

# Model Predictive Robot- Environment Interaction Control for Mobile Manipulation Tasks

**Conference Paper****Author(s):**

Minniti, Maria V.; Grandia, Ruben; Fäh, Kevin; Farshidian, Farbod; [Hutter, Marco](#) 

**Publication date:**

2021

**Permanent link:**

<https://doi.org/10.3929/ethz-b-000476635>

**Rights / license:**

[In Copyright - Non-Commercial Use Permitted](#)

**Originally published in:**

<https://doi.org/10.1109/ICRA48506.2021.9562066>

**Funding acknowledgement:**

780883 - subTerranean Haptic INvestiGator (EC)

# Model Predictive Robot-Environment Interaction Control for Mobile Manipulation Tasks

Maria Vittoria Minniti, Ruben Grandia, Kevin Föh, Farbod Farshidian, Marco Hutter

**Abstract**—Modern, torque-controlled service robots can regulate contact forces when interacting with their environment. Model Predictive Control (MPC) is a powerful method to solve the underlying control problem, allowing to plan for whole-body motions while including different constraints imposed by the robot dynamics or its environment. However, an accurate model of the robot-environment is needed to achieve a satisfying closed-loop performance. Currently, this necessity undermines the performance and generality of MPC in manipulation tasks. In this work, we combine an MPC-based whole-body controller with two adaptive schemes, derived from online system identification and adaptive control. As a result, we enable a general mobile manipulator to interact with unknown environments, without any need for re-tuning parameters or pre-modeling the interacting objects. In combination with the MPC controller, the two adaptive approaches are validated and benchmarked with a ball-balancing manipulator in door opening and object lifting tasks.

## I. INTRODUCTION

Mobile service robots deployed in human habitats will face a wide variety of manipulation tasks. Assisted lifting of objects, cleaning of windows, or opening of doors, all require deliberate interaction between the robot and its surroundings. Solving such robot-environment interaction control tasks in a stable and repeatable manner is still an open problem. For fixed-base robots, state-of-the-art control schemes [1], [2], [3] can define and limit the interaction forces in a reactive manner. However, if the manipulator is mounted on a mobile base, control action is additionally demanded for the stabilization of the robotic platform itself. This is especially the case for underactuated mobile bases, such as humanoids, quadrupeds or wheeled balancing systems, that require a planning strategy to maintain their balance [4]–[6]. For known environments [7] or simple interactions [8], planning for the robot motion and interaction has been addressed by a Model Predictive Controller (MPC), which deals with the constraints and dynamics by solving an optimal control problem in a receding horizon fashion.

While robust to certain model mismatches, the lack of a precise system model can profoundly affect the MPC’s performance. For interaction tasks, this includes a model of

This work was supported in part by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics (NCCR Robotics), in part by the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab), in part by Intel Network on Intelligent Systems, and in part by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 780883.)

All authors are with Robotic Systems Lab, ETH Zurich, Zurich 8092, Switzerland {mminniti, rgrandia, farbodf, mahutter}@ethz.ch



Fig. 1: A ball-balancing manipulator opening a hinged door. The robot is required to maintain its balance and, at the same time, follow a desired end-effector trajectory while interacting with an unknown, external environment.

the environment that the robot should interact with, which, in general, is not available and needs to be extracted in advance. To tackle this issue, we extend an SLQ-MPC framework introduced in [9] by two adaptive control strategies, that are based on model identification adaptive control (MIAC) [10] and model reference adaptive control (MRAC) [11], [1]. The design of the proposed methods relies on the assumption that the environment behaves as a linear mechanical system along the interaction trajectory direction. We benchmark the performance of these two adaptive approaches and the baseline non-adaptive MPC for opening a door and lifting a weight. Such applications are typically used to evaluate robotic manipulation skills [12]. We then conclude that while both techniques improve the MPC’s performance during the interaction, the MRAC method demonstrates a more reliable and robust performance across the two different tasks.

For the experiments, a *ballbot* with an arm has been employed, as shown in Fig. 1. Due to its underactuation, a ballbot requires a whole-body control strategy that coordinates the movement and balance of the manipulator. Thus, controlling such a platform is an intriguing problem, that has received growing attention in the recent years [13], [14], [15].

### A. Related Work

Classical methods in robot-environment interaction control are based on direct and indirect force control. In direct force control [1], the manipulation task is performed by tracking a

desired force. This method is effective in all those scenarios where specific force profiles have to be applied. However, the generation of suitable force trajectories is not trivial without prior knowledge about the environment and the task. Indirect force control allows for tracking a desired force by exploiting its relation with deviations in position and can be formulated as an impedance or admittance controller [16], [17]. Impedance control allows tuning the controller's stiffness and damping based on the desired task, but its performance degrades when the environment's model is unknown or time-varying. Recent approaches propose to learn an error force term to compensate for the uncertainties in the robot and the environment model [18]. Another possibility is to change the controller gains based on the force tracking error [19], or an online estimate of the environment model [20], [21]. The MIAC method presented in this letter fits in this category, with the novelty of using the environment estimates to improve the internal model of the MPC planner in real-time.

