

Whole-Body Walking Generation using Contact Parametrization: A Non-Linear Trajectory Optimization Approach

Stefano Daffarra^{1,2}, Giulio Romualdi^{1,2}, Giorgio Metta¹, and Daniele Pucci¹

Abstract—In this paper, we describe a planner capable of generating walking trajectories by using the centroidal dynamics and the full kinematics of a humanoid robot model. The interaction between the robot and the walking surface is modeled explicitly through a novel contact parametrization. The approach is *complementarity-free* and does not need a predefined contact sequence. By solving an optimal control problem we obtain walking trajectories. In particular, through a set of constraints and dynamic equations, we model the robot in contact with the ground. We describe the objective the robot needs to achieve with a set of tasks. The whole optimal control problem is transcribed into an optimization problem via a Direct Multiple Shooting approach and solved with an off-the-shelf solver. We show that it is possible to achieve walking motions automatically by specifying a minimal set of references, such as a constant desired Center of Mass velocity and a reference point on the ground.

I. INTRODUCTION

Planning locomotion trajectories for humanoid robots requires considering high-dimensional multi-body systems instantiating contacts with the surrounding environment. Given their intrinsic under-actuation, these robots have to exploit the interaction with the environment and their ability to change “shape” in order to move.

During the DARPA Robotics Challenge [1], it became popular to approach the locomotion problem with a hierarchical control architecture. In our previous works [2], [3] we explored this concept by adopting a three layer control architecture. The outer layer, i.e. the *trajectory optimization for foot-step planning*, is in charge of generating walking trajectories starting from high-level commands, such as those coming from a joystick. The output of this layer is served to the *receding horizon controller (RHC)*, also referred to as Model Predictive Control (MPC) [4]. Its aim is to generate *centroidal* [5] quantities obtainable by the robot without incurring into an undesired fall state. Compared to the first, this second layer adopts more complex models with a shorter prediction horizon. The last stage, the *whole-body quadratic programming control* is in charge of stabilizing the planned trajectories exploiting the full robot model with a suitable *Quadratic Programming* formulation.

In this paper, we merge the first two layers, generating locomotion trajectories adopting the full kinematics of the robot and the centroidal dynamics. The approach follows

what presented by authors of [6]. Nevertheless, no prior knowledge is injected on the system to generate walking trajectories, but the whole-body motions result from a particular choice of cost function.

When planning locomotion trajectories, the definition of contacts plays a central role. Several strategies are available in literature, here roughly summarized in four categories.

Fixed contact sequence, timing and location. A common approach consists of assuming to know in advance where the contacts will be established and in which instant [7], [8], [9], [10], [11]. Such choice simplifies the planning problem, leading to a lower computational effort. However, they need to rely on external contact planners.

Predefined contact sequence. During locomotion, it can be assumed to know in advance the contact sequence. As an example, for a biped robot, it can be assumed that a contact with the left foot will be followed by another one with the right foot. In other words, the phases are predefined while the remaining quantities (like positions and timings) are an output of the planner [12], [13], [14]. By specifying a different set of equations depending on the contact state, the hybrid nature arising from the establishment of contacts is easily modeled. The time spent by each phase can be turned into an optimization variable. Nevertheless, in case several point contacts, the definition of the various phases could become intractable.

Mixed Integer Programming. Instead of receiving the contact sequence as input, it is possible to use integer variables to determine when a particular contact has to be considered active or not [15], [16]. This approach requires *Mixed Integer Programming* tools. While providing enhanced modeling capabilities, the exploitation of integer variables strongly affects the computational performances, especially in case several contacts are available. In addition, the availability of specialized solvers is limited.

Complementarity-free. Authors of [17], [18] presented an approach which allows simulating multi-body systems subject to contacts, without enforcing complementarity conditions directly. Equivalently accurate results are obtained by maximizing the rate of energy dissipation. Such approach can be used to generate complex movements [19], [20].

In this paper, we present a planner where neither contact sequences, locations or timings are fixed a priori. Additionally, we adopt a complementarity-free approach. Through a tailored parametrization of contacts, we impose complementarity conditions indirectly. Since the full robot kinematics is used, planned footsteps are within the robot work-space.

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 731540 (An.Dy).

