

Adaptive CLF-MPC With Application To Quadrupedal Robots

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

Abstract—Modern robotic systems are endowed with superior mobility and mechanical skills that make them suited to be employed in real-world scenarios, where interactions with heavy objects and precise manipulation capabilities are required. For instance, legged robots with high payload capacity can be used in disaster scenarios to remove dangerous material or carry injured people. It is thus essential to develop planning algorithms that can enable complex robots to perform motion and manipulation tasks accurately. In addition, online adaptation mechanisms with respect to new, unknown environments are needed. In this work, we impose that the optimal state-input trajectories generated by Model Predictive Control (MPC) satisfy the Lyapunov function criterion derived in adaptive control for robotic systems. As a result, we combine the stability guarantees provided by Control Lyapunov Functions (CLFs) and the optimality offered by MPC in a unified adaptive framework, yielding an improved performance during the robot's interaction with unknown objects. We validate the proposed approach in simulation and hardware tests on a quadrupedal robot carrying un-modeled payloads and pulling heavy boxes.

Index Terms—Robust/Adaptive Control; Optimization and Optimal Control; Legged Robots

I. INTRODUCTION

One major challenge for the real-world deployment of robots is the absence of a perfect model of all the objects that the robot should manipulate. In addition, the model of the robot itself may be uncertain, since it is difficult to perfectly identify the dynamic parameters of all its components. Based on this motivation, a fruitful, vast amount of literature has been dedicated to the development of adaptive control algorithms [1] with a particular emphasis on direct approaches, such as the notable Slotine-Li adaptive controller for robot manipulators [2]. However, when it comes to the control of poly-articulated mobile platforms, such as quadrupedal robots (Fig. 1), it is necessary to devise methods that can reliably handle more complex, nonlinear performance objectives (such as foothold selection and end-effector tracking), as well as a set of safety constraints. Thus, much analysis is currently being devoted to the study of approaches aimed at combining adaptive control laws with powerful learning and optimization algorithms [3].

Manuscript received: September, 9, 2021; Accepted October, 30, 2021.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments.

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

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

Digital Object Identifier (DOI): <https://doi.org/10.1109/LRA.2021.3128697>



Figure 1: The quadrupedal robot ANYmal pulling a box of 21 kg (at the top), and holding a payload of 10.86 kg (at the bottom).

In [4], the disturbance experienced by the robot is parametrized through a neural network; the inner features parameters are meta-learned offline in closed-loop simulations of model ensembles under adaptive control. A deep network is used in [5] for quadrotor control; adaptation to different flight regimes is achieved using an MRAC law for the last layer weights. Research has also been conducted to add adaptive control stability guarantees in Quadratic Programs (QPs). Specifically for legged robots, an L1 adaptive controller is designed in [6], [7] to track a nominal model based on a QP. For systems in strict feedback form, online adaptation and safety can be obtained by incorporating adaptive Control Lyapunov Functions (aCLFs) [8], [9] and adaptive Control Barrier Functions (aCBFs) in the QP formulation [10]. Stochastic CBFs and stochastic CLFs have been used in QPs to obtain stability and safety conditions where model uncertainty is approximated via Bayesian model learning [11]. CLFs have already been used as a powerful tool to synthesize direct adaptive controllers [12]. In the context of infinite horizon optimal control, a direct adaptive controller can be obtained as the sum of the nominal optimal input, and an adaptive input that uses the optimal value function as a CLF [13].

Thanks to recent computational advances, Model Predictive Control (MPC) has shown real-time applicability on high-dimensional systems [14], [15]. However, the performance of nominal MPC degrades in the presence of model uncertainties. Robust MPC can cope with disturbances in a known set, although only few works have shown applications on real-world,

robotic systems [16]. Moreover, it has been demonstrated that online adaptation to the unknown parameters can provide MPC with robustness properties with respect to disturbances in a larger domain [17]. In robotics, adaptive MPC controllers have mostly been designed in combination with system identification [18], [19], [20], or learning methods [21]. For mobile-manipulation problems, analysis on the combination of MPC with adaptive control has been conducted in [22].

CLF stability criteria have also been added as constraints in MPC [23], [24], [25], where stability guarantees are usually obtained with a properly designed terminal cost and terminal constraint [26]. In the absence of such terminal conditions, which are complex to obtain for non-linear problems, MPC relies on the choice of the time horizon, with decreasing performance for smaller look-ahead. Incorporating CLF constraints within MPC allows to exploit the stability properties of CLF-QP controllers, and eases the tuning of the prediction horizon [25]. However, so far such advantages have not been shown in the presence of uncertainties in the robot non-linear MPC model.

Inspired by recent literature [25], we propose here to combine CLFs with MPC, targeting robotic applications that involve interaction with unknown environments/objects. The contributions of this letter are the following:

- We consider the Lyapunov condition derived in adaptive control for robotic manipulators [2], and we incorporate it as an inequality constraint to an MPC problem. Thus, the controller plans for an approximate system model, while ensuring asymptotic tracking in the presence of matched uncertainties. As a result, we combine the optimality of MPC with the advantages provided by CLFs. These include the stability guarantees and a reduced dependence on the prediction horizon.
- We present the implementation of the proposed formulation on a floating-base model of a quadrupedal robot.
- We validate the proposed setup in a variety of simulations and real-world hardware experiments. The simulations aim to demonstrate the advantages of the proposed approach with respect to some baseline methods. These characterize themselves for the presence/absence of a terminal penalty, the awareness of the model mismatch, or the use of a different adaptation mechanism. The hardware tests showcase a quadrupedal robot carrying bricks and pulling heavy boxes.

