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. IEEE-RAS International Conference on Humanoid Robots (Humanoids 2018), Nov 2018, Beijing, China. 10.1109/HUMANOIDS.2018.8624925 . hal-01851596v2

HAL Id: hal-01851596

https://hal.science/hal-01851596v2

Submitted on 13 Oct 2018

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 [1], Justin Carpentier [1,2,3], Carlos Mastalli [1] and Nicolas Mansard [1]

***Abstract*— A common strategy 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 step computes the whole-body trajectory that follows the centroidal pattern. While the second step is generally handled by a simple program such as an inverse kinematics solver, we propose in this paper 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 contact 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], [7] while other strategies optimize the centroidal trajectory and contact information together [8], [9], [10]. 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 usually solve quadratic optimization problems written with task-space dynamics (Inverse Kinematics (IK) / Inverse Dynamics (ID)) [11], [12], [13]. While this scheme has shown great experimental results (e.g. [3], [5], [14]), it is still not able to correctly handle the AM produced by the extremities of the limbs. This is notable for humanoid robots which have important masses in the limbs that are put in motion during (for instance) walking. This effect is neither properly captured by the centroidal model, nor by the instantaneous time-invariant linearization.

In [12] an alternative scheme aims to compensate the AM 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 AM, it is necessary 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 struggle to discover 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 AM 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 [15] 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 AM conservation can be seen as a non-holonomic constraint on the robot orientation. Of course, one can design a control law that counterbalances the lower-body AM. However this will create tracking errors (and potentially instabilities) without mentioning the cases where the arms need to be used for multi-contact locomotion. In fact, as shown in [16], a system under non-holonomic constraints cannot be controlled with a time-invariant feedback law. Thus AM 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 the AM as needed. For instance, the centroidal optimizer cannot anticipate a high demand of the linear part (Center of Mass (CoM)) by delaying the limb movement, or exploit the movement of the arms to compensate for large forces acting for a short duration. Nonetheless, these methods are still valid since they provide an efficient way to compute the CoM motion while keeping balance and avoiding slippage.

C. Overview of our method

Instead of relying on instant linearization using IK/ID, we propose to rely on optimal control [18] [19], namely Differential Dynamic Programming (DDP), to compute the whole-body motion while tracking the centroidal trajectory.

[1] CNRS, LAAS, 7 Avenue du Colonel Roche, Toulouse, France.
[2] Département d'informatique de l'ENS, École Normale Supérieure, CNRS, PSL Research University, Paris, France.
[3] INRIA, France.
email: rohan.budhiraja@laas.fr.

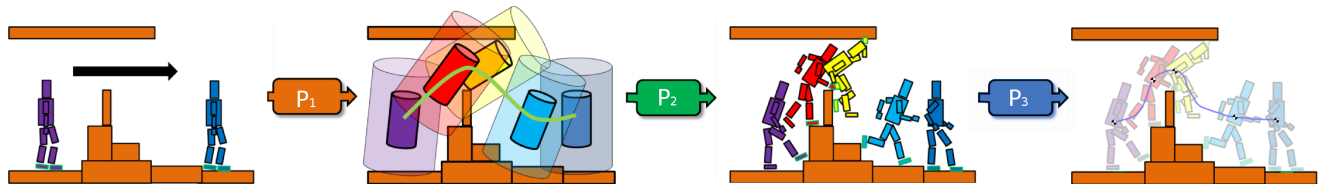

Fig. 1: Overview of our multi-stage locomotion framework [17]. Given a requested path request between start and goal positions (left image), $\mathcal{P}_1$ is the problem of computing a guide path in the space of equilibrium feasible root configurations. $\mathcal{P}_2$ is then the problem of extending 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.

DDP has been made popular by the proof of concept [20], and by the demonstration in simulation that it can meet the control-loop timings constraint [21]. However, locomotion movements computed by DDP have not yet been transferred to a real full-size humanoid. Contrary to [20] that optimizes the motion from scratch with a regularized dynamics (thanks to a smooth contact model [22]), we propose to impose the contact phases as decided by the centroidal optimization. As DDP does not need to discover the contact switching instants, we can use 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 [23]. In [24], 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 a 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 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. Lastly, Section VI summarizes the work conclusions.