¹ Dynamic Interaction Control, Istituto Italiano di Tecnologia, 16163 Genova, Italy (e-mail: name.surname@iit.it)

² Università degli Studi di Genova, DIBRIS, Genova, Italy

A. Notation

- The i_{th} component of a vector x is denoted as x_i .
- The transpose operator is denoted by $(\cdot)^\top$.
- \mathcal{I} is a fixed inertial frame with respect to (w.r.t.) which the robot's absolute pose is measured. Its z axis points against gravity, while the x direction points forward.
- Given a function of time $f(t)$ the dot notation denotes the time derivative, i.e. $\dot{f} := \frac{df}{dt}$. Higher order derivatives are denoted with a corresponding amount of dots.
- $\mathbf{1}_n \in \mathbb{R}^{n \times n}$ denotes the identity matrix of dimension n .
- $\mathbf{0}_{n \times n} \in \mathbb{R}^{n \times n}$ denotes a zero matrix.
- $S(\cdot)$ is the skew-symmetric operation associated with the cross product in \mathbb{R}^3 .
- The weighted L2-norm of a vector $v \in \mathbb{R}^n$ is denoted by $\|v\|_W$, where $W \in \mathbb{R}^{n \times n}$ is a weight matrix.
- ${}^A R_B \in SO(3)$ and ${}^A H_B \in SE(3)$ denote the rotation and transformation matrices which transform a vector expressed in the B frame into one expressed in A .
- $\mathbf{n}(\cdot), \mathbb{R}^3 \rightarrow \mathbb{R}^3$ returns the direction normal to the walking plane given the argument's x and y coordinates.
- $\mathbf{t}(\cdot), \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 2}$ returns two perpendicular directions normal to $\mathbf{n}(\cdot)$. The composition of $\mathbf{t}(\cdot)$ and $\mathbf{n}(\cdot)$, $[\mathbf{t}(\cdot) \mathbf{n}(\cdot)]$, defines the rotation matrix ${}^{\mathcal{I}}R_{plane}$.
- The function $h(p), \mathbb{R}^3 \rightarrow \mathbb{R}$ defines the distance between p and the walking surface.
- $e_1 := [1, 0, 0]^\top$, $e_2 := [0, 1, 0]^\top$ and $e_3 := [0, 0, 1]^\top$ denote the canonical basis vectors of \mathbb{R}^3 .
- ${}^D V_{A,D} \in \mathbb{R}^6$ is the relative velocity between frame A and D , whose coordinates are defined in frame D .
- $diag(\cdot), \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ is a function casting the argument into the corresponding diagonal function.

II. SYSTEM MODELING

A. Contact Interface Modeling

When performing a step, the foot can impact the ground in a not flat configuration, reducing the amount of contact wrenches obtainable from the ground. At the same time, toe-off motions can be used to increase the work-space available during double support phases [21]. Given these reasons, all the various contact configurations should be taken into account when planning step motions.

In order to reduce the complexity, it is possible to consider the foot as composed by a set of points, for example four points located at the corners of the foot [6], [14]. Thanks to this choice, the several contact configurations can be modeled independently, depending on the number of points in contact.

A pure force can be applied on each contact point. In case of four points, twelve variables define a six dimensional quantity, i.e. the resulting contact wrench acting on the foot. This is a drawback that will be addressed later in Sec. III-D.

B. Contacts Force Constraints

Define $i p \in \mathbb{R}^3$ as the i -th contact point location in an inertial frame \mathcal{I} , and $i f \in \mathbb{R}^3$ as the force exerted on that point. Such force is expressed on a frame located in $i p$ and with orientation parallel to \mathcal{I} . Since it results from the interaction of the foot with the ground, it is subject to

constraints. Being a reaction force, its normal component with respect to the walking ground is supposed to be non-negative. In particular, $\mathbf{n}(i p)^\top i f \geq 0$. Additionally, in order to avoid slippage, friction constraints should be satisfied:

$$\|\mathbf{t}(i p)^\top i f\| \leq \rho \mathbf{n}(i p)^\top i f \quad (1)$$

where ρ is the static friction coefficient.

C. Contact Parametrization

The contact force $i f$ applied to the i -th contact point is supposed to be not-null only if the point is in contact with the walking surface. This condition could be represented by the following equality:

$$h(i p) \mathbf{n}(i p)^\top i f = 0. \quad (2)$$

Such constraint can be difficult to be tackled in an optimization framework. This is due to the fact that the feasible set is only constituted by two lines, namely $h(i p) = 0$ and $\mathbf{n}(i p)^\top i f = 0$, which are intersecting in the origin. In this point, the constraint Jacobian gets singular, thus violating the *linear independence constraint qualification* (LICQ), on which most off-the-shelf solvers rely upon [22].

