2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Multi-Contact Motion Planning and Control Strategy for Physical Interaction Tasks Using a Humanoid Robot

Francesco Ruscelli[1,2], Matteo Parigi Polverini[1], Arturo Laurenzi[1], Enrico Mingo Hoffman[1] and Nikos G. Tsagarakis[1]

***Abstract*— This paper presents a framework providing a full pipeline to execute a complex physical interaction behaviour of a humanoid bipedal robot, both from a theoretical and a practical standpoint. Building from a multi-contact control architecture that combines contact planning and reactive force distribution capabilities, the main contribution of this work consists in the integration of a sample-based motion planning layer conceived for transitioning movements where obstacle and self-collisions avoidance is involved. To plan these motions we use Rapidly Exploring Random Tree (RRT) projected on the contacts manifold and validated through the Centroidal Statics (CS) model, to ensure static balance on non-coplanar surfaces. Finally, we successfully validate the presented planning and control architecture on the humanoid robot COMAN+ performing a *wall-plank* task.**

I. Introduction

As opposed to classical fixed-base robots, controlling free-floating platforms such as bipeds is considerably harder due to the inherent under-actuation that generates a strong coupling between physical interaction and body stability. Humanoid research seeks to obtain dexterous and autonomous robots capable of mimicking the motion skills of a human being: in this respect, the two-fold problem of whole-body motion and environment interaction needs to be tackled to achieve robust physical mobility and interaction. A general high-level task is composed of unique movements that need to be planned and executed taking into account multiple contacts, balancing, and collisions. Hence, the building blocks of a controller for non-gaited multi-contact actions need to comprise a planning and a control layer, often organized in a hierarchical fashion [1], [2]. Depending on the task, each layer adopts a suitable model of the humanoid robot, spanning from reduced models, such as the Centroidal Dynamics (CD) [3], [4], to whole-body descriptions [5]. We separate the motion planning problem for humanoid robots in *Cartesian-space planning*, which generates contact points and the relative interaction forces at each contact, and *Joint-space planning*, which connects the sequence of planned stances with a continuous motion without violating a set of constraints. Two approaches to the planning problem can be found in literature: sample-based and gradient-based. The first uses discrete search methods such as A* or RRT [6]–[8]. While being a powerful tool, rising the complexity of the model greatly impacts the computation time for query a solution, and a guiding heuristic to span the search space may not be trivial to be determined, since the majority of the randomly sampled configurations do not satisfy all the constraints. The second relies on continuous optimization to find feasible solutions. It avoids unnatural motions and, if fed with the full dynamic model, produces refined behaviours [9], but its computational complexity makes it unpractical for many real-world scenarios. The problem can be also formulated by using simplified models such as the Centroidal Dynamics (CD), trading dynamic consistency for efficiency. To compute the contacts and the interaction forces, we rely on a gradient-based method using the robot CD under quasi-static assumptions, valid for any task which involves quasi-static poses. During the motion towards the planned contact, preventing joint limits, (self-)collisions and singular configurations while ensuring kinematic reachability and stability condition is critical, especially for a humanoid robot moving in a confined space. We exploit RRT using the full kinematic model of the robot to query a suitable joint-space in the heavily constrained configuration space.

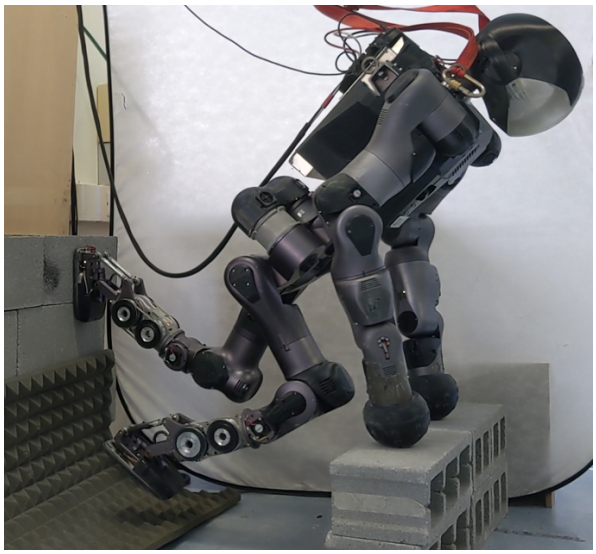

Fig. 1: COMAN+ reaching the *wall plank* pose. The weight of the robot is distributed among the hands and the left foot pushing against the wall.

Motion control is responsible for the closed-loop tracking of the trajectories and the interaction forces planned. In executing the planned trajectories, this layer provides compliance and robustness, critical requirements for the balancing

[1]Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova
[2]DIBRIS, Università di Genova, Via Opera Pia 13, 16145, Italy
{francesco.ruscelli, matteo.parigi, arturo.laurenzi, enrico.mingo, nikos.tsagarakis}@iit.it

of the robot. Two categories of balancing controllers exist: the first directly relies on full-body inverse dynamics [10]–[12] to solve the optimal force distribution problem. The second approximates the model used in the previous methods: it is based on a two-stage approach which involves either a pre-optimization of the contact forces [13]–[15] or a post-optimization [16] before mapping them to joint torques. Our *Interaction controller* is based on the last approach.

The contribution of this work lies in the development of a system for planning and control of multi-contact interaction tasks for hyper-redundant floating-base robots and its implementation on the humanoid COMAN+ (see Figure 1). In particular, we extend the work in [17] by providing:

- a sample-based planner to generate a sequence of feasible robot configurations, in terms of (self-)collisions and balance, from a multi-contact stance to another;
- an optimization-based interpolation of the planned sequence to obtain a time-optimal joint trajectory satisfying velocity and acceleration constraints.

Each layer is integrated into a framework that allows to execute complex motion actions while guaranteeing quasi-static equilibrium. Despite great progression in simulation, these complex physical interaction tasks are still difficult to replicate in real-world: we take a step towards bridging the gap between simulation and real-hardware achievements by successfully validating our control architecture on the robot COMAN+. We choose a challenging action for the average human being that consists in putting both hands on a support while rising both feet on a wall and subsequently balancing by distributing its weight on the four limbs keeping the whole body suspended from the ground, from here on referred as the *wall plank*. The designated motion highlights the critical role that contact interactions play in any motion that goes beyond nominal walking. Furthermore, it is highly non-linear and it is performed in a confined space, making it particularly challenging for the geometry of the robot, which doesn't have the full flexibility and mobility of a human being.

