Hybrid direct collocation and control in the constraint-consistent subspace for dynamic legged robot locomotion

Diego Pardo, Michael Neunert, Alexander Winkler, Ruben Grandia, Jonas Buchli

Agile & Dexterous Robotics Lab, Swiss Federal Institute of Technology ETH, Zürich

{depardo,mneunert,winklera,rgrandia,buchlij}@ethz.ch

Abstract—In this paper, we present an algorithm for planning and control of legged robot locomotion. Given the desired contact sequence, this method generates gaits and dynamic motions for legged robots without resorting to simplified stability criteria. The method uses direct collocation for searching for solutions within the constraint-consistent subspace defined by the robot's contact configuration. For the differential equation constraints of the collocation algorithm, we use the so-called direct dynamics of a constrained multibody system. The dynamics of a legged robot is different for each contact configuration. Our method deals with such a hybrid nature, and it allows for velocity discontinuities when contacts are made. We introduce the *projected impact dynamics constraint* to enforce consistency during mode switching. We stabilize the plan using an inverse dynamics controller compatible with the optimal feed-forward control of the motion plan. As a whole, this approach reduces the complexity associated with specifying dynamic motions of a floating-base robot under the constant influence of contact forces. We apply this method on a hydraulically-actuated quadruped robot. We show two types of gaits (walking and trotting) as well as diverse jumping motions (forward, sideways, turning) on the real system. The results presented here are one of the few examples of an optimal control problem solved and satisfactorily transferred to a real torque-controlled legged robot.

I. INTRODUCTION

The field of planning and control for legged robots has progressed remarkably over the last decades. Methods to make robots walk [1], run [2] and jump [3] exist and they are well understood [4]. However, such methods are usually specifically designed for a given task and robot morphology. Generalized methods are based on simplified models [5], and advanced robots are not using all their potential to move efficiently, but their motion is driven by the premise of 'not to fall'. The current challenge is to improve its performance (e.g., speed, versatility, robustness) and to find generic frameworks able to generate diverse types of dynamic motions from simple specifications.

With this motivation, many approaches based on Trajectory Optimization (TO) have been proposed [6], [7], [8], [9], [10], [11], [12] demonstrating that optimization is a valid framework to derive motion plans that are not driven by quasi-static stability criteria. However, legged robots are floating-base, high dimensional systems, underactuated and permanently influenced by contact forces. These challenging aspects hinder the effectiveness and generalization capabilities of many planning approaches.

Contact forces are essential for the motion of a legged robot. Stabilizing contact forces in the physical system is very difficult as sensing is problematic and actuation systems are arguably not ideal. Motion plans must be robust towards contact disturbances and uncertainty of the contact model. Realistic plans and feedback controllers should cope with these constraints to produce feasible accelerations.

In this paper we propose a generalized framework based on direct collocation to generate dynamic motions on legged robots. We model the robot as a constrained multibody system using the direct dynamics derived by Aghili [13]. Such a model is based on the projection of the dynamics into the null space of the constraints. This projection restricts the search domain to the subspace of valid accelerations. Given a sequence of contact configurations, we formulate a multi-phase optimization problem in which the dynamic of each phase differs depending on the constraints. Modeling the hybrid nature of the robot enables plans with impacts. The projection eliminates the contact forces from the equation of motion, and makes the implicit assumption of a hard contact model. Then, our method optimizes states and controls neither using the contact forces as decision variables nor a parametric soft contact model. The method provides state and control trajectories that satisfy the contact constraints and are transferable to the physical system. Finally, we implement a stabilizing controller based on the ideas of Mistry et al. [14], compatible with the feed-forward control provided by the motion planner.

We demonstrate our approach on a torque-controlled quadruped robot and present two types of gaits (walking and trotting). We also generate different types of jumping motions (forward, sideways, turning) on the real system, showing that this method can generate aperiodic motions, and that it exploits the physical capabilities of the robot. In the results section we show evidence that our approach provides motion plans consistent with the contact constraints. The software package that implements the method presented in this paper is available at http://www.diegopardo.net/rss2017.html.

In the following section we formally describe the TO problem and review related work. In Section III we describe Direct Collocation and the dynamics of a floating-base constrained multibody system as a new framework for optimizing in the constraint-consistent subspace. We introduce the hybrid direct collocation method for dynamic locomotion of legged robots

in Section IV. It includes the derivation of the new *projected impact dynamics constraint* as a mechanism for enabling swing to stance transitions. Experimental results are described in Section VI, followed by the discussion and conclusions in Section VII and VIII.

II. BACKGROUND AND RELATED WORK

A. Optimal control of legged robots

In general, the dynamics of a robot can be modeled by a set of nonlinear differential equations,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{1}$$

where $\mathbf{x} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T \in \mathbb{R}^{n_x}$ represents the state and $\mathbf{u} \in \mathbb{R}^{n_u}$ the vector of inputs.

A trajectory optimization problem is formulated in [15] as

$$
\begin{aligned}
\min_{\mathbf{u}(t), \mathbf{x}(t)} \quad & J = \Psi\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\
\text{s.t.,} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \\
& \mathbf{g}_{0,l} \leq \mathbf{g}_0(\mathbf{x}(t_0), \mathbf{u}(t_0), t_0) \leq \mathbf{g}_{0,u} \\
& \mathbf{g}_{f,l} \leq \mathbf{g}_f(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f) \leq \mathbf{g}_{f,u} \\
& \phi_l \leq \phi(\mathbf{x}(t), \mathbf{u}(t), t) \leq \phi_u \\
& \mathbf{x}_{min} \leq \mathbf{x}(t) \leq \mathbf{x}_{max} \quad \mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}.
\end{aligned}
\tag{2}
$$

It consists of finding a finite-time input trajectory $\mathbf{u}(t), \forall t \in [0, t_f]$, such that a given criteria J is minimized. The optimization is subject to the dynamics of the system, as well as to boundary conditions $\mathbf{g}(\cdot)$, path constraints $\phi(\cdot)$, and bounds on the states and controls.

