On Time Optimization of Centroidal Momentum Dynamics

Brahayam Ponton\textsuperscript{1}, Alexander Herzog\textsuperscript{1}, Stefan Schaal\textsuperscript{1,2} and Ludovic Righetti\textsuperscript{1,3}

Abstract—Recently, the centroidal momentum dynamics has received substantial attention to plan dynamically consistent motions for robots with arms and legs in multi-contact scenarios. However, it is also non convex which renders any optimization approach difficult and timing is usually kept fixed in most trajectory optimization techniques to not introduce additional non convexities to the problem. But this can limit the versatility of the algorithms. In our previous work, we proposed a convex relaxation of the problem that allowed to efficiently compute momentum trajectories and contact forces. However, our approach could not minimize a desired angular momentum objective which seriously limited its applicability. Noticing that the non-convexity introduced by the time variables is of similar nature as the centroidal dynamics one, we propose two convex relaxations to the problem based on trust regions and soft constraints. The resulting approaches can compute time-optimized dynamically consistent trajectories sufficiently fast to make the approach realtime capable. The performance of the algorithm is demonstrated in several multi-contact scenarios for a humanoid robot. In particular, we show that the proposed convex relaxation of the original problem finds solutions that are consistent with the original non-convex problem and illustrate how timing optimization allows to find motion plans that were difficult to plan with fixed timing. Source code is available at https://git.amd.tuebingen.mpg.de/bponton/timeoptimization.

I. INTRODUCTION

Motion optimization for robots with arms and legs such as humanoids is a challenging task for many reasons including very-high dimensionality, problem discontinuities due to intermittent contacts, non convex optimization landscapes prone to local minima, realtime constraints to find a solution, quality of the solution for execution on a real robot, etc. Yet, these challenges have inspired researchers for many years to develop optimization methods that show really impressive simulation results \cite{1}, \cite{2}, \cite{3}, \cite{4}, \cite{5}.

On the more practical side however, the most successful methods have not been the most complex and computationally expensive ones, but the ones that provide sufficient flexibility to perform a desired task while being well suited for model predictive control \cite{6}, \cite{7}, \cite{8}, \cite{9}, \cite{10}, \cite{11}. Indeed the ability to compute motions in a receding horizon fashion is very important to provide the necessary reactivity to the robot behavior in uncertain environments.

In recent years, the centroidal momentum dynamics model \cite{12}, \cite{13} has become a popular model for multi-contact dynamic full-body motions \cite{1}, \cite{14}. Indeed, this model, under the assumption of enough torque authority, provides sufficient conditions for planning dynamically feasible motions \cite{15}, \cite{16}, and is simple enough such that the problem could be solved with close to realtime rates \cite{6}, \cite{17}. Notably, it was shown in \cite{16} that such plans could be successfully used on a humanoid robot.

However, the centroidal momentum dynamics is not convex, which renders the optimization problem difficult. Recent works have looked at the mathematical structure of the problem in order to find more efficient optimization algorithms. In \cite{16}, a multiple-shooting method is used to efficiently optimize the centroidal dynamics. In \cite{18} a convex upper bound on the angular momentum is used in order to minimize the $l_1$ norm of the angular momentum. In \cite{2}, it was shown that the non convex part could be decomposed as a difference of quadratic functions. This allows an efficient convex approximation of the problem that can be used in sequential quadratic programming approaches. The paper also proposed a method to efficiently compute dynamically consistent full-body motions by alternating centroidal dynamics optimization with full robot kinematics optimization.

In our previous work \cite{17}, we proposed a convex relaxation of the problem that allowed to find solutions at realtime rates. However, our approach was using a proxy function to minimize the angular momentum: it was minimizing the sum of the squares of the quadratic functions composing the non-convex part of the equations. Thus, it was not possible to include an explicit target momentum in the cost function, which can limit the space of desirable solutions where momentum is effectively minimized. Moreover, our approach could not be used directly in the alternating full-body optimization method proposed in \cite{2} to do full-body optimization.

Another under-studied aspect is the importance of timing for centroidal momentum optimization. Several works have realized the importance of including time as an optimization variable \cite{1}, \cite{4}, \cite{6} and in doing so have shown very nice simulation and experimental results. However, including time optimization is usually computationally very costly due to the non convexity introduced in the discretized dynamics.