II. MATHEMATICAL BACKGROUND

A. Floating-base dynamic model

Humanoid robots fall in the category of hyper-redundant under-actuated robots. The vector of generalized variables of the n-link robot is composed of an un-actuated and an actuated part:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_u \\ \boldsymbol{q}_a \end{bmatrix} \tag{1}$$

where $\boldsymbol{q}_u \in \boldsymbol{R}^6$ is the (un-actuated) configuration of the virtual chain describing the pose of the robot w.r.t. to an inertial world frame and $\boldsymbol{q}_a \in \boldsymbol{R}^n$ are the coordinates of the n (actuated) joints. The model describing the robot dynamics is the following:

$$\begin{bmatrix} \boldsymbol{B}_u(\boldsymbol{q}) \\ \boldsymbol{B}_a(\boldsymbol{q}) \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{h}_u(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{h}_a(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{6\times n} \\ \boldsymbol{I}_{n\times n} \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \boldsymbol{J}_{\mathrm{C},u}^T \\ \boldsymbol{J}_{\mathrm{C},a}^T \end{bmatrix} \boldsymbol{F}_{\mathrm{C}} \tag{2}$$

which entails a selection matrix that differentiate actuation torques $\boldsymbol{\tau}$ from the full set of torques of the model. The unilateral contact forces $\boldsymbol{F}_{\mathrm{C}} \in \mathbb{R}^k$ are mapped in the joint-space by the Jacobian of all the links in contact with the environment $\boldsymbol{J}_{\mathrm{C}} \in \mathbb{R}^{k\times(n+6)}$ In the case of full planar contact $k = 6$, while for a point contact $k = 3$.

B. Reduced model

Full dynamic models grant kinematic and dynamic consistency at the expenses of a higher computational complexity. Reduced methods, on the other hand, rely on a lighter description: the most commonly used are the *Centroidal Dynamics* (CD) model or the *Centroidal Statics* (CS), a simplified version which discards acceleration and velocity. The CD model simplifies the full model by projecting the first six rows of (2) on the robot Center of Mass (CoM):

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

where $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{6\times(n+6)}$ is the *centroidal momentum matrix* (CMM). Terms $\boldsymbol{g} \in \mathbb{R}^3$ and $m \in \mathbb{R}$ are the vector of the gravity acceleration and the mass of the robot, respectively. Finally, $\boldsymbol{G}_{\mathrm{CD}} \in \mathbb{R}^{6\times k}$ is the the centroidal dynamics grasp matrix:

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

where n_{C} is the number of contacts and $\boldsymbol{r}_{\mathrm{C,i}} \in \mathbb{R}^3$ is the i-th contact position, while $\boldsymbol{S}$ is a skew-symmetric matrix. For any application in which only static equilibrium is required, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ can be discarded for a further simplification. Under quasi-static conditions, (5) yields the formulation of the CS:

$$m\boldsymbol{g} + \boldsymbol{G}_{\mathrm{CD}}\boldsymbol{F}_{\mathrm{C}} = \boldsymbol{0} \tag{5}$$

This reduced description assumes fixed contact placements with associated friction models to obtain each interaction force required to compensate for gravity and achieve static balancing.

C. Coloumb friction cones

Deriving contact-stability conditions for the robot requires to model the interaction between the robot and the environment. In any contact point C_i with the associated normal to the surface $\boldsymbol{n}_{\mathrm{C},i} \in \boldsymbol{R}^3$, the normal and tangential component of the interaction forces are defined as follows:

$$\begin{aligned} \boldsymbol{F}_{\mathrm{C},i}^n &= (\boldsymbol{F}_{\mathrm{C},i} \cdot \boldsymbol{n}_{\mathrm{C},i})\, \boldsymbol{n}_{\mathrm{C},i} \\ \boldsymbol{F}_{\mathrm{C},i}^t &= \boldsymbol{F}_{\mathrm{C},i} - (\boldsymbol{F}_{\mathrm{C},i} \cdot \boldsymbol{n}_{\mathrm{C},i})\, \boldsymbol{n}_{\mathrm{C},i} \end{aligned} \tag{6}$$

A contact is guaranteed as long as $\boldsymbol{F}_{\mathrm{C},i}$ lies inside the friction cone directed by $n_{\mathrm{C},i}$:

$$\mathcal{F}\left(\boldsymbol{F}_{\mathrm{C},i}, \boldsymbol{n}_{\mathrm{C},i}, \mu_i\right) := \begin{cases} \boldsymbol{F}_{\mathrm{C},i} \cdot \boldsymbol{n}_{\mathrm{C},i} > F_{\mathrm{thr}} \\ \left\|\boldsymbol{F}_{\mathrm{C},i}^t\right\|_2 \leq \mu_i \left(\boldsymbol{F}_{\mathrm{C},i} \cdot \boldsymbol{n}_{\mathrm{C},i}\right) \end{cases} \tag{7}$$

where μ_i is the static friction coefficient at the contact point C_i. The Euclidean norm $\|\cdot\|_2$ simplifies the model, by assuming a circular friction cone, i.e. the isotropy of friction.

III. Method overview

The proposed strategy extends the work in [17]: it is divided into a motion planning layer, which accounts for contact position, interaction forces, joint-space planning and a motion control layer that realizes the planned trajectory while granting continuous multi-contact equilibrium of the robot. Its hierarchical layered architecture is illustrated in Figure 3:

- *Cartesian-space planner*: prescribes reference values for contact force distribution $\boldsymbol{F}_{\mathrm{C},i}$ to achieve static equilibrium. It also outputs the position of each contact $\boldsymbol{r}_{\mathrm{C},i}$ and CoM $\boldsymbol{r}_{CoM}$, given the desired values. The solution is found via a non-linear optimization which reasons about the CS of the robot to reduce the problem complexity.
- *Joint-space planner*: generates a suitable joint trajectory $\overline{\boldsymbol{q}}(t)$ to reach the desired contact without violating balancing constraints and avoiding kinematic singularities as well as collisions with the environment and the robot itself.

