Multi-Fidelity Receding Horizon Planning for Multi-Contact Locomotion

Multi-Fidelity Receding Horizon Planning for Multi-Contact Locomotion

Jiayi Wang [1] Sanghyun Kim [2] Sethu Vijayakumar [1,3] Steve Tonneau [1]

***Abstract*— When traversing uneven terrain, humans consider their future steps for choosing the best location and timing of their current step. Likewise, when planning multi-contact motions for legged robots (e.g. humanoids), a 'prediction horizon' has to be considered. However, planning several steps ahead increases the dimensionality and non-linearity of an already challenging problem, which makes online planning intractable. We propose to reduce the problem complexity by using convex relaxations in the prediction horizon. We realize this idea within a Receding Horizon Planning (RHP) framework to plan dynamically consistent centroidal trajectories of humanoid walking on uneven terrain. This results in a novel formulation that combines an accurate non-convex model with a relaxed convex model, which we call RHP with multiple levels of model fidelity. We evaluate three candidate multi-fidelity RHPs with convex relaxations of the centroidal dynamics in the prediction horizon. The best candidate is 1.4x-3.0x (average 2.4x) faster than the traditional RHP that employs a single dynamics model over the entire look-ahead horizon. We also validate the resultant centroidal trajectories by tracking them with a whole-body inverse dynamics controller in simulation. Lastly, we find that incorporating angular dynamics in the prediction horizon is important to the success of multi-fidelity RHP.**

I. Introduction

We consider the problem of planning the motion of legged robots (e.g. humanoids) on uneven terrain under the framework of multi-contact motion planning. This problem is high-dimensional, non-linear and subject to discrete changes in the dynamics due to the making and breaking of contacts [1], [2], [3], [4], [5], [6]. Traditionally, multi-contact motions from a start position to a goal are pre-planned offline and then tracked by a controller [2], [7], [8], [9], [10]. However, due to imperfect control, state estimation errors and other unforeseen perturbations that inevitably happen in the real world, pre-planned motions can become invalid during their execution. Hence, online motion re-planning is often required to adapt to these perturbations. To facilitate reliable locomotion in challenging environments, our long term objective is to provide robots with the capability to re-plan online multi-contact motions.

Towards this end, local re-planning strategies—such as Receding Horizon Planning (RHP) [11]—can be a promising approach. Similar to Model Predictive Control (MPC) [12], [13], [14], [15], [16], RHP framework constantly plans online a few steps ahead of the current state and then partially executes the motion plan with a controller. The motion plan is usually computed by solving a finite-horizon Trajectory Optimization (TO) problem [17]. For RHP frameworks, having a look-ahead horizon with sufficient length is critical [18]. This is because the prediction of the future can guide the immediate actions for execution to avoid short-sighted decisions (e.g. causing the robot to fall, unable to traverse gaps or the solver getting stuck in a local minima).

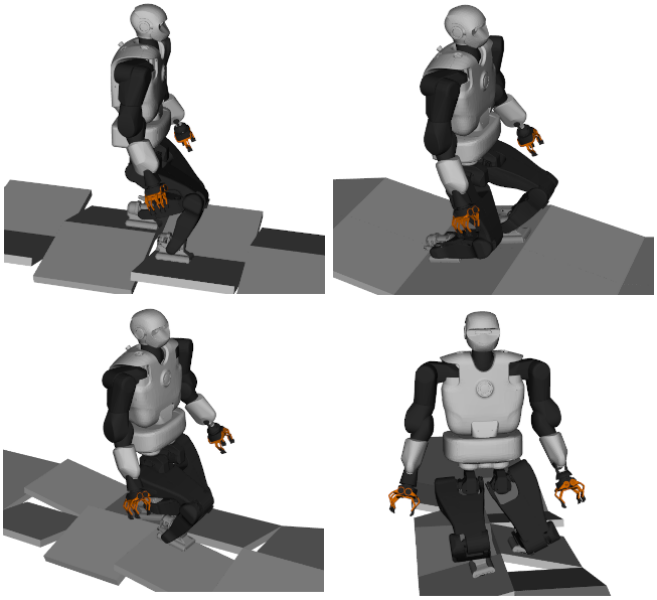


Fig. 1: Snapshots of our simulations (video available in https://youtu.be/3FjKPmnYxvU). The centroidal trajectory is generated from our multi-fidelity RHP framework and then tracked by a whole-body inverse dynamics controller.

RHP usually plans over the entire look-ahead horizon with a single dynamics model that captures the system dynamics as accurately as possible. This results in expensive computation, especially for planning multi-contact motions on uneven terrain, where complex dynamics (usually non-convex) needs to be considered.

To improve the computation efficiency of RHP for multi-contact motions, we explore the possibility to relax the complexity of the model along the look-ahead horizon. We split the look-ahead horizon into execution horizon that considers the first few steps of the motion (to be executed), and prediction horizon that regards the remaining steps of the motion (not executed). Although the execution horizon needs to be planned accurately, we hypothesize that the

[1] The authors are with the School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom.

2 The author is with the department of AI Machinery, System Engineering Research Division, Korea Institute of Machinery & Materials, Daejeon, South Korea.

3 The author is with Artificial Intelligence Programme, The Alan Turing Institute, London, United Kingdom.

e-mail: jiayi.wang@ed.ac.uk

prediction horizon can instead use simplified models to guide the decisions of the execution horizon. This results in a novel type of RHP formulation for planning multi-contact motions on uneven terrain, where appropriate selection of contact locations and timings are critical. In this work, we call this formulation as RHP with Multiple Levels of Model Fidelity.

Recently a similar approach [19] has successfully been demonstrated for quadrupedal locomotion and humanoid running on flat surfaces. Our work focuses on computing centroidal trajectories [20] for the humanoid robot Talos [21] walking on multi-contact scenarios, where the contact surfaces can be non-coplanar. In this context, the quality of the angular dynamics approximation significantly impacts the performances of the framework. We thus propose and compare three candidate multi-fidelity RHPs that each relies on a different approximation of the dynamics. In our experiments, the best candidate is 1.4x-3.0x (average 2.4x) faster than the baseline (traditional single-fidelity RHP). This is achieved by employing a convex relaxation based on [22], [23] in the prediction horizon. Although the optimal cost of the resultant trajectories is higher, we show that these centroidal trajectories can be successfully tracked by a whole-body inverse dynamics controller [24] in simulation.