This is a complex problem for the case of legged robots. These systems are underactuated, meaning that in certain configurations, arbitrary accelerations are not achievable [16]. During locomotion these systems establish and break contacts with the environment, generating reaction forces that act on the system. Depending on the contact configuration, the equation of motion (1) changes and only accelerations consistent with the contact constraints are feasible. Moreover, when contacts are made, impacts may occur and generate impulsive forces and velocity discontinuities.

B. Related work

Numerical methods for solving (2) for the case of legged robots have been vastly explored. In general, methods differ on three fundamental aspects: i) Type of model (continuous, hybrid, whole-body, centroidal), ii) Representation of the contact forces (decision variables, soft or hard contact model) and iii) Type of numerical optimization algorithm (direct transcription/collocation/multiple shooting, Differential Dynamic Programming). Interestingly, these three dimensions cannot be chosen independently. Folding (2) into a numerical optimization framework requires coordination among the aspects mentioned above.

Schultz and Mombaur [6] use direct multiple shooting to generate the running gait of a multi-link 3D anthropomorphic model. They use an explicit hybrid model for each phase

of the motion and a soft contact model. Other multiple-shooting approaches [17], [18] have been employed for simplified locomotion tasks. Early work on the use of a Linear Complementary Problem for solving trajectories of discrete contact systems was formulated in [19]. Posa et al. [7] apply direct transcription and an implicit hard contact model to a locomotion problem using the contact forces as decision variables. Similarly, Dai et al. [9] used direct transcription over a reduced model of the dynamics together with kinematic constraints. Xi and Remy [20] applied direct collocation in a planar model using soft contact models. A series of approaches based on Differential Dynamic Programming (DDP) have presented alternatives to direct methods, where computational efficiency is the main motivation [10], [21].

On the other hand, transfer to real robots and thus ultimately proving usefulness of the methods is still a major challenge. Optimal trajectories that depend on specific contact forces are difficult to stabilize. At the same time, sensitivity to a parametric soft contact model is not desired, as a precise interaction between the robot and the ground cannot be expected. Recently, Neunert et al. [11] presented a very efficient implementation of a Sequential Linear Quadratic (SLQ) algorithm for online motion planning on a real robot. However, an elaborated contact model is critical for its performance. DDP approaches are not favorable for handling constraints, therefore SLQ requires non-trivial cost functions to describe the desired task. Hereid et al. [12] implemented a direct collocation framework for optimizing a dynamic walking of a human-size biped. However, [12] focuses on the convergence of a periodic cycle through the implementation of Hybrid Zero Dynamics virtual constraints, and only a periodic motion is demonstrated. Importantly, in our framework the contact constraints are implicitly enforced along the continuous trajectory and not only at specific points. As a consequence, the guard conditions and reset maps proposed in this paper are significantly simpler.

C. Contribution

In practice, legged robots are dynamical systems with kinematic constraints. In this paper we provide a method to solve the problem in (2) recognizing this condition. With the same purpose, Posa et al. [8] recently presented *DIRCON*, a modified direct collocation algorithm for optimizing in the manifold of feasible states. Instead of modifying the algorithm, we use the constraint-consistent direct dynamics of a legged robot using the projection method derived by Aghili [13]. This model incorporates the contact constraints into the equation of motion. We show that applying direct collocation to this model provides motion plans that are consistent with the kinematic constraints along the trajectory and not only at the discretization points of the collocation algorithm. Our method achieves this without adding artificial decision variables as in [8]. With a similar motivation, early work on optimal biped walking using minimal coordinates models was explored by Hardt et al. [22]. However, this modeling approach is only valid for simple cases.

Our approach also presents a new alternative for handling

contact forces in a TO problem: It does not depend on soft contact models, neither does it use the contact forces as decision variables. The effects of the contact forces are implicit in the states, controls and contact configuration at any given time. Regarding computation times, the current implementation of our method is not significantly different from other direct methods. A rigorous comparison of the NLP performance is outside of the scope of this paper. The main advantage of using the projected dynamics is that the resulting plans are constraint-consistent with the contact constraints, and therefore easy to execute on the real robot.

Posa et al. [8] address the problem of properly stabilizing a legged robot as a constrained system. They propose to use the optimal trajectories from the planner on a low-level Quadratic Programming feedback controller using the Linear Quadratic Regulator solutions as cost functions. Instead, here we tackle the nonlinearity of the robot with an implementation of the inverse dynamics methods proposed by Mistry et al. [14].

We use constraints on the final configuration of the robot to specify the desired motion, avoiding the use of elaborate cost functions as in DDP approaches. As a whole, we present a new form of hybrid direct collocation for legged robots that handles impacts and that produces accurate solutions that can be stabilized on a real robot.

III. DIRCOL IN THE CONSTRAINED SUBSPACE

A. Method Overview

We use the projected forward dynamics [13] within a direct collocation [23] framework. We assume that a motion can be specified by a sequence of contact configurations. Using the projection, each configuration corresponds to a different equation of motion, resulting in a multi-phase optimization setting.

B. Direct collocation

In direct methods, the continuous formulation in (2) is translated into a nonlinear programming problem (NLP) that can be solved using nonlinear optimization solvers.

The decision variables of the NLP are the discrete values of the state and control trajectories sampled at certain points or nodes. We include the time step between nodes ΔT in the decision variables, i.e., $\mathbf{y} = \{\mathbf{x}_l, \mathbf{u}_l, \Delta T_l\}$ for $l = 1, ..., N$.

The resulting NLP is formulated as follows,

$$\begin{aligned} \min_{\mathbf{y}} \quad & f_0(\mathbf{y}) \\ \text{s.t.,} \quad & \zeta(\mathbf{y}) = \mathbf{0} \\ & \mathbf{b}_{min} \leq \mathbf{b}(y_1, y_N) \leq \mathbf{b}_{max} \\ & \mathbf{c}_{min} \leq \mathbf{c}(\mathbf{y}) \leq \mathbf{c}_{max} \\ & \mathbf{y}_{min} \leq \mathbf{y} \leq \mathbf{y}_{max}. \end{aligned} \quad (3)$$

It contains a scalar and derivable objective function $f_0(\mathbf{y})$, a set of boundary $\mathbf{b}(\cdot)$ and path constraints $\mathbf{c}(\cdot)$, and bounds on the decision variables. The set of differential equations in (2) representing the dynamics is transcribed using a vector of dynamic constraints or *defects*, $\zeta(\cdot) \in \mathbb{R}^{(N-1)n_x}$.