The planned references are tracked by the motion control layer: planned trajectories can be either directly realized in joint-space or provided by a Cartesian controller based on Hierarchical Inverse Kinematics (HIK) when the joint-space planner is not required. The *Interaction controller* continuously distributes the planned contact force on each contact end-effector, generating feed-forward torque references $\boldsymbol{\tau}_{\mathrm{ff}}$ for the joint-level control loop. Both feet and hands are required to reach for a surface and establish a robust contact: in doing so, a collision detection algorithm senses if the controlled end-effector is in contact with the environment according to a prescribed threshold.

IV. Motion planning

In this section we present our strategy for planning environment interactions: a gradient-based method efficiently computes the Cartesian position of the contacts and the forces required at each supporting link for static equilibrium. To reach the desired contacts a set of feasible configurations of the robot is found by a sample-based algorithm, particularly suited when obstacle and self-collision avoidance are involved. Finally, an optimization problem interpolates the planned sequence of configurations ensuring a joint trajectory with bounded accelerations.

A. Cartesian-space planner

To grant quasi-static stability for a given robot configuration a critical role is played by the interaction forces. The first layer relies on non-linear programming (NLP) to compute force distribution among the subset of end-effector in contact with the environment. Additionally, given a CoM position reference, it returns a feasible CoM position and the active contact positions that satisfy the balancing conditions. At each stance the algorithm is fed with:

- *active contacts* C_a, specifying each contact position $\boldsymbol{r}_{\mathrm{C},a}$ and its associated normal $\boldsymbol{n}_{\mathrm{C},a}$;
- *lifted contacts*, C_l, enforcing the constraint $\boldsymbol{F}_{\mathrm{C},l} = 0$, to redistribute the robot weight on the remaining contacts. Note that, if no lifted contact are required, constraint (8c) not necessary;
- optionally, a desired CoM position $\boldsymbol{r}_{\mathrm{CoM}}^{\mathrm{des}}$.

The resulting NLP in the variables $\boldsymbol{x} = [\boldsymbol{r}_{\mathrm{CoM}}, \boldsymbol{r}_{\mathrm{C},a}, \boldsymbol{F}_{\mathrm{C}}]$ is formulated as:

$$\min_{\boldsymbol{x}} \|\boldsymbol{F}_{\mathrm{C}}\|^2_{2,W_1} + \|\boldsymbol{r}_{\mathrm{CoM}} - \boldsymbol{r}^{\mathrm{des}}_{\mathrm{CoM}}\|^2_{2,W_2} + \|\boldsymbol{r}_{\mathrm{C},a} - \boldsymbol{r}^{des}_{\mathrm{C},a}\|^2_{2,W_3}$$

$$\text{s.t.:} \quad m\boldsymbol{g} + \boldsymbol{G}_{\mathrm{CD}}\boldsymbol{F}_{\mathrm{C}} = \boldsymbol{0} \tag{8a}$$

$$\{\boldsymbol{F}_{\mathrm{C},a}, \boldsymbol{n}_{\mathrm{C},a}\} \in \mathcal{F}\left(\boldsymbol{F}_{C,a}, \boldsymbol{n}_{\mathrm{C},a}, \mu_a\right) \tag{8b}$$

$$\boldsymbol{F}_{C,l} = 0 \tag{8c}$$

$$\boldsymbol{r}_{\mathrm{C},i} \in S_{\mathrm{C}}(\boldsymbol{r}_{\mathrm{C},i}) \tag{8d}$$

Constraint (8a) ensures quasi-static stability based on the CS of the robot, while (8b) guarantees contact-stable stances given desired contact position computing the friction cone $\mathcal{F}$ as a function of the contact positions $\boldsymbol{r}^{\mathrm{des}}_{\mathrm{C},a}$, the corresponding normals $\boldsymbol{n}^{\mathrm{des}}_{\mathrm{C},a}$ and the given friction coefficient μ_a. Constraints (8c) set to zero the required force at the lifted contact l. Finally, constraint (8d) requires the contact to lay on the environment. To get the benefit of continuous optimization, a significantly faster method compared to Mixed Integer Programming where computational complexity rapidly increases as the number of decision variables grows, we choose a continuous description of the environment through superquadric functions $S_{\mathrm{C}}(\boldsymbol{r}_{\mathrm{C},i})$, as proposed in [17].

B. Joint-space planner

This layer is in charge of generating feasible transition motions between the static stances. However, these transitions are highly non-linear and constrained, as highlighted by the experiments in simulation (Figure 2). As a matter of fact, a standard HIK incurs in local minima and should be augmented with obstacle, self-collisions and singularities avoidance. To prevent these issues, a naive approach is tentatively selected, to validate whether two stances can be connected: a sequence of Cartesian end-effector sub-poses are hand-picked to achieve the final configuration. Although viable, this approach is burdensome and highly specific, since it requires a fine-tuning for each different robot action. For these reasons, we choose a more general approach that accounts for environmental and balancing constraints as well as self-collision and kinematic singularities. Sample-based planner such as RRT are particularly suited for this task:

- self-collision are rapidly checked and the relative joint configurations discarded;
- contacts can be guaranteed by projecting the robot configuration on a desired manifold;
- validity checks can be applied to impose stability constraints and (self-)collision free paths.

