2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Optimal Scheduling of Models and Horizons for Model Hierarchy Predictive Control

Charles Khazoom[1], Steve Heim[1], Daniel Gonzalez-Diaz[1] and Sangbae Kim[1]

***Abstract*— Model predictive control (MPC) is a powerful tool to control systems with non-linear dynamics and constraints, but its computational demands impose limitations on the dynamics model used for planning. Instead of using a single complex model along the MPC horizon, model hierarchy predictive control (MHPC) reduces solve times by planning over a sequence of models of varying complexity within a single horizon. Choosing this model sequence can become intractable when considering all possible combinations of reduced order models and prediction horizons. We propose a framework to systematically optimize a model schedule for MHPC. We leverage trajectory optimization (TO) to approximate the accumulated cost of the closed-loop controller. We trade off performance and solve times by minimizing the number of decision variables of the MHPC problem along the horizon while keeping the approximate closed-loop cost near optimal. The framework is validated in simulation with a planar humanoid robot as a proof of concept. We find that the approximated closed-loop cost matches the simulated one for most of the model schedules, and show that the proposed approach finds optimal model schedules that transfer directly to simulation, and with total horizons that vary between 1.1 and 1.6 walking steps.**

I. INTRODUCTION

A. Motivation

Model predictive control (MPC) can provide closed-loop stability for systems with complex, nonlinear dynamics and state and input constraints by repeatedly solving a trajectory optimization problem (TO) in a receding horizon fashion. The main bottleneck is the solve time of the TO. It should be fast enough such that the closed-loop control bandwidth matches the timescale of the system dynamics. To reduce the solve time, the predictive horizon of MPC can be shortened and a terminal value function can be added to the cost to account for the truncated horizon [1]–[3]. Another common approach is to plan with a reduced order model [4]–[6] or even a hierarchy of models of varying complexity [7]–[10]. Choosing the sequence of models and their corresponding horizons is typically done by trial and error. We propose a framework to systematically optimize a schedule of reduced order models, and validate it with a proof of concept on a planar, 13-DoF humanoid robot.

B. Related Work

Full-body models can exploit the full dynamic capabilities of the robot [11], [12], but it remains challenging to optimize full-body trajectories in real time for high-dimensional systems like legged robots. Due to the complexity of these systems, roboticists have adopted reduced order models (ROMs) to simplify analysis and control [13], [14].

For instance, the linear inverted pendulum (LIP) [15], [16] and the spring-loaded inverted pendulum (SLIP) [5], [17] are common ROMs for humanoid locomotion, and have inspired many extensions [16], [18], [19]. The LIP, the SLIP and the single rigid body (SRB) have been used in MPC to rapidly optimize trajectories, which are typically tracked by a whole-body controller that reasons about the instantaneous full-body dynamics [6], [20]–[22]. Despite its utility, this approach can limit performance, as it forces the robot to behave like a ROM that cannot account for whole-body constraints [23].

To find less restrictive ROMs, Chen and Posa [23] proposed a bilevel optimization to synthesize ROMs that min-