In order to avoid the complications related to Eq. (2), we adopt a simple parametrization. In particular, we assume to have full control over the derivative of both contact point positions and forces:

$$i \dot{p} = u_{i p} \quad (3a)$$

$$i \dot{f} = u_{i f}, \quad (3b)$$

where $u_{i p}, u_{i f} \in \mathbb{R}^3$ are control inputs. Then, we can impose Eq. (2) dynamically through the following constraints:

$$-M_f \leq u_{i f} \leq M_f \quad \text{if } h(i p) = 0, \quad (4a)$$

$$u_{i f} = -K_f i f \quad \text{if } h(i p) \neq 0. \quad (4b)$$

When the point is in contact, $u_{i f}$ is free to take any value in $[-M_f, M_f]$ with $M_f \in \mathbb{R}^3$ defining control bounds (Eq. (4a)). On the other hand, if the contact point is not on the walking surface, the control input makes the contact force decreasing exponentially (Eq. (4b)). Defining $\delta^*(i p)$ as a binary function such that

$$\delta^*(i p) = \begin{cases} 1 & \text{if } h(i p) = 0 \\ 0 & \text{if } h(i p) \neq 0 \end{cases}, \quad (5)$$

it is possible to write Eq. (4) as a set of two inequalities:

$$-K_f (1 - \delta^*(i p)) i f - \delta^*(i p) M_f \leq u_{i f} \quad (6a)$$

$$-K_f (1 - \delta^*(i p)) i f + \delta^*(i p) M_f \geq u_{i f}. \quad (6b)$$

Even if $\delta^*(i p)$ would require the adoption of integer variables, it is possible to use a continuous approximation, $\delta(i p)$, namely the hyperbolic secant:

$$\delta(i p) = \text{sech}(\mathbf{k}_h h(i p)), \quad (7)$$

where \mathbf{k}_h is a user-defined scaling factor. Since $\delta(i p) = 0$ only when $i p \rightarrow \infty$, Eq. (6) satisfies the LICQ condition.

Given Eq. (1), it is enough to apply Eq. (6) only to the force component normal to the ground: if it decreases to zero, also planar force components have to vanish.

Since contact points are not supposed to penetrate the walking ground, we can impose $h(i p) \geq 0$.

None of the constraints defined above could prevent the contact points to move on the walking plane when in contact. In fact, even if friction constraints defined in Eq. (1) are satisfied, the contact points are still free to move on the contact surface. Force and position variables are (almost) independent at this stage. It is possible to prevent planar motions when in contact by limiting the effect of the control input u_{ip} along the planar components:

$$\mathbf{t}(i p)^\top \dot{i p} = \tanh(\mathbf{k}_t h(i p)) [e_1 \ e_2]^\top u_{ip} \quad (8)$$

where $\mathbf{k}_t \in \mathbb{R}$ is a user-defined scaling factor. Eq. (8) multiplies the control input along the planar direction to zero when $h(i p)$ is null and, at the same time, it will reduce the velocity when the contact point is approaching the ground. It is possible to rewrite Eq. (8) as

$$\dot{i p} = \tau(i p) u_{ip}, \quad (9)$$

where the function $\tau(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ is defined as:

$$\tau(i p) = {}^{\mathcal{I}}R_{plane} \text{diag} \left(\begin{bmatrix} \tanh(\mathbf{k}_t h(i p)) \\ \tanh(\mathbf{k}_t h(i p)) \\ 1 \end{bmatrix} \right). \quad (10)$$

Note that, from now on, u_{ip} is assumed to be defined in *plane* coordinates. Thus, the normal component of the velocity is directly affected by $e_3^\top u_{ip}$. Also, it is necessary to bound this control input, $u_{ip} \in [-M_V, M_V]$, $M_V \in \mathbb{R}^3$, to properly exploit the effect of the hyperbolic tangent. Note that Eq. (8) allows avoiding the use of complementarity conditions along planar directions.

D. Contact Point Position Consistency

While each contact point is supposed to be independent from the control point of view, they all need to remain on the same surface and maintain a constant relative distance, since they belong to the same rigid body. At the same time, we want them to be within the workspace reachable by the robot legs. We can achieve both the objectives by enforcing the following constraint on each of the contact points:

$$i p = {}^{\mathcal{I}}H_{\text{foot}}^{\text{foot}} i p, \quad (11)$$

where ${}^{\text{foot}}i p$ is the (fixed) position of the contact point within the foot surface, expressed in foot coordinates. Here, the transformation matrix ${}^{\mathcal{I}}H_{\text{foot}}$ would depend on the base position ${}^{\mathcal{I}}p_B \in \mathbb{R}^3$, the base quaternion ${}^{\mathcal{I}}\rho_B \in \mathbb{H}$ and the joints configuration $s \in \mathbb{R}^n$, with n being the number of joints. As a consequence, the full kinematics of the robot is taken into consideration and the following dynamic equations have to be considered:

$${}^{\mathcal{I}}\dot{p}_b = {}^{\mathcal{I}}R_B {}^B v_{\mathcal{I},B} \quad (12a)$$

$${}^{\mathcal{I}}\dot{\rho}_B = u_\rho \quad (12b)$$

$$\dot{s} = u_s. \quad (12c)$$

Here ${}^B v_{\mathcal{I},B} \in \mathbb{R}^3$, $u_\rho \in \mathbb{R}^4$ and $u_s \in \mathbb{R}^n$ are considered control inputs defining the base linear velocity, the quaternion derivative and the joints velocity, respectively. More specifically, ${}^B v_{\mathcal{I},B}$ is the linear part of ${}^B V_{\mathcal{I},B} \in \mathbb{R}^6$ the left-trivialized (i.e. measured in body coordinates) base velocity.