The proposed planner uses a probabilistically complete sample-based algorithm to find a feasible solution. Given the starting configuration $\boldsymbol{q}_{start}$ and final configuration $\boldsymbol{q}_{goal}$, the planner finds a collision-free joint trajectory: since the motion is realized in joint-space, kinematic singularities do

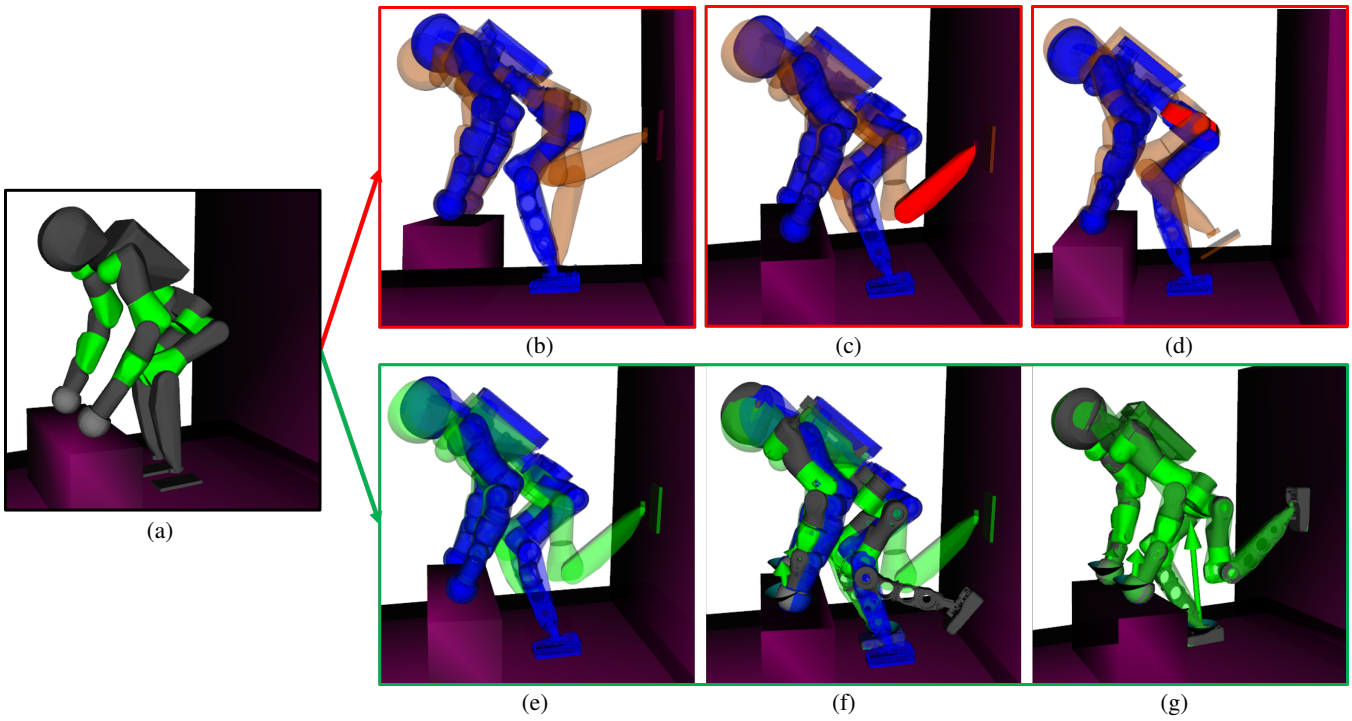

Fig. 2: Simulations of COMAN+ positioning a foot on the wall. The initial quadrupedal pose is shown in (a), fed to the Joint-space planner as the starting configuration $\boldsymbol{q}_{start}$ (in blue). A feasible configuration is green, or red otherwise. Without any trajectory planning the robot incurs in kinematic singularities, e.g. full extension of the knee joint of the supporting leg (b), environmental collisions, e.g. foot impacting the wall (c) or self-collision, e.g. hip joints colliding (d). The Joint-space planner checks if the goal configuration $\boldsymbol{q}_{end}$ is feasible, as shown in (e) and produces a joint trajectory from $\boldsymbol{q}_{start}$ to $\boldsymbol{q}_{goal}$ without violating the CS constraints. Note in (f) and (g) the friction cones and the relative contact forces satisfying the CS.

not arise. Contact consistency, due to the floating-base nature of our robot, is injected in the problem using manifold projection: robot configurations $\boldsymbol{q}_i$ are mapped on an implicit manifold [18] defined by the constraint function $F(\boldsymbol{q}) = 0$ that describes a set of end-effectors in contact with the the environment. Physical limitations, such as balancing, are subsequently considered by validating the projected configuration. In the general case of multiple non-coplanar contacts, where the support polygon doesn't suffice to grant static equilibrium, we formulate the following optimization problem in the contact force variables:

$$\begin{aligned} \min_{\boldsymbol{F}_\mathrm{C}} \quad & \left\| \boldsymbol{J}_{\mathrm{C},u}^T \boldsymbol{F}_\mathrm{C} - \boldsymbol{g}_u(\boldsymbol{q}) \right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{b}_l \leq \boldsymbol{D}\boldsymbol{R}_\mathrm{C}\boldsymbol{F}_\mathrm{C} \leq \boldsymbol{b}_\mathrm{u} \end{aligned} \tag{9}$$

which corresponds to enforcing the CS (5) at the floating base, under linearized friction cones constraints. Notice that the problem as formulated in (9) has always a solution inside the friction cones. The projected pose is considered statically balanced if the cost function in (9) doesn't exceed a certain threshold ε, and, if collision free, the pose is added to the search tree. Any configuration that violates the CS is discarded. The procedure exploits the OMPL library [18] to set up the randomized planning problem and spans the constrained configuration space of the humanoid until it reach a feasible goal (see Algorithm 1). The planning scene consists of all the obstacles involved, so to avoid any stance in collision with the environment. The robot configuration is projected on the manifold using an atlas-based methodology [18]. Self-collision are detected using the FCL package [19]. To speed up the algorithm, we discard unnecessary checks for self-colliding configurations and by identifying all possibly colliding pairs of links. Future works will allow to specify solely the contacts to automatically get a feasible goal configuration instead of providing a pre-planned one. The final set of stances found by the planner is then interpolated to get a joint trajectory.