II. Multi-contact Motion Generation

Locomotion synthesis is a difficult 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. In our previous works, we have proposed a multi-stage strategy that decouples the global problem into successive subproblems of smaller dimensions [17]. The global problem is thus split into an interactive acyclic contact planner [25], a centroidal pattern generator [4] [5] which takes the contact sequence as an input, and a whole-body motion generator.

Centroidal pattern generator [4] [5] by itself is unable to account for the AM effect generated by the limb motions. Indeed, AM of a body is accounted by both, as a result of the contact forces, and by *only* the limb movement. Consider a floating-base robot. Even with no external forces acting on the robot, a constant AM can be maintained by the non-holonomic constraint on the joint velocities [15]:

$$\sum_{k=0}^{n_j} m_i[\mathbf{r}_k - \mathbf{r}]_\times \dot{\mathbf{x}}_k + \mathbf{R}_k \mathbf{I}_k \boldsymbol{\omega}_k = \text{Constant}, \tag{1}$$

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 links.

While we would like to account for this "gesticulation" [15] during the centroidal optimization, it is a difficult problem to solve in real-time. Instead, we assume that this effect is small [4], but yet important to consider, and we track it with a whole-body Model Predictive Control (MPC). DDP [26] is a reasonable choice between the two, because it allows us to generate additional AM using (1), while efficiently tracking the reference trajectories provided by the centroidal solver. DDP has been shown [27] to be efficient in solving online Optimal Control (OC) in legged systems. Fig. 1 demonstrates our multi-stage locomotion pipeline.

Generating a whole-body trajectory requires finding a trajectory which is subject to dynamic-consistency, the friction-cone constraints, the self-collision avoidance and the joint limits. This can be formulated as a single OC problem:

$$\begin{aligned} \begin{Bmatrix} \mathbf{x}_0^*, \cdots, \mathbf{x}_N^* \\ \mathbf{u}_0^*, \cdots, \mathbf{u}_N^* \end{Bmatrix} = \underset{\mathbf{X},\mathbf{U}}{\arg\min} \quad & \sum_{k=1}^{N} \int_{t_k}^{t_k+\Delta t} l_k(\mathbf{x},\mathbf{u}) dt \\ \text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}), \\ & \mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}, \boldsymbol{\lambda} \in \mathcal{K}. \end{aligned} \tag{2}$$

The state and control are defined as $\mathbf{x} = (\mathbf{q}, \mathbf{v})$ and $\mathbf{u} = \boldsymbol{\tau}$, where $\mathbf{q} \in SE(3) \times \mathbb{R}^{n_j}$ is the configuration vector for a floating base robot with n_j Degrees of Freedom (DoF). $\mathbf{v}$ is its derivative, $\boldsymbol{\tau}$ is the vector of joint torques, and $\boldsymbol{\lambda}$ is the friction force corresponding to $(\mathbf{x}, \mathbf{u})$. $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{K}$ are the admissible sets: joint configurations and joint velocities bounds, joint torque commands limits, and contact forces constraints (e.g. the friction cone constraint), respectively.

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 [26]. DDP belongs to the family of OC handled with a sparse structure thanks to the Bellman principle. Concretely, instead of finding the entire optimal trajectory (2), we make recursive decisions:

$$V_i(\mathbf{x}_i) = \min_{\mathbf{u}_i} [l(\mathbf{x}_i, \mathbf{u}_i) + V_{i+1}(\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i))], \tag{3}$$

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:

$$V_i(\mathbf{x}_i) = \min_{\mathbf{u}_{i:N-1}} J_i(\mathbf{x}_i, \mathbf{u}_{i:N-1}). \tag{4}$$

DDP searches locally the optimal state and control sequences of the above problem. It uses a quadratic approximation $\mathbf{Q}(\delta\mathbf{x}, \delta\mathbf{u})$ of the differential change in (3), i.e.