In this paper, we propose two methods for convex relaxation of the centroidal momentum dynamics optimization problem that allows to include an explicit angular momentum objective. Moreover, noticing that the non-convexity introduced by the time variable is of similar nature as the torque cross product allows us to use the same relaxation approaches for time optimization. The resulting algorithm allows to compute dynamically consistent plans close to realtime. Experiments demonstrate the computational efficiency of our approach in multi-contact scenarios. In particular, we show that the numerical solutions found in our method are very close to the original dynamics, suggesting that the convex relaxation is close enough to the real problem. Then we show that optimizing time allows to find solutions that could not be easily found with fixed time optimization. We also combine

\textsuperscript{1}Max Planck Institute for Intelligent Systems, Tuebingen, Germany
\textsuperscript{2}University of Southern California, Los Angeles, USA
\textsuperscript{3}New York University, New York, USA
our approach with the alternating approach proposed in [2] to compute dynamically consistent full-body motions.

The remainder of this paper is structured as follows. In Sec. II, we present the problem formulation. Then, in Sec. III, we show how to efficiently find a solution to the centroidal momentum dynamics problem. We show experimental results in Sec. IV and conclude the paper in Sec. V.

II. PROBLEM FORMULATION

The equations of motion that describe the dynamic evolution of a floating-base rigid body system are given by

\[ M(q)\ddot{q} + N(q, \dot{q}) = S^T \tau_j + J^T_{e} \lambda \]

where the robot state is denoted by \( q = [q^T \dot{q}^T]^T \), and comprises the position and orientation of a floating base frame in the robot relative to an inertial frame \( x \in SE(3) \), and the joint configuration \( \eta_j \in \mathbb{R}^{n_j} \). The inertia matrix is denoted by \( M(q) \in \mathbb{R}^{(n+6) \times (n+6)} \), the vector of nonlinear terms \( N(q, \dot{q}) \in \mathbb{R}^{n+6} \) includes Coriolis, centrifugal, gravity and friction forces, the selection matrix \( S \in \mathbb{R}^{(n+6) \times \eta_j} \) represents the system under-actuation, namely that \( x \in SE(3) \) is not directly actuated by the vector of joint torques \( \tau_j \in \mathbb{R}^{n_j} \), but indirectly through a vector of generalized forces \( \lambda \) and the Jacobian of the contact constraints \( J_e \).

The system under-actuation leads to a dynamics decomposition into an actuated (subscript \( a \)) and un-actuated parts (subscript \( u \)) as follows:

\[ M_a(q)\ddot{q} + N_a(q, \dot{q}) = \tau_j + J^T_{e,a} \lambda \quad (1a) \]
\[ M_u(q)\ddot{q} + N_u(q, \dot{q}) = J^T_{e,u} \lambda \quad (1b) \]

Equation (1b), known as the Newton-Euler equations, tells us that the systems’ change of momentum depends on external contact forces. Any combination of forces \( \lambda \) and accelerations \( \ddot{q} \) can be realized, if are consistent with the underactuated dynamics (1b), and there is enough torque authority (1a) [19], [20]. This natural decomposition suggests that satisfaction of the momentum equation (1b) is sufficient to guarantee dynamic feasibility, and equation (1a) ensures kinematic feasibility and torque limits.

A. Dynamics Model

As a consequence of the last observation, under the assumption of enough torque authority and kinematic reachability, a necessary condition for planning physically consistent motions is that the total wrench generated by external and gravitational forces (2) equals the rate of momentum computed from the robot joint angles and velocities (3) [1]. On the one hand, the centroidal momentum, computed from external forces, expressed at the robot center of mass is

\[ h = \begin{bmatrix} r \\ l \\ k \end{bmatrix} = \begin{bmatrix} l^T \\ \frac{1}{2} l^T \Sigma c(p_c + R_{x,y}z_c - r) x_c + R_{x,y} \tau_c \end{bmatrix} \quad (2) \]