Our main contribution is the introduction and comparison of several RHP formulations with multiple levels of model fidelity for planning dynamically consistent centroidal trajectories of humanoid walking in challenging multi-contact scenarios. These formulations feature convex relaxation of the system dynamics in the prediction horizon that allow faster computation than the traditional formalism. Besides, our investigation suggests that considering angular dynamics in the prediction horizon is vital to the success of RHP.

II. Related Work

Planning multi-contact motions in challenging environments (i.e. non-horizontal surfaces) necessarily requires the consideration of the whole-body dynamics of the robot. Such models consider the mass and inertia of every link and relates the joint torques to base and joint accelerations. In the past, impressive motions were demonstrated using this model within TO [1], [25], [26], [27], [28], [29], [30]. However, given their high-dimensionality and non-convexity, these approaches are usually computationally challenging for RHP unless contact timings and locations are fixed [12].

An alternative for multi-contact motion planning is to use a centroidal model [20]. The centroidal model is low-dimensional as it only considers the dynamics of the total linear and angular momenta expressed at the Center of Mass (CoM). Further, approximations are introduced on the geometric constraints of the robot and on the momentum variation, which results from the motion of each individual rigid body of the robot. While these approximations can lead to failures to achieve a corresponding whole-body motion, centroidal model is widely used for generating multi-contact motions [31], [32], [33], [34], [35], [36], [37] due to its reduced dimensionality. Unfortunately, the centroidal model is non-convex[1] except when strong limiting assumptions (e.g. pre-defined gait, flat/co-planar surfaces) are made [38], [39], [40], which makes efficient resolution of TO problems challenging.

To improve the computational efficiency, two families of approaches have been proposed. The first family comprises conservative approaches that look for a solution in a subset of all possible trajectories [37], [41], [42]. Although fast computations are demonstrated, these methods usually assume pre-defined contact locations and/or timings, which limits their flexibility. In contrast, our approach only assumes the sequence of the contact surfaces where the swing foot will land on, while we simultaneously optimize centroidal trajectories, contact locations and contact timings.

The second family introduces convex relaxations [22], [23], [43], [8]. Despite the fact that the model complexity is dramatically reduced, a potential issue is that these methods may generate motions with significant violations of the angular dynamics constraint. Although it is possible to gradually tighten the convex relaxation using iterative schemes, this may require the design of customized solvers [22], [23]. In our work, we propose a multi-fidelity RHP approach where, in a single optimization formulation, a high-fidelity model is employed in the execution horizon and a low-fidelity model in the prediction horizon. This formulation is straightforward and can be interfaced directly with off-the-shelf solvers. Also, the combination of the high-fidelity and the low-fidelity models can ensure the dynamic consistency of the motion to be executed, while at the same time facilitate efficient computation.

A similar framework to the one we propose was introduced recently by [19]. The authors present a MPC framework based on Differential Dynamic Programming (DDP) that combines whole-body dynamics and a non-convex model with reduced order (single-rigid body model [2], [34]) in a single formulation. Successful MPC of 2D quadrupedal locomotion and humanoid running has been demonstrated on flat surfaces. In contrast, the emphasis of our work is RHP of centroidal trajectories for uneven-terrain locomotion of humanoid robots. This problem requires careful selection of contact locations and timings to modulate the centroidal momenta. In this regard, the simplified model applied in the prediction horizon need to be carefully designed, as the quality of the model can significantly affect the performance of the framework. Further, instead of searching for non-convex models with reduced order, we focus on finding appropriate convex relaxations for the prediction horizon.

III. Rationale

Traditional RHP (see Fig. 2-a) employs a single-fidelity model over the entire look-ahead horizon; in contrast, our multi-fidelity RHP (Fig. 2-b) aims to obtain a computationally efficient TO formulation by relaxing the dynamic constraints of our model along the look-ahead horizon.

[1]The centroidal model is non-convex due to the bilinear terms that arise from the cross product operation.

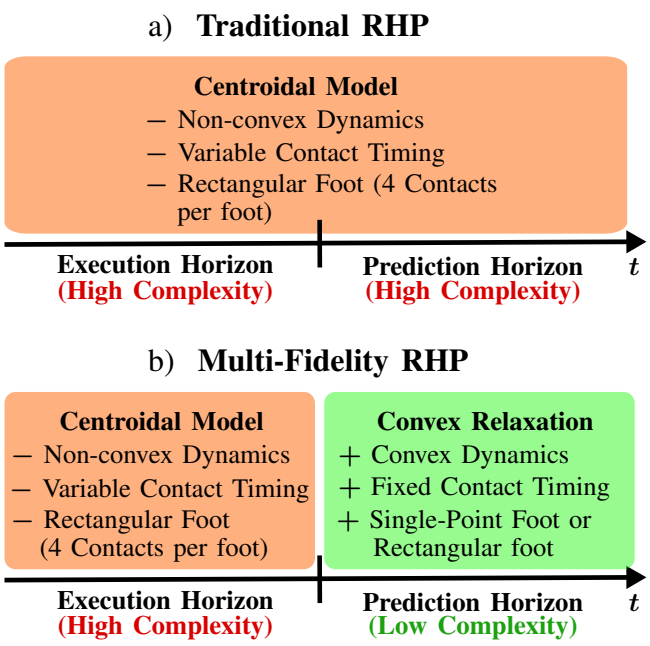


Fig. 2: Complexity comparison between traditional RHP and our multi-fidelity RHP. We use orange to denote higher computation complexity, while green means lower computation complexity. Our multi-fidelity RHP formulation has reduced complexity due to the introduction of convex relaxations in the prediction horizon.