II. BACKGROUND

A. Nonlinear Model Predictive Control

We consider the following nonlinear optimal control problem (OCP):

$$\underset{u(\cdot)}{\text{minimize}} \quad \int_0^T l(\mathbf{x}(t), \mathbf{u}(t)) dt + \phi(\mathbf{x}(T)) \quad (1a)$$

$$\text{subject to:} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u} \quad (1b)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (1c)$$

$$\mathbf{g}_1(\mathbf{x}) = \mathbf{0} \quad (1d)$$

$$\mathbf{g}_2(\mathbf{x}, \mathbf{u}) = \mathbf{0} \quad (1e)$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \geq \mathbf{0} \quad (1f)$$

where $\mathbf{x}(t) \in \mathcal{X}$, $\mathbf{u}(t) \in \mathcal{U}$ are the state and input to the system, $l(\cdot)$ is an intermediate cost, $\phi(\cdot)$ is a terminal cost, and T is the time horizon. The OCP is solved in real-time by updating the initial conditions (1c) with the measured state of the system. Eq. (1b) describes the system dynamics, Eqs. (1d)–(1f) describe the state-only equality constraints, state-input equality constraints and inequality constraints, respectively. Here, we consider an MPC implementation where inequality constraints (1f) are treated according to a relaxed-barrier method [27]: a relaxed log-barrier function is computed as a function of the inequality constraint, and added to the cost minimization. Without loss of generality, we present the following derivation in the continuous-time domain. However, the proposed method does not depend on the specific implementation (e.g., continuous-time or discrete-time DDP, direct methods, etc...). In addition, in Sec. III we assume that Eq. (1b) represents a general mechanical system. In Sec. IV, we apply this general formulation on the simplified kino-dynamic model of a quadrupedal robot.

B. Control Lyapunov Functions

We consider an error function $\sigma : \mathcal{X} \times \mathbb{R}_+ \rightarrow \mathbb{R}^l$, with $l \leq \dim(\mathcal{X})$. Let $\mathcal{C} = \{\sigma(\mathbf{x}, t), \mathbf{x} \in \mathcal{X}, t \in \mathbb{R}_+\} \subseteq \mathbb{R}^l$. We assume that each component of σ has relative degree 1.

Definition 1 (Class \mathcal{K} -functions [25]). A continuous function $\alpha : [0, a) \rightarrow \mathbb{R}_+$, with $a > 0$, is a class- \mathcal{K} function ($\alpha \in \mathcal{K}$) if $\alpha(0) = 0$ and α is strictly monotonically increasing. If $a = \infty$ and $\lim_{r \rightarrow \infty} \alpha(r) = \infty$, then α is said to be a class- \mathcal{K}_∞ function.

Definition 2 (Control Lyapunov Functions [28], [25]). A continuously differentiable function $V : \mathbb{R}^l \rightarrow \mathbb{R}_+$ is said to be a Control Lyapunov Function (CLF) for (1b) on \mathcal{C} if there exist $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$ such that $\forall \sigma \in \mathcal{C}$:

$$\alpha_1(\|\sigma\|) \leq V(\sigma) \leq \alpha_2(\|\sigma\|) \quad (2)$$

$$\dot{V}(\sigma, \mathbf{u}) \leq -\alpha_3(\|\sigma\|) \quad (3)$$

The existence of a CLF $V(\sigma)$ is a necessary and sufficient condition for the existence of a state-feedback controller $k : \mathcal{X} \rightarrow \mathcal{U}$ that makes σ globally asymptotically converge to $\mathbf{0}$. As shown in Sec. III, for affine robotic systems and by an appropriate definition of σ , convergence to the error surface $\sigma = \mathbf{0}$ can also lead to the convergence of the full-system state \mathbf{x} to a desired state \mathbf{x}_d . In QP-based controllers, a method to obtain stability guarantees is to add inequality (3) to the set of constraints of the QP [29]. A natural extension for MPC problems consists in adding (3) to the set of inequality constraints of Eq. (1f) [23], [24], [25]. We build from this idea for the following derivation.

III. ADAPTIVE CLF-MPC FOR ROBOTIC SYSTEMS

A. Adaptive Optimal Control Problem

We want to control mechanical systems in the form:

$$\mathbf{M}_n(\mathbf{q})\dot{\mathbf{v}} + \mathbf{C}_n(\mathbf{q}, \mathbf{v})\mathbf{v} + \mathbf{g}_n(\mathbf{q}) + \boldsymbol{\tau}_u(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}) = \mathbf{S}(\mathbf{q})\boldsymbol{\tau} \quad (4)$$

where $\mathbf{q}, \mathbf{v} \in \mathbb{R}^n$, $\boldsymbol{\tau} \in \mathbb{R}^m$ are the generalized positions, velocities and torques of the robot, respectively, $\mathbf{S} \in \mathbb{R}^{n \times m}$

stabilizing input for (4). Theoretically, such a constraint is sufficient to guarantee global asymptotic stability. However, inequalities are usually treated as soft constraints [27]. On the other hand, the MPC dynamics (7b) is a hard constraint. As a consequence, if (7b) would only be based on the robot nominal model, it could conflict with the CLF constraint, leading to the latter not being satisfied. Thus, the term $\mathbf{Y}_u(\mathbf{q}, \mathbf{v}, \mathbf{v}_r, \dot{\mathbf{v}}_r)\hat{\boldsymbol{\pi}}_u$ is used in place of the true disturbance $\mathbf{Y}_u(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}})\boldsymbol{\pi}_u$ in the dynamic constraint of Eq. (10). The adaptive component $\mathbf{Y}_u\hat{\boldsymbol{\pi}}_u$ serves both as an estimate of the disturbance, and to generate the adaptation needed in the system dynamics to allow the satisfaction of the CLF constraint. Furthermore, this choice allows obtaining an MPC controller which is also a certainty equivalence controller [17]. Indeed, Eq. (10) is equivalent to incorporate the following two constraints in the MPC problem:

