Whole-body MPC for highly redundant legged manipulators: experimental evaluation with a 37 DoF dual-arm quadruped

Ioannis Dadiotis, Arturo Laurenzi and Nikos Tsagarakis

Abstract-Recent progress in legged locomotion has rendered quadruped manipulators a promising solution for performing tasks that require both mobility and manipulation (locomanipulation). In the real world, task specifications and/or environment constraints may require the quadruped manipulator to be equipped with high redundancy as well as whole-body motion coordination capabilities. This work presents an experimental evaluation of a whole-body Model Predictive Control (MPC) framework achieving real-time performance on a dual-arm quadruped platform consisting of 37 actuated joints. To the best of our knowledge this is the legged manipulator with the highest number of joints to be controlled with real-time whole-body MPC so far. The computational efficiency of the MPC while considering the full robot kinematics and the centroidal dynamics model builds upon an open-source DDPvariant solver and a state-of-the-art optimal control problem formulation. Differently from previous works on quadruped manipulators, the MPC is directly interfaced with the low-level joint impedance controllers without the need of designing an instantaneous whole-body controller. The feasibility on the real hardware is showcased using the CENTAURO platform for the challenging task of picking a heavy object from the ground. Dynamic stepping (trotting) is also showcased for first time with this robot. The results highlight the potential of replanning with whole-body information in a predictive control loop.

I. INTRODUCTION

Model Predictive Control (MPC) has proved to be a powerful strategy for addressing the motion planning and control problem of quadrupeds, which have now achieved a basic level of autonomy in real world [1]. The underlying reason of deploying MPC is the intuitiveness to formulate the problem as a constrained model-based optimal control problem (OCP) that is continuously solved over a receding finite time horizon. Optimizing over the time horizon enables anticipating future expected events (e.g. make/break of contacts) and continuous replanning compensates for model errors and environment changes, while satisfying the imposed constraints. Recently there has been an increasing interest in extending these capabilities to quadruped manipulators towards enabling the execution of complex loco-manipulation tasks. Legged manipulators, in general, promise the realworld sensorimotor skills needed for interaction not only between the robot legs and the terrain to be traversed (*legged*

This work was supported from the European Union's Horizon Europe Framework Programme under grant agreement No. 101070596 (euROBIN).

All authors are with Humanoid and Human-Centered Mechatronics Research Line, Italian Institute of Technology, Genoa 16163, Italy (emails: name.surname@iit.it). Ioannis Dadiotis is also with the Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genoa, Genoa 16145, Italy.



Fig. 1. The dual-arm quadruped CENTAURO robot controlled with the proposed whole-body MPC for a variety of motions: a) trotting b) 5 kg object picking from the ground and c) large height reaching. Corresponding video: https://youtu.be/8XIAtw42010

locomotion) but between one or more mounted robotic arms and its surrounding objects (*manipulation*), as well.

Unfortunately, the promising applicability when augmenting a quadruped with one or more robotic arms comes at the cost of increased complexity, since quadruped manipulators consist of multiple kinematic chains and Degrees-of-Freedom (DoF). This increased system dimension combined with the non-linear, hybrid and floating-base dynamics render real-time MPC a hard problem, which highly depends on the MPC design specifications, e.g. the robot model and the OCP solver. This difficulty has motivated a number of approaches which plan and/or control the motion for locomotion (legged body) and manipulation (armed body) separately, therefore splitting the problem into two tractable ones [2]-[6]. In some cases the effect of manipulation planning has been considered during locomotion control but not vice versa [5]. These approaches are considerably limited when it comes to real-world tasks, which may require coordination of the robot's whole-body for leveraging the available redundancy, e.g. interacting with objects close to the limit of the robot's workspace while satisfying multiple constraints. In fact, humans' advanced mobile manipulation skills are based on whole-body coordination even for simple tasks like opening

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

a door or picking up an object from the ground.

To that end it is easily understood that a MPC capable for generating whole-body motion behaviors should consider the full robot kinematics as well as a dynamics model with as much fidelity as required by the targeted task (ideally full dynamics). In this direction, a number of works on bipeds [7]–[9], quadrupeds [10]–[13] as well as other mobile manipulators [14] have achieved real-time performance. For what concerns quadruped manipulators the work of [15] (applied, also, to [16], [17]) has achieved similar results, considering full kinematics and the centroidal dynamics, for a quadruped with an up to 6 DoF robotic arm¹. Despite this progress, whole-body MPC has not been yet evaluated for quadruped manipulators that go beyong the usual 3 DoF leg structure and a single 6 DoF mounted arm, which can be only partially explained by the limited current availability of this kind of platforms. In fact, there exist highly redundant quadrupeds (bi-manual and/or with more complex leg kinematics) but they are mainly planned and controlled through simplified model-based techniques [21] or teleoperation schemes [22], [23] that rely on instantaneous control (i.e. there is no prediction). As a result, it is not clear to what extent, in terms of real-time performance, whole-body MPC can be applied to such systems and the advantages of this method have not been deployed for a variety of applications that require higher redundancy. Such tasks can be bi-manual locomanipulation ones (where at least two arms are needed) or heavy physical interaction tasks (e.g. where higher-DoF legged articulation can establish suitable support contacts [24]). This paper consists a first step towards this direction.