$$\mathbf{Q}(\delta\mathbf{x}, \delta\mathbf{u}) \approx \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}^\top \begin{bmatrix} \mathbf{0} & \mathbf{Q_x}^\top & \mathbf{Q_u}^\top \\ \mathbf{Q_x} & \mathbf{Q_{xx}} & \mathbf{Q_{xu}} \\ \mathbf{Q_u} & \mathbf{Q_{ux}} & \mathbf{Q_{uu}} \end{bmatrix} \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} \tag{5}$$

where

$$\begin{aligned} \mathbf{Q_x} &= \mathbf{l_x} + \mathbf{f_x}^\top \mathbf{V}'_\mathbf{x}, \\ \mathbf{Q_u} &= \mathbf{l_u} + \mathbf{f_u}^\top \mathbf{V}'_\mathbf{x}, \\ \mathbf{Q_{xx}} &= \mathbf{l_{xx}} + \mathbf{f_x}^\top \mathbf{V}'_\mathbf{xx} \mathbf{f_x} + \mathbf{V}'_\mathbf{x} \mathbf{f_{xx}}, \\ \mathbf{Q_{uu}} &= \mathbf{l_{uu}} + \mathbf{f_u}^\top \mathbf{V}'_\mathbf{xx} \mathbf{f_u} + \mathbf{V}'_\mathbf{x} \mathbf{f_{uu}}, \\ \mathbf{Q_{ux}} &= \mathbf{l_{ux}} + \mathbf{f_u}^\top \mathbf{V}'_\mathbf{xx} \mathbf{f_x} + \mathbf{V}'_\mathbf{x} \mathbf{f_{ux}}, \end{aligned} \tag{6}$$

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 (3). In an unconstrained setting the solution is:

$$\delta\mathbf{u}^* = \arg\min_{\delta\mathbf{u}} \mathbf{Q}(\delta\mathbf{x}, \delta\mathbf{u}) = \mathbf{k} + \mathbf{K}\delta\mathbf{x}, \tag{7}$$

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

$$\begin{aligned} \mathbf{V_x}(i) &= \mathbf{Q_x} + \mathbf{K}^\top \mathbf{Q_{uu}} \mathbf{k} + \mathbf{K}^\top \mathbf{Q_u} + \mathbf{Q_{ux}}^\top \mathbf{k}, \\ \mathbf{V_{xx}}(i) &= \mathbf{Q_{xx}} + \mathbf{K}^\top \mathbf{Q_{uu}} \mathbf{K} + \mathbf{K}^\top \mathbf{Q_{ux}} + \mathbf{Q_{ux}}^\top \mathbf{K}. \end{aligned} \tag{8}$$

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 $\{\mathbf{k}_i, \mathbf{K}_i\}$:

$$\begin{aligned} \hat{\mathbf{u}}_i &= \mathbf{u}_i + \alpha\mathbf{k}_i + \mathbf{K}_i(\hat{\mathbf{x}}_i - \mathbf{x}_i), \\ \hat{\mathbf{x}}_{i+1} &= \mathbf{f}(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i), \end{aligned} \tag{9}$$

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 $\mathbf{Q_{uu}}$) and its update using the Levenberg-Marquardt algorithm are typically used [28]. Tassa et al. [27] propose a regularization scheme over $\mathbf{V_{xx}}$, 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:

$$\mathbf{M}\dot{\mathbf{v}}_{free} = \boldsymbol{\tau}_b, \tag{10}$$

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 [29], [15]. Under this principle, the constrained motion evolves in such a way that it minimizes the deviation in acceleration from the unconstrained motion $\mathbf{a}_{\text{free}}$, i.e.:

$$\begin{aligned} \dot{\mathbf{v}} = \arg\min_{\mathbf{a}} \quad & \frac{1}{2}\|\dot{\mathbf{v}} - \dot{\mathbf{v}}_{free}\|_\mathbf{M} \\ \text{subject to} \quad & \mathbf{J}_c\dot{\mathbf{v}} + \dot{\mathbf{J}}_c\mathbf{v} = \mathbf{0}, \end{aligned} \tag{11}$$

in which $\mathbf{M}$ is formally the metric tensor over the configuration manifold $\mathbf{q}$. Note that we differentiate twice the holonomic contact constraint $\boldsymbol{\phi}(\mathbf{q})$ 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. $\mathbf{J}_c = \begin{bmatrix}\mathbf{J}_{c_1} & \cdots & \mathbf{J}_{c_f}\end{bmatrix} \in \mathbb{R}^{kp\times n}$ is a stack of the f contact Jacobians.

