

# Contact-Implicit Trajectory Optimization using an Analytically Solvable Contact Model for Locomotion on Variable Ground

Iordanis Chatzinikolaïdis<sup>1</sup>, Yangwei You<sup>2</sup>, and Zhibin Li<sup>1</sup>

**Abstract**—This paper presents a novel contact-implicit trajectory optimization method using an analytically solvable contact model to enable planning of interactions with hard, soft, and slippery environments. Specifically, we propose a novel contact model that can be computed in closed-form, satisfies friction cone constraints and can be embedded into direct trajectory optimization frameworks without complementarity constraints. The closed-form solution decouples the computation of the contact forces from other actuation forces and this property is used to formulate a minimal direct optimization problem expressed with configuration variables only. Our simulation study demonstrates the advantages over the rigid contact model and a trajectory optimization approach based on complementarity constraints. The proposed model enables physics-based optimization for a wide range of interactions with hard, slippery, and soft grounds in a unified manner expressed by two parameters only. By computing trotting and jumping motions for a quadruped robot, the proposed optimization demonstrates the versatility for multi-contact motion planning on surfaces with different physical properties.

## I. INTRODUCTION

Physical interactions in humans, animals, and robots require models of the contact properties in the environment to plan movements. The necessary contact forces are generated due to intricate interactions between the contact media and are in practice difficult to model. Therefore, proper modeling of these interactions is important for a motion planning framework that aims to produce contact-rich behaviors on variable grounds, such as locomotion on soft floors.

Environments in our daily life exhibit many properties: they can be hard, soft, slippery, or combinations thereof, as shown in Fig. 1. In terms of modeling, some of these aspects are usually missing in typical motion planning because most contact interactions are assumed rigid. For example, two usual approaches are using either spring-damper [1] or ad hoc penalization schemes [2]. In this work, we present a novel contact model in a principled formalism that can capture such properties without the drawbacks of spring-damper models.

A common solution in complex multi-contact planning is to split the original problem in a series of stages, obtaining more tractable subproblems that are still able to solve the original one. Examples are the work in [3] for general contact plans or the work in [4] with a pre-specified contact

This research is supported by the EPSRC as part of the CDT in Robotics and Autonomous Systems (EP/L016834/1) and the EPSRC UK RAI Hub in Offshore Robotics for Certification of Assets (ORCA) (EP/R026173/1).

<sup>1</sup>Authors are with the Edinburgh Centre for Robotics and School of Informatics, The University of Edinburgh, Edinburgh, UK. Corresponding authors' emails: iordanis.cs@gmail.com, zhibin.li@ed.ac.uk.

<sup>2</sup>Author is with the Institute for Infocomm Research, Agency for Science, Technology and Research (A\*STAR), Singapore.

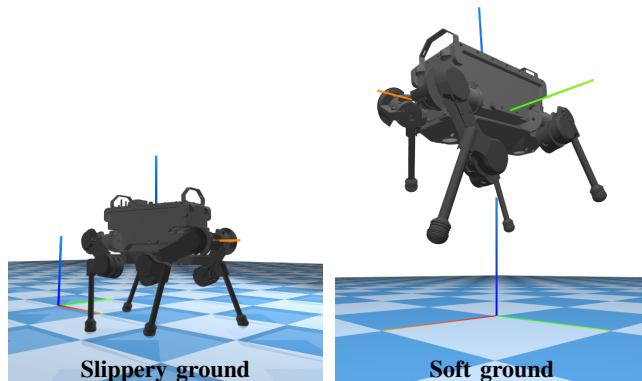


Fig. 1: Dynamic motions computed by the proposed framework: trotting on slippery ground (left); jumping on soft ground (right). (See slippery, soft ground effects in video).

sequence. Their main benefit is fast computation since each stage is usually designed to be efficiently solvable. However, it is challenging to properly design these stages to compute general plans, without restricting the solution space or leading to infeasibilities for the subsequent stages. The focus here is on approaches that avoid such decompositions and can reason about the generated motion plans holistically.

Trajectory optimization (TO) has emerged as a powerful framework to design locally optimal trajectories for highly dynamical and underactuated systems [5], [6]. One of its main benefits is that it allows the setting of high-level goals, expressed as cost functions, while outputting a variety of motions as solutions. This is especially important for legged locomotion and multi-contact motion. Traditional approaches struggle to generalize across different scenarios or non-periodic motions, while TO methods are significantly more versatile [7], [8]. Expressing complex multi-contact planning through a TO lens enables the solution of a wide range of problems with minimum modifications. Though learning approaches demonstrate similar generalization [9], [10], this letter focuses on the model-based optimization paradigm.

A significant problem in TO with contacts is proper modeling as planning requires discontinuous and combinatorial reasoning. Thus, some approaches focus on embedding part of the problem in the TO description. This is possible for simple legged systems; for instance, one-leg hoppers [11] and bipeds with a pre-defined periodic gait pattern like running [12]. While this leads to problems with a very specific structure that are usually easier to solve, adapting to different legged configurations is difficult. This work focuses on approaches that work for arbitrary legged systems. This is possible for contact-implicit formulations that do not require

a priori specification of the contact sequence.

An alternative approach is to describe the problem in a bilevel fashion: The outer level updates the state of the model, while the inner level computes the contact information [13], [14], [15]. Bilevel methods are usually solved by formulating the Karush–Kuhn–Tucker conditions of the inner level, leading to a mathematical program with complementarity constraints (MPCC) [16]. Contrary to bilevel approaches, our work focuses on direct methods that formulate and solve the problem in a single level, such that the optimizer can reason about contacts by optimizing forces, with benefits for long-term physical reasoning [17].

Enabling the optimizer to directly reason about contact forces was proven very powerful for generating complicated contact-implicit motion plans. One way is to allow contact forces to act from a distance [18]; while this is important for discovering contacts, penalizing these forces for physically realistic motion can be challenging. A more principled formalism is introduced in [19], where the problem is elegantly posed as an MPCC. This allows leveraging relaxations for this class of problems already studied in the optimization literature. Albeit these relaxations, a fundamental problem lies in the complementarity constraints, which usually violate constraint qualification tests [6]. Some of these constraints are due to the contact model used.