The main contribution of this work is presenting the experimental results of applying whole-body MPC and achieving real-time performance to a highly redundant dual-arm quadruped. The aim is to bridge the gap between motion control of today's most redundant quadruped manipulators and current MPC capabilities. The adopted approach is inspired from [15], however the CENTAURO platform [25] used in this work consists, at its current version, of much more actuated joints. In particular, it has 37 actuated DoFs which is more than double than the 18 actuated DoFs quadruped manipulator of [15]-[17] and much higher than the 22 actuated DoFs considered in state-of-the-art MPC schemes for bipeds [8], [9]. Besides the technical details for scaling whole-body MPC to such a high-dimensional system the proposed framework, differently from [15], [16], [21], does not rely on a WBC. This is only feasible due to the whole-body information considered in the MPC formulation. The motivation for exploring an approach without a WBC arises from the fact that the latter is a separate controller with different control objective and constraints, thus, does not explicitly satisfy compliance with the optimal MPC policy, as has been argued in [26]. Finally tuning a WBC consists an additional task requiring high expertise. Summing up the contributions of this paper are:

- An implementation and experimental evaluation study using a whole-body MPC framework for a highly redundant dual-arm quadruped that considers full kinematics as well as the centroidal dynamics. To the best of our knowledge this is the most redundant quadruped manipulation platform (37 actuated DoFs) that has been ever controlled using real-time whole-body MPC.
- A MPC-based pipeline providing motion plans that are executed without an instantaneous WBC. Shown in Fig. 2, the pipeline is characterized by two closed loops: the whole-body MPC loop and the joints' impedance control loop at a lower level. The MPC is interfaced with the low-level joint impedance controllers through a low-level reference generator module.

II. PRELIMINARIES

A. The CENTAURO robot

CENTAURO robot is a human-size dual-arm quadruped entirely designed at the Italian Institute of Technology [25]. Its lower body consists of four 6 DoF legs. The first 5 leg joints can be controlled in joint position or impedance control mode while the last one is an actuated wheel, which is controlled in velocity mode. Although in this work no wheeled motion is studied, wheels are considered in the OCP so that the formulation can be easily extended by just imposing different contact constraints. The robot at its current version is equipped with two 6 DoF arms (one DoF less with respect to (wrt) the initial 7 DoF prototype). Finally, the robot has a torso yaw joint that connects the upper body with the base of the platform. The above make up a total 37 actuated DoF platform with weight 118 kg.

B. Robot model

In this paper, the approach of [15] is adopted where the full kinematics and the centroidal dynamics [27] are considered while the use of Single Rigid Body Dynamics (SRBD) model [13] is, also, evaluated. The target is to demonstrate how whole-body MPC scales to such a highly redundant robot and not a comparison of the two dynamic models (which has been already presented in [15]). We briefly revisit the modeling approach; the reader is advised to the original work for detailed presentation. The dynamic model considered for CENTAURO robot assumes that the robot has six possible contact points, four at the leg and two at the arm EEs. The Newton-Euler equations can be, therefore, expressed wrt the center of mass (CoM) as shown in Eq. (1), where $h_{com} \in \mathbb{R}^6$ is the centroidal momentum about a reference frame attached to the robot's center of mass and aligned with the world frame. r_{com,c_i} is defined as the position of the contact c_i wrt the center of mass, f_{c_i} is the corresponding contact force vector and $n_c = 6$ the number of EEs. Additionally, centroidal momentum can be expressed wrt to the joint position and velocity vectors, through the centroidal momentum matrix (CMM) $A(q) \in \mathbb{R}^{6 \times (6+n_a)}$ with $n_a = 37$, as shown in Eq. (2). The CMM $A(q) = [A_b(q) \ A_i(q)]$ consists of the base part $A_b(q)$ and the actuated joints part $A_i(q)$. Full

¹Most of the mentioned works rely on recent advances on rigid multibody dynamics software [18] as well as numerical Differential Dynamic Programming (DDP) based solvers [19], [20].



Fig. 2. The motion planning and control framework for the dual-arm quadruped CENTAURO platform. The whole-body MPC, considering the centroidal dynamics, computes optimal state and control input trajectories at 10-22 Hz which are then mapped to references for the joint impedance controllers through an inverse dynamics-based low-level reference generator module. Each MPC iteration is updated with the latest available state estimate.

kinematics are considered since the base pose $q_b \in \mathbb{R}^6$ and actuated joint positions vector $q_j \in \mathbb{R}^{n_a}$ emerge in Eq. (2).

$$\dot{\boldsymbol{h}}_{com} = \begin{bmatrix} m\boldsymbol{g} + \sum_{i=1}^{n_c} \boldsymbol{f}_{c_i} \\ \sum_{i=1}^{n_c} \boldsymbol{r}_{com,c_i} \times \boldsymbol{f}_{c_i} \end{bmatrix}$$
(1)

$$\dot{\boldsymbol{q}}_{b} = \boldsymbol{A}_{b}^{-1} \left(\boldsymbol{h}_{com} - \boldsymbol{A}_{j} \dot{\boldsymbol{q}}_{j} \right)$$
(2)

Eq. (1) and Eq. (2) describe the robot model that is considered in the MPC. By selecting the state vector of the model $\boldsymbol{x} = (\boldsymbol{h}_{com}, \boldsymbol{q}_b, \boldsymbol{q}_j) \in \mathbb{R}^{12+n_a}$ and control input vector $\boldsymbol{u} = (\boldsymbol{f}_{c_1}, ..., \boldsymbol{f}_{c_{n_c}}, \boldsymbol{\dot{q}}_j) \in \mathbb{R}^{3n_c+n_a}$ the robot model can be expressed in the state space as $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$. As mentioned above, CENTAURO robot consists of 37 actuated joints which leads to a model with state dimension $n_x = 49$ and control input dimension $n_u = 55$.