where \( r \) denotes the center of mass position, \( l \) the linear and \( k \) the angular momentum. \( h \) is a shortcut vector comprising \( r, l, \) and \( k \). The total robot mass is \( m \) and \( g \) the gravity vector. The position of the \( e \) end-effector is denoted \( p_e, z_e \in \mathbb{R}^2 \) is the center of pressure (CoP) expressed in local end-effector coordinates. \( R_{x,y}^e \in \mathbb{R}^{3\times2} \) represents the first two columns of the rotation matrix \( R_{e} \in \mathbb{R}^{3\times3} \) that maps quantities from end-effector frame to the inertial coordinate frame. \( f_e \in \mathbb{R}^3 \) and \( R_{x}^e \tau_e \in \mathbb{R}^3 \) are forces and torques acting at contact point \( p_e + R_{x,y}^e z_e \), represented in inertial frame. \( \tau_e \in \mathbb{R}^3 \) is the torque around the \( z \) upward pointing, axis expressed in end-effector frame. \( R_{x}^e \) maps \( \tau_e \) to the inertial coordinate frame.

On the other hand, the centroidal momentum, computed from the robot joint angles and velocities, is given by [21]

\[ \begin{bmatrix} 1 \\ k \end{bmatrix} = A(q)q \quad (3) \]

where \( A(q) \in \mathbb{R}^{6 \times n_j + 6} \) is the centroidal momentum matrix. It is important to highlight that momentum, as given by (2), only depends on dynamic quantities, while the momentum, as given by (3), only depends on kinematic quantities. This separation, originally suggested in [2], allows the use of an iterative procedure, where one can alternate between a kinematic and a dynamic optimization to solve the joint problem. Both optimization procedures need only agree on the center of mass trajectory, momentum and contact locations and such agreement is enforced by a cost function in each optimization algorithm. The benefit of this separation is that the optimization problems can be solved more easily separately than as a joint problem and the inherent structure of each problem can be exploited in dedicated solvers.

B. Trajectory Optimization

In this paper we focus on the centroidal dynamics optimization problem and only briefly comment on the alternating full-body optimization procedure as we will use it in the experiments section. First, we discretize the differential equations (2)-(3) into algebraic equations and then use the iterative procedure described in [2] to alternate between a dynamics and a kinematics optimization.

1) Kinematics Optimization Problem: We will not focus on the kinematic optimization. However, we would like to mention that we use at each time step an inverse kinematics procedure, whose objective is to track a desired center of mass position, linear and angular momenta, regularize rates of momentum, track desired motions for unconstrained end-effectors, regularize joint posture towards a default posture and regularize joint velocities and accelerations. The constraints include the evolution of linear and angular momenta using the centroidal momentum dynamics (3), evolution of center of mass according to linear momentum, evolution of end-effectors based on the end-effector Jacobians, joint limits and constraints for active end-effectors.

2) Dynamics Optimization Problem: We are interested in the efficient optimization of dynamic motions including momentum trajectories, contact forces and timings, under the non-convex and non-linear centroidal momentum dynamics, which could later be realized by a low-level controller such as an inverse dynamics controller [15]. Formally, the objective to be minimized in our optimization problem is:

\[ \min_{\tau_c, \tau_e, z_c, \Delta t} \phi_N(h_n - h_{n_{des}}) + \sum_{i=1}^{n-1} \ell_i(h - h_{des}, z_c, \tau_e, \tau_c, \Delta t) \quad (4) \]

More specifically, we would like to minimize a terminal cost \( \phi_N(h_n - h_{n_{des}}) \) that penalizes the difference between
the final state \( \mathbf{h}_n \) and the desired final state \( \mathbf{h}_{\text{des}} \), and a running cost \( \ell_f (\mathbf{h} - \mathbf{h}_{\text{des}}, \mathbf{z}_e, \mathbf{f}_e, \tau_e, \Delta_e) \), that penalizes the tracking performance of a desired linear and angular momentum trajectories \( \mathbf{h} - \mathbf{h}_{\text{des}} \), and regularizes the available controls, namely, forces \( \mathbf{f}_e \), torques \( \tau_e \) and time discretizations \( \Delta_e \).