In direct collocation, the state trajectory at each interval is approximated using a cubic Hermite spline, $x_l^s(t)$. The slope of

the spline at the middle of the time interval, $t_c = t_l + 0.5\Delta T_l$, must be equal to the dynamics evaluated at that point,

$$\zeta_l = \dot{x}_l^s(t_c) - f(\mathbf{x}_c, \mathbf{u}_c) = 0. \quad (4)$$

This constraint implicitly verifies that the first derivatives of the spline at the nodes equals the dynamics.

C. Constraint consistent accelerations of a legged robot

We assume that a legged robot obeys the rigid body dynamics. Then, the equations of motion are given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T \tau + \mathbf{F}_c, \quad (5)$$

subject to k kinematic constraints

$$\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}. \quad (6)$$

The vector of generalized coordinates, $\mathbf{q} = \begin{bmatrix} \mathbf{q_b}^T & \mathbf{q_r}^T \end{bmatrix}^T$, includes the body pose, $\mathbf{q_b} \in SE(3)$, and the joint angles $\mathbf{q_r} \in \mathbb{R}^n$. $\mathbf{M} \in \mathbb{R}^{(n+6)\times(n+6)}$ represents the inertia matrix, $\mathbf{h} \in \mathbb{R}^{n+6}$ is a generalized force vector, gathering gravitational, Coriolis and centrifugal effects. $\tau \in \mathbb{R}^n$ is the vector of joint torques and $\mathbf{S} = \begin{bmatrix} \mathbf{0}_{n\times6} & \mathbf{I}_{n\times n} \end{bmatrix} \in \mathbb{R}^{n\times(n+6)}$ is the joint selection matrix that reflects the underactuation. $\mathbf{F}_c \in \mathbb{R}^{n+6}$, represents the generalized constraint forces acting on the robot's DOF.

The set of constraints, $\mathbf{\Phi}(\mathbf{q}) \in \mathbb{R}^k$, represents certain points of the robot, $\mathbf{p}_c \in \mathbb{R}^3$, that are in contact with the environment and therefore have a velocity equal to zero,

$$\dot{\mathbf{p}}_c = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{0}, \quad (7)$$

where $\mathbf{J}_c = \partial\mathbf{\Phi}/\partial\mathbf{q} \in \mathbb{R}^{k\times(n+6)}$ is the Jacobian of the constraints. Differentiating (7), the constraints can be expressed in acceleration form,

$$\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = \mathbf{0}. \quad (8)$$

Accelerations are constraint-consistent if and only if they satisfy (8).

$\mathbf{F_c}$ is related to the forces at the contact points $\lambda \in \mathbb{R}^k$,

$$\mathbf{F}_c = \mathbf{J}_c^T \lambda. \quad (9)$$

Notice that there is no assumption about the rank of $\mathbf{J}_c$, and it may contain redundancies.

Aghili [13] proposed to use a linear operator to project the equation of motion (5) onto the tangent space with respect to the constrained manifold. Given a projector operator, $\mathbf{P}(\mathbf{q}) \in \mathbb{R}^{(n+6)\times(n+6)}$, such that its range space equals the null space of the Jacobian of the constraints, i.e., $\mathcal{R}(\mathbf{P}) = \mathcal{N}(\mathbf{J}_c)$, the so called projected inverse dynamics of a constrained multibody system is given by,

$$\mathbf{P}\mathbf{M}\ddot{\mathbf{q}} = \mathbf{P}(\mathbf{S}^T \tau - \mathbf{h}). \quad (10)$$

The generalized constraint forces are annihilated by the projector operator i.e., $\mathbf{P}\mathbf{F}_c = \mathbf{0}$.

Accelerations cannot be uniquely determined from (10) as $\mathbf{P}\mathbf{M}$ might be singular. Aghili [13] observed that the null space orthogonal component of the acceleration, $\ddot{\mathbf{q}}_\perp \in \mathcal{N}(\mathbf{J_c})^\perp$, is

not necessarily zero, and that it can be used to complement the set of equations in (10). We show the details of this derivation as it supports the contributions presented in the next section.

According to (7), the null space orthogonal component of the velocity $\dot{\mathbf{q}}_\perp \in \mathcal{N}(\mathbf{J_c})^\perp$ must be equal to zero,

$$\dot{\mathbf{q}}_\perp = (\mathbf{I} - \mathbf{P})\dot{\mathbf{q}} = \mathbf{0}. \tag{11}$$

Differentiating with respect to time,

$$\ddot{\mathbf{q}}_\perp = (\mathbf{I} - \mathbf{P})\ddot{\mathbf{q}} = \mathbf{C}\dot{\mathbf{q}}, \tag{12}$$

where $\mathbf{C} = d\mathbf{P}/dt$. These accelerations are produced exclusively by the constraints and not by dynamics. Aghili [13] derived a complete equation of motion by combining the perpendicular equations (10) and (12).

$$\ddot{\mathbf{q}} = \mathbf{M}_c^{-1} \left(\mathbf{P}\mathbf{S}^T \tau - \mathbf{P}\mathbf{h} + \mathbf{C}_c\dot{\mathbf{q}} \right), \tag{13}$$

where $\mathbf{C}_c = \mathbf{M}\mathbf{C}$. The so-called *constraint inertia matrix* $\mathbf{M}_c \in \mathbb{R}^{(n+6)\times(n+6)}$, is defined as,

$$\begin{aligned} \mathbf{M}_c &= \mathbf{M} + \tilde{\mathbf{M}} \\ \tilde{\mathbf{M}} &= \mathbf{P}\mathbf{M} - (\mathbf{P}\mathbf{M})^T. \end{aligned} \tag{14}$$

D. Body pose representation and derivatives

The dynamics in (1) also includes the derivatives of the positions,

$$\dot{\mathbf{q}} = \left[\frac{d\mathbf{q_b}}{dt}^T \quad \dot{\mathbf{q}}_\mathbf{r}^T \right]^T. \tag{15}$$