III. WHOLE-BODY MODEL PREDICTIVE CONTROL

Model Predictive Control has been a popular strategy for generating robot motions through iteratively solving an OCP. Solving the OCP at each MPC iteration consists of computing the control input that minimizes an objective function over a finite-time horizon while satisfying a number of constraints (equality, inequality, dynamics). The optimized control input is applied on the robot until the OCP of the next MPC iteration is solved and a new optimal control policy is available to be used. In mathematic terms each MPC iteration can be formulated as the optimization problem in Eq. (3).

$$\begin{cases} \min_{\boldsymbol{u}(.)} \Phi(\boldsymbol{x}(T)) + \int_0^T l(\boldsymbol{x}(t), \boldsymbol{u}(t), t) dt \\ \text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{g}_1(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = 0 \\ \boldsymbol{g}_2(\boldsymbol{x}(t), t) = 0 \\ \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \ge 0 \\ \boldsymbol{x}(0) = \boldsymbol{x}_0, \end{cases}$$
(3)

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ and $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ are the state and control input vectors, respectively, and $\dot{\mathbf{x}}(t) =$ $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ is the dynamics model described in Section II-B. The terms $l(\mathbf{x}, \mathbf{u}, t)$ and $\Phi(\mathbf{x}(T))$ are the stage and terminal cost, respectively. The terminal cost is a common practice used to mimic the "tail" of the cost of the original infinite horizon problem. The terms $g_1(x(t), u(t), t)$, $g_2(x(t), t)$, h(x(t), u(t), t), x_0 in Eq. (3) form state-input equality, state-only equality, state-input inequality and initial state constraints, respectively. In this work the Sequential Linear Quadratic (SLQ) solver, detailed in [28], [29] and available in [20] is used with selected time horizon of 1 sec. The SLQ algorithm is a DDP variation in the continuous-time domain. As common practise in real-time MPC iteration schemes [30], at each MPC iteration only the solution of the first SLQ iteration is obtained.

A. Constraints

A shown in Eq. (3), the MPC formulation considers the robot dynamics and a number of equality and inequality constraints that are handled through a projection and an augmented-Lagrangian method [31], respectively. These constraints are summarized in Table I. For the rest of this paper time dependency is omitted for simplicity.

1) Contact-related constraints: Given a user desired gait (i.e. the contact state of all EEs along the time horizon), friction cone (inequality) and contact complementarity (equality) constraints are imposed for the EEs. Differently from leg EEs the velocity of the arm EEs in contact is not constrained to zero in order to include potential interaction with movable objects. It is worth mentioning that the friction cone constraint includes a regularization term which provides as well a level of stability robustness by imposing a positive lower bound at the forces along the normal direction of the contact-related constraints is omitted, since they currently consist textbook knowledge for legged planning and control, however the reader can refer to [15] for the details.

2) Swing leg EE normal velocity: For each leg EE motion tracking along the normal direction is achieved through a state-input equality constraint, shown in Eq. (4). The horizontal components of the motion are not constrained.

$$v_{ee}^{z}(\boldsymbol{x}, \boldsymbol{u}) - v_{ee}^{ref, z} + \alpha_{p} \cdot (p_{ee}^{z}(\boldsymbol{x}) - p_{ee}^{ref, z}) = 0$$
 (4)

where $p_{ee}^{z}(\boldsymbol{x})$ and $v_{ee}^{z}(\boldsymbol{x}, \boldsymbol{u})$ are the corresponding normal components of the state-dependent and state-input-dependent position and velocity of the leg EE, respectively. The superscript $(.)^{ref}$ indicates the desired values at each time according to the desired trajectory and α_{p} is a tuned gain that is multiplied by the position error. Feedback on the position

is necessary since a velocity-only constraint would result in drifting. A constraint of this type can be as well used for tracking the motion of the arm EEs, as well. Due to the limited space only one of the showcased arm EE motions in Section V is evaluated using such kind of constraints while for the rest the MPC formulation makes use of cost terms to track the arm EE motion, as described in Section III-B.

B. Cost function

The cost function for the OCP is selected to be:

$$l(\boldsymbol{x}, \boldsymbol{u}) = ||\boldsymbol{x} - \boldsymbol{x}^{ref}||_{\boldsymbol{Q}}^2 + ||\boldsymbol{u} - \boldsymbol{u}^{ref}||_{\boldsymbol{R}_j + \boldsymbol{R}_{ts}}^2 + ||\boldsymbol{p}_{ee}(\boldsymbol{x}) - \boldsymbol{p}_{ee}^{ref}||_{\boldsymbol{Q}_{e_P}}^2 + ||\boldsymbol{e}_O(\boldsymbol{x})||_{\boldsymbol{Q}_{e_O}}^2 + l_{sc}(\boldsymbol{x})$$
(5)

where $||\cdot||_{H}^{2}$ denotes the weighted (with weight matrix H) squared L_2 norm. The state is regularized with weight matrix Q while the weight matrix related with the control input consists of a positive definite sum of two terms $R_j + R_{ts}$. The matrix $R_j \in \mathbb{R}^{n_u \times n_u}$ is a positive definite weight matrix for regularizing all the control inputs while $R_{ts} \in \mathbb{R}^{n_u \times n_u}$ is related with task specifications, as described below.

