

Contact-Implicit Planning for Contact-Rich Tasks

Stefanos Charalambous

Jia-Ruei Chiu

Jean-Pierre Sleiman

Abstract—Controlling high-dimensional and underactuated robotic systems to execute dynamic, contact-rich motions remains a challenge for optimization-based methods due to the discontinuous dynamics introduced from inelastic impact. Our work exploits the formulation of direct contact-implicit trajectory optimization to smoothen out discontinuities in the first stage of our framework. We are thus able to optimize given a non-informative initialization, while through the use of a continuation method the generated motion plan has strict physical feasibility. The smooth optimization landscape in the first stage enables us to efficiently discover a crude motion plan that places the second and final stage in a desired basin of attraction. In contrast to prior works that rely on different forms of guidance and only optimize locally, we present a task-agnostic contact-implicit trajectory optimization framework that works with minimal guidance, without any task-specific initialization nor references. We also present a novel contact-implicit trajectory optimization formulation that gives us a direct handle on the contact modes and is a crucial tool in overcoming the sim2real gap.

Index Terms—contact-implicit, trajectory optimization, motion planning

I. INTRODUCTION

Legged loco-manipulation systems demonstrate great potential in removing humans from dull, dirty and dangerous environments. However, their capabilities are currently limited to conservative walking and careful interaction with the environment. The next level of applications we are looking to unlock includes physically intensive tasks that require human level athleticism, which pose a challenge in contrast to the current tasks most robots are designed for, like inspection and surveillance. The execution of such tasks by robotic systems will require long horizon contact-exploiting planning to successfully complete the task instead of instantaneous reactive thinking.

II. PRIMARY CONTRIBUTIONS

We present a task-agnostic contact-implicit trajectory optimization framework that automatically discovers the mode schedule and generates motion plans that are realizable on hardware, given only a terminal goal. Our framework works with minimal guidance as the only task-specific input is a high-level goal provided by the user without any task-specific initialization nor references.

Our work builds on current contact-implicit approaches through a three-fold contribution:

- We present a continuation method for contact-implicit trajectory optimization that works with minimal guidance

and without handcrafted initialization nor references. Our framework gradually introduces the contact physics and constraints eliminating the need to handcraft references and a good initial guess. This allows us to automatically discover motions and contact schedules given only a user-specified goal at the end of the task horizon.

- We present a novel formulation for contact-implicit trajectory optimization (TO) that allows us to explicitly encode constraints or preferences on contact modes, without the combinatorial complexity of mixed-integer programming. In contrast to prior work on contact-implicit TO by Posa et al. [1], our relaxed Mixed-Integer formulation with additional relaxed orthogonality constraints enables us to encode constraints and costs directly on the contact modes. We demonstrate its use on eliminating unrealistic gaits from the returned solutions to achieve a minimum sim2real gap.
- We test the motion plans generated by our framework on hardware for a complex, under-actuated legged system.

III. RELATED WORK

A. Control for complex, long-horizon tasks

Long horizon, contact-rich tasks on legged systems remain a challenge for both learning-based and optimization based approaches. Such tasks are a challenge for optimization-based methods that optimize through contact because discontinuities or stiff dynamics arising from contact phenomena lead to a numerically challenging optimization landscape. The current state of the art in optimization-based methods for athletic loco-manipulation tasks by Sleiman et al. [2] avoids optimizing through contact by pre-defining the mode schedule of the feet contacts. Current approaches that optimize through contact bypass this challenge by only optimizing locally such that the solver does not have to overcome problematic areas such as bad local minima or flat regions to reach a good solution. This approach often includes handcrafting references for complementarity-constrained variables [3], and/or providing a good initial guess that place the solver in a desired basin of attraction [1].

Zero-order methods, such as RL, are not prone to issues with jagged optimization landscapes as they do not use gradient information and have shown impressive results in contact-rich tasks. However, zero-order methods over large action spaces suffer from sample complexity. In addition,

in the context of long horizon tasks on legged systems, sparse rewards and sensitive dynamics lead to uninformative signals. Hence, approaches in this class of methods often require extensive guidance in the form of reward shaping [4] or demonstrations [5] to guide the search to a desired subspace of solutions, which requires extensive task-specific engineering effort.

The success of RL in through-contact optimization has already hinted that stochasticity and zero order optimization are beneficial in the presence of discontinuities or stiff dynamics which inspired researchers to employ a combination of noise injection, sampling and interpolation schemes [6, 7] to optimize over the non-smooth landscapes. However, in the context of contact-implicit trajectory optimization, the discontinuous contact physics appear as constraints on the decision variables. Hence, it adds unnecessary complexity to use stochasticity or interpolation schemes to optimize over a non-smooth optimization landscape, while we can easily smoothen the optimization landscape using standard penalty methods.