C. Trajectory interpolation

Because the selected RRT-based planner is a *geometric* planner, its output is a simple sequence of points $\{\boldsymbol{q}_k\}_{k=0}^N$ in the robot configuration space, that does not carry any information about its time parametrization. Consequently, there is no guarantee whatsoever that output points will accumulate in correspondence to high-acceleration trajectory segments. The direct interpolation of the output sequence assuming an evenly-spaced time grid will therefore lead to high-acceleration, jerky motions, especially when changes of directions occur to any of the robot degrees of freedom. To address this issue, we propose to perform an optimization-based interpolation relying on polynomial splines of the

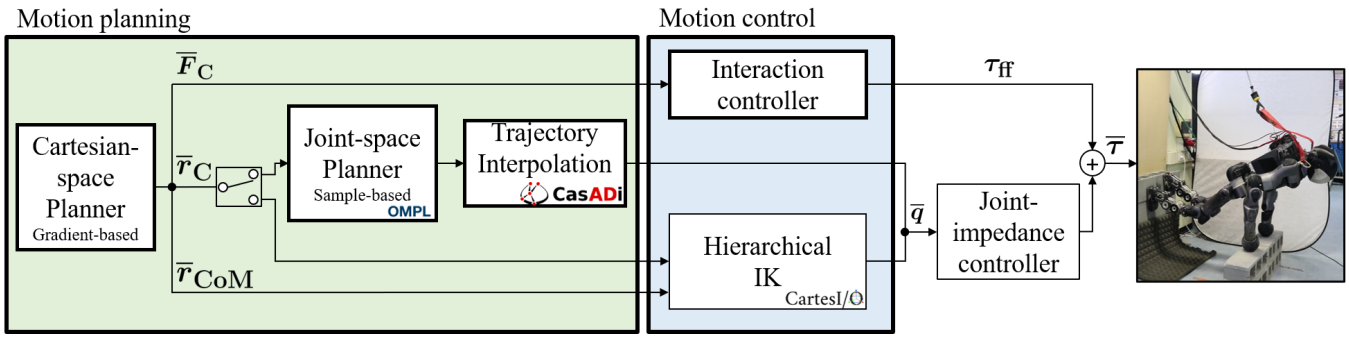

Fig. 3: Scheme of the proposed controller. The Joint-space planner uses OMPL [18] to generate suitable robot configurations. The optimization problem for joint trajectory interpolation is set up using CasADi [20], while CartesI/O [21] is responsible for the whole-body Inverse Kinematics problem.

following form:

$$\boldsymbol{q}(t) = \boldsymbol{s}_k(t), \quad t \in [t_k,\, t_{k+1}]\,,\; k = 0, \ldots, N-1 \tag{10}$$

$$\boldsymbol{s}_k(t) = \sum_{i=0}^{d} \boldsymbol{a}_{k,i}(t - t_k)^i, \tag{11}$$

where (i) $\boldsymbol{a}_{k,i} \in \mathbb{R}^n$ are the spline coefficients, (ii) $d \in \mathbb{N}^+$ is the spline degree, and (iii) $t_k \in \mathbb{R}$ is the k-th time knot. The spline degree is fixed to $d = 3$ in this work, which yields a continuous velocity profile with bounded acceleration.

Algorithm 1: Multi-Contact Motion Planning and Control

Data: $\boldsymbol{q}_{goal}$, $\mathrm{C}_a, \mathrm{C}_l, \boldsymbol{r}^{des}_{\mathrm{C},a}$, obstacles
Result: collision-free, statically balanced motion
for *multi-contact action in task* **do**
1. set C_l to Cartesian-space Planner
2. run Cartesian-space Planner
   $\rightarrow$ get $\{\overline{\boldsymbol{r}}_{\mathrm{C},a}, \overline{\boldsymbol{r}}_{CoM}, \overline{\boldsymbol{F}}_{\mathrm{C},a}\}$
3. send $\overline{\boldsymbol{r}}_{CoM}$ to HIK
4. set up *Planning Scene*
5. set $\boldsymbol{q}_{start}$ to Joint-space Planner
6. generate *contact manifold* from $\{\boldsymbol{q}_{start}, \mathrm{C}_a\}$
7. set $\boldsymbol{q}_{goal}$, project it on the *contact manifold*
   while *(exact solution found)* **do**
8. run RRT constrained with CS $\rightarrow$ get $\boldsymbol{q}_k$
   end
9. optimal trajectory interpolation: $\boldsymbol{q}_k \rightarrow \boldsymbol{q}(t)$
10. lift contact C_l with HIK
11. send $\boldsymbol{q}(t)$ to robot
12. run HIK for surface-reaching
13. add new contact C_a to Cartesian-space Planner
14. send $\boldsymbol{F}_{\mathrm{C},a}$ to Interaction Controller

end

With the aim to compute a *time-optimal* trajectory satisfying any given joint velocity and acceleration bounds, we therefore set up the following optimization problem:

$$\min_{\boldsymbol{a},\boldsymbol{\Delta t}} \quad \mathbf{1}^T \boldsymbol{\Delta t} + \lambda \|\boldsymbol{a}\|^2 \tag{12}$$

$$\text{s.t.:} \quad -\dot{\boldsymbol{q}}_{\max} \le \dot{\boldsymbol{s}}_k(t_k) \le \dot{\boldsymbol{q}}_{\max} \quad \forall k = 0, \ldots, N \tag{13}$$

$$-\ddot{\boldsymbol{q}}_{\max} \le \ddot{\boldsymbol{s}}_k(t_k) \le \ddot{\boldsymbol{q}}_{\max} \quad \forall k = 0, \ldots, N \tag{14}$$

$$\boldsymbol{s}_{k-1}(t_k) = \boldsymbol{s}_k(t_k) = \boldsymbol{q}_k \quad \forall k = 1, \ldots, N \tag{15}$$

$$\dot{\boldsymbol{s}}_{k-1}(t_k) = \dot{\boldsymbol{s}}_k(t_k) \quad \forall k = 1, \ldots, N \tag{16}$$

$$\boldsymbol{\Delta t} \ge \mathbf{0} \tag{17}$$

where the minimization of (12) is carried out over the vector of all spline coefficients $\boldsymbol{a} \in \mathbb{R}^{n \cdot d \cdot N}$ and the vector of time interval *durations* $\boldsymbol{\Delta t} \in \mathbb{R}^N$, whereas the constraints (13)-(17) enforce velocity and acceleration limits, plus the interpolation and continuity conditions, and positive interval durations. In order to regularize the solution, the spline coefficients norm is penalized via a small scalar $\lambda > 0$. The resulting non-linear problem is high-dimensional but highly sparse as well, and it can be solved in a fraction of the time required by the sample-based planner.