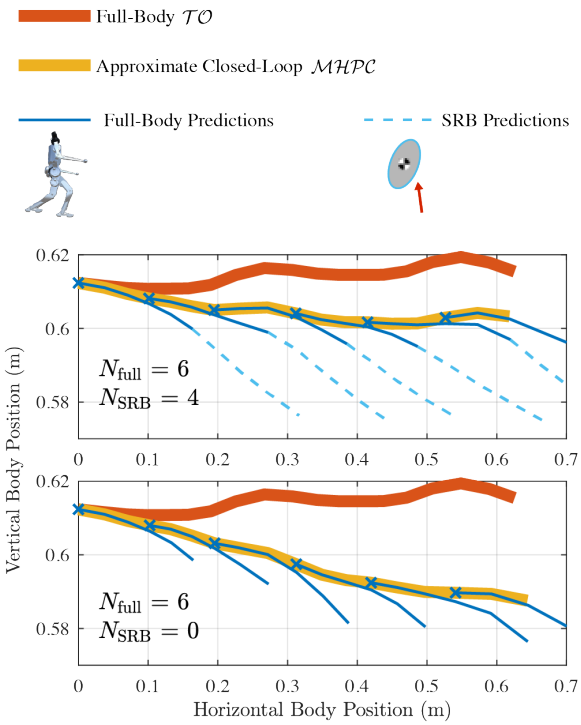

Fig. 1. Body position from full-body trajectory optimization ($\mathcal{TO}$) compared to an approximate closed-loop trajectory using ($\mathcal{MHPC}$). The goal of our proposed framework is to minimize the dimensionality of ($\mathcal{MHPC}$) while keeping the closed-loop cost near optimal. The framework is validated using a case study where we optimize the horizons of whole-body (N_{full}) and single rigid body dynamics (N_{SRB}).

This work was supported by Naver Labs, Disney Research Imagineering, NSERC, and the Swiss National Science Foundation (Grant No P2SKP2_194954).

[1]Department of Mechanical Engineering Department, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA, United States ckhaz@mit.edu, sheim@mit.edu,dgdiaz@mit.edu , sangbae@mit.edu

imize the cost incurred by a full-body model. For a chosen ROM parameterization, they find parameters that are least restrictive for a distribution of tasks. However, the mapping from the ROM to the full dimensionality of the high-order model is not well-defined.

A recent strategy that combines the advantages of ROMs and full-body models is model hierarchy predictive control (MHPC), which consists of planning over a hierarchy of models across the MPC horizon [7]–[9]. Both the performance and solve times of the closed-loop controller are affected by the complexity of the models used and their prediction horizon. In previous work, a balance between the number of failures [7], [9] and the average solve time was used to judge the performance of various model schedules. Wang *et al.* [8] used the closed-loop accumulated cost instead of the number of failures. These approaches require testing multiple MHPC configurations by trial and error to select one that best trades off solve time and model complexity. This ad hoc process, however, becomes intractable when considering the plethora of possible ROMs and all possible combinations across the horizon, each having different equations of motion.

C. Contribution

The goal of this paper is to provide a practical framework to optimize model schedules for MHPC offline. An optimal schedule should minimize the solve time when deployed for online MPC, with little impact on the closed-loop performance. The framework relies on two contributions to enable tractability. First, we formulate the model schedule optimization as a constrained problem that minimizes the solve time of the MHPC trajectory optimization subject to a suboptimality bound on the closed-loop cost. We identify computable proxies for solve time and closed-loop cost that can be obtained from trajectory optimization. Specifically, we leverage the low-resolution solutions of the MHPC predictions to approximate the closed-loop dynamics and estimate the closed-loop performance of MHPC, which reduces the computational and implementation burden of the model schedule optimization. Second, we represent the ROMs as a subset of the full order model by enforcing additional holonomic constraints, instead of using hybrid dynamics, which simplifies the implementation of the MHPC optimization. As a case study, we optimize the horizons of three models for a planar humanoid walking robot. This case study is amenable to brute-force, and allows us to validate our framework against higher-fidelity simulation.

The remainder of the paper is organized as follows. Section II provides background on TO, ROMs and MPC. Section III presents the problem statement, Section IV detail the specific implementation, and Section V presents the results.

II. BACKGROUND

This section introduces TO and MPC, explains how ROMs are typically used in MPC, and details the MHPC approach. We use the symbol $\hat{(\cdot)}$ to denote variables and functions that relate to ROMs, and $(\cdot)^*$ for optimal costs or solutions.

