

Hierarchical Planning of Dynamic Movements without Scheduled Contact Sequences

Carlos Mastalli¹, Ioannis Havoutis², Michele Focchi¹, Darwin G. Caldwell¹, Claudio Semini¹

Abstract—Most animal and human locomotion behaviors for solving complex tasks involve dynamic motions and rich contact interaction. In fact, complex maneuvers need to consider dynamic movement and contact events at the same time. We present a hierarchical trajectory optimization approach for planning dynamic movements with unscheduled contact sequences. We compute whole-body motions that achieve goals that cannot be reached in a kinematic fashion. First, we find a feasible CoM motion according to the centroidal dynamics of the robot. Then, we refine the solution by applying the robot’s full-dynamics model, where the feasible CoM trajectory is used as a warm-start point. To accomplish the unscheduled contact behavior, we use complementarity constraints to describe the contact model, i.e. environment geometry and non-sliding active contacts. Both optimization phases are posed as Mathematical Program with Complementarity Constraints (MPCC). Experimental trials demonstrate the performance of our planning approach in a set of challenging tasks.

I. INTRODUCTION

Legged robots are able to traverse areas that are inaccessible to wheeled vehicles, such as complex and unstructured environments. Such environments are common to search and rescue scenarios, one promising application of legged systems. From the legged locomotion point of view, natural disaster scenarios require planning and execution of complex behaviors in environments with high uncertainty. Complex behaviors require whole-body movements and multiple contact interactions at the same time. Indeed, whenever an articulated body is in contact with the environment (e.g. legged robots) the set of contact forces and joint commands describe the evolution of the motion. Dynamic maneuvers such as jumping and rearing need to consider different sequences of contacts (mode-switching), which cannot be tackled with traditional predefined gaits [1], [2].

Locomotion in complex environments requires reasoning about terrain conditions, planning and execution of movements through a sequence of contacts, i.e. footholds and/or handholds. These can be posed as separate problems (decoupled approach), i.e. motion and contact planning, and control. Such approaches reduce the combinatorial search space at the expense of the richness of complex behaviors. In contrast, highly-dynamic movements need to consider contact forces and robot dynamics. For instance, contact forces play an important role for predicting the ballistic

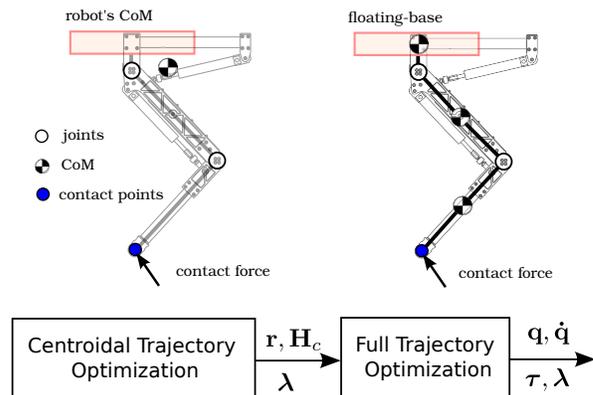


Fig. 1. The proposed hierarchical trajectory optimization reduces the complexity of the motion planning problem by considering two different optimization phases: centroidal and full trajectory optimization. First, the centroidal trajectory optimization phase produces a locally optimal CoM motion using the centroidal dynamics model [3], which does not consider joint dynamics (i.e. link’s CoM). Second, the full trajectory optimization phase refines the CoM trajectory by applying the robot’s full-dynamics and joint limits. Both optimization phases use complementarity constraints to model the contact interactions.

trajectory in a jumping task. A decoupled motion planner has to explore different plans in the space of feasible movements, which is often defined by physical (friction properties), stability (whole-body balance), dynamic (inertial properties) and task constraints (goal positions and orientations). On the other hand, coupled motion planners compute simultaneously contacts and body movements by posing the problem as a hybrid system or a mode-invariant optimization problem.

In this paper, we are concerned with finding feasible trajectories for complex tasks, i.e. tasks that require the exploration of different mode sequences through highly-dynamic movements. We choose a set of jumping tasks as examples, as these highlight the ability to explore the dynamical capabilities of the robot in order to reach goals that are unreachable in a kinematic manner.