B. Karush-Kuhn-Tucker (KKT) conditions

The Gauss minimization in (11) corresponds to an equality-constrained convex optimization problem[1], 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

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_b \\ -\dot{\mathbf{J}}_c\mathbf{v} \end{bmatrix}. \tag{12}$$

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

[1] $\mathbf{M}$ is a positive-definite matrix.

in terms of the robot state and actuation. As compared to previous approaches which would introduce the contact constraints in the whole-body optimization [3] [11], here we solve for the contact constraints at the level of the dynamics, and not the solver. In other words, this would free the solver to find an unconstrained solution to the KKT dynamics (12), 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 (12), we can see the augmented KKT dynamics as a function of the state $\mathbf{x}_i$ and the control $\mathbf{u}_i$:

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i), \\ \boldsymbol{\lambda}_i &= \mathbf{g}(\mathbf{x}_i, \mathbf{u}_i), \end{aligned} \tag{13}$$

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 (12). 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 (3) and (6)) are updated as follows:

$$J_i(\mathbf{x}_i, \mathbf{U}_i) = l_f(\mathbf{x}_N) + \sum_{k=i}^{N-1} l(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\lambda}_k), \tag{14}$$

where $\mathbf{U}_i = \{\mathbf{u}_i, \mathbf{u}_{i+1}, \cdots, \mathbf{u}_{N-1}\}$ 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

$$\begin{aligned} \mathbf{Q_x} &= \mathbf{l_x} + \mathbf{g_x}^\top \mathbf{l}_{\boldsymbol{\lambda}} + \mathbf{f_x}^\top \mathbf{V}'_{\mathbf{x}}, \\ \mathbf{Q_u} &= \mathbf{l_u} + \mathbf{g_u}^\top \mathbf{l}_{\boldsymbol{\lambda}} + \mathbf{f_u}^\top \mathbf{V}'_{\mathbf{x}}, \\ \mathbf{Q_{xx}} &\approx \mathbf{l_{xx}} + \mathbf{g_x}^\top \mathbf{l}_{\boldsymbol{\lambda}\boldsymbol{\lambda}} \mathbf{g_x} + \mathbf{f_x}^\top \mathbf{V}'_{\mathbf{xx}} \mathbf{f_x}, \\ \mathbf{Q_{uu}} &\approx \mathbf{l_{uu}} + \mathbf{g_u}^\top \mathbf{l}_{\boldsymbol{\lambda}\boldsymbol{\lambda}} \mathbf{g_u} + \mathbf{f_u}^\top \mathbf{V}'_{\mathbf{xx}} \mathbf{f_u}, \\ \mathbf{Q_{ux}} &\approx \mathbf{l_{ux}} + \mathbf{g_u}^\top \mathbf{l}_{\boldsymbol{\lambda}\boldsymbol{\lambda}} \mathbf{g_x} + \mathbf{f_u}^\top \mathbf{V}'_{\mathbf{xx}} \mathbf{f_x}. \end{aligned} \tag{15}$$

The set of equations (15) 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 (13).

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 100 cm) which require large amount of angular momentum (due to the fast swing of the 6-kg leg) and reach the HRP-2 limits. This demonstrates the ability of the solver to handle contact constraints, as well as to *generate* excessive AM required by the leg motions. Then, to again emphasize the need for a horizon based optimizer such as DDP, and the ability of our solver to handle these AM requirements, we regulate the robot attitude in absence of contact forces

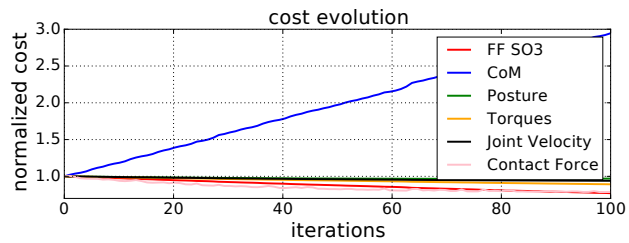

Fig. 2: Evolution of the different cost functions (normalized) with respect to iterations. DDP reduces the applied torques by recalculating the CoM tracking. It improves the contact force by taking into account the whole-body angular momentum. The result is a continuous improvement in the performance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

