2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Fast multi-contact whole-body motion planning with limb dynamics

Jonathan Arreguit[1], Salman Faraji[1] and Auke Jan Ijspeert[1]

***Abstract*— We present a new method for multi-contact motion planning which efficiently encodes internal dynamics of the robot without needing to use full models. Our approach is based on a five-mass model which is formulated by Cartesian points instead of joint angles. We solve direct optimization problems which include distance constraints between these points, Newtonian equations and integration constraints. We consider a given rhythm of contact switches but leave the phase-timings and contact positions free inside the optimization to provide more flexibility. Due to simpler equations and sparser problem structures, we can achieve very short optimization times in the order of few hundred milliseconds, which make the method suitable for application of online model predictive control. Aside from contact position and time adjustment properties, we can include precise foothold regions and synthesize dynamic motions by taking internal dynamics and momentums into account.**

I. INTRODUCTION

Planning multi-contact locomotion in unstructured environments involves various challenging tasks, ranging from perception, environment modeling, motion planning and compliant control. Despite well-established techniques for different blocks in this complex pipeline [1], [2], the planning part is yet time-consuming and computationally slow compared to human locomotion. The underlying complexity has different folds: limited foothold locations, robot's floating-based dynamics, unilateral frictions, reachability constraints, collisions and planning the sequence of contacts. These complexities produce a very complex mathematical problem in case all mechanical details are considered. Solving such problem for a simple task of sitting down on a chair may, in fact, take up to a few hours [3], [4]. However, it is possible to speed-up this problem with different simplifications and achieve faster planning speeds while compromising for certain features. Taking inspiration from state of the art algorithms, we propose a simple formulation based on a previously developed technique [5] based on molecular modeling, that can improve dynamics of the motion in an efficient and meaningful way. Before introducing the proposed method, we review the literature by identifying essential challenges involved and discussing possible improvements.

Manuscript received: June, 1, 2018; Accepted X.
The first two authors contributed equally to this work. This paper was recommended for publication by Editor Nikos Tsagarakis upon evaluation of the Associate Editor and Reviewers' comments. This work was funded by the WALK-MAN project.
[1]Biorobotics Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL) jonathan.arreguitoneill@epfl.ch
Digital Object Identifier (DOI): see the top of this page.

A. The curse of dimensionality

Humanoids or multi-legged robots have floating-based dynamics described by $(n+6)$ states where n refers to the number of joints and 6 denotes the dimension of global states which describe the position and orientation of the robot in world-frame. Finding time-trajectories for all these degrees of freedom produces a high-dimensional planning problem. One of the simplifications widely popular in the literature is to use single-mass models which could approximate the overall behavior in slow motions [6]. In these models, one can optimize all contact forces to produce a physically meaningful Center of Mass (CoM) trajectory. However, these forces produce a moment on the body according to Centroidal dynamics [7]. Enforcing the resulting angular momentum rate to zero is possible, but it is more natural to associate it with whole-body inertia like [8] (shown in Fig. 1A). This simplification is close to reality if the legs are light-weight compared to the main body, e.g., in quadruped robots like ANYmal [8]. However, in many humanoid robots including Coman [9], the legs and arms are comparably heavy.

B. Simplified dynamics, full kinematics

Simplified models can speed-up calculations by reducing problem dimensions. However, they possibly over-simplify fast motions and especially dynamic effects induced by heavy limbs. To mitigate this problem, it is possible to plan end-effector and CoM trajectories in Cartesian space for a simplified model, and match it kinematically with the full model (shown in Fig. 1B). Mordatch et al. [10] converted such abstract trajectories to joint angles via inverse kinematics. By feeding the resulting joint angles into a full dynamics model, they were able to find the whole-body momentum rate which was then constrained to be equal to the abstract momentum rate. In their approach, there is no need to explicitly include individual joint angles and contact forces in the optimization. These variables are directly calculated from the Cartesian trajectories through inverse kinematics and inverse dynamics respectively. Mordatch et al. used soft constraints to enforce dynamic equations originally [10]. However, Dai et al. included the contact forces explicitly and used hard constraints [11]. These methods assume no limitation for the joint torques which significantly reduces the number of inequality constraints needed. They also need very efficient kinematic solvers, possibly in closed-form, which could be iteratively used inside bigger optimizations. Herzog et al. also used a similarly decoupled dynamic-kinematic approach and achieved faster optimizations with more efficient solvers [12].