Following the conventions in [24], we represent the pose of the base $\mathbf{q_b} = [\psi_\mathbf{b}^T \quad \mathbf{p_b}^T]^T$, as the position $\mathbf{p_b} \in \mathbb{R}^3$ and orientation $\psi_\mathbf{b} \in SO(3)$ of a base frame B expressed in a fixed inertial frame O. We decide to locate the inertial frame at the initial configuration of the base for convenience. We represent the base orientation using Euler angles (roll, pitch and yaw). On the other hand, the base twist $\dot{\mathbf{q}}_\mathbf{b} = \begin{bmatrix} \mathbf{w_b}^T & \dot{\mathbf{p}}_\mathbf{b}^T \end{bmatrix}^T$, is expressed in the base frame. Therefore, the twist and the rate of change of the base pose, $\frac{d\mathbf{q_b}}{dt}$, are related by the rotation matrice $^O\mathbf{E}(\cdot)_B \in SO(3)$, and the mapping $\mathbf{\Gamma}(\cdot)$,

$$\frac{d\mathbf{q_b}}{dt} = \begin{bmatrix} \mathbf{\Gamma}(\psi_\mathbf{b}) & \mathbf{0} \\ \mathbf{0} & ^O\mathbf{E}_B(\psi_\mathbf{b}) \end{bmatrix} \dot{\mathbf{q}}_\mathbf{b}. \tag{16}$$

We propose to use (13), (15) and (16) as representation of the dynamics of a legged robot, together with the direct collocation framework in (3) and (4).

IV. HYBRID DIRECT COLLOCATION FOR LEGGED ROBOTS

Legged robots are hybrid systems, and the modes are defined by the set of active contacts at any given time. Notice that the structure of $\mathbf{J}_c$ changes depending on the number of points in contact with the environment. This is reflected in (13) as the rank of $\mathbf{P}$ also changes. However, the number of equations describing the dynamics remains the same. Given a contact sequence, i.e., a sequence of versions of (13), we formulate a hybrid collocation problem over the state and control trajectories.
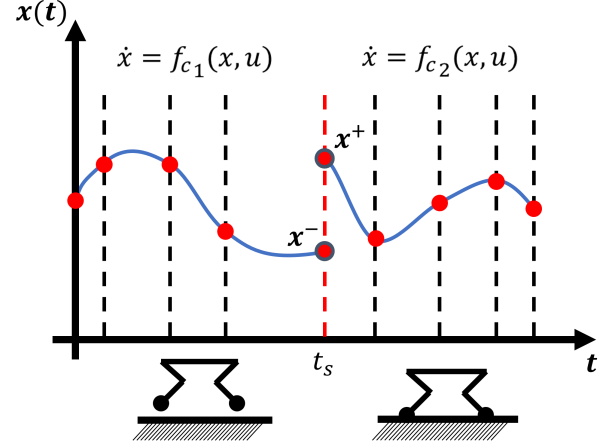

Fig. 1: Multi-phase direct collocation. Two contact configurations are shown as example. The dynamic equation used for each phase is given by the corresponding contact configuration. Each phase has one node at the boundary, allowing for discontinuities. Times between nodes are not even. Time of switching (t_s) is not prespecified.

A. Hybrid Collocation

Applying (2) to a hybrid system results in a sequence of TO problems or phases. Phases have different sets of equations of motion and they are connected with constraints. In a hybrid collocation approach boundary nodes are added. For a problem with s phases we add $N_b = s - 1$ boundary nodes. They represent the state (and controls) before, $\mathbf{x}(t_l^-) = (\mathbf{q}^-, \dot{\mathbf{q}}^-)$ and after, $\mathbf{x}(t_l^+) = (\mathbf{q}^+, \dot{\mathbf{q}}^+)$ a phase switch. Using this framework, discontinuities on the velocities are allowed. Fig. 1 illustrates the multi-phase direct collocation framework used in this research.

B. Guards and inter-phase constraints

Guards are functions determining a switching between modes. Such functions are directly transcribed to the NLP as guard constraints evaluated at the boundary nodes,

$$\mathbf{g}(\mathbf{x}^-) \geq \mathbf{0}. \tag{17}$$

Therefore phase switches occur given an event and not at fixed times. For example, the event verifying the end of a swing leg phase and the beginning of its support phase is given by the position of the feet p_c^z with respect to the ground. The guard validating that the feet is in contact with the ground at the switching point is given by,

$$p_c^z = f_{kin}(\mathbf{q}^-) \leq z_{ground} + \epsilon. \tag{18}$$

We use $\epsilon = 5 \times 10^{-4}$m to represent the tolerance at which we assume a contact has been made.

The discrete transition between modes is given by edge functions. These functions are transcribed as *inter-phase constraints*, and it might depend on the two boundary points of the phase switching. For instance, the configuration of the robot is invariant throughout phase switches, and a constraint enforcing this condition needs to be used in all phase transitions

($\mathbf{q}^+ - \mathbf{q}^- = \mathbf{0}$). Task conditions can be included as inter-phase constrains. For example, feet in contact can be constrained to have zero (or very small) vertical contact force at the left boundary point when switching from stance to swing phase.

C. Projected impact dynamics constraint

When contacts are made at nonzero velocity an impact occurs and generates an impulse and discontinuities in the velocities [25]. The impulse $\mathbf{\Lambda}$ is related to the change of momentum $\mathbf{\Delta h}$ during the impact,

$$\mathbf{\Delta h} = \mathbf{M}\left(\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-\right) = \mathbf{J}_c^T \mathbf{\Lambda}. \qquad (19)$$

Together with (7), a complete set of equations provides the transition function for the impact event,

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_c^T \\ \mathbf{J}_c & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}^+ \\ \mathbf{\Lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\dot{\mathbf{q}}^- \\ \mathbf{0} \end{bmatrix}. \qquad (20)$$

We project (19) using the projector operator of the corresponding contact mode,

$$\mathbf{P}\mathbf{M}\left(\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-\right) = \mathbf{0}. \qquad (21)$$

According to (21), there is no change of momentum in the null space of the constraints, or similarly, all the change of momentum has to be perpendicular to the null space of the constraints.