More specifically, we employ an accurate model of the system (in our case, centroidal model) in the execution horizon. This is to ensure that in each RHP cycle, the motion for execution is dynamically consistent. Further, in the prediction horizon, we introduce a convex relaxation of the centroidal model. In this work, we explore and test three convex relaxations. The resultant TO problems are partially non-convex, and can be efficiently solved by off-the-shelf Non-Linear Programming (NLP) solvers. We use the following definitions and assumptions:

- We focus on the humanoid walking, where a step consists of three contact phases: i) Pre-Swing (Double Support) ii) Swing (Single Support) iii) Landing (Double Support). This gives rise to a multi-phase TO formulation, where the active contacts in each phase establish phase-specific constraints (dynamics and kinematics).
- We do not consider impact dynamics (e.g. state jumps when making contacts) in our formulation. This assumption is reasonable for the walking gait, as the feet can have near-zero touch-down velocity, which creates negligible impacts.
- We define the execution horizon to always cover a one-step motion plan, while the prediction horizon can look ahead for multiple steps.
- The environment is modeled as a set of convex surfaces [44]. We pre-specify the sequence of contact surfaces where the swing foot will land upon a priori, but the contact locations (within each surface) and the contact timings are optimized along with the centroidal trajectory.
- The kinematic constraints of the CoM and relative positions of active contacts are approximated as convex polytopes attached to the foot in contact and thus implemented as linear inequalities. These polytopes are computed by the method described in [45].

IV. Receding Horizon Planning with Multiple Levels of Model Fidelity

In this section, we first formulate the problem of finding a feasible centroidal trajectory for a humanoid robot as a TO problem. We then explain how this formulation is adapted in the context of RHP with multiple levels of model fidelity.

Given a finite look-ahead horizon of n steps, the sequence of contact surfaces $\mathcal{S}_1, ..., \mathcal{S}_n$, the current state $\boldsymbol{x}_{cur}$ and a goal position of the robot $\boldsymbol{x}_g$, the TO computes a motion plan divided into N_{ph} phases (described in Section III). The motion plan consists of the following terms[2]: 1) state trajectory $\mathcal{X} = [\boldsymbol{x}^{\{1\}}, ..., \boldsymbol{x}^{\{N_{ph}\}}]^T$ of all phases; 2) control input of all phases $\mathcal{U} = [\boldsymbol{u}^{\{1\}}, ..., \boldsymbol{u}^{\{N_{ph}\}}]^T$; 3) contact locations of all footsteps $\mathcal{P} = [\boldsymbol{p}^{\{1\}}, ..., \boldsymbol{p}^{\{n\}}]^T$ with $\boldsymbol{p} \in \mathbb{R}^3$; 4) phase switching timings $\mathcal{T} = [\tilde{\boldsymbol{t}}^{\{1\}}, ..., \tilde{\boldsymbol{t}}^{\{N_{ph}\}}]^T$. We use superscript $\{q\} \in \{1, ..., N_{ph}\}$ to indicate quantities associated to a phase $\{q\}$. In each phase, the state trajectory and control are further discretized into knots and we use $k \in \{1, ..., N_k\}$ to denote the knot index. The discretized formulation is given by:

$$\min_{\mathcal{X},\mathcal{U},\mathcal{T},\mathcal{P}} \quad \sum_{q=1}^{N_{ph}} J^{\{q\}}(\boldsymbol{x}^{\{q\}}, \boldsymbol{u}^{\{q\}}) + \phi(\boldsymbol{x}_T) \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{x}_0 = \boldsymbol{x}_{cur} \tag{1b}$$

$$0 \leq \tilde{\boldsymbol{t}}^{\{1\}} \leq \cdots \leq \tilde{\boldsymbol{t}}^{\{N_{ph}\}} \leq T_{max} \tag{1c}$$

$$\forall \{i\} \in \{1, ..., n\}:$$

$$\boldsymbol{p}^{\{i\}} \in \mathcal{S}_i \tag{1d}$$

$$\boldsymbol{p}^{\{i\}} \in \mathcal{R}_{\{i-1\}} \tag{1e}$$

$$\forall \{q\} \in \{1, ..., N_{ph}\}, \forall k \in \{1, ..., N_k\}:$$

$$\boldsymbol{c}_k^{\{q\}} \in \mathcal{K}_l, \quad \forall l \in \mathcal{C}^{\{q\}} \tag{1f}$$

$$\boldsymbol{x}_{k+1}^{\{q\}} = \boldsymbol{f}_{dyn}^{\{q\}}(\boldsymbol{x}_k^{\{q\}}, \boldsymbol{u}_k^{\{q\}}), \tag{1g}$$

where $\boldsymbol{c} \in \mathbb{R}^3$ is the CoM position, $\phi(\boldsymbol{x}_T) = (\boldsymbol{x}_T - \boldsymbol{x}_g)^2$ is the terminal cost which encourages the robot to approach a user-defined goal position $\boldsymbol{x}_g$, and $J^{\{q\}}$ is the running cost for each phase which is a function of state and control.

The control input in our formulation is defined as the collection of all contact forces $\boldsymbol{u} = [\boldsymbol{f}_1, ..., \boldsymbol{f}_{N_c}]^T$, where $\boldsymbol{f} \in \mathbb{R}^3$, N_c is the total number of contact points of the robot. The definition of the state vector $\boldsymbol{x}$ in each phase depends on the chosen dynamics model[3].

[2]For simplicity, we do not explicitly write the time dependency of continuous trajectories (i.e. state trajectory and control input)

[3]The state vector for centroidal model is $\boldsymbol{x} = [\boldsymbol{c}, \boldsymbol{L}]^T$, while the state vector of our convex relaxations are $\boldsymbol{x} = [\boldsymbol{c}]^T$.

TABLE I: Knot-wise model complexity of the centroidal dynamics model and the three convex relaxations.