B. Contact-implicit trajectory optimization

To the best of our knowledge, it’s not obvious yet how learning-based methods can be improved to work without reward shaping or expert demonstrations. Thus, our work focuses on optimization-based methods, namely contact-implicit trajectory optimization. Contact-implicit approaches have had the most success in the form of constrained optimization over continuous decision variables, where the contact physics are encoded as constraints on the decision variables. In contrast to Mixed-Integer Programming approaches [8, 9], and hybrid dynamics methods, the advantage of this formulation is that it does not suffer from combinatorial mode complexity. In addition, they offer the benefit of automatically discovering the mode sequence and number of steps, in contrast to phase-based parametrizations [10].

Contact-implicit approaches that rely on the hard contact model lead to a non-convex Non-Linear Program (NLP) with stiff contact physics encoded through the complementarity constraints, which create an optimization landscape with flat gradients, and infeasible or other ‘bad’ local minima. Hence, existing approaches in this category require a handcrafted initialization and/or references to avoid running into infeasibility issues [1, 11]. In our work, through smoothening the optimization landscape during the early stages of the optimization, we smooth out ‘bad’ and infeasible local minima and can reach a ‘good’ and feasible basin of attraction that would have otherwise required a good initial guess to reach.

Similar work to ours is prior work by Mordatch et al. [12], where they presented results on contact-rich tasks using soft

constraints to encode the contact physics and thus render the optimization landscape smooth for gradient descent. However, they only presented results in simulation as their generated motion plans lack physical accuracy. Our work builds on this work by integrating an analogous formulation in the discovery stage, and augmenting it with a subsequent refinement stage that enforces physical accuracy.

C. Warm-starting TO through soft-contact relaxation

Marcucci et al. [13] present a continuation method where the relaxation relies on relaxing the contact parameters, on using a coarser discretization and on using an approximate pseudo-static dynamic model. Similarly, Suh and Wang [14] study the effectiveness of warm-starting physically accurate methods with methods that rely on a relaxed approximation of the contact model for a toy problem. Overall, Marcucci et al. [13] and Suh and Wang [14] are the closest work to ours, but they rely on the soft contact model (also called penalty-based contact model) originally proposed in [15] which lacks physical accuracy as it models contact interactions with a spring(-damper) model. They also only presented results on simple 2D systems which lack the complexity of legged robotic systems and the generated insights (such as about the effectiveness of single shooting) are not expected to share the same success in highly non-linear and high-dimensional systems. We build on their work as we use their generated insights to our advantage. Specifically, we choose the augmented over the embedded scheme including contact forces as explicit decision variables. The idea of using a coarse discretization grid in the first stage is also present in our work, but we employ splines to further improve computational efficiency without vastly sacrificing expressiveness. We also emulate the relaxation of the contact parameters through a soft-to-hard constraint strategy as there are no such parameters that explicitly determine the stiffness in the hard contact model. In contrast, we use a hard contact model that is physically accurate and has shown to transfer successfully to complex robotic systems in the real world [1], and present results on hardware.

IV. MODELLING

A. Complementarity Constraints

Our method relies on a hard contact model where the contact forces are included as continuous decision variables, and the contact physics are encoded as constraints. In the scope of our work, we explicitly do not allow sliding, so our formulation consists of two complementarity constraints: a separation and a sliding complementarity constraint. The separation constraint encodes that there must be no ground penetration, no forces can be applied at a distance and that the forces are unilateral. The sliding constraint encodes that no force can be applied when the tangential velocity of the end-effector is non-zero.

B. Single Rigid Body (SRB) Model

Optimization through contact is governed by the aforementioned complementarity constraints. These constraints are the most numerically challenging component in our formulation and live in Cartesian space. We chose the SRB model as it also lives in Cartesian space, and we augment it with the end-effector positions and velocities, analogous to prior work by Winkler et al. [10].