978-1-5386-7283-9/18/$31.00 ©2018 IEEE

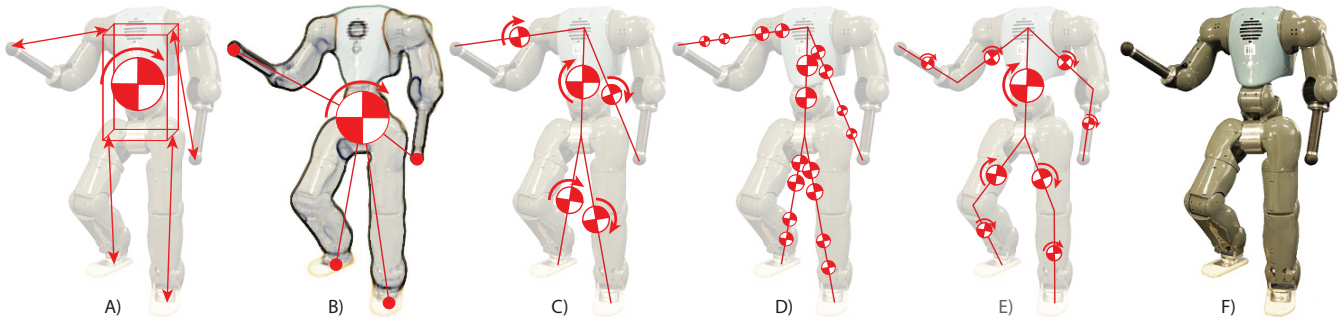

Fig. 1: Different template models used in the literature: A) A single mass and inertia with kinematic length constraints [8], B) Centroidal dynamics with full kinematics [11], C) the proposed five-mass model with stretchable limbs, D) same as previous, but representing inertias with two masses, E) a nine-mass model with elbows and knees and F) the full robot COMAN [9]. The borders in B) indicate that the simplified Centroidal dynamics match with the full system by using full kinematics.

C. Trajectory parameterization

Due to nonlinearity, it is very common to use splines to encode trajectories over time. This can be done for joint positions [4], Cartesian positions [10] and even contact forces [8]. However, it requires binding trajectories together at regular intervals via kinematic and dynamic constraints [8]. An alternative is to define all variables at every time-sample and link them via integration constraints [11]–[13]. This gives more flexibility in the design of trajectory shapes but increases the dimensionality of the problem. A vast part of literature also aims at optimizing transition postures only, which further simplifies the spline method [14]–[17]. Although these methods may potentially be faster from a computational point of view, they are prone to producing quasi-static motions.

D. Fixed contact sequences

Given reachability constraints and environmental complexities, finding a kinematically and dynamically feasible set of contacts is not always trivial. The Linear Inverted Pendulum (LIP) model [18] can provide closed-form equations which enable for online Model Predictive Control (MPC) in flat-ground walking conditions [19], [20], but it is too limited for multi-contact locomotion. Without considering dynamical consistency, it is possible to perform a faster kinematic search for footstep locations and orientations in a first stage. Rapidly-exploring random trees can achieve reasonable plans with simple distance constraints [14], [17] and energy-based cost functions [21], but these approaches are also limited to bipedal walking. A library of recorded human motions can give a good initial guess [22]. However, the generalization to different environments remain a challenge. Hauser et al. proposed a graph search method in the first stage to find a path of (kinematically) adjacent nodes spanning between the start and end points [6]. The second stage in their algorithm then fixed dynamic consistencies. It is possible to link them back after computing them separately [16], but at the cost of increasing computations in mixed-integer optimizations. However, extraction of convex hulls in the environment to represent supporting surfaces would allow for contact adjustment and result in faster optimization problems [15]. Given a simple environment model represented by such convex hulls, one can use mixed-integer convex optimization to find bipedal walking footholds [23] or more complex multi-contact motions [24]. The approach presented in [23] tackles kinematics and dynamics separately in different optimization stages while [24] combines both in a single mixed-integer optimization. Depending on the number of existing support surfaces, the discrete search problem may grow in size exponentially.

E. Emergent contact sequence

To find more elegant contact sequences with dynamic consistency and flexible timing, Mordatch et al. introduced the Contact-Invariant Optimization (CIO) approach [10]. To explain it briefly, consider a single contact point of the robot. Without any assumption on the phase sequence or timing for this point, it is possible to minimize the multiplication between the present contact force and the distance (to the adjacent environment surface) at each instance of time. When the distance is zero, the algorithm is allowed to increase the contact force while it is forced to zero when the distance is non-zero (i.e., a swing motion). Both contact positions and forces are indeed explicit optimization variables. While this method was initially formulated with soft constraints in [10], Posa et al. used complementarity constraints and formulated the problem with hard constraints which allowed for sliding contacts as well [13]. This formulation was later used in [11] to produce arbitrary contact sequences for different tasks on the Atlas robot. A more restrictive alternative to this flexible approach is to look at each contact individually. The sequence of each contact is fixed, *swing* and then *stance* and vice versa. Winkler et al. assumed the same number of switches for each contact and hard-coded the trajectories in each phase with splines [8]. This approach considers variable phase durations per contact point which sum up to the same total motion duration determined beforehand. This approach allows for arbitrary contact sequences like the CIO method, but with a predetermined number of contact switches.

F. Environment complexity

We mentioned that representing the environment with simple convex-hulls is useful in discrete search methods [17],

[23]. The approach presented by Winkler et al. [8] combines all these surfaces and approximates them with a smooth height-map which allows for continuous optimizations. However, in case of large gaps or restricted footholds, a good initial guess is needed to avoid converging to local minima. Mordatch et al. calculated a soft distance-to-surface measure in the CIO method to consider environment complexities, but their simulations were restricted to flat-ground or stair-climbing cases and not traversing huge gaps. A similar limitation also appears in the approach of [11] and [13] where the contact sequence in restricted foothold cases or monkey-bar scenarios might be pre-determined. Therefore, the CIO approach and the alternative proposed in [8] are probably powerful in finding emergent behaviors in simple environments, but more prone to local minima in complex terrains.