The main contribution of this paper is a novel hierarchical trajectory optimization approach based on the principle of *divide and conquer* (Fig. 1). Our hierarchical trajectory optimization is capable of producing a wide range of complex behaviors without scheduling a contact sequence by reducing the search space to a fixed sequence of commands, which can be used to find continuous motion plans. The trajectory optimization approach is posed as a MPCC. First, we prune the search space by finding a feasible trajectory according to the robot’s centroidal dynamics. Second, we find a feasible trajectory in terms of the full dynamics and joint limits of

¹Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. email: {carlos.mastalli, michele.focchi, darwin.caldwell, claudio.semini}@iit.it.

²Robot Learning and Interaction Group, Idiap Research Institute, Martigny, Switzerland email: ioannis.havoutis@idiap.ch

This work was in part supported by the DexROV project through the EC Horizon 2020 programme (Grant #635491).

the robot.

The rest of the paper is structured as follows: after discussing previous research in the field of dynamic motion planning for legged systems (Section II) we describe the direct hierarchical trajectory optimization approach proposed for dynamic motion planning in Section III. In Section IV we evaluate the performance of our trajectory optimization approach in experimental trials with a robotic leg before Section V summarizes this work and presents ideas for future work.

II. RELATED WORK

State-of-the-art legged motion planning methods are often constrained to non-dynamic regimes with scheduled contact sequences [4][5][6][7] and decoupled models [8][9]. For example, most of the current locomotion approaches follow predefined contact sequences, specified by fixed patterns of locomotion. These assumptions reduce the capability of traversing complex environments with legged systems. We believe that natural locomotion in complex environments requires dynamic movements with unscheduled contact sequences. Nevertheless, these approaches are often posed as non-linear optimization problems in which kinematic and dynamic models of the robot, and task information are stated as hard [10][11][12] or soft constraints [13][14]. Currently, there are many approaches that reduce the complexity of the problem. Different dynamic formulations have been proposed such as: point-mass, cart-table [5][7], spring-loaded inverted pendulum model [15][16] and contact wrench sum [11][3]. However, most of these approaches use simplified models that do not capture the full-body dynamics of the robot, and reduce the richness of the solutions. On the other hand, using full-body dynamic models [10][17], one can produce richer movements but often such formulations require an excessively long amount of time to compute solutions, they are prone to reaching and becoming trapped in local minima and infeasible regions.

Defining a scheduled contact sequence considerably reduces the search space by constraining the plans to a fixed pattern of locomotion [8]. On the other hand, with an unscheduled contact sequence, we can increase the possibilities of locomotion [18][19], which could in turn be crucial for finding solutions in complex environments. From the optimization point of view, unscheduled contact sequence approaches include the contact forces as decision variables, allowing us to consider friction cone constraints. In the literature there exist two main approaches that incorporate contact forces in the optimization; the first one, smooths the contact forces but has the drawback of allowing the contact to be enabled when there is a distance w.r.t. the contact surface [13], the second one, uses complementarity constraints to model inelastic contacts [10][11]. Complementarity constraints describe inelastic contact forces without resulting in stiff differential equations that require an extremely small time step [20].

III. HIERARCHICAL DYNAMIC PLANNING

Our work was motivated by the observation that most animal and human locomotion behaviors involve dynamic

motions through contact interactions. For instance, kangaroos are *dynamic jumpers* that use hopping as the main locomotion strategy. Indeed, in kangaroo locomotion, highly-dynamic movements and contact forces play an important role for finding efficient locomotion trajectories.

Although such dynamic maneuvers are undoubtedly beneficial, planning and execution of these whole-body trajectories is challenging due to the loss of control authority during flight phases and undefined contact events. We tackle it by generating a whole-body trajectory toward a body goal state (desired body height) that ensures a dynamic motion plan through contact interactions. To accomplish this, we describe the contact model using complementarity constraints which defines our approach as a mode-invariant trajectory planner.

A. Generating Dynamic Motions

Consider a rigid body system with n degrees of freedom, of which n_b are floating-base degrees of freedom. The state of the robot is represented by its floating-base and actuated joint components, $\mathbf{q} = (\mathbf{q}_b, \mathbf{q}_q)$. Additionally, the robot has p end-effector or contact points.

The system's evolution depends on the internal joint torques, $\boldsymbol{\tau}_q$, and the contact force, $\boldsymbol{\lambda}_j$, applied at the j^{th} end-effector. This evolution is subject to robot and environmental constraints such as: joint limits and environment geometry. Exploring different mode switches (contact events) and dynamic movements could produce unsuccessful locally optimal solutions. We improve the solutions by applying a hierarchical trajectory optimization. In the first phase, we model the system's evolution with the centroidal dynamics, i.e. in the CoM space. Then, we impose joint dynamics and limits using a full-dynamic model. Figure 1 presents an overview of our hierarchical trajectory optimization approach.