$$\mathbf{S}(\mathbf{q})\mathbf{w} := \mathbf{S}(\mathbf{q})\boldsymbol{\tau} - \mathbf{Y}_u(\mathbf{q}, \mathbf{v}, \mathbf{v}_r, \dot{\mathbf{v}}_r)\hat{\boldsymbol{\pi}}_u \quad (16)$$

$$\dot{\mathbf{v}} = \mathbf{M}_n(\mathbf{q})^{-1}[-\mathbf{C}_n(\mathbf{q}, \mathbf{v})\mathbf{v} - \mathbf{g}_n(\mathbf{q}) + \mathbf{S}(\mathbf{q})\mathbf{w}] \quad (17)$$

with the introduction of an auxiliary input $\mathbf{w} \in \mathcal{U}$. Eq. (17) represents the nominal system, while Eq. (16) can be solved exactly with respect to $\boldsymbol{\tau}$ because of the assumption of matched uncertainty. As a result, the optimal input $\boldsymbol{\tau}^*$ is given by the sum of a nominal input \mathbf{w}^* , and an adaptive disturbance estimate.

C. Convergence of the adaptive dynamics

In adaptive control, it is not required that the estimated parameters converge to the true values. However, for linear systems, there exist persistent excitation conditions that can be incorporated as MPC constraints to obtain such a convergence [31]. As pointed out in [30], analogous simple conditions do not exist for nonlinear systems. Thus, here we aim to achieve approximate convergence of the adaptive uncertainty $\mathbf{Y}_u\hat{\boldsymbol{\pi}}_u$ to the true uncertainty $\mathbf{Y}_u\boldsymbol{\pi}_u$ by an appropriate cost function tuning. We refer again to the auxiliary input \mathbf{w} defined in Eq. (16). Because of the constraint (7f), in closed-loop $\boldsymbol{\sigma} \rightarrow \mathbf{0}$, and thus the robot dynamics (13) reduces to:

$$\mathbf{S}(\mathbf{q})\boldsymbol{\tau} \approx \mathbf{Y}_n\boldsymbol{\pi}_n + \mathbf{Y}_u\boldsymbol{\pi}_u \Rightarrow \mathbf{S}(\mathbf{q})\mathbf{w} + \mathbf{Y}_u\hat{\boldsymbol{\pi}}_u \approx \mathbf{Y}_n\boldsymbol{\pi}_n + \mathbf{Y}_u\boldsymbol{\pi}_u.$$

We suppose that the MPC cost is designed such that the difference $\|\mathbf{S}(\mathbf{q})\mathbf{w} - \mathbf{Y}_n\boldsymbol{\pi}_n\|$ is minimized. We are left with:

$$\mathbf{Y}_u(\mathbf{q}, \mathbf{v}, \mathbf{v}_r, \dot{\mathbf{v}}_r)\hat{\boldsymbol{\pi}}_u \approx \mathbf{Y}_u(\mathbf{q}, \mathbf{v}, \mathbf{v}_r, \dot{\mathbf{v}}_r)\boldsymbol{\pi}_u + \epsilon$$

where ϵ is a residual error from the minimization of $\|\mathbf{S}(\mathbf{q})\mathbf{w} - \mathbf{Y}_n\boldsymbol{\pi}_n\|$. If such an error is small, the adaptive term $\mathbf{Y}_u\hat{\boldsymbol{\pi}}_u$ converges to the true uncertainty $\mathbf{Y}_u\boldsymbol{\pi}_u$. In the absence of persistently exciting references, this does not imply $\hat{\boldsymbol{\pi}}_u \approx \boldsymbol{\pi}_u$. However, it implies that the MPC dynamics approximately converges to the true dynamics.

IV. IMPLEMENTATION FOR A QUADRUPEDAL ROBOT

In this section, we describe the implementation of the MPC problem given in Eqs. (7a)–(8) for the 6-dimensional floating-base model of a quadrupedal robot interacting with un-modeled objects at its base (Fig. 3).

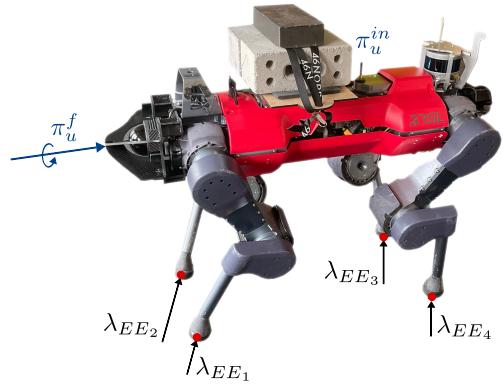


Figure 3: Illustration of the considered scenario for a quadrupedal robot. The MPC control inputs for the single rigid-body model are the feet contact forces λ_{EE_i} . The uncertain parameters are due to un-modeled payloads ($\boldsymbol{\pi}_u^{in}$) and constant wrenches acting on the base ($\boldsymbol{\pi}_u^f$).

