Multi-Contact Heavy Object Pushing With a Centaur-Type Humanoid Robot: Planning and Control for a Real Demonstrator

Matteo Parigi Polverini, Arturo Laurenzi, Enrico Mingo Hoffman, Francesco Ruscelli, and Nikos G. Tsagarakis

***Abstract*—Performing a demanding manipulation task with a multi-legged loco-manipulation platform may require the exploitation of multiple external contacts with different environment surfaces for counteracting the manipulation forces. This is the case of the pushing task of a heavy object, where the grip forces at ground may not be adequate and establishing leg contacts against a wall turns out to be an effective solution to the manipulation problem. In order to produce such behaviour, this letter presents a control architecture that is able to freely exploit the environment complexity to perform loco-manipulation actions, e.g. pushing a heavy object, while meeting the implementation requirements to achieve a real demonstrator. The proposed approach, conceived for torque-controlled platforms, combines the planning capabilities of nonlinear optimization over the robot centroidal statics, based on a continuous description of the environment through superquadric functions, with the instantaneous capabilities of hierarchical inverse kinematics and reactive contact force distribution. Experimental validation has been performed on the pushing task of a wooden cabinet loaded with bricks, using the CENTAURO robot developed at Istituto Italiano di Tecnologia (IIT).**

***Index Terms*—Humanoid robots, kinematics, optimization and optimal control, redundant robots.**

I. Introduction

As HIGHLIGHTED by the DARPA Robotics Challenge (DRC), humanoids robots and loco-manipulation platforms in general, are on the way to find their natural employment in disaster scenarios. They are expected to enter environments conceived for humans and to perform dangerous and physically demanding tasks that require locomotion and manipulation capabilities. Note also that, besides the tasks staged at DRC, disaster scenarios often demand for a high level of interaction between a humanoid and the environment, where all the available end-effectors simultaneously contact different surfaces for a successful task completion. This is the case of e.g. locomotion in unstructured and cumbersome environments [1], balancing in confined spaces [2], pushing of heavy objects [3], [4]. In this respect, the problem of establishing multiple contacts with the environment in order to perform a loco-manipulation task (hereafter referred to as *multi-contact loco-manipulation*) needs to be treated from two different perspectives, hierarchically related: planning and control.

From a *motion planning* perspective, there exist two families of approaches: contacts planning and dynamics motion planning. Given the kinematic reachability of the robot and the environment complexity, the problem of finding a kinematically feasible set of contacts can be tackled either through discrete search methods or continuous optimization methods. The discrete search approaches, such as A* and RRT, applied to multi-contact planning [5], [6] have been verified in unstructured environments [7], [8] and implemented on real humanoid robots in structured indoor environments [9]. Nevertheless, they can be challenging to apply due to the difficulty of finding informative, admissible heuristics. By simplifying the environment through a set of convex regions, mixed-integer convex optimization can be used to find bipedal walking footholds or more complex multi-contact motions [10], [11]. This formulation leads to efficient solutions, depending on the number of considered support surfaces. The second family of motion planning approaches relies on the adoption of the robot dynamics model. In this respect, the approaches exploiting the full hybrid dynamics of a legged system can produce impressive results [12]–[14], but remain computationally expensive. At the other end, the methods based on simplified dynamical models [15]–[17], e.g. the *centroidal dynamics* [18], allow to speed up calculations by reducing problem dimensions and can be solved efficiently using off-the-shelf nonlinear solvers, e.g. IPOPT [19], at the expense of kinematic and dynamic consistency.

From a *control perspective*, the planned trajectories, produced by one of the aforementioned planning methods, need to be tracked at best by a suitable control layer. This layer is also responsible for the instantaneous balancing problem, which, for the case of a torque-controlled platform as the CENTAURO robot [20], entails a distribution of contact forces among the available end-effectors. In this respect, a first category of methods performs a *pre-optimization* of contact forces with respect to the robot centroidal dynamics, and subsequently maps the computed contact forces to the actuated joint torques, see [3],

Manuscript received September 9, 2019; accepted January 2, 2020. Date of publication January 13, 2020; date of current version January 27, 2020. This letter was recommended for publication by Associate Editor Y. Zhao and Editor D. Song upon evaluation of the reviewers' comments. This work was supported by the European Unions Horizon 2020 Research and Innovation Program under Grant 644839 (CENTAURO). *(Corresponding author: Matteo Parigi Polverini.)*

M. P. Polverini, E. M. Hoffman, and N. G. Tsagarakis are with the Humanoids & Human Centered Mechatronics Research Line (HHCM), Istituto Italiano di Tecnologia (IIT), 16163 Genova, Italy (e-mail: matteo.parigi@iit.it; Enrico.Mingo@iit.it; nikos.tsagarakis@iit.it).

A. Laurenzi and F. Ruscelli are with the Humanoids & Human Centered Mechatronics Research Line (HHCM), Istituto Italiano di Tecnologia (IIT), 16163 Genova, Italy, and also with the Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), Universitá di Genova, 16126 Genova, Italy (e-mail: arturo.laurenzi@iit.it; francesco.ruscelli@iit.it).

Digital Object Identifier 10.1109/LRA.2020.2965906

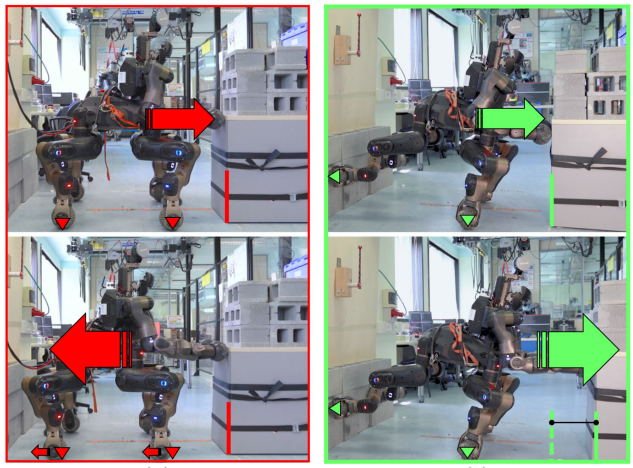


Fig. 1. CENTAURO pushing a heavy object exceeding 120 kg. If all the legs of the robot stay in contact with the ground (left-hand side), it is not possible to effectively push the object by simply imposing a forward motion with the two hands (here the robot is controlled in position mode). In fact (see bottom left figure), the robot loses grip and slides backwards, while the object keeps its position. Instead, as an outcome of the proposed control architecture, a pushing behaviour is produced with the rear legs touching a wall located behind the robot (right-hand side). This way CENTAURO can accomplish the manipulation task and effectively move the object.

[21]–[23]. In contrast, full-body inverse dynamics can be directly exploited, see e.g. [24]–[26]. The balancing problem for torque-controlled legged robots can be alternatively addressed through a *post-optimization* of contact forces [27], [28].