The desired linear and angular momentum trajectories could be as trivial as zeros, if the optimization algorithm is to be used by itself, or could for instance come from the kinematic optimization. The constraints of the optimization problem include a discrete form of the centroidal momentum dynamics (2):

\[
\mathbf{h} = \begin{bmatrix} \mathbf{r}_t \\ \ell_t \\ \mathbf{k}_t \\ \mathbf{l}_t \\ \mathbf{i}_t \\ \mathbf{m}_t + \sum_{e} \mathbf{f}_{e,t} \\ \sum_e \kappa_{e,t} \end{bmatrix}
\]

(5)

The variable \( \kappa_{e,t} \) (end-effector contribution to angular momentum rate \( \mathbf{k}_t \)) has been defined as

\[
\kappa_{e,t} = (p_{e,t} + R_{e,t}^y z_{e,t} - r_t) \times \mathbf{f}_{e,t} + R_{e,t}^x \tau_{e,t}
\]

\[
= \ell_{e,t} \times \mathbf{f}_{e,t} + R_{e,t}^x \tau_{e,t}
\]

\[
= \begin{bmatrix} \ell_{x,t}^y & 0 & \ell_{y,t} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{e,t}^x \\ \mathbf{f}_{e,t}^y \\ \mathbf{f}_{e,t}^z \end{bmatrix} + R_{e,t}^x \tau_{e,t}
\]

(6)

where for simplicity of notation, we have introduced the change of variable \( \ell_{e,t} = (p_{e,t} + R_{e,t}^y z_{e,t} - r_t) \). Physical constraints such as friction cone, CoP within region of support and torque limits are given by

\[
\| \mathbf{f}_{e,t}^x + \ell_{e,t}^x \mathbf{f}_{e,t}^z \|_2 \leq \mu \ell_{e,t}^x, \quad \ell_{e,t}^x \geq 0,
\]

(7a)

\[
\mathbf{z}_{e,t}^x, \mathbf{z}_{e,t}^y \in [\mathbf{z}_{e,t}^x, \mathbf{z}_{e,t}^y],
\]

(7b)

\[
\tau_{e,t} \in [\tau_{\text{min}}, \tau_{\text{max}}]
\]

(7c)

\[
\Delta_t \in [\Delta_{\text{min}}, \Delta_{\text{max}}]
\]

(7d)

\[
\| \mathbf{p}_{e,t} - r_t \| - \ell_{e,t}^x \leq \ell_{e,t}^x
\]

(7e)

where \( \mathbf{f}_{e,t}^x = R_{e,t}^x \mathbf{f}_{e,t} \) is the end-effector force in local coordinate frame. (7a) expresses that forces belong to a friction cone with friction coefficient \( \mu \). Note that friction cones could alternately be approximated by the usual pyramids as a polyhedral approximation. (7b) expresses that the CoP should be within a conservative region with respect to the real physical available region. (7c) constrains the torque to a bounded region, but it could also be approximated using a second-order cone constraint of the form \( \| \tau_{e,t} \| \leq \ell_{e,t}^x \). [22] also provides precise closed-form formulas for it under polyhedral approximation of the friction cone. Equation (7d) constrains the time discretization variable to a bounded region. Finally, equation (7e) constrains the distance between the current position of the center of mass \( \mathbf{r}_t \) and the end-effector contact point \( \mathbf{p}_{e,t} \) to be less than the maximum length of the end-effector \( \ell_{e,t}^x \). It includes a constant offset, when the end-effectors of interest are the arms of the robot.

We would like to conclude this section by highlighting that the non-convexities of this problem are the bilinear terms of eq. (5). For instance, terms where the time discretization variable \( \Delta_t \) appears or the cross products of the angular momentum rate terms \( \kappa_{e,t} \). In traditional momentum optimization [2], [16], [23], the focus is on the cross products of the angular momentum, that introduce bilinear constraints to the optimization problem. It is important to note that including time discretizations as optimization variables introduces nonconvex constraints of the same bilinear nature. Therefore, the goal of the next section will be to devise methods that can approximate bilinear constraints, while still allow us to efficiently find a solution to the optimization problem.

III. APPROACH

In this section, we describe a tool to express bilinear constraints (5) as a difference of convex functions, and then we show how to approximate these still nonconvex constraints, using the knowledge about their positive curvature.