G. Rooms for improvement

As discussed so far, popular approaches proposed in the literature might share certain positive aspects, but they might be limited from other perspectives. An ideal planner should be able to re-plan the motion online within a few milliseconds and ideally adjust the timing, contact locations and even contact surfaces which involve discrete decisions. Besides, it is expected to generate dynamic and fast motions by considering robot dynamics. Most of the methods proposed in the literature cannot reach such time performance due to many reasons. The mixed-integer nature of the problem is probably the most restricting factor. Additionally, although simplified models speed up the problem, they cannot produce dynamic motions for robots with heavy limbs unless the full dynamics model [3] or the kinematic model is used [10]–[12]. Consider Fig. 1 which summarizes the important modeling methods used in literature. The single mass-inertia model of Winkler et al. [8] is shown in Fig. 1A together with the decoupled dynamic-kinematic approach of [10]–[12] shown in Fig. 1B. These approaches speed-up simulations considerably compared to using the full model [3] shown in Fig. 1F. However, it is possible to elaborate the simplified model slightly to better account for internal body dynamics in fast motions.

H. Contributions, novelties

In this paper, by splitting the modeling and discrete search challenges apart, we only focus on the simplified model and propose a method to improve the performance from this perspective. We restrict the study to finding dynamic trajectories and use a predetermined contact sequence. We also use simple collision avoidance techniques and do not consider possible knee/elbow contacts with the environment. We propose a simple five-mass model with a torso and four limbs to represent different body parts shown in Fig. 1C. Using a previously developed optimization technique [5], we set up a motion planning problem with contact forces, phase times, end-effector, pelvis and mid-shoulder positions as main variables. These trajectories are bound together with dynamic equations, geometric length constraints, reachability constraints and contact friction cones. We show that our method can plan dynamic multi-contact motions in the order of few hundred milliseconds with adjustable contact location and phase timing properties. Although we use a predefined contact sequence which could be the outcome of human motion libraries [22], random search [14], [17], [21] or mixed-integer problems [23], the novelty of our method lies in the new simplified model which could be used in many other optimization problems as well. It can efficiently describe internal robot dynamics while allowing for implementation of reachability constraints. Despite using time-integration to parametrize trajectories instead of splines which are probably more efficient, we show that we can achieve fast optimization times comparable to single-mass simplified models. The next section describes the model and optimization formulation in details. Next, we demonstrate results over different planning scenarios and analyze the performance from various perspectives. Finally, we conclude with a discussion of advantages and limiting factors as well as future directions for potential improvements.

II. PROBLEM FORMULATION

We start this section by introducing our simplified mechanical model. This model is directly formulated with Cartesian position variables instead of joint angles. In a previous work [5], we discussed that such modeling technique is popular in computational chemistry to calculate complex molecular conformations. Despite different scales of mechanical forces in humanoids versus molecules, the overall geometries have important similarities, especially the tree-like topology and the fixed-length constraints between points. In [5], we formulated a posture optimization problem in which world-frame positions of skeletal joints were the main variables instead of actual joint angles. We considered a mass and inertia for each limb which spans between two adjacent Cartesian points, constrained to have a fixed length inside the optimization. Compared to joint-space models, this new formulation is sparse and mathematically less complicated. Therefore, optimizations of arbitrary multi-contact postures could be done with better convergence properties and up to ten times faster with the new formulation [5].

In the present work, however, we use a simpler five-mass model (shown in Fig. 1C) for motion planning. This model is derived from a nine-mass model similar to the humanoid anatomy (Fig. 1E). Also, the inertias we assume for each body segment are equivalent to splitting the mass into two half-masses shown in Fig. 1D. We consider stretchable limbs in the robot where masses also change location along the limb proportionally. More formally, we assume there are two ideal prismatic (massless) actuators on both sides regulating the entire limb length proportionally. We use a fixed length for the torso segment and restrict limbs to a certain range to approximate the effective workspace. The upper limit of this range encodes reachability constraint while the lower limit prevents self-collision.

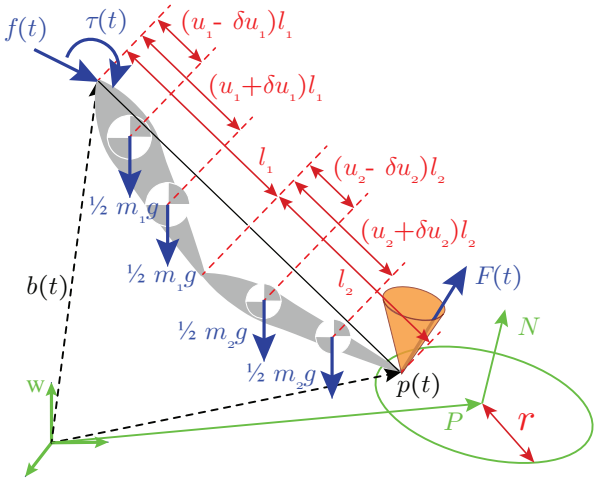

Fig. 2: Demonstration of a single limb in the robot with internal forces $f(t)$ and $\tau(t)$, and external forces $F(t)$. This limb could be one arm for example without hands, where the vector $b(t)$ is shoulder position and the vector $p(t)$ is contact position both expressed in the world-frame w. The surface can be approximated by a circle at position P, normal vector N and radius r. The friction cone of this contact is also described by a coefficient μ.