On the other hand, adaptive controllers have proven to be very effective when dealing with the uncertainty of the inertial parameters in the dynamic model of robotic manipulators [1], [22]. In such methods, the adaptive parameters do not necessarily converge to the real values, but are updated to achieve convergence of the tracking error. Analysis on adaptive control has also been conducted in the context of force/position control in unknown interactions [23], [24], and it has mainly considered fixed-base, fully-actuated manipulators. The MRAC controller presented in this work focuses on obtaining an adaption law for the environment model, to be set as a constraint in an MPC module that can plan for motion and interaction force for a general mobile manipulator system. There are previous works that have used an adaptive law to update a parametric MPC model [25], [26]. However, so far this combination has not been addressed in a mobile manipulation scenario, especially when dealing with underactuated platforms.

## B. Contributions

The main contribution of this work is to enable an MPC strategy to plan and control mobile-manipulation tasks that involve interaction with an unknown environment, without the need for re-tuning the control parameters for different tasks or offline modeling. To this end, we propose to combine a whole-body MPC scheme with two adaptive control techniques, based on system identification and model reference adaptive control. We benchmark the performance of the two described methods in two different representative manipulation tasks. We experimentally demonstrate that the combination of MPC and the model reference adaptive controller enables the robot to complete the desired tasks, while the performance of the baseline controller degrades, and the MPC scheme relying on online system identification fails to generalize across different tasks.

## II. MPC PROBLEM

In our proposed method, an MPC solver manages the on-line switching between subsystems (e.g., the change between cost functions and constraints) and the system model update at the start of every MPC iteration. We employ an SLQ-MPC method [27], which solves the underlying optimal control problem using the Sequential Linear Quadratic (SLQ) algorithm [28], while allowing to switch between subsystems.

Let  $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n$  be the generalized positions and velocities of the robot, and let  $\boldsymbol{\tau} \in \mathbb{R}^m$  be the actuation torques. In addition, the end-effector pose is defined as  $\boldsymbol{\chi}_{ee} \in SE(3)$ . The employed non-linear optimal control formulation aims at minimizing the cost function

$$J^*(\mathbf{x}_{ss}(0)) = \min_{\mathbf{u}(\cdot)} \int_0^T \|\mathbf{x}_{ss}^d - \mathbf{x}_{ss}\|_{\mathbf{Q}_{ss}}^2 + \|\mathbf{u}^d - \mathbf{u}\|_{\mathbf{R}}^2 + \|\boldsymbol{\chi}_{ee}^d \boxminus \boldsymbol{\chi}_{ee}\|_{\mathbf{Q}_{ee}}^2 dt, \quad (1)$$

where the state and input at time  $t$  are defined as  $\mathbf{x}_{ss} := (\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{2n}$ ,  $\mathbf{u} := \boldsymbol{\tau} \in \mathbb{R}^m$ , and we use the symbol  $\boxminus$  to denote the task-space pose error, as defined in [1] for the quaternion rotation part. The optimization is subject to a set of constraints:

$$\mathbf{x}_{ss}(0) = \mathbf{x}_{ss0} \quad (2)$$

$$\dot{\mathbf{x}}_{ss} = \mathbf{f}(\mathbf{x}_{ss}, \mathbf{u}) \quad (3)$$

$$\mathbf{h}(\mathbf{x}_{ss}, \mathbf{u}, t) \geq \mathbf{0}. \quad (4)$$

The initial condition of the MPC problem (2) coincides with the robot measured state. The system in (3) consists of the robot's dynamical model and the interaction model, which are discussed in the following section. Inequality constraints (4) include joint angle and torque limits, and are treated as soft constraints according to a relaxed log-barrier function method [29].

## III. INTERACTION CONTROL

### A. Control Task

The control task considered in this work is to follow a trajectory with the robot end-effector while having contact with an unknown environment. For trajectory generation, we use the time-optimal algorithm described in [30], which allows for generating position and velocity trajectories for the end-effector with bounded acceleration and velocity magnitudes. Its simple implementation also allows fast recomputation. This is important for the door opening task, where the door hinge position and current angle are estimated online and thus the trajectory needs to be continuously updated.

### B. Assumptions and Notation

Let  $\boldsymbol{\lambda}_{ee} \in \mathbb{R}^3$  be the interaction force between the robot end-effector and the environment, and let  $\mathbf{v}(t) \in \mathbb{R}^3$  be the unit vector tangent to the manipulation trajectory. Without loss of generality, we suppose  $\boldsymbol{\lambda}_{ee} = \lambda \mathbf{v}$ , with  $\lambda \in \mathbb{R}$ ; possible normal components could be compensated using force feedback, without modifications to the following derivation.

We state the following assumptions:

- The contact between the end-effector and the environment is rigid.
- The environment model can be described by a 3D second-order system:

$$\lambda_{ee} = \bar{M}\ddot{x}_{\text{env}} + \bar{B}\dot{x}_{\text{env}} + \bar{K}(x_{\text{env}} - x_0) + f_s, \quad (5)$$

where  $x_{\text{env}} \in \mathbb{R}^3$  is the position of the environment interaction point,  $\bar{M}, \bar{B}, \bar{K} \in \mathbb{R}^{3 \times 3}$  are positive symmetric impedance matrices,  $f_s \in \mathbb{R}^3$  is a static force, and  $x_0 \in \mathbb{R}^3$  is a given spring equilibrium position.

