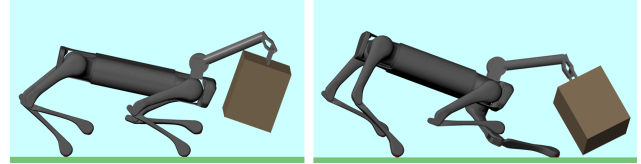
Fig. 4: **Improvements using loco-manipulation MPC.** In the plots, we compare the results using the loco-manipulation MPC and the baseline MPC. The robot is lifting from the ground a 3Kg object and reaches a predefined arm configuration,

### A. Effect of Loco-manipulation MPC

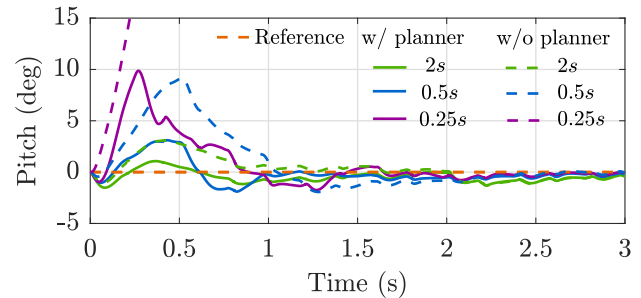
In this section, we want to highlight the importance of considering the object in the robot MPC model, especially when the object is heavy. To do this, we compare the loco-manipulation MPC in our approach to a baseline locomotion MPC, where there is no information about the object the robot is carrying. We perform this task on the real robot and during the task, the robot picks up a 3kg object from the ground using the arm DOF and lifts it to a predefined arm angle in 2 seconds. We can see snapshots of the task and COM height and robot pitch tracking with the two controllers in Fig. 4. Using the proposed loco-manipulation MPC, the robot can successfully lift the heavy object from the ground and maintain the desired height and pitch for the robot throughout the entire task. Instead, the baseline MPC cannot follow the desired quantities and struggles to maintain balance. In fact, with objects heavier than 5kg, the baseline MPC would fail, while our proposed controller can still handle them. Due to the mass-efficient arm design, we have a maximum payload capability of 8 kg, which is almost



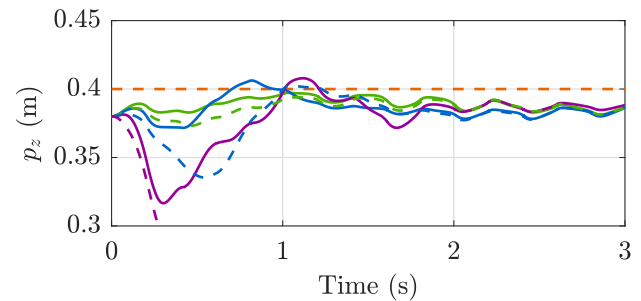
(a) Using object manipulation planner



(b) Using fixed manipulation force



(c) Pitch angle of the robot during the dynamic lifting task



(d) Height of the robot COM during the dynamic lifting task

Fig. 5: **Improvements using object manipulation planning.** In the plots, we compare the results using the loco-manipulation MPC with the object manipulation planner or a fixed manipulation force without the object manipulation planner. The robot lifts a 10 Kg object from the ground and reaches a predefined arm configuration in shorter times, making the lifting more dynamic.

50% of the robot's weight. Our payload-to-robot weight ratio is higher than other robotic arms used in state-of-the-art legged loco-manipulation research.

### B. Effect of Planner for Object Manipulation

We then investigated the effectiveness of the planner for object manipulation. This component of the proposed approach becomes critical when performing very dynamic tasks because the effect of the object on the robot cannot be represented by a fixed manipulation force anymore. To highlight the importance of the component, we performed different simulations of dynamic object-lifting tasks, where various lifting times were specified for each case. The results are presented in Fig. 5 regarding robot COM height and pitch angle, using the object planner or without the planner with