E. Momentum Dynamics

In Sec. II-A, we consider the contact points as if they have the possibility of exerting a force with the environment. We can describe the effect of these contact forces through the momentum, or centroidal, dynamics. This choice is supported by the fact that the momentum dynamics depends only on the contact forces, their location and on the center of mass (CoM) position, $x_{\text{CoM}} \in \mathbb{R}^3$. Define $\bar{G}h \in \mathbb{R}^6$ as the robot total momentum, with $\bar{G}h = [\bar{G}h^p{}^\top \ \bar{G}h^\omega{}^\top]^\top$ where $\bar{G}h^p \in \mathbb{R}^3$ and $\bar{G}h^\omega \in \mathbb{R}^3$ are respectively the linear and angular momentum. This quantity is expressed in a frame oriented as the inertial frame \mathcal{I} , with the origin placed on the CoM position. Such frame is called $G[\mathcal{I}]$ or simply \bar{G} . The momentum dynamics has the following form:

$$\bar{G}\dot{h} = m\bar{g} + \sum_i \left[S(i p - x_{\text{CoM}}) \right]_i f \quad (13a)$$

$$\dot{x}_{\text{CoM}} = \frac{1}{m} (\bar{G}h^p) \quad (13b)$$

with m the total mass of the robot, $\bar{g} = [0 \ 0 \ -g \ 0 \ 0 \ 0]^\top$. We also need to make sure that the integrated CoM corresponds to the one obtained via the joint variables. This is done through an additional constraint:

$$x_{\text{CoM}} = \text{CoM}({}^{\mathcal{I}}p_B, {}^{\mathcal{I}}\rho_B, s) \quad (14)$$

where $\text{CoM}({}^{\mathcal{I}}p_B, {}^{\mathcal{I}}\rho_B, s)$ is the function mapping base pose and joint positions to the CoM position. While this constraint defines a link between the linear momentum and the joint variables, the same would not hold for the angular part. To this end, we can exploit the Centroidal Momentum Matrix [5] (J_{CMM}). In fact, the robot angular momentum can be defined as a function of the base and joints velocity:

$$\bar{G}h^\omega = [0_{3 \times 3} \ \mathbf{1}_3] J_{\text{CMM}} \nu \quad (15)$$

where $\nu = [{}^B V_{\mathcal{I},B} \ u_s]^\top$. Here, the base angular velocity ${}^B \omega_{\mathcal{I},B}$ can be substituted with the quaternion derivative through the map \mathcal{G} [23, Section 1.5.4], such that

$${}^B \omega_{\mathcal{I},B} = 2\mathcal{G}({}^{\mathcal{I}}\rho_B) u_\rho.$$

Some additional constraints can be considered:

$$x_{\text{CoM},z} \min \leq e_3^\top x_{\text{CoM}} \quad (16a)$$

$$-M_{h_\omega} \leq \bar{G}h^\omega \leq M_{h_\omega} \quad (16b)$$

Eq (16a) avoids solutions which would bring the CoM position too close or below the ground. Eq. (16b) provides an upper and lower bound $M_{h_\omega} \in \mathbb{R}^3$ to the angular momentum. These constraints avoid trajectories that would cause excessive motions or let the robot falling.

F. Feet Minimum Lateral Distance

While taking steps, we need to make sure that the robot legs do not collide with each other. Self collision constraints are usually hard to be considered and may slow down consistently the determination of a solution. A simpler solution consists in avoiding the left leg to be on the right of the other leg. Consequently, cross steps are forbidden. Let us consider a frame attached to the right foot with the positive y -direction pointing toward left. In this case, it is sufficient to impose the y -component of the ${}^r x_l$ (i.e. the relative position of the left foot expressed in the right foot frame) to be greater than a given quantity, i.e. $e_2^\top {}^r x_l \geq d_{\min}$.

III. TASKS

We present the set of tasks used to plan a walking trajectory. While constraints define the model and the control limitations, the tasks embed the planning objectives.

A. Contact point centroid position task

In order to make the robot moving toward a desired position, we minimize the L2 norm of the error between a point attached to the robot and its desired position in an absolute frame. In particular, we select the centroid of the contact points as target, thus avoiding to specify a desired placement for each foot:

$$\Gamma_{\#p} = \frac{1}{2} \|\#p - \#p^*\|_{W_{\#}}^2 \quad (17)$$

where $\#$ is the number of contact points in a single foot. Thus, we have $\#p = \frac{1}{2\#} \sum_{l,r} \sum_i^{\#} i p$ and $\#p^* \in \mathbb{R}^3$ is a user-defined reference value.