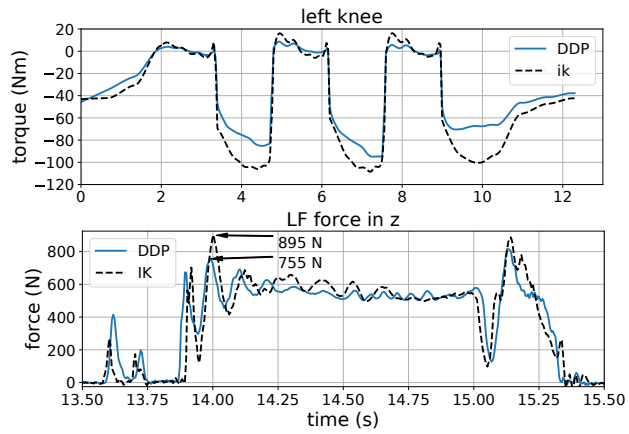

Fig. 3: Comparison between the IK and DDP trajectory for 100 cm stride on the HRP-2 robot. *Top*: Knee torques generated in the left leg. *Bottom*: Ground Reaction Forces (GRFs) generated in the left foot. The DDP formulation allows us to utilize the angular momentum of the upper body, which reduces 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.

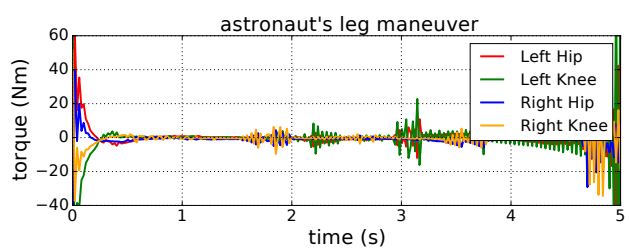

Fig. 4: Joint torques for the astronaut maneuver. Our method plans a smart strategy by kick-starting the rotation, and then tries to maintain the velocity by small bang-bang control signals. Towards the end, it changes again the velocity of the lower legs in order to bring the system to a stop.

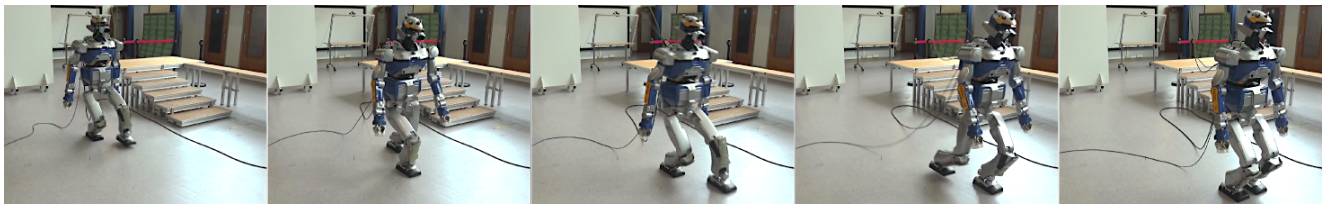

Fig. 5: Snapshots of 100 cm stride on a flat terrain used to evaluate the performance of our whole-body trajectory optimization method. The DDP trajectory reduces significantly the normal forces peaks compared with classic whole-body IK.

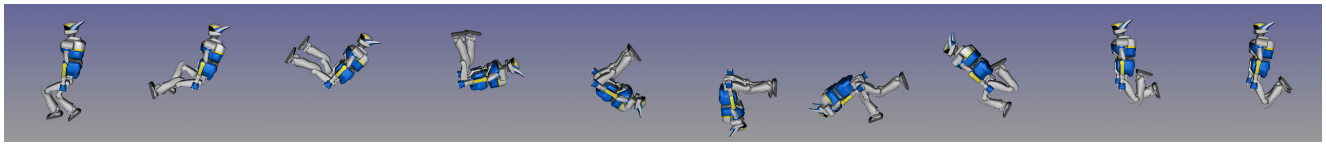

Fig. 6: Attitude adjustment maneuver conducted by the robot in gravity free space. DDP solver takes into account the non-holonomic angular momentum constraint and uses internal actuation to rotate 360° without the need for contact forces.