However, and as in (10), this system of equations does not provide a unique solution. We then add the condition that the velocities after the impact should be in the null space of the constraint or, equivalently, that the null space perpendicular component should be zero. Multiplying (11) by $\mathbf{M}$ and adding it to (21),

$$\left(\mathbf{M} + \mathbf{P}\mathbf{M} - (\mathbf{P}\mathbf{M})^T\right)\dot{\mathbf{q}}^+ - \mathbf{P}\mathbf{M}\dot{\mathbf{q}}^- = \mathbf{0},$$

i.e.,

$$\mathbf{M}_c\dot{\mathbf{q}}^+ - \mathbf{P}\mathbf{M}\dot{\mathbf{q}}^- = \mathbf{0}, \qquad (22)$$

we obtain the *Projected impact dynamics constraint*. This constraint uniquely relates the velocity of the system before and after the impact without explicitly reasoning about the impulse.

D. NLP Constraints

Here we provide details of additional constraints required by our algorithm to generate feasible motion plans.

1) Time step boundaries: The time step $\Delta T_{min} < \Delta T_l < \Delta T_{max}$ is part of the decision variables, and the algorithm can distribute the number of nodes along the trajectory. For the experiments reported in this paper we use $\Delta T_{min} = 0.005$s, and $\Delta T_{max} = 0.1$s.

2) Trajectory time constraint: We implement a mechanism for bounding the total time of the motion,

$$T_{min} < \sum_{l=1}^{N-1} \Delta T_l < T_{max}. \qquad (23)$$

We noticed that the solver uses less iterations when allowing $T_{min} = 0.01s$, and $T_{max} = \infty$.

3) Friction cone constraint: All phases include a friction cone constraint to avoid unfeasible contact forces. Contact forces are uniquely defined as a nonlinear function of the state and control, as derived by Righetti et al. [26],

$$\lambda_l = \mathbf{R}^{-1}\mathbf{S}_c\mathbf{Q}^T(\mathbf{M} \cdot f(t_l) + \mathbf{h} - \mathbf{S}^T\tau_l) \qquad (24)$$

where $\mathbf{Q} \in \mathbb{R}^{(n+6) \times (n+6)}$ and $\mathbf{R} \in \mathbb{R}^{k \times k}$ derive from the QR decomposition of the transpose of the constraint Jacobian $\mathbf{J}_c^T = \mathbf{Q}[\mathbf{R}^T \ \mathbf{0}]^T$, and $\mathbf{S}_c = \begin{bmatrix} \mathbf{I}_{k \times k} & \mathbf{0}_{k \times (n+6-k)} \end{bmatrix}$ is a selection matrix.

$$-\infty \leq ||\lambda_l^{x,y}||_2 - \mu \cdot \lambda_l^z \leq 0 \qquad (25)$$

where μ models the friction coefficient. For the experiments presented in this paper we typically use $\mu = 0.4$.

Notice that the friction cone constraint is not contradictory with the use of the projected dynamics, which does not depend on the contact forces.

4) Surface limits: All phases must include a constraint preventing the robot from penetrating the contact surface.

$$p_c^z \geq z_{ground} - \varepsilon. \qquad (26)$$

We use $\varepsilon = 1 \times 10^{-4}$m to represent a small penetration/deformation tolerance.

5) Final phase constraint: We add a state constraint to the final phase of the motion. This allows us to provide motion specifications to the algorithm. For example, we use a constraint on the x-component of the body position to specify the desired forward motion at the final phase,

$$q_b^x \geq x_{min}.$$

6) State and control bounds: Bounds on the states and controls are implemented for safety and to limit the search space of the algorithm. Moreover, torque limits may depend on the configuration of the robot, i.e., $\tau_{max} = \varphi(\mathbf{q})$. In all the experiments shown in this paper we use $-120\mathrm{N\,m} \leq \tau(t) \leq 120\mathrm{N\,m}$.

V. WHOLE-BODY FEEDBACK LINEARIZATION

A stabilizing controller is required to execute the plans. We use a whole-body feedback linearization method that provides asymptotic stability on the position error.

A. Motivation

A series of contributions on inverse dynamics for controlling legged robots without depending on the contact forces was presented in [14, 26]. These methods evolved providing a framework for the optimal distribution of contact forces [27], e.g., obtaining minimum tangential components for avoiding slippage.

Although these approaches are valid to stabilize the trajectories generated by our algorithm, we choose to use the optimal feed-forward controls $\mathbf{u}^*$, that already satisfy the friction constraints.
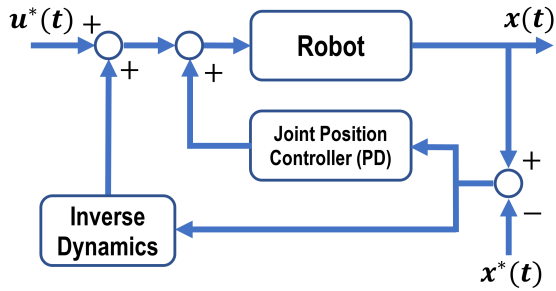
Fig. 2: Control framework diagram. The feedback linearization controller in (28) works at 250Hz. The default joint position controller of the robot works at 1.0kHz, and it is active during the execution of the tasks.