- The manipulation direction  $v$  is a principal direction for  $\bar{M}, \bar{B}, \bar{K}$ , with  $m, b, k \in \mathbb{R}$  as associated eigenvalues. These are the mass, damping and stiffness of the environment object.

The model in Eq. (5) applies to a number of interaction objects used in service robotics tasks. For instance, doors and windows usually have a spring and damping behavior. Carts and payloads can also be described by Eq. (5), setting  $\bar{B} = \mathbf{0}$ ,  $\bar{K} = \mathbf{0}$ . The stated assumptions yield:

$$\lambda = v^T \lambda_{ee} = mv^T \ddot{x}_{ee} + bv^T \dot{x}_{ee} + kv^T (x_{ee} - x_0) + v^T f_s,$$

where  $x_{ee} = x_{\text{env}}$ , since the contact is rigid. To derive a controller for the environment system, this equation is written as a one-dimensional second-order linear system. Many state space representations are possible, depending on the trajectory  $v$ . One possibility that allows to generalize among many trajectories consists in letting  $x := v^T x_{ee}$ , which yields:

$$\lambda = m\ddot{x} + b\dot{x} + k(x - x_0) + f_s + \mathcal{O}(\dot{v}), \quad (6)$$

where  $f_s = v^T f_s$  is the projected static force. We now assume the direction  $v(t)$  to be constant or slowly varying; this applies to a number of relevant tasks, which have usually linear or circular paths on relatively small angles. Thus, we obtain a robot-environment model as

$$M(q)\ddot{q} + n(q, \dot{q}) = S^T \tau - J_{ee}(q)^T (f_{ee} + \lambda_{ee}), \quad (7a)$$

$$m\ddot{x} + b\dot{x} + k(x - x_0) + f_s = \lambda, \quad (7b)$$

where  $M \in \mathbb{R}^{n \times n}$ , and  $n \in \mathbb{R}^n$  are the robot's mass matrix and non-linear terms, respectively.  $S^T \in \mathbb{R}^{n \times m}$  maps the actuator torques to generalized torques and, in the case of a ballbot system, is a non-linear function of the joint positions  $q$ .  $J_{ee}(q) \in \mathbb{R}^{3 \times n}$  is the position jacobian of the end-effector frame. Since  $x$  is a function of  $q$ , (7b) can be written in terms of  $q, \dot{q}, \ddot{q}$ . Substituting  $\lambda_{ee} = \lambda v$  from (7b) into (7a) and converting to state-space form, the system dynamics in (3) is obtained.

The term  $f_{ee} \in \mathbb{R}^3$  is an additional virtual force, used as a control term. In its simplest implementation it takes the form of a PD controller:

$$f_{ee} = K(x_d - x) + D(\dot{x}_d - \dot{x}). \quad (8)$$

Similar to what is done in inverse dynamics-based impedance controllers, this term is not part of the true robot equations of motion, but modifies them so that the planned control

torques produce an additional force in the operational space. The terms  $m, b, k, f_s, x_0, v, f_{ee}$  are updated in the system dynamics at the beginning of each MPC iteration.

### C. Model Identification Adaptive Control (MIAC)

The objective of MIAC is to estimate the environment model parameters by means of an online system identification method, and to use these estimates to improve control performance. Since MPC is employed as the main controller module, the estimated parameters can be used in the system dynamics to model the environment. The block diagram of the resulting closed-loop system is indicated in Fig. 2. A simpler version of this method would be to directly compensate the external force using measurements from a force sensor. However, estimating the parameters allows for a more accurate prediction by the MPC since a model of the environment is available that can be used during the prediction and optimization over the time horizon.

Estimating the environment impedance parameters is a classical problem in robotics [31]. Available methods include Recursive [31], Least Mean Square [32] estimation, and algebraic manipulations [33]. Here, we propose the use of a Kalman filter as online system identification method to estimate the environment's mass  $m$ , damping  $b$ , spring stiffness  $k$  and static force  $f_s$ . Thus, Eq. (7b) is transformed into a state-space model with the substitutions  $x_1 := x - x_0$ ,  $x_2 := \dot{x}$ . Converting the resulting model in discrete time with sampling time  $T_s$ , the following measurement update equation can be extracted:

$$\lambda[k] = \begin{bmatrix} \frac{x_2[k] - x_2[k-1]}{T_s} & x_2[k] & x_1[k] & 1 \end{bmatrix} \begin{bmatrix} m \\ b \\ k \\ f_s \end{bmatrix} + w[k], \quad (9)$$

with zero mean Gaussian measurement noise  $w$ . Eq. (9) can be compactly written as:

$$z_k = H_k \pi_k + w_k, \quad (10)$$

with measurement  $z$  and estimated parameters  $\pi$ . The process model of the filter is chosen as a random walk of the parameters  $\pi$  with process noise  $n \sim \mathcal{N}(\mathbf{0}, Q)$ :

$$\pi_k = \pi_{k-1} + n. \quad (11)$$

As shown in Eq. (9), the filter measurement is the interaction force  $\lambda$  along the manipulation trajectory. This can be acquired with a force sensor, or with an observer approach [34]. Regarding the measurement update equation (9), depending on the task, the two terms  $m$  and  $f_s$  may be linearly dependent on each other. For instance, if the manipulation is in vertical direction with a free object, then  $f_s = mg$ . Or, if the manipulated object is pushed in horizontal direction,  $f_s = kmg$ , where  $k$  is a Coulomb friction coefficient. However, to avoid loss of generality, in this formulation they are considered separate from each other. As a result,  $m$  is observable only in the presence of persistent accelerations  $\ddot{x}$ .