A. Disciplined Convex-Concave Programming

In this subsection, we will describe a useful tool [24] for dealing with bilinear constraints in optimization problems. The method decomposes a bilinear or nonconvex quadratic expression into a difference of convex functions; in other words, it decomposes the bilinear expression into a difference of two terms, each of which is a convex function. Once the decomposition has been performed, the constraint will continue to be nonconvex; however, the terms composing it will have known curvature, which will let us perform an efficient approximation.

This approach, previously used and detailed in [2], [17], analytically decomposes the nonconvex quadratic expressions of the angular momentum (6) into a difference of convex (quadratic) functions. The set of difference of convex functions \( C^\pm \) is defined as:

\[
C^\pm = \left\{ C^+ (\mathbf{x}) - C^- (\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n, \right\}
\]

\[
C^+, C^- : \mathbb{R}^n \rightarrow \mathbb{R}, \text{ are convex functions}\}
\]

Expressions such as scalar products \( x^T y \) can be decomposed into a difference of convex functions, such as \( Q^+ - Q^- \)

\[
Q^+ = \frac{1}{4} \| x+y \|^2, \quad Q^- = \frac{1}{4} \| x-y \|^2
\]

where \( Q^+ \in C^+ \) and \( Q^- \in C^- \) are convex functions. Using this idea, we can easily identify scalar products and transform quadratic expressions in cross products into scalar products, which can then be defined as elements of \( C^\pm \). As an example, we present the decomposition of a cross product:

\[
\ell \times \mathbf{f} = \begin{bmatrix} \ell_x \mathbf{f}_x + d_{\mathbf{f}x} \mathbf{f}_x \\ \ell_y \mathbf{f}_y - e_{\mathbf{f}y} \mathbf{f}_y \\ \ell_y \mathbf{f}_x - \ell_x \mathbf{f}_y \\ \ell_x \mathbf{f}_y - \ell_y \mathbf{f}_x \end{bmatrix}^T
\]

and then each scalar product is defined as an element of \( C^\pm \).

\[
\ell \times \mathbf{f} = \frac{1}{4} \left[ \| a_{\mathbf{f}x} + d_{\mathbf{f}x} \|^2 - \| a_{\mathbf{f}x} - d_{\mathbf{f}x} \|^2 \right]
\]

\[
| b_{\mathbf{f}x} + e_{\mathbf{f}x} |^2 - | b_{\mathbf{f}x} - e_{\mathbf{f}x} |^2
\]

\[
\| c_{\mathbf{f}x} + f_{\mathbf{f}x} \|^2 - \| c_{\mathbf{f}x} - f_{\mathbf{f}x} \|^2
\]
For simplicity of presentation, we denote \( p = a_{cvx} + d_{cvx} \) and \( q = a_{cvx} - d_{cvx} \) \((p, q \in \mathbb{R}^2)\). With this notation the first component of the torque contribution of an end-effector to the angular momentum rate dynamics \( \kappa \) becomes

\[
\kappa^x = \frac{1}{4} [p^T p - q^T q] + \tau^x,
\]

which is equivalent to the following formulation

\[
\kappa^x = \frac{1}{4} [\bar{p}^T \bar{q}] + \tau^x,
\]

\[
\bar{p} = p^T p - q^T q
\]

(8a)

where we have introduced the scalar variables \( \bar{p}, \bar{q} \in \mathbb{R}_+ \).

Under this formulation the original nonconvex quadratic constraint has been now separated into a linear constraint, and two non-convex constraints (8a). The difference is that, while in the original constraint, the hessian is an indefinite matrix, the hessian of the new constraints is positive semi-definite and therefore each term has known curvature.

B. Approximations of Quadratic Equality Constraints

In the last section, we have presented a method to analytically decompose a bilinear constraint into a linear constraint and two nonconvex quadratic equality constraints with known curvature, see (8a). In this section, we present alternatives for dealing with each of the two nonconvex quadratic equality constraints with known curvature, namely, approximation using a trust region and use of soft constraints.