A. Model formulas

To derive the equations, consider Fig. 1D. As mentioned earlier, we consider mass and inertial for both limb segments (upper arm and forearm as well as thigh and shank segments). Since in human [25] and our robot COMAN [19], transversal moments of inertia in each limb segment are much smaller than sagittal and longitudinal, and since the last two components are almost the same, we approximate each limb segment by two half-masses. This is equivalent to considering only two nonzero diagonal elements in the inertia matrix. Consider one limb of the robot shown in Fig. 2 which attaches to the main body at point $b(t)$ (representing the pelvis or the mid-shoulder point), and creates contact with the environment at point $p(t)$. To improve readability of the paper, we avoid using indices for individual limbs. In our simplified model, we only consider contact forces $F(t)$ and not wrenches. We leave this actuation possibility (realized by ankle and wrist joints) to a lower-level inverse-dynamics algorithm for a better track of the simplified trajectories. We also assume point-contacts which ideally prevent transversal moments. However, the proposed optimization framework is ideally able to consider all these moments and include constraints to ensure their feasibility with respect to the foot/hand size. Coming back to Fig. 2, further assume that internal forces $f(t)$ and torques $\tau(t)$ are applied to the limb at point $b(t)$. We denote segment masses by m, lengths by l, relative mass positions by $0 \leq u \leq 1$ (refer to Fig. 2), and sagittal and longitudinal inertias by i. Splitting the half-masses by δul around the actual mass location, one can obtain:

$$\delta ul = \sqrt{\frac{i}{ml^2}} \tag{1}$$

Individual limb segments are identified by subscripts 1 and 2. It is obvious that the overall limb length L and mass M are:

$$L = l_1 + l_2, \quad M = m_1 + m_2 \tag{2}$$

The overall relative limb mass location U is:

$$U = \frac{m_1 u_1 l_1 + m_2 (l_1 + u_2 l_2)}{(l_1 + l_2)(m_1 + m_2)} \tag{3}$$

and the overall inertia around this mass is calculated by:

$$\begin{aligned} I &= \frac{m_1}{2}(UL - (u_1 - \delta u_1) l_1)^2 \\ &+ \frac{m_1}{2}(UL - (u_1 + \delta u_1) l_1)^2 \\ &+ \frac{m_2}{2}(UL - l_1 - (u_2 - \delta u_2) l_2)^2 \\ &+ \frac{m_2}{2}(UL - l_1 - (u_2 + \delta u_2) l_2)^2 \end{aligned} \tag{4}$$

which could be described in the simplest form as:

$$I = i_1 + i_2 + \frac{m_1 m_2}{m_1 + m_2}((1 - u_1) l_1 + u_2 l_2)^2 \tag{5}$$

Now, having L, M, U and I as a function of individual segment properties, we can define $x(t) = p(t) - b(t)$ and write Newton equations:

$$\begin{aligned} F(t) + f(t) + M[g - U\ddot{x}(t) - \ddot{b}(t)] &= 0 \\ \tau(t) + x(t) \times [(1 - U)F(t) - Uf(t) - \frac{I}{|x(t)|^2}\ddot{x}(t)] &= 0 \end{aligned} \tag{6}$$

These equations describe normal rigid body dynamics. We use $|x(t)| = L$ which is valid for the torso, since such constraint is explicitly included in the optimization. For other stretchable limbs, although $|x(t)|$ is variable in the range $L_{min} \leq |x(t)| \leq L$, we still use $|x(t)| = L$ which is an approximation, but simplifies the symbolic equations considerably.

The entire robot model is composed of five body segments (torso + four limbs) whose properties are calculated as described before. For each segment, we have to include one equality or inequality in the optimization to control the length. However, endpoint positions are all decision variables in the optimization to be found. Note that in the pelvis and mid-shoulder points, the sum of internal forces and torques are zero. Therefore, we can combine all dynamic equations of the limbs (i.e., the set of equations (6) for each limb) and obtain the six usual Centroidal dynamic equations. In the single mass models used in [10]–[12], CoM and end-effector positions, as well as contact forces, are optimization variables. The extra computational cost of our model is only replacing the single CoM position with two pelvis and mid-shoulder positions. The rest of six Centroidal dynamic equations remain present. The advantage of having individual limb dynamics is that end-effector movements can now directly influence the six equations, whereas the approaches in [10]–[12] implicitly create this linkage between the inverse kinematics and the full dynamical model. In these approached, more precision could be obtained at the cost of more calculations. We avoid including all the model details in this section for the sake of readability and only provide our optimization problem in abstract form.

B. Optimization setup

As mentioned, we use a predefined contact sequence in our planning scenarios. The sequence naturally starts from a given state and ends in a terminal posture. The environment is also modeled by contact surfaces given in circular shapes with certain positions P, radius r and normal vectors N similar to [14] (refer to Fig. 2). The sequence also involves the rhythm of contact changes while directly assigning contact points in the robot to those contact surfaces. The optimization can then find dynamically consistent end-effector position and force trajectories, proper contact locations inside the surfaces and variable phase durations. We divide the motion into M phases in which both the upper and lower limbs either perform a swing motion or establish a contact and receive supporting forces from the environment. Each phase is divided into K sub-phases which aim at providing a good resolution for dynamic and kinematic constraints. Sub-phases (indexed by i where $1 \leq i \leq MK$) have variable durations h_i encoded as optimization decision variables which all sum to MT. Here, the parameter T represents an average phase duration and determines the frequency of motion. Exact phase durations h_i are not forced to be exactly equal to T however. End-effector, pelvis and mid-shoulder positions p_i^j and velocities $\dot{p}_i^j$ (where $1 \leq j \leq 6$) are also decision variables at the beginning of each sub-phase. Therefore, trajectories are formed by time-integration constraints over individual sub-phases which form a direct optimization setup [13]. The contact points are free to move inside the contact circles, and contact forces F_i^j should lie inside friction cones. We also define accelerations by differentiation of velocities $h_i \ddot{p}_i^j = \dot{p}_i^j - \dot{p}_{i-1}^j$. Altogether, the optimization setup is as follows:

$$
\begin{aligned}
&\min_{h_i, p_i^j, \dot{p}_i^j, F_i^j} \sum_{i,j} h_i \ddot{p}_i^{j2} + h_i F_i^{j2} + H(p_i^j) && \text{(objective)} \\
&s.t. \\
&\textstyle\sum h_i = MT && \text{(motion duration)} \\
&\Gamma(p_i, \ddot{p}_i, F_i) = 0 && \text{(dynamics)} \\
&L_{min} \leq L(p_i) \leq L_{max} && \text{(kinematics)} \\
&G(p_0, \dot{p}_0) = S_0 && \text{(initial condition)} \\
&\text{Contact phase:} \\
&p_i^j \in S(P_k, N_k, r_k) && \text{(circular surfaces)} \\
&F_i^j \in F(P_k, N_k, \mu_k) && \text{(friction cones)} \\
&\dot{p}_i^j = 0 && \text{(no sliding)} \\
&\text{Swing phase:} \\
&F_i^j = 0 && \text{(free swing)} \\
&h_i \dot{p}_i = p_i - p_{i-1} && \text{(time integration)} \\
&C(p_{i+[\frac{K}{2}]}) \geq 0, \quad i \bmod K = 0 && \text{(collision)} \qquad (7)
\end{aligned}
$$

Here, S_0 is the initial state. The vectors p_i and F_i contain individual quantities p_i^j and F_i^j. The function $G(.)$ maps Cartesian points to a given state and $L(.)$ encodes length constraints. The vectors L_{min} and L_{max} simply contain limb length constraints. For the torso length, the corresponding elements are set to be equal to implement an equality constraint. The function $\Gamma(.)$ encodes dynamic equations (6) for each sub-phase i, $F(.)$ denotes friction cones and $S(.)$ denotes contact circles. In practice, we define accelerations by a direct differentiation of velocities and thus, exclude them from the list of optimization decision variables. Zero velocities, accelerations and forces are also defined parametrically to make the optimization faster.

We normalize forces and accelerations by total mass and gravity in the objective function. The function $H(p) = -\hat{z}^T p$ is only applied to the mid-shoulder point to lift the robot up and make the motion more energy optimal. The vector $\hat{z}$ denotes world-frame vertical direction. Since we do not have knee joints in our five-mass model, we cannot minimize knee torques to naturally lift the robot up [5]. In a previous work [5], where we optimized postures with a more complete model that had knee joints, we showed that such lifting term could produce similar postures compared to a setup in which only a torque-minimizing objective was used. Alternative approaches to avoid collapsing are constraining end-effector positions in certain limited regions with respect to the body [8], relying on initial trajectories [12] or using nominal knee angles [11].

The collision function $C(.)$ simply makes sure that the mid-trajectory end-effector positions always lie on one side of both starting and ending contact surfaces. More precisely:

$$
C(p) = \begin{bmatrix} (p - P^-)^T N^- \\ (p - P^+)^T N^+ \end{bmatrix} - c \qquad (8)
$$

where superscripts $-$ and $+$ represent starting and ending contact surfaces for the corresponding swing phase. The scaler $c = 5\%$ of leg length also indicates a desired amount of ground clearance or leg lift for the swing trajectories. Finally, we start optimizations by an intuitive (possibly infeasible) initial solution where $h_i = T$, swing trajectories connect the centers of contact surfaces together, velocities are differentiations of these positions and contact forces are set to the total weight divided by the number of active contacts in each phase. The optimization therefore adjusts all these variables optimally. We use the package CasADi [26] to formulate our problem with automatic symbolic differentiation and solve it via IPOPT [27].

III. RESULTS

In this section, we study the optimization results for different motion planning tasks. We consider changing terrain complexity, planning horizon and trajectory precisions. In all experiments, we keep the motion sequence fixed, i.e. we use the same order of limb motions all the time. This sequence is set to move one limb at a time, although it could be easily generalized to moving arbitrary number of limbs and even going to flight phases.

A. Swing dynamics

As motivated previously, the aim of using a five-mass model in our method is to study the exchange of energy between the base (torso) and the limbs. In another work

[28] we showed that in bipedal walking motions, swing and torso balancing dynamics can influence torso motions mainly by reducing the speed. Mechanically, a transfer of kinetic energy from the torso to the limbs enables them to move, but slows down the torso itself. Single-mass models cannot include this phenomena in optimizations [8] while Centroidal dynamic approaches require inverse kinematics and full-body momentum equations to produce dynamically consistent motions [10]–[12]. Our method can easily model this phenomena while reducing the complexity and considerably speeding up optimizations. Depending on the frequency of motion and the robot properties, this exchange of energy might become important. In other words, Cartesian motions that ignore swing dynamics might only be realizable on robots with lightweight limbs [8]. Otherwise, the choice of frequency should be slow enough to allow for a feasible tracking of Cartesian trajectories with inverse kinematics or dynamics. Fig. 3 demonstrates an example of motion where limb-swing dynamics can influence torso motions to some extent. One can observe in Fig. 3B that torso velocities are generally decreased when the limbs start moving.