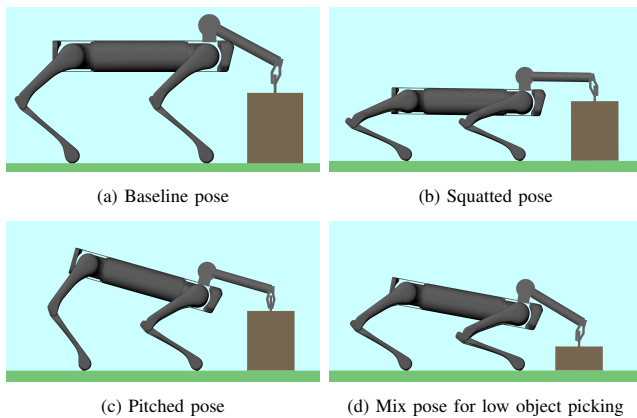


Fig. 6: **Various optimized poses for object lifting.** These figures represent the optimal poses computed for the task based on the object dimensions and the weights in the optimization cost function.

fixed manipulation force, together with the loco-manipulation MPC. We start from a 2-second lifting that can be treated as a quasi-static movement, and both simulations are successful. But if we reduce the lifting time to 0.5s, we start to see that the controller that uses the fixed manipulation force cannot maintain the robot’s reference height and pitch. When the lifting time decreases to 0.25s, this controller fails, while our proposed approach using the manipulation planner can consider the increased force acting on the robot from the large acceleration imparted on the object and still maintain balance to complete the task.

### C. Effect of Pose Optimization for Coordination

The next component we want to showcase is the pose optimization for coordinated loco-manipulation. This part of the approach is important because it links the output of the object manipulation planner to the loco-manipulation MPC. In Fig. 6, we can see various solutions that the pose optimization block can compute to solve the same task based on the choice of weights for the objectives in the cost function. We can consider the pose that tracks COM height and pitch and only uses the arm to reach the object as the baseline. Then, if we reduce the weight related to the COM height in the pose optimization cost function, we obtain the pose in Fig. 6b, where the controller trades the robot COM height to reach the object with the gripper. Similarly, in Fig. 6c, we can see the effect of reducing the weight related to robot body rotation. A balanced choice of weights in the cost function gives us a pose that uses all possible DOFs of the system to reach the object with the gripper, as illustrated in Fig. 6d. All these starting poses can complete the task of lifting the object to a predefined height.

### D. Door Opening Using Proposed Approach

To illustrate the effectiveness of our approach, we conducted tests involving the challenging task of opening a heavy, resistant hinged door. Fig. 7 presents snapshots of the task successfully solved in simulation. Within the manipulation planner, we have subtasks such as turning the

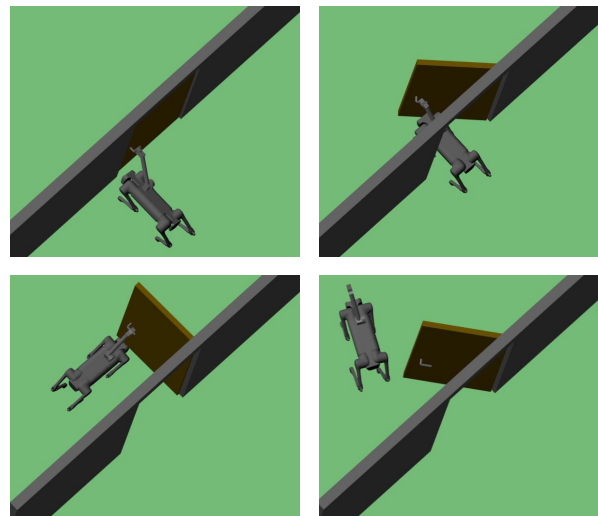


Fig. 7: **Door opening with the proposed framework.** In the initial phase, the object manipulation planner and pose optimization dictate the robot to roll to manipulate the door handle. Subsequently, the robot applies the necessary force to push the door open, with the manipulation planner considering kinematic and spatial constraints.

door handle and pushing the door open. The pose optimization orchestrates the robot’s motion to align with the desired manipulation actions. Notably, in this task, we can account for potential collisions with the door frame directly within the pose optimization, resulting in viable collision-free trajectories for the robot’s COM. One key advantage of our approach, with its distinct hierarchical components working together, is the ability to incorporate additional constraints into the problem without significantly increasing complexity and computational overhead. This is in contrast to a unified planner for both the object and robot, where the nonlinear structure would considerably complicate the process of obtaining an online planning solution.