While the trust region could be introduced as a trivial box constraint with a threshold, that would constraint the value of \( \bar{p} \) to values near \( p^T p \), the best affine trust region that exploits the knowledge about the curvature of the function and the information contained in the current value of the optimal vector is a linear approximation such as \( p^T_{val} p_{val} + 2 p_{val} (p - p_{val}) \geq \bar{p} - \sigma \), where \( \sigma \) is a positive value representing a threshold, enough to provide a feasible interior to the intersection of the constraints, and \( p_{val} \) is any value taken by the variables \( p \) coming from the solution of the relaxed problem. Notice that if the hessian of the quadratic equality constraint where an indefinite matrix, this trust region would not constraint the problem as desired and instead lead to unbounded regions.

While this approximation is still a relaxation of the constraint, it allows us to approximate the terms \( \bar{p}, \bar{q} \), and consequently \( \kappa^x \). This formulation has the advantage that we do not trade-off expressiveness of the cost, allowing for example quadratic terms over \( \kappa^x \) without losing the convexity properties.

2) Approximation using soft constraints: This method is similar in spirit to the last one. It will first drop all the nonconvex terms to find an initial guess for the optimal vector and then it introduces heuristics in the cost, whose purpose is to bias the solutions towards the boundaries. Unlike the previous method, this method does not restrict strictly the search space, but instead biases the solutions towards the boundaries by pulling the variables towards an underestimator of the function.

As in the previous case, the best affine underestimator, that exploits the knowledge about the curvature of the function and the current value of the optimal vector is a linearization \( p^T_{val} p_{val} + 2 p_{val} (p - p_{val}) \). The heuristic is a quadratic term in the cost that penalizes the difference between the variable \( \bar{p} \) and the linearization. This rewards the optimization for selecting values of \( \bar{p} \) that are close to the boundaries of the constraint and are therefore feasible for the nonconvex constraint. As in the previous case, we do not trade-off expressiveness of the cost, allowing for example quadratic terms over \( \kappa^x \) without losing the convexity properties.

IV. EXPERIMENTS

We have tested the algorithm in several multi-contact scenarios, including walking on an uneven terrain (Fig. 2), walking under a bar using also hands (Fig. 3), and a walking motion with low friction coefficient (Fig. 4). The resulting plans are visible in the attached video https://youtu.be/oFI54WE8O3E.

A. Walking on uneven terrain

The first motion has been built so that double support time after a single support on left foot is very short, while double support time after single support on right foot is longer. It then allows to see the effect of double support duration on momentum optimization. Stairs are also close to each other, such that legs have to generate momentum to be lifted up, and minimum jerk trajectories that guide this kinematic motion have not been well tuned such that the effect of it is more visible. We test three optimizations: with fixed time (Momentum Optimization), with time as an
Fig. 2: Walking motion in uneven terrain. This figure shows optimization results for the motion optimizing only momentum (blue), including time in the optimization (green) and fixed time horizon optimization (red). Plots show normalized quantities of linear and angular momentum by the robot mass, and forces by robot mass times gravity. Bottom plot shows optimal time discretizations. For momentum plots, we show the dynamically optimized momentum and its kinematic tracking. For force plots, the forces at each end-effector are shown. Vertical colored bars represent endeffector activations. For example, at the beginning the robot is in double support, then single support on left foot, then double support again and finally single support on right foot. From there on, the cycle repeats itself.
Fig. 3: Walking under a bar using hands. This figure shows optimization results for the motion optimizing only momentum (blue) with soft constraint heuristic, with trust region heuristic (green) and including time optimization (red). Plots show normalized quantities of linear and angular momentum by the robot mass, and forces by robot mass times gravity. Bottom plot shows optimal time discretizations. For momentum plots, we show the dynamically optimized momentum and its kinematic tracking. For force plots, the forces at each end-effector are shown. Vertical colored bars represent endeffector activations. The colors on the timesteps plot correspond to the colors of forces in previous plots. For example, blue is right foot, orange left foot, green right hand and yellow left hand.
optimization variable and no constraints on the total duration of the motion (Time and Momentum Optimization) and with time as an optimization variable and a fixed total duration (Fixed Time Horizon and Momentum Optimization)