Therefore, our principal motivation is to introduce a contact model that does not require the specification of complementarity constraints. Such an idea is discussed in [20], where they propose a pair of convex optimization problems that compute the contact forces for simulation purposes. For direct TO, the previous work either focused on MPCC formulations or used a spring-damper model [7], [21].

In this work, we present a TO formulation with a contact model expressed as a pair of quadratic problems that can be computed in closed form. Thus, complementarity constraints are not required while problems associated with spring-damper models such as energy injections, stiff differential equations, and difficulties imposing the friction cone constraint are mitigated. Furthermore, our framework allows deriving the equations of motions for physical interaction with environments characterized by different stiffness, viscosity, and friction. By using the proposed framework, a variety of motion plans can be computed for hard, soft, and slippery surfaces by setting a small number of parameters. The contributions are summarized as follows:

- An analytically solvable contact model suitable for direct contact-implicit TO, which can be utilized in formulations without complementarity constraints, while satisfying unilaterality and friction cone constraints.
- The proposed contact model is generic and can be used to compute motion plans on hard, soft, and slippery surfaces in a unified manner.
- A TO framework that integrates the new contact model for generating contact-implicit motion plans for a high degree of freedom robot, demonstrating the advantages of the proposed method with extensive comparisons performed against the rigid contact model and a TO

formulation with complementarity constraints.

The remaining sections are organized as follows. [Sec. II](#) describes how contacts are resolved in a simulation setting and introduces two contact models. In [Sec. III](#), the proposed contact model is derived and the overall direct TO formulation is elaborated. [Sec. IV](#) presents the comparisons with (i) a different contact model, (ii) an alternative direct TO formulation based on complementarity constraints, (iii) followed by a variety of computed quadrupedal motions on terrains with different properties. Finally, we summarize and discuss future outlooks in [Sec. V](#).

## II. BACKGROUND AND PRELIMINARIES

### A. Dynamics with contacts

The equations of motion of a typical robot model are

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = S\boldsymbol{\tau} + \sum_i J_i^T(\mathbf{q})\mathbf{f}_i, \quad (1)$$

where  $M$  is the mass matrix,  $\mathbf{H}$  the vector of nonlinear forces (Coriolis, centrifugal, and gravitational),  $S$  is a selection matrix that maps actuated joint torques  $\boldsymbol{\tau}$  to generalized coordinates, while  $J_i$  denotes the Jacobian of the  $i$ -th contact and  $\mathbf{f}_i$  the corresponding force. We simplify the notation by dropping explicit dependence on quantities and write  $M$  instead of  $M(\mathbf{q})$ , *etc.* Furthermore, we denote the generalized velocity and acceleration as the time derivative of the configuration, though this is not necessarily the case; for example, if the floating base is represented using quaternions, the angular velocity of the base is not equal to the rate of change of the quaternions.

Next, we follow a similar treatment with time-stepping approaches, *e.g.* [22], [23]. Equation (1) is discretized using an Euler approximation to obtain the discrete version

$$M_k(\dot{\mathbf{q}}_{k+1} - \dot{\mathbf{q}}_k) = h(S\boldsymbol{\tau}_k - \mathbf{H}_k) + J_k^T \boldsymbol{\lambda}_k, \quad (2)$$

where  $h$  is the time step and  $\boldsymbol{\lambda}_k$  corresponds to the concatenation of the contact impulses at time step  $k$ . Next, the discrete equations of motions are projected in contact space

$$J_k(\dot{\mathbf{q}}_{k+1} - \dot{\mathbf{q}}_k) = J_k M_k^{-1} [h(S\boldsymbol{\tau}_k - \mathbf{H}_k) + J_k^T \boldsymbol{\lambda}_k]. \quad (3)$$

Alternatively, (3) can be expressed as

$$\mathbf{v}^+ = A\boldsymbol{\lambda} + \mathbf{b}(\boldsymbol{\tau}) + \mathbf{v}^-, \quad (4)$$

with  $\mathbf{v}^+ = J_k \dot{\mathbf{q}}_{k+1}$ ,  $\mathbf{v}^- = J_k \dot{\mathbf{q}}_k$ ,  $\mathbf{b} = h J_k M_k^{-1} (S\boldsymbol{\tau}_k - \mathbf{H}_k)$ , and  $A = J_k M_k^{-1} J_k^T$ .

Based on (4), two cases are identified [22]: (i) The *forward contact dynamics* case, where we want to compute  $(\mathbf{v}^+, \boldsymbol{\lambda})$  given  $(A, \mathbf{b}, \mathbf{v}^-)$ ; (ii) The *inverse contact dynamics* case, where we want to compute  $(\mathbf{b}, \boldsymbol{\lambda})$  given  $(A, \mathbf{v}^-, \mathbf{v}^+)$ . The forward contact dynamics is more relevant in simulation, while the inverse is more relevant in a TO setting. In the latter case, decomposing actuation from contact impulses for an underactuated robot model is challenging, which is an aspect addressed in this work.

Finally, the contact impulses should satisfy unilateral and friction cone constraints, *i.e.* each impulse must satisfy

$$F_{\mu_i} = \{ \boldsymbol{\lambda}_i \mid \lambda_{n(i)} \geq 0, \|\boldsymbol{\lambda}_{t(i)}\|^2 \leq \mu_i^2 \lambda_{n(i)}^2 \}, \quad (5)$$

where  $\lambda_i = [\lambda_{t(i)} \ \lambda_{n(i)}]^T$  are the tangential and normal components. Equations (4) and (5) form the backbone of impulse-based time-stepping methods [24]. The contact models discussed next provide different approaches on how to solve them.