Model \ No. of	Decision variables	Non-convex Constraints	Convex Constraints
Centroidal Dynamics	36	12	0
Convex (CoM only)	18	0	6
Convex (Rectangular Foot)	78	0	48
Convex (Point Foot)	12	0	12

Constraints are introduced in Eq.(1) as follows: (1b) constrains the trajectory to start from the current state x_{cur}; (1c) constrains the phase switching timings to increase monotonically and bounds the maximum duration by T_{max}; (1d) constrains each contact location $\boldsymbol{p}^{\{i\}}$ to a pre-specified contact surface $\mathcal{S}_i = \{\boldsymbol{p} \in \mathbb{R}^3, \boldsymbol{d}_i^T\boldsymbol{p} = e_i, S_i\boldsymbol{p} \leq s_i\}$; (1e) constrains each contact location $\boldsymbol{p}^{\{i\}}$ to stay in the reachable space $\mathcal{R}$ with respect to the other foot $\boldsymbol{p}^{\{i-1\}}$ and (1f) constrains the CoM to stay in the reachable space $\mathcal{K}$ established by each stance foot $l \in \mathcal{C}^{\{q\}}$. These reachable spaces are defined in a matrix form: $\mathcal{R} = \{\boldsymbol{p} \in \mathbb{R}^3, \boldsymbol{R}\boldsymbol{p} \leq \boldsymbol{r}\}$, $\mathcal{K} = \{\boldsymbol{c} \in \mathbb{R}^3, \boldsymbol{K}\boldsymbol{c} \leq \boldsymbol{k}\}$. Lastly, (1g) implements the system dynamics constraints defined by $\boldsymbol{f}_{dyn}^{\{q\}}$, which is realized by one or a combination of the models as described below. Next, we elaborate on the formulation of the dynamics constraints (1g) with multiple levels of model fidelity.

1) Single-Fidelity RHP with Centroidal Dynamics (baseline): The baseline follows the traditional TO paradigm (See Fig. 2-a), in which we enforce centroidal dynamics constraints across the entire look-ahead horizon. The centroidal dynamics constraints in discrete form with forward Euler integration are given by:

$$\boldsymbol{c}_{k+1}^{\{q\}} = \boldsymbol{c}_k^{\{q\}} + \Delta_k^{\{q\}}\dot{\boldsymbol{c}}_k^{\{q\}}, \tag{2a}$$

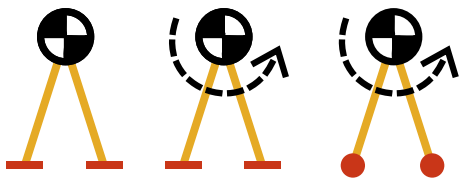
$$\dot{\boldsymbol{c}}_{k+1}^{\{q\}} = \dot{\boldsymbol{c}}_k^{\{q\}} + \Delta_k^{\{q\}}\left(\frac{1}{m}\sum_{c\in\mathcal{C}^{\{q\}}} \boldsymbol{f}_{c,k}^{\{q\}} - \boldsymbol{g}\right), \tag{2b}$$

$$\boldsymbol{L}_{k+1}^{\{q\}} = \boldsymbol{L}_k^{\{q\}} + \Delta_k^{\{q\}}\sum_{c\in\mathcal{C}^{\{q\}}} (\boldsymbol{p}_c - \boldsymbol{c}_k^{\{q\}}) \times \boldsymbol{f}_{c,k}^{\{q\}}, \tag{2c}$$

where m is the total mass, $\boldsymbol{g}$ is the gravitational acceleration, $\boldsymbol{L} \in \mathbb{R}^3$ is the angular momentum, $\boldsymbol{p}_c$ and $\boldsymbol{f}_{c,k}^{\{q\}}$ are the location and force associated to the active contact point c, $\Delta_k^{\{q\}} = (\tilde{\boldsymbol{t}}^{\{q\}} - \tilde{\boldsymbol{t}}^{\{q-1\}})/N_k$ is the time step. The robot feet are rectangular and we use one contact force at each vertex of the rectangle. Contact forces are further constrained by the linearized friction cone $-\mu\boldsymbol{f}_c^{\hat{n}} \leq \boldsymbol{f}_c^{\hat{t}_1,\hat{t}_2} \leq \mu\boldsymbol{f}_c^{\hat{n}}$, where μ is the friction coefficient, $\boldsymbol{f}_c^{\hat{n}}$ and $\boldsymbol{f}_c^{\hat{t}_1,\hat{t}_2}$ respectively denote the normal and tangential components of the contact force $\boldsymbol{f}_c$. Further, to encourage a smooth trajectory, we penalize acceleration and angular momentum in the running cost $J^{\{q\}} = \sum_{k=1}^{N_k}[\Delta_k^{\{q\}}(\ddot{\boldsymbol{c}}_k^{\{q\}})^2 + \Delta_k^{\{q\}}(\boldsymbol{L}_k^{\{q\}})^2]$.

To increase computational speed, we propose to replace the exact centroidal dynamics described above with convex relaxations in the prediction horizon.

2) Linear CoM Dynamics (Candidate 1): Our first candidate only considers the linear CoM dynamics defined by (2a)–(2b). As a result, the state vector reduces to $\boldsymbol{x} = [\boldsymbol{c}]^T$ and the running cost only penalizes acceleration $J_1^{\{q\}} = \sum_{k=1}^{N_k}\Delta_k^{\{q\}}(\ddot{\boldsymbol{c}}_k^{\{q\}})^2$. Further, to remove the non-convexity introduced by time optimization, we simply fix the phase switching time $\tilde{\boldsymbol{t}}^{\{q\}}, \forall\{q\} \in [4, N_{ph}]$ in the prediction horizon.



(a) (b) (c)

Fig. 3: Schematics of models used in the prediction horizon: a) linear CoM dynamics (Candidate 1); b) convex relaxation of angular momentum rate dynamics (dashed arrow) with rectangular contacts (Candidate 2); c) convex relaxation of angular momentum rate dynamics (dashed arrow) with point contacts (Candidate 3).

3) Convex Relaxation of Angular Momentum Rate Dynamics with Rectangular Contacts (Candidate 2): Building upon the first candidate, the second candidate adds a convex outer approximation of the angular momentum rate dynamics based on [22], [23]. Next, we briefly describe the formulation.