By providing the estimates  $\hat{m}, \hat{b}, \hat{k}, \hat{f}_s$  to the MPC solver, at the next iteration, the optimal control planner becomes aware of the evolution of the robot-environment model.

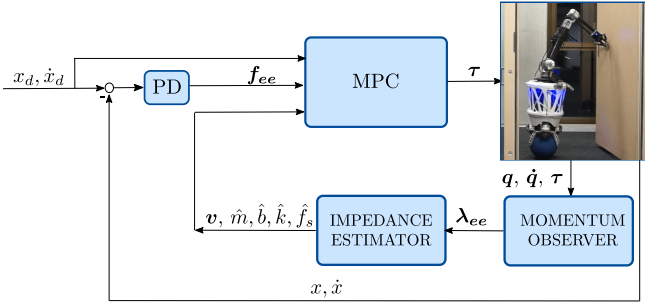


Fig. 2: Block diagram of the closed-loop system under the MIAC controller.

#### D. Model Reference Adaptive Control (MRAC)

As described in [1], [11], the design method for an MRAC controller consists of finding a control law that is dependent on the reference tracking error and includes adaptive parameters. Then, a Lyapunov function is formulated to derive both the parameters adaption law and a certification of the system stability. Here, this procedure is simplified since the model (7b) is linear in the environment parameters. Thus, it can be written as:

$$m\ddot{x} + b\dot{x} + k(x - x_0) + f_s = \mathbf{Y}(\ddot{x}, \dot{x}, x)\boldsymbol{\pi} = \lambda, \quad (12)$$

where  $\boldsymbol{\pi} := [m, b, k, f_s]^T$ ,  $\mathbf{Y}(\ddot{x}, \dot{x}, x) := [\ddot{x}, \dot{x}, x - x_0, 1]$ . The control law is defined as

$$\begin{aligned} \lambda &= \hat{m}\ddot{x}_r + \hat{b}\dot{x}_r + \hat{k}(x - x_0) + \hat{f}_s + k_s\sigma, \\ &= \mathbf{Y}(\ddot{x}_r, \dot{x}_r, x)\hat{\boldsymbol{\pi}} + k_s\sigma, \end{aligned} \quad (13)$$

with  $k_s > 0$  as a tuning parameter and with the following definitions:

$$\begin{aligned} \tilde{x} &= x_d - x, & \dot{x}_r &= \dot{x}_d + \Lambda\tilde{x}, & \Lambda &> 0, \\ \ddot{x}_r &= \ddot{x}_d + \Lambda\dot{\tilde{x}}, & \sigma &= \dot{x}_r - \dot{x} = \dot{\tilde{x}} + \Lambda\tilde{x}. \end{aligned} \quad (14)$$

A Lyapunov function for the controlled system can be defined as

$$V(\sigma, \tilde{x}, \tilde{\boldsymbol{\pi}}) = \frac{1}{2}m\sigma^2 + \Lambda k_s \tilde{x}^2 + \frac{1}{2}\tilde{\boldsymbol{\pi}}^T \mathbf{K}_\pi \tilde{\boldsymbol{\pi}} > 0, \quad (15)$$

where  $\tilde{\boldsymbol{\pi}} \in \mathbb{R}^4$  is the vector of adaptive parameters,  $\tilde{\boldsymbol{\pi}} := \hat{\boldsymbol{\pi}} - \boldsymbol{\pi}$ , and  $\mathbf{K}_\pi \in \mathbb{R}^{4 \times 4}$  is a positive definite matrix. Imposing  $\dot{V} \leq 0$ , the adaption law can be derived:

$$\dot{\hat{\boldsymbol{\pi}}} = \mathbf{K}_\pi^{-1} \mathbf{Y}^T(\ddot{x}_r, \dot{x}_r, x)\sigma. \quad (16)$$

The control law (13) is the sum of an adaptive term  $\mathbf{Y}(\ddot{x}_r, \dot{x}_r, x)\hat{\boldsymbol{\pi}}$  and a PD-feedback term  $k_s\sigma$ . The control force to be tracked by the MPC controller is computed as:

$$\mathbf{f}_{ee} = (\mathbf{Y}(\ddot{x}_r, \dot{x}_r, x)\hat{\boldsymbol{\pi}} + k_s\sigma)\mathbf{v}. \quad (17)$$

With respect to the MIAC controller, here the estimates of the environment parameters are not considered in the MPC. The system dynamics thus reduces to Eq. (7a) with  $\boldsymbol{\lambda}_{ee} = \mathbf{0}$ . Indeed, the goal of this method is not to exactly compensate the environment model in Eq. (7b), but rather to modify the controller with an adaptive law that guarantees the convergence of the tracking error along the manipulation trajectory.