## B. Contact models

*Nonlinear complementarity problem (NCP).* This contact model augments (4) and (5) with an additional constraint: That either the contact normal distance or the normal contact impulse at the next time instant is zero. This is usually described succinctly as  $0 \leq \lambda_n \perp d^+ \geq 0$ , where  $d^+$  is the next step normal gap distance required to be nonnegative to avoid penetration. In simulation  $d^+$  is generally unknown, so this condition is approximated by performing its Taylor expansion and forming a complementarity constraint between the gap velocity and normal contact impulse [23]. Apart from the friction cone constraint, the tangential components of the impulse are specified via the *maximum dissipation principle* (MDP). It specifies that friction forces maximize the rate of the kinetic energy dissipation. This principle was introduced in [25] for a frictional contact at a single point.

*Convex optimization formulation.* The convex contact model relaxes the complementarity condition by forming a quadratic problem that penalizes movement in contact space [22]. In the forward dynamics case, it is specified by the quadratically constrained quadratic program [23]

$$\begin{aligned} \min_{\lambda} \quad & \frac{1}{2} \lambda^T (A + R) \lambda + \lambda^T (\mathbf{b} + \mathbf{v}^-) \\ \text{s.t.} \quad & \lambda_i \in \mathbb{F}_{\mu_i}, \forall i \end{aligned} \quad (6)$$

that computes the contact impulses, where  $R$  is a positive definite matrix that makes the solution unique and invertible. Physically this makes the contact model soft; for hard contacts, there can be an infinite number of contact impulse combinations. The inverse dynamics case is well-defined and the impulses are given by

$$\begin{aligned} \min_{\lambda} \quad & \frac{1}{2} \lambda^T R \lambda + \lambda^T \mathbf{v}^+ \\ \text{s.t.} \quad & \lambda_i \in \mathbb{F}_{\mu_i}, \forall i. \end{aligned} \quad (7)$$

For a diagonal  $R$ , the latter optimization problem splits into independent problems, one for each contact.

## III. TRAJECTORY OPTIMIZATION FORMULATION

### A. Optimal control problem

The continuous optimal control problem (OCP) can be expressed as

$$\min_{\mathbf{q}(t), \boldsymbol{\tau}(t)} \quad l_f(\mathbf{q}_T) + \int_0^T l(\mathbf{q}) + c(\boldsymbol{\tau}) dt \quad (8a)$$

$$\text{s.t.} \quad M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = S\boldsymbol{\tau} + J^T(\mathbf{q})\mathbf{f} \quad (8b)$$

$$\mathbf{f} \in \begin{cases} \arg \min_{\mathbf{f}} k(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}) \\ \text{s.t.} \quad \mathbf{f} \in F_{\mu} \end{cases} \quad (8c)$$

$$\mathbf{g}(\mathbf{q}, \boldsymbol{\tau}) \in \mathbb{Z} \quad (8d)$$

$$\mathbf{q}(0) = \mathbf{q}^0 \quad (8e)$$

$$\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}^0 \quad (8e)$$

$$t \in [0, T], \quad (8f)$$

where  $l(\mathbf{q})$  is an additive cost associated with the joint trajectory,  $l_f(\mathbf{q}_T)$  is the final state cost, and  $c(\boldsymbol{\tau})$  is the cost of the joint torques. These can be general sufficiently smooth functions but we focus on positive definite quadratic forms. Equation (8b) specifies the dynamics of the system, where  $J(\mathbf{q})$  and  $\mathbf{f}$  are the concatenated Jacobians and contact forces, respectively. The constraints (8d) specify general path constraints imposed on the optimal trajectory, *e.g.* joint and torque limits. Finally, (8e) and (8f) specifies the initial state and the time.

The optimization problem in (8c) specifies the contact forces via a mathematical program which makes (8) a bilevel optimization problem. The first approach is to provide gradients to the upper level via sensitivity analysis, but this can make long-term physical reasoning hard [17]. The second approach is to introduce the contact forces as variables, and then describe (8c) via its first-order necessary (KKT) conditions; that is, by imposing complementarity constraints. For example, the NCP contact model requires constraints for avoiding penetrations and the KKT conditions of the MDP [26]. A third approach is to solve (8c) for the contact forces, which are then not introduced as variables, and the associated complementarity constraints become unnecessary. We present such an approach next.

### B. Contact model with analytical solution

First, the solution for the normal components is specified. We call it the *frictionless case*. To obtain a unique solution, the strict non-penetration constraint is replaced with a quadratic program that has a unique solution and penalizes penetrations and the magnitude of the normal forces, while satisfying unilateral contact impulse constraints.

Given this solution, the tangential components are computed for what we call the *friction case*. Instead of the MDP, the velocities in contact space are minimized as in [22]. Since we focus on TO, the advantage of this approach is that an invertible contact model for the tangential components analogous to (7) can be formulated, and its unique minimum can be analytically derived.

1) *Frictionless case:* The following quadratic problem specifies each normal component as

$$\begin{aligned} \min_{\lambda_{n(i)}} \quad & \frac{1}{2r_{n(i)}} \lambda_{n(i)}^2 + \lambda_{n(i)} d_i^+(q) \\ \text{s.t.} \quad & \lambda_{n(i)} \geq 0. \end{aligned} \quad (9)$$

Since the next step gap distance is available, the velocity complementarity constraint becomes unnecessary. The solution is

$$\lambda_{n(i)} = r_{n(i)} \max \{-d_i^+(q), 0\}. \quad (10)$$

Notice that the model depends on information from the next time instant, in contrast to traditional spring models. By examining the solution, penetration occurs when the contact impulses are activated and become positive; otherwise, they are zero. Furthermore,  $r_{n(i)}$  expresses the trade-off between the magnitude of the normal impulse and the penetration depth. The larger  $r_{n(i)}$ , the smaller the penetration.