1) *State variables*: The state consists of the base pose and twist, and end-effector positions and velocities. The base pose includes the CoM position, denoted by \mathbf{c} , and orientation expressed as ZYX Euler angles, denoted by $\boldsymbol{\theta}$. The base twist includes the CoM velocity, denoted by $\dot{\mathbf{c}}$, and the angular velocity, denoted by $\boldsymbol{\omega}$. The position and velocity for end-effector i are denoted by \mathbf{r}_i and $\dot{\mathbf{r}}_i$, respectively. All quantities are expressed in the world frame. We use the following convention for the order of the end-effectors: left hind, right hind, left front, right front.

$$\mathbf{x} = [\mathbf{c} \ \boldsymbol{\theta} \ \dot{\mathbf{c}} \ \boldsymbol{\omega} \ \mathbf{r}_1 \ \dots \ \mathbf{r}_{n_c} \ \dot{\mathbf{r}}_1 \ \dots \ \dot{\mathbf{r}}_{n_c}] \quad (1)$$

2) *Input variables*: The input consists of the contact forces at each of the n_c end-effectors, denoted by \mathbf{f}_i , as well as the end-effector accelerations, denoted by $\ddot{\mathbf{r}}_i$. We do not consider the torques at the end-effectors, under the assumption of point contacts.

$$\mathbf{u} = [\mathbf{f}_1 \ \dots \ \mathbf{f}_{n_c} \ \ddot{\mathbf{r}}_1 \ \dots \ \ddot{\mathbf{r}}_{n_c}] \quad (2)$$

3) *Dynamics*: The SRB model is governed by the SRB centroidal dynamics. \mathbf{I} denotes the rotational inertia, which we assume it's constant by using its value at a nominal configuration. $\mathbf{T}(\boldsymbol{\theta})$ denotes the Jacobian that transforms the angular velocity to the time-derivative of the ZYX Euler angles. ${}_{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$ denotes the rotation matrix of the base in the world frame.

$$\dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\theta}) \boldsymbol{\omega} \quad (3)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}_{\mathcal{W}}^{-1}(\boldsymbol{\theta}) \left(\sum_{i=1}^{n_c} \mathbf{f}_i \times (\mathbf{r}_i - \mathbf{c}) - \boldsymbol{\omega} \times \mathbf{I}_{\mathcal{W}}(\boldsymbol{\theta}) \boldsymbol{\omega} \right) \quad (4)$$

$$\ddot{\mathbf{c}} = \frac{1}{m} \sum_{i=1}^{n_c} (\mathbf{f}_i + m\mathbf{g}) \quad (5)$$

where

$$\mathbf{I}_{\mathcal{W}}(\boldsymbol{\theta}) = {}_{\mathcal{W}}\mathbf{R}_{\mathcal{B}}(\boldsymbol{\theta}) \mathbf{I}_{\mathcal{B}} {}_{\mathcal{W}}\mathbf{R}_{\mathcal{B}}(\boldsymbol{\theta})^T \quad (6)$$

V. METHOD

We propose a framework that consists of a motion discovery (MD) and a motion refinement (MR) module. Given a high-level goal provided by the user, the MD module gradually introduces the physics and nonlinear constraints while the MR module enforces strict physical feasibility to ensure a minimum sim2real gap. The first module, MD, is

allowed to break the laws of physics and violate the nonlinear constraints at intermediate stages. Thus, the MD stage doesn't require a good initial guess. Since the solution of the MD module is then used as an initialization to the MR module, our framework works with minimal guidance. For example, for the task of local navigation and locomotion, we only need to provide a goal (e.g., a desired x-y position and yaw) at the end of the task horizon. This eliminates the need for the human expert to handcraft reference trajectories and an initial guess.

The role of the motion refinement stage is to ensure a minimum sim2real gap. On one hand, it enforces strict physical feasibility. On the other hand, it allows us to explicitly penalize unrealistically fast contact switches and enforce any other constraints on the contact modes, producing realistic motion plans. We achieve this through a novel relaxed Mixed-Integer formulation. Having a direct handle on the contact mode decision variables, our method allows us to exclude unrealistic contact modes from our search space, leading to motion plans that are realizable on hardware despite the highly non-convex optimization landscape. We validate the quality and strict physical accuracy of the motion plans generated by our framework by tracking the generated trajectory with a whole-body tracking controller both in simulation and on hardware. Even though we present results in legged locomotion only, our framework is task-agnostic: given a corresponding Cartesian space model, it can be used for any contact-rich task on any robotic platform such as legged locomotion or loco-manipulation.

Both modules consist of an Optimal Control Problem (OCP). The goal in each stage is to find a trajectory that satisfies a set of constraints while minimizing a specified objective. We have selected simultaneous methods for the transcription of the OCP of both modules, as including the state variables as decision variables leads to better numerical conditioning with respect to state and state-input constraints (such as the complementarity constraints). We use the CasADi MATLAB library [16] to solve the NLPs as an interface to IPOPT [17].