In this model, each bilinear term $\alpha\beta$ results from the expansion of the cross product between the lever arm $(\boldsymbol{p}_c - \boldsymbol{c})$ and the contact force vector $\boldsymbol{f}_c$ is reformulated as the difference between two bounded quadratic terms $\psi^+ \in \mathbb{R}$ and $\psi^- \in \mathbb{R}$, along with two convex trust-region constraints as shown below:

$$\alpha\beta = \frac{1}{4}(\psi^+ - \psi^-), \psi^+ \geq (\alpha+\beta)^2, \psi^- \geq (\alpha-\beta)^2 \tag{3}$$

Further, instead of explicitly considering the angular momentum dynamics (2c), we decide to penalize ψ^+ and ψ^- in the running cost to minimize the angular momentum rate, along with the acceleration. An additional benefit is that we avoid explicit modeling of angular momentum and thus retain a low-dimensional model with the state vector of $\boldsymbol{x} = [\boldsymbol{c}]^T$.

4) Convex Relaxation of Angular Momentum Rate Dynamics with Point Contacts (Candidate 3): To further reduce the dimensionality of the TO problem, we propose the third candidate in which we switch the rectangle foot to point foot and apply the same dynamics modeling as in the second candidate. As a result, the control input reduces to $\boldsymbol{u} = [\boldsymbol{f}_L, \boldsymbol{f}_R]^T$, where $\boldsymbol{f}_L$ and $\boldsymbol{f}_R$ refers to the contact force vector of the left and right point foot, respectively. This reduces the number of auxiliary variables (ψ^+ and ψ^-) as well as the associated trust region constraints introduced [22], [23].

To provide an intuition of the complexity of these candidate models, we illustrate their schematics in Fig. 3, and compare their model complexity in terms of dimensionality, number of non-convex and convex constraints in Table I.

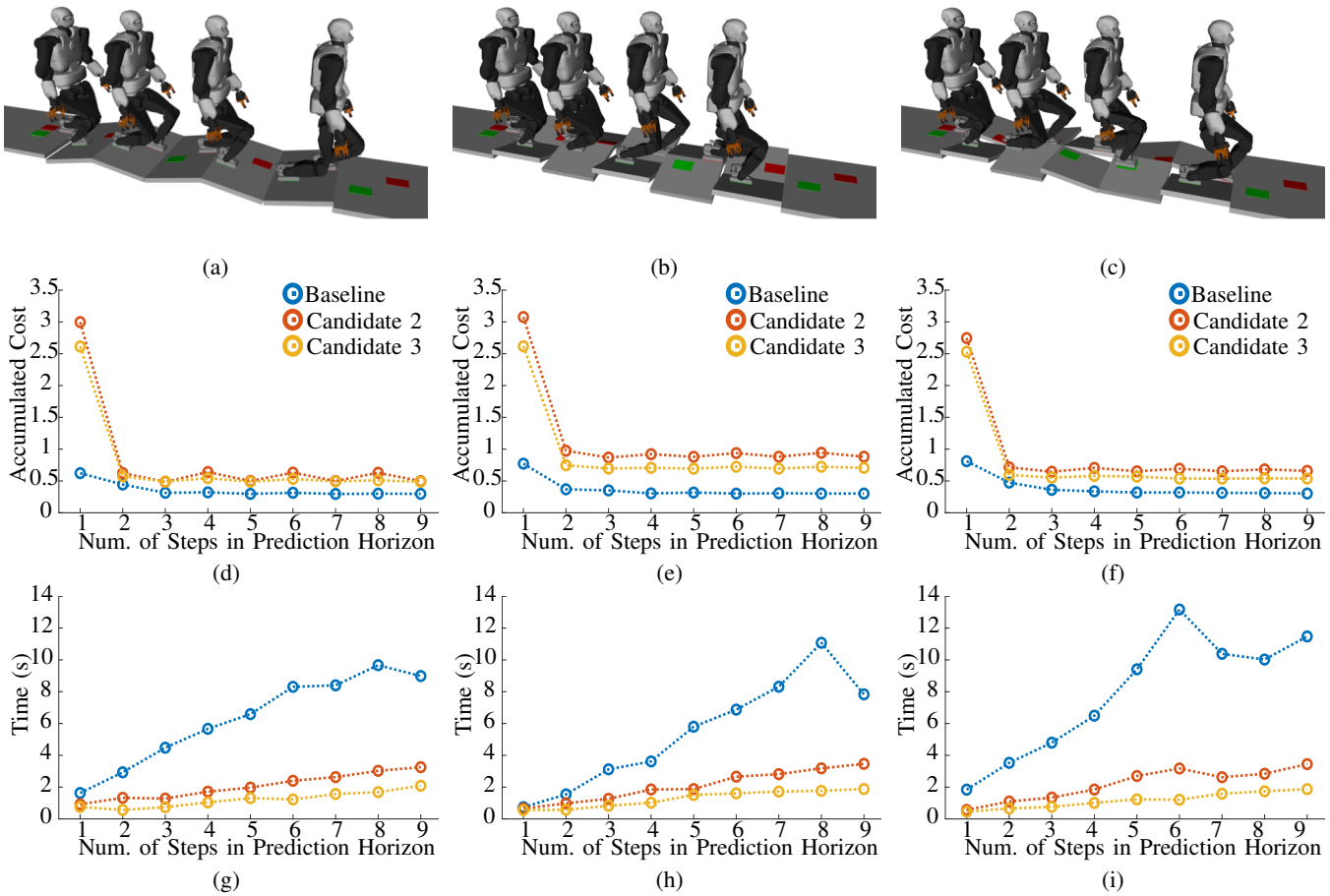


Fig. 4: (a) Scenario 1: up-and-down hill with 11.3° slopes; (b) Scenario 2: V-shape terrain with 13.5° slopes; (c) Scenario 3: Rubbles with 11.3°-13.5° slopes; (d)-(f) Accumulated cost of each RHP for scenario 1-3; (g)-(i) Average Computation time of different RHPs for scenario 1-3.

V. Experiments and Results

To identify and analyze the advantages and the disadvantages of our approach, we compare the performance of our multi-fidelity RHP candidates against the single-fidelity RHP (baseline) over a set of multi-contact scenarios.