A. Trajectory Optimization and Model Predictive Control

Trajectory optimization solves for a trajectory of states and control inputs that minimize the cost along a horizon of N time-steps while respecting systems dynamics $\boldsymbol{f}(\cdot)$, path constraints $\boldsymbol{g}(\cdot)$ and terminal constraints $\boldsymbol{g}_N(\cdot)$:

$$V^*(\boldsymbol{x}_1) = \min_{\boldsymbol{\mathcal{X}}} \sum_{k=1}^{N-1} c(\boldsymbol{x}_k, \boldsymbol{u}_k) + c_N(\boldsymbol{x}_N) \qquad (\mathcal{TO})$$

subject to

$$\begin{aligned} &\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \;\forall\; k \in \{1, ..., N-1\} \\ &\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{0} \;\forall\; k \in \{1, ..., N-1\} \\ &\boldsymbol{g}_N(\boldsymbol{x}_N) \leq \boldsymbol{0}, \end{aligned}$$

where $V^*(\boldsymbol{x}_1)$ is the optimal accumulated cost starting from the current state $\boldsymbol{x}_1$, the system state and control input at time k are $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ respectively, $c(\cdot)$ is the running cost, $c_N(\cdot)$ is the terminal cost, and $\boldsymbol{\mathcal{X}} = \{\{\boldsymbol{x}, \boldsymbol{u}\}_{k=1}^{N-1}, \boldsymbol{x}_N\}$ describes the set of all decision variables: the trajectories of states and inputs.

MPC closes the loop by solving $(\mathcal{TO})$ from the current state, applying the first control input $\boldsymbol{u}_1$ from the solution $\boldsymbol{\mathcal{X}}$, letting the system dynamics evolve and then repeating.

B. MPC with Reduced Order Models

To achieve fast enough solve times for MPC, a single ROM is often used. In this case, the measured state of the robot $\boldsymbol{x}_1$ must be mapped to the lower-order ROM state $\hat{\boldsymbol{x}}_1$ using the state mapping $\boldsymbol{\phi}(\cdot)$,

$$\hat{\boldsymbol{x}}_1 = \boldsymbol{\phi}(\boldsymbol{x}_1). \qquad (1)$$

Given $\hat{\boldsymbol{x}}_1$, the optimized ROM trajectory $\hat{\boldsymbol{\mathcal{X}}}^*$ must be mapped back to the full robot for execution using the control mapping $\boldsymbol{\Phi}(\cdot)$:

$$\boldsymbol{u} = \boldsymbol{\Phi}(\hat{\boldsymbol{\mathcal{X}}}^*). \qquad (2)$$

If offline TO is used, the entire ROM trajectory $\hat{\boldsymbol{\mathcal{X}}}^*$ must be mapped back. In MPC, only the first timestep of the trajectory is used. For instance, $\boldsymbol{\Phi}(\cdot)$ can be a whole-body controller that reasons about the instantaneous full-body dynamics at a higher frequency than MPC and tracks the ROM trajectories [21], [22], leading to a hierarchy of controllers. A major hurdle to this approach is the complex interplay between both MPC and mapping $\boldsymbol{\Phi}(\cdot)$, which can lead to time-consuming tuning [7].

1) Model Hierarchy Predictive Control: A recent approach to improve the solve time of MPC is model hierarchy predictive control (MHPC) [7]–[9]. Instead of planning with a single ROM and using $\boldsymbol{\Phi}(\cdot)$ to track the plan, MHPC plans over a hierarchy of models within a single TO, with hybrid dynamics that switch between reduced order models according to a model schedule. In this paper, we define $\boldsymbol{\mathcal{S}} = \left\{\{\hat{\boldsymbol{f}}, \hat{\boldsymbol{g}}\}_{k=1}^{\hat{N}-1}, \hat{\boldsymbol{g}}_{\hat{N}}\right\}$ as a model schedule composed of different ROM dynamics and constraints at each time-step.

MHPC takes the following form:

$$\hat{V}^{\boldsymbol{\mathcal{S}}}(\hat{\boldsymbol{x}}_1) = \min_{\hat{\boldsymbol{\mathcal{X}}}} \sum_{k=1}^{\hat{N}-1} \hat{c}_k(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k) + \hat{c}_{\hat{N}}(\hat{\boldsymbol{x}}_{\hat{N}}) \quad (\mathcal{MHPC})$$