The present work has been driven by the realization of a demonstrator of the pushing task of a heavy object with a centaur-type humanoid. This is an example of a task that requires multi-contact loco-manipulation capabilities. As a matter of fact, see Fig. 1(a), due to the heavy weight of the object (exceeding 120 kg) combined with the ground friction coefficient, it is not possible to push the object with all legs staying on the ground. Based on this, we aim at devising a control architecture that is able to freely exploit the environment complexity, e.g. a wall located behind the robot, in order to accomplish the pushing task, see Fig. 1(b), and similar loco-manipulation actions, while meeting the requirements for actual implementation on real hardware. The proposed control architecture, conceived for torque-controlled platforms, relies on the planning capabilities of a nonlinear programming (NLP) *multi-contact planner*, combined with the reactive capabilities of a *multi-contact controller*. Besides the experimental challenge inherent in this task, the innovative contributions of this letter with respect to state-of-the-art approaches are listed in the following:

- The *multi-contact planner* belongs to the family of nonlinear planners that reason about a simplified model of the robot dynamics, as in [15]–[17], in order to reduce the problem complexity. As a novel contribution we propose to employ the robot centroidal dynamics under quasi-static assumptions. This further reduces the problem dimension, at the expense of producing quasi-static poses.
- The *multi-contact planner* adopts a continuous description of the environment complexity in order to allow for continuous optimization, as in [16], instead of mixed-integer optimization [10], [11]. We propose to use *superquadric* functions as a convenient choice, w.r.t. [16], to model perpendicular planes, like the ground/wall scenario for the pushing task, but also narrow passages or corridors. Note that, while superquadrics have been employed in robot grasping, see [29], their adoption in motion planning for legged robots represents a novel contribution of this work.
- The *multi-contact controller* is responsible for the tracking of the planned contacts and for reactive balancing, by means of hierarchical Inverse Kinematics (HIK) and instantaneous contact force distribution, respectively. Although these are conventional elements, based on the authors' works [27], [30], their combination is the result from our experimental trials on the hardware in order to bridge the gap between simulations and experiments.

II. Mathematical Background

A. Floating-Base Dynamic Model

The dynamics of a floating-base robot is expressed by the following equation:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{S}\boldsymbol{\tau} + \boldsymbol{J}_{\mathrm{C}}^{T}\boldsymbol{F}_{\mathrm{C}} \tag{1}$$

Differently from a fixed-base robot, an *actuation matrix* $\boldsymbol{S} \in \mathbb{R}^{(n+6)\times n}$ models the system under-actuation, while contact forces $\boldsymbol{F}_{\mathrm{C}} \in \mathbb{R}^{k}$ are taken into account by concatenating the Jacobian of all support links $\boldsymbol{J}_{\mathrm{C}} \in \mathbb{R}^{k\times(n+6)}$ and the corresponding overall contact wrench. Due to the wheeled-legged nature of CENTAURO, we will hereafter assume n_{C} point contacts, thus $k = 3n_{\mathrm{C}}$. We can split (1) into 6 unactuated and n actuated rows, denoted with subscript u and a, respectively.

$$\boldsymbol{B}_{u}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}_{u}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{J}_{\mathrm{C,u}}^{T}\boldsymbol{F}_{\mathrm{C}} \tag{2a}$$

$$\boldsymbol{B}_{a}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}_{a}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{J}_{\mathrm{C,a}}^{T}\boldsymbol{F}_{\mathrm{C}} \tag{2b}$$

Note that the transpose of the unactuated part of the contact Jacobian $\boldsymbol{J}_{\mathrm{C},u}^{T} \in \mathbb{R}^{6\times k}$ acts as a *grasp matrix* [31], where $\boldsymbol{J}_{\mathrm{C},u}$ encodes the relationship between the linear velocity of the contact points and the virtual joint velocities.

B. Centroidal Dynamics Model

By expressing the rate of change of the total linear and angular momentum of the system in a frame anchored at the Center of Mass (CoM), that has origin $\boldsymbol{r}_{\mathrm{CoM}} \in \mathbb{R}^{3}$ and orientation of the inertial frame, see [18], called *centroidal momentum*, the dynamics expressed in (2a) can be rewritten as:

$$\boldsymbol{A}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{A}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = m\boldsymbol{g} + \boldsymbol{G}_{\mathrm{CD}}\boldsymbol{F}_{\mathrm{C}} \tag{3}$$

This formulation is called *centroidal dynamics* (CD) [18], where $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{6\times(n+6)}$ is denoted as *centroidal momentum matrix* (CMM), which maps the velocities, and therefore momentum, of each individual body into the common reference frame at the CoM. The gravity acceleration is given by $\boldsymbol{g} \in \mathbb{R}^{3}$, while $m \in \mathbb{R}$ is the mass of the entire robot. The left-hand side of (3) denotes the rate of change of the centroidal momentum $\frac{d}{dt}(\boldsymbol{A}\dot{\boldsymbol{q}})$, which for all mechanical systems must equal the net external wrench on the system (right-hand side). Finally, the centroidal dynamics grasp matrix $\boldsymbol{G}_{\mathrm{CD}} \in \mathbb{R}^{6\times k}$ is given by:

$$\boldsymbol{G}_{\mathrm{CD}} = \begin{bmatrix} \boldsymbol{I}_3 & \cdots & \boldsymbol{I}_3 \\ \boldsymbol{S}(\boldsymbol{r}_{\mathrm{CoM}} - \boldsymbol{r}_{\mathrm{C},1}) & \cdots & \boldsymbol{S}(\boldsymbol{r}_{\mathrm{CoM}} - \boldsymbol{r}_{\mathrm{C},n_{\mathrm{C}}}) \end{bmatrix}, \tag{4}$$

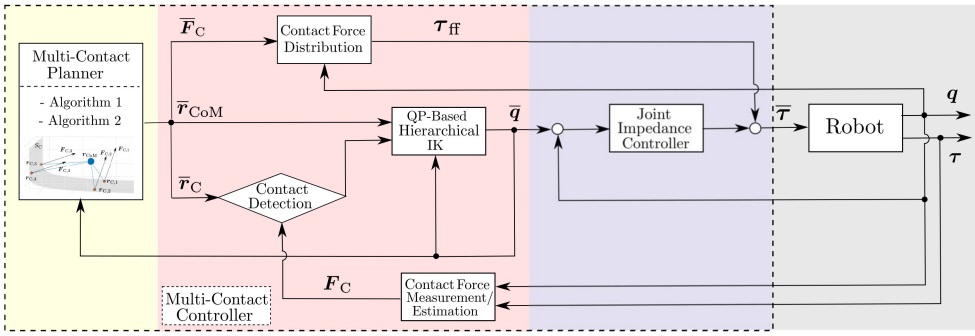


Fig. 2. Overview of the control architecture.

where $\boldsymbol{r}_{\mathrm{C,i}} \in \mathbb{R}^3$ is the i-th contact position, for $i = \{1, \ldots, n_{\mathrm{C}}\}$, while $\boldsymbol{S}$ is a skew-symmetric matrix such that $\boldsymbol{S}(\boldsymbol{a})\,\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$.