A. Adaptive dynamics and constraints

To derive the adaptive modification of the MPC dynamics, we base ourselves on the kino-dynamic model described in [27]. The adaptive equations of motion are given by:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v}_p \\ \dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\theta})\boldsymbol{\omega} \end{cases} \quad (18)$$

$$\dot{\mathbf{v}}_p = \mathbf{g} + \frac{1}{m}(\sum_{i=1}^4 \mathbf{R}_{WB}\lambda_{EE_i} - \mathbf{R}_{WB}\mathbf{f}_u) \quad (19)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(-\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \sum_{i=1}^4 \mathbf{r}_{EE_i} \times \lambda_{EE_i} - \mathbf{t}_u) \quad (20)$$

$$\dot{\boldsymbol{\theta}}_j = \boldsymbol{\xi}_j \quad (21)$$

$$\dot{\boldsymbol{\theta}}_j = \boldsymbol{\xi}_j \quad (22)$$

where $\mathbf{p}, \boldsymbol{\theta} \in \mathbb{R}^3$ are the position of the robot center of mass (CoM) in world frame $\{W\}$ and the base Euler angles, respectively. $\mathbf{T}(\boldsymbol{\theta})$ is the mapping from the base angular velocity $\boldsymbol{\omega}$, expressed in base frame $\{B\}$, and the Euler angles time derivative $\dot{\boldsymbol{\theta}}$. \mathbf{I} and m are the moment of inertia about the CoM and the mass of the robot, respectively. \mathbf{r}_{EE_i} is the position of the foot i with respect to the CoM, and $\boldsymbol{\theta}_j$ are the legs' joint positions. The MPC control input is given by $\mathbf{u} = (\boldsymbol{\xi}_j, \lambda_{EE_i})$, where $\boldsymbol{\xi}_j$ are the legs' joint velocities and λ_{EE_i} is the reference contact force for the i th-foot, expressed in base frame. The kino-dynamic model described in Eqs. (18)–(22) is composed of a free-floating, single rigid-body model and a kinematic model for the legs' joints. Here, we concentrate on the problem of tracking a desired pose (center of mass position and base orientation) for the single rigid-body subsystem (Eqs. (18)–(21)) which has the properties described in Sec. III-A. Let $\mathbf{q} := (\mathbf{p}, \boldsymbol{\theta}) \in \mathbb{R}^6$ and $\mathbf{v} := (\mathbf{v}_p, \boldsymbol{\omega}) \in \mathbb{R}^6$, respectively. We define an *adaptive wrench* (combined force and torque) acting on the robot base:

$$\begin{bmatrix} \mathbf{f}_u \\ \mathbf{t}_u \end{bmatrix} = \mathbf{Y}_u(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}})\boldsymbol{\pi}_u = \mathbf{Y}_u^{in}(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}})\boldsymbol{\pi}_u^{in} + \mathbf{Y}_u^f(\mathbf{q}, \mathbf{v})\boldsymbol{\pi}_u^f \quad (23)$$

In Eq. (23), $\boldsymbol{\pi}_u^f \in \mathbb{R}^6$ includes constant forces and torques acting on the robot, while $\boldsymbol{\pi}_u^{in} \in \mathbb{R}^{10}$ is a vector of unknown inertial parameters (a combination of the mass of the unknown payload, center of mass and inertia). \mathbf{Y}_u^{in} can be derived

through the standard procedure for the Slotine-Li regressor [32], and distinguishing between the robot nominal parameters π_n^{in} and those of the payload π_u^{in} . Since a legged robot is under-actuated in many walking scenarios [27], the minimization described in Sec. III-C is not perfectly achievable for the full 6-dimensional set of generalized coordinates. Thus, here we impose it only for the translational part of the floating-base model. This is equivalent to designing the cost function so that it equally distributes the robot weight among the feet in contact, such that: $\|\sum_{i=1}^4 \mathbf{R}_{WB} \boldsymbol{\lambda}_{EE_i} - m\mathbf{v}_p^r + mg - \mathbf{R}_{WB} \mathbf{f}_u\| \approx 0$.

Eqs. (7c)–(7e) include end-effector velocity constraints for swing and stance legs, as well as friction cone constraints for the feet in contact; we refer the reader to [27] for a more detailed description of the nominal model and constraints.

As described in Sec. III-A, we make the assumption that the parametric uncertainty $\mathbf{Y}_u \pi_u$ is in a matched form. In practice, for a quadrupedal robot this requires that the robot has at least three feet in contact, or that the disturbance $\mathbf{Y}_u \pi_u$ acts along a controllable direction (for instance, during a trotting gait, only the angle around the line of the two contact points is not controllable).

B. Sliding surface for pose control

The implementation of the adaptive Lyapunov constraint requires the definition of the composite error $\boldsymbol{\sigma} := (\boldsymbol{\sigma}_l, \boldsymbol{\sigma}_o)$ for the floating-base linear and rotational dynamics. For the linear part, we can simply use $\boldsymbol{\sigma}_l := \tilde{\mathbf{v}}_p + \Lambda_l \tilde{\mathbf{p}}$. Indeed, the position error $\tilde{\mathbf{p}}$ evolves on the surface $\boldsymbol{\sigma}_l = \mathbf{0}$ with an asymptotically stable dynamics. As discussed in [30], the implementation of an adaptive controller for the rotational dynamics requires the choice of a suitable rotation error. Here, we use the quaternion error as defined in [32]:

$$\mathbf{e}_o := \eta \boldsymbol{\epsilon}_d - \eta_d \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_d^\times \boldsymbol{\epsilon} \quad (24)$$

where $\mathcal{Q} = (\eta, \boldsymbol{\epsilon})$, $\mathcal{Q}_d = (\eta_d, \boldsymbol{\epsilon}_d)$ are the base actual and desired quaternions, respectively, and $(\cdot)^\times$ is the skew operator. As demonstrated in [32], if we define $\boldsymbol{\sigma}_o := \tilde{\boldsymbol{\omega}} + \Lambda_o \mathbf{e}_o$, we have that $\mathbf{e}_o \rightarrow \mathbf{0}$ on the surface $\boldsymbol{\sigma}_o = \mathbf{0}$.

