



HAL
open science

Differential Dynamic Programming for Multi-Phase Rigid Contact Dynamics

Rohan Budhiraja, Justin Carpentier, Carlos Mastalli, Nicolas Mansard

► **To cite this version:**

Rohan Budhiraja, Justin Carpentier, Carlos Mastalli, Nicolas Mansard. Differential Dynamic Programming for Multi-Phase Rigid Contact Dynamics. 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Nov 2018, Beijing, China. hal-01851596v1

HAL Id: hal-01851596

<https://hal.science/hal-01851596v1>

Submitted on 30 Jul 2018 (v1), last revised 13 Oct 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Differential Dynamic Programming for Multi-Phase Rigid Contact Dynamics

Rohan Budhiraja, Justin Carpentier, Carlos Mastalli, Nicolas Mansard

Abstract—A common strategy today to generate efficient locomotion movements is to split the problem into two consecutive steps: the first one generates the contact sequence together with the centroidal trajectory, while the second one computes the whole-body trajectory that follows the centroidal pattern. Yet the second step is generally handled by a simple program such as an inverse kinematics solver. In contrast, we propose to compute the whole-body trajectory by using a local optimal control solver, namely Differential Dynamic Programming (DDP). Our method produces more efficient motions, with lower forces and smaller impacts, by exploiting the Angular Momentum (AM). With this aim, we propose an original DDP formulation exploiting the Karush-Kuhn-Tucker constraint of the rigid contact model. We experimentally show the importance of this approach by executing large steps walking on the real HRP-2 robot, and by solving the problem of attitude control under the absence of external forces.

I. INTRODUCTION

A. Goal of the paper

Trajectory optimization based on reduced centroidal dynamics [1] has gained a lot of attention in the legged robotics community. Some approaches use it after precomputing the contact sequence and placements [2], [3], [4], [5], [6] while other strategies optimize the centroidal trajectory and contact information together [7], [8], [9]. In both cases, the transfer from centroidal dynamics to whole-body dynamics is achieved using instantaneous feedback linearization to locally take into account the constraints of the robot. These solvers are usually solve quadratic optimization problems written with task-space dynamics (Inverse Kinematics (IK) / Inverse Dynamics (ID)) [10], [11]. While this scheme has shown great experimental results (e.g. [3], [12]), it is still not able to correctly handle the angular momentum produced by the body extremities. This is notably important for humanoid robots which have heavy masses in the limbs, as the angular momentum mostly varies during locomotion due to the motions of the legs and arms. This effect is neither properly handled by the centroidal model, nor by the instantaneous time-invariant linearization.

In [11] an alternative scheme aims to compensate the angular momentum variations. Indeed, it properly compensates the momentum changes produced by the flying limbs, however it is not yet able to trigger additional momentum to enable very dynamic movements. This would be needed for generating long steps, running, jumping or salto motions. To properly handle the angular momentum, it is necessary

The Authors are with CNRS, LAAS, 7 avenue du colonel Roche, Toulouse, France. *email*: {rohan.budhiraja, justin.carpentier, carlos.mastalli, nicolas.mansard}@laas.fr.

to jointly optimize the whole-body kinematics and the centroidal dynamics [2]. However whole-body trajectory optimization approaches suffer from two problems that prevent the replacement of IK/ID solvers. Namely, they have trouble in discovering a valid motion, in particular the gait and its timings; and they are slow to converge.

In this paper, we propose to combine the advantages of centroidal dynamics optimization (to decide the gait, the timings and the main shape of the centroidal trajectory) with a whole-body trajectory optimizer based on multi-phase rigid contact dynamics. In what follows, we first discuss the importance of properly handling the angular momentum during locomotion, before introducing our method.

B. On the importance of angular momentum

Consider an astronaut, floating in space, without any external forces. If he/she mimics the normal human walk, he/she will start spinning in his/her sagittal plane. Indeed, contact forces are not the only way to change the robot orientation. It is known [13] that robot orientation can be controlled without the need of contact forces (i.e. only with the internal joint actuators). Under the action of only internal forces, the angular momentum conservation can be seen as a nonholonomic constraint on the robot orientation. Of course, one can design a control law that counterbalances the lower-body angular momentum. However this will create tracking errors (and potentially instabilities) without mentioning the cases where the arms can be used for multi-contact locomotion. In fact, as shown in [14], a system under nonholonomic constraints cannot be controlled with a time-invariant feedback law. Therefore, angular momentum quantity requires a preview control strategy to be correctly regulated or triggered.

It is often (wrongly) understood that centroidal optimization provides the answer to this problem. The centroidal optimizer can neither anticipate nor modify the limb movements in order to change, as needed, the angular momentum. For instance, the centroidal optimizer cannot anticipate a high demand of the linear part by delaying the limb movement, or exploit the movement of the arms to compensate for excessive forces during a short instant. Nonetheless, these methods are still valid since they provide an efficient way to compute the Center of Mass (CoM) motion while keeping balance and avoiding slippage.