1) Leg end-effector velocity: \mathbf{R}_{ts} is selected such that the task space (cartesian) velocity of the leg EEs wrt the base link are penalized. This is achieved by mapping a weight matrix $\mathbf{R}_{leg,ee} \in \mathbb{R}^{3n_{c,leg} \times 3n_{c,leg}}$ from the cartesian space into the joint space through the stacked Jacobians of the leg EEs wrt the base $\mathbf{J}_{c,leg} \in \mathbb{R}^{3n_{c,leg} \times (n_{c,leg} \cdot n_{a,leg})}$, as shown in Eq. (6). This input cost affects only the leg joints, hence the zero submatrices corresponding to the forces and upper body joints. The number of leg EEs is $n_{c,leg} = 4$ with $n_{a,leg} = 6$ actuated joints per leg. The jacobians are computed at the nominal joint configuration which is included in \mathbf{x}^{ref} .

$$\boldsymbol{R}_{ts} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_{c,leg}(\boldsymbol{x}^{ref})^T \boldsymbol{R}_{leg,ee} \boldsymbol{J}_{c,leg}(\boldsymbol{x}^{ref}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \quad (6)$$

The cost term related with the leg EE cartesian velocities is particularly important for stepping motions, since large velocities can create large impacts at leg touch-down. The horizontal components are as well penalized since they can produce lateral disturbance to the robot. This is demonstrated in Fig. 3 with data from a dynamic trotting motion in simulation. In this simulation the robot was not able to continue stepping without the proposed cost term.

TABLE I CONSTRAINTS INCLUDED IN THE MPC FORMULATION

| constraint | Туре | Complexity | |
|------------------------------|----------------------|------------|--|
| robot dynamics | state-input equality | nonlinear | |
| friction cone | input inequality | nonlinear | |
| zero force at free-motion | input equality | linear | |
| contact leg EE zero velocity | state-input equality | nonlinear | |
| swing leg EE normal velocity | state-input equality | nonlinear | |
| arm joint position limits | state inequality | linear | |



Fig. 3. Cartesian velocity of a leg EE (rear right) with (red) and without (blue) the proposed cost term during a dynamic trotting motion in simulation. Penalizing cartesian leg EE velocities results in reducing impacts during leg touch-down. The shaded regions correspond to leg in contact.

2) Arm EE pose tracking: The weights Q_{e_P} , $Q_{e_O} \in \mathbb{R}^{3\times3}$ penalize the position and orientation error between the arm EE frame and the desired pose at each time instant, $p_{ee}(x) - p_{ee}^{ref}$ and $e_O(x)$ respectively. The orientation error is computed as follows [32]:

$$\boldsymbol{e}_O(\boldsymbol{x}) = \eta_e(\boldsymbol{x})\boldsymbol{\epsilon}_{ref} - \eta_{ref}\boldsymbol{\epsilon}_e(\boldsymbol{x}) - \boldsymbol{S}(\boldsymbol{\epsilon}_{ref})\boldsymbol{\epsilon}_e(\boldsymbol{x})$$
(7)

where $Q_e(x) = \{\eta_e(x), \epsilon_e(x)\}, Q_{ref} = \{\eta_{ref}, \epsilon_{ref}\}$ are unit quaternions of the achieved and desired orientation, respectively, and $S(\cdot)$ the skew-symmetric matrix operator.

3) Self-collision avoidance: The term $l_{sc}(x)$ in Eq. (5) is a state cost for self-collision avoidance which is particularly important for robots with multiple kinematic chains. The advantage of such a consideration within a MPC, in contrast to instantaneous approaches, is that collision-free motions are optimized along the receding horizon thus self-collision distances are predicted. In this work, the approach of [33] is used where the cost is the sum of relaxed barrier function evaluations, one for each pair of links to be considered. This way the cost is acting as a soft constraint; more details can be found in the corresponding publication. For CENTAURO multiple link pairs are considered, mainly between the arm links and the robot's torso, base link as well as the front leg on the same side (e.g. between left arm and left leg). The MPC formulation considers 22 link pairs for each arm and the rest of the robot body, as well as the EE links of the two arms. This makes up a total number of 45 link pairs.

The terminal cost $\Phi(\boldsymbol{x}(T))$ of Eq. (3) includes only a term related to arm EE tracking (similar to the one described above) to make sure that the MPC does not continuously postpone tracking until after the end of the receding horizon.

IV. FROM MPC TO MOTOR CONTROL

This section describes how the optimal MPC state and control input policy is interfaced with the low-level joint impedance controllers of the joint actuators.