subject to

$$\begin{aligned} &\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{f}}_k(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k) \ \forall \ k \in \{1, ..., \hat{N}-1\} \\ &\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k) \leq \boldsymbol{0} \ \forall \ k \in \{1, ..., \hat{N}-1\} \\ &\hat{\boldsymbol{g}}_N(\hat{\boldsymbol{x}}_N) \leq \boldsymbol{0}, \end{aligned}$$

where $\hat{\boldsymbol{\mathcal{X}}} = \left\{\{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}\}_{k=1}^{N-1}, \hat{\boldsymbol{x}}_N\right\}$ is the set of decision variables, and $\hat{V}^{\boldsymbol{\mathcal{S}}}(\hat{\boldsymbol{x}}_1)$ is the optimal open-loop cost of the trajectory composed of the model schedule $\boldsymbol{\mathcal{S}}$. Note the dependency of the dynamics and constraints on the time-step k.

The policy induced by $(\mathcal{MHPC})$ is still given by (2), and the corresponding cost accumulated by the robot under this policy is

$$V^{\boldsymbol{\mathcal{S}}}(\boldsymbol{x}_1) = \sum_{k=1}^{N-1} c\left(\boldsymbol{x}_k, \boldsymbol{\Phi}(\hat{\boldsymbol{\mathcal{X}}}^*)\right) + c_N(\boldsymbol{x}_N), \tag{3}$$

where $V^{\boldsymbol{\mathcal{S}}}(\boldsymbol{x}_1)$ (note the absence of $\hat{(\cdot)}$) is the closed-loop cost incurred by the robot under the MHPC controller. This is the quantity we wish to keep low, as opposed to the open-loop cost $\hat{V}^{\boldsymbol{\mathcal{S}}}$ of the MHPC predictions [24].

Note that using a full-body model at $k = 1$ simplifies control as $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{\Phi}(\cdot)$ can be treated as identity.

III. PROBLEM STATEMENT

We wish to identify offline a model schedule $\boldsymbol{\mathcal{S}}$ that minimizes solve time while bounding the closed-loop performance of the robot under the MHPC controller. Thus, we consider the model schedule $\boldsymbol{\mathcal{S}}$ to be optimal when

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{S}}^* = \ \arg\min_{\boldsymbol{\mathcal{S}}} (\textit{solve time}) \quad (\mathcal{MS})$$

subject to

$$\frac{V^{\boldsymbol{\mathcal{S}}}(\boldsymbol{x}_1) - V^*(\boldsymbol{x}_1)}{V^*(\boldsymbol{x}_1)} \leq \epsilon.$$

The left-hand side in the inequality constraint represents the relative cost error (RCE), which must be within some chosen bound ϵ. The optimal cost $V^*(\boldsymbol{x}_1)$ provides a reference value to more easily choose ϵ. Note that we do not bound the norm of the RCE, because in practice, we found a few model schedules where $V^{\boldsymbol{\mathcal{S}}}(\boldsymbol{x}_1) < V^*(\boldsymbol{x}_1)$.

As a proxy for solve time, we minimize the number of decision variables of $(\mathcal{MHPC})$. To approximate $V^*(\boldsymbol{x}_1)$ and $V^{\boldsymbol{\mathcal{S}}}(\boldsymbol{x}_1)$, we use trajectory optimization.

IV. IMPLEMENTATION

The problem $(\mathcal{MS})$ is a bilevel optimization, with $(\mathcal{MHPC})$ in its inner loop. We solve it by computing offline:

1) the optimal cost $V^*(\boldsymbol{x}_1)$ from $(\mathcal{TO})$ for the full robot
2) the solution to $(\mathcal{MHPC})$ and induced policy (2), for multiple $\boldsymbol{\mathcal{S}}$
3) the cost $V^{\boldsymbol{\mathcal{S}}}(\boldsymbol{x}_1)$ from (3), for multiple $\boldsymbol{\mathcal{S}}$.

This section details how these three terms are approximated using TO and how they are used together to solve for $\boldsymbol{\mathcal{S}}^*$.

A. Full-Body Optimization

To estimate $V^*(\boldsymbol{x}_1)$ offline, we formulate $(\mathcal{TO})$ with a direct transcription method with full-body dynamics. We treat $V^*(\boldsymbol{x}_1)$ as a lower-bound on the cost incurred by the robot using MHPC.