V. Interaction Control

This controller is based on [16], which post-optimizes the contact forces to account for the under-actuated component of the system. As depicted in Figure 3, it consists in feeding the inner joint impedance control loop:

$$\boldsymbol{\tau} = \mathbf{K}_P\,(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_D\,(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \boldsymbol{\tau}_{\mathrm{ff}} \tag{18}$$

the feed-forward torque:

$$\boldsymbol{\tau}_{\mathrm{ff}} = -\boldsymbol{J}^T_{\mathrm{C,a}} \boldsymbol{F}_{\mathrm{C}} \tag{19}$$

The reference contact force is obtained solving an optimization problem to track the planned contact force $\overline{\boldsymbol{F}}_{\mathrm{C}}$ under two conditions for balancing:

$$\begin{gathered} \min_{\boldsymbol{F}_{\mathrm{C}}} \left\| \boldsymbol{F}_{\mathrm{C}} - \overline{\boldsymbol{F}}_{\mathrm{C}} \right\|_2^2 \\ \text{s.t.} \\ \boldsymbol{J}^T_{\mathrm{C},u} \boldsymbol{F}_{\mathrm{C}} = \boldsymbol{g}_u(\boldsymbol{q}) \\ \boldsymbol{b}_l \le \boldsymbol{D}\boldsymbol{R}_{\mathrm{C}} \boldsymbol{F}_{\mathrm{C}} \le \boldsymbol{b}_{\mathrm{u}} \end{gathered} \tag{20}$$

The first equality constraint enforces centroidal statics balancing (5). The second constraint bounds the contact forces

to satisfy the friction cones contact condition. In particular, the matrix $\boldsymbol{R}_{\mathrm{C}}$ accounts for the contact orientation: given the normal to the surface in the contact point C, a rotation matrix $\boldsymbol{R}_{\mathrm{C}}$ can be found such that ${}^{\mathrm{C}}\boldsymbol{F}_{\mathrm{C}} = \boldsymbol{R}_{\mathrm{C}}\boldsymbol{F}_{\mathrm{C}}$, where ${}^{\mathrm{C}}\boldsymbol{F}_{\mathrm{C}} \in \mathbb{R}^3$. The matrix $\boldsymbol{D}$ is related to the static friction coefficient:

$$\begin{aligned} {}^{\mathrm{C}}\boldsymbol{F}^{z}_{\mathrm{C}} &\geq 0, \\ \left| {}^{\mathrm{C}}\boldsymbol{F}^{x,y}_{\mathrm{C}} \right| &\leq \tilde{\mu}\, {}^{\mathrm{C}}\boldsymbol{F}^{z}_{\mathrm{C}} \end{aligned} \tag{21}$$

where $\tilde{\mu} = \frac{\sqrt{2}}{2}\mu$ is an inner approximation for the isotropic Couloumb friction cone in (7). To recover the contact forces the Jacobian $\boldsymbol{J}^{T}_{\mathrm{C},u}$ and the gravitational term $\boldsymbol{g}_u(\boldsymbol{q})$ are computed in closed-loop using the robot model and the IMU measurements.

VI. Experimental validation

In order to show the capabilities of the proposed architecture we conceived an uncommon and challenging task for a bipedal humanoid, the *wall plank*: the task consists in reaching a pose with two hands close to the ground level and two feet on an adjacent wall in the minimum number of actions, starting from a standing position. Algorithm 1 describes a general procedure that we exploit to execute this particular physical interaction task. The robot crouches to reach a quadrupedal stance with each hand resting on a pile of two bricks (at 30 cm from the ground, on a surface of 50x20 cm). The distance between the middle of the robot soles and the wall is approximately 65 cm, as shown in Figure 1. In the initial configuration the robot is standing on its feet, while in the goal position the body is held slanted to the ground, supported by the hands and pushing on the wall with its feet. The overall motion is composed of strongly non-linear manoeuvres due to the robot kinematics, the environment geometry and the inherently cramped poses the robot needs to adopt. Hence a strong emphasis is put in generating feasible trajectories that satisfy joint limits and avoid self-collisions and singularities. In particular, to reach the goal position the robot goes through a discrete sequence of multi-contact stances:

- *Standing stance*: initial configuration, the robot is standing on its feet;
- *Crouched stance*: bent, with feet on the ground and hands on two piled bricks;
- *Single leg stance*: one leg is lifted on the wall;
- *Double leg stance*: goal configuration, both feet are on the wall while the hands are on the bricks.

The robot performing the above-mentioned task is COMAN+, a humanoid built at Istituto Italiano di Tecnologia. The robot has 28 DoFs: the arms are based on a 7-DoF kinematics, providing one degree of redundancy, while the legs rely on 6 DoF, 2 of which are given by a particular four-bar mechanism for ankle actuation [22]. The last 2 DoF are situated in the torso, allowing rotation only around the longitudinal and sagittal axis. COMAN+ weights 70 kg and is 1.7 m tall. Each joint is equipped with a torque sensor, enabling the implementation of torque and impedance control, while standard position control is also available. A force/torque sensor is positioned on each foot. The soles are covered with rubber, while the anthropomorphic hands are replaced with rubber-coated spherical end-effectors. Their extremities are sensor-less, but the resulting configuration is particularly suited for a task in which ruggedness and high friction are essential. COMAN+ is powered by the real-time software architecture XBotCore [23]. Trajectory generation for the transition to a quadrupedal stance relies on a whole-body hierarchical inverse kinematics framework named CartesI/O [21] to define multiple Cartesian tasks organized in a stack of task (SoT) fashion. For simple trajectories, this local method is preferred, since it is efficient and deterministic. However, the last two poses entail a highly non-linear motion in a narrow space, limited by environmental obstacles and kinematic constraints: the HIK, being a local method, runs up against ill-behaviours such as singularities and self-collisions, detrimental in these delicate phases. The integration of joint-space and Cartesian-space control enhances the framework adaptability: straightforward behaviours such as CoM shifting and surface-reaching are managed by CartesI/O while complex motions are autonomously planned and sent to the robot as joint-space trajectories. The proposed framework is organized as follows:

- the *Cartesian-space planner* is based on IFOPT, an Eigen-based C++ interface to the Non-linear Programming (NLP) solver IPOPT [24]. At each stage, it produces the contact forces and the end-effector (contacts) positions to achieve static equilibrium, given a desired CoM and end-effector reference positions and the associated bounds.
- the *Joint-space planner* runs the RRT-Connect algorithm based on the OMPL library [18] to find a suitable set of robot configuration for transitioning from one stance to another without violating the multiple constraints. The interpolation between the resulting configurations is formulated as an optimal control problem using CasADi [20] which generates a NLP solved with IPOPT.
- the *Interaction controller* is responsible for the force distribution problem, running in a ROS node at 100 Hz. The forces at each supporting link are computed using the closed-loop joint coordinates and the IMU readings from the robot.