A. Experiment Setup

We consider the following multi-contact scenarios: 1) up-and-down hill (Fig. 4a); 2) v-shape terrain (Fig. 4b); 3) rubbles (Fig. 4c). These scenarios feature non-horizontal surfaces with different orientations, which impose challenges in terms of angular momentum modulation. For instance, up-and-down hill and v-shape terrain require special care on the pitch-axis and roll-axis, respectively; while the rubbles need careful momentum modulation around all axes. In each of these scenarios, we use both the single-fidelity RHP (baseline) and our multi-fidelity RHP candidates to plan offline centroidal trajectories of the humanoid robot Talos [21] for multiple planning cycles. Within each planning cycle, a TO problem is solved given the current state of the robot and the first step of the planned motion (execution horizon) is extracted. We validate the dynamic feasibility of these trajectories by tracking them via an inverse dynamics controller [24] in simulation (video: https://youtu.be/3FjKPmnYxvU). The TO problems are implemented in Python and solved by the interior-point algorithm of KNITRO 10.10 [46]. Further, we use the automatic differentiation framework CasADi [47] to provide the gradients and the Hessian. All computations are carried out on a laptop with an Intel Xeon E3-1535M v6 (Maximum 4.20 GHz) processor and 32 GB memory.

TABLE II: Minimum prediction horizon required for success for each RHP framework under different scenarios.

Method	up-and-down hill	v-shape terrain	rubbles
baseline	1	1	1
Candidate 1 (CoM)	Fail	Fail	Fail
Candidate 2 (Rectangle)	2	1	2
Candidate 3 (Point)	2	2	2

B. Performance Comparison

The relevant criteria for performance comparison are:

1) Minimum Prediction Horizon Required: We declare a chosen RHP framework as successful, if the generated

centroidal trajectory can be tracked by the whole-body controller in our simulation. Table II lists the minimum prediction horizon (PH)[4] required for a RHP framework to be successful. We recall here that the length of PH is defined by the total number of footsteps included.

Among all considered RHP frameworks, the candidate 1 (multi-fidelity RHP with linear CoM dynamics) always fails to converge after a few planning cycles, no matter how many steps look-ahead we include in the PH. In contrast, with only 1-step look-ahead in the PH, the baseline can successfully plan trajectories across all considered scenarios. These trajectories can be successfully tracked in our simulation. This suggests that considering the angular dynamics in the PH is necessary.

On the other hand, although candidates 2 and 3 can successfully generate the centroidal trajectories with 1-step PH, the whole-body controller fails to track the resultant trajectories in most of the cases. To obtain centroidal trajectories that can be tracked successfully in our simulation, candidate 2 and 3 require at least two steps look-ahead in PH.

2) Accumulated Cost: Next, we compare the quality of the trajectories generated from each RHP framework. The quality of the trajectories is measured by the accumulated running cost reported from RHP frameworks. To obtain this cost, we sum the running cost J of execution horizon for all planning cycles. In our work, the cost encourages smooth centroidal trajectories by penalizing large CoM acceleration and angular momentum. To have a fair comparison, we normalized the accumulated cost over the distance traveled along the forward direction (x-axis). Fig. 4d-4f plot accumulated cost for all RHP frameworks under different PH lengths. Usually, the accumulated cost of each RHP framework tends to improve if we increase the PH length. However, for the cases with more than four steps look-ahead in the PH, there is only minor improvements, which suggests that long look-ahead horizon in RHP can be unnecessary for the scenarios considered.

Further, we can find that the baseline always has the least accumulated cost for all scenarios. In comparison, the accumulated cost of candidate 2 and 3 are higher than the baseline. Especially, much higher cost can be observed when we only assign 1-step look-ahead in the PH. We find that such centroidal trajectories tend to exhibit jerky movements in our robot simulation, which causes failures of tracking most of the time. With increased PH length (two steps look-ahead in the PH and more), although the cost is still higher than the baseline, the resultant centroidal trajectories can be successfully executed in our whole-body simulation.

We speculate that, the convex outer approximation allows violation on system dynamics in PH, which gives rise to optimistic value function prediction of PH and in turn results in higher accumulated cost of execution horizon. Further, we notice candidate 3 have smaller cost than candidate 2. We guess this is due to the conservative nature of the point foot model, which may regularise the violation on system dynamics in PH. We believe further investigation is needed for understanding how the model simplifications in PH affect the quality of resultant trajectories.

[4]In the result section only, we refer to the prediction horizon with PH for brevity.

TABLE III: Mean and standard deviation of the computation time. We run each RHP framework ten times for each scenario with randomly selected initial guesses for the first and the second planning cycles.

Method	up-and-down hill	v-shape terrain	rubbles
baseline (1-step PH)	1.63±1.56s	0.78±0.22s	1.84±1.67s
baseline (2-step PH)	2.93±2.02s	1.45±0.58s	3.53±3.16s
Candidate 2 (2-step PH)	1.30±0.59s	1.00±0.27s	1.12±0.24s
Candidate 3 (2-step PH)	**0.58±0.03**s	**0.57±0.06**s	**0.62±0.07**s

3) Computation Time: To highlight the computational benefit of our approach, we compare the best computation time of each RHP framework in Table III. For all RHP frameworks, the best computation time is achieved when the PH is set to the minimum length for success (See Table II). We recall here from Section V-B.1 that for the baseline the minimum PH length for success is one step (1-step PH), while for candidate 2 and 3 the minimum PH length for success is two steps (2-step PH). Further, we also include the computation time for the baseline with 2-step PH for completeness. In Table III, we observe that with the same PH length (2-step PH), both candidate 2 and 3 achieve faster computation than the baseline. More importantly, candidate 3 with 2-step PH achieves the best computation speed, which is 1.4x-3.0x (average 2.4x) faster than the fastest baseline (1-step PH).