2) *Friction case*: The optimization problem that minimizes the velocities in the contact frame for the inverse dynamics case has the form

$$\begin{aligned} \min_{\boldsymbol{\lambda}_{t(i)}} \quad & \frac{1}{2} \boldsymbol{\lambda}_{t(i)}^T R_{t(i)}^{-1} \boldsymbol{\lambda}_{t(i)} + \boldsymbol{\lambda}_{t(i)}^T \mathbf{v}_{t(i)}^+ \\ \text{s.t.} \quad & \|\boldsymbol{\lambda}_{t(i)}\|^2 \leq \mu_i^2 \lambda_{n(i)}^2, \end{aligned} \quad (11)$$

where  $R_{t(i)} = \text{diag}\{r_{t(i)}, r_{t(i)}\}$ . Essentially,  $r_t$  trades off tangential velocity and force; the smaller its value, the more tangential forces are penalized.

This is a projection to circle problem and two cases can be identified. If the solution lies inside the cone, the problem is an unconstrained quadratic one, and the solution is

$$\boldsymbol{\lambda}_{t(i)} = -R_{t(i)} \mathbf{v}_{t(i)}^+. \quad (12)$$

Otherwise, the solution lies on the boundary

$$\boldsymbol{\lambda}_{t(i)} = -\mu_i \lambda_{n(i)} \hat{\mathbf{v}}_{t(i)}^+, \quad (13)$$

where  $\hat{\mathbf{v}}_{t(i)}^+ = \mathbf{v}_{t(i)}^+ / \|\mathbf{v}_{t(i)}^+\|$ .

For  $r_{t(i)} \rightarrow 0$  the solution approaches the frictionless case, while for  $r_{t(i)} \gg 0$  energy dissipation is increased. The tangential components of the impulse are given by (12) if  $r_{t(i)} \|\mathbf{v}_{t(i)}^+\| \leq \mu_i \lambda_{n(i)}$ ; otherwise, they are given by (13). Finally, the tangential force is opposite to the tangential velocity; thus, the reaction force is dissipative.

3) *Smoothing*: In continuous optimization, smoothness of the objective and the constraints for at least the second derivative is required [27]. From the previous analysis, it is clear that the computed impulses contain switches that can cause problems for the optimizer. Therefore, a procedure to remove the discontinuities is discussed below. First, the solution for the *friction case* is expressed using a max function,

$$\boldsymbol{\lambda}_{t(i)} = \hat{\mathbf{v}}_{t(i)}^+ \max \left\{ -\mu_i \lambda_{n(i)}, -r_{t(i)} \|\mathbf{v}_{t(i)}^+\| \right\}. \quad (14)$$

As a result, a smooth approximation to the max function is required, which is now present in the solutions for both cases. Multiple definitions for a smooth max function exist, and the selected `softmax` function is

$$\text{softmax}(\alpha, \beta; \epsilon) = \frac{\alpha + \beta + \sqrt{(\alpha - \beta)^2 + \epsilon^2}}{2}, \quad (15)$$

where for  $\epsilon > 0 \rightarrow 0$  the approximation becomes stricter.

### C. Direct transcription

According to the solutions (10) and (14), the contact impulses are described as a function of joint configurations and velocities, and the Jacobian can be used to map these quantities to the contact velocity in (14). Afterwards, these

TABLE I: Parameters for the unactuated rigid body models

Model	Position [m]	Orientation (MRP)	Body angular vel. [rad/s]	Body linear vel. [m/s]
Ball	$\begin{bmatrix} 0.1 \\ 0.75 \\ 0.3 \end{bmatrix}$	$\begin{bmatrix} 0.1617 \\ 0.566 \\ 0.0809 \end{bmatrix}$	$\begin{bmatrix} 0.372 \\ 1.208 \\ 0.834 \end{bmatrix}$	$\begin{bmatrix} 1.379 \\ 1.386 \\ 0.743 \end{bmatrix}$
Brick	$\begin{bmatrix} 0.1 \\ 0.75 \\ 1.7 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0.2 \\ 0.126 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

terms can be substituted in (8b), which becomes a function of the joint positions, velocities, and accelerations only.

Note that (8) is an infinite-dimensional continuous problem that can be transcribed to a finite discrete one [6]. An implicit Euler discretization is selected due to its numerical properties, e.g. it is an A-stable method. Thus, problem (8) can be expressed as

$$\min_{\mathbf{q}_k, \dot{\mathbf{q}}_k, \boldsymbol{\tau}_k} \quad l_f(\mathbf{q}_N) + h \sum_{k=1}^{N-1} l_k(\mathbf{q}_k, \dot{\mathbf{q}}_k) + c_k(\boldsymbol{\tau}_k) \quad (16a)$$

$$\text{s.t.} \quad M_{k+1}(\dot{\mathbf{q}}_{k+1} - \dot{\mathbf{q}}_k) = h(S\boldsymbol{\tau}_{k+1} - \mathbf{H}_{k+1} + J_{k+1}^T \boldsymbol{\lambda}(\mathbf{q}_{k+1}, \dot{\mathbf{q}}_{k+1})) \quad (16b)$$

$$h\dot{\mathbf{q}}_{k+1} = \mathbf{q}_{k+1} - \mathbf{q}_k \quad (16c)$$

$$\mathbf{g}(\mathbf{q}_k, \boldsymbol{\tau}_k) \in \mathbb{Z} \quad (16d)$$

$$\mathbf{q}_0 = \mathbf{q}^0 \quad (16e)$$

$$\dot{\mathbf{q}}_0 = \dot{\mathbf{q}}^0 \quad (16e)$$

$$k \in [0, N-1]. \quad (16f)$$

Since (16c) is linear to joint velocities, they can be removed from the optimization problem by substituting the right-hand side. Similarly, joint torques are split into actuated  $\boldsymbol{\tau}$  and underactuated  $\boldsymbol{\tau}^u$  parts using (16b) [15]. Then  $\boldsymbol{\tau}$  is given as a function of the joint configurations and can be substituted directly on the objective (16a). Finally, the underactuated dynamics should be zero, yielding the overall trajectory optimization problem

$$\min_{\mathbf{q}_k} \quad l_f(\mathbf{q}_N) + h \sum_{k=1}^{N-1} l_k(\mathbf{q}_k) + c_k(\boldsymbol{\tau}(\mathbf{q}_k, \mathbf{q}_{k+1})) \quad (17a)$$

$$\text{s.t.} \quad M_{k+1}^u(\mathbf{q}_{k+1} - 2\mathbf{q}_k + \mathbf{q}_{k-1}) = h^2(J_{k+1}^{uT} \boldsymbol{\lambda}_{k+1} - \mathbf{H}_{k+1}^u) \quad (17b)$$

$$\mathbf{g}(\mathbf{q}_k, \boldsymbol{\tau}_k) \in \mathbb{Z} \quad (17c)$$

$$\mathbf{q}_0 = \mathbf{q}^0 \quad (17d)$$

$$\mathbf{q}_{-1} = \mathbf{q}^0 - h\dot{\mathbf{q}}^0 \quad (17d)$$

$$k \in [0, N-1], \quad (17e)$$

where  $(\cdot)^u$  denotes the unactuated part of the quantity. It is worth noting that the Hessian and Jacobian remain sparse.