## IV. EXPERIMENTS

The control structures presented in the previous sections III-C, III-D are evaluated in two experimental scenarios. In the first scenario, the robot is tasked with opening a hinged door. In the second one, the robot is tasked with lifting an object of unknown mass. The experiments are used to assess how modifying the MPC controller using the environment estimated/adapted parameters can improve the closed-loop tracking performance. A video showcasing the results accompanies this paper<sup>1</sup>.

### A. Experimental Setup

The robot employed for the tests is a ballbot with a 4-degree of freedom (DOF) arm attached to the base and a Robotiq-2F85 gripper as end-effector. The system is shown in Fig. 1. It is modeled with 8 DOFs, resulting in a dimensionality of the MPC system of 16 states. The reference torques sent to the actuators are computed as  $\boldsymbol{\tau}^{ref} = \boldsymbol{\tau}^* + \mathbf{K}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}})$ , where  $\boldsymbol{\tau}^*$ ,  $\mathbf{q}^*$ ,  $\dot{\mathbf{q}}^*$  are the optimal joint torques, positions and velocities. The MPC solver computes its optimal solution at a frequency of 200 Hz over a time horizon of 1.5 s, and is implemented with the OCS2 toolbox [35]. Rigid-body dynamics and kinematics equations are generated using the RobCoGen library [36]. State estimation, motion planning and control run on the onboard PC, which is an Intel Core i7, NUC10i7FNK Mini PC.

The interaction force used as a measurement for the Kalman Filter of Sec. III-C is computed by means of a momentum observer [34], which allows to estimate the generalized external torques  $\boldsymbol{\tau}_{ext}$  from joint position, velocity, and torque measurements. Then, an augmented Jacobian method is employed to retrieve the external force on the end-effector. This is implemented by defining

$$\mathbf{J}_{aug} := \begin{bmatrix} \mathbf{J}_{base} \\ \mathbf{J}_{ee} \end{bmatrix}, \quad (18)$$

where  $\mathbf{J}_{base} = [\mathbf{I}_{n_b \times n_b}, \mathbf{0}_{n_b \times (n - n_b)}]$  is the Jacobian corresponding to the  $n_b$  degrees of freedom of the base. Then,  $\boldsymbol{\lambda}_{ee}$  is computed from:

$$\begin{bmatrix} \boldsymbol{\tau}_{base} \\ \boldsymbol{\lambda}_{ee} \end{bmatrix} = \mathbf{J}_{aug}^{-T} \boldsymbol{\tau}_{ext}. \quad (19)$$

### B. Door Opening

In this experiment, the robot should open a door up to a desired angle of 70°. The desired end-effector path is a circular trajectory. An extended Kalman Filter (EKF) estimates the door radius, angle, and hinge position, which are used for the re-planning of the time-optimal trajectory generator of Sec. III-A. Based on given velocity and acceleration bounds and on the current door measurements, the time optimal trajectory generator determines the door angle references and the remaining door-opening time. The parameters of the PD controller (8) are specified in a frame rigidly attached to the door, in order to achieve a higher compliance in the radial

<sup>1</sup>Available at [https://youtu.be/A7\\_e-UWkfXo](https://youtu.be/A7_e-UWkfXo).



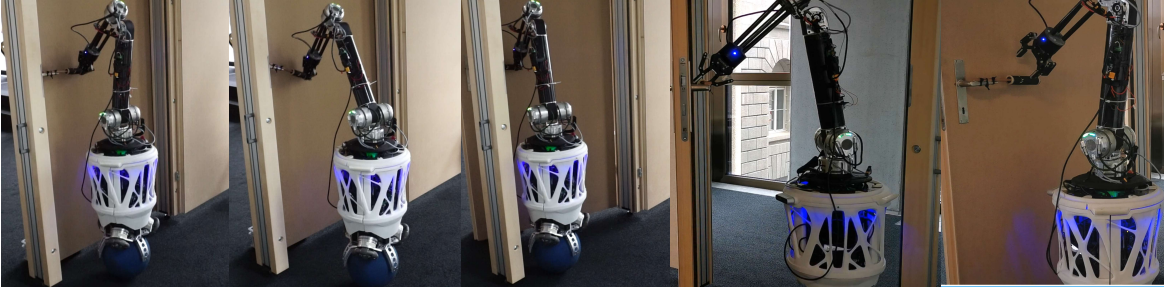


Fig. 3: Door opening experiment with a ball-balancing manipulator. The robot is tasked with opening the door up to a desired angle of  $70^\circ$ , while tracking a circular trajectory.