B. CoM linear velocity task

While walking, we want the robot to keep a constant forward motion. In fact, since the positions of the feet are not scripted, it may be possible to plan two consecutive steps with the same foot. By requiring a constant forward velocity, such phenomena can be avoided. This task is defined as:

$$\Gamma_{\bar{G}h^p} = \frac{1}{2} \|\bar{G}h^p - mv_G^*\|_{W_v}^2 \quad (18)$$

with $v_G^* \in \mathbb{R}^3$ a desired CoM velocity. The matrix W_v selects and weights the different directions separately.

C. Frame orientation task

While moving, we want a robot frame to be oriented in a specific orientation ${}^I R_{\text{frame}}^*$. In particular, we weight the distance of the rotation matrix ${}^I \tilde{R}_{\text{frame}} = {}^I R_{\text{frame}}^{*\top} {}^I R_{\text{frame}}$ from the identity. Having to express this task in vector form, we convert ${}^I \tilde{R}_{\text{frame}}$ into a quaternion (through the function `quat` which implements the Rodriguez formula) and weight its difference from the identity quaternion I_q . Namely:

$$\Gamma_{\text{frame}} = \frac{1}{2} \left\| \text{quat} \left({}^A \tilde{R}_{\text{frame}} \right) - I_q \right\|^2. \quad (19)$$

This corresponds to a simplified version of the quaternion difference metric listed in [24], under the assumption for `quat` to always return a quaternion with positive real value.

D. Force regularization task

While considering each single contact force in a foot as independent, they still belong to a single body part. Thus, we prescribe the contact forces in a foot to be as similar as possible, refraining from using partial contacts if not strictly necessary. This can be obtained through the following:

$$\Gamma_{\text{regf}} = \sum_{l,r} \sum_i^{\#} \frac{1}{2} \left\| i f - \frac{1}{\#} \sum_j^{\#} j f \right\|^2. \quad (20)$$

E. Joint regularization task

The joint configuration s is part of the optimization variables. In order to prevent the planner from providing solutions with huge joint variations, we introduce a regularization task for joint variables:

$$\Gamma_{\text{reg}s} = \frac{1}{2} \|\dot{s} + K_s(s - s^*)\|_{W_j}^2 \quad (21)$$

with s^* the desired joint configurations and W_j a weight matrix. The minimum of this cost is when $\dot{s} = -K_s(s - s^*)$, with $K_s \in \mathbb{R}^{n \times n}$. When this equality holds, joint values converge exponentially to their reference s^* . Hence, joint velocities and joint positions are regularized at the same time.

F. Swing height task

When performing a step, the swing foot clearance usually ensures some level of robustness with respect to ground asperity. Nevertheless, since the soil profile is supposed to be known in advance, a solution satisfying all the equations described in Sec. II may require the swing foot to be raised just few millimeters from the ground. In order to specify a desired swing height, we impose the following cost:

$$\Gamma_{\text{swing}} = \sum_{l,r} \sum_i^{\#} \frac{1}{2} \left\| \left(e_3^\top i p - {}_s h^* \right) [e_1 \ e_2]^\top u_{ip} \right\|^2. \quad (22)$$

It penalizes the distance between the z -component of each contact point position from a desired height ${}_s h^* \in \mathbb{R}$ when the corresponding planar velocity is not null. Trivially, this cost has two minima: when the planar velocity is zero (thus the point is not moving) or when the height of the point is equal to the desired one.

IV. OPTIMAL CONTROL PROBLEM

Given the set of equations listed in Sec. II and the tasks described in Sec. III it is possible to define an optimal control problem, whose complete formulation is presented below. Here, the vector \mathbf{w} contains the set of weights defining the relative “importance” of each task.

A. Problem definition

$$\begin{aligned} & \underset{\mathcal{X}, \mathcal{U}}{\text{minimize}} \quad \left[\Gamma_{\#p} \ \Gamma_{\bar{G}h^p} \ \Gamma_{\text{frame}} \ \Gamma_{\text{regf}} \ \Gamma_{\text{reg}s} \ \Gamma_{\text{swing}} \right]^\top \mathbf{w} \\ & \text{subject to :} \end{aligned}$$