1) Full-Body Dynamics: The configuration of a robot composed of a floating base with n_b degrees of freedom and n_j joints is described by $\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}_b^\top & \boldsymbol{q}_j^\top\end{bmatrix}^\top$, where $\boldsymbol{q}_b \in \mathbb{R}^{n_b}$ and $\boldsymbol{q}_j \in \mathbb{R}^{n_j}$ are the unactuated base and actuated generalized coordinates, respectively. For n_h holonomic constraints, the robot's constrained dynamics are described by

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{S}_a\boldsymbol{u} + \boldsymbol{J}^\top(\boldsymbol{q})\boldsymbol{F}, \tag{4}$$

where $\boldsymbol{H}(\boldsymbol{q})$ is the mass matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the Coriolis and gravitational term, $\boldsymbol{S}_a = \begin{bmatrix}\boldsymbol{0}^{n_b \times n_j} & \boldsymbol{I}^{n_j \times n_j}\end{bmatrix}$ is a matrix that selects the actuated joints torques $\boldsymbol{u} \in \mathbb{R}^{n_j}$, and $\boldsymbol{J}(\boldsymbol{q})$ is the Jacobian of the holonomic constraints $\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{0}$, and $\boldsymbol{F} \in \mathbb{R}^{n_h}$ is the force that enforces the holonomic constraint.

2) Cost and Decision Variables: The decision variables are

$$\boldsymbol{\mathcal{X}} = \{\{\boldsymbol{x}, \boldsymbol{p}\}_{k=1}^{N}, \{\boldsymbol{u}, \boldsymbol{F}_c\}_{k=1}^{N-1}\}, \tag{5}$$

where the state $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{q}^\top & \dot{\boldsymbol{q}}^\top\end{bmatrix}^\top$ and $\boldsymbol{p}$ and $\boldsymbol{F}_c$ are additional variables corresponding to contact positions and ground reaction force, respectively. We use a quadratic cost designed to track a desired velocity.

3) Constraints: The robot's configuration must be consistent with the contact positions determined by the forward kinematics $\boldsymbol{f}_{\text{kin}}(\cdot)$.

$$\boldsymbol{f}_{\text{kin}}(\boldsymbol{q}_k) = \boldsymbol{p}_k. \tag{6}$$

The dynamics constraint is given by

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{q}_k)(\dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}_k) + \boldsymbol{C}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)dt_k \\ = \boldsymbol{B}\boldsymbol{u}_k dt_k + \boldsymbol{J}_c^\top(\boldsymbol{q}_k)\boldsymbol{F}_{c,k}dt_k, \end{aligned} \tag{7}$$

where $\boldsymbol{J}_c$ is the Jacobian of the forward kinematics. The trapezoidal integration constraint

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \frac{1}{2}(\dot{\boldsymbol{q}}_{k+1} + \dot{\boldsymbol{q}}_k)dt_k \tag{8}$$

ensures continuity between the timesteps of duration dt_k. The ground reaction forces must respect the friction cone and unilateral force constraint (9) and the predetermined contact schedule (10)

$$\boldsymbol{F}_{c,k} \in \boldsymbol{\mathcal{F}}(\boldsymbol{\theta}_k) \tag{9}$$

$$\boldsymbol{p}_k \in \boldsymbol{\mathcal{C}}(\boldsymbol{\theta}_k), \tag{10}$$

where $\boldsymbol{\theta}_k$ is a contact schedule parameter that modifies the feasible sets of ground reaction forces and foot locations $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{C}}$ along the horizon.

B. Optimization With a Hierarchy of Models

In the scope of this paper, we consider three candidate models to be used in ($\mathcal{MHPC}$), in the following order: 1) the full-body model; 2) the single rigid body (SRB) model, where all joints are locked and only the floating base is allowed to move; 3) a void model, where both the joints and the floating base are locked and the cost is zero until the end of the horizon. The void model effectively truncates the horizon.