C. Overview of our method

Rather than relying on IK/ID, we propose to use the optimal control framework for computing the whole-body mo-

tion while tracking the centroidal trajectory. Concretely, we propose a local optimal controller solver, namely Differential Dynamic Programming (DDP). DDP has been made popular by the proof of concept [15], and by the demonstration in simulation that it can meet the control-loop timings constraint [16]. Yet it has not been possible to transfer locomotion movements computed by DDP on a real full-size humanoid. Contrary to [15] that optimizes the motion from scratch with a regularized dynamics (thanks to a smooth contact model [17]), we propose to impose the contact phases as decided by the centroidal optimization. As the DDP does not need to discover the contact switching instants, we can then use a rigid contact dynamics which is faster to compute and easier to implement.

Other works have shown that DDP is able to discover locomotion gaits applied on a real quadruped [18]. In [19], DDP is coupled with Monte Carlo tree search to compute the bipedal locomotion pattern of an avatar. While not yet demonstrated on a real humanoid, we might wonder whether this should be pushed further, instead of relying on a decoupling between contact computation, centroidal and whole-body optimization. We believe that DDP is a mature solution to replace IK/ID and is very complementary to centroidal optimization. Indeed, contact and centroidal problems can be efficiently handled within a global search thanks to the low dimension, while DDP is efficient to accurately handle the whole-body dynamics in large space but locally.

The rest of the paper is organized as follows: after discussing the locomotion framework in which our method takes place, we describe and justify in deep our technical choices in Section II. Section III briefly introduces the DDP algorithm, we then describe our novel DDP formulation for rigid contact dynamics in Section IV. Then, in Section V we show experimental trials and realistic simulation on the HRP-2 robot and compare them against a whole-body IK solver. Last, Section VI summarizes the work conclusions.

II. MULTI-CONTACT MOTION GENERATION

Locomotion synthesis is a hard problem because of a) the combinatorial nature of contact planning, b) the high-dimensionality of the search-space, c) the instabilities, discontinuities and non-convexity of the robot dynamics, d) the non-convexity of the terrain environment, among others. To synthesize a multi-contact motion, we follow a multi-stage strategy that decouples the global problem into various subproblems of smaller dimensions which are easier to solve in real-time as introduced in [20]. The first two stages describe an interactive acyclic contact planner [21]. Later a third stage computes a dynamic-physical centroidal trajectory while taking the discrete contact sequence as input. The centroidal pattern generator is formulated inside an Optimal Control (OC) framework [20], [4]. Fig. 1 illustrates the scheme of our multi-stage locomotion pipeline.

Henceforth, we focus on the whole-body motion generation. For that, we present an OC formulation under a generic form in order to describe this problem. This form is not suitable for efficient resolution of the problem, but it

helps us to sustain our particular technical choices. Finally we describe our proposed formulation which makes an interesting trade-off between efficiency and complexity.

A. Generic whole-body OC problem

We consider a floating-base system of $6 + n_j$ Degrees of Freedom (DoF). Its configuration vector $\mathbf{q} \in SE(3) \times \mathbb{R}^{n_j}$ describes the placement of the floating-base relatively to the inertial frame \mathcal{W} , $\mathbf{p} = (\mathbf{R}, \mathbf{r}) \in SE(3)$, and the joint configuration $\mathbf{q}_j \in \mathbb{R}^{n_j}$. The first and second time derivatives of \mathbf{q} belong to the so-called tangent bundle of the configuration manifold and they generally have a dimension different from the one of \mathbf{q} ; we choose to describe them as \mathbf{v} and $\dot{\mathbf{v}}$, respectively. These derivatives correspond to $\mathbf{v} = (\dot{\mathbf{r}}, \boldsymbol{\omega}, \dot{\mathbf{q}}_j)$ and $\dot{\mathbf{v}} = (\ddot{\mathbf{r}}, \dot{\boldsymbol{\omega}}, \ddot{\mathbf{q}}_j)$, where \mathbf{r} and $\boldsymbol{\omega}$ are the floating-base linear and angular velocity, respectively. As shown in [22], the Lagrangian dynamics may get simplified if it is expressed using centroidal coordinates:

$$\begin{bmatrix} m\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times n_j} \\ \mathbf{0}_{3 \times 3} & \bar{\mathbf{I}}_r & \mathbf{0}_{3 \times n_j} \\ \mathbf{0}_{n_j \times 3} & \mathbf{0}_{n_j \times 3} & \bar{\mathbf{M}}_j \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{r}} \\ \dot{\boldsymbol{\omega}} \\ \ddot{\mathbf{q}}_j \end{bmatrix} + \begin{bmatrix} m\mathbf{g} \\ \bar{\mathbf{I}}_r \boldsymbol{\omega} \\ \bar{\mathbf{h}}_j \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} + \bar{\mathbf{J}}_c^T \boldsymbol{\lambda}, \quad (1)$$