III. Proposed Control Architecture

Assuming an a-priori knowledge on the environment geometry, our goal is to propose a control architecture for multi-contact loco-manipulation tasks that is able to freely exploit the environment complexity, while meeting the requirements for implementation on the real hardware, e.g. computational load and robustness. Considering e.g. the pushing task of a heavy object with a centaur-type humanoid, where the environment consists of a wall located behind the robot, see Fig. 1, this control architecture allows to:

- produce a sequence of quasi-static poses, automatically choosing which environment surface interact with, e.g. pushing with the rear legs touching the wall, see Fig. 1(b);
- autonomously reach the planned quasi-static poses, while being robust to environment uncertainties and external perturbations.

To this end, the proposed control architecture consists of the following components, organized in a hierarchical structure, see Fig. 2 :

- *Multi-contact planner* (see Sec. IV): plans the CoM position $\overline{\boldsymbol{r}}_{CoM}$, the contact positions $\overline{\boldsymbol{r}}_{\mathrm{C}} \in \mathbb{R}^k$, and the contact forces $\overline{\boldsymbol{F}}_{\mathrm{C}} \in \mathbb{R}^k$ for a multi-contact loco-manipulation task. A nonlinear optimization is performed using the robot centroidal dynamics under quasi-static assumptions, based on a continuous description of the environment through superquadric functions.
- *Multi-contact controller* (see Sec. V): tracks at best the output of the multi-contact planning layer by producing the reference joint positions $\overline{\boldsymbol{q}}$ and a feed-forward torque $\boldsymbol{\tau}_{\mathrm{ff}}$ to the inner joint-level controller. This control layer provides robustness to the environment surface location and to external disturbances, by means of the following components:
  1) *QP-based hierarchical IK* (see Sec. V-A): tracks the planned CoM position and contact positions;
  2) *Contact force distribution* (see Sec. V-B): tracks at best the planned contact forces, while ensuring reactive balancing.
  3) *Contact detection:* detects if contact with the environment has been established, based on estimated or measured contact forces.
- *Joint-level control:* a decentralized joint impedance controller, with a torque feed-forward term, feeding the inner torque control loop.

IV. Multi-Contact Planner

Our planning approach relies on the adoption of a continuous nonlinear optimization method formulated w.r.t. the robot centroidal dynamics under quasi-static assumptions, i.e. $\dot{\boldsymbol{q}} = \boldsymbol{0}$ and $\ddot{\boldsymbol{q}} = \boldsymbol{0}$. Let $\boldsymbol{\tau}_{\mathrm{ext}} \in \mathbb{R}^6$ be a desired manipulation wrench expressed w.r.t. the CoM frame, one obtains from (3):

$$m\boldsymbol{g} + \boldsymbol{G}_{\mathrm{CD}}\boldsymbol{F}_{\mathrm{C}} + \boldsymbol{\tau}_{\mathrm{ext}} = \boldsymbol{0} \tag{7}$$

which we hereafter refer to as *centroidal statics* (CS). In doing so, we end up in an optimization problem with the following decision variables:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}_{\mathrm{CoM}} & \boldsymbol{r}_{\mathrm{C}} & \boldsymbol{F}_{\mathrm{C}} \end{bmatrix}^T \in \mathbb{R}^{2k+3} \tag{8}$$

Note that, compared to an optimization that includes the full robot dynamics (1) under the same assumptions, hence $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} & \boldsymbol{F}_{\mathrm{C}} \end{bmatrix}^T \in \mathbb{R}^{n+6+k}$, the problem dimension significantly decreases for $n \gg k$. This choice additionally circumvents the nonlinearity introduced by trigonometric functions appearing in (1). As an heuristic observation, this can be connected to a reduced occurrence of local minima inherent in continuous optimization methods compared to discrete search methods. On the other hand, the price for this reduced computation is the possibility to produce quasi-static poses only and the inability to reason about joint limits.

A. Continuous Environment Description: Superquadrics

Representing the environment with separate convex-hulls can be a useful simplification in contacts planning approaches, although it inherently requires mixed-integer optimization [10], [11]. In agreement with [16], we here consider a hard contact model for the environment based on a unique continuous function. This choice allows for continuous optimization, and consequently speeds up calculations compared to mixed-integer programming. Soft contact models can be alternatively employed, see [12], [14], although they are prone to produce physical inconsistencies, e.g. surface penetrations with contacts. In particular, we propose to employ *superquadric* functions as a convenient description of perpendicular surfaces, e.g. the ground/wall environment involved in the pushing task, but also narrow passages or corridors. While superquadrics have been recently employed in robot grasping, see [29], their adoption in motion planning for legged robots is novel. Note that, compared to [16], which employs smooth height-maps to combine different environment surfaces, superquadrics are more effective in modelling vertical walls, whereas [16] would require to consider

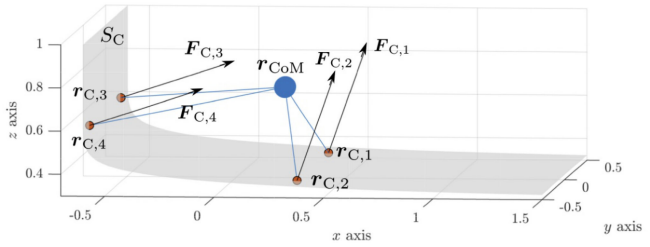


Fig. 3. Graphical representation of the quantities involved in the *multi-contact planner*, see Algorithm 1. The environment complexity, consisting of the ground and a wall (see Fig. 1), has been modeled through a single superquadric function (9) with $\boldsymbol{P} = [8\ 8\ 8]^T$.

Algorithm 1: Multi-Contact Manipulation.

Input: $\boldsymbol{\tau}_{\text{ext}}$, S_{C} - optionally: $\boldsymbol{r}_{\text{CoM}}^{\text{des}}$, $\boldsymbol{r}_{\text{C}}^{\text{des}}$
Output: $\overline{\boldsymbol{r}}_{\text{CoM}}$, $\overline{\boldsymbol{r}}_{\text{C}}$, $\overline{\boldsymbol{F}}_{\text{C}}$

1: Solve the following NLP problem:

$$\min_{\boldsymbol{x}} \|\boldsymbol{r}_{\text{CoM}} - \boldsymbol{r}_{\text{CoM}}^{\text{des}}\|^2_{2,W_{CoM}} + \|\boldsymbol{r}_{\text{C}} - \boldsymbol{r}_{\text{C}}^{\text{des}}\|^2_{2,W_r} + \|\boldsymbol{F}_{\text{C}}\|^2_{2,W_F}$$

subject to

$$m\boldsymbol{g} + \boldsymbol{G}_{\text{CD}}\boldsymbol{F}_{\text{C}} + \boldsymbol{\tau}_{\text{ext}} = \boldsymbol{0} \tag{5a}$$

$$\boldsymbol{r}_{\text{C},i} \in S_{\text{C}}(\boldsymbol{r}_{\text{C},i}) \tag{5b}$$

$$\{\boldsymbol{F}_{\text{C},i}, \boldsymbol{r}_{\text{C},i}\} \in \mathcal{F}(\boldsymbol{F}_{\text{C},i}, \boldsymbol{r}_{\text{C},i}, \mu_i) \tag{5c}$$

$$\underline{\boldsymbol{r}_{\text{C}}} \leq \boldsymbol{r}_{\text{C}} \leq \overline{\boldsymbol{r}_{\text{C}}} \tag{5d}$$

tilted planes. This, on the other end, comes at the expense of curved corners.