1) *Centroidal-dynamic model:* The robot dynamics can be projected at the CoM, i.e. the *centroidal dynamics* of the robot. In a full floating-base system ($n_b = 6$ DoF), the centroidal-dynamic model describes the rate of change of linear and angular momentum of CoM with respect to the inertial frame of reference [3]. The rate of change of linear and angular momentum is determined by contact forces $\boldsymbol{\lambda}_j$, gravitational force $m\mathbf{g}$ and the motion of the robot's links

$$m\dot{\mathbf{r}} = \sum_{j=0}^p \boldsymbol{\lambda}_j + m\mathbf{g} \quad (1)$$

$$\dot{\mathbf{H}}_c(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{j=0}^p (\mathbf{x}_j - \mathbf{r}) \times \boldsymbol{\lambda}_j \quad (2)$$

where m is the total mass of the robot, $\mathbf{r} \in \mathbb{R}^3$ is the CoM position, $\boldsymbol{\lambda}_j \in \mathbb{R}^3$ is the contact force applied at the j^{th} end-effector, $\mathbf{H}_c \in \mathbb{R}^3$ is the centroidal angular momentum and $\mathbf{x}_j \in \mathbb{R}^3$ is the end-effector position. The centroidal angular momentum is computed through the computation of the Centroidal Momentum Matrix (CMM) as defined in [3].

2) *Full-dynamic model:* The full-dynamic model enables us to compute the joint efforts given a motion state $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ subject to contact forces $\boldsymbol{\lambda}_j$. We partition the dynamics equation of the robot into the unactuated floating-base DoFs \mathbf{q}_b

(n_b equations) and the active robot joints \mathbf{q}_q (n_q equations):

$$\underbrace{\mathbf{H}(\mathbf{q}) \begin{bmatrix} \ddot{\mathbf{q}}_b \\ \ddot{\mathbf{q}}_q \end{bmatrix} + \begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_q \end{bmatrix}}_{\mathbf{b}=ID(model, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})} (\mathbf{q}, \dot{\mathbf{q}}) - \sum_{j=0}^p \begin{bmatrix} \mathbf{J}_{b_j}^T \\ \mathbf{J}_{q_j}^T \end{bmatrix} \boldsymbol{\lambda}_j = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau}_q \end{bmatrix} \quad (3)$$

where $\mathbf{H} \in \mathbb{R}^{n \times n}$ is the floating-base inertial matrix, $\mathbf{c} = (\mathbf{c}_b, \mathbf{c}_q) \in \mathbb{R}^n$ is the force vector that accounts for Coriolis, centrifugal, and gravitational forces, $\boldsymbol{\lambda}_j \in \mathbb{R}^3$ are the ground contact forces at the j^{th} end-effector (i.e. point feet), and their corresponding Jacobian, $\mathbf{J}_j = [\mathbf{J}_{b_j} \quad \mathbf{J}_{q_j}] \in \mathbb{R}^{3p \times n}$ and $\boldsymbol{\tau}_q \in \mathbb{R}^{n_q}$ are the joint efforts that we wish to calculate.

The left-hand term $\mathbf{b} = (\mathbf{b}_b, \mathbf{b}_q)$ is computed efficiently using the Featherstone implementation of the Recursive Newton-Euler Algorithm (RNEA) [21].

B. Contact Model

In dynamic movements, contact forces play an important role, e.g. a jumping or hopping task. Traditional approaches compute trajectories given a predefined contact sequence. These approaches do not exploit the fact that an optimized mode switching could be required for the success of a certain task.

A contact event occurs when a signed distance to the surface is strictly zero, and additionally, there is a contact force acting along the surface normal. Moreover, it is expected a null normal contact force when the contact is inactive, i.e. a positive signed distance. In other words, normal contact forces and signed distances are orthogonal and positives functions (4). Additionally, we desire that active contacts do not slide. Such condition implicates an orthogonality between normal contact forces and tangential velocities (5). In the optimization literature, constraints with combinatorial nature, such as the set of contact model equations (4)(5), can be described as complementarity constraints

$$0 \leq \lambda_j^{\hat{n}} \perp \phi_j(\mathbf{q}) \geq 0 \quad (4)$$

$$\mathbf{0} \leq \lambda_j^{\hat{n}} \perp \dot{\mathbf{x}}_j^{\hat{t}}(\mathbf{q}, \dot{\mathbf{q}}) \geq \mathbf{0} \quad (5)$$

where $\lambda_j^{\hat{n}}$ is the contact force acting along the surface normal at the j^{th} end-effector (i.e. contact point), $\phi_j(\mathbf{q})$ is the signed distance between the j^{th} contact point \mathbf{x}_j and the surface S_i , and $\dot{\mathbf{x}}_j^{\hat{t}}(\mathbf{q}, \dot{\mathbf{q}})$ is the velocity of the contact point along the tangential surface. Contact-point positions and velocities are calculated efficiently using spatial algebra.