B. Whole-body feedback linearization of legged robots

It can be assumed that $\tau = \mathbf{u}^*$ generates the optimal accelerations $\ddot{\mathbf{q}}^*$. i.e.,

$$\mathbf{PS}^T\mathbf{u}^* = \mathbf{PM}\ddot{\mathbf{q}}^* + \mathbf{Ph}. \quad (27)$$

Our goal is to work out an expression for the feedback term $\phi(\mathbf{x})$ to stabilize the system together with the feed-forward controls, i.e.,

$$\tau(\mathbf{x}) = \mathbf{u}^* + \phi(\mathbf{x}). \quad (28)$$

Substituting (28) and (27) in (10)

$$\mathbf{PM}\ddot{\mathbf{q}} = \mathbf{PM}\ddot{\mathbf{q}}^* + \mathbf{PS}^T\phi(\mathbf{x}), \quad (29)$$

the projected inverse dynamics derived in [14] provides the correction term,

$$\phi(\mathbf{x}) = \left[\mathbf{PS}^T\right]^\dagger \mathbf{PM} \cdot \xi_{\mathbf{fb}}, \quad (30)$$

where $[\cdot]^\dagger$ is the right pseudo-inverse operator. We use $\xi_{\mathbf{fb}} = (\mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e})$ as a feedback term based on the whole-body position error, $\mathbf{e} \in \mathbb{R}^{n+6}$, such that, from (29),

$$\mathbf{PM}\left(\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}\right) = 0. \quad (31)$$

Given that the body positions and velocities are expressed in different frames, it is important to transform the error terms before feeding them back into the controller, such that the velocity error is the derivative of the position error. We use the quaternion stabilization convention proposed in [28].

VI. RESULTS

We use the method to generate gaits (walking and trotting) and different types of jumping motions for a point feet quadruped robot. This robot weights 80kg, and each leg has 3 actuated joints, i.e., $n = 12, n_x = 36, n_u = 12$. We use a state estimator for sensing the body positions and velocities.

Results are obtained using the control diagram shown in Fig. 2. The feedback linearization controller in (28) works at 250Hz. The default joint position controller (PD) of the robot runs at 1.0kHz, and it is active during the execution of the tasks. Such controller compensates for modeling inaccuracies and improves the tracking performance of the legs during

TABLE I: Gait parameters and Optimization results

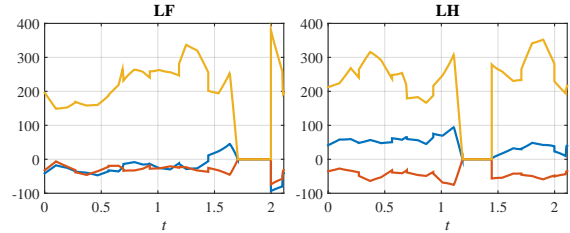|	Walking	Trotting
Nodes	36	26
Phases	9	6
$\mathcal{C}_i$	1111, 1110, 1111, 1011, 1111, 1011, 1111, 0111, 1111	1111, 1001, 0110, 1001, 0110, 1111
Sol. time (s)	140.33	535.16
Iterations (M/m)	14/95590	296/233144
Motion time (s)	2.1	2.0

Fig. 3: Motion Plan: Ground reaction forces during walking. Left Front (LF) and Right Hind (RH) legs. F_x:blue, F_y:red, F_z:yellow. The vertical force tends to zero before lifting off.

swing phase. Hardware results are demonstrated in a video available at http://www.diegopardo.net/rss2017.html.

A. Motion plans

The output of the algorithm are the states and controls at the nodes as well as the time between nodes. Therefore neither the total trajectory time nor the phase duration are pre-specified. However, these times are bounded given the total number of nodes and the bounds set for the time step. The location of the footholds is also not specified beforehand and it results from the optimal state trajectories. The input parameters to the algorithm are the total number of nodes and the contact sequence: number of phases and the contact configuration for each phase. For the case of the point feet quadruped, we represent the contact configuration at phase i as the binary 4-tuple denoting whether a foot is in contact or not, $\mathcal{C}_i = [LF, RF, LH, RH]$, where LF: Left Front, RF: Right Front, LH: Left Hind, RH: Right Hind. Table I summarizes the parameters used for each gait.

We generate the final state trajectories interpolating at 250Hz using the third order polynomial spline implicitly used in the collocation algorithm. Similarly, the final control trajectories are obtained using linear interpolation. All variables of interest can be computed from these trajectories.

1) Walking: We set the final phase constraint with the desired final configuration of the body. We also use $x_{min} = 0.2$m as the minimum distance to walk. Additionally, we use inter-phase constraints during transitions to swing phases constraining the vertical force at the boundary point to be almost zero before lifting the leg,

$$\lambda_z(\mathbf{x}^-, \mathbf{u}^-) \leq 0.01\text{N}.$$

Although this constraint is not necessary to generate the walking gait, the resulting motion is qualitatively less rigid. Fig. 3 shows the ground reaction forces of two feet during
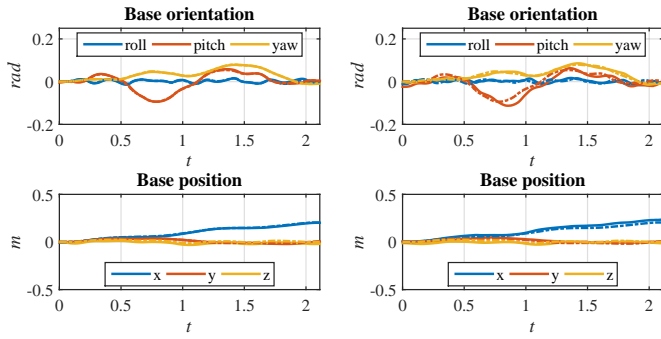
Fig. 4: Trajectories of the robot base, $\mathbf{q_b}$, during walking. Left: Simulation, Right: Real Robot. Top: Base orientation, Bottom: base position. Dashed lines: motion plan.
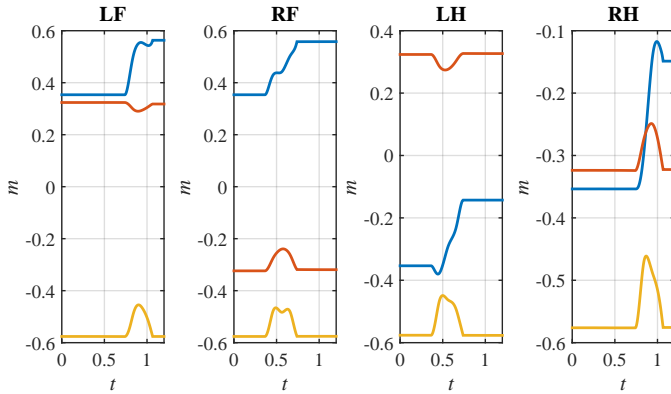

Fig. 5: Trotting motion plan: Feet position trajectories x:blue, y:red, z:yellow. Values are expressed in the inertial frame. $z_{gound} = -0.578$m.

walking. It can be observed how the vertical component decreases before lifting-off.