The general form of a superquadric function, hereafter denoted with $S_{\text{C}}(\boldsymbol{r}_{\text{C},i})$, with fully independent axis orders is given by:

$$\left|\frac{r_{\text{C,i}_x} - C_x}{R_x}\right|^{P_x} + \left|\frac{r_{\text{C,i}_y} - C_y}{R_y}\right|^{P_y} + \left|\frac{r_{\text{C,i}_z} - C_z}{R_z}\right|^{P_z} = 1 \tag{9}$$

where the superquadric center is defined by the vector $\boldsymbol{C} = \begin{bmatrix} C_x & C_y & C_z \end{bmatrix}^T$, while the superquadric axial radii are defined by the vector $\boldsymbol{R} = \begin{bmatrix} R_x & R_y & R_z \end{bmatrix}^T$. Finally, $\boldsymbol{P} = \begin{bmatrix} P_x & P_y & P_z \end{bmatrix}^T$ defines the axial order, or curvature, which determines the actual shape of the superquadric. If each element of $\boldsymbol{P}$ is greater than 2, (9) is continuously differentiable and defines a cube with smooth edges and corners, see Fig. 3. Clearly enough, the advantage of circumventing the need for mixed-integer optimization comes at a price, as there exists a trade-off between the curvature of the superquadric and the ability of the solver to find a solution. In the following, we will consider the environment superquadric function to be manually specified, assuming the surfaces locations to be known. Alternatively, the superquadric could be automatically generated from point cloud data. This remains a future research direction.

Algorithm 2: Contact-Lifting.

Input: $\boldsymbol{r}_{\text{C}}^{\text{des}}$, $\boldsymbol{n}_{\text{C}}^{\text{des}}$, lifting contact/s - optionally: $\boldsymbol{r}_{\text{CoM}}^{\text{des}}$
Output: $\overline{\boldsymbol{r}}_{\text{CoM}}$, $\overline{\boldsymbol{F}}_{\text{C}}$

1: Solve the following NLP problem:

$$\min_{\boldsymbol{x}} \|\boldsymbol{r}_{\text{CoM}} - \boldsymbol{r}_{\text{CoM}}^{\text{des}}\|^2_{2,W_{CoM}} + \|\boldsymbol{F}_{\text{C}}\|^2_{2,W_F}$$

subject to

$$m\boldsymbol{g} + \boldsymbol{G}_{\text{CD}}\boldsymbol{F}_{\text{C}} = \boldsymbol{0} \tag{6a}$$

$$\boldsymbol{F}_{\text{C,lift}} = \boldsymbol{0} \tag{6b}$$

$$\boldsymbol{r}_{\text{C,stance}} = \boldsymbol{r}_{\text{C,stance}}^{\text{des}} \tag{6c}$$

$$\{\boldsymbol{F}_{\text{C},i}, \boldsymbol{n}_{\text{C},i}^{\text{des}}\} \in \mathcal{F}(\boldsymbol{F}_{\text{C},i}, \boldsymbol{n}_{\text{C},i}^{\text{des}}, \mu_i) \tag{6d}$$

B. Coulomb Friction Cones

To encode the impact of the environment geometry on contact forces, friction constraints must be satisfied. Let us consider the contact force $\boldsymbol{F}_{\text{C},i} \in \mathbb{R}^3$ and the related contact-surface normal $\boldsymbol{n}_{\text{C,i}} \in \mathbb{R}^3$, at the i-th contact point. Being $\boldsymbol{F}_{\text{C,i}}^n \in \mathbb{R}^3$ and $\boldsymbol{F}_{\text{C,i}}^t \in \mathbb{R}^3$, the normal and tangential component of the contact force, respectively, given by:

$$\boldsymbol{F}_{\text{C,i}}^n = (\boldsymbol{F}_{\text{C,i}} \cdot \boldsymbol{n}_{\text{C,i}})\boldsymbol{n}_{\text{C,i}} \tag{10a}$$

$$\boldsymbol{F}_{\text{C,i}}^t = \boldsymbol{F}_{\text{C,i}} - (\boldsymbol{F}_{\text{C,i}} \cdot \boldsymbol{n}_{\text{C,i}})\boldsymbol{n}_{\text{C,i}} \tag{10b}$$

the i-th point contact remains in rest contact mode if $\boldsymbol{F}_{\text{C,i}}$ lies inside the friction cone directed by $\boldsymbol{n}_{\text{C,i}}$, i.e.:

$$\mathcal{F}(\boldsymbol{F}_{\text{C},i}, \boldsymbol{n}_{\text{C},i}, \mu_i) := \begin{cases} \boldsymbol{F}_{\text{C,i}} \cdot \boldsymbol{n}_{\text{C,i}} > F_{\text{thr}} \\ \|\boldsymbol{F}_{\text{C,i}}^t\|_2 \leq \mu_i(\boldsymbol{F}_{\text{C,i}} \cdot \boldsymbol{n}_{\text{C,i}}) \end{cases} \tag{11}$$

where μ_i is the Coulomb friction coefficient, while $F_{\text{thr}} \geq 0$ is a scalar force threshold. The Euclidean norm $\|\cdot\|_2$ models Coulomb friction cones with circular section.

Note that from (9), one can express the contact-surface normal $\boldsymbol{n}_{\text{C},i}$ as a function of the corresponding contact point $\boldsymbol{r}_{\text{C},i}$ through the following relation:

$$\boldsymbol{n}_{\text{C},i} = -\frac{\nabla S_{\text{C}}(\boldsymbol{r}_{\text{C,i}})}{\|\nabla S_{\text{C}}(\boldsymbol{r}_{\text{C,i}})\|_2} \tag{12}$$

By plugging (12) into (11) one obtains an expression for the friction cones, denoted with $\mathcal{F}(\boldsymbol{F}_{\text{C},i}, \boldsymbol{r}_{\text{C},i}, \mu_i)$, which is entirely dependent on the set of decision variables $\boldsymbol{x}$ in (8). Its analytical expression will be here omitted for brevity.

C. Nonlinear Optimization Problems

Based on: the robot centroidal statics model (7), a continuous description of the environment (9), and Coulomb friction cones (11), it is possible to set up different nonlinear programming (NLP) problems:

- *Multi-contact manipulation*, see Algorithm 1: produces a single quasi-static pose, in terms of CoM position $\overline{\boldsymbol{r}}_{CoM}$, contact positions $\overline{\boldsymbol{r}}_{\text{C}}$, and contact forces $\overline{\boldsymbol{F}}_{\text{C}}$, given a desired manipulation wrench at the CoM frame $\boldsymbol{\tau}_{\text{ext}}$, and the environment function S_{C}. Bounds on the contact positions (5d)

are introduced to approximate the reachable work-space through box constraints.