The scenario in Fig. 3 is of particular interest, because the torso almost has to stop in order to move hands. In such flat terrain conditions, ideally, the robot does not need to use hands at all. However, since the rhythm of motion is fixed and given, the planner cannot skip hand-support phases and let the robot naturally walk. It is also not able to shorten these phases, because by design, the hands cannot just move constantly with the body and instead, they have to stop at corresponding contact surfaces for some time. Consequently, because of the lifting terms which bring the legs to their maximum length, the robot cannot keep both feet in contact and move the torso with a high velocity forward. One of the limbs may geometrically violate its reachability limits during double-feet support. As a result, the torso almost stops completely when the arms move. This is indeed the limitation of our predefined motion sequence which requires stationary hand positions and could be solved by deciding a different sequence.

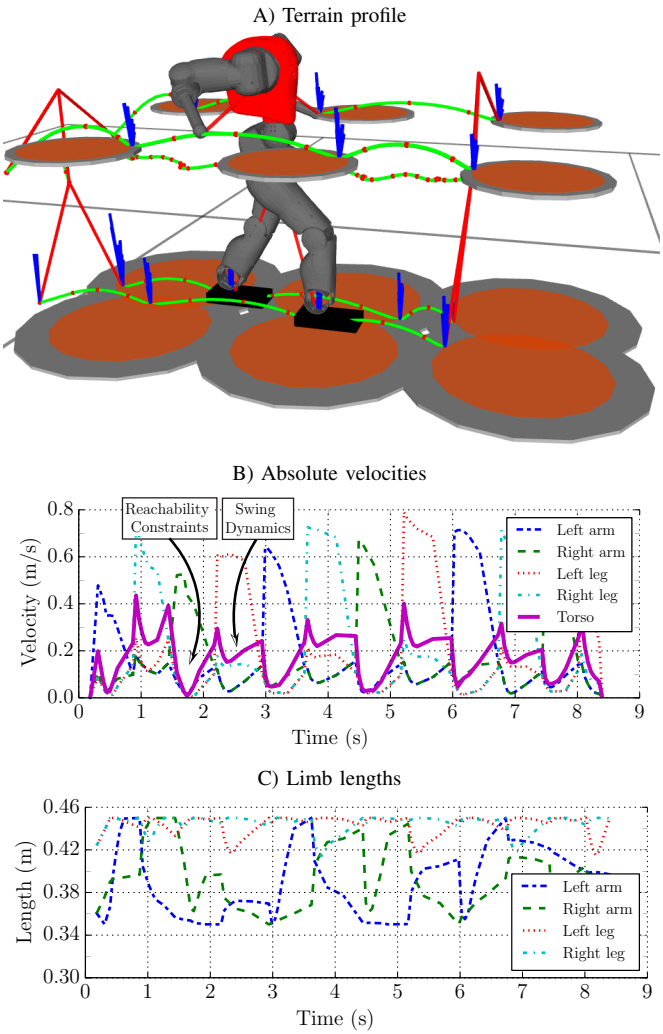

Fig. 3: Planning $M = 12$ steps with a fine resolution of $K = 7$ and a nominal phase duration of $T = 0.7$s. In this simple environment shown in A), trajectories are rather simple, but limb swing dynamics influences the torso velocity considerably. In B), we have shown absolute velocities for all the five masses in our model. In this scenario, since hand-support surfaces are placed slightly high, the elbows are normally bent which means the arms hardly reach the boundaries of their workspace. On the other hand, because of the lifting term, the legs are normally stretched and limited. Therefore, when both feet are in contact, the pelvis and torso cannot progress much because of such reachability constraints and thus slow down. This can be seen in the accompanied video. The double-support duration is relatively small in human walking as well [29]. When the two hands are fixed, the pelvis and torso have more freedom to move, though leg-swing dynamics influences their speed slightly.

B. Static vs. dynamic motions

In the second test, we change the frequency of motion and investigate the resulting trajectories. The frequency of normal walking in a child with the same height as our COMAN robot (approximately 90cm) is about $T = 0.45$s [30]. Here, we consider two gaits of $M = 12$ steps with different average phase durations of $T = 0.3$s and $T = 0.7$s. As demonstrated in Figures 4 and 5, the choice of $T = 0.7$s produces a slow motion with conservative contact forces and non-smooth trajectories. The choice of $T = 0.3$s produces forces that reach friction cone boundaries, but resulting in smoother trajectories and similar limb acceleration patterns.

C. Optimization performance

In the final set of experiments, we change different motion parameters and report the trends in optimization performance. In particular, the horizon length M, trajectory precisions determined by K and motion dynamics determined by T have considerable influences on the optimization performance. Table I lists different optimization setups to investigate these parameters. It is observed that the number of future steps almost linearly increases the optimization time which is due to the sparsity of our technique similar to [24]. Increasing the number of sub-phases has a similar effect according to Table I. Also, generating more static motions is generally faster, since velocities and accelerations are small in magnitude.