As can be seen in Fig. 2 (Momentum Optimization, in blue), the angular momentum in the Y direction has peaks for short time double supports, while during longer double supports less momentum is distributed along a longer timespan. For the optimization of both time and momentum optimization (green), timestep discretizations are included in the optimization, without a fixed total duration. As can be seen in the bottom plot, time discretization is increased during the initial short time double supports. The optimization thus automatically distribute the angular momentum in the Y direction without any peaks. The last optimization (Fixed time horizon and momentum optimization, in red), is also capable of adapting the time discretizations, however, to be comparable to the original motion, the time horizon is fixed to the same value as in the first experiment. Here, we can also see the tendency of increasing the timesteps at the short time double supports, while reducing time spent at the beginning and end of the motion, and when walking on the straight line. This also allows to decrease angular momentum in the Y direction, which was the goal of the optimization. We also note that in all plots, the momentum coming from the kinematic optimizer can be tracked by the dynamic optimization. These experiments show how time can be used by the optimizer to reduce the overall momentum of the system and result in potentially easier to execute plans.

B. Walking under a bar using hands

In this example, we compare both relaxation methods (trust region and soft constraint) using fixed time (Fig. 3). We see that both relaxations lead to very similar solutions and these solutions lead to very good tracking of the kinematic momentum. This suggests both methods are similarly applicable, however in our current implementation the soft constraint relaxation was always significantly faster than the trust region one. For this specific problem, we also notice that time optimization brings marginal changes to the motion.

C. Walking with low friction coefficient

In this example (Fig. 4), time optimization was critical to find a dynamically feasible motion. In this example, the friction coefficient is 0.4 and the original time horizon is around 10 sec. Under these conditions, the motion cannot be realized without optimizing time. However, time optimization allows to automatically increase the time horizon and the time available during double supports, which generates a physically realizable motion.

D. Discussion on time optimization

As has been shown in the previous experiments, including time as an optimization variable is useful, and depending on the kind of motion it can really produce much lower cost solutions or even make them feasible which is an infinite improvement. However, it increases the dimensionality of the problem to be solved and consequently the time required to solve it. Our convergence criteria is that the average errors on the center of mass, linear and angular momenta, computed comparing the values for the corresponding variables coming from the relaxed optimization problem and the values computed integrating optimal torques and forces are lower than a certain residual error. Notice that the values of angular momentum depend on the convergence of center of mass and linear momentum, therefore, instead of using a relaxed quadratic version for the time related constraints, we use only a linear approximation of them, to significantly speed up the convergence, allowing to come to residual errors in the order of numerical precision for the center of mass and linear momentum, and to modest accuracy for the angular momentum.

1) Time complexity: Table I shows information about the optimal problems being solved and time required to find a solution. Among the parameters being shown are the
time horizon, the number of timesteps, number of variables, linear equality and inequality constraints, second order cones, size and nonzeros of the KKT matrix, and finally the time required to solve the problem. $M_{time}$ is the motion under low friction coefficient with time adaptation, while $M_1$ is the same motion but with normal friction coefficient because time is fixed. $M_2$ is the motion for walking under a bar using hands without time adaptation and $M_{time}$ with time adaptation, which can be easily recognized by looking at the number of second order cone constraints. We can observe that the timings required to solve a dynamics optimization problem are in the order of a second, which is double the time required in [17], because in this method, we also refine the solution after the relaxation, which allows us to directly penalize and track momentum, which increases the time required to find a solution, however improves its quality. Our solving time is comparable to the one reported in [16], which corresponds to 1.23 sec for a motion duration of 8 sec using the centroidal wrench and 3.89 sec for the same motion using contact forces as control input, as in our case. Note that the time taken by the optimization is always faster than the duration of the motion and that, with appropriate warm start of the optimizer, receding horizon control is attainable.

The solve time for a motion that includes time optimization is larger because more iterative refinements are required to converge to a good threshold of constraint violation. However, it could be speed up doing only a few iterations of time optimization and then fixing time discretizations; increasing the accepted approximation error tolerance used as convergence criteria; or warm-starting a time optimization with a fixed-time dynamic optimization. The computation time is still lower than the plan time horizon, what makes it possible to run the algorithm online (the next plan can be computed, while the current one is being executed).