Fig. 4 shows the resulting base trajectory during walking (simulation and experiment). It can be observed in the plots that the body angles change along the trajectory. In the video it can be observed that such motions release the vertical contact forces from the leg that is about to be lifted. This underlines the importance of the feedback controller. Stabilizing the body trajectory is paramount in order to execute whole-body motion plans in a legged robot.

2) Trotting: During trotting only two feet are in contact with the ground at each phase. Fig. 5 shows the resulting feet trajectories for the case of a two steps trotting. In this plot, the feet position is expressed in the inertial frame, which is aligned with the initial position of the base frame at the default stance configuration of the robot. Therefore, the position of the ground expressed in the inertial frame is negative $z_{ground} = -0.578$m.

In Fig.6 we show the control trajectories obtained during the trotting gait shown in the video (20cm). It can be seen that the contribution of the joint position controller is present along the trajectory, compensating for modeling errors. The contribution of this controller is more important during swing phases. These can be explained by the limitations of the torque-tracking low
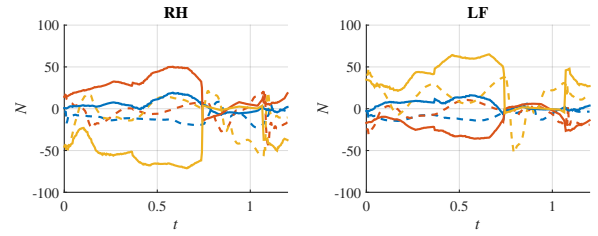

Fig. 6: Control signals during the execution of the trotting task (20cm) in the real robot. Joints corresponding to RH and LF legs. Dashed lines : PD controller, Continuous Lines: $\mathbf{u}^*$

TABLE II: Trotting gaits: specifications and results

Spec./Results	T1	T2	T3
Nodes	22	22	26
Phases	4	4	6
x_{min}(m)	0.20	0.30	0.60
Max linear velocity (m/s)	0.49	0.7	0.72
Motion time (inc.standing phases)	1.22	1.372	2.0

level controller together with the fast motions required when the feet are in the air.

It is possible to set the specifications of the desired motions. In the video we show three different trotting motions. The parameters and resulting characteristics of such trotting motions are summarized in Table II.

Finally, we use this gait to show that the algorithm generates constraint-consistent accelerations. Fig. 7 shows the violation of the acceleration constraints as the difference between the two terms in (8) along a trotting trajectory. Notice that this is not an explicit constraint, but it is implicitly satisfied by using the projected dynamics. Fig. 7 shows that accelerations are consistent along the final splined trajectory, and not only at the nodes of the collocation algorithm. All the motions obtained with our algorithm show the same consistency.

3) Jumping motions: We generate different jumping motions: forward, sideways and turning. The resulting motions of the real robot are shown in the video available at http://www.diegopardo.net/rss2017.html. The contact sequence for jumping is $\mathcal{C}_{jump} = 1111, 0000, 1111$. The robot executes forward jumps with three different distances specifications $x_{jump} = 0.2$m, 0.4m, 0.5m. By changing only one component of the desired final configuration of the body, a completely different body motion is obtained. Similarly, we produce a sideways jumping motion specifying $y_{jump} = 0.2$m. Finally, a very dynamic jumping motion with turn is obtained by specifying the final configuration of the yaw angle of the body $yaw_{jump} = 45°$. In the resulting motion, the 80Kg robot jumps while turning its body on the air before landing.

B. Optimization

All the results were obtained using SNOPT [29] as the NLP solver. Table I shows information about the performance of the optimization in terms of time and number of major (M) and minor (m) iterations required by SNOPT to solve the problem. We also report the resulting trajectory time as reference.
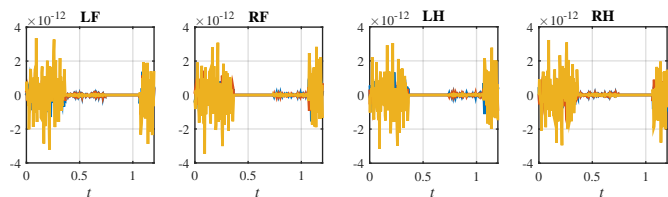
Fig. 7: Acceleration constraint violation, $\eta = \mathbf{J}_c\ddot{\mathbf{q}} - \dot{\mathbf{J}}_c\dot{\mathbf{q}} \in \mathbb{R}^{12}$. During a swing phase, the corresponding legs are not constrained and therefore the constraint violation is null.
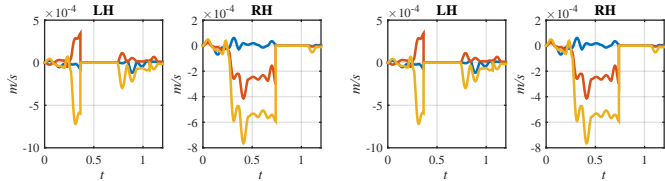

Fig. 8: Velocity constraint violation of the motion plan. Feet velocity, x:blue, y:red, z:yellow.

The cost functions used to generate the motions are rather simple. Minimizing a quadratic cost on the joint velocities or in the body accelerations produces smooth motions. Even not using a cost function at all produces feasible motions connecting the phases.

VII. DISCUSSION

1) Integration error: The approximation of the trajectories used in the direct collocation algorithm results in integration errors. The error over a single time step is $\mathcal{O}(\Delta T_l^4)$ [8]. Integrating the acceleration constraint violation shown in the previous section (4×10^{-12}) inevitable leads to drift. In Fig.8 we show the magnitude of the velocity constraint violation. The accurate representation used by direct collocation together with the consistent accelerations provided by the forward dynamics in (13) and the penetration constraints result in a feasible motion plan.