- *Contact lifting*, see Algorithm 2: produces a single quasi-static pose, in terms of CoM position $\overline{\boldsymbol{r}}_{CoM}$ and the contact forces $\overline{\boldsymbol{F}}_{\mathrm{C}}$, given the desired (pre-planned) contact positions $\boldsymbol{r}_{\mathrm{C}}^{\mathrm{des}}$, the related contact normals (not necessarily co-planar) $\boldsymbol{n}_{\mathrm{C}}^{\mathrm{des}}$ and the lifting contact/s. The force on the lifting contact/s $\boldsymbol{F}_{\mathrm{C,lift}}$ is set to zero (6b), while the positions of the stance contacts $\boldsymbol{r}_{\mathrm{C,stance}}$ are constrained to be equal to the desired value (6c). This algorithm can be applied iteratively to perform quasi-static stepping on flat or uneven terrains, once a set of planned contact positions and normals are provided.

Note that, in order to produce a sequence of n_{s} quasi-static poses, Algorithm 1 can be modified by extending the decision variable vector as follows:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}_{\mathrm{CoM}}^{[1]} & \dots & \boldsymbol{r}_{\mathrm{CoM}}^{[n_{\mathrm{s}}]} & \boldsymbol{r}_{\mathrm{C,i}}^{[1]} & \dots & \boldsymbol{r}_{\mathrm{C,i}}^{[n_{\mathrm{s}}]} & \boldsymbol{F}_{\mathrm{C,i}}^{[1]} & \dots & \boldsymbol{F}_{\mathrm{C,i}}^{[n_{\mathrm{s}}]} \end{bmatrix}^T$$

and by constraining the positions of the stance contacts at each step. This eventually allows for quasi-static locomotion in unstructured environments, e.g. a gap crossing task, assuming a suitable continuous function for the environment is provided, at the expense of computational complexity due to the increased problem dimension.

V. Multi-Contact Controller

A. QP-Based Hierarchical Inverse Kinematics

A humanoid platforms as CENTAURO allows to simultaneously perform multiple Cartesian tasks, organized in a mixed hard/soft priority relation. This hierarchy of tasks is generally referred to as Stack-of-Tasks (SoT) [32]. Given a prescribed SoT, it is possible to introduce a Quadratic Programming (QP) formulation of a hierarchical inverse kinematics (HIK) problem [33] as follows:

$$\begin{aligned} &\min_{\dot{\mathbf{q}}_{\mathrm{i}}} \|\mathbf{J}_{\mathrm{i}}\dot{\mathbf{q}}_{\mathrm{i}} - \dot{\mathbf{x}}_{\mathrm{i}}\|_2^2 + \lambda\|\dot{\mathbf{q}}_{\mathrm{i}}\|_2^2 \\ &\text{subject to} \\ &\underline{\mathbf{c}} \leq \mathbf{C}\dot{\mathbf{q}}_{\mathrm{i}} \leq \overline{\mathbf{c}} \\ &\underline{\mathbf{u}} \leq \dot{\mathbf{q}}_{\mathrm{i}} \leq \overline{\mathbf{u}} \\ &\mathbf{J}_{\mathrm{j}}\dot{\mathbf{q}}_{\mathrm{j}} = \mathbf{J}_{\mathrm{j}}\dot{\mathbf{q}}_{\mathrm{i}} \end{aligned} \tag{13}$$

where the i-th priority level is considered; $\mathbf{J}_{\mathrm{i}} \in \mathbb{R}^{m\times n}$ is the task Jacobian, $\dot{\mathbf{q}}_{\mathrm{i}} \in \mathbb{R}^n$ are the joint velocities, representing the problem variables, $\underline{\mathbf{u}}\backslash\overline{\mathbf{u}} \in \mathbb{R}^n$ are bounds on the problem variables and $\mathbf{C} \in \mathbb{R}^{l\times n}$ and $\underline{\mathbf{c}}\backslash\overline{\mathbf{c}} \in \mathbb{R}^l$ are the problem constraints. The scalar $\lambda \in \mathbb{R}^+$ weights a regularization term which ensures numerical stability near kinematic singularities. Being $\mathbf{x}_{\mathrm{d}} \in \mathbb{R}^m$, the desired end-effector position, the Cartesian velocities $\dot{\mathbf{x}}_{\mathrm{i}} \in \mathbb{R}^m$ are computed as $\dot{\mathbf{x}}_{\mathrm{i}} = \mathbf{K}\mathbf{e}$, where $\mathbf{e} = \mathbf{x}_{\mathrm{d}} - \mathbf{f}(\mathbf{q}_i)$ is the operational space tracking error (here $\mathbf{f}(\cdot)$ indicates the direct kinematics function) and $\mathbf{K} \in \mathbb{R}^{m\times m}$ is a positive definite (usually diagonal) matrix. The equality constraints $\mathbf{J}_{\mathrm{j}}\dot{\mathbf{q}}_{\mathrm{j}} = \mathbf{J}_{\mathrm{j}}\dot{\mathbf{q}}_{\mathrm{i}}$ in (13) enforces the priorities from all the previous j levels, with $j = \{0, \dots, i-1\}$, for $i \geq 1$. Notice that $\dot{\mathbf{q}}_{\mathrm{j}}$ is the solution given by the previously solved QPs.

For the proposed control architecture (see Fig. 2), the HIK is responsible to track the planned CoM position $\overline{\boldsymbol{r}}_{\mathrm{CoM}}$ and contact positions $\overline{\boldsymbol{r}}_{\mathrm{C}}$, producing the reference joint positions $\overline{\boldsymbol{q}} \in \mathbb{R}^n$ that feed the inner joint impedance controller.

B. Contact Force Distribution

Building upon the authors' work in [27], it is possible to formulate the following contact force distribution problem in order to track the planned contact forces $\overline{\boldsymbol{F}}_{\mathrm{C}}$, while providing reactive balance capabilities:

$$\begin{aligned} &\min_{\boldsymbol{F}_{\mathrm{C}}} \ \|\boldsymbol{F}_{\mathrm{C}} - \overline{\boldsymbol{F}}_{\mathrm{C}}\|_2^2 \\ &\text{subject to} \\ &\boldsymbol{J}_{\mathrm{C},u}^T\boldsymbol{F}_{\mathrm{C}} = \boldsymbol{g}_{\mathrm{u}}(\boldsymbol{q}) \\ &\boldsymbol{b}_{\mathrm{l}} \leq \boldsymbol{D}\,\boldsymbol{R}_{\mathrm{C}}\,\boldsymbol{F}_{\mathrm{C}} \leq \boldsymbol{b}_{\mathrm{u}} \end{aligned} \tag{14}$$