1) Cost and Decision Variables: The decision variables are

$$\hat{\boldsymbol{\mathcal{X}}} = \{\{\boldsymbol{x}, \boldsymbol{p}\}_{k=1}^{N}, \{\boldsymbol{u}, \boldsymbol{F}_c, \boldsymbol{F}_{\text{SRB}}, \boldsymbol{F}_{\text{void}}\}_{k=1}^{N-1}\}, \tag{11}$$

where $\boldsymbol{F}_{\text{SRB}}$ and $\boldsymbol{F}_{\text{void}}$ are the additional constraint forces for the SRB and the void models. The same running cost function as in ($\mathcal{TO}$) is used for the full-body and SRB models and we focus on the constraints.

2) Binary Parameters: To allow for time-varying dynamics and constraints along the horizon, we define the model schedule $\boldsymbol{\mathcal{S}}$ as the set of binary parameters

$$\boldsymbol{\mathcal{S}} = \Big\{ s_{m,k} \in \{0,1\} | \sum_{m \in \{\text{full,SRB,void}\}} s_{m,k} = 1 \ \forall \ k \Big\}, \tag{12}$$

where the constraint in (12) ensures that only one model is active at each time-step k. Each element $s_{m,k}$ is used to activate the dynamics and constraints of a ROM $m \in \{\text{full}, \text{SRB}, \text{void}\}$ at time-step k.

In (7), the forward kinematics of the contact foot and its Jacobian $\boldsymbol{J}_c$ are used as the holonomic constraint. Similarly, we use holonomic constraints to achieve ROM behaviors without having to derive their equations of motion.

3) SRB Model Constraints: The SRB model can be viewed as a floating-base model with all joints locked, as expressed by the constraint $\boldsymbol{h}_{\text{SRB}}(\boldsymbol{q}_k) : \mathbb{R}^{(n_b+n_j)} \to \mathbb{R}^{n_j}$:

$$\boldsymbol{h}_{\text{SRB}}(\boldsymbol{q}_k) = \begin{bmatrix} \boldsymbol{0}_{6\times n_j} & \boldsymbol{I}_{n_j \times n_j} \end{bmatrix} \boldsymbol{q}_k = \text{constant}. \tag{13}$$

We use the big-M method [25, Section 9.7] to enforce that time derivative of (13) is zero when the SRB is active ($s_{\text{SRB},k} = 1$):

$$-\boldsymbol{M}(1 - s_{\text{SRB},k}) \leq \boldsymbol{J}_{\text{SRB}} \dot{\boldsymbol{q}}_k \leq \boldsymbol{M}(1 - s_{\text{SRB},k}), \tag{14}$$

where $\boldsymbol{J}_{\text{SRB}} = \begin{bmatrix} \boldsymbol{0}_{6\times n_j} & \boldsymbol{I}_{n_j \times n_j} \end{bmatrix}$ is the Jacobian of $\boldsymbol{h}_{\text{SRB}}$, and $\boldsymbol{M}$ is a big number that deactivates the constraint when $s_{\text{SRB},k} = 0$. Additionally, the contact positions $\boldsymbol{p}_k$ are constrained by (6) only when $s_{\text{full}} = 1$,

$$-\boldsymbol{M}(1 - s_{\text{full},k}) \leq \boldsymbol{f}_{\text{kin}}(\boldsymbol{q}_k) - \boldsymbol{p}_k \leq \boldsymbol{M}(1 - s_{\text{full},k}), \tag{15}$$

and the simpler maximum-distance constraint is used when $s_{\text{SRB},k} = 1$:

$$\begin{aligned} ||\boldsymbol{p}_k - \boldsymbol{q}_{xyz}||^2 - \ell_{\max}^2 &\leq \boldsymbol{M}(1 - s_{\text{SRB},k}) \\ ||\boldsymbol{p}_k - \boldsymbol{q}_{xyz}||^2 - \ell_{\max}^2 &\leq -\boldsymbol{M}(1 - s_{\text{SRB},k}), \end{aligned} \tag{16}$$

where $\boldsymbol{q}_{xyz}$ is the position of the floating base and $\ell_{\max}$ is a maximum leg length.