where m is the total mass, $\bar{\mathbf{I}}_r \in \mathbb{R}^{3 \times 3}$ is the total rotational inertia expressed around the robot CoM, and the contact Jacobian expressed in centroidal coordinates has the form $\bar{\mathbf{J}}_{c_i} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & [\mathbf{r}_i^b - \mathbf{r}]_{\times} & \mathbf{J}_i - \mathbf{J}_c \end{bmatrix} \in \mathbb{R}^{3 \times 6 + n_j}$. Note that $[\cdot]$ operator denotes matrices/vectors recomputed after the coordinate transform, and $[\cdot]_{\times}$ is the skew-symmetric operator. Additionally, $\mathbf{r}_i^b \in \mathbb{R}^3$ is the i^{th} contact point position expressed in \mathcal{B} , $\mathbf{J}_c = \partial \mathbf{r} / \partial \mathbf{q}_j$ maps joint velocities into CoM velocities expressed in \mathcal{B} , \mathbf{J}_i maps joint velocities into the Cartesian velocity of the i^{th} contact, \mathbf{g} is the gravity vector, $\bar{\mathbf{M}}_j$ and $\bar{\mathbf{h}}_j$ are respectively the joint-space inertia matrix and the Coriolis and centrifugal terms resulting from the centroidal coordinate transform.

Generating a whole-body trajectory requires to find a valid trajectory of the centroidal momenta along its joint motion. A valid trajectory might be defined whenever it satisfies the dynamic-consistency, the friction-cone constraints, the self-collision avoidance and the joint limits. From (1) the first two rows describe the evolution of the centroidal linear and angular momenta $\mathcal{H} = (\mathcal{P}, \mathcal{L}) \in \mathbb{R}^6$, and they represent the under-actuated part of the dynamics (a.k.a. centroidal dynamics). One can formulate this huge dimensional problem as a single OC problem:

$$\begin{cases} \left\{ \mathbf{x}_0^*, \dots, \mathbf{x}_N^* \right\} \\ \left\{ \mathbf{u}_0^*, \dots, \mathbf{u}_N^* \right\} \end{cases} = \arg \min_{\mathbf{x}, \mathbf{u}} \sum_{k=1}^N \int_{t_k}^{t_k + \Delta t} l_k(\mathbf{x}, \mathbf{u}) dt$$

s.t. $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda})$,
 $\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}, \boldsymbol{\lambda} \in \mathcal{K}.$ (2)

The state and its time derivative are $\mathbf{x} = (\mathbf{q}, \mathbf{v})$ and $\dot{\mathbf{x}} = (\mathbf{v}, \dot{\mathbf{v}})$, respectively, and the control is defined as the torque commands $\mathbf{u} = \boldsymbol{\tau}$. Additionally, \mathcal{X} , \mathcal{U} and \mathcal{K} are the admissible sets: joint configurations and joint velocities bounds, joint torque commands limits, and contact forces

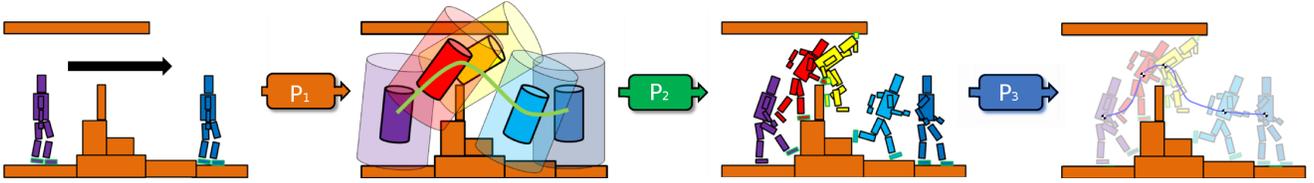


Fig. 1: Overview of our multi-stage locomotion framework [20]. Given a requested path request between start and goal positions (left image), \mathcal{P}_1 is the problem of computing a guide path in the space of equilibrium feasible root configurations. We achieve this by defining a geometric condition, the reachability condition (abstracted with the transparent cylinders on the middle image). \mathcal{P}_2 is then the problem of extending the path into a discrete sequence of contact configurations. Finally, \mathcal{P}_3 attends to compute a dynamic-physical whole-body trajectory given as input the discrete contact sequence.

constraints (e.g. the friction cone constraint), respectively. Despite the fact that it represents a compact and unified formulation, it is not the most efficient way of solving it as we discuss below.

In our previously explained centroidal trajectory optimization [4], we enforce friction cone constraints and the kinematics feasibility of the trajectory (self-collision and joint limits) using data-driven techniques. However, we are unable to consider the right angular momentum effect produced by the limb motions. Consider the second row of (1). If the contact forces are zero, then we observe a conservation of the angular momentum as a non-holonomic constraint i.e.

$$\sum_{k=0}^{n_j} m_i [\mathbf{r}_k - \mathbf{r}]_{\times} \dot{\mathbf{x}}_k + \mathbf{R}_k \mathbf{I}_k \boldsymbol{\omega}_k = \text{Constant}, \quad (3)$$

where k denotes the index of a rigid limb and \mathbf{I}_k corresponds to its inertia matrix expressed in the body's CoM frame. $\dot{\mathbf{x}}_k$ and $\boldsymbol{\omega}_k$ are the linear and angular velocities of the body in \mathcal{W} . From this equation, we can conclude that it is possible to regulate the robot orientation by only changing the joint configuration. As a consequence of this, in [4] we aim to lower this quantity (regulation around a zero reference).