C. Conversion to torque commands

For a robotic arm, the MPC control torques $\boldsymbol{\tau}$ can be directly commanded to the robot (Fig. 2). However, for a floating-base system, a step is needed to convert MPC optimal control inputs to torque commands. Here, desired contact forces, along with desired accelerations obtained from the forward simulation of Eqs. (20), (21) are tracked by a hierarchical whole-body QP controller [33], where the learned dynamics $\mathbf{Y}_u \hat{\pi}_u$ is also compensated to take care of the model mismatch.

V. RESULTS

In this section, we validate the proposed approach in simulation and hardware tests on the quadrupedal robot ANYmal (Fig. 1). A video showcasing the results accompanies this letter ¹. For the underlying MPC computations, we use a

Multiple Shooting algorithm provided by the OCS2 toolbox [34]. The problem is formulated as an OCP for switched systems [15], where the CLF constraint in Eq. (7f) depends on different contact conditions and is thus affected by the transitions between subsystems. During the hardware tests, all computations run on a single on-board PC (Intel i7-8850H, 2.6 GHz, hexa-core 64-bit) with the MPC solver running at 100 Hz with a time horizon of 1 s and the whole-body QP controller and state estimation running at 400 Hz. In all the proposed tests, we assume that the regressor related to the unknown parameters is as in (23). Unless otherwise stated, the uncertainties include both un-modeled inertial parameters $\pi_u^{in} \in \mathbb{R}^{10}$, and constant disturbance forces applied on the base $\mathbf{f}_u^f \in \mathbb{R}^3$. These are either due to forces generated on purpose in the simulation environment (Sec. V-B), to the static friction torque present in the actuators (Sec. V-C), or to the static/dynamic force between the ground and the object that the robot carries (Sec. V-D).

A. Comparative analysis

We perform a comparative analysis of different methods which are feasible to be applied on the high-dimensional model of a quadrupedal robot. The analysis is conducted in a physics simulation, where we can precisely quantify the amount of model mismatch. We assume that the MPC model under-estimates the actual ANYmal mass of 20 kg ($\approx 40\%$ of the nominal mass), and that the true center of mass is displaced by 0.3 m along the base x-direction from the center of mass of the MPC model. We test the following five baselines:

- 1) ACLF-MPC
- 2) ACLF-MPC without terminal cost
- 3) CLF-MPC
- 4) Perfect-model MPC with terminal cost
- 5) Perfect-model MPC without terminal cost

The controllers 1) and 2) correspond to the presented method, tested with and without a terminal cost. This is set equal to the value function of an LQR, obtained from the linearization of the original problem. Indeed, although the terminal cost has been presented as part of the proposed method in Sec. III, here we show that the CLF constraint helps reducing the controller dependency on such terminal component. The CLF-MPC without adaptive ability is tested in 3), where the stability constraint is obtained from the CLF of the nominal model.

In terms of tracking performance, ideal results are attainable with an MPC problem that is perfectly aware of the model mismatch. Baselines 4) and 5) correspond to this ideal case, which does not include a CLF constraint and where the model mismatch is known to the controller.

All the methods employ the same intermediate cost function, which was tuned so that the best performance is obtained in the nominal case. For the ACLF-MPC, the adaptation gain matrix is chosen as $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\Gamma}_m, \boldsymbol{\Gamma}_{com}, \boldsymbol{\Gamma}_I)$, where $\boldsymbol{\Gamma}_m = 5.0$, $\boldsymbol{\Gamma}_{com} = I_{3 \times 3}$, and $\boldsymbol{\Gamma}_I = 0.01 I_{6 \times 6}$ relate to the uncertainties in the robot mass, combined center of mass and rotational inertia, respectively. The initial values of the estimated parameters are set to zero. The tuning gains $\boldsymbol{\Lambda}_l, \boldsymbol{\Lambda}_o$ were chosen as $\boldsymbol{\Lambda}_l = \boldsymbol{\Lambda}_o = 5I_{3 \times 3}$, while $\boldsymbol{K}_D = \text{diag}(50, 50, 50, 80, 80, 80)$. During the

¹Available at <https://youtu.be/Gu2mfAAvT0A>.

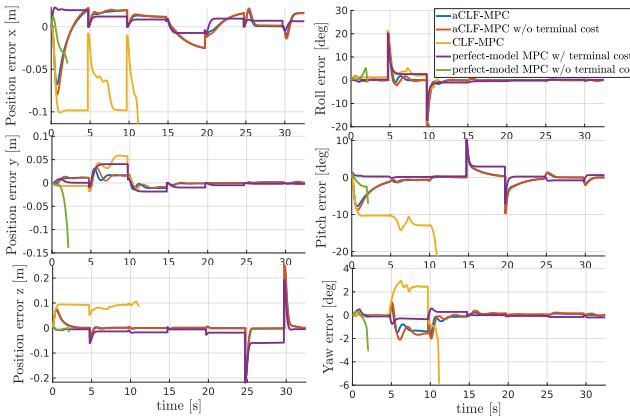


Figure 4: Position and rotation errors for a simulation test with a quadrupedal robot under the five baseline MPC controllers introduced in Sec. V-A, with a time horizon of 0.5 s.

Table I: Linear and angular RMSE errors for a simulation test with a quadrupedal robot under the five baseline methods described in Sec. V-A, with a time horizon of 1 s.