- Dynamical Constraints

$${}_i\dot{f} = u_{if} \quad \forall \text{ contact point} \quad (23a)$$

$${}_i\dot{p} = \tau({}_ip)u_{ip} \quad \forall \text{ contact point} \quad (23b)$$

$$\bar{G}\dot{h} = m\bar{g} + \sum_i \begin{bmatrix} \mathbb{1}_3 \\ S({}_ip - x_{\text{CoM}}) \end{bmatrix} {}_if \quad (23c)$$

$$\dot{x}_{\text{CoM}} = \frac{1}{m} (\bar{G}h^p) \quad (23d)$$

$$\mathcal{I}\dot{p}_B = \mathcal{I}R_B^B v_{\mathcal{I},B} \quad (23e)$$

$$\mathcal{I}\dot{\rho}_B = u_\rho \quad (23f)$$

$$\dot{s} = u_s \quad (23g)$$

- Equality Constraints

$${}_ip = {}^A H_{\text{foot}}^{\text{foot}} {}_ip \quad \forall \text{ contact point} \quad (24a)$$

$$x_{\text{CoM}} = \text{CoM}(\mathcal{I}p_B, \mathcal{I}\rho_B, s) \quad (24b)$$

$$\bar{G}h^\omega = [0_{3 \times 3} \quad \mathbb{1}_3] J_{\text{CMM}} \begin{bmatrix} {}^B v_{\mathcal{I},B} \\ 2\mathcal{G}(\mathcal{I}\rho_B)u_\rho \\ u_s \end{bmatrix} \quad (24c)$$

$$\|\mathcal{I}\rho_B\|^2 = 1 \quad (24d)$$

- Inequality Constraints

$$\mathbf{n}({}_ip)^\top {}_if \geq 0 \quad (25a)$$

$$\|\mathbf{t}({}_ip)^\top {}_if\| \leq \rho \mathbf{n}({}_ip)^\top {}_if \quad (25b)$$

$$u_{if} \geq -K_f (1 - \delta({}_ip)) {}_if - \delta({}_ip)M_f \quad (25c)$$

$$u_{if} \leq -K_f (1 - \delta({}_ip)) {}_if + \delta({}_ip)M_f \quad (25d)$$

$$-M_V \leq u_{ip} \leq M_V \quad (25e)$$

$$h({}_ip) \geq 0 \quad (25f)$$

$$e_2^\top x_l \geq d_{\min} \quad (25g)$$

$$x_{\text{CoM},z} \min \leq e_3^\top x_{\text{CoM}} \quad (25h)$$

$$-M_{h_\omega} \leq \bar{G}h^\omega \leq M_{h_\omega} \quad (25i)$$

Here, the state variables \mathcal{X} are those derived in time, \mathcal{U} all the others. More specifically:

$$\mathcal{X} = \begin{bmatrix} {}_if \\ {}_ip \\ \vdots \\ \bar{G}h \\ x_{\text{CoM}} \\ \mathcal{I}p_B \\ \mathcal{I}\rho_B \\ s \end{bmatrix}, \quad \mathcal{U} = \begin{bmatrix} u_{if} \\ u_{ip} \\ \vdots \\ {}^B v_{\mathcal{I},B} \\ u_\rho \\ u_s \end{bmatrix} \quad (26)$$

where the symbol \vdots represents the repetition of variables for each contact point.

The optimal control problem is solved using a Direct Multiple Shooting method [22]. The system dynamics, defined in Eq. (23), is discretized adopting an implicit trapezoidal method with a fixed integration step. The corresponding optimization problem is solved thanks to Ipopt [25].

The walking trajectories are generated using the Receding Horizon Principle [26], adopting a fixed prediction window.

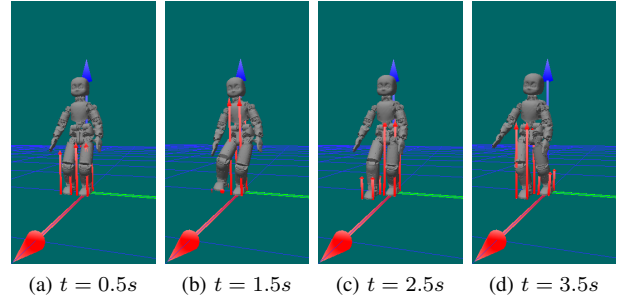


Fig. 1: Snapshots of the generated walking motion. The red arrows indicate the force required at each contact point scaled by a factor of 0.01.

B. Considerations

The optimal control problem described in Sec. IV-A is built such that (almost) no constraint is task specific. As a consequence, it is particularly important to define the cost function carefully since the solution will be a trade-off between all the various tasks. On the other hand, the detailed model of the system allows achieving walking motions without specifying a desired CoM trajectory or by fixing the angular momentum to zero. Nevertheless, due to the limited time horizon, it is better to prevent the solver from finding solutions which would bring the robot to unfeasible states in future planner iterations. To this end, Eq. (16a) and Eq. (16b) have been added, using reasonably large bounds.