The equality constraint ensures closed-loop balancing by employing (2a) under quasi-static assumptions. The Jacobian $\boldsymbol{J}_{\mathrm{C},u}$ and the floating-base gravitational term $\boldsymbol{g}_{\mathrm{u}}(\boldsymbol{q}) \in \mathbb{R}^6$ are computed w.r.t. closed-loop joint coordinates, based on IMU feedback. The matrix $\boldsymbol{R}_{\mathrm{C}}$ maps the contact forces in a local reference frame such that the z-axis represents the normal component of the contact force. Note that, given the contact-surface normal $\boldsymbol{n}_{\mathrm{C},i}$ in (12), it is possible to find a rotation matrix $\boldsymbol{R}_{\mathrm{C},i} \in \mathbb{R}^{3\times3}$ satisfying $\boldsymbol{R}_{\mathrm{C},i}\,\boldsymbol{n}_{\mathrm{C},i} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$. Being ${}^{\{\mathrm{C,i}\}}\boldsymbol{F}_{\mathrm{C,i}} \in \mathbb{R}^3$ the i-th contact force expressed w.r.t. the local reference frame $\{\mathrm{C,\ i}\}$:

$${}^{\{\mathrm{C,i}\}}\boldsymbol{F}_{\mathrm{C,i}} = \boldsymbol{R}_{\mathrm{C,i}}\,\boldsymbol{F}_{\mathrm{C,i}} \tag{15}$$

the matrix $\boldsymbol{D}$ can be retrieved from the expression of the polyhedral linearized friction cones, i.e.:

$${}^{\{\mathrm{C,i}\}}F_{\mathrm{C,i}}^z \geq 0, \ |{}^{\{\mathrm{C,i}\}}F_{\mathrm{C,i}}^{x,y}| \leq \tilde{\mu}_i\ {}^{\{\mathrm{C,i}\}}F_{\mathrm{C,i}}^z \tag{16}$$

where $\tilde{\mu}_i = \frac{\sqrt{2}}{2}\mu_i$ models the inner approximation of the circular Coulomb friction cone in (11). Finally, the actuated torque $\boldsymbol{\tau}_{\mathrm{ff}} \in \mathbb{R}^n$ to be sent to the *joint-level controller* as a torque feed-forward term (see Fig. 2) is computed as:

$$\boldsymbol{\tau}_{\mathrm{ff}} = -\boldsymbol{J}_{\mathrm{C,a}}^T\boldsymbol{F}_{\mathrm{C}}. \tag{17}$$

Results from a balancing experiment are illustrated in the accompanying video.

VI. Experimental Validation

The manipulation task addressed in this letter consists in the pushing of a wooden cabinet, loaded with bricks, for a total weight exceeding 120 kg, see Fig. 1. Considering a Coulomb friction coefficient of 0.5, this task requires a pushing force greater than 600 N along the x-axis.

The used humanoid platform is the CENTAURO robot [20]. CENTAURO is a 39 degrees-of-freedom (DoF) hybrid wheeled-legged quadruped equipped with a bimanual humanoid upper-body, and has a weight of 92 kg. The robot is fully torque-controlled with direct sensing of the link side torque. Standard position control mode is also available. In its current setup no force/torque sensors are mounted at the robot end-effectors. CENTAURO is powered by the *XBotCore* middleware [34], while the *CartesI/O* framework [30], which relies on the hierarchical IK library *OpenSoT* [35], is responsible for Cartesian control.

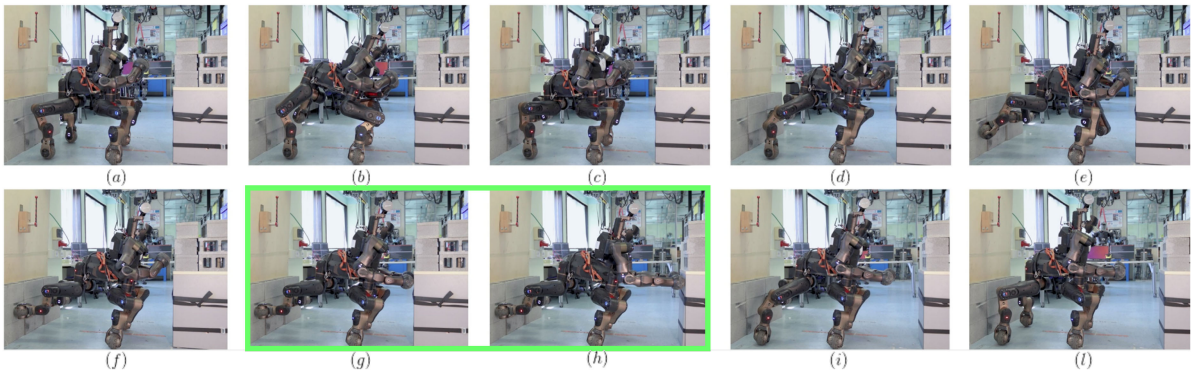


Fig. 4. Screen shots from a pushing experiment. In (a)–(f) CENTAURO reaches the final pose produced by Algorithm 1. This is done by iteratively applying Algorithm 2 in order to be able to lift and move the rear legs toward the planned contact locations on the wall, see (d) and (f). In (g) the two arms approach the object in velocity control mode, until contact is detected. In (h) the contact forces $\boldsymbol{F}_{\mathrm{C}}^{(1)}$ planned by Algorithm 1 are applied to the legs, while the manipulation force is distributed between the arms, effectively producing a forward motion of the object. Finally, in (i), (j) CENTAURO returns to the initial pose, again by iteratively applying Algorithm 2.

The *multi-contact planner* is executed in a ROS node running at 10 Hz. Algorithm 1 and Algorithm 2 have been implemented using IFOPT [36], an Eigen-based C++ interface to the nonlinear programming solver IPOPT [19]. To decrease the nonlinearity of $\mathcal{F}(\boldsymbol{F}_{\mathrm{C},i}, \boldsymbol{r}_{\mathrm{C},i}, \mu_i)$ and speed up the solver computation, we included the contact-surface normals $\boldsymbol{n}_{\mathrm{C}}$ in the set of decision variables and consequently added (12) as equality constraint in Algorithm 1. The considered set of contact points $\{\mathrm{C}_i\}$ comprises the four legs, while a pure manipulation force of $800\ N$ along the x-axis is applied at the two hands. The considered manipulation wrench is $\boldsymbol{\tau}_{\text{ext}} = \begin{bmatrix}800\ N & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T . \boldsymbol{r}_{\mathrm{CoM}}^{\text{des}}$ and $\boldsymbol{r}_{\mathrm{C}}^{\text{des}}$ in Algorithm 1 have been set to be equal to the related positions in the initial robot configuration. The box constraints (5d) have been selected according to the following guidelines: no overlapping should arise between each leg work-space; the box constraints must intersect the superquadric to avoid optimization unfeasibility; simple forward kinematics considerations are made using *CartesI/O* library. Finally, $W_{CoM} = 100,\ W_r = 10,\ W_F = 1$. The environment superquadric axial curvature has been chosen equal to $\boldsymbol{P} = \begin{bmatrix}8 & 8 & 8\end{bmatrix}^T$, while the vectors $\boldsymbol{C}$ and $\boldsymbol{R}$ have been chosen based on an approximate knowledge of the wall position w.r.t. the robot base.