A. Motion Discovery Module

The OCP of the MD module is transcribed using direct Hermite-Simpson collocation, where the state and input trajectories are parametrized by cubic Hermite splines. We chose spline parametrization for computational efficiency - its higher order of accuracy allows for a reduced number of knot points. In this module, the contact physics are only softly enforced to prevent any convergence issues.

1) *Objective*: The objective function, $L_{MD}(X, U, \dot{U})$, is decomposed to a stage cost and a terminal cost. The stage cost is linear combination of a regularization cost, L_{reg} , a contact physics violation cost, L_{CP} , and a ground clearance cost,

L_{GC} , each term weighted by a corresponding weight, w_s :

$$L_{MD}(X, U, \dot{U}) = L_f(\mathbf{x}(N)) + \int_0^T \sum_s w_s L_s(\mathbf{x}(t), \mathbf{u}(t), \dot{\mathbf{u}}(t)) dt \quad (7)$$

$$L_f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_{ref}\|_{Q_f}^2 \quad (8)$$

$$L_{reg}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}) = \|\mathbf{x} - \mathbf{x}_{ref}\|_Q^2 + \|\mathbf{u}\|_R^2 + \|\dot{\mathbf{u}}\|_{RR}^2 \quad (9)$$

$$L_{CP}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}) = \sum_{i=1}^{n_c} g(f_i^N(\mathbf{u})) (\phi_i(\mathbf{x})^2 + \|\mathbf{v}_i^T(\mathbf{x})\|_2^2) \quad (10)$$

$$L_{GC}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}) = \sum_{i=1}^{n_c} g(-f_i^N(\mathbf{u})) (\hat{B}(\phi_i(\mathbf{x}) - h_{nominal})) \quad (11)$$

where $s \in \{reg, CP, GC\}$. $h_{nominal}$ denotes the desired swing height. $\phi_i(\mathbf{x})$, $f_i^N(\mathbf{u})$, and $f_i^T(\mathbf{u})$ denote the gap function, and the normal and tangential components of the contact force of the i th end-effector respectively. The tangential velocity of the i th end-effector is denoted by $\mathbf{v}_i^T(\mathbf{x})$. We use $\hat{B}(\cdot)$ to denote the relaxed log barrier function.

The complementarity constraints that encode (our approximate model of) the contact physics appear in the objective through the L_{CP} term as they are enforced as soft constraints using a quadratic penalty. Similar to [18], the contact modes are not independent decision variables but are a function of the normal contact forces. Contact activation is defined as a smooth step function of the normal contact force, which is used to softly enforce the orthogonality component of the complementarity constraints. The smooth step function is denoted by $g(\cdot)$, and in our implementation we use the following transformation of the tanh function:

$$g(x) = 0.5 + 0.5 \tanh(x/a - b) \quad (12)$$

2) *Constraints*: The dynamics are enforced through the derivative method, i.e., as an equality constraint between the dynamics evaluated at a collocation point and the slope of the state spline (13). We enforce the initial condition as a constraint (14), a unilateral and a maximum limit constraint on the normal component of the contact force $f_i^N(\mathbf{u})$ (16), and a non-penetration constraint on the gap function (18). As a proxy for actuation bounds, we enforce a limit on the end-effector velocities (17). In addition, we constrain the contact force to lie in an outer linear approximation of the friction cone (15). We also enforce a terminal constraint on the state that encodes the user-specified command (20). This constraint is constructed by specifying a box constraint around the commanded x-y position (and yaw if applicable). We can thus guarantee that the returned motion plan reaches the commanded state (within a specific tolerance). Furthermore, we enforce a box constraint on the end-effector positions in the base frame, abstracted in notation by a non-linear constraint $\Psi(\mathbf{x})$ (19).

In addition, we have found it necessary to divide the motion discovery phase into two stages, where in the first preliminary stage the nonlinear constraints are also enforced as soft constraints with quadratic penalty and the limits ($f_{N,limit}$ and v_{limit}) are relaxed to larger values. The solution of this stage is then used as initialization for the main stage where the nonlinear constraints are hard constraints and the actual limits are used.

$$\dot{\mathbf{x}}(t) = f_{CT}(\mathbf{x}(t), \mathbf{u}(t)) \quad (13)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (14)$$

$$\|\mathbf{f}_i^T(\mathbf{u}(t))\|_\infty \leq \mu f_i^N(\mathbf{u}(t)) \quad (15)$$

$$0 \leq f_i^N(\mathbf{u}(t)) \leq f_{N,limit} \quad (16)$$

$$\|\mathbf{v}_i(\mathbf{x})\|_\infty \leq v_{limit} \quad (17)$$

$$0 \leq \phi_i(\mathbf{x}(t)) \quad (18)$$

$$\mathbf{x}(t) \in \Psi(\mathbf{x}(t)) \quad (19)$$

$$\mathbf{A}_{x_N} \mathbf{x}(T) \leq \mathbf{b}_{x_N} \quad (20)$$

3) Direct Collocation:

$$\begin{aligned} \min_{X, U, \dot{U}} \quad & L_{MD}(X, U) \\ \text{s.t.} \quad & (13) - (20) \end{aligned}$$