Steps: M	Avg. Durations: T	Sub-Phases: K	Terrain	Variables	Constraints	Iterations	Optim. Time	Iter. Time	Description
4	0.5s	4	Rough	431	606	25	0.12s	4.8ms	Short horizon
8	0.5s	4	Rough	879	1254	23	0.29s	12.6ms	Medium horizon
12	0.5s	4	Rough	1327	1902	27	0.52s	19.3ms	Long horizon
12	0.3s	4	Rough	1327	1902	35	0.69s	19.7ms	High frequency
12	0.5s	4	Rough	1327	1902	27	0.52s	19.3ms	Natural frequency
12	0.7s	4	Rough	1327	1902	20	0.36s	18.0ms	Slow frequency
12	0.5s	4	Rough	1327	1902	27	0.52s	19.3ms	Normal gait
12	0.5s	7	Rough	2317	3258	35	1.39s	39.7ms	High resolution
12	0.7s	7	Flat	2317	3258	32	1.21s	37.8ms	Flat gait

TABLE I: Dimensions and performance metrics of different optimization setups. The first group of three experiments investigates the effect of horizon length. The second group explores different gait frequencies. The third group also investigates the effect of motion resolution. Increasing the horizon almost linearly increases the optimization time. Producing quasi-static gaits can be done faster. Increasing the resolution also linearly increases optimization time which is slightly better for simpler environments.

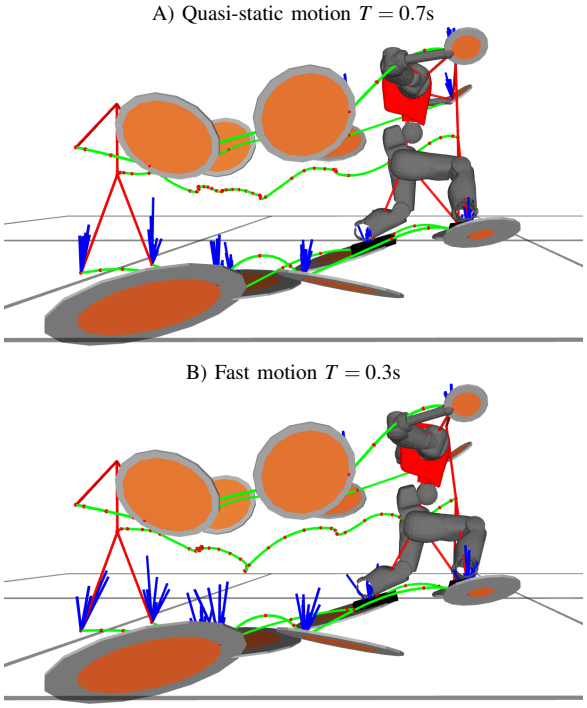

Fig. 4: Rough-terrain locomotion with different average phase durations. With a choice of $T = 0.7$s, the robot spends more time in each phase while with $T = 0.3$s, the motion becomes much faster. In this case, contact forces are also less conservative and may reach friction cone boundaries.

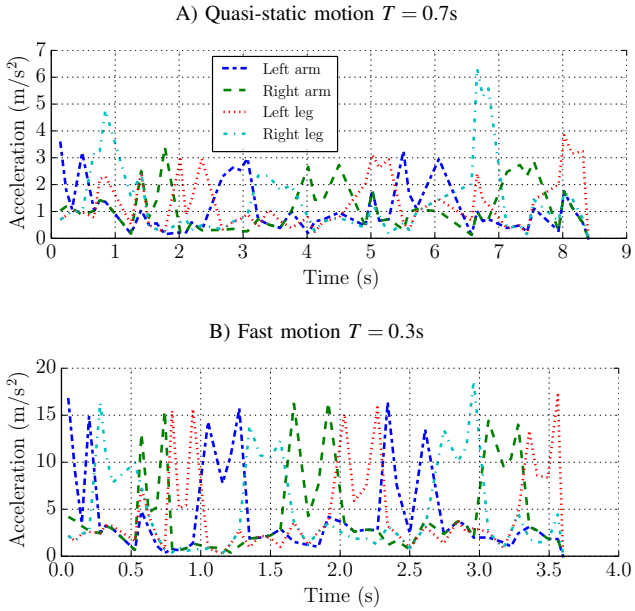

Fig. 5: Rough-terrain locomotion with different average phase durations similar to Fig. 4. With a choice of $T = 0.7$s, absolute limb accelerations follow a jerky pattern whereas with $T = 0.3$s, the motion is fast and smooth. In this case, different limb accelerations follow similar patterns.

IV. CONCLUSIONS

Similar to our posture planning work earlier [5], we observe that representing geometries by Cartesian points instead of joint angles can speed up motion planning optimizations as well. In the most complicated case, we have $M = 12$ steps and $K = 7$ sub-phases which gives a total of 84 sub-phases and optimization time of 1.39s. In comparison, the combination of Centroidal dynamics and inverse kinematics used in [12] optimizes 100 sub-phases in 30s. While this combination can account for swing dynamics in fast motions, a reduced version without inverse kinematics [24] can run much faster and optimize a static plan (of 4 steps in 8s duration and with 80 sub-phases) in 0.421s. Our method, however, can find more dynamic motions in similar optimization setups thanks to inclusion of internal dynamics in the five-mass model. It also leaves the choice of sub-phase durations free in optimizations to find more optimal solutions compared to [12], [24].