Finally, the modified Rodrigues parameters (MRP) are selected [28] to represent the floating base. As any three-parameter orientation parameterization, it possesses a singularity. For the MRP representation, this singularity is located after a full revolution. That places the singularity as far as possible from the origin. The polynomial expression and lack of unit norm constraints make this representation a suitable candidate for nonlinear optimization.

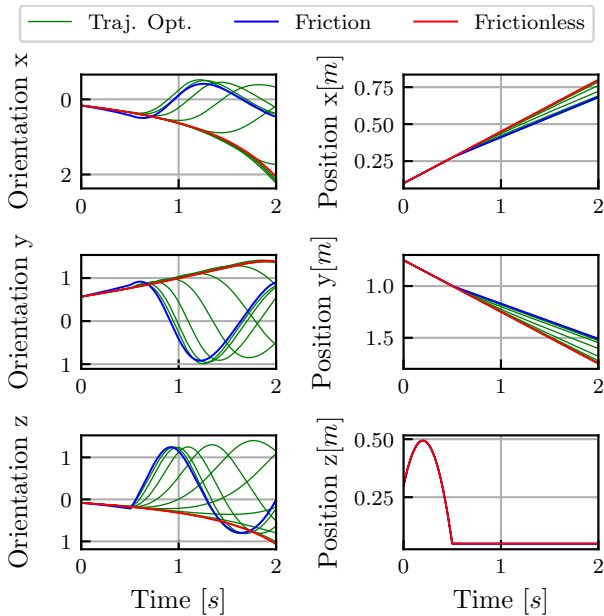


Fig. 2: By changing the value of the parameter  $r_t$ , we can obtain a family of solutions that range from the frictionless to the dissipative friction case.

#### IV. RESULTS AND VALIDATION

Next, simulations are conducted to quantitatively validate the proposed formulation. For all the cases, the optimization problems are formulated with `CasADi` [29] (for automatic differentiation), and solved by `IPOPT`, a large scale interior-point solver [30]. `IPOPT` allows the selection of a linear solver for computing the Newton steps; we used the `MA57` solver when performing comparisons and `MA86` otherwise [31]. The rigid body dynamics of the models are computed using the `RigidBodyDynamics.jl` library [32].

As in all nonlinear optimization problems, proper scaling is important. Since configurations are the only variables, scaling here is straightforward and only the position of the floating joint requires special treatment. For the constraints and objective function, we use the default gradient-based scaling available in `IPOPT`.

First, the results for two basic models are studied before proceeding to a complex robot model: A rigid ball model that constitutes the simplest 3D floating model with one contact point and a rigid brick with eight contact points. These models are suitable for benchmarking and require minimum parameter tuning; the small number of states and the unique state solution provide a framework for direct comparisons. For benchmarking, we avoided more complicated models which can make the results less comparable due to the high-dimensional representations and non-trivial choices of parameters.

The initial state for all ball and brick simulations is summarized in Table I, where the root body spatial velocity is defined in the body frame, while we use  $\epsilon = 0.001$  for smoothing. Regarding the simulated motions, the ball and the brick were dropped on a flat ground. Parameter  $r_n$  is selected so that no bouncing occurs. Thus, for the frictionless case,

TABLE II: Running time and iterations of the MPCC versus our proposed formulations

Method	MPCC		Proposed		n
	Wall time [s]	Iter.	Wall time [s]	Iter.	
Ball frictionless	$0.773 \pm 0.011$	24	$0.507 \pm 0.016$	15	5
Ball friction	$3.78 \pm 0.047$	104	$2.374 \pm 0.023$	64	5
Brick frictionless	$1340 \pm 15.27$	6034	$508.83 \pm 0.56$	1893	3
Brick friction	$1151 \pm 20.18$	3015	$317.68 \pm 4.69$	819	3

the ball slides on the surface, while it rolls when friction is present. The dropped brick touches the plane with a line contact. It slowly rotates until settling on a large side—where four vertices are active contact points<sup>1</sup>.

##### A. Comparison with a physics simulation

To evaluate the contact impulses computed by the proposed approach, we performed a comparison with the rigid contact model which is typically used in simulation engines. The aim is to demonstrate that the model can represent different environmental interactions by an intuitive selection of its parameters. Assuming that the frictionless case corresponds to an extremely slippery interaction, our model can simulate very slippery conditions (numerically identical to the frictionless one), up until minimally slippery (numerically identical to the friction case).

For the primitive models, the optimization is equivalent to a root-finding problem. We perform comparisons against the nonlinear complementarity model with the Projected Gauss-Seidel (PGS) solver [23]. This implementation uses a semi-implicit Euler integration scheme for the dynamics (so that the problem remains linear) while we use an implicit one. Furthermore, that trajectory is computed step by step since the computation is done in a simulation setting, whereas our TO formulation computes the whole trajectory simultaneously.

Fig. 2 illustrates the position and orientation of the ball. First, the PGS solver was executed with and without friction and the resulting two solutions are plotted. Afterwards,  $r_n = 100^N/m$  is fixed for the normal component, and a parameter sweep is performed for  $r_t$ , with  $\mu = 0.5$ . It is verified that by changing the parameter  $r_t$ , we obtain the friction and frictionless solutions, as well as additional in-between solutions; these correspond to slipping motions if  $r_t$  is small, while more dissipative as  $r_t$  increases.