The decision variables of the MD OCP are X , U and \dot{U} . The MD OCP is transcribed to a constrained NLP using Hermite-Simpson direct collocation. The objective function is integrated using Simpson quadrature and all state and input decision variables are parametrized by Cubic Hermite splines. The cubic splines are fully defined by the knot points and the slopes at those knot points. The slopes for the splines of the state variables are constrained to match the dynamics at the knot points, while for the splines of the input variables they are part of the decision variables (\dot{U}).

4) *Initialization*: Our goal was to just use a zero initialization perturbed with Gaussian noise. However, such an initialization does not lead to a desirable gait as it can lead to a gait that includes long swing phases and fast contact switches. To bias the solution towards a regular gait, we provide an initialization that includes a periodic pattern. Our framework does not require to carefully handcraft a good initial guess specific to each task: the same initialization scheme is used for all tasks. Current results show that our planner produces a solution that does not resemble the initialization provided. For example, in the task presented the gait differs by 27.5%.

The initialization is crudely constructed by alternating between positive and negative slopes at the knot points of the normal contact force and the normal component of the end-effector positions. This pattern is shifted by one knot point between the diagonal pairs of end-effectors. For the

remaining terms, we use zeros except for the terms included in the user command (such as x-y CoM position) where we use a linear interpolation between the initial and commanded values. In addition, the aforementioned initialization scheme leads to an initialization that violates the contact physics, and thus the initialization need not be strictly feasible.

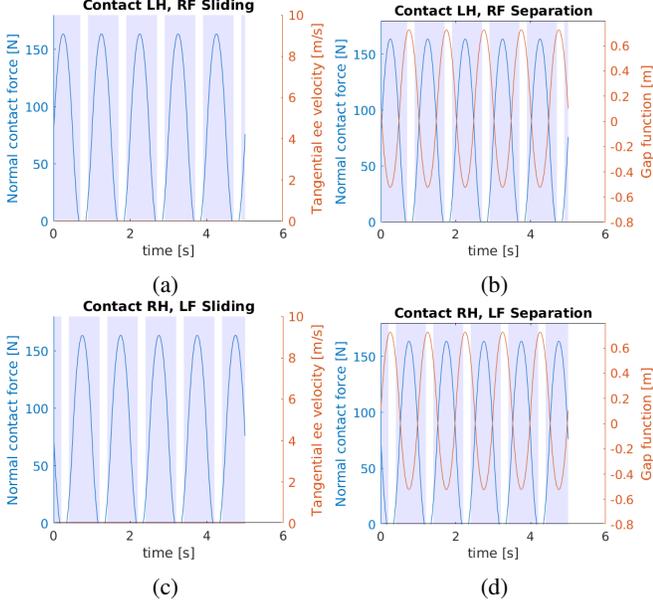


Fig. 1: Initialization used for the complementarity constrained variables of the LH-RF (a-b) and RH-LF (c-d) end-effectors. Inspired from a trot gait, the periodic pattern in the splines is shifted by one knot point for the two diagonal pairs of end-effectors (LH-RF and RH-LF).

B. Motion Refinement Module

We formulate the OCP of the MR Stage as a relaxed Mixed-Integer program, with relaxed integer variables, \mathbf{z} , to encode the contact modes. Specifically, we relax the boolean constraint $z_i \in \{0, 1\}$ to its convex hull $z_i \in [0, 1]$. As a result, our method optimizes over continuous decision variables and thus does not suffer from the combinatorial complexity of mixed-integer programming. We additionally enforce an epsilon-relaxed orthogonality constraint to recover the orthogonality from the original boolean constraint. Subsequently, our formulation is not relaxed (assuming a zero epsilon) as in standard relaxed Mixed-Integer formulations and can thus achieve strict physical accuracy.

Overall, the major advantage of our formulation is that in, in contrast to prior work on contact-implicit TO by Posa et al. [1], we can encode constraints and costs on the contact modes. Specifically, we found it necessary to exclude or penalize gaits with unrealistically fast contact switches. This is needed as when optimizing over the gait, it is not possible to encode phase-dependent costs or constraints. Hence, we

are not able to enforce ground clearance constraints during swing, as commonly done in multi-phase formulations [19], to avoid foot scuffing. In absence of these constraints, the returned motion plan can include a gait with very short swing phases, which we found to often be the case in our experiments with the state-of-the-art contact-implicit TO formulation [1].