C. Trajectory Optimization

Planning problems without scheduled contact sequences are often hard to solve due to the contact forces producing discontinuities in the dynamics. Here, we tackle this issue by applying a hierarchical trajectory optimization scheme, which uses different dynamic models in a two-phase manner. Using a different (simpler) dynamic model in the first optimization phase, we impose a dynamic relaxation, that helps to explore different mode switches. Thus, we find a feasible CoM motion in terms of the robot's centroidal dynamics. Then, we refine it by applying the full-dynamic model in the second, more complex, trajectory optimization phase.

1) *Centroidal trajectory optimization:* The centroidal trajectory optimization step computes a feasible CoM trajectory through the mapping of contact forces inside the centroidal dynamics. The CMM maps the robot's generalized velocities to its spatial momentum (for more details see [3]). We sample the trajectory in N knot-points with a fixed-time duration h . In this optimization phase, the decision variables of the optimization problem are the robot position \mathbf{q} , the robot velocity $\dot{\mathbf{q}}$, the CoM position \mathbf{r} , the CoM velocity $\dot{\mathbf{r}}$, the contact forces $\boldsymbol{\lambda}$, and the end-effector (contact) positions \mathbf{x} . Our cost function evaluates the trajectory in terms of the desired high-level goal of the task $\mathbf{w}(\mathbf{q})$ as:

$$\min_{\substack{\mathbf{q}[k], \dot{\mathbf{q}}[k], \mathbf{r}[k], \dot{\mathbf{r}}[k], \\ \mathbf{H}_c[k], \dot{\mathbf{H}}_c[k], \boldsymbol{\lambda}[k], \mathbf{x}[k]}} h \sum_{k=1}^N (\|\mathbf{w}(\mathbf{q}[k]) - \mathbf{w}(\mathbf{q}^*[k])\|_{\mathbf{Q}_q}) \quad (6)$$

where $\mathbf{w}(\mathbf{q})$ constructs a task-specific value from relevant features of the task, and $\|\mathbf{w}(\mathbf{q}[k]) - \mathbf{w}(\mathbf{q}^*[k])\|_{\mathbf{Q}_q}$ computes its associated quadratic cost given a desired robot position \mathbf{q}^* . Note that $\|\mathbf{x}\|_{\mathbf{Q}}$ is an abbreviation for the quadratic cost $\mathbf{x}^T \mathbf{Q} \mathbf{x}$.

We transcribe the centroidal dynamics differential equations (2) to algebraic ones by applying an Euler-backward integration rule with a fixed-time step h

$$\mathbf{r}[k-1] - \mathbf{r}[k] + h\dot{\mathbf{r}}[k] = \mathbf{0} \quad (7)$$

$$\mathbf{H}_c[k-1] - \mathbf{H}_c[k] + h\dot{\mathbf{H}}_c[k] = \mathbf{0} \quad (8)$$

$$m(\dot{\mathbf{r}}[k] - \dot{\mathbf{r}}[k-1]) - h \left(\sum_{j=0}^p \boldsymbol{\lambda}_j[k] + m\mathbf{g} \right) = \mathbf{0} \quad (9)$$

$$\dot{\mathbf{H}}_c[k] - \sum_{j=0}^p (\mathbf{x}_j[k] - \mathbf{r}[k]) \times \boldsymbol{\lambda}_j[k] = \mathbf{0} \quad (10)$$

where the centroidal angular momentum is computed from the CMM, $\mathbf{A}(\mathbf{q})$, as is explained in [3], i.e. $\mathbf{H}_c[k] = \mathbf{A}(\mathbf{q}[k])\dot{\mathbf{x}}[k]$. Additionally, we impose contact position constraints in order to describe the contact interactions

$$\mathbf{x}_j[k] - \boldsymbol{\kappa}_j(\mathbf{q}[k]) = \mathbf{0} \quad (11)$$

where $\boldsymbol{\kappa}_j(\cdot)$ is the direct kinematic function which computes the position of the j^{th} end-effector. We also include joint position and velocity limits.