While, we would like to account for this effect during the optimization process, it is a hard problem to solve in real-time. Instead we can assume that this effect is small, but yet important to consider, and track it with a whole-body Model Predictive Control (MPC). DDP is a reasonable choice between the two, because it allows us to optimize both trajectory and control commands [23], and it uses the Bellman principle to exploit the sparse structure of the problem. DDP has been shown [24] to be efficient in solving online OC in legged systems.

III. DIFFERENTIAL DYNAMIC PROGRAMMING

In this section, we give a formal description of the DDP algorithm for completeness. For more elaborate explanations and derivations, the reader is referred to [23]. DDP belongs to the family of OC handled with a sparse structure thanks to the Bellman principle. Concretely speaking, instead of finding the entire optimal trajectory (2), the Bellman principle makes recursively individual decisions:

$$V_i(\mathbf{x}_i) = \min_{\mathbf{u}_i} [l(\mathbf{x}_i, \mathbf{u}_i) + V_{i+1}(\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i))], \quad (4)$$

This is possible through a forward simulation of the system dynamics $\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$. Note that V_i denotes the value function which describes the minimum cost-to-go:

$$V_i(\mathbf{x}_i) = \min_{\mathbf{u}_{i:N-1}} J_i(\mathbf{x}_i, \mathbf{u}_{i:N-1}). \quad (5)$$

DDP searches locally the optimal state and control sequences of the above problem. For that aim, it uses a quadratic approximation $\mathbf{Q}(\delta\mathbf{x}, \delta\mathbf{u})$ of the differential change in (4), i.e.

$$\mathbf{Q}(\delta\mathbf{x}, \delta\mathbf{u}) \approx \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}^{\top} \begin{bmatrix} \mathbf{0} & \mathbf{Q}_{\mathbf{x}}^{\top} & \mathbf{Q}_{\mathbf{u}}^{\top} \\ \mathbf{Q}_{\mathbf{x}} & \mathbf{Q}_{\mathbf{xx}} & \mathbf{Q}_{\mathbf{xu}} \\ \mathbf{Q}_{\mathbf{u}} & \mathbf{Q}_{\mathbf{ux}} & \mathbf{Q}_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} \quad (6)$$

where

$$\begin{aligned} \mathbf{Q}_{\mathbf{x}} &= \mathbf{l}_{\mathbf{x}} + \mathbf{f}_{\mathbf{x}}^{\top} \mathbf{V}'_{\mathbf{x}}, \\ \mathbf{Q}_{\mathbf{u}} &= \mathbf{l}_{\mathbf{u}} + \mathbf{f}_{\mathbf{u}}^{\top} \mathbf{V}'_{\mathbf{x}}, \\ \mathbf{Q}_{\mathbf{xx}} &= \mathbf{l}_{\mathbf{xx}} + \mathbf{f}_{\mathbf{x}}^{\top} \mathbf{V}'_{\mathbf{xx}} \mathbf{f}_{\mathbf{x}} + \mathbf{V}'_{\mathbf{x}} \mathbf{f}_{\mathbf{xx}}, \\ \mathbf{Q}_{\mathbf{uu}} &= \mathbf{l}_{\mathbf{uu}} + \mathbf{f}_{\mathbf{u}}^{\top} \mathbf{V}'_{\mathbf{xx}} \mathbf{f}_{\mathbf{u}} + \mathbf{V}'_{\mathbf{x}} \mathbf{f}_{\mathbf{uu}}, \\ \mathbf{Q}_{\mathbf{ux}} &= \mathbf{l}_{\mathbf{ux}} + \mathbf{f}_{\mathbf{u}}^{\top} \mathbf{V}'_{\mathbf{xx}} \mathbf{f}_{\mathbf{x}} + \mathbf{V}'_{\mathbf{x}} \mathbf{f}_{\mathbf{ux}}, \end{aligned} \quad (7)$$

and the primes denotes the values at the next time-step.

A. Backward pass

The backward pass determines the search direction of the Newton step by recursively solving (4). In an unconstrained setting the solution is:

$$\delta\mathbf{u}^* = \arg \min_{\delta\mathbf{u}} \mathbf{Q}(\delta\mathbf{x}, \delta\mathbf{u}) = \mathbf{k} + \mathbf{K}\delta\mathbf{x}, \quad (8)$$

where $\mathbf{k} = -\mathbf{Q}_{\mathbf{uu}}^{-1} \mathbf{Q}_{\mathbf{u}}$ and $\mathbf{K} = -\mathbf{Q}_{\mathbf{uu}}^{-1} \mathbf{Q}_{\mathbf{ux}} \delta\mathbf{x}$ are the feed-forward and feedback terms. Recursive updates of the derivatives of the value function are done as follows:

$$\begin{aligned} \mathbf{V}_{\mathbf{x}}(i) &= \mathbf{Q}_{\mathbf{x}} + \mathbf{K}^{\top} \mathbf{Q}_{\mathbf{uu}} \mathbf{k} + \mathbf{K}^{\top} \mathbf{Q}_{\mathbf{u}} + \mathbf{Q}_{\mathbf{ux}}^{\top} \mathbf{k}, \\ \mathbf{V}_{\mathbf{xx}}(i) &= \mathbf{Q}_{\mathbf{xx}} + \mathbf{K}^{\top} \mathbf{Q}_{\mathbf{uu}} \mathbf{K} + \mathbf{K}^{\top} \mathbf{Q}_{\mathbf{ux}} + \mathbf{Q}_{\mathbf{ux}}^{\top} \mathbf{K}. \end{aligned} \quad (9)$$