A. Low-level reference generator

The whole-body MPC of Section III is interfaced with the joint impedance controllers of the robot through a low-level reference generator. The reference generator is responsible for sampling the latest optimal MPC policy at 500 Hz (using linear interpolation where needed along the time horizon) and mapping the optimal state and control input vectors to

joint impedance controller references for the actuated joints, namely feedforward torque $\tau_j^{ff} \in \mathbb{R}^{n_a}$, joint position $q_j^{ref} \in \mathbb{R}^{n_a}$ and joint velocity $\dot{q}_j^{ref} \in \mathbb{R}^{n_a}$ references, respectively. These are then sent to the controller of Section IV-B.

The position and velocity references $oldsymbol{q}_{j}^{ref}, \ oldsymbol{\dot{q}}_{j}^{ref}$ can be directly accessed as part of the sampled optimal state and control input vectors, respectively. The feedforward torque vector τ_i^{ff} , differently from previous works that use an instantaneous WBC [15], is computed by solving the unconstrained inverse dynamics. This is efficiently done using the Recursive Newton-Euler Algorithm (RNEA), an implementation of which is available in Pinocchio library [18]. In practise, this consists a simple mapping of the MPC policies to torques without any additional feedback as in WBC. Given the generalized reference configuration $q^{ref} \in \mathbb{R}^{6+n_a}$, velocity $\dot{q}^{ref} \in \mathbb{R}^{6+n_a}$, acceleration $\ddot{q}^{ref} \in \mathbb{R}^{6+n_a}$ and external forces $oldsymbol{f}_c \in \mathbb{R}^{3n_c}$ the RNEA computes the actuation torques τ_j^{ff} that satisfy Eq. (8). $M(q) \in \mathbb{R}^{(6+n_a) \times (6+n_a)}$ is the inertia matrix, $h(q^{ref}, \dot{q}^{ref}) \in \mathbb{R}^{6+n_a}$ is the vector of centrifugal, Coriolis and gravity forces, $oldsymbol{S}_a \in \mathbb{R}^{n_a imes (6+n_a)}$ the actuated joint selection matrix and $oldsymbol{J}_c \in \mathbb{R}^{3n_c imes (6+n_a)}$ includes the Jacobians mapping external forces to generalized coordinates. The generalized acceleration vector is not available at the MPC and joint acceleration is not measurable on the actuators, thus zero generalized accelerations $\ddot{q}^{ref} =$ 0 are assumed in Eq. (8). This assumption is compensated by the feedback part of the control law in Eq. (9).

$$\boldsymbol{S}_{a}^{T}\boldsymbol{\tau}_{j}^{ff} = \boldsymbol{M}(\boldsymbol{q}^{ref})\boldsymbol{\ddot{q}}^{ref} + \boldsymbol{h}(\boldsymbol{q}^{ref},\boldsymbol{\dot{q}}^{ref}) - \boldsymbol{J}_{c}^{T}\boldsymbol{f}_{c} \quad (8)$$

B. Joint Impedance Control

In contrast to previous optimization-based work on CEN-TAURO [21], the actuated joints are controlled in joint impedance mode, described by Eq. (9) in a vector form.

$$\boldsymbol{\tau}_{j}^{ref} = \boldsymbol{\tau}_{j}^{ff} + \boldsymbol{K}_{p} \cdot (\boldsymbol{q}_{j}^{ref} - \boldsymbol{q}_{j}^{m}) + \boldsymbol{K}_{d} \cdot (\dot{\boldsymbol{q}}_{j}^{ref} - \dot{\boldsymbol{q}}_{j}^{m})$$
(9)

where q_j^m , $\dot{q}_j^m \in \mathbb{R}^{n_a}$ consist the vectors of measured joint positions and velocities. These controllers are running at 2 kHz. The feedback gain matrices K_p , $K_d \in \mathbb{R}^{n_a \times n_a}$ express stiffness and damping, respectively, at the joint level. Stiffness was tuned by trial and error to be 30, 120, 200 Nm/rad and damping 30, 60, 60 $Nm \cdot s/rad$ for the small, medium and large-size motors of CENTAURO, respectively. Stiffness is set at relatively low levels since the reference joint positions and velocities have been already considered for the generation of the feedforward torque in Eq. (8). For what concerns the velocity controlled wheel joints, only velocity references \dot{q}_j^{ref} are passed to them which are tracked using a proportional feedback at the velocity level. Thus, the control law of each wheel motor can be still described by a scalar version of Eq. (9) with $\tau_j^{ff} = 0 Nm$, $K_p = 0 Nm/rad$ with selected damping $K_d = 10 Nm \cdot s/rad$.

C. State Estimation

The state estimation module consists of an algorithm which estimates the robot base pose and twist using IMU,

joint position and velocity measurements. Upon estimating the base pose and twist, an estimate of the OCP state $\hat{x} \in \mathbb{R}^{n_x}$ is computed which includes the centroidal momentum. The computed state estimate is then passed to the wholebody MPC to be used as an initial condition in case a new MPC iteration is about to start. The state estimator needs to be aware of the contact state of the EEs and so the userselected gait command sent to the whole-body MPC is as well passed to the state estimator module.

V. EXPERIMENTS & EVALUATION

A. Implementation

The MPC is developed using the OCS2 toolbox [20] which provides an implementation of the used SLQ algorithm and automatic differentiation capabilities based on CppADCode-Gen for the derivatives of the dynamics, cost and constraints. The toolbox relies on Pinocchio [18] for the dynamics and kinematics. Synchronization mechanisms regarding the different processes that an MPC-based pipeline requires (e.g. running the solver, policy sampling, observation update) are provided. To enable deploying with Gazebo simulator and the real platform the MPC is interfaced through ROS with the XBot2 middleware [34] used for controlling CENTAURO.