$$\mathbf{q}_q^l \leq \mathbf{q}_q \leq \mathbf{q}_q^u \quad (12)$$

$$\dot{\mathbf{q}}_q^l \leq \dot{\mathbf{q}}_q \leq \dot{\mathbf{q}}_q^u \quad (13)$$

To describe different possible mode switches, we add contact position and velocity constraints. These constraints are described as complementarity constraints as follows

$$\lambda_j^{\hat{n}}[k], \phi_j(\mathbf{q}[k]) \geq 0 \quad (14)$$

$$\lambda_j^{\hat{n}}[k] \phi_j(\mathbf{q}[k]) = 0 \quad (15)$$

$$\lambda_j^{\hat{n}}[k] (\mathbf{x}_j^{\hat{t}}[k] - \mathbf{x}_j^{\hat{t}}[k-1]) = \mathbf{0} \quad (16)$$

We approximate the contact velocity as contact displacement along the tangential surface. Note that a contact velocity constraint does not guarantee zero displacement between knots.

2) *Full trajectory optimization*: Once a feasible, and locally optimal, CoM trajectory is computed, we use this CoM trajectory as a warm-start point of the full trajectory optimization phase. We transcribe the full-dynamic model with the same time step value of the centroidal trajectory optimization phase. In this optimization phase, we formulate the problem with the following decision variables: the robot position \mathbf{q} , the robot velocity $\dot{\mathbf{q}}$, the joint efforts τ_q and the contact forces λ .

In this stage, the cost function also considers the joint effort energy of the movement τ_q as

$$\min_{\substack{\mathbf{q}[k], \dot{\mathbf{q}}[k], \\ \tau[k], \lambda[k]}} h \sum_{k=1}^N (\|\mathbf{w}(\mathbf{q}[k]) - \mathbf{w}(\mathbf{q}^*[k])\|_{\mathbf{Q}_q} + \|\tau_q[k]\|_{\mathbf{R}}) \quad (17)$$

We apply the same integration rule to the full-dynamic differential equation (3). Additionally, we add a selection matrix \mathbf{B} in order to impose a null wrench vector to the floating-base:

$$\mathbf{q}[k-1] - \mathbf{q}[k] + h\dot{\mathbf{q}}[k] = \mathbf{0} \quad (18)$$

$$\mathbf{H}[k] (\dot{\mathbf{q}}[k] - \dot{\mathbf{q}}[k-1]) + h \left(\mathbf{c}[k] - \sum_{j=0}^p \mathbf{J}_j[k]^T \lambda_j[k] \right) - \mathbf{B}\tau[k] = \mathbf{0} \quad (19)$$

where the contact forces are determined using the complementarity constraints (14)(15)(16).

In the full trajectory optimization phase, we impose position and velocity bounds (12)(13), and additionally joint efforts bounds

$$\tau_q^l \leq \tau_q \leq \tau_q^u \quad (20)$$

We derive a continuous motion plan, from the N optimized knot-points, using a polynomial interpolation. Both optimization phases model contact interactions using complementarity constraints. In general, optimization problems with complementarity constraints are difficult to solve because constraint qualifications are hard to satisfy. We solve the MPCC using interior point method as this is faster than a Sequential Quadratic Programming (SQP) algorithm when the number of complementarity constraints increases [22]. We use the IPOPT library [23]. We relax the orthogonality between the complementarities, for example $\lambda_j^{\hat{n}}[k] \phi_j(\mathbf{q}[k]) = 0$ is posed as $\lambda_j^{\hat{n}}[k] \phi_j(\mathbf{q}[k]) \leq 0$. For more information about different interior point methods see [22].

IV. EXPERIMENTAL RESULTS

This section describes the experiments conducted to validate the effectiveness and quantify the performance of the proposed hierarchical optimization approach.

A. Experimental Setup

We use the hydraulically-actuated robot leg, HyL, in our experiments. HyL weighs approximately 11 kg, is fully-torque controlled and equipped with precision joint encoders, and load cells. HyL is a 1D floating-base system with 2 actuated joints as is shown in Fig. 2. Controller computations

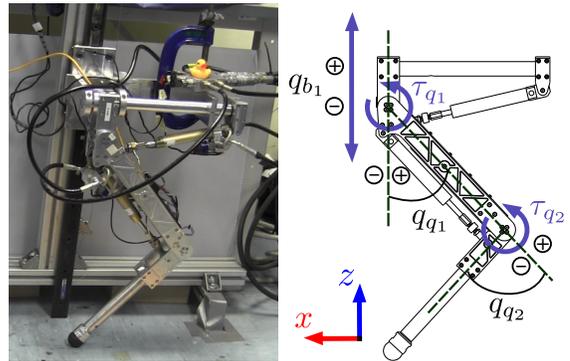


Fig. 2. HyL: one hydraulically actuated and fully torque controlled leg of the quadruped robot HyQ [24]. The HyL robot has a total number of 3 DOF: 1D floating-base system q_{b1} with 2 actuated joints (q_{q1}, q_{q2}).

are done on-board in an i7/2.8 GHz PC with a real-time kernel. Motion plans are computed off-line using a predefined terrain model.