We consider this computational improvement as significant, because it suggests the potential of using candidate 3 in an online fashion. Rather than controlling the robot with a high frequency, RHP can run in a slower frequency to constantly update the reference trajectories for a controller to track. Nevertheless, to ensure continuous operation of the robot, it is required that the online RHP should at least compute the motion plan for the next cycle before the robot completes the execution of the current cycle. In Fig. 5, we compare the computation time of each planning cycle—for both the baseline with 1-step PH and candidate 3 with 2-step PH—against the duration of the motion plan (to be executed) in each cycle. These computation time and motion duration are averaged over all the scenarios considered. We can observe that the computation time of the baseline is often larger than the motion duration and it exhibits large standard deviation (also shown in Table III). This indicates that the baseline is not suitable for online RHP. In contrast, the mean computation time of candidate 3 is always smaller than the motion duration, and the standard deviation of the computation time is rather small (also shown in Table III). This strongly supports the potential of employing candidate 3 for online RHP.

In addition, we evaluate how the computation time of each RHP framework scales with respect to the length of PH.

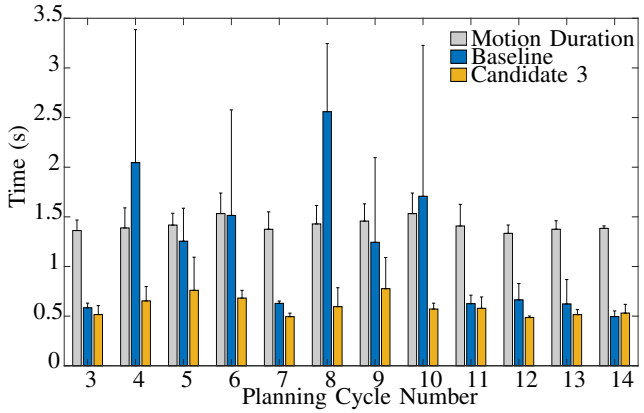


Fig. 5: Computation time of baseline with 1-step PH, candidate 3 with 2-step PH, and the duration of the planned motion in each cycle. Each bar represents the average value across all the scenarios considered. Since the humanoid robot swings the same leg per two cycles, we warm-start each cycle with the solution computed in two cycles before (except the first and the second cycle whose initial guesses are not readily available).

Fig. 4g-4i shows the average computation time[5] for each RHP framework with different lengths of PH. The average computation time of all RHP frameworks grows roughly linearly as the length of PH increases. This is expected as the number of decision variables and constraints increases too. More importantly, as the baseline (single-fidelity RHP) is fully non-convex, its computation time grows rapidly. In contrast, the computation time of multi-fidelity RHPs (candidate 2 and 3) increases at a much smaller rate. Also, although both candidate 2 and baseline assume rectangular foot, candidate 2 achieves a much faster computation speed than the baseline. This suggests that the convexification in the prediction horizon can improve the computation time. Finally, since candidate 3 reduces the model's dimensionality due to the point foot assumption in the prediction horizon, further computational improvement is achieved.

VI. Discussion

Our experiments first confirm that a prediction horizon is vital to the RHP frameworks. As Table II lists, at least 1-step look-ahead in the prediction horizon is required to achieve successful RHPs. Indeed, from an optimal control point of view, the prediction horizon can be seen as a value function approximation, which evaluates whether the motion decisions made in the execution horizon are beneficial for the future.

Second, we empirically demonstrate that it is possible to achieve successful RHP while considering simplified dynamics model (convex relaxations) in the prediction horizon. We demonstrate this can significantly improve the computation time, but comes with a price—the cost of the resultant trajectories (jerkiness) is increased. Nevertheless, with the proposed relaxations the resultant trajectories can be successfully tracked by a whole-body controller. On the other hand, the multi-fidelity RHP can fail to converge if we only consider linear CoM dynamics in the prediction horizon. This suggests that evaluating the evolution of angular dynamics in the prediction horizon is important. We believe that further investigations regarding the trade-off between the computation efficiency and model complexity, as well as the motion quality are necessary.

Lastly, we show that for the scenarios we considered, in each planning cycle our multi-fidelity RHP can always complete the computation of feasible trajectories within the time window of the motion for execution. This suggests that our multi-fidelity RHP has the potential to be applied in an online fashion. To further improve the computation speed, we can explore avenues of research. First, we can design customized numerical solvers (e.g. hierarchical optimization [48]) to better exploit the convexity of the formulation. This can be done by solving the convex and non-convex parts of the formulation alternatively until convergence. Secondly, we can reduce the look-ahead horizon by learning the value function approximation provided by the prediction horizon.

In our approach, we pre-define the sequence of contact surfaces [44], [49] and the selection of gait pattern (i.e. the sequence in which the feet make and break contacts with the environment) [50]. Ideally, these discrete decisions should be automatically resolved by the optimization. However, this gives rise to combinatorial problems which are generally difficult to solve. We believe that handling the combinatorial complexity of the multi-contact problem in a computationally efficient manner would be beneficial, as it can further augment the flexibility of our approach.

VII. Conclusion & Future Work

In this work, we present a novel RHP formulation that achieves efficient computation of centroidal trajectories for bipedal walking in multi-contact scenarios. The proposed formulation features system dynamics constraints with multiple levels of fidelity. More specifically, the execution horizon uses an accurate, non-linear centroidal model, while the prediction horizon relaxes the centroidal model into a convex approximation. As a result the executed motion is always dynamically consistent (at the centroidal level) while the overall complexity of the RHP formulation is reduced. Using the proposed formulation, we achieve a computational improvement in the range of 1.4x-3.0x (average 2.4x) against the baseline. We also illustrate that our approach can be potentially used for online RHP, owing to its fast computation. In the future, we plan to further improve the computation time by explicitly exploiting the convexity or reducing the problem dimensionality via value function learning.

Acknowledgements

This research is supported by EU H2020 project Memory of Motion (MEMMO, 780684), EPSRC UK RAI Hub for Offshore Robotics for Certification of Assets (ORCA, EP/R026173/1) and The Alan Turing Institute.

[5]Each data point refers to the mean computation time averaged over fourteen planning cycles.