##### B. Comparison with an MPCC formulation

An approach most related with our proposed is the one presented in [26], which is an extension of [19]. Therefore, we perform numerical comparisons to understand their differences. The results are analyzed in terms of the number of iterations and solution time. Two comparisons are performed for each model, one in a frictionless setting and one with friction. The mean, standard deviation, and iterations are shown in Table II, where each simulation is repeated  $n$  times.

<sup>1</sup>A video of all simulated cases is available at [youtu.be/eLx1DebDHmY](https://youtu.be/eLx1DebDHmY)

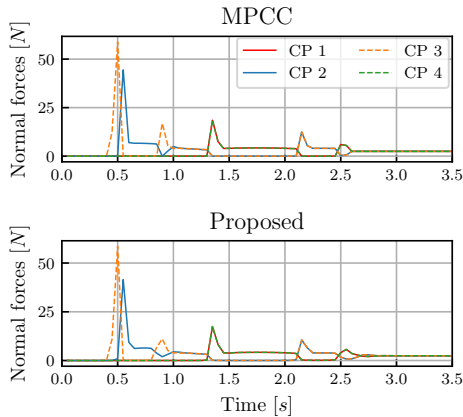


Fig. 3: Comparison of the normal impulses at the four contact points (CP) of the brick with friction. The contact impulses at the other four unactivated vertices are zero and not displayed.

For the comparison, the relaxation suitable for interior-point methods is utilized [26]. The dynamics for both methods are enforced using the inverse dynamics approach in Sec. III-C. Regarding the slack variables weighing in [26],  $\alpha = 1$  is selected. The other aspects of the mathematical program with complementarity constraints remain intact.

1) *Ball model*: For both methods, the friction coefficient is  $\mu = 0.5$ , and our model parameters were chosen as  $r_n = 100^N/m$  and  $r_t = 1^N/m/s$  in the friction case, to match the response with the contact model used by the MPCC. We initialize both methods with zero variables, while we select a time step  $h = 0.1s$  and final time  $t_f = 1s$ .

The root-mean-square error (RMSE) for the frictionless case is  $0.0218N$  (only normal force component), while for the friction case is  $(0.0037, 0.007, 0.0218) N$ ; the ball’s mass is  $0.2kg$ . Note that tangential forces are generated only at the knot after the contact event and are zero otherwise. Also, the MPCC method uses a linearized friction cone, whereas we use the full cone model. The linearized version does not properly capture solutions that lie on the boundary.

2) *Brick model*: We select  $\mu = 0.6$ ,  $r_n = 1000^N/m$  and  $r_t = 10^N/m/s$  for our model’s parameters. We initialize both methods with zero variables, with a time step  $h = 0.05s$  for a horizon of  $t_f = 3.5s$ .

In the frictionless case, the number of variables for the MPCC formulation is 1540, while our formulation has 420. Both methods have the same number of equalities (420), while MPCC additionally includes 2240 inequalities. It is also very common to experience plateaus during the iterations with the MPCC approach, while we avoid such a problem by adapting the parameters. These plateaus are the main reason for the increased number of iterations in the brick frictionless case.

Similarly, the number of variables of the MPCC problem is 6580 for the case with friction, while for the proposed method is again 420. The MPCC problem has 2660 equality and 8960 inequality constraints, while ours has 420 equalities and zero inequalities. The computed normal contact impulses are shown in Fig. 3, where the results show only the four activated contact points for clarity.

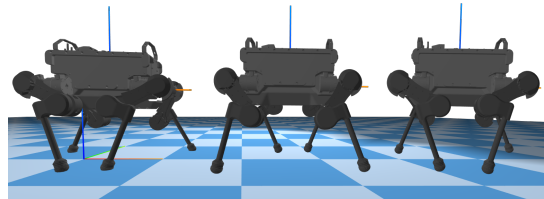


Fig. 4: Trotting on hard ground snapshots (left to right).

3) *Overall remarks*: Based on the comparison, there are two characteristics that are advantageous:

- The size of the optimization problem is kept minimum as we do not introduce extra variables (*e.g.* slack variables, Lagrange multipliers). This is important because general-purpose nonlinear solvers usually demonstrate higher than linear in time complexity since they do not utilize block factorization for solving the problem. Also, the optimizer’s search space is smaller.
- The parameters in our proposed method are few and have direct physical meaning, which makes selection intuitive.

For the same problem instantiation, our method generally converges faster and requires a smaller number of steps. In our experience, choosing parameters that produce similar behaviors is not hard. For the MPCC method, there are no parameters to select and the performance is fixed for a specific problem instantiation.

Finally, our comparison did not include a cost function for reasons explained before. This is favorable for the MPCC formulation because selecting  $\alpha$  in general can be challenging. For actuated systems with multiple solutions,  $\alpha$  needs to be correctly tuned to drive the slack variables to values that sufficiently minimize complementarity violations, without affecting the optimized task. In our approach, there are no such parameters. Once appropriate values for  $r_n$  and  $r_t$  are selected to model the environment, they do not change between different tasks.

### C. ANYmal trotting on hard and slippery surfaces

The proposed method is applied to the quadrupedal robot ANYmal. We use similar gains as in [7] to generate trotting gaits. The torque limits of the system ( $40Nm$ ) are set as inequality constraints. Since the limits are provided, the torque penalization term in the cost function is decreased while the joint velocity penalization term is increased.

Furthermore, a step size of  $h = 0.08s$  and a horizon of  $t_f = 4s$  is selected, resulting in a problem with 900 variables. By selecting this step size, we aim to demonstrate a positive aspect of our contact model: it is able to handle such large step sizes while not suffering from numerical stiffness. The step size is  $\times 40$  bigger than the one used in [7] and might not capture the maximum impulse. Here, we aim to compute a feasible motion plan which can be afterwards interpolated to generate reference trajectories for commanding such a motion on a robot. Since most robots have rubbers on the feet, a certain level of shock absorption not captured by the model is expected. Moreover, problems of short impulses are mitigated in case of locomotion on soft ground.