1) *Objective*: The objective function, $L_{MR}(X, U, Z)$, is decomposed to a stage cost and a terminal cost. The stage cost is a sum of a regularization cost, L_{reg} , and a cost on contact switches, L_{FC} :

$$L_{MR}(X, U, Z) = L_f(\mathbf{x}(N)) + dt \sum_{k=0}^{N-1} L_{reg}(\mathbf{x}(k), \mathbf{u}(k)) + L_{FC}(\mathbf{z}(k), \mathbf{z}(k+1)) \quad (21)$$

$$L_f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_{ref}\|_{Q_f}^2 \quad (22)$$

$$L_{FC}(\mathbf{z}, \mathbf{z}') = \|\mathbf{z}' - \mathbf{z}\|_P^2 \quad (23)$$

2) *Constraints*: The dynamics are enforced through the integral method using forward Euler. The initial condition (25), friction cone (26), terminal (34) and kinematic constraints (32) are equivalent to the corresponding ones of the MD module. We encode the complementarity constraints through the big-M formulation, where M_f , M_ϕ and M_v denote the big-M for the normal component of the contact force, for the gap function and for the tangential velocity of the end-effector respectively. The indexing of the complementarity constrained variables is important as we use the time-stepping integration scheme [20] to deal with contact impulses. In absence of any additional constraints, this formulation resembles a relaxed Mixed-Integer NLP which suffers from physical inaccuracy as it relaxes the boolean constraint to its convex hull (30). In our formulation we enforce an additional constraint (31) to recover the boolean constraint that encodes the orthogonality between the pairs of complementarity constrained variables. Furthermore, we leverage the direct handle on the contact mode of each end-effector to enforce stance at start and end of motion, penalize fast contact switches and require at least 2 feet to be in contact at all times. (33) denotes the collection of all the linear (equality and inequality) constraints on the contact modes.

$$\mathbf{x}(k+1) = f_{DT}(\mathbf{x}(k), \mathbf{u}(k)) \quad (24)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (25)$$

$$\|\mathbf{f}_i^T(\mathbf{u}(k))\|_\infty \leq \mu f_i^N(\mathbf{u}(k)) \quad (26)$$

$$0 \leq f_i^N(\mathbf{u}(k)) \leq M_f z_i(k) \quad (27)$$

$$0 \leq \phi_i(\mathbf{x}(k+1)) \leq M_\phi (1 - z_i(k)) \quad (28)$$

$$\|\mathbf{v}_i^T(\mathbf{x}(k+1))\|_\infty \leq M_v (1 - z_i(k)) \quad (29)$$

$$\mathbf{z}(k) \in [0, 1] \quad (30)$$

$$\mathbf{z}_i(k)^T \cdot (1 - z_i(k)) \leq \varepsilon \quad (31)$$

$$\mathbf{x}(k) \in \Psi(\mathbf{x}(k)) \quad (32)$$

$$\mathbf{A}_{z,k}\mathbf{z}(k) \leq \mathbf{b}_{z,k} \quad (33)$$

$$\mathbf{A}_{x_N}\mathbf{x}(N) \leq \mathbf{b}_{x_N} \quad (34)$$

3) *Direct Transcription:*

$$\begin{aligned} \min_{X,U,Z} \quad & L_{MR}(X,U,Z) \\ \text{s.t.} \quad & (24) - (34) \end{aligned}$$

The decision variables of the MR OCP are X , U and Z . The MR OCP is transcribed into a constrained NLP using direct transcription, where the input variables and the state variables are approximated with piecewise constant and piecewise linear trajectories respectively. The objective is integrated using forward Euler.

4) *Initialization:* X and U are initialized from the solution of the MD module by interpolating the splines of the respective variables. Z is initialized with a matrix of 0.5's, as otherwise a heuristic would have been needed to extract the contact modes from the solution of the MD module.

C. Tracking Controller

The task-space references (base position (x,y) and base yaw) generated by our contact-implicit planner must then be tracked by a tracking controller that operates in joint-space. This can be achieved by any type of controller, learning- or optimization-based. In this work, we use a whole-body planning framework [2] together with a whole-body controller, where the former generates and the latter tracks joint positions and velocities, and contact forces. In the planning framework, we pre-define the gait using the solution of the integer variables of the MR module, which enables us to use efficient optimal control methods for switched systems [21]. For the whole-body controller, we use the hierarchical QP-based framework from [22].