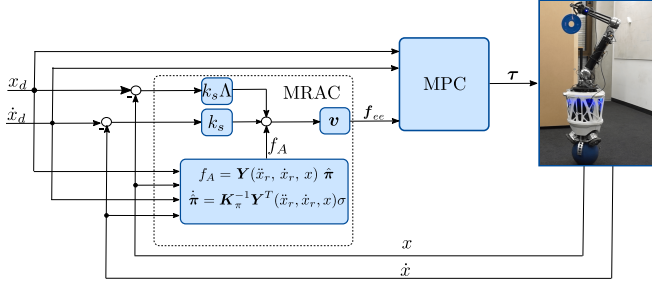


Fig. 4: Block diagram of the closed-loop system under the cascade MRAC-MPC controller.

direction and a lower compliance in the tangential direction. Since the mass of the original door is not heavy enough to result in a challenging application, it is modified by attaching two boxes of intermediate weight (1.5 kg) and larger weight (4.2 kg), respectively, behind the door.

During the heavy-door experiment, both the end-effector and the ball-trajectory are commanded to track a desired circular trajectory. The reference for the ball is necessary to avoid a collision with the door frame when the ballbot leans forward while pushing. Both of the tests are repeated under three different controllers, which are: 1) the baseline MPC controller, with an additional PD force input as in (8). 2) the MIAC controller presented in section III-C. 3) the MRAC controller presented in section III-D. To compare the tracking performance of the different controllers for the opening of the two doors, we compute the root mean square tracking error (RMSE), and the final error between the desired and estimated door angle which are visible in Table I.

1) *Light door*: For this experiment, forces in the range of 10 – 15 N need to be applied to open the door. The door angle tracking performance is shown in Fig. 5. For the baseline controller, the tracking error is large at all times. On the contrary, the tracking error is comparably small for all the other controllers, with a final door error of less than  $4^\circ$ .

2) *Heavy door*: In this experiment, forces in the range of 20 – 25 N need to be applied to open the door. The door angle tracking performance for this test is visualized in Fig. 5. We notice that the desired and measured door angles exhibit more oscillations than in the previous test. In fact, due to the higher required pushing force, the controller plans to lean the base forward at a larger angle. Tilting the base additionally

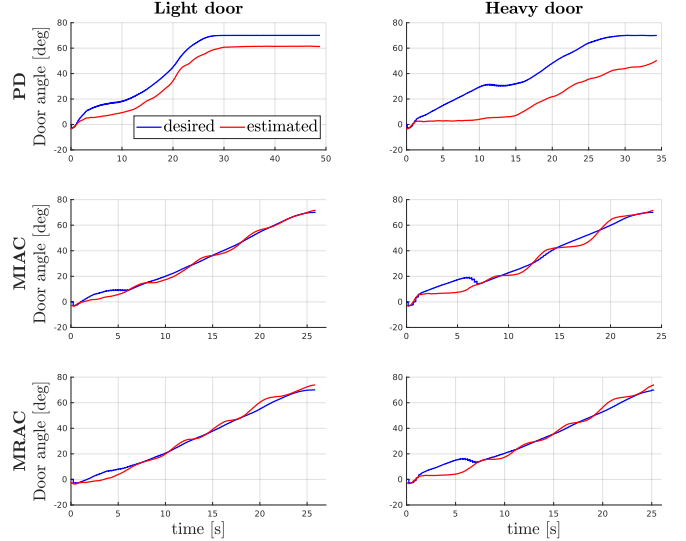


Fig. 5: From top to bottom: desired and estimated door angle under the PD, MIAC, MRAC controllers, for the door opening experiments. The plots are referred to the light and heavy door experiment on the left and right, respectively.

TABLE I: RMSE [deg] and final door errors [deg] of the three interaction controllers in the two door-opening test scenarios.

Controller	Light Door		Heavy Door	
	RMSE	Final Error	RMSE	Final Error
Impedance (PD)	9.5	8.74	19.5	19.7
MIAC	2.7	<b>-1.64</b>	<b>4.9</b>	<b>-1.7</b>
MRAC	<b>2.6</b>	-3.9	5.8	-3.9

influences the ball position, resulting in a trade-off between this effect and the door frame collision avoidance term.

Both the MIAC controller and the MRAC-MPC controller produce a considerable reduction of the tracking error, as it can be verified in Table I. The MRAC controller can fast adapt to the unknown environment due to the balance between the PD and adaptive terms, whose contribution is shown in Fig. 6. As the time passes, the PD term becomes less relevant and most of the control action comes from the adaptive input. Regarding the MIAC estimates, since the employed door has no spring behaviour, the most significant parameters are the door damping and static friction, whose

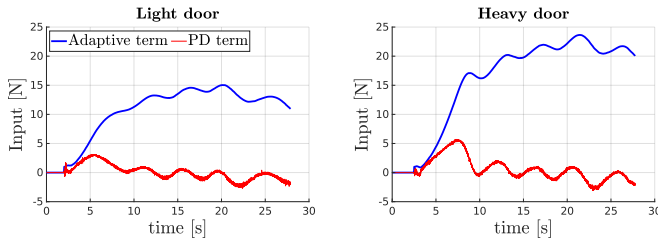


Fig. 6: Contribution of the adaptive and PD terms in the MRAC control law (13) for the light door opening test (on the left) and heavy door opening test (on the right).