The computational load of the pipeline during the wall-plank task is negligible: the complexity of the NLP in the *Cartesian-space planner* is low, thanks to the choice of a continuous description of the environment and the simplified centroidal dynamics under quasi-static assumption. Similarly, the computational time of the RRT in the Joint-space planner is, on average, less than one second and, as highlighted in Section IV-C, the NLP for trajectory interpolation is solved in a fraction of the time required by the Joint-space planner. CartesI/O runs on separate node at 100 Hz, and a specific

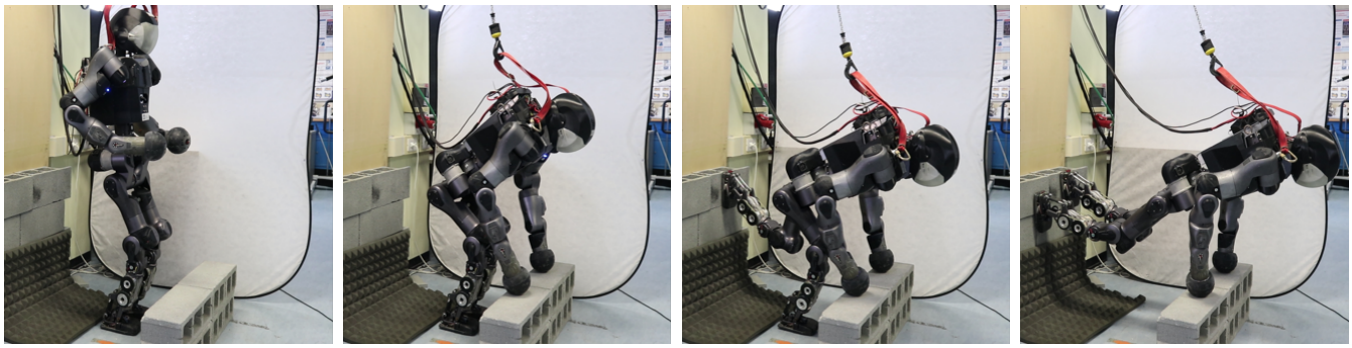

Fig. 4: Sequence of frames of COMAN+ performing the *wall plank*. The horizontal gap between the bricks and the wall is approximately 65 cm, a narrow space for a 1.7 m tall robot. The pile of bricks is 30 cm high.

SoT was tailored for this work:

$$\begin{pmatrix} \left({}^{\mathrm{W}}\mathcal{T}_{\mathrm{LFoot}} +^{\mathrm{W}} \mathcal{T}_{\mathrm{RFoot}} +^{\mathrm{W}} \mathcal{T}^{RPY}_{\mathrm{Torso}}\right) / \\ {}^{\mathrm{Waist}}\mathcal{T}_{\mathrm{LArm}} +^{\mathrm{Waist}} \mathcal{T}_{\mathrm{RArm}} / \\ {}^{\mathrm{W}}\mathcal{T}^{XY}_{\mathrm{CoM}} +^{\mathrm{W}} \mathcal{T}^{Z}_{\mathrm{Waist}} / \\ \mathcal{T}_{\mathrm{Posture}} \end{pmatrix} << \left(\mathcal{C}_{\substack{\mathrm{Joint} \\ \mathrm{Lims}}} + \mathcal{C}_{\substack{\mathrm{Vel.} \\ \mathrm{Lims}}}\right) \tag{22}$$

where ${}^{(A)}\mathcal{T}_{(B)}$ represents a Cartesian tasks of the frame B w.r.t. the frame A or expressed in joint space if A is not specified. The $/$ and $+$ symbols are used to impose *strict* or *soft* priority, respectively, among sets of tasks. The symbol $<<$ is used to enforce desired constraint in the task execution. During the motion, the SoT is modified online to adapt the robot behaviour in each transition phase: specifically, the parent link of a desired task can be selected or the whole task can be disabled, effectively removing it from the stack.

The contacts are detected using a simple algorithm that checks if a contact force $\boldsymbol{F}_{\mathrm{C}}$ along the normal to the surface n_{C} exceeds a certain threshold, set to 15 N for the hands and 50 N for the feet. $\boldsymbol{F}_{\mathrm{C}}$ is measured from the force/torque sensor mounted at the sole and estimated for the arm end-effector by exploiting the readings from the torque sensors at the arms joint level.

Finally, the joint level controller, composed of a decentralized impedance controller with a torque feed-forward term runs at 2 kHz.

The sequence of stances composing the wall-plank task is depicted in Figure 4, while Figure 5 reports the values of the contact forces at the legs during the experiment, computed by the *Cartesian-space planner* and the the *Interaction controller*. Videos of the experiments can be found at `https://www.youtube.com/playlist?list=PL7c1ZKncPan7g0tHDx5jP0kpt5si8G3pR`, showing the hand-crafted motion and the one using the RRT trajectory planner. Notice also how the final configuration of the robot, in the wall plank pose, is robust to external pushes, thanks to the *Interaction controller*. The reactive balance granted by this control layer and the surface-reaching algorithm, which uses the force/torque sensors to detect impacts, provide robustness to modeling error of the surfaces and to external disturbances. The velocity for each phase of the experiment can be increased, as long as the quasi-static stability is not violated.