Another possible effect resulting from the application of the Receding Horizon principle, is the emergence of “procrastination” phenomena. Due to the moving horizon, the solver may continuously delay in actuating motions, since the task keeps being shifted in time. A simple fix to this phenomena is to increase the weights \mathbf{w} with time, such that it is more convenient to reach a goal position earlier.

Finally, given that the problem under consideration is non-convex, the optimizer will find a local minimum. This may result in a sub-optimal solution for the given tasks.

During the first iteration, the solver is initialized by simply translating the whole robot in the desired position. In successive iterations, the solver is warm-started with the solution previously computed.

V. RESULTS

The optimal control problem described in Sec. IV is used to generate a walking motion. The integration step is set to 100ms, while the prediction horizon is 2s. After each iteration of the MPC controller, the previously computed state is used as a feedback. Regarding the scaling factors, we use $\mathbf{k}_h = 300$ and $\mathbf{k}_t = 10.0$. They appeared to be reasonable values for having meaningful simulations of contacts. In addition, these values are robot and ground independent, since they depend only on the position of a contact point with respect to the ground.

The trajectories have been generated using the iCub humanoid [27] robot model on a 7th generation Intel® Core

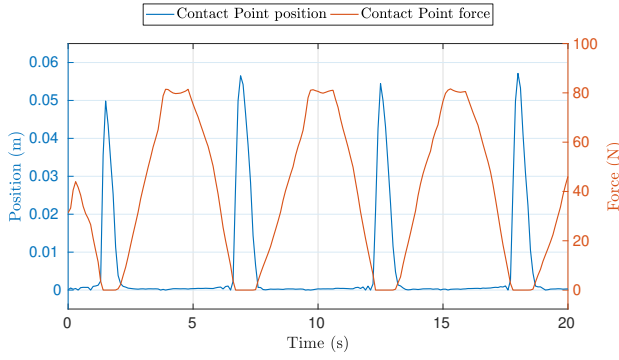


Fig. 2: Normal force and normal position of a contact point of the right foot plotted together.

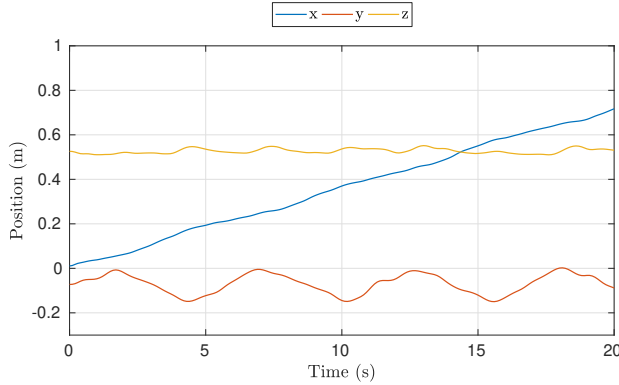


Fig. 3: Planned CoM position.

i7@2.8GHz laptop. We assume the ground to be flat, while we control 23 of the robot joints. For each foot, we consider four contact points located at the vertexes of the rectangle enclosing the robot foot. Concerning the references, the desired position for the centroid of the contact points is moved 10cm along the walking direction every time the robot performs a step. A simple state machine, where the reference is moved as soon as a step is completed, is enough to generate a continuous walking pattern. The speed is modulated by prescribing a fixed desired CoM forward velocity equal to 5cm/s.

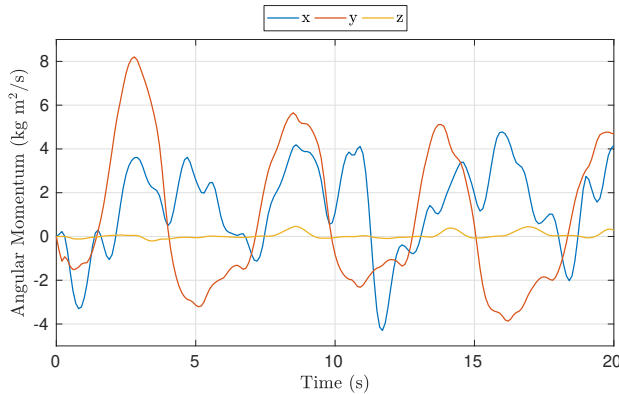


Fig. 4: Planned angular momentum.

Figure 1 shows some snapshots of the first generated step. Also, it can be observed the effect of the contact parametrization described in Sec. II-C from Fig. 2. The normal force decreases to zero as soon as the foot starts leaving the ground, and then it grows again at touchdown. It is possible to recognize the different walking phases, even though they are not planned a priori.

Figure 3 presents the planned CoM position. Here, it is possible to notice that x position grows at a constant rate. This is a direct consequence of the task on the CoM velocity. Figure 4 shows the planned angular momentum, which is not fixed to zero. Although it is limited to $10 \text{ kg m}^2/\text{s}$, such limit is never reached. Similarly, the bound on the CoM height, $x_{\text{CoM},z \text{ min}}$, is set to half of the initial robot height, but such constraint is never activated.