## VII. CONCLUSIONS

This paper introduces a practical approach for addressing loco-manipulation challenges in legged robots. Our hierarchical approach simplifies the problem by breaking it into three components that work together effectively. We validate our approach with numerical and experimental scenarios investigating the effect of the three components separately. We show the importance of considering the object’s dynamic effect on the robot controller, highlight the significance of our online object manipulation planner, and demonstrate the flexibility of our pose optimization component. Our framework, when applied to a Unitree Aliengo equipped with a custom-made robotic arm, achieves successful tasks such as lifting and carrying an 8 Kg object and opening a resistive hinged door, underscoring the significance of our three components. Future directions include extending the framework to more practical tasks to explore the boundaries of the current arm design.

## REFERENCES

- [1] J. Di Carlo, P. M. Wensing, B. Katz, G. Bleidt, 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] 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.
- [3] 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.
- [4] D. Kim, J. Di Carlo, B. Katz, G. Bleidt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.
- [5] M. Sombolostan, 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.
- [6] M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli, and N. G. Tsagarakis, "Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 859–866, 2020.
- [7] A. Rigo, Y. Chen, S. K. Gupta, and Q. Nguyen, "Contact optimization for non-prehensile loco-manipulation via hierarchical model predictive control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9945–9951.
- [8] M. Sombolostan and Q. Nguyen, "Hierarchical adaptive locomotion control for quadruped robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 156–12 162.
- [9] H. Ferrolho, W. Merkt, V. Ivan, W. Wolfslag, and S. Vijayakumar, "Optimizing dynamic trajectories for robustness to disturbances using polytopic projections," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7477–7484.
- [10] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter, "Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2377–2384, 2022.
- [11] 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.
- [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] A. Gams, A. Ude, and J. Morimoto, "Accelerating synchronization of movement primitives: Dual-arm discrete-periodic motion of a humanoid robot," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2754–2760.
- [16] G. Heppner, A. Roennau, J. Oberländer, S. Klemm, and R. Dillmann, "Laupe-six legged walking robot for planetary exploration participating in the spacebot cup," *WS on Advanced Space Technologies for Robotics and Automation*, vol. 2, no. 13, pp. 69–76, 2015.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] J. Li and Q. Nguyen, "Multi-contact mpc for dynamic loco-manipulation on humanoid robots," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 1215–1220.
- [21] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986.
- [22] 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.
- [23] H. Sun, L. Yang, Y. Gu, J. Pan, F. Wan, and C. Song, "Bridging locomotion and manipulation using reconfigurable robotic limbs via reinforcement learning," *Biomimetics*, vol. 8, no. 4, p. 364, 2023.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] Q. Yao, J. Wang, S. Yang, C. Wang, L. Meng, Q. Zhang, and D. Wang, "A transferable legged mobile manipulation framework based on disturbance predictive control," in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2022, pp. 2093–2097.
- [28] M. P. Murphy, B. Stephens, Y. Abe, and A. A. Rizzi, "High degree-of-freedom dynamic manipulation," in *Unmanned Systems Technology XIV*, vol. 8387. SPIE, 2012, pp. 339–348.
- [29] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch!-dynamic grasps using boston dynamics spot with external robotic arm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4488–4494.
- [30] W. J. Wolfslag, C. McGreavy, G. Xin, C. Tiseo, S. Vijayakumar, and Z. Li, "Optimisation of body-ground contact for augmenting the whole-body loco-manipulation of quadruped robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3694–3701.
- [31] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1647–1654.
- [32] C. D. Bellicoso, K. Krämer, M. Stäubli, 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.
- [33] P. Ewen, J.-P. Sleiman, Y. Chen, W.-C. Lu, M. Hutter, and R. Vasudevan, "Generating continuous motion and force plans in real-time for legged mobile manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4933–4939.
- [34] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A collision-free mpc for whole-body dynamic locomotion and manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4686–4693.
- [35] 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.
- [36] 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, 2023.
- [37] J. Li and Q. Nguyen, "Kinodynamics-based pose optimization for humanoid loco-manipulation," *arXiv preprint arXiv:2303.04985*, 2023.