The *multi-contact control* layer runs at $100\ Hz$. The HIK is managed by *CartesI/O* framework within a ROS node. The considered SoT can be written using the Math of Task (MoT) formalism [35] as follows:

$$\begin{pmatrix} \left(\sum_i {}^{\text{World}}\mathcal{T}_{\text{Foot}_i}^{[\text{XYZ}]}\right) / \\ \left({}^{\text{World}}\mathcal{T}_{\text{CoM}} + {}^{\text{World}}\mathcal{T}_{\text{LHand}} + {}^{\text{World}}\mathcal{T}_{\text{RHand}}\right) / \\ \left(\mathcal{T}_{\text{Posture}} + \sum_i {}^{\text{World}}\mathcal{T}_{\text{Ankle}_i}^{[\text{RPY}]} + {}^{\text{World}}\mathcal{T}_{\text{Waist}}^{[\text{RPY}]}\right) \end{pmatrix} << \left(\mathcal{C}_{\substack{\text{Pos.}\\\text{Lims}}} + \mathcal{C}_{\substack{\text{Vel.}\\\text{Lims}}}\right), \tag{18}$$

Here the $+$ and $/$ symbols are used to specify *soft* and *hard* priorities among tasks, respectively. ${}^{\mathrm{A}}\mathcal{T}_{\mathrm{B}}$ denotes a Cartesian task of the frame B relative to the frame A, while the $<<$ symbol is used to specify constraints to the whole stack or to single levels of the stack. A separate ROS node is dedicated to the *contact force distribution* problem for the considered set of contact points, i.e. the four legs. The *contact detection* has been implemented by evaluating the following logic condition:

$$\text{sign}\left(\boldsymbol{F}_{\mathrm{C,i}}^{(\text{est})} \cdot \boldsymbol{n}_{\mathrm{C,i}} - F_{\text{cnt}}\right) \tag{19}$$

where $F_{\text{cnt}} > 0$ is a scalar contact threshold. Since CENTAURO is not equipped with any force/torque sensors mounted at the end-effectors, the estimated contact forces $\boldsymbol{F}_{\mathrm{C}}^{(\text{est})}$ are computed under quasi-static assumptions as:

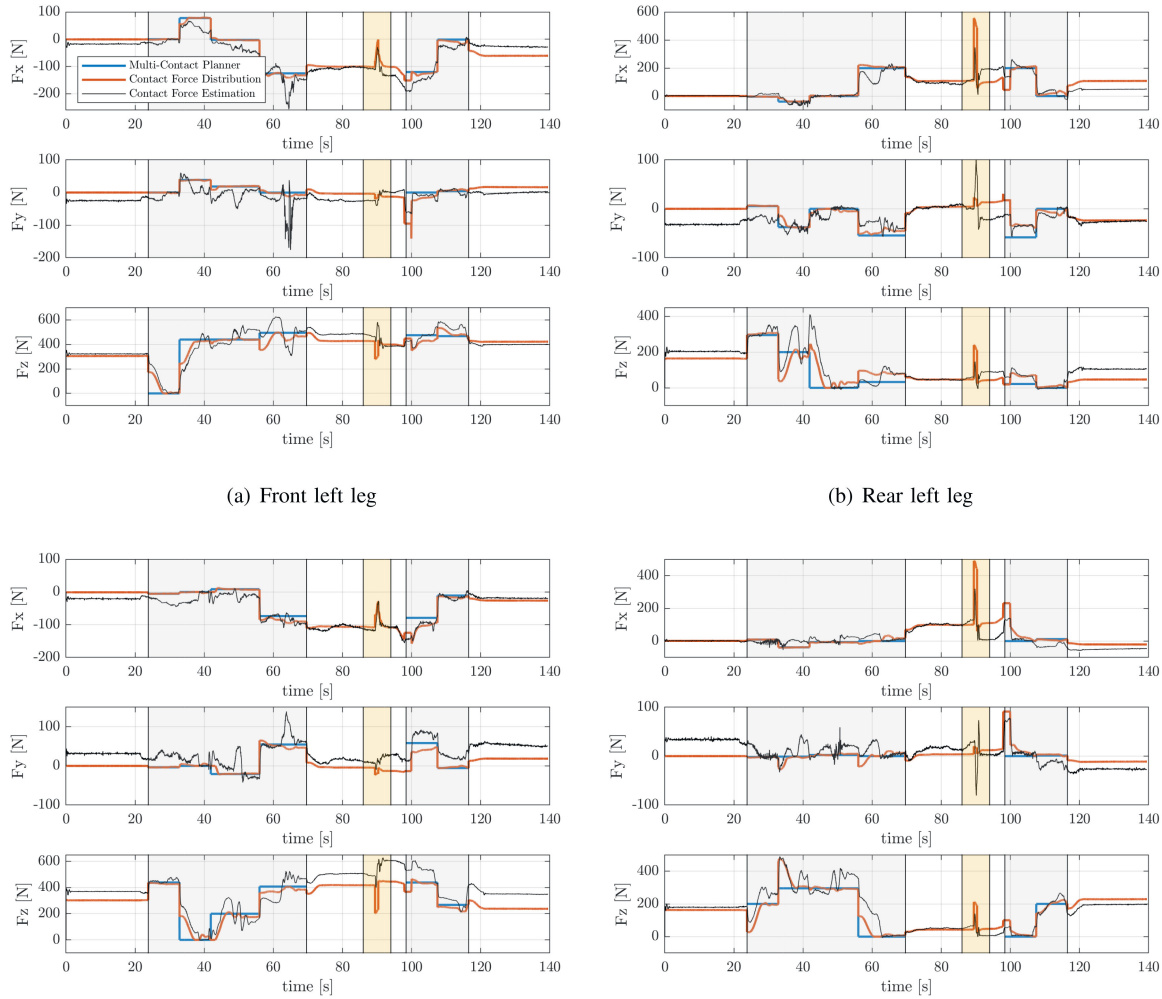
$$\boldsymbol{F}_{\mathrm{C}}^{(\text{est})} = \boldsymbol{J}_{\mathrm{C,a}}^{\dagger T}\left(\boldsymbol{g}_a - \boldsymbol{\tau}_{\text{mes}}\right) \tag{20}$$

being $\boldsymbol{\tau}_{\text{mes}} \in \mathbb{R}^n$ the measured link-side joint torques and $\boldsymbol{g}_a \in \mathbb{R}^n$ the actuated gravitational term computed based on closed-loop joint coordinates, while † stands for the Moore-Penrose pseudo-inverse. Finally, the *joint-level controller* runs at 2 kHz.

The state machine pseudo-code governing the pushing task execution is described in Algorithm 3, while snapshots from a complete experiment are shown in Fig. 4 (these results are also

Algorithm 3: Heavy-Object Pushing State Machine.