B. Forward pass

The forward pass determines the step size along the Newton direction by adjusting the line search parameter α . It computes a new trajectory by integrating the dynamics along the computed feed-forward and feedback commands $\{\mathbf{k}_i, \mathbf{K}_i\}$:

$$\begin{aligned} \hat{\mathbf{u}}_i &= \mathbf{u}_i + \alpha \mathbf{k}_i + \mathbf{K}_i (\hat{\mathbf{x}}_i - \mathbf{x}_i), \\ \hat{\mathbf{x}}_{i+1} &= \mathbf{f}(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i), \end{aligned} \quad (10)$$

in which $\hat{\mathbf{x}}_1 = \mathbf{x}_1$, and $\{\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i\}$ are the new state-control pair. Note that if $\alpha = 0$, it does not change the state and control trajectories.

C. Line search and regularization

We perform a *backtracking* line search by trying the full step ($\alpha = 1$) first. The choice of α is dual to the choice of regularization terms, and both are updated between subsequent iterations to ensure a good progress toward the (local) optimal solution. We use two regularization schemes: the Tikhonov regularization (over $\mathbf{Q}_{\mathbf{u}\mathbf{u}}$) and its update using the Lavenberg-Marquardt algorithm are typically used [25]. Tassa et al. [24] propose a regularization scheme over $\mathbf{V}_{\mathbf{x}\mathbf{x}}$, which is equivalent to adding a penalty in the state changes.

IV. DDP WITH CONSTRAINED ROBOT DYNAMICS

A. Contact dynamics

Let's consider the case of rigid contact dynamics with the environment. Given a predefined contact sequence, rigid contacts can be formulated as holonomic scleronomic constraints to the robot dynamics (i.e. equality-constrained dynamics). The unconstrained robot dynamics is typically represented as:

$$\mathbf{M}\dot{\mathbf{v}}_{free} = \boldsymbol{\tau}_b, \quad (11)$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the joint-space inertia matrix, $\dot{\mathbf{v}}_{free}$ is the unconstrained joint acceleration vector, $\boldsymbol{\tau}_b = \mathbf{S}\boldsymbol{\tau} - \mathbf{b} \in \mathbb{R}^n$ is the force-bias vector that accounts for the control $\boldsymbol{\tau}$, the Coriolis and gravitational effects \mathbf{b} , and \mathbf{S} is the selection matrix of the actuated joint coordinates.

We can account for the rigid contact constraints by applying the Gauss principle of least constraint [26], [13]. Under this principle, the constrained motion evolves in such a way that it minimizes the deviation in acceleration from the unconstrained motion \mathbf{a}_{free} , i.e.:

$$\begin{aligned} \dot{\mathbf{v}} = \arg \min_{\mathbf{a}} \quad & \frac{1}{2} \|\dot{\mathbf{v}} - \dot{\mathbf{v}}_{free}\|_{\mathbf{M}} \\ \text{subject to} \quad & \mathbf{J}_c \dot{\mathbf{v}} + \dot{\mathbf{J}}_c \mathbf{v} = \mathbf{0}, \end{aligned} \quad (12)$$

in which \mathbf{M} is formally the metric tensor over the configuration manifold \mathbf{q}^1 . Note that we differentiate twice the holonomic contact constraint $\phi(\mathbf{q})$ in order to express it in the acceleration space. In other words, the rigid contact condition is expressed by the second-order kinematic constraints on the contact surface position. $\mathbf{J}_c = [\mathbf{J}_{c_1} \ \cdots \ \mathbf{J}_{c_f}] \in \mathbb{R}^{kp \times n}$ is a stack of the f contact Jacobians.

B. Karush-Kuhn-Tucker (KKT) conditions

The Gauss minimization in (12) corresponds to an equality-constrained convex optimization problem², and it has a unique solution if \mathbf{J}_c is full-rank. The primal and dual optimal solutions $(\dot{\mathbf{v}}, \boldsymbol{\lambda})$ must satisfy the so-called KKT conditions given by

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_b \\ -\dot{\mathbf{J}}_c \mathbf{v} \end{bmatrix}. \quad (13)$$

¹The dimension of the configuration manifold \mathbf{q} and its tangent \mathbf{v} are not the same in general (e.g. system evolving in a $SE(3)$ manifold).

² \mathbf{M} is a positive-definite matrix.