REFERENCES

[1] M. Posa and R. Tedrake, "Direct trajectory optimization of rigid body dynamical systems through contact," in *Algorithmic foundations of robotics X*. Springer Berlin Heidelberg, 2013, pp. 527–542.

[2] 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 Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1560–1567, 2018.

[3] M. Toussaint, K. Allen, K. Smith, and J. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," in *Proc. Robotics: Science and Systems*, 2018.

[4] T. Stouraitis, I. Chatzinikolaidis, M. Gienger, and S. Vijayakumar, "Dyadic collaborative manipulation through hybrid trajectory optimization," in *Conf. Robot Learning (CoRL)*, 2018, pp. 869–878.

[5] H. Ferrolho *et al.*, "Inverse dynamics vs. forward dynamics in direct transcription formulations for trajectory optimization," 2021.

[6] J. Ding, S. Xin, T. L. Lam, and S. Vijayakumar, "Versatile locomotion by integrating ankle, hip, stepping, and height variation strategies," in *Proc. Int. Conf. Robot. Automat.*, 2021.

[7] S. Tonneau *et al.*, "An efficient acyclic contact planner for multiped robots," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 586–601, 2018.

[8] B. Aceituno-Cabezas *et al.*, "Simultaneous Contact, Gait and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization," *IEEE Robot. Automat. Lett.*, 2018.

[9] H. Ferrolho *et al.*, "Whole-body end-pose planning for legged robots on inclined support surfaces in complex environments," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2018, pp. 944–951.

[10] "Optimizing dynamic trajectories for robustness to disturbances using polytopic projections," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2020, pp. 7477–7484.

[11] H.-W. Park, P. M. Wensing, and S. Kim, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *Proc. Robotics: Science and Systems*, 2015.

[12] C. Mastalli *et al.*, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *Proc. Int. Conf. Robot. Automat.*, 2020.

[13] M. Neunert *et al.*, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1458–1465, 2018.

[14] J. Di Carlo *et al.*, "Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[15] E. Dantec *et al.*, "Whole body model predictive control with a memory of motion: Experiments on a torque-controlled Talos," in *Proc. Int. Conf. Robot. Automat.*, 2021.

[16] T. Dinev *et al.*, "Modeling and control of a hybrid wheeled jumping robot," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2020, pp. 2563–2570.

[17] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[18] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE Trans. on Automatic Control*, vol. 50, no. 5, pp. 674–678, 2005.

[19] H. Li, R. J. Frei, and P. M. Wensing, "Model hierarchy predictive control of robotic systems," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3373–3380, 2021.

[20] D. E. Orin, A. Goswami, and S. H. Lee, "Centroidal dynamics of a humanoid robot," *Aut. Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.

[21] O. Stasse *et al.*, "TALOS: A new humanoid research platform targeted for industrial applications," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2017, pp. 689–695.

[22] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of momentum dynamics for multi-contact motion generation," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2017.

[23] B. Ponton *et al.*, "On time optimization of centroidal momentum dynamics," in *Proc. Int. Conf. Robot. Automat.*, 2018, pp. 1–7.

[24] A. Del Prete and N. Mansard, "Robustness to joint-torque-tracking errors in task-space inverse dynamics," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1091–1105, 2016.

[25] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 4906–4913.

[26] G. Schultz and K. Mombaur, "Modeling and optimal control of human-like running," *IEEE/ASME Trans. on mechatronics*, vol. 15, no. 5, pp. 783–792, 2009.

[27] K. H. Koch, K. Mombaur, and P. Soueres, "Optimization-based walking generation for humanoid robot," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 498–504, 2012.

[28] T. Erez and E. Todorov, "Trajectory optimization for domains with contacts using inverse dynamics," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 4914–4919.

[29] J. Koschorreck and K. Mombaur, "Modeling and optimal control of human platform diving with somersaults and twists," *Optimization and Engineering*, vol. 13, no. 1, pp. 29–56, 2012.

[30] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, 2012.

[31] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2014.

[32] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2016, pp. 2703–2710.

[33] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *IEEE Trans. Robot.*, 2018.

[34] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *Proc. Int. Conf. Robot. Automat.*, 2013, pp. 3103–3109.

[35] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multi-contact momentum-control," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2015.

[36] J. Carpentier *et al.*, "A versatile and efficient pattern generator for generalized legged locomotion," in *Proc. Int. Conf. Robot. Automat.*, 2016, pp. 3555–3561.

[37] P. Fernbach *et al.*, "C-CROC: Continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 676–691, 2020.

[38] S. Kajita *et al.*, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. Int. Conf. Robot. Automat.*, vol. 2, 2003, pp. 1620–1626.

[39] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 355–368, 2015.

[40] P.-B. Wieber, "Viability and predictive control for safe locomotion," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2008, pp. 1103–1108.

[41] S. Caron and A. Kheddar, "Multi-contact walking pattern generation based on model preview control of 3D COM accelerations," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2016, pp. 550–557.

[42] S. Caron and Q.-C. Pham, "When to make a step? tackling the timing problem in multi-contact locomotion by topp-mpc," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2017, pp. 522–528.

[43] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2016, pp. 579–586.

[44] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2014, pp. 279–286.

[45] S. Tonneau *et al.*, "2PAC: Two-point attractors for center of mass trajectories in multi-contact scenarios," *ACM Trans. on Graphics (TOG)*, vol. 37, no. 5, pp. 1–14, 2018.

[46] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*, G. Di Pillo and M. Roma, Eds. Springer US, 2006, pp. 35–59.

[47] J. A. E. Andersson *et al.*, "CasADi – a software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, Mar. 2018.

[48] T. Stouraitis, I. Chatzinikolaidis, M. Gienger, and S. Vijayakumar, "Online hybrid motion planning for dyadic collaborative manipulation via bilevel optimization," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1452–1471, 2020.

[49] S. Tonneau *et al.*, "SL1M: Sparse L1-norm minimization for contact planning on uneven terrain," in *Proc. Int. Conf. Robot. Automat.*, 2020, pp. 6604–6610.

[50] J. Wang *et al.*, "Automatic Gait Pattern Selection for Legged Robots," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2020.