Method	1	2	3	4	5
Linear RMSE [m]	0.036	0.036	0.10	0.04	0.04
Angular RMSE [deg]	3.23	3.32	10.2	3.12	3.14

simulation, we command the robot to stand while following a desired center of mass position and base orientation trajectory. The reference trajectory consists of a sequence of step references for the center of mass vertical component, and the base pitch and roll angles. To validate the advantages that the ACLF-MPC also provides with respect to the prediction horizon tuning, we repeat the same test under a time horizon of 1 s and 0.5 s, respectively. The root mean square linear and rotational errors (RMSE) are reported in Tab. I and II.

With a time horizon of 1.0 s, the ACLF-MPC formulation performs comparatively well as the perfect-model MPC controller. In particular, the position tracking error even slightly improves with the proposed formulation with respect to the theoretical perfect-model baseline. In contrast, the ACLF-MPC outperforms the CLF-MPC that does not use any adaptation mechanism.

Interesting conclusions can be drawn from the behaviour of the five MPC controllers under a smaller time horizon (Fig. 4 and Table II). In fact, the system becomes unstable under the perfect-model MPC controller that does not use a terminal cost. In addition, the CLF-MPC problem without adaptation becomes unfeasible due to the planned contact forces not adapting to the unknown payload distribution. On the contrary, the ACLF-MPC controller determines a closed-loop stable behavior even without the need for a terminal cost.

B. Comparison with an online adaptation method

As an additional baseline comparison, we consider an adaptive method that estimates external forces and torques exerted on the robot base according to a generalized momentum approach, based on [20]. As in the proposed method, the estimated wrench is compensated online in the MPC dynamics.

Table II: RMSE for a simulation test with a quadrupedal robot under the five baseline MPC controllers described in Sec. V-A, with a time horizon of 0.5 s. A bar - indicates that the controller failed.

Method	1	2	3	4	5
Linear RMSE [m]	0.035	0.037	-	0.04	-
Angular RMSE [deg]	3.16	3.56	-	2.62	-

Such an approach is purely based on system identification, while our method is driven by the convergence of the pose tracking error. We consider a Gazebo simulation scenario where the robot walks up a slope with a statically stable gait, while it is subject to a constant disturbance force F acting in the opposite direction to the robot motion. Here, walking on an inclined surface is proposed as a representative scenario where the base orientation tracking has a relevant influence on the balance of the robot. We gradually increase the value of the constant external force to validate which of the two methods can tolerate the largest amount of un-modeled disturbance. We find that, under the momentum observer method, the robot loses its balance along the slope when $F \geq 70N$, while the robot remains stable under the proposed approach. In fact, the momentum observer approach does not include the CLF constraint in the MPC problem. Consequently, the contact forces are computed based only on the cost function minimization, which tends to distribute the forces equally among the feet in contact. In a scenario where an external force is applied on the robot in the direction opposite to the motion, a torque around the base is generated, leading to a larger orientation error. On the other hand, the ACLF-MPC method explicitly adds an additional tracking requirement, which is embedded in the Lyapunov constraint. This in turn allows the robot to maintain balance while walking on the inclined surface.

C. Base tracking under heavy bricks

We also validate how the ACLF-MPC formulation performs in a set of real-world hardware experiments. Here, we place two heavy bricks on top of the robot; the weight of each brick is 5.43 kg. Moreover, we add a third brick of 3.4 kg at random positions on top of the base, in such a way as to cause a perturbation to the center of mass of the system. An additional source of modeling errors comes from the effects that the payload has on the dynamics of legs' joints, and that is neglected in the MPC kino-dynamic model. Initially, we command the robot to stand on four feet while keeping a desired constant base position and orientation. We repeat this experiment under the baseline nominal MPC controller (i.e., based on the robot nominal model), and the ACLF-MPC approach. As a compact measure of the tracking performance, in Fig. 5 we show the constraint h_{clf} from Eq. (7f). Under the nominal MPC controller, h_{clf} settles at a non-zero value. On the contrary, under the proposed controller, the value of the constraint converges to 0. This implies that the tracking error converges to the surface $\sigma = 0$. As described in Sec. III, convergence of the tracking error is achieved by the combination of a soft inequality constraint, and by compensating

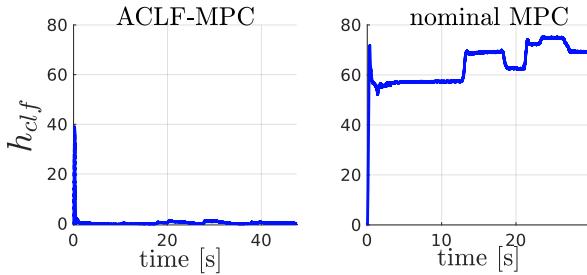


Figure 5: CLF constraint for the experiment described in Sec. V-C. The plots refer to two different tests where the robot is controlled with the ACLF-MPC approach (on the left), and with the nominal MPC (on the right).