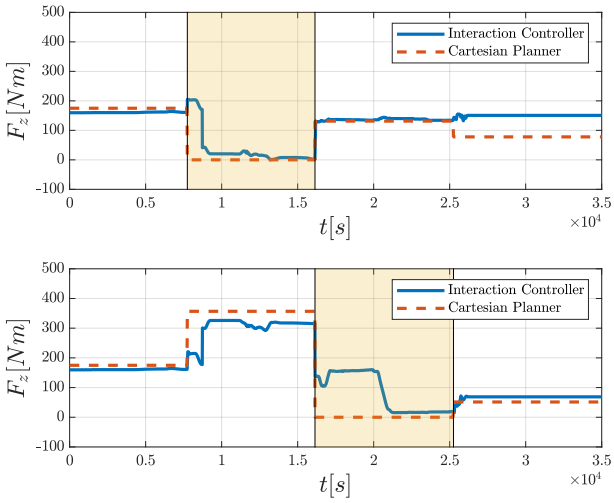

Fig. 5: Values for the contact forces on the z-axis of left leg (above) and right leg (below) while performing Algorithm 1 for the *wall plank* task. The *Interaction controller* runs in closed-loop, continuously distributing the forces on the supporting links: this generates discrepancies w.r.t. the planned forces provided by the *Cartesian-space planner*, since they are computed based on a simplified model (CS) *before* the actual motion. The highlighted areas correspond to the flight phase of each leg: when a foot is lifted, its relative contact force is set to zero.

VII. Conclusions and Future Works

We presented a complete planning and control architecture that allows a humanoid robot to perform advanced real-world physical interaction tasks. The *Cartesian-space planner* uses a gradient-based optimization algorithm based on the CS to compute the contact forces for static equilibrium. The *Interaction controller* is responsible for the tracking of the references without violating the continuous balancing conditions imposed by the CS and the Coloumb friction cones. A joint-

space planner is used in combination with the *Cartesian-space planner*: while ad-hoc motion can be hand-crafted to reach the desired contacts, our algorithm guarantees general (self-)collision-free motions avoiding kinematics singularities. We decide to rely on a global planning RRT algorithm to find a suitable trajectory, given the full-body model of the robot, and physical feasibility conditions, i.e. stability constraints (enforced using the CS) as well as environmental constraints, i.e. obstacle and contact constraints and a self-collision map. The realization of the aforementioned complex behaviour on a humanoid platform revealed the intrinsic limitations of its kinematic structure. The inherent poor flexibility of the robot, together with the lack of the torso pitch DoF increased the complexity of this particularly challenging task. The presented framework is capable to find a robust solution for this demanding task. Nevertheless, future enhancement will allow for a more organic pipeline towards physical autonomy. To reach the goal configuration each intermediate motion will be autonomously planned in terms of number of contacts and body posture, without the need to manually outlining them at each phase of the sequence of motions. Since the joint-space planner randomly searches through the configuration space, a strategy to penalize burdensome and awkward movements will be conceived in favor of more natural-looking ones. The application of the presented pipeline to different scenarios is straightforward as long as they involve quasi-static motions and a continuous description of the environment can be provided: we will extend the task of wall plank to more articulated and dynamic motions, re-positioning the contacts to climb the wall with the feet or walk on the hands.

REFERENCES

[1] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, no. 3, pp. 429–455, 2016.

[2] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, and F. Kanehiro, "Multi-contact vertical ladder climbing with an hrp-2 humanoid," *Autonomous Robots*, vol. 40, 02 2016.

[3] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*, nov 2014, pp. 295–302. [Online]. Available: http://ieeexplore.ieee.org/document/7041375/

[4] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 4030–4035.

[5] S. Dalibard, A. El Khoury, F. Lamiraux, A. Nakhaei, M. Taïx, and J.-P. Laumond, "Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1089–1103, 2013.

[6] K. Hauser, T. Bretl, K. Harada, and J. C. Latombe, "Using motion primitives in probabilistic sample-based planning for humanoid robots," *Springer Tracts in Advanced Robotics*, vol. 47, pp. 507–522, 2008.

[7] T. Nishi and T. Sugihara, "Motion planning of a humanoid robot in a complex environment using rrt and spatiotemporal post-processing techniques," *International Journal of Humanoid Robotics*, vol. 11, 06 2014.

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

[9] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, 2012.

[10] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 4906–4913.

[11] 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 Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, apr 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6482266/

[12] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical systems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1366–1373.

[13] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.

[14] A. Werner, B. Henze, D. A. Rodriguez, J. Gabaret, O. Porges, and M. A. Roa, "Multi-contact planning and control for a torque-controlled humanoid robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2016, pp. 5708–5715. [Online]. Available: http://ieeexplore.ieee.org/document/7759840/

[15] 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," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 3157–3162, 2010.

[16] A. Laurenzi, E. Mingo Hoffman, M. Parigi Polverini, and N. G. Tsagarakis, "Balancing control through post-optimization of contact forces," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Beijing, China, Nov 2018, pp. 320–326. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8625013

[17] M. Parigi Polverini, A. Laurenzi, E. Mingo Hoffman, F. Ruscelli, and N. G. Tsagarakis, "Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator," *IEEE Robotics and Automation Letters, presented at the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 5, no. 2, pp. 859–866, 2020.

[18] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, http://ompl.kavrakilab.org.

[19] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 3859–3866.

[20] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[21] A. Laurenzi, E. Mingo Hoffman, L. Muratore, and N. G. Tsagarakis, "CartesI/O: A ROS Based Real-Time Capable Cartesian Control Framework," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02017773

[22] F. Ruscelli, A. Laurenzi, E. Mingo Hoffman, and N. G. Tsagarakis, "A fail-safe semi-centralized impedance controller: Validation on a parallel kinematics ankle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, Oct 2018, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8594112

[23] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "XBotCore: A Real-Time Cross-Robot Software Platform," in *First IEEE International Conference on Robotic Computing (IRC)*. IEEE, apr 2017, pp. 77–80.

[24] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, mar 2006. [Online]. Available: http://link.springer.com/10.1007/s10107-004-0559-y