These dual variables $\boldsymbol{\lambda}^k \in \mathbb{R}^p$ render themselves nicely in mechanics as the external forces at the contact level. This relationship allows us to express the contact forces directly in terms of the robot state and actuation. In other words, this would free the solver to find an unconstrained solution to the KKT dynamics (13), without worrying about the contact constraint. Fast iterative Newton and quasi-Newton methods can then be easily applied to achieve real-time performance.

C. KKT-based DDP algorithm

From (13), we can see the augmented KKT dynamics as a function of the state \mathbf{x}_i and the control \mathbf{u}_i :

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i), \\ \boldsymbol{\lambda}_i &= \mathbf{g}(\mathbf{x}_i, \mathbf{u}_i), \end{aligned} \quad (14)$$

where the state $\mathbf{x} = (\mathbf{q}, \mathbf{v})$ is represented by the configuration vector and its tangent velocity, \mathbf{u} is the torque-input vector, and $\mathbf{g}(\cdot)$ is the dual solution of (13). In case of legged robots, the placement of the free-floating link is described using the special Euclidean group $SE(3)$.

Given a reference trajectory for the contact forces, the DDP backward-pass cost and its respective Hessians (see (4) and (7)) are updated as follows:

$$J_i(\mathbf{x}_i, \mathbf{U}_i) = l_f(\mathbf{x}_N) + \sum_{k=i}^{N-1} l(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\lambda}_k), \quad (15)$$

where $\mathbf{U}_i = \{\mathbf{u}_i, \mathbf{u}_{i+1}, \dots, \mathbf{u}_{N-1}\}$ is the tuple of controls that acts on the system dynamics at time i , and the Gauss-Newton approximation of the \mathbf{Q} coefficients (i.e. first-order approximation of $\mathbf{g}(\cdot)$ and $\mathbf{f}(\cdot)$) are

$$\begin{aligned} \mathbf{Q}_x &= \mathbf{l}_x + \mathbf{g}_x^\top \mathbf{l}_\lambda + \mathbf{f}_x^\top \mathbf{V}'_{\mathbf{x}}, \\ \mathbf{Q}_u &= \mathbf{l}_u + \mathbf{g}_u^\top \mathbf{l}_\lambda + \mathbf{f}_u^\top \mathbf{V}'_{\mathbf{x}}, \\ \mathbf{Q}_{xx} &\approx \mathbf{l}_{xx} + \mathbf{g}_x^\top \mathbf{l}_{\lambda x} + \mathbf{f}_x^\top \mathbf{V}'_{xx} \mathbf{f}_x, \\ \mathbf{Q}_{uu} &\approx \mathbf{l}_{uu} + \mathbf{g}_u^\top \mathbf{l}_{\lambda u} + \mathbf{f}_u^\top \mathbf{V}'_{xx} \mathbf{f}_u, \\ \mathbf{Q}_{ux} &\approx \mathbf{l}_{ux} + \mathbf{g}_u^\top \mathbf{l}_{\lambda x} + \mathbf{f}_u^\top \mathbf{V}'_{xx} \mathbf{f}_x. \end{aligned} \quad (16)$$

The set of equations (16) takes into account the trajectory of the rigid contact forces inside the backward-pass. The system evolution needed in the forward-pass is described by (14).

V. RESULTS

In this section, we show that our DDP formulation can generate whole-body motions which require regulation of the angular momentum. The performance of our algorithm is assessed on realistic simulations and aggressive experimental trials on the HRP-2 robot. First, we perform very large strides (from 80 to 100cm) which require large amount of angular momentum (due to the fast swing of the 6-kg leg) and reach the HRP-2 limits. Then, we show how our method can regulate the robot attitude in absence of contact forces and gravitation field. These motions cannot be generated through a standard time-invariant IK/ID solver, as the system becomes non-holonomic as shown in (3).

All the motions were computed offline. Contact sequence [21] and the centroidal trajectory [27] are pre-computed and provided to the solver for the large stride

experiments. We used the standard controller OpenHRP [28] for tracking the motions on the real robot. The large strides produced by DDP are compared with those produced by an IK solver [10], showing the benefit of our approach.

A. Large stride on a flat ground

In these experiments, we generate a sequence of cyclic contact for 80 cm to 100 cm stride. These are very big steps for HRP-2 compared to its height (160 cm). For the contact location, we use the OC solver reported in [3] to compute the contact timings and the centroidal trajectory. Then we use our proposed DDP to generate the full robot motion.

The cost function is composed of various tasks in order to keep balance and to increase efficiency and stability: (a) CoM, foot position and orientation and contact forces tracking of centroidal motion, (b) torque commands minimization and (c) joint configuration and velocity regularization. All these tasks are residual functions of the state, control and contact forces which are penalized quadratically, i.e. $\|r_i(\mathbf{x}, \mathbf{u}, \lambda)\|_{Q_i}$. The evolution of the different normalized task costs with iterations is shown in Fig. 2. Our method adapts the CoM to create a more efficient torque and contact force trajectory.