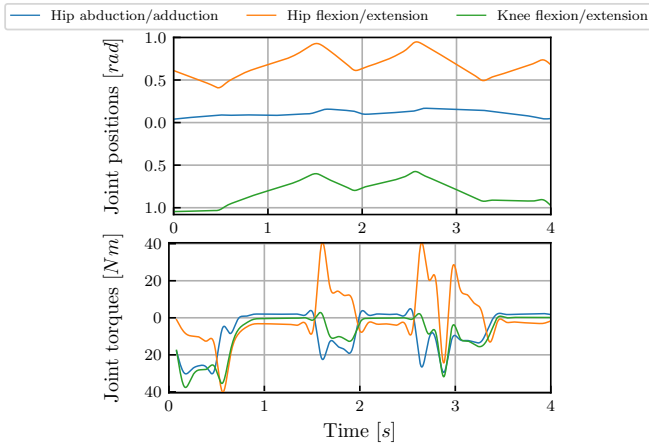


Fig. 5: The joint position and torque ( $40Nm$  limit) trajectories of the left front leg (top) and left hind leg (bottom) for the trotting motion on hard ground.

We initialize the optimizer with a nominal standing configuration for the whole duration; the same configuration is set as initial and desired final. This initialization is used only for the simulation on hard ground. Using this solution, we initialize the same optimization problem on a slippery surface. The purpose of this is to demonstrate the motion adaptations due to different environmental properties. Snapshots of the computed motions are shown in Fig. 4, while in Fig. 6 the position in the  $x$  axis for both cases is shown.

1) *Hard ground*: For this case, we select parameters  $r_n = 20N/m$  and  $r_t = 20N/m/s$ . The diagonal legs step together as expected in a trotting gait, while four steps are taken in total. The duration of each step is different, as shown in Fig. 5; a possible reason is due to the requirement of stopping at the end of the motion. In this case, to start and finish the periodic gait on time, a transient state of fast stepping is necessary.

2) *Slippery ground*: To simulate a slippery ground, we select  $r_t = 4N/m/s$ . Compared to the previous solution, there are two notable differences. First, the ground clearance from the moving legs is significantly smaller. Since the slipping motions are abrupt, the optimizer tries to keep the contact points closer to the surface for fast activation. Second, the solution now relies more heavily on the hind legs to push the body forward, while the front legs are mostly used for stabilization. Similar results are reported in [2].

#### D. ANYmal dynamic jumping

Next, we compute a jumping motion using the ANYmal model on a hard and a soft ground. We use a waypoint at the middle of the trajectory for reaching a  $0.8m$  height. As in the previous case, we specify only the initial and final state, and adapt the gains regarding torque and joint velocity penalization, without a maximum torque inequality constraint. This is because the specified motion is fast and with the current torque limitations the optimizer would struggle to find a solution that reaches the desired height.

We select a time horizon of  $t_f = 3s$  and a step size of  $h = 0.06s$ , resulting in 900 variables. Again, we use the motion computed on the hard ground to initialize the jumping on the

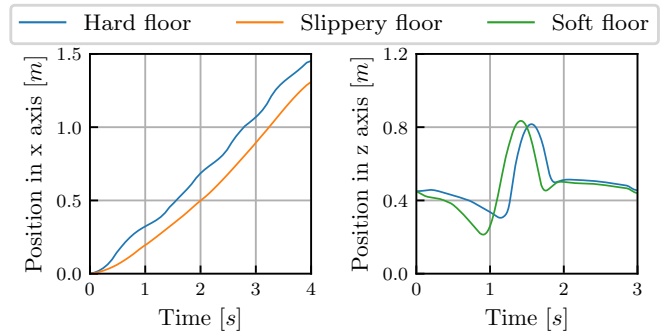


Fig. 6: Body position in the  $x$  axis during trotting on hard and slippery ground, and body position in the  $z$  axis during jumping on hard and soft ground.

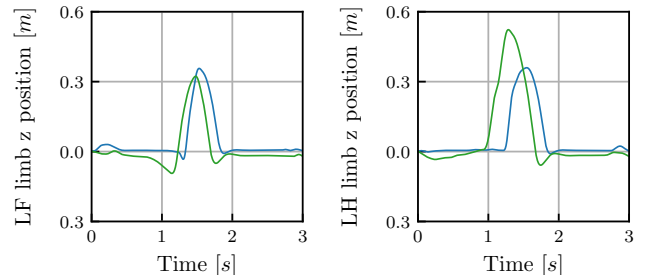


Fig. 7: The foot height during jumping on hard and soft ground for the left front (LF) and left hind (LH) legs.

soft ground, aiming to demonstrate the motion adaptation. Unless specified, we use the same contact model parameters as in the trotting case. The position of the body in the  $z$  axis for both cases is shown in Fig. 6.

1) *Hard ground*: The snapshots from the computed solution are shown in Fig. 8. We identify 6 phases: Lowering the body to prepare for lift-off, the lift-off, reaching the desired height and configuration, touch-down, and transitioning to the desired final state. Changing the torque penalization weight affects the apex height and time instant that this is reached.

2) *Soft ground*: For simulating a soft ground, we select  $r_n = 2N/m$ . The penetration of the limb is shown in Fig. 7, while the snapshots of the motion are shown in Fig. 9. The salient aspects of the motion are the same; differences are identified during the lift-off and touch-down phases. Specifically, during the lift-off preparation, the body is lowered in a similar manner but the feet penetrate deeper into the soft ground. Finally, a small oscillation occurs after touch-down due to the ground's softness and is quickly damped.