B. Experiments

In this section the conducted experiments are described, which can be found in the accompanying video. These are evaluated in Section V-C. The OCP problem of each experiment is different due to the different constraints/costs of the executed task.

1) Arm end-effector free motion : The performance of the framework is firstly evaluated with a whole-body motion where the left arm EE of the robot is tracking a cartesian linear motion while all leg EEs are in contact with the ground. Position tracking for the arm EE is achieved using using two different approaches: the cost term in Eq. (5) and equality constraints similar to Eq. (4). The cost term related with EE orientation tracking is not included. The resulted motion highlights the advantage of whole-body coordination for increasing the robot's workspace since the robot reaches poses close to the ground and up to 2 m height (see Fig. 1b).



Fig. 4. Snapshots of the heavy object retrieval from ground experiment described in Section V-B.2

2) Heavy object retrieval from ground: The second experiment involves a manipulation task where the robot reaches a 5 kg object on the ground with its arm EE, grasps it (using a suitable gripper) and places it on top of a table, as shown in Fig. 4. This task is challenging since the object has a significant mass and, thus, the force exerted from the object to the arm EE has to be considered in the MPC. Moreover, both arm EE position and orientation tracking costs are included in Eq. (5). In terms of tuning, position tracking errors are assigned with a slightly higher cost wrt orientation. The used gripper has a rather large aperture and, thus, can achieve grasping of the object despite any small angular misalignment while this way the MPC can take advantage of a less stiff orientation tracking for avoiding self-collision when needed. For EE tracking equality constraints similar to Eq. (4) can be used instead of the cost terms, however this is omitted due to lack of space. The video demonstrates how the arm fastly moves towards the object and close to the robot legs without self-colliding. This natural behavior of the MPC emerges from the corresponding self-collision related state cost term while the user only sends a desired trajectory (linear interpolation between the current and target pose) without worrying for potential self-collisions.

3) Dynamic stepping: A number of dynamic stepping experiments are also presented where the robot is performing crawling and trotting without including any arm EE motion tracking. Self-collision avoidance and arm joint position limits are, also, not included to simplify the OCP. The arms remain always close to the reference configuration due to the state regularization cost. The stepping motions mainly include leg swings of 0.3 sec duration. For a CENTAURO robot, where leg distal mass is significant, this is quite challenging and is achieved for first time on the real hardware.

4) Loco-manipulation in simulation: A highly demanding motion where the robot is dynamically trotting and at the same time following a reference position trajectory with its arm EE is presented in a full-physics numerical simulator (Gazebo). This motion uses the centroidal dynamics formulation with self-collision and position tracking cost terms. Such an experiment is not presented on the real hardware but consists a straightforward extension of this work towards combining legged mobility with whole-body manipulation.

C. Evaluation

1) Computational efficiency: The MPC is first evaluated in simulation on a machine with an Intel Core i7-10700 CPU @ 2.90GHz using the same dynamics for both planning and rollout. The achieved MPC frequencies (mean values) are shown in Table II. Simulations of the arm EE free motion experiment are performed using two different approaches for tracking the arm EE position: through the position cost in Eq. (5) and three equality constraints similar to Eq. (4). For both cases the SRBD and the centroidal dynamics models as well as the computational overhead of including selfcollision avoidance and joint position limits are evaluated. Dynamic stepping motions are, as well, included in the table. As expected, including additional costs/constraints and/or

| TABLE II |
|---|
| MPC ACHIEVED FREQUENCY (HZ) IN SIMULATION - SAME DYNAMICS |
| MODEL USED FOR BOTH PLANNING AND AS ROLLOUT INSTANCE |
| (SRBD / CENTROIDAL DYNAMICS) |

| | arm EE free motion with position tracking cost | arm EE free motion with position tracking constraint | crawl | trot |
|----------------------------------|---|--|--------|--------|
| full formulation | 38.3 / 28.3 | 48.5 / 27.7 | | |
| $l_{sc} = 0$ | 44.9 / 35.0 | 58.0 / 34.3 | 34.9 / | 25.9 / |
| w/o arm joint position limits | 40.8 / 31.8 | 51.0 / 30.3 | 21.8 | 17.3 |

TABLE III MPC ACHIEVED FREQUENCY (HZ) ON THE REAL ROBOT

| arm EE free motion with position tracking constraint | arm EE free motion with position tracking cost | crawl | trot | heavy object retrieval |
|--|---|-------|------|---------------------------|
| 10.0 | 22.4 | 18.0 | 44.0 | 10.0 |

more complex dynamics in the formulation decelerates the SLQ solver which, however, manages to achieve frequencies more than 17 Hz in all cases. Difference in the achieved performance (especially for the SRBD model) can be noticed between the tracking cost and constraint cases which may be, also, explained by the fact that different tunings were needed to render the motion successful.