4) Void Model Constraints: The void model locks both the joints and the floating base, effectively truncating the planning horizon. The constraint $\boldsymbol{h}_{\text{void}}(\boldsymbol{q}_k) : \mathbb{R}^{(n_b+n_j)} \to \mathbb{R}^{(n_b+n_j)}$ is

$$\boldsymbol{h}_{\text{void}}(\boldsymbol{q}_k) = \boldsymbol{I}_{(n_b+n_j)\times(n_b+n_j)} \boldsymbol{q}_k = \text{constant}, \tag{17}$$

and its time derivative is enforced when $s_{\text{void},k} = 1$:

$$-\boldsymbol{M}(1 - s_{\text{void},k}) \leq \boldsymbol{J}_{\text{void}} \dot{\boldsymbol{q}}_k \leq \boldsymbol{M}(1 - s_{\text{void},k}), \tag{18}$$

where $\boldsymbol{J}_{\text{void}} = \boldsymbol{I}_{(n_b+n_j)\times(n_b+n_j)}$ is an identity matrix.

5) Constrained Full-body Dynamics: The full-body dynamics are constrained to be consistent with the holonomic constraints defining each model. Thus, the constrained dynamics are

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{q}_k)(\dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}_k) + \boldsymbol{C}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k) dt_k &= \boldsymbol{B}\boldsymbol{u}_k dt_k && \text{(19a)} \\ + s_{\text{full},k} \boldsymbol{J}_c^\top(\boldsymbol{q}_k) \boldsymbol{F}_{c,k} dt_k && \text{(19b)} \\ + s_{\text{SRB},k} \boldsymbol{J}_{\text{SRB}}^\top \boldsymbol{F}_{\text{SRB},k} dt_k && \text{(19c)} \\ + s_{\text{void},k} \boldsymbol{J}_{\text{void}}^\top \boldsymbol{F}_{\text{void},k} dt_k && \text{(19d)} \\ + s_{\text{SRB},k} \boldsymbol{J}_{\text{fb}}^\top \boldsymbol{F}_{c,k} dt_k. && \text{(19e)} \end{aligned}$$

The term (19e) maps the ground reaction force $\boldsymbol{F}_{c,k}$ to the torque about the floating base when the SRB model is used. The constraint forces $\boldsymbol{F}_{\text{SRB},k}$ and $\boldsymbol{F}_{\text{void},k}$ are set to zero when the corresponding model is not used:

$$-\boldsymbol{M} s_{\text{SRB},k} \leq \boldsymbol{F}_{\text{SRB},k} \leq \boldsymbol{M} s_{\text{SRB},k} \tag{20}$$

$$-\boldsymbol{M} s_{\text{void},k} \leq \boldsymbol{F}_{\text{void},k} \leq \boldsymbol{M} s_{\text{void},k}. \tag{21}$$

C. Finding the Optimal Model Schedule

Because our implementation of ($\mathcal{MHPC}$) uses full-body dynamics with holonomic constraints for each ROM, we cannot directly minimize the number of decision variables, as they are equal across the horizon. Instead, we use the dimensionality of each model $\dim(m)$ and minimize

$$\min_{\boldsymbol{\mathcal{S}}} \sum_{m \in \{\text{full,SRB,void}\}} \sum_{k=1}^{N} s_{m,k} \dim(m). \tag{22}$$

The dimensionality $\dim(m)$ depends on the number of DoFs of the robot and the size of the holonomic constraint defining each model:

$$\dim(\text{full}) = 2n_b + 3n_j \tag{23}$$

$$\dim(\text{SRB}) = 2n_b \tag{24}$$

$$\dim(\text{void}) = 0 \tag{25}$$

To simplify the computation of the closed-loop cost $V^{\boldsymbol{\mathcal{S}}}$ induced by ($\mathcal{MHPC}$), the full-body model is used at the first time-step. Additionally, we set the minimum horizon of the full-body model to two. This restriction allows us to directly use the first timestep of ($\mathcal{MHPC}$) as a coarse simulation step. As a result, solving ($\mathcal{MHPC}$) lets us approximate the closed-loop dynamics of the robot and the policy (2) using:

$$\{\boldsymbol{u}_k, \boldsymbol{x}_{k+1}\} \subset \arg\min_{\hat{\boldsymbol{\mathcal{X}}}} \hat{V}^{\boldsymbol{\mathcal{S}}}(\boldsymbol{x}_k). \tag{26}$$