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 (1).

All the motions were computed offline. Contact sequence [25] and the centroidal trajectory [30] are pre-computed and provided to the solver for the large stride experiments. We used the standard controller OpenHRP [31] for tracking the motions on the real robot. The large strides produced by DDP are compared with those produced by an IK solver [11], 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. As the centroidal solver is able to provide feasible contact forces for individual contacts in the phase, we use a damped cholesky inverse to deal with the rank deficient $\mathbf{J}_c$ matrix. Then we use our proposed DDP to generate the full robot motion.

The cost function is composed of various quadratic residues (i.e. $\|\mathbf{r}_i(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda})\|^2_{\mathbf{Q}_i}$) 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. 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. 5.

B. Attitude regulation through joint motion

The angular momentum equation (1) shows that it is possible to regulate the robot attitude without the need of contact forces [15]. It can be seen that the gravity field does not affect this property. Thus, we analyze how our 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. 6 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. 4 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 behavior is repeated in the reverse.

VI. Conclusion and Perspective

Typically, reduced centroidal trajectory optimization does not take into account the AM produced by the limb motions.

Proper regulation of the AM exploits the counterbalancing effect of limb motion in order to reduce the contact forces and torque inputs. It also improves the stability during flight phases where the momentum control can only be made through joint motions. Excessive AM cannot be produced by a simple IK/ID solver. OC provides the required tools for solving it. Our proposed solver is an extension of our previous work [3].

In this paper, we have proposed a novel DDP formulation based on the augmented KKT dynamics (see (13)) 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 whole-body 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. 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.

While the solver is able to generate AM, and track the centroidal trajectories, the current approach still lacks an assurance that the additional AM generated by the DDP solver is accounted for in the centroidal optimization, and vice versa. Upcoming work from our team is focused on this guarantee. While this is proof-of-concept, an implementation based on Analytical Derivatives [32] is also our next goal.

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*, 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] P. Fernbach, S. Tonneau, and M. Taïx, "CROC: Convex Resolution Of Centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem," in *International Conference on Intelligent Robots and Systems*, 2018.

[6] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum control," in *IEEE International Conference on Humanoid Robots*, 2015.

[7] 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*, 2015.

[8] 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*, 2017.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, and F. Kanehiro, "Multi-contact vertical ladder climbing with an HRP-2 humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 561–580, mar 2016.

[14] 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.

[15] 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.

[16] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, 1983.

[17] J. Carpentier, A. Del 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.

[18] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, 2013.

[19] G. Schultz and K. Mombaur, "Modeling and Optimal Control of Human-Like Running," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 783–792, oct 2010.

[20] 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*, 2015.

[21] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE International Conference on Robotics and Automation*, 2014.

[22] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[23] M. Neunert, M. Stauble, M. Giftthaler, 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.

[24] J. Rajamäki, K. Naderi, V. Kyrki, and P. Hämäläinen, "Sampled differential dynamic programming," in *IEEE International Conference on Intelligent Robots and Systems*, 2016.

[25] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An Efficient Acyclic Contact Planner for Multiped Robots," *IEEE Transactions on Robotics (TRO)*, 2018.

[26] D. Q. Mayne, "Differential Dynamic ProgrammingA Unified Approach to the Optimization of Dynamic Systems," *Control and Dynamic Systems*, 1973.

[27] 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*, 2012.

[28] 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.

[29] F. Udwadia and R. Kalaba, "A new perspective on constrained motion," in *Mathematical and Physical Sciences*, 1992.

[30] J. Carpentier and N. Mansard, "Multi-contact locomotion of legged robots," *IEEE Transactions on Robotics (To Appear)*, 2018.

[31] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture humanoid robotics platform," *The International Journal of Robotics Research*, 2004.

[32] J. Carpentier and N. Mansard, "Analytical Derivatives of Rigid Body Dynamics Algorithms," in *Robotics: Science and Systems (RSS)*, 2018.