Next, the computational efficiency with the real robot in the loop is evaluated in real experiments. The focus in on using centroidal dynamics since it is a higher fidelity model that considers the updated inertia on each robot configuration as well as the momentum due to the joint velocities. The achieved frequencies on CENTAURO's Intel Core i9-10900K CPU @ 3.70 GHz are shown in Table III. As seen, the wholebody MPC achieves replanning at 10 Hz in the most complex scenario of heavy object retrieval (where arm EE position and orientation is controlled). For other motions this frequency can increase up to 22.4 Hz. It is noted that from all the included motions, trotting is the only one for which the MPC considers the SRBD and not the centroidal dynamics. By the time of writing this paper these have been the best results we have managed to record.

2) Low-level reference generation & joint impedance control: The low-level reference generator and joint impedance controller consist an important part of the pipeline rendering the MPC optimal policy useful for the torque controlled joints. To understand their behavior, for each joint the modulus of each torque term in Eq. (9) (i.e. torque term based on position feedback, velocity feedack and torque feedforward, respectively) is calculated as a percentage α_i of the sum of the modulus of all terms, as shown in Eq. (10).

$$\alpha_i = \frac{100 \ \% \cdot |i|}{\sum_{k \in S} |k|} \tag{10}$$

where $i \in S = \{\tau_j^{ff}, K_p \cdot (q_j^{ref} - q_j^m), K_d \cdot (\dot{q}_j^{ref} - \dot{q}_j^m)\}$. In Fig. 5 data from a single joint (the front left leg hip pitch joint) of the robot are included. The top part of the Fig. 5 shows that during an arm EE motion the feedforward torque originated from the RNEA algorithm contributes most on the total reference that is the output of the joint impedance control law. The velocity-based torque term has a significant contribution at some parts of the motion where the joints need to accelerate significantly. Since generalized accelerations are neglected at the lowlevel reference generator, the feedback terms, and mainly the velocity-related term, of the joint impedance control are compensating for them whenever high acceleration has to be achieved. The bottom part of the figure demonstrates that during a dynamic stepping motion the feedback part becomes significantly important. In particular, during the swing phases of the leg, the feedforward contribution is diminished and most of the torque reference is due to the feedback part. This is explained due to the need for achieving high acceleration on the swinging leg joints. Another reason for the significant contribution of the velocity-related term is due to the velocity jumps that characterize the lift off and touch down events.

3) Joint level tracking: Joint torque and velocity tracking of a single leg during the trotting motion, which consists the most dynamic motion achieved on the hardware, is shown in Fig. 6. Joint velocities are part of the MPC control input and, in a sense, have to be tracked adequately. The torque tracking plots demonstrate how the overall framework explores the large torque capacity of the actuators when a dynamic motion requires it, especially at the pitch joints since these are highly contributing to leg lifting and landing. Leveraging this range of joint torque capacity for dynamic stepping has not been possible using the framework of previous work [21] since, in practice, the robot was losing stability at the very first dynamic steps. Although torque limits are not explicitly satisfied at the MPC, it is worth mentioning that during the presented motions all references are within the actuation limits. Despite this, there are significant torque tracking errors at the hip pitch joint, probably due to limited torque controller bandwidth at high torque amplitudes.

4) Arm EE motion tracking: In Fig. 7 the arm EE motion tracking during the heavy object retrieval experiment is



Fig. 5. Contribution of the different torque terms (position-based feedback, velocity based feedback and feedforward torque) to the generated torque reference during free motion experiment (top) and the dynamic crawl stepping (bottom). The shaded regions denote leg in contact. Only data from the front left hip pitch joint are presented.



Fig. 6. Joint torque and velocity tracking for the front left (FL) leg joints during a trotting motion. The ankle yaw and wheel joints are omitted since they exhibit minimal motions during stepping, as can be easily understood by the leg kinematics. Shaded regions correspond to leg in contact.



Fig. 7. Reference (red) and achieved (blue) arm EE pose during the heavy object retrieval task. The shaded region denotes arm in contact with the object. The reference frame is shown in Fig. 4.

shown. The references sent to the MPC (red color) consist a linear interpolation and spherical linear interpolation (for the position and quaternion orientation, respectively) between discrete target poses defined for this task. Notice that orientation in Fig. 7 is depicted in roll-pitch-yaw angles where any rapid variation does not imply non-smooth reference. As specified in Section III, the MPC formulation was on purpose designed for slightly stricter position wrt orientation tracking, which results in larger steady state errors in orientation. Another interesting observation is that the tracking capability degrades while the robot is carrying the 5 kg object. This is, partially, expected since the model of the object is unknown. Succeeding the task only relies on the limited ability of the MPC to predict the external force exerted at the arm EE through continuously receiving state observations and replanning. However, there are significant errors in the X and Y directions and orientations since the MPC compromises tracking in this directions to avoid self-collision with the leg. Finally, there is an impact at the time of releasing the object which causes tracking errors. At the last part, the user disturbs the robot body and arm EE, as can be seen in the accompanying video. Since low stiffness gains were used at the joint impedance controller the robot behaves relatively compliant. Most importantly the plot demonstrates that the achieved MPC frequency, which has an order magnitude of tens of Hz, provides the necessary feedback for achieving EE motion tracking and constraint satisfaction.