## V. DISCUSSION & CONCLUSION

This paper proposed a contact model suitable for direct TO formulations, followed by simulation validations with a wide range of models and settings: from simple objects (ball, brick) to a complex multi-body robot model in various locomotion modes (trotting, jumping) and ground conditions (hard, soft, and slippery). It was demonstrated that the proposed contact-implicit TO method can compute complicated motion plans for multi-contact interactions. An important

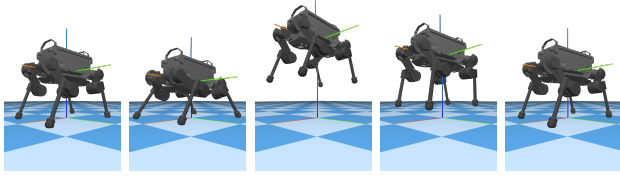


Fig. 8: Optimized jumping motion on hard ground, at the same time instances as in the soft ground case.

feature in the new formulation is an improved, principled contact model which is solved in closed-form and expressed as a function of the state. Furthermore, this contact model avoids complementarity constraints for its description and automatically satisfies friction cone constraints, while not suffering from problems of energy injections and small step sizes. Moreover, it is described by two parameters which have intuitive physical interpretation and can be straightforwardly selected.

Nevertheless, there are several aspects worth of discussion and future improvement. First, the parameters  $r_n$  and  $r_t$  need to be tuned for different robot models or new conditions in the environment's characteristics. This is a common property among the soft contact models. Second, the presented method is suitable for solving contact-implicit planning problems in an offline setting, since the computation can not be performed in real-time yet. The computational time can be improved by taking into account the stage-wise nature of the decision problem. Lastly, a common feature of TO formulations is that the cost function needs to be specified for each task, *i.e.* the cost function is task-dependent.

The motion adaptations for various ground conditions demonstrate the importance of integrating environmental properties into motion planning. Future work will focus on systematic identification of parameters  $r_n$  and  $r_t$  and hardware validations. Another extension of the framework will focus on improving the numerical accuracy by using higher-order integration methods as in [33].

#### REFERENCES

- [1] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, "STANCE: Locomotion adaptation over soft terrain," 2020.
- [2] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization for legged robots with slipping motions," *IEEE Robot. Autom. Lett.*, 2019.
- [3] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," 2018.
- [4] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," 2018.
- [5] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," 2017.
- [6] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, 2010.
- [7] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robot. Autom. Lett.*, 2017.
- [8] A. Radulescu, I. Havoutis, D. G. Caldwell, and C. Semini, "Whole-body trajectory optimization for non-periodic dynamic motions on quadrupedal systems," 2017.
- [9] C. Yang, K. Yuan, S. Heng, T. Komura, and Z. Li, "Learning natural locomotion behaviors for humanoid robots using human bias," *IEEE Robot. Autom. Lett.*, 2020.

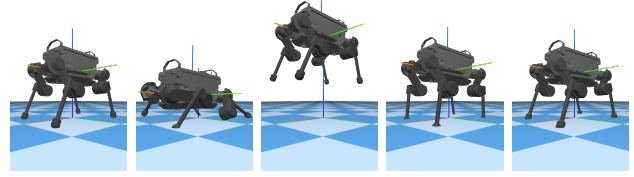


Fig. 9: Optimized jumping motion on soft ground, where the feet penetrate the ground surface.

- [10] H. Dallali, P. Kormushev, Z. Li, and D. Caldwell, "On global optimization of walking gaits for the compliant humanoid robot, COMAN using reinforcement learning," *Cybernetics and Information Technologies*, 2012.
- [11] I. Chatzinikolaïdis, T. Stouraitis, S. Vijayakumar, and Z. Li, "Non-linear optimization using discrete variational mechanics for dynamic maneuvers of a 3d one-leg hopper," 2018.
- [12] K. Mombaur, "Using optimization to create self-stable human-like running," *Robotica*, 2009.
- [13] K. Yunt and C. Glocker, "Trajectory optimization of mechanical hybrid systems using SUMT," in *Proc. IEEE Int. Workshop on Adv. Motion Control*, 2006.
- [14] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization with implicit hard contacts," *IEEE Robot. Autom. Lett.*, 2018.
- [15] T. Erez and E. Todorov, "Trajectory optimization for domains with contacts using inverse dynamics," 2012.
- [16] S. Dempe and J. Dutta, "Is bilevel programming a special case of a mathematical program with complementarity constraints?" 2012.
- [17] M. Toussaint, J.-S. Ha, and D. Driess, "Describing physics for physical reasoning: Force-based sequential manipulation planning," *IEEE Robot. Autom. Lett.*, 2020.
- [18] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," 2012.
- [19] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," 2014.
- [20] E. Drumwright and D. A. Shell, "Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation," in *Algorithmic Foundations of Robotics IX*. Springer, Berlin, Heidelberg, 2010.
- [21] A. Ö. Öno, P. Long, and T. Padr, "Contact-implicit trajectory optimization based on a variable smooth contact model and successive convexification," 2019.
- [22] E. Todorov, "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo," 2014.
- [23] P. C. Horak and J. C. Trinkle, "On the similarities and differences among contact models in robot simulation," *IEEE Robot. Autom. Lett.*, 2019.
- [24] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with Coulomb friction," 2000.
- [25] J. J. Moreau, "On unilateral constraints, friction and plasticity," in *New Variational Techniques in Mathematical Physics*. Springer, Berlin, Heidelberg, 2011.
- [26] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," 2019.
- [27] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Operations Research and Financial Engineering. Springer, 2006.
- [28] G. Terzakis, M. Lourakis, and D. Ait-Boudaoud, "Modified Rodrigues parameters: An efficient representation of orientation in 3D vision and graphics," 2018.
- [29] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADI – a software framework for nonlinear optimization and optimal control," 2018.
- [30] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," 2006.
- [31] "HSL. A collection of Fortran codes for large scale scientific computation." [Online]. Available: <http://www.hsl.rl.ac.uk/>
- [32] T. Koolen and R. Deits, "Julia for robotics: Simulation and real-time control in a high-level programming language," 2019.
- [33] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, "Contact-implicit trajectory optimization using orthogonal collocation," *IEEE Robot. Autom. Lett.*, 2019.