2) Constraint violations: Table I compares the average error in the center of mass, linear and angular momentum, computed integrating forces and torques with the original model and the approximate values in our relaxed formulation. As can be observed, the error on center of mass and linear momentum are marginal, while the error on the angular momentum is really small. The values shown are in an absolute scale. If they are compared to values shown in Figs. 2-4, they are also small, because values in the figures are normalized by the robot mass.

| Time [sec] | $M_{time}$ | $M_1$ | $M_2$ |
|------------|------------|-------|-------|
| 1.28       | 1.28       | 1.28  |
| 1.28       | 1.28       | 1.28  |

V. CONCLUSION

We have presented two convex relaxation methods for the optimization of the centroidal dynamics and motion timing of a legged robot. Our approach is efficiently solvable and could therefore be used in receding horizon control. Moreover, the convex relaxation deviates only marginally from the original dynamics. Our approach has not been yet tested on a real robot, but this is the step coming.

REFERENCES

[1] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *Humanoids*, 2014.
[2] A. Herzog, S. Schaal, and L. Righetti, “Structured contact force optimization for kino-dynamic motion generation,” *IROS*, 2016.
[3] I. Mordatch, E. Todorov, and Z. Popovic, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Trans.*, 2012.
[4] M. Posa and R. Tedrake, “Direct trajectory optimization of rigid body dynamical systems through contact,” in *WAFR*, 2012.
[5] Y. Tassa, T. Erez, and E. Todorov, “Synthesis and stabilization of complex behaviors through online trajectory optimization,” in *IROS*, 2012.
[6] J. Carpenter, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, “A versatile and efficient pattern generator for generalized legged locomotion,” in *ICRA*, 2016.
[7] J. Englsberger, C. Ott, and A. Albu-Schäffer, “Three-dimensional bipedal walking control based on divergent component of motion,” *IEEE Trans. Robotics*, vol. 31, no. 2, pp. 355–368, 2015.
[8] A. Sherikov, D. Dimitrov, and P. Wieber, “Whole body motion controller with long-term balance constraints,” in *Humanoids*, 2014.
[9] P. Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations,” in *Humanoids*, 2006.
[10] S. Caron and A. Kheddar, “Multi-contact walking pattern generation based on model preview control of 3D COM accelerations,” *Humanoid Robots (Humanoids)*, pp. 550–557, 2016.
[11] 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 *IROS 2014*, pp. 4030–4035, 2014.
[12] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Autonomous Robots*, June 2013.
[13] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Resolved momentum control: humanoid motion planning based on the linear and angular momentum,” in *IROS 2003*, pp. 1644–1650, IEEE, 2003.
[14] P. M. Wensing and D. E. Orin, “Generation of dynamic humanoid behaviors through task-space control with conic optimization,” in *ICRA*, pp. 3103–3109, 2013.
[15] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, “Trajectory generation for multi-contact momentum-control,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Nov. 2015.
[16] J. Carpenter and N. Mansard, “Multi-contact locomotion of legged robots,” in *Report IAAAS n. 17172 submitted to IEEE Transaction on Robotics*, 2017.
[17] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, “A convex model of humanoid momentum dynamics for multi-contact motion generation,” in *Humanoids*, 2016.
[18] H. Dai and R. Tedrake, “Planning robust walking motion on uneven terrain via convex optimization,” in *Humanoids*, pp. 579–586, MIT, Cambridge, USA, IEEE, Dec. 2016.
[19] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, “Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid,” *Aaon. Robots*, vol. 40, no. 3, pp. 473–491, 2016.
[20] P.-B. Wieber, “Holonomy and nonholonomy in the dynamics of articulated motion,” *Fast Motions in Biomechanics and Robotics*, 2006.
[21] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Aaon. Robots*, vol. 35, pp. 161–176, Oct. 2013.
[22] S. Caron, Q. Pham, and Y. Nakamura, “Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas,” in *ICRA*, pp. 5107–5112.
[23] S. Caron and A. Kheddar, “Multi-contact walking pattern generation based on model preview control of 3d com accelerations,” in *Humanoid Robots, 2016 IEEE-RAS International Conference on*, November 2016.
[24] X. Shen, S. Diamond, Y. Gu, and S. P. Boyd, “Disciplined convex-concave programming,” in *CDC 2016*, 2016.