Increasing the upper-body angular momentum helps to counterbalance the swing leg motion, this in turn reduces GRFs and improves the locomotion stability. Our experimental results show a reduction on the GRFs peaks compared to the IK solver. Fig. 3 shows the measured normal contact forces and the knee torques in case of DDP solver and IK solvers. Our DDP reduced the normal forces peaks of the IK solver from 895 N to 755 N. This represents a significant improvement, considering that the minimum possible contact forces are 650 N (the total mass of the HRP-2 robot is 65 kg) and the maximum safe force allowed by the sensors on the foot is 1000 N. An overview of the motion is shown in Fig. 4.

B. Attitude regulation through joint motion

The angular momentum equation (3) shows that it is possible to regulate the robot attitude without the need of contact forces [13]. It can be seen that the gravity field does not affect this property. Thus, we analyze how our

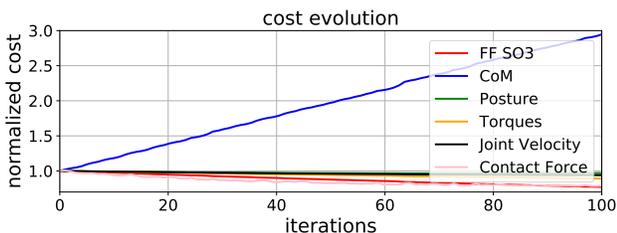


Fig. 2: Evolution of the different cost functions (normalized) with respect to iterations. DDP reduces the applied torques by recalculating the CoM tracking. It improves the contact force by taking into account the whole-body angular momentum. The result is a continuous improvement in the performance as compared to IK. We stop after 100 iterations.

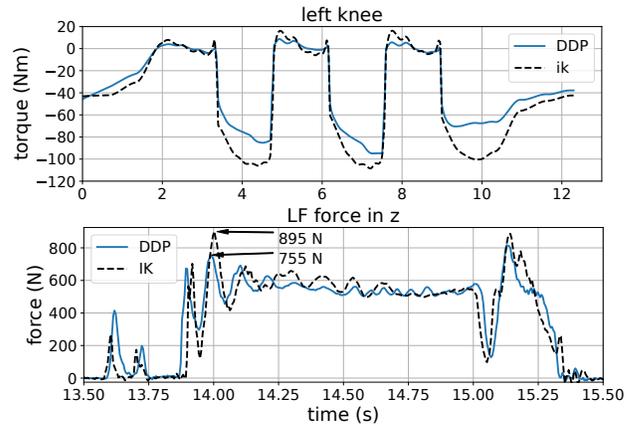


Fig. 3: Comparison between the IK and DDP trajectory for 100 cm stride on the HRP-2 robot. *Top*: Knee torques generated in the left leg. *Bottom*: Ground Reaction Forces (GRFs) generated in the left foot. The DDP formulation allows us to utilize the angular momentum of the upper body, which reduces the requirement on the lower body to create a counterbalancing motion. This results in a lower torque in the lower body, as well as lower GRFs. Around $t = 14s$ we can see high peaks for the IK and DDP trajectories of 895 N and 755 N, respectively.

DDP solver regulates the attitude in zero-gravity condition, we named this task *astronaut reorientation*. The astronaut reorientation (similar to cat falling) is an interesting motor task due to fact that it depends on a proper exploitation of the angular momentum based on the coordination of arms and legs motions. Fig. 5 demonstrates the motion found by the solver to rotate the body 360° . Unlike an instantaneous tracking solver like IK, the solver is willing to bend in the opposite direction, in order to obtain an ability to create sufficient angular momentum by the legs. It is important here to note that such motion cannot be obtained by a time-invariant control law which does not take the future control trajectory into account.

The cost matrices for this problem require a barrier function on the robot configuration to avoid self-collision. Final cost on the body orientation provides the goal, and a running cost on the posture is added for regularization. No warm start is given to the solver, the initial control trajectory is a set of zero vectors. For the ease of demonstration, we used only the leg joints in the sagittal plane. Fig. 6 shows the torques produced by the hip and the knee joints. Our method creates a rotation of the upper body by a quick initial motion in the legs. Then it maintains the angular velocity by small correctional torque inputs during the rest of the trajectory. At the end, to bring the rotation to a halt, the same behaviour is repeated in the reverse.

VI. CONCLUSION

Typically, reduced centroidal trajectory optimization does not take into account the angular momentum produced by the limb motions. Proper regulation of the angular momentum



Fig. 4: Snapshots of 100 cm stride on a flat terrain used to evaluate the performance of our whole-body trajectory optimization method. The DDP trajectory reduces significantly the normal forces peaks compared with classic whole-body IK.



Fig. 5: Attitude adjustment maneuver conducted by the robot in gravity free space. DDP solver takes into account the non-holonomic angular momentum constraint and uses internal actuation to rotate 360° without the need for contact forces.