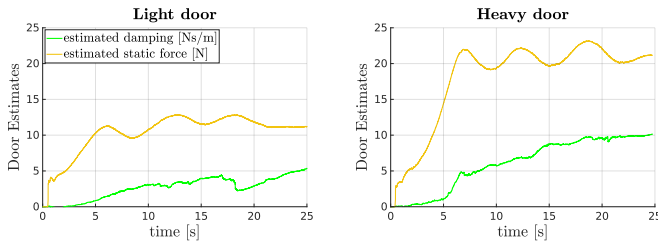


Fig. 7: Estimation of the damping and static force parameters for the light door opening experiment (on the left), and heavy door opening experiment (on the right) using the MIAC controller.

estimates are plotted in Fig. 7. The non-adaptive baseline impedance controller performs poorly without re-tuning of the stiffness and damping gains and is not able to open the door up to the desired angle.

### C. Lifting Unknown Objects

In this experiment, the robot is tasked with lifting an unmodeled payload of 2 kg. One of the assumptions under which the interaction controllers are derived in section III is that the robot dynamic model in (7a) is perfectly known. To evaluate how this assumption may influence the controller performance, the same experiment is executed both in a Gazebo simulation, where the robot model is perfect, and on the physical system, where some degree of model mismatch is present. The Gazebo simulation model is based on [8], [37]. While in the door opening scenario the robot interacts with a statically stable object, the equilibrium of the object being lifted is attained at a non-zero interaction force, which adds an element of complexity to the task.

1) *Simulation*: The measured and desired trajectory along the end-effector vertical direction is depicted in Fig. 8. Analyzing the plots for the simulation test under the baseline controller, it can be inferred that, if the MPC is unaware of the lifted payload, the robot is not able to track the desired trajectory. Conversely, the end-effector position error is minimized if the environment online estimates are included in the optimization, in both an estimated or adaptive manner, as it can be noticed from the plots referring to the MIAC and MRAC controllers.

2) *Hardware*: The same experiment is performed on the physical system, with corresponding plots shown in Fig. 8. The baseline MPC controller has an increasing tracking error.

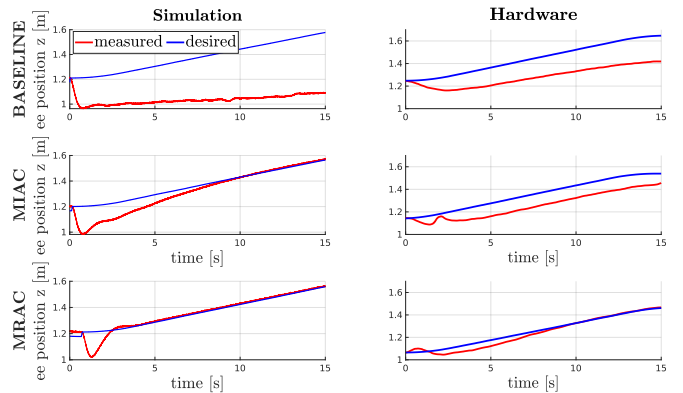


Fig. 8: From top to bottom: measured and desired end-effector position for the unknown object lifting experiment under the non-adaptive baseline MPC, MIAC, and MRAC, respectively. The plots are referred to simulations (on the left), and hardware tests (on the right).

This gets reduced if the whole-body MPC is combined with the proposed interaction control formulation. However, the MIAC controller is strongly influenced by imprecise actuator torque measurements used for the force estimation, and consequent degradation of the environment identification. As a result, the closed-loop system under the MIAC controller has a slower convergence than in the simulation. On the contrary, the MRAC controller manages to consistently reduce the tracking error in the same amount of time, with the payload being successfully moved to the desired height. Furthermore, we point out that the same gains were used for the MRAC in both the door opening experiment from section IV-B and the lifting test. However, as shown in figure Fig. 8, for the MIAC controller, direct transfer to a different task while maintaining the same tracking behavior was not possible with the same parameters used for system identification.

## V. CONCLUSIONS AND FUTURE WORK

This paper focuses on making a whole-body MPC controller adaptive to mobile manipulation tasks in unknown environments. Thus, a system identification and an adaptive control method are proposed to extend the MPC formulation. From experimental tests, it can be concluded that both the MIAC method and the cascade MRAC-MPC outperform the original MPC controller in mobile manipulation tasks applied to unknown environments. However, the MRAC method generalizes better across tasks. In fact, it is independent of interaction forces estimation or the convergence of the MIAC system identification method. The employed modeling strategy was derived under the assumption that the environment can be described by a linear mass-spring-damper system. As a future direction, we intend to generalize the formulation to general multi-DOF tasks (e.g. a collaborative task). Furthermore, so far, the objective has been to achieve the desired closed-loop performance in practice. Providing theoretical guarantees for stability and feasibility in MPC is an active research area, which we will further pursue in the context of robot-environment interaction.