It is worth stressing that none of the tasks described above define how and when to raise the foot. By prescribing a reference for the centroid of the contact points and by preventing the motion on the contact surface, swing motions are planned automatically. Nevertheless, this advantage comes with a cost. It is difficult to define a desired swing time and, more importantly, the relative importance of each task, i.e. the values of w , must be chosen carefully. During experiments, we adopted an incremental approach. We added the tasks one by one, starting from $\Gamma_{\#p}$ and then we gradually refine the walking motion by tuning a cost at a time.

VI. CONCLUSIONS

This paper presents a planner capable of generating walking trajectories using a minimal set of references. It considers the centroidal momentum of the robot and its full kinematics to plan dynamically consistent step motions. The modelling of contacts makes use of a novel parametrization approach, allowing to model the interface between the robot and the ground with a set of continuous equations. Currently, this model does not consider impacts nor contact sliding. The consideration of slip-turn motions [28] is left as future work.

The results have to be considered as an initial validation of the generated trajectories when the iCub model is adopted. In particular, it is shown that walking trajectories can emerge by specifying a moving reference for the centroid of the contact points and the desired CoM velocity only.

The planner considers relatively large time-steps. This enables the insertion of another control loop at higher frequency, whose goal is to stabilize the planned trajectories. As a future work, we will consider connecting this planner to the whole-body controller presented in [2].

The main bottleneck is represented by the computational time. A single planner iteration may take from slightly less than a second to more than a minute. This prevents an online implementation on the real robot. On the other hand, the continuous formulation of the optimal control problem allows the application of techniques, like those presented in [29], [30], which do not rely on the transcription to a non-linear programming problem.

REFERENCES

- [1] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the darpa robotics challenge," *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.
- [2] S. Dafarra, G. Nava, M. Charbonneau, N. Guedelha, F. Andradel, S. Traversaro, L. Fiorio, F. Romano, F. Nori, G. Metta, *et al.*, "A control architecture with online predictive planning for position and torque controlled walking of humanoid robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.
- [3] G. Romualdi, S. Dafarra, Y. Hu, and D. Pucci, "A benchmarking of dcm based architectures for position and velocity controlled walking of humanoid robots," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.
- [4] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.
- [5] D. E. Orin and A. Goswami, "Centroidal momentum matrix of a humanoid robot: Structure and properties," *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 653 – 659, 2008.
- [6] 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*, pp. 295–302.
- [7] P. Fernbach, S. Tonneau, and M. Taïx, "Croc: Convex resolution of centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.
- [8] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 874–880.
- [9] 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)*. IEEE, 2016, pp. 1366–1373.
- [10] M. De Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–10, 2010.
- [11] S. Dafarra, F. Romano, and F. Nori, "A receding horizon push recovery strategy for balancing the icub humanoid robot," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2017, pp. 297–305.
- [12] A. W. Winkler, D. C. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, pp. 1560–1567, July 2018.
- [13] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, "A versatile and efficient pattern generator for generalized legged locomotion," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3555–3561.
- [14] S. Caron and Q.-C. Pham, "When to make a step? tackling the timing problem in multi-contact locomotion by topp-mpc," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 522–528.
- [15] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*.
- [16] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2018.
- [17] E. Todorov, "A convex, smooth and invertible contact model for trajectory optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1071–1076.
- [18] E. Drumwright and D. A. Shell, "Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation," in *Algorithmic foundations of robotics IX*. Springer, 2010, pp. 249–266.
- [19] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.
- [20] 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*. IEEE, 2012, pp. 4906–4913.
- [21] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. Pratt, "Straight-leg walking through underconstrained whole-body control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–5.
- [22] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. Siam, 2010, vol. 19.
- [23] B. Graf, "Quaternions and dynamics," Available at <https://arxiv.org/pdf/0811.2889.pdf>, 2008.
- [24] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.
- [25] 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, 2006.
- [26] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Transactions on automatic control*, vol. 35, no. 7, pp. 814–824, 1990.
- [27] L. Natale, C. Bartolozzi, D. Pucci, A. Wykowska, and G. Metta, "icub: The not-yet-finished story of building a robot child," *Science Robotics*, vol. 2, no. 13, 2017.
- [28] K. Miura, F. Kanehiro, K. Kaneko, S. Kajita, and K. Yokoi, "Slip-turn for biped robots," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 875–887, 2013.
- [29] M. Neunert, M. Stäubli, M. Gifftaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.
- [30] F. Farshidian, E. Jelavic, A. Satapathy, M. Gifftaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 577–584.