For each experiment, we specify the goal state of the robot's trunk, and the desired final joint position as a terminal cost. The hierarchical trajectory optimizer finds a sequence of footsteps through dynamic movements without a predefined order, which the controller then executes dynamically. We use a PD controller, using the planned joint efforts as feedforward inputs. We validate the performance of our framework in 3 different examples as seen in Fig. 3, and compare against the full dynamic optimization (Table I) on the same benchmark examples. The first example consists of reaching a goal that is kinematically not feasible, called the *jumping task*. In the next examples, the *step-jumping tasks*, the robot has to reach and keep the desired trunk height, which is done through two different steps: a small step (10 cm) and a big step (15 cm). Additionally, the reader is strongly encouraged to view the accompanying video as it provides the most intuitive way to demonstrate the performance of our approach.

B. Results and Discussion

1) *Motion planning through dynamical system relaxation*: We focus on finding trajectories that are only feasible when dynamics and contact forces are considered. Our motion planning approach describes contact events in a MPCC problem. Since the problem is non-convex, our hierarchical optimization tends to guide the exploration away from infeasible regions through dynamical system relaxation, i.e. centroidal to full dynamics. This dynamical relaxation helps to reduce the computation time and cost value.

Our experiments suggest that dynamical system relaxation is key for finding successful motion plans. Table I shows the time and cost reduction of our approach compared with a single full trajectory optimization. We can see that the hierarchical optimization approach tends to have better performance in complex tasks. Nevertheless, in general, the central tendency (median) of the computation time reduction is decreased, while, on average, we improve the quality of the solution (cost reduction). We define a set of 8 different goal states (i.e. trunk height) for computing the time and

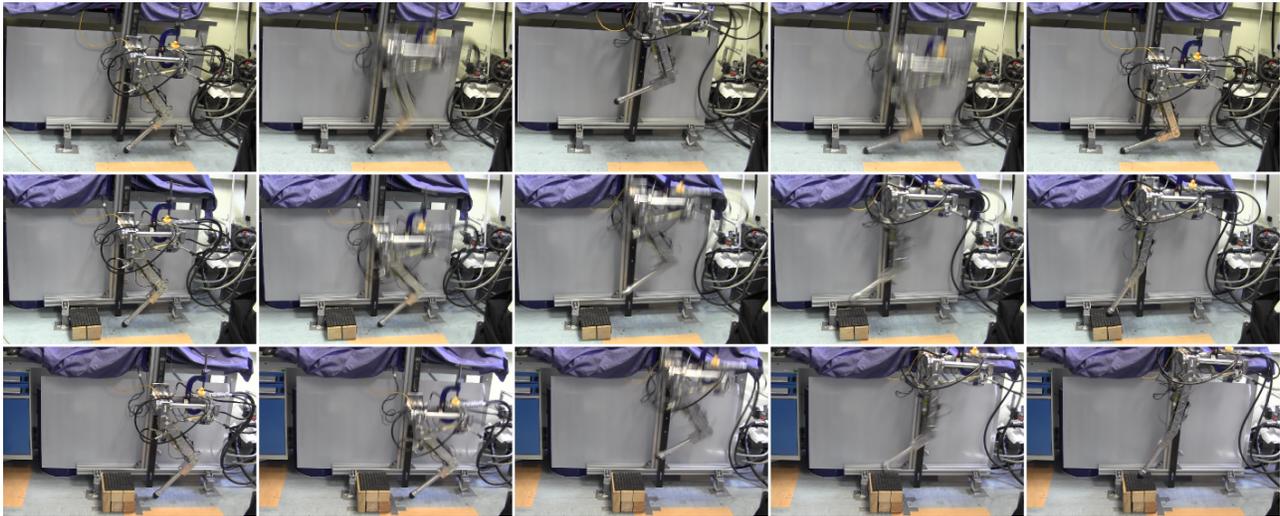


Fig. 3. Snapshots of three experimental trials used to evaluate the performance of our hierarchical trajectory optimization approach. From top to bottom: jumping task; small step jumping (10 cm of height); big step jumping (15 cm of height).