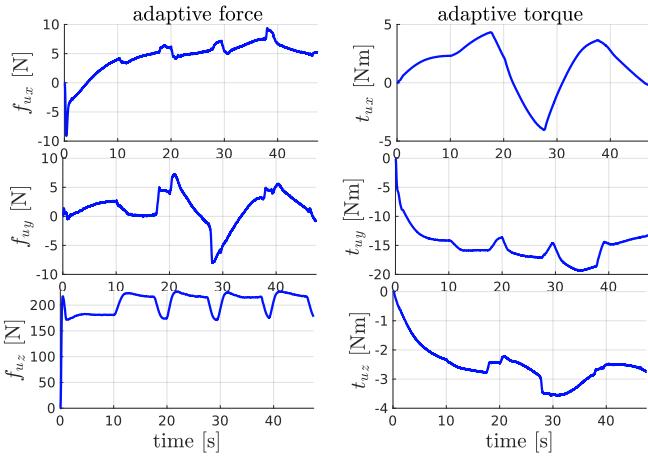


Figure 6: Adaptive force (on the left) and torque (on the right) for an experiment where the robot holds two un-modelled bricks, while the system center of mass is modified by placing a third brick at different positions on the base.

the adaptive wrench $\mathbf{Y}_u \hat{\pi}_u = (\mathbf{f}_u, \mathbf{t}_u)$ in the MPC dynamics (Eq. (10)). The latter is displayed in the plots of Fig. 6. These show how all the components of the adaptive force and torque contribute to compensate for the center of mass perturbation.

As shown in the attached video, we also send position and orientation references to the robot during walking. We repeat such experiment with two bricks on top of the robot, under the nominal MPC controller and the ACLF-MPC controller, respectively. Under the nominal MPC controller, the robot exhibits an unbalanced behavior, due to not being able to track the desired pose commands (especially with respect to the z position and pitch angle). On the other hand, it can be seen from the video that, under the proposed formulation, the robot walks stably.

D. Box pulling

In this experiment, we are interested in making the robot pull a heavy box. We repeat the same experiment twice for two different weights of the box. In the first experiment, the box weighs 16.17 kg. In the second test, we make the task more challenging by increasing the weight up to 21.6 kg. We repeat both the experiments under the ACLF-MPC method and the nominal MPC. Base rotation and center of mass position errors

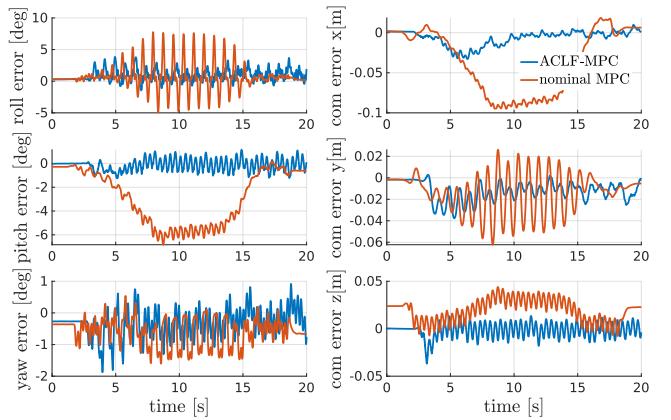


Figure 7: Base rotation and center of mass position errors for an experiment where the robot pulls an un-modelled box of 16.17 kg.

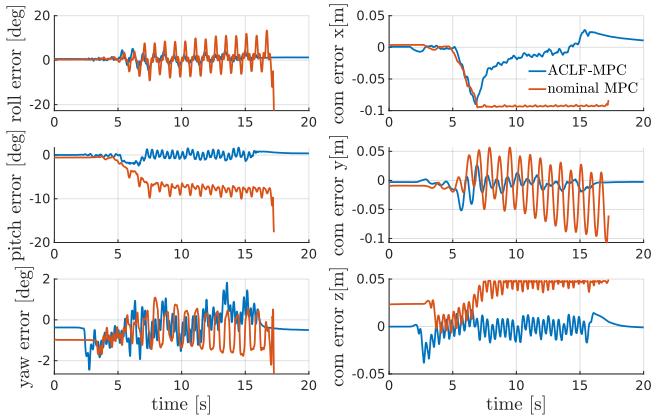


Figure 8: Base rotation and center of mass position errors for an experiment where the robot pulls an un-modelled box of 21.03 kg.

are displayed in Fig. 7 and 8, for the scenarios with the lighter box and heavier box, respectively. During the experiments, the desired position is computed by integrating a desired velocity of approximately 0.2 m/s, which is generated with a joystick. For safety reasons, position and orientation commands are reset whenever the errors grow larger than a threshold (for instance, for the center of mass position error along the base x-direction, the threshold is 0.1 m). As visible from Fig. 7 and 8, the proposed formulation succeeds in improving the base tracking, and as a consequence the robot is able to pull the box forward. Also, it helps to remove oscillations that could compromise the stability of the system. This is particularly visible from the plots displaying the roll angular error and the y position error. When pulling the heavier box (Fig. 8) such oscillations are especially strong and make the robot unstable. In Fig. 9, we also show the adaptation of the wrench translational component with respect to the different friction forces that the robot experiences in the two scenarios.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an optimal approach that unifies MPC with the online adaptation and the global stability con-

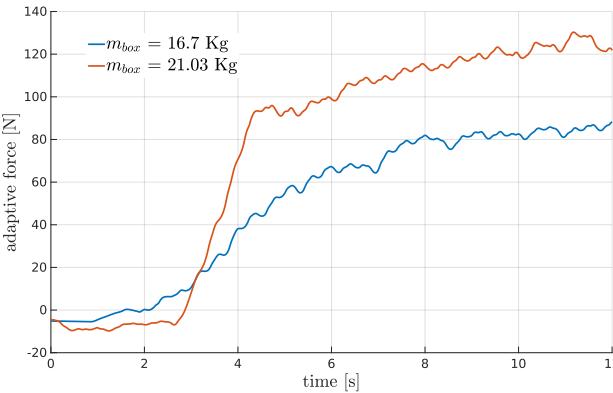


Figure 9: Adaptive force $Y_u \hat{\pi}_u$ for two experiments where the robot pulls two heavy boxes.

ditions derived in adaptive control [2]. We described a general formulation and an implementation for floating-base systems interacting with objects of unknown dynamic properties. We performed a number of simulations and hardware experiments on a quadrupedal robot, that demonstrated the effectiveness and the necessity of the proposed formulation.