2) Velocity discontinuities: Impulsive impacts should not be avoided when planning a dynamic motion. However, in practice high impact forces create big disturbances that may hinder the performance of the controller. We have implemented a mechanism to bound the resulting impulse, by adding a suplementary inter-phase constraint. We bound the square of the difference between velocities before and after the impact,

$$(\dot{q}_i^+ - \dot{q}_i^-)^2 < d^2 \quad \forall i = 1, ..., n + 6.$$

3) Transferring trajectories to the real robot: The main benefit of using the projected dynamics in the TO formulation is that the resulting optimal solution is transferable to the real robot. It avoids the assumptions introduced when using soft contact models, and it allows to explicitly handle impacts. The resulting plans depend only on the states and control trajectories, which are easier to stabilize. Executing such trajectories on the real robot generates consistent contact forces. Moreover, there are no assumptions regarding the behavior of the contact forces in between the nodes of the trajectory.

4) Contact forces are not decision variables: It is not clear what is the impact, in terms of computation times, of this new way of handling contact forces in a direct method. It avoids the use of a complementary constraint that introduces nonconvexity to the NLP. Moreover, our approach also lowers the dimensionality of the problem. However, additional nonlinear constraints are introduced to ensure unilateral contact forces. An in-depth comparison against [7],[8] is necessary to determine the differences in terms of computation times.

On the other hand, our method requires a pre-specified sequence of contact configurations, which is not mandatory when contact forces are part of the decision variables. Nevertheless, this does not represent a limitation to generate diverse gaits and dynamic motions.

VIII. CONCLUSIONS

We present a generalized framework to generate and control dynamic motions of legged robots. The framework can produce and stabilize periodic and aperiodic motions. Task specifications are given by the number of phases, the contact configuration at each phase and the desired pose of the robot at the end of the motion. The same interface controls all the motions, without implementing a dedicated controller for specific tasks. Moreover, the approach is independent of a specific robot morphology.

Our method assumes the hybrid nature of a legged robot, allowing velocity discontinuities when switching from swing to stance phases. The method does neither depend on a parametric contact model nor on a specific representation of contact forces. We demonstrated that this method produces motions that are consistent with the contact constraints. This makes the plans feasible to be stabilized on the robot using a inverse dynamics approach.

REFERENCES

[1] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal. Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research*, 30(2):236–258, nov 2010.

[2] 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 *2013 IEEE International Conference on Robotics and Automation*, pages 2554–2561, May 2013.

[3] Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D Running of MIT Cheetah 2. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June): 5163–5170, 2015.

[4] Pierre-Brice Wieber, Russ Tedrake, and Scott Kuindersma. *Modeling and Control of Legged Robots*, pages 1203–1234. Springer International Publishing, Cham, 2016.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking

pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation*, volume 2, pages 1620–1626 vol.2, Sept 2003.

[6] G. Schultz and K. Mombaur. Modeling and Optimal Control of Human-Like Running. *Mechatronics, IEEE/ASME Transactions on*, 15(5):783–792, 2010.

[7] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, 2014.

[8] M. Posa, S. Kuindersma, and R. Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1366–1373, May 2016.

[9] Hongkai Dai, Andres Valenzuela, and Russ Tedrake. Whole-body Motion Planning with Centroidal Dynamics and Full Kinematics. In *IEEE-RAS International Conference on Humanoids Robots*, 2014.

[10] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, Oct 2012.

[11] Michael Neunert, Farbod Farshidian, Alexander W. Winkler, and Jonas Buchli. Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds. *arxiv*, abs/1607.04537, 2016.

[12] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames. 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1447–1454, May 2016.

[13] F. Aghili. A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *IEEE Transactions on Robotics*, 21(5):834–849, Oct 2005.

[14] M. Mistry, J. Buchli, and S. Schaal. Inverse dynamics control of floating base systems using orthogonal decomposition. In *2010 IEEE International Conference on Robotics and Automation*, pages 3406–3412, May 2010.

[15] John T. Betts. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[16] Marco Hutter, Hannes Sommer, Christian Gehring, Mark Hoepflinger, Michael Bloesch, and Roland Siegwart. Quadrupedal locomotion using hierarchical operational space control. *The International Journal of Robotics Research*, 33(8):1047–1062, 2014.

[17] Kai Henning Koch, Katja Mombaur, and Philippe Soueres. Optimization-based walking generation for humanoid robot. *IFAC Proceedings Volumes*, 45(22):498 – 504, 2012.

[18] Sbastien Lengagne, Joris Vaillant, Eiichi Yoshida, and Abderrahmane Kheddar. Generation of whole-body optimal dynamic multi-contact motions. *The International Journal of Robotics Research*, 32(9-10):1104–1119, 2013.

[19] K. Yunt and C. Glocker. A Combined Continuation and Penalty Method for the Determination of Optimal Hybrid Mechanical Trajectories. In *Symposium on Dynamics and Control of Nonlinear Systems with Uncertainty*, 2007.

[20] W. Xi and C. D. Remy. Optimal gaits and motions for legged robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3259–3265, Sept 2014.

[21] Farbod Farshidian, Michael Neunert, Alexander W. Winkler, Gonzalo Rey, and Jonas Buchli. An Efficient Optimal Planning and Control Framework For Quadrupedal Locomotion. *arxiv*, abs/1609.09861, 2016.

[22] M. Hardt, K. Kreutz-Delgado, and J. W. Helton. Optimal biped walking with a complete dynamical model. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 3, pages 2999–3004 vol.3, 1999.

[23] C. Hargraves and S. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.

[24] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 0387743146.

[25] F. Pfeiffer and C. Glocker. *Multibody Dynamics with Unilateral Contacts*. Wiley, Berlin,Germany, 1996.

[26] L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Inverse dynamics control of floating-base robots with external constraints: A unified view. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1085–1090, 2011.

[27] Ludovic Righetti, Jonas Buchli, Michael Mistry, Mrinal Kalakrishnan, and Stefan Schaal. Optimal distribution of contact forces with inverse-dynamics control. *The International Journal of Robotics Research*, 32(3):280–298, 2013.

[28] Dongeun Seo and Maruthi Ram Akella. Separation Property for the Rigid-Body Attitude Tracking Control Problem. *Journal of Guidance, Control, and Dynamics*, 30(6):1569–1576, 2007.

[29] P. Gill, W. Murray, and M. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review*, 47(1):99–131, 2005.