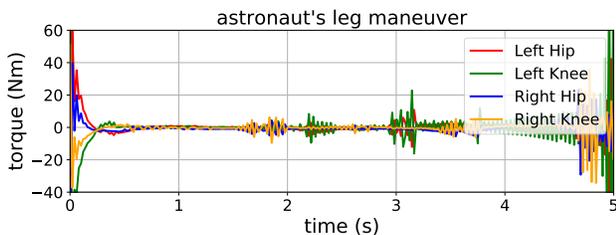


Fig. 6: Joint torques for the astronaut maneuver. Our method plans a smart strategy by kick-starting the rotation, and then tries to maintain the velocity by small bang-bang control signals. Towards the end, it changes again the velocity of the lower legs in order to bring the system to a stop.

exploits the counterbalancing effect in order to reduce the contact forces and torque commands. It also improves the stability during flight phases where the momentum control can only be made through joint motions. Also, it is well known that under absence of external forces the momentum conservation represents a nonholonomic constraint. And this system cannot be controlled with simple time-invariant feedback law. OC provides the required tools for solving it, however, this big problem is hard to solve in real-time. Thus, we have proposed a whole-body generation approach that splits the problem into the under-actuated and actuated dynamics. Our whole-body locomotion framework is an extension of our previous work [3].

In this paper, we have also proposed a novel DDP formulation based on the augmented KKT dynamics (see (14)) which is a product of holonomic contact constraints. It represents the first application of motion generated by DDP solver on a real humanoid locomotion. Our whole-body motion generation pipeline enables us to potentially regulate angular momentum dynamics during the wholebody motion in real-time. We have observed a reduction of the contact forces compared to the IK solver, even though we had to restrict the angular momentum in the sagittal plane, for the stride on flat ground task, due to robot limits. While these first

results look promising, there is still a considerable extra work such as: efficient DDP based on the analytical derivatives of rigid-body dynamics [29], faster numerical convergence through a better globalization strategy, robustness study against model errors, etc. A more revealing experiment, the *astronaut reorientation*, demonstrates further the limits of the previous approaches and the advantages of using DDP. The solver generates the desired motion from scratch in this case by manipulating joint velocities within the non-holonomic angular momentum constraints.

REFERENCES

- [1] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, 2013.
- [2] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE International Conference on Humanoid Robots*, 2014.
- [3] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, "A versatile and efficient pattern generator for generalized legged locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [4] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility Constraints for Multicontact Locomotion of Legged Robots," in *Robotics: Science and Systems (RSS)*, 2017.
- [5] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *IEEE International Conference on Humanoid Robots*, 2015.
- [6] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Planning and Execution of Dynamic Whole-Body Locomotion for a Hydraulic Quadruped on Challenging Terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [7] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and Foothold Optimization using Low-Dimensional Models for Rough Terrain Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [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 (RAL)*, 2018.
- [9] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernandez-Lopez, and C. Semini, "Simultaneous Contact, Gait and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization," *IEEE Robotics and Automation Letters (RAL)*, 2018.

- [10] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soures, and J. Y. Fourquet, "Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints," *IEEE Transactions on Robotics*, 2013.
- [11] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid," *Autonomous Robots*, 2016.
- [12] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Motion planning for quadrupedal locomotion: coupled planning, terrain mapping and whole-body control," 2017, working paper or preprint.
- [13] P.-B. Wieber, "Holonomy and nonholonomy in the dynamics of articulated motion," in *Proceedings of the Ruperto Carola Symposium on Fast Motion in Biomechanics and Robotics*, 2005.
- [14] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, 1983.
- [15] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [16] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [17] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [18] M. Neunert, M. Stauble, M. Gifftthaler, 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 (RAL)*, 2018.
- [19] J. Rajamäki, K. Naderi, V. Kyrki, and P. Hämmäläinen, "Sampled differential dynamic programming," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [20] J. Carpentier, A. D. Prete, S. Tonneau, T. Flayols, F. Forget, A. Mifsud, K. Giraud, D. Atchuthan, P. Fernbach, R. Budhiraja, M. Geisert, J. Solà, O. Stasse, and N. Mansard, "Multi-contact Locomotion of Legged Robots in Complex Environments The Loco3D project," 2017, RSS Workshop on Challenges in Dynamic Legged Locomotion.
- [21] S. Tonneau, A. D. Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An Efficient Acyclic Contact Planner for Multiped Robots," *IEEE Transactions on Robotics (TRO)*, 2018.
- [22] C. Ott, M. A. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *IEEE International Conference on Humanoid Robots*, 2011.
- [23] D. Q. Mayne, "Differential Dynamic Programming A Unified Approach to the Optimization of Dynamic Systems," *Control and Dynamic Systems*, 1973.
- [24] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [25] M. Toussaint, *A Tutorial on Newton Methods for Constrained Trajectory Optimization and Relations to SLAM, Gaussian Process Smoothing, Optimal Control, and Probabilistic Inference*. Springer, 2017.
- [26] F. Udawadia and R. Kalaba, "A new perspective on constrained motion," in *Mathematical and Physical Sciences*, 1992.
- [27] J. Carpentier and N. Mansard, "Multi-contact locomotion of legged robots," *IEEE Transactions on Robotics (To Appear)*, 2018.
- [28] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture humanoid robotics platform," *The International Journal of Robotics Research*, 2004.
- [29] J. Carpentier and N. Mansard, "Analytical Derivatives of Rigid Body Dynamics Algorithms," in *Robotics: Science and Systems (RSS)*, 2018.