TABLE I

TIME AND COST REDUCTION OVER 8 TRIALS COMPARED TO A SINGLE FULL TRAJECTORY OPTIMIZATION.

Task	Time reduction [%]		Cost reduction [%]	
	Md.	Av.	Md.	Av.
Jumping	10.36	0.0	0.0	1.44
Step Jumping	48.29	30.46	0.0	12.91

cost reduction of our approach. For computing the time and cost reduction, we compare the time and cost of a single full trajectory optimization trial against our hierarchical optimization approach in the set of goals defined.

2) *Reaching goals that are kinematically not feasible:*

The jumping task demonstrates the ability of exploring the dynamical capabilities of the robot in order to reach goals that are not kinematically possible. In this particular case, we desire to reach with the trunk, a height of 0.85 cm w.r.t. the ground (or 0.27 cm w.r.t. the initial position), which is kinematically not feasible. Thus, our hierarchical motion planner explores different mode sequences in order to plan a dynamically feasible motion. In Fig. 4, the robot plans a countermovement jump (around 7 cm) without being predefined. In countermovement jumps, a preliminary downward movement is executed which increases the jump height because the robot is carried by its own inertia. Then, a fast movement of the foot is planned considering a desired task behavior, e.g. joint position in the apex point.

3) *Discovering of new contacts:*

For the success of some tasks, it is crucial to exploit the environmental conditions, e.g. reaching and keeping a desired trunk position that is kinematically not reachable. So, imagine that we want to keep a desired trunk position but due to gravitational forces this is not possible with just a vertical jump, for instance we need to climb onto an obstacle to accomplish this. With the hierarchical trajectory optimization, we can plan such

kind of maneuvers. In fact, Fig. 5 shows that our motion planner solves these tasks by defining a foothold on top of an available step. Note that a pre-defined footstep sequence is not required to arrive to such a solution.

V. CONCLUSION

In this paper we presented a hierarchical trajectory optimization approach for planning dynamic movements through unscheduled contact sequence. First, the hierarchical trajectory optimization finds a feasible CoM motion according to the centroidal dynamics of the robot. Then, a second phase of optimization considers the full-dynamics of the robot. In both phases we use complementarity constraints for contact modeling. We demonstrated that, with our approach, the robot can plan a wide range of movements that consider the full-dynamics and joint effort limits of the robot. We believe that these considerations are crucial for highly-dynamic locomotion tasks, i.e. *step-jumping* tasks that are cannot be accomplished in a kinematic fashion. We showed how the hierarchical trajectory optimization improves the solutions and significantly reduces the computation time, compared with the full dynamic optimization. Experimental trials with a robotic leg performing highly-dynamic and challenging tasks demonstrate the capability of our planning approach.

In general, trajectory optimization produces motion plans for a fixed sequence of points, N knot-points. Here, we arrive at a continuous motion plan by applying a polynomial interpolation. Nevertheless, polynomial interpolations cannot predict accurately changes in the contact forces, which generally happen in less than 10 ms. We are currently working on generating motion plans given a library of synthesized motions, improving the quality of the solutions, and permitting on-line computation and execution.

REFERENCES

[1] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, "A reactive controller framework for quadrupedal locomotion on challenging terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013,

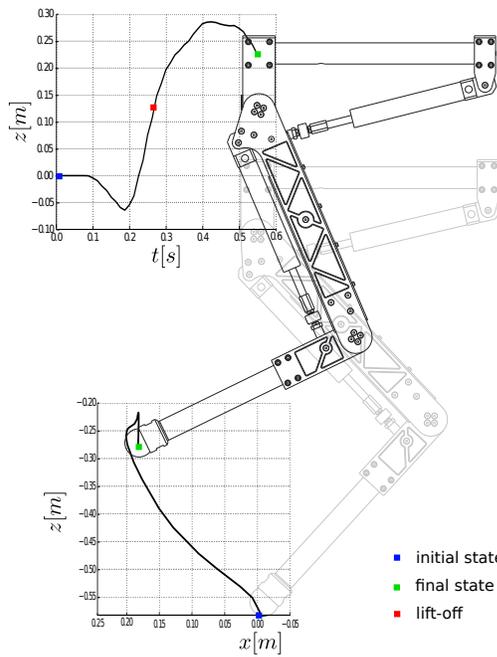


Fig. 4. Optimized CoM and foot trajectory for a jumping task that shows a dynamic movement through different phases: thrust and flight. We can see that the hierarchical optimization maximizes the jump energy by planning a countermovement jump, i.e. a preliminary downward movement.