Solve Algorithm 1 $\mathbf{I}$:$\{\boldsymbol{\tau}_{\text{ext}},\ S_{\mathrm{C}}\}$ - $\mathbf{O}$:$\{\boldsymbol{r}_{\mathrm{C}}^{(1)},\ \boldsymbol{n}_{\mathrm{C}}^{(1)}, \boldsymbol{F}_{\mathrm{C}}^{(1)}\}$
for leg : {FL, FR, RL, RR} **do**
  Solve Algorithm 2 $\mathbf{I}$:$\{\boldsymbol{r}_{\mathrm{C}}^{(1)}$, leg$\}$ - $\mathbf{O}$:$\{\boldsymbol{r}_{\mathrm{CoM}}^{(2)},\ \boldsymbol{F}_{\mathrm{C}}^{(2)}\}$
  Send $\boldsymbol{F}_{\mathrm{C}}^{(2)}$ to *contact force distribution*
  Send $\boldsymbol{r}_{\mathrm{CoM}}^{(2)}$ to *HIK*
  while *contact detection* = false **do**
    Send $\boldsymbol{r}_{\mathrm{C,leg}}^{(1)}$ to *HIK*
    Move leg in velocity mode along $-\boldsymbol{n}_{\mathrm{C,leg}}^{(1)}$
  end while
end for
while *contact detection* = false **do**
  Move LHand, RHand in velocity mode along x-axis
end while
Set: $\boldsymbol{J}_{\mathrm{C},u}^T \boldsymbol{F}_{\mathrm{C}} = \boldsymbol{g}_{\mathrm{u}}(\boldsymbol{q}) + \boldsymbol{\tau}_{\text{ext}}$ in *contact force distribution*
$\boldsymbol{\tau}_{\text{ff}} \mathrel{+}= \left(0.5\ \boldsymbol{J}_{\mathrm{LHand,a}}^T \boldsymbol{\tau}_{\text{ext}} + 0.5\ \boldsymbol{J}_{\mathrm{RHand,a}}^T \boldsymbol{\tau}_{\text{ext}}\right)$



(a) Front left leg

(b) Rear left leg

(c) Front right leg

(d) Rear right leg

Fig. 5. Time histories of the legs' contact forces from a complete pushing experiment. During the stepping phase (grey shaded areas), the planned contact forces produced by the *multi-contact planner* (blue solid lines), through iterative application of Algorithm 2, are reactively tracked by the *contact force distribution* control layer (red solid lines). Black solid lines depict the estimated contact forces. It is worth pointing out that the contact force produced the *contact force distribution* problem can be driven to zero only after the CoM has reached the reference position planned by Algorithm 2. Finally, yellow shaded areas highlight the pushing phase.

illustrated in the accompanying video). Time histories of the contact forces at the legs are reported in Fig. 5.

Algorithm 1 produces the final quasi-static pose for the pushing task, automatically planning which surface to touch, i.e. pushing with the rear legs touching the wall, see Fig. 4(f). Algorithm 2 is then applied iteratively in order perform quasi-static stepping toward the pushing pose. Note that, the lifting sequence, see line 3 in Algorithm 3, is a heuristic. Such heuristic could be avoided by adopting an optimal control formulation of the contact planning problem, e.g. as in [16]. This represents a future research direction.

VII. Conclusion and Future Works

Motivated by the realization of the pushing task of a heavy object with a centaur-type humanoid, this work has presented a control architecture, conceived for torque-controlled platforms, that is able to freely exploit the environment complexity to perform similar multi-contact loco-manipulation actions, while meeting the implementation requirements to achieve a real demonstrator. Planning capabilities are offered by a continuous nonlinear optimization over the robot centroidal statics, relying on a continuous description of the environment complexity through superquadric functions. Reactive control capabilities are provided by a HIK control layer, combined with a contact force distribution problem ensuring closed-loop balancing. Experimental validation has been performed on the pushing task of a heavy object using the CENTAURO robot.

Extension of the proposed control architecture to position-controlled humanoids represents a future work, together with the validation on different platforms, e.g. bipeds, in order to prove its generality. Finally, an optimal control formulation of the multi-contact planner, an inverse dynamics implementation of the multi-contact controller and the automatic generation of the continuous environment function from point cloud data remain promising research directions.

REFERENCES

[1] S.-Y. Chung and O. Khatib, "Contact-consistent elastic strips for multi-contact locomotion planning of humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 6289–6294.

[2] B. Henze, A. Dietrich, M. A. Roa, and C. Ott, "Multi-contact balancing of humanoid robots in confined spaces: Utilizing knee contacts," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 697–704.

[3] F. Abi-Farraj, B. Henze, C. Ott, P. R. Giordano, and M. A. Roa, "Torque-based balancing for a humanoid robot performing high-force interaction tasks," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2023–2030, Apr. 2019.

[4] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5682–5689.

[5] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited humanoid locomotion planning," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2005, pp. 7–12.

[6] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 794–807, Aug. 2008.

[7] K. Bouyarmane and A. Kheddar, "Multi-contact stances planning for multiple agents," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 5246–5253.

[8] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robot. Auton. Syst.*, vol. 61, no. 5, pp. 428–442, 2013.

[9] J. Vaillant *et al.*, "Multi-contact vertical ladder climbing with an HRP-2 humanoid," *Auton. Robots*, vol. 40, no. 3, pp. 561–580, 2016.

[10] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2014, pp. 279–286.

[11] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 842–849.

[12] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, no. 4, p. 43, 2012.

[13] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 69–81, 2014.

[14] M. Neunert *et al.*, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1458–1465, Jul. 2018.

[15] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2014, pp. 295–302.

[16] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1560–1567, Jul. 2018.

[17] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2703–2710.

[18] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Auton. Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.

[19] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.

[20] N. Kashiri *et al.*, "CENTAURO: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1595–1602, Apr. 2019.

[21] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1522–1543, 2016.

[22] S.-H. Lee and A. Goswami, "Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 3157–3162.

[23] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1248–1255.

[24] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *Int. J. Humanoid Robot.*, vol. 2, no. 04, pp. 505–518, 2005.

[25] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 346–362, Apr. 2013.

[26] A. Herzog *et al.*, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Auton. Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[27] A. Laurenzi, E. M. Hoffman, M. P. Polverini, and N. G. Tsagarakis, "Balancing control through post-optimization of contact forces," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2018, pp. 320–326.

[28] M. Parigi Polverini, E. M. Hoffman, A. Laurenzi, and N. Tsagarakis, "Sparse optimization of contact forces for balancing control of multi-legged humanoids," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1117–1124, Apr. 2019.

[29] G. Vezzani, U. Pattacini, and L. Natale, "A grasping approach based on superquadric models," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1579–1586.

[30] A. Laurenzi, E. Hoffman, L. Muratore, and N. Tsagarakis, "CartesI/O: A ROS based real-time capable cartesian control framework," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 591–596.

[31] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016, pp. 955–988.

[32] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 670–685, Jun. 2009.

[33] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 785–792, Aug. 2011.

[34] L. Muratore, A. Laurenzi, E. Mingo Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "Xbotcore: A real-time cross-robot software platform," in *Proc. IEEE Int. Conf. Robotic Comput.*, 2017, pp. 77–80.

[35] E. Mingo Hoffman, A. Rocchi, A. Laurenzi, and N. G. Tsagarakis, "Robot control for dummies: insights and examples using OpenSoT," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2017, pp. 736–741.

[36] A. W. Winkler, "Ifopt - A modern, light-weight, Eigen-based C++ interface to Nonlinear Programming solvers Ipopt and Snopt." 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1135046