We also found it beneficial to include a polishing step to refine the trajectories generated by the contact-implicit optimization prior to tracking them online. We expect that a transition to a learning-based tracking controller will deem this step unnecessary. The polishing step is formulated as a hybrid optimization that includes the full centroidal dynamics and the kinematics of the robot, in contrast to the contact-implicit planner that relies on a SRB model. It thus leads to trajectories that are of higher quality and have higher physical accuracy.

In this step, constraints are enforced such that the force vectors lie in the friction cone, forces are zero during swing, sliding is not allowed, gap function is zero during stance and that the desired foothold positions are tracked during stance. The end-effector positions are also constrained in the normal direction during the swing phase to achieve the necessary ground clearance. Since accuracy is of higher importance than computational efficiency in this step, we use a high order integration method (RK4). The polishing step is transcribed using multiple shooting, and solved using SQP.

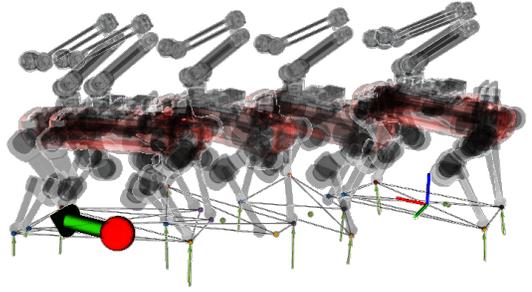


Fig. 2: XY position command.

Both the polishing step and the MPC planning framework are implemented using the OCS2 framework [21].

VI. RESULTS

A. Local Navigation and Locomotion

We have tested our framework on a range of motions for the task of local navigation and locomotion, analogous to the task considered by Rudin et al. [23]. Given only a high-level goal, our planner generates reference trajectories for the base and end-effectors (from which contact locations can also be extracted) and the mode schedule. The goal is specified by the user, and it can include x -position, y -position, and/or yaw orientation for the base.

In the examples illustrated in Figures 2 and 3, the foothold locations and the changes in yaw emerge from the planner and neither any references nor the initialization were modified to hint that. The red spheres illustrate the position command and the green arrows the yaw orientation command.

B. Sim2real gap

Physical feasibility is gradually introduced to prevent any feasibility or convergence issues earlier on in the pipeline. The motion plan generated by the MD module is not strictly feasible, but strict feasibility is enforced by the MR module. As also seen in Figure 4, both complementarity constraints are satisfied strictly in the final solution. We validated the physical feasibility of the generated motion plans by also executing them on hardware (Figure 5). Phases with less than 2 feet in contact proved to be a source of failure in executing the generated motion plans in the physical simulator, which we addressed by modifying constraint (33) accordingly.

VII. CONCLUSION

Overall, our framework is able to generate contact-rich motions that are realizable on hardware given a non-informative initialization. We exploit the contact-implicit TO formulation that gives us direct access to the source of discontinuities, the complementarity constraints, to smoothen the optimization landscape. We thus eliminate the need for

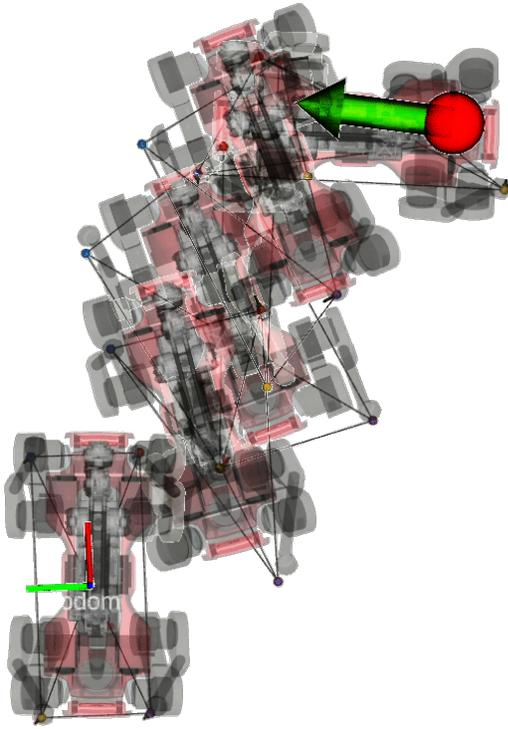


Fig. 3: XY position, and yaw orientation command.