Our implementation on a quadrupedal robot can handle external payloads and unknown constant wrenches applied on the base. In future work, we want to extend the method to adapt to extra rigid loads on the legs. In addition, a possible extension of this work would be to consider higher degree-of-freedom manipulators performing highly dynamic motions, such as catching objects, where unknown loads could be attached to any link, and the non-linear effects of more than one body would need to be adapted.

REFERENCES

- [1] J.-J. E. Slotine, W. Li, et al., *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.
- [2] 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.
- [3] N. M. Boffi and J.-J. E. Slotine, “Implicit regularization and momentum algorithms in nonlinearly parameterized adaptive control and prediction,” *Neural Computation*, vol. 33, no. 3, pp. 590–673, 2021.
- [4] S. M. Richards, N. Azizan, J.-J. Slotine, and M. Pavone, “Adaptive-Control-Oriented Meta-Learning for Nonlinear Systems,” in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.
- [5] G. Joshi, J. Virdi, and G. Chowdhary, “Asynchronous deep model reference adaptive control,” *Conference on Robot Learning*, 2020.
- [6] Y. Chen and Q. Nguyen, “Adaptive force-based control for legged robots,” *arXiv preprint arXiv:2011.06236*, 2020.
- [7] 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.
- [8] M. Krstić and P. V. Kokotović, “Control lyapunov functions for adaptive nonlinear stabilization,” *Systems & Control Letters*, vol. 26, no. 1, pp. 17–23, 1995.
- [9] J. Moore and R. Tedrake, “Adaptive control design for underactuated systems using sums-of-squares optimization,” in *2014 American Control Conference*. IEEE, 2014, pp. 721–728.
- [10] A. J. Taylor and A. D. Ames, “Adaptive safety with control barrier functions,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1399–1405.
- [11] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A.-a. Agha-mohammadi, and E. A. Theodorou, “Bayesian learning-based adaptive control for safety critical systems,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4093–4099.
- [12] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhwani, “Learning stability certificates from data,” *Conference on Robot Learning*, 2020.
- [13] B. T. Lopez and J.-J. E. Slotine, “Adaptive variants of optimal feedback policies,” *arXiv preprint arXiv:2104.02709*, 2021.
- [14] Y. Ding, A. Pandala, and H.-W. Park, “Real-time model predictive control for versatile dynamic motions in quadrupedal robots,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8484–8490.
- [15] F. Farshidian, E. Jelavic, A. Satapathy, M. Gifthalter, 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.
- [16] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, “Safe and fast tracking on a robot manipulator: Robust mpc and neural network control,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.
- [17] R. Sinha, J. Harrison, S. M. Richards, and M. Pavone, “Adaptive robust model predictive control with matched and unmatched uncertainty,” *arXiv preprint arXiv:2104.08261*, 2021.
- [18] C. Ding, L. Zhou, X. Rong, and Y. Li, “Online center of mass detection for quadruped robots in trot gait motion,” in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 2140–2145.
- [19] G. Tournois, M. Focchi, A. Del Prete, R. Orsolino, D. G. Caldwell, and C. Semini, “Online payload identification for quadruped robots,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4889–4896.
- [20] 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.
- [21] Y. Sun, W. Ubellacker, W. Ma, X. Zhang, C. Wang, N. Csomay-Shanklin, M. Tomizuka, K. Sreenath, and A. Ames, “Online learning of unknown dynamics for model-based controllers in legged locomotion,” *IEEE Robotics and Automation Letters*, 2021.
- [22] M. V. Minniti, R. Grandia, K. Fäh, F. Farshidian, and M. Hutter, “Model predictive robot-environment interaction control for mobile manipulation tasks,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1651–1657.
- [23] J. A. Primbs, V. Nevistic, and J. C. Doyle, “A receding horizon generalization of pointwise min-norm controllers,” *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 898–909, 2000.
- [24] J. Yu, A. Jadbabae, J. Primbs, and Y. Huang, “Comparison of nonlinear control design techniques on a model of the caltech ducted fan,” *Automatica*, vol. 37, no. 12, pp. 1971–1978, 2001.
- [25] R. Grandia, A. Taylor, A. Singletary, M. Hutter, and A. Ames, “Nonlinear Model Predictive Control of Robotic Systems with Control Lyapunov Functions,” in *Proceedings of Robotics: Science and Systems*, Corvallis, Oregon, USA, July 2020.
- [26] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.
- [27] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, “Feedback mpc for torque-controlled legged robots,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4730–4737.
- [28] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, “Episodic learning with control lyapunov functions for uncertain robotic systems,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6878–6884.
- [29] A. D. Ames and M. Powell, “Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs,” in *Control of Cyber-Physical Systems*. Springer, 2013, pp. 219–240.
- [30] P. Culbertson, J.-J. Slotine, and M. Schwager, “Decentralized adaptive control for collaborative manipulation of rigid bodies,” *IEEE Transactions on Robotics*, pp. 1–15, 2021.
- [31] X. Lu, M. Cannon, and D. Koksal-Rivet, “Robust adaptive model predictive control: Performance and parameter estimation,” *International Journal of Robust and Nonlinear Control*, 2019.
- [32] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [33] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, “Perception-less terrain adaptation through whole body control and hierarchical optimization,” in *Humanoids*, 2016, pp. 558–564.
- [34] “OCS2: An open source library for optimal control of switched systems,” [Online]. Available: <https://github.com/leggedrobotics/ocs2>.