We only used an intuitive start point without any prior knowledge. Running these optimizations with a warm start in receding-horizon optimizations can be much faster and allow for a reactive online control of up to 50Hz update rate, depending on the horizon length and the desired resolution. In the smallest setup, we can optimize $M = 4$ steps in 120ms while spending 4.8ms per iteration. For a typical MPC control application with receding horizon, we might only need up to 5 iterations which may take 20-25ms. This convincingly extends our previous MPC controllers [19] for bipedal walking which used the linear inverted pendulum model and could adjust footsteps in 0.2ms. Comparatively, with a cost of longer computations which is still much faster than the normal frequency of motion ($T = 0.5$s), we can optimize a much more complex multi-contact gait. We can adjust contact positions, swing trajectories and phase timings altogether. More importantly, we are not limited to linearity

assumptions anymore.

Our method is able to produce motions with different properties despite using a given fixed rhythm. It can also adjust the timing of different phases in order to smoothen trajectories. However, we consider using Cost of Transport (CoT) terms in the objective function and remove the motion duration constraint to naturally find an optimal frequency. For online push-recovery applications which may require activation or deactivation of certain contacts on the fly, assuming that only few integer combinations are decided, our method is potentially fast enough to be used in a mixed-integer optimization solver. A possible scenario would be receiving external pushes during locomotion while available foothold regions are not enough to recover. In this case, the robot may decide to use hands against the wall which is a new discrete decision. We do not have a mixed-integer optimization setup yet, but we plan to add this feature in future work. Ultimately, we would like to integrate this planner with a perception pipeline that extracts environment geometries [1], and a compliant inverse dynamics controller that provides precise and fast tracking [31].

V. ACKNOWLEDGMENT

This work was funded by the WALK-MAN project (European Community's 7th Framework Programme: FP7-ICT 611832).

REFERENCES

[1] P. Kaiser, E. E. Aksoy, M. Grotz, D. Kanoulas, N. G. Tsagarakis, and T. Asfour, "Experimental evaluation of a perceptual pipeline for hierarchical affordance extraction," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 136–146.

[2] J. R. Rebula, P. D. Neuhaus, B. V. Bonnlander, M. J. Johnson, and J. E. Pratt, "A controller for the littledog quadruped walking on rough terrain," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1467–1473.

[3] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.

[4] M. Al Borno, M. De Lasa, and A. Hertzmann, "Trajectory optimization for full-body movements with complex contacts," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 8, pp. 1405–1414, 2013.

[5] S. Faraji and A. J. Ijspeert, "Modeling Robot Geometries Like Molecules, Application to Fast Multicontact Posture Planning for Humanoids," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 289–296, 2018.

[6] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited humanoid locomotion planning," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*. IEEE, 2005, pp. 7–12.

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

[8] 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 Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[9] L. Colasanto, N. G. Tsagarakis, and D. G. Caldwell, "A compact model for the compliant humanoid robot COMAN," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, IEEE. IEEE, 2012, pp. 688–694.

[10] 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.

[11] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 295–302.

[12] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2703–2710.

[13] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[14] S.-Y. Chung and O. Khatib, "Contact-consistent elastic strips for multi-contact locomotion planning of humanoid robots," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6289–6294.

[15] S. Brossette, J. Vaillant, F. Keith, A. Escande, and A. Kheddar, "Point-cloud multi-contact planning for humanoids: Preliminary results," in *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*. IEEE, 2013, pp. 19–24.

[16] K. Bouyarmane and A. Kheddar, "Multi-contact stances planning for multiple agents," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5246–5253.

[17] 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 *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 5708–5715.

[18] H. Hemami and C. Golliday, "The inverted pendulum and biped stability," *Mathematical Biosciences*, vol. 34, no. 1, pp. 95–110, 1977.

[19] S. Faraji, S. Pouya, and A. Ijspeert, "Robust and agile 3d biped walking with steering capability using a footstep predictive approach," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[20] A. Herdt, N. Perrin, and P.-B. Wieber, "Walking without thinking about it," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 190–195.

[21] W. Huang, J. Kim, and C. G. Atkeson, "Energy-based optimal step planning for humanoids," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3124–3129.

[22] J. Borràs, C. Mandery, and T. Asfour, "A whole-body support pose taxonomy for multi-contact humanoid robot motions," *Science Robotics*, vol. 2, no. 13, p. eaaq0560, 2017.

[23] 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*. IEEE, 2014, pp. 279–286.

[24] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 842–849.

[25] P. de Leva, "Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters," *J Biomech*, vol. 29, no. 9, pp. 1223–1230, Sept. 1996.

[26] J. Andersson, J. Åkesson, and M. Diehl, "Casadi: A symbolic package for automatic differentiation and optimal control," in *Recent advances in algorithmic differentiation*. Springer, 2012, pp. 297–307.

[27] 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.

[28] S. Faraji and A. J. Ijspeert, "3lp: A linear 3d-walking model including torso and swing dynamics," *the international journal of robotics research*, vol. 36, no. 4, pp. 436–455, Apr. 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364917708248

[29] G. Cappellini, Y. P. Ivanenko, R. E. Poppele, and F. Lacquaniti, "Motor patterns in human walking and running," *Journal of neurophysiology*, vol. 95, no. 6, pp. 3426–3437, 2006.

[30] J. Hausdorff, L. Zemany, C.-K. Peng, and A. Goldberger, "Maturation of gait dynamics: stride-to-stride variability and its temporal organization in children," *Journal of Applied Physiology*, vol. 86, no. 3, pp. 1040–1047, 1999.

[31] S. Faraji, L. Colasanto, and A. J. Ijspeert, "Practical considerations in using inverse dynamics on a humanoid robot: Torque tracking, sensor fusion and Cartesian control laws," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1619–1626.