VI. CONCLUSION AND FUTURE WORK

This work presents the experimental results of controlling the highly redundant dual-arm quadruped CENTAURO with whole-body MPC. This is the highest DoF robot that has been controlled in real-time with whole-body MPC so far. At the lower-level, the MPC plans are interfaced with the joint impedance controllers without the need of a WBC. The results indicate that the approach can be successfully applied to control the motion of highly complex robots which are, traditionally, teleoperated or controlled with simplified techniques. Real-time performance and replanning frequency of tens of Hz are achieved for a variety of complex tasks i.e. dynamic stepping and heavy object manipulation.

For future work, deploying the Riccati-like feedback gains provided by the solver could potentially increase the state feedback frequency [9], [13], [26]. In terms of targeted tasks, using MPC for manipulation of unknown environments remains an open challenge, which few works have tried to address [17], [35]. Another direction is to investigate how joint torque sensing can be considered within the proposed MPC where joint torques are not explicitly available.

REFERENCES

- M. Tranzatto *et al.*, "Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge," *arXiv preprint arXiv:2201.07067*, 2022.
- [2] C. D. Bellicoso *et al.*, "Alma-articulated locomotion and manipulation for a torque-controllable robot," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 8477–8483.
- [3] "Boston dynamics: Dynamic robot manipulation," https://youtu.be/ 2jyLalY6ubc, accessed: 2023-07-21.
- [4] M. P. Murphy *et al.*, "High degree-of-freedom dynamic manipulation," in *Unmanned Systems Technology XIV*, vol. 8387. International Society for Optics and Photonics, 2012, p. 83870V.
- [5] Y. Ma et al., "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.
- [6] R. Parosi *et al.*, "Kinematically-decoupled impedance control for fast object visual servoing and grasping on quadruped manipulators," *arXiv* preprint arXiv:2307.04918, 2023.
- [7] J. Koenemann et al., "Whole-body model-predictive control applied to the hrp-2 humanoid," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 3346–3351.
- [8] E. Dantec et al., "Whole body model predictive control with a memory of motion: Experiments on a torque-controlled talos," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 8202–8208.
- [9] E. Dantec *et al.*, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4448–4455, 2022.
- [10] M. Neunert et al., "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.
- [11] C. Mastalli et al., "Inverse-dynamics mpc via nullspace resolution," IEEE Transactions on Robotics, pp. 1–20, 2023.
- [12] A. Assirelli *et al.*, "Whole-Body MPC without Foot References for the Locomotion of an Impedance-Controlled Robot," Sep. 2022, working paper or preprint. [Online]. Available: https://hal.laas.fr/hal-03778738

- [13] R. Grandia et al., "Feedback mpc for torque-controlled legged robots," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019, pp. 4730–4737.
- [14] M. V. Minniti *et al.*, "Whole-body mpc for a dynamically stable mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3687–3694, 2019.
- [15] J.-P. Sleiman *et al.*, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.
- [16] J. Cheng et al., "Haptic teleoperation of high-dimensional robotic systems using a feedback mpc framework," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022, pp. 6197–6204.
- [17] E. Arcari et al., "Bayesian multi-task learning mpc for robotic mobile manipulation," IEEE Robotics and Automation Letters, 2023.
- [18] J. Carpentier et al., "The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in 2019 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2019, pp. 614–619.
- [19] C. Mastalli *et al.*, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 2536–2542.
- [20] F. Farshidian *et al.*, "OCS2: An open source library for optimal control of switched systems," [Online]. Available: https://github.com/ leggedrobotics/ocs2.
- [21] I. Dadiotis *et al.*, "Trajectory optimization for quadruped mobile manipulators that carry heavy payload," in 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids). IEEE, 2022, pp. 291–298.
- [22] T. Klamt *et al.*, "Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance," *Journal* of Field Robotics, vol. 37, no. 5, pp. 889–919, 2020.
- [23] "Ground demonstration of gitai's lunar robotic rover r1 in a simulated lunar environment," https://youtu.be/mZcwuypT6rU, accessed: 2023-07-21.
- [24] M. P. Polverini *et al.*, "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.
- [25] N. Kashiri *et al.*, "Centauro: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1595–1602, 2019.
- [26] C. Mastalli *et al.*, "Agile maneuvers in legged robots: a predictive control approach," *arXiv preprint arXiv:2203.07554*, 2022.
- [27] D. E. Orin et al., "Centroidal dynamics of a humanoid robot," Autonomous robots, vol. 35, pp. 161–176, 2013.
- [28] F. Farshidian *et al.*, "An efficient optimal planning and control framework for quadrupedal locomotion," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 93–100.
- [29] F. Farshidian *et al.*, "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.
- [30] M. Diehl et al., "Real-time iterations for nonlinear optimal feedback control," in Proceedings of the 44th IEEE Conference on Decision and Control. IEEE, 2005, pp. 5871–5876.
- [31] J.-P. Sleiman et al., "Constraint handling in continuous-time ddp-based model predictive control," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 8209–8215.
- [32] B. Siciliano et al., Robotics: Modelling, Planning and Control. Springer, 2008.
- [33] J.-R. Chiu et al., "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.
- [34] A. Laurenzi et al., "The xbot2 real-time middleware for robotics," *Robotics and Autonomous Systems*, vol. 163, p. 104379, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0921889023000180
- [35] M. V. Minniti *et al.*, "Model predictive robot-environment interaction control for mobile manipulation tasks," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 1651–1657.