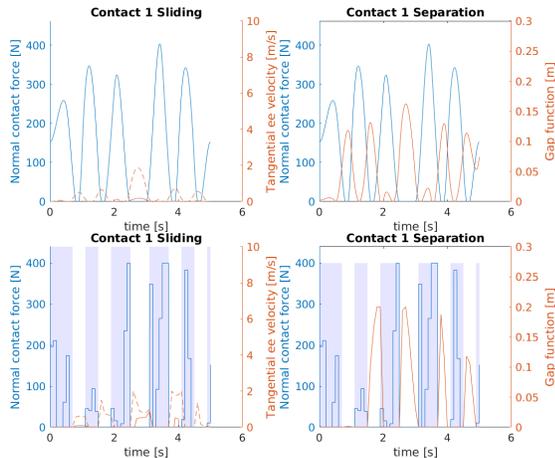


Fig. 4: Feasibility of complementarity constraints in solution from (Top) MD module and (Bottom) MR module.

a human expert to guide the solver to a desired basin of attraction through handcrafted references or initialization. In addition, we present a novel formulation for contact-implicit TO that gives us a direct handle on the contact modes for the primary purpose of eliminating unrealistic gaits from the space of feasible solutions. As a result, the final output of our continuation method are motion plans that have strict physical accuracy and gaits that are realizable on hardware.

The current bottleneck of our pipeline is the multi-phase



Fig. 5: Generated motions were realizable on hardware.

planning framework which is sensitive to any fast contact switches in the generated mode schedule, as well as to any dynamic footholds in the generated references. We expect a learning-based controller to overcome these limitations and enable us to execute more dynamic motions. Finally, we will be exploring the application of the framework to legged loco-manipulation. Since the workspace of the arm can not be approximated with task-space constraints easily, we will investigate including the full kinematics to replace the proxy kinematic constraints, similar to [24].

REFERENCES

- [1] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *International Journal of Robotics Research*, vol. 33, pp. 69–81, 2014.
- [2] J. P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, pp. 4688–4695, 7 2021.
- [3] A. Aydinoglu, A. Wei, and M. Posa, "Consensus complementarity control for multi-contact mpc," 4 2023.
- [4] Y. Ji, G. B. Margolis, and P. Agrawal, "Reinforcement learning for quadrupedal dribbling in the wild."
- [5] X. Xiao, T. Zhang, K. Choromanski, E. Lee, A. Francis, J. Varley, S. Tu, S. Singh, P. Xu, F. Xia, S. M. Persson, D. Kalashnikov, L. Takayama, R. Frostig, J. Tan, C. Parada, and V. Sindhvani, "Learning model predictive controllers with real-time attention for real-world navigation," 9 2022.
- [6] H. J. T. Suh, T. Pang, and R. Tedrake, "Bundled gradients through contact via randomized smoothing," 9 2021.
- [7] H. J. T. Suh, M. Simchowit, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" 2 2022.
- [8] A. K. Valenzuela, "Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain," 2016.

- [9] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Febru, pp. 279–286, 2015.
- [10] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1560–1567, 2018.
- [11] Y. Fuchioka, Z. Xie, and M. van de Panne, "Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors," 10 2022.
- [12] I. Mordatch, E. Todorov, and Z. P. Popović, "Discovery of complex behaviors through contact-invariant optimization."
- [13] T. Marcucci, M. Gabiccini, and A. Artoni, "A two-stage trajectory optimization strategy for articulated bodies with unscheduled contact sequences," *IEEE Robotics and Automation Letters*, vol. 2, pp. 104–111, 1 2017.
- [14] H. J. T. Suh and Y. Wang, "Comparing effectiveness of relaxation methods for warm starting trajectory optimization through soft contact."
- [15] M. Gabiccini, A. Artoni, G. Pannocchia, and J. Gillis, "A computational framework for environment-aware robotic manipulation planning."
- [16] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 3 2019.
- [17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 5 2006.
- [18] I. Mordatch, J. M. Wang, E. Todorov, and V. Koltun, "Animating human lower limbs using contact-invariant optimization," *ACM Transactions on Graphics*, vol. 32, 11 2013.
- [19] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *arXiv preprint arXiv:2211.11644*, 2022.
- [20] D. Stewart and J. C. Trinkle, "Implicit time-stepping scheme for rigid body dynamics with coulomb friction," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 162–169, 2000.
- [21] F. Farshidian *et al.*, "OCS2: An open source library for optimal control of switched systems," [Online]. Available: <https://github.com/leggedrobotics/ocs2>.
- [22] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization." IEEE Computer Society, 12 2016, pp. 558–564.
- [23] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," 9 2022.
- [24] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," vol. 2015-February. IEEE Computer Society, 2 2015, pp. 295–302.