## REFERENCES

- [1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [2] C. Natale, *Interaction control of robot manipulators: six degrees-of-freedom tasks*. Springer Science & Business Media, 2003, vol. 3.
- [3] M. Vukobratovic, D. Surdilovic, Y. Ekalo, and D. Katic, *Dynamics and Robust Control of Robust Robot-Environment Interaction*. World Scientific Publishing, 2009.
- [4] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3346–3351.
- [5] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *arXiv preprint arXiv:2103.00946*, 2021.
- [6] N. Gupta, J. Smith, B. Shrewsbury, and B. Børnich, "2d push recovery and balancing of the eve r3-a humanoid robot with wheel-base, using model predictive control and gain scheduling," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 365–372.
- [7] D. Lee, H. Seo, D. Kim, and H. J. Kim, "Aerial manipulation using model predictive control for opening a hinged door," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1237–1242.
- [8] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, "Whole-body mpc for a dynamically stable mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3687–3694, 2019.
- [9] F. Farshidian, E. Jelavic, A. Satapathy, M. Gifftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *Humanoids*, 2017, pp. 577–584.
- [10] Z. Öreg, H.-S. Shin, and A. Tsourdos, "Model identification adaptive control-implementation case studies for a high manoeuvrability aircraft," in *2019 27th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2019, pp. 559–564.
- [11] N. T. Nguyen, "Model-reference adaptive control," in *Model-Reference Adaptive Control*. Springer, 2018, pp. 83–123.
- [12] A. H. Quispe, H. B. Amor, and H. I. Christensen, "A taxonomy of benchmark tasks for robot manipulation," in *Robotics Research*. Springer, 2018, pp. 405–421.
- [13] Z. Li and R. Hollis, "Toward a ballbot for physically leading people: A human-centered approach," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4827–4833.
- [14] F. Sonnleitner, R. Shu, and R. L. Hollis, "The mechanics and control of leaning to lift heavy objects with a dynamically stable mobile robot," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9264–9270.
- [15] M. Shomin, J. Forlizzi, and R. Hollis, "Sit-to-stand assistance with a balancing mobile robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3795–3800.
- [16] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," 1985.
- [17] H. Kazerooni, T. Sheridan, and P. Houpt, "Robust compliant motion for manipulators, part i: The fundamental concepts of compliant motion," *IEEE Journal on Robotics and Automation*, vol. 2, no. 2, pp. 83–92, 1986.
- [18] W. Amanhoud, M. Khoramshahi, M. Bonnesoeur, and A. Billard, "Force adaptation in contact tasks with dynamical systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6841–6847.
- [19] K. Lee and M. Buss, "Force tracking impedance control with variable target stiffness," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6751–6756, 2008.
- [20] N. Diolaiti, C. Melchiorri, and S. Stramigioli, "Contact impedance estimation for robotic systems," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 925–935, 2005.
- [21] V. Mallapragada, D. Erol, and N. Sarkar, "A new method of force control for unknown environments," *International Journal of Advanced Robotic Systems*, vol. 4, no. 3, p. 34, 2007.
- [22] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.
- [23] L. Villani, C. C. De Wit, and B. Brogliato, "An exponentially stable adaptive control for force and position tracking of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 798–802, 1999.
- [24] J. Roy and L. L. Whitcomb, "Adaptive force control of position/velocity controlled robots: theory and experiment," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 121–137, 2002.
- [25] B. Zhu and X. Xia, "Lyapunov-based adaptive model predictive control for unconstrained non-linear systems with parametric uncertainties," *IET Control Theory & Applications*, vol. 10, no. 15, pp. 1937–1943, 2016.
- [26] B. Zhu, Z. Zheng, and X. Xia, "Constrained adaptive model-predictive control for a class of discrete-time linear systems with parametric uncertainties," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 2223–2229, 2019.
- [27] F. Farshidian, E. Jelavic, A. Satapathy, M. Gifftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 577–584.
- [28] F. Farshidian, M. Kamgarpour, D. Pardo, and J. Buchli, "Sequential linear quadratic optimal control for nonlinear switched systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1463–1469, 2017.
- [29] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," *arXiv preprint arXiv:1905.06144*, 2019.
- [30] F. Ramos, M. Gajamohan, N. Huebel, and R. D'Andrea, "Time-optimal online trajectory generator for robotic manipulators," 2013.
- [31] L. J. Love and W. J. Book, "Environment estimation for enhanced impedance control," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1995, pp. 1854–1859.
- [32] I. F. Jasim, P. W. Plapper, and H. Voos, "Gaussian filtering for enhanced impedance parameters identification in robotic assembly processes," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2015, pp. 1–4.
- [33] J. Beccedas, G. Mamani, V. Feliu, and H. Sira-Ramírez, *Estimation of Mass-Spring-Dumper Systems*. Dordrecht: Springer Netherlands, 2009, pp. 411–422.
- [34] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.
- [35] "OCS2: An open source library for optimal control of switched systems," [Online]. Available: <https://bitbucket.org/leggedrobotics/ocs2>.
- [36] M. Gifftthaler, M. Neunert, M. Stäuble, M. Frigerio, C. Semini, and J. Buchli, "Automatic differentiation of rigid body dynamics for optimal control and estimation," *Advanced Robotics*, vol. 31, no. 22, pp. 1225–1237, 2017.
- [37] P. Fankhauser and C. Gwerder, "Modeling and control of a ballbot," B.S. thesis, Eidgenössische Technische Hochschule Zürich, 2010.