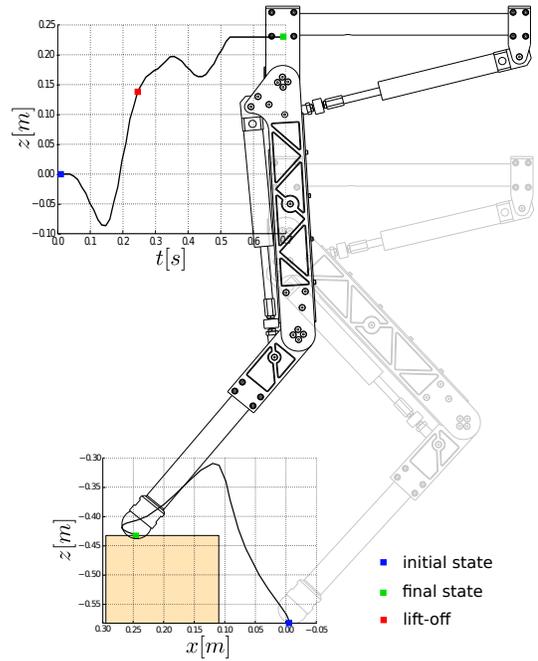


Fig. 5. Optimized CoM and contact sequence for reaching and keeping a desired trunk position (big step jumping), which is not reachable with just a vertical motion. The hierarchical trajectory optimization finds a transitory foothold in order to keep the desired trunk position. After landing in the planned foothold, the trunk moves up until the desired goal.

- [2] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, D. G. Caldwell, and C. Semini, "Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [3] D. E. Orin, A. Goswami, and S. H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.
- [4] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, "A control architecture for quadruped locomotion over rough terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 811–818.
- [5] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Fast, robust quadruped locomotion over challenging terrain," in *IEEE international conference on Robotics and Automation (ICRA)*, 2010.
- [6] I. Havoutis, J. Ortiz, S. Bazeille, V. Barasuol, C. Semini, and D. G. Caldwell, "Onboard Perception-Based Trotting and Crawling with the Hydraulic Quadruped Robot (HyQ)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.
- [7] A. 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)*, May 2015.
- [8] C. Mastalli, A. Winkler, I. Havoutis, D. G. Caldwell, and C. Semini, "On-line and On-board Planning and Perception for Quadrupedal Locomotion," in *IEEE International Conference on Technologies for Practical Robot Applications (TEPRA)*, May 2015.
- [9] C. M. Dellin and S. S. Srinivasa, "A framework for extreme locomotion planning," *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012.
- [10] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research (IJRR)*, Oct. 2013.
- [11] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with simple dynamics and full kinematics," in *IEEE/RAS International Conference on Humanoid Robots (IROS)*, 2014.
- [12] Y. Tassa and E. Todorov, "Stochastic Complementarity for Local Control of Discontinuous Dynamics," in *Robotics: Science and Systems*, 2010.
- [13] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–8, July 2012.
- [14] M. Gabbicini, A. Artoni, G. Pannocchia, and J. Gillis, "A Computational Framework for Environment-Aware Robotic Manipulation Planning," in *International Symposium on Robotics Research (ISRR)*, Sept. 2015.
- [15] I. Poulakakis and J. W. Grizzle, "The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009.
- [16] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An integrated system for real-time Model Predictive Control of humanoid robots," in *IEEE/RAS International Conference on Humanoid Robots (IROS)*, 2013.
- [17] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012.
- [18] K. Hauser, V. Ng-Thow-Hing, and H. Gonzalez-Baños, "Multi-modal motion planning for a humanoid robot manipulation task," *The International Journal of Robotics Research (IJRR)*, 2011.
- [19] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428–442, May 2013.
- [20] D. Stewart and J. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with Coulomb friction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.
- [21] R. Featherstone, *Rigid Body Dynamics Algorithms*. Boston, MA: Springer US, 2008.
- [22] A. U. Raghunathan and L. T. Biegler, "An Interior Point Method for Mathematical Programs with Complementarity Constraints (MPCCs)," *Journal on Optimization*, vol. 15, no. 3, pp. 720–750, 2005.
- [23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [24] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of HyQ – a Hydraulically and Electrically Actuated Quadruped Robot," *Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.