By recursively solving (26), we generate a full-body trajectory over a horizon of N timesteps (as in Fig. 1), for which we evaluate the closed-loop cost (3). Hence, the inequality in ($\mathcal{MHPC}$) can be evaluated for every $\boldsymbol{\mathcal{S}}$ and we can find $\boldsymbol{\mathcal{S}}^*$ that minimizes (22).

V. RESULTS

This section first details the experimental setup and simulations that validate the method. Next, the results are presented.

A. Experimental Setup

We implement our approach for a walking gait using a 13-DoF planar biped based on the MIT Humanoid [26]. All optimizations are designed with a fixed gait schedule and an integration timestep of 50 ms. We use a horizon of $\hat{N} = 23$ time-steps for ($\mathcal{MHPC}$) and $N = 17$ (two walking steps) for the full-body optimization ($\mathcal{TO}$) and for computing $V^{\boldsymbol{\mathcal{S}}}$ in ($\mathcal{MS}$).

We fix the ordering of the models with the full-body model first, followed by the SRB and the void models, and we compute the relative cost error term (RCE) in ($\mathcal{MHPC}$) for all possible horizons of the full and SRB models, corresponding to 253 possibilities. We refer to a given model schedule by the tuple $(N_{\text{full}}, N_{\text{SRB}})$, corresponding to the number of timesteps used for each model.

We use CasADi to formulate our trajectory optimizations [27], and the non-linear interior-point solver Knitro 13.1 [28]. MHPC is simulated in MATLAB for all model schedules using ode45. The same dynamics model is used in both optimizations and simulations, but a different integration scheme is used: the TO has a fixed simulation step of 50 ms using trapezoidal integration, whereas the simulation uses a higher-order Runge-Kutta adaptive integration scheme, queries ($\mathcal{MHPC}$) at 100 Hz, and simulates foot touchdown impacts. In all simulations, the solution from the first control input of MHPC is used as feedforward torque, and two walking steps are simulated.

B. Approximate Closed-Loop Trajectories

Recursively solving (26) generates trajectories that approximate the closed-loop behavior of ($\mathcal{MHPC}$). This allows us to directly leverage TO in order to estimate the closed-loop cost for a given model schedule. For example, Fig. 1 compares body trajectories from ($\mathcal{MHPC}$) with the approximate closed-loop for two model schedules (6,4) and (6,0). We can see that for the (6,4) schedule, although the predicted SRB states continue to deviate from the optimal trajectory, the discrepancy of the closed-loop trajectory is significantly decreased compared to the (6, 0) schedule. In other words, accurate state-predictions are not critical for choosing ROMs.

C. Relative Cost Error

Fig. 2a illustrates the relative cost error (RCE) evaluated from the approximated closed-loop trajectory for each model schedule. There is a sharp change in RCE, at $N_{\text{full}} = 5$, above which many model schedules achieve low RCE and successful walking in simulation. The diagonal lines indicate isolines for the total horizon $N_{\text{full}} + N_{\text{SRB}}$. Note that the RCE is high below 1 walking step and rapidly drops when planning above 1 walking step. Additionally, some model schedules achieve negative RCE, implying that the cost from the approximated closed-loop trajectory is lower than from ($\mathcal{TO}$). This result could be explained by the presence of local minima, from which the more complex full-body ($\mathcal{TO}$) is more likely to suffer. In contrast, the accumulated cost $V^{\boldsymbol{\mathcal{S}}}$ from the closed-loop approximations is obtained by solving many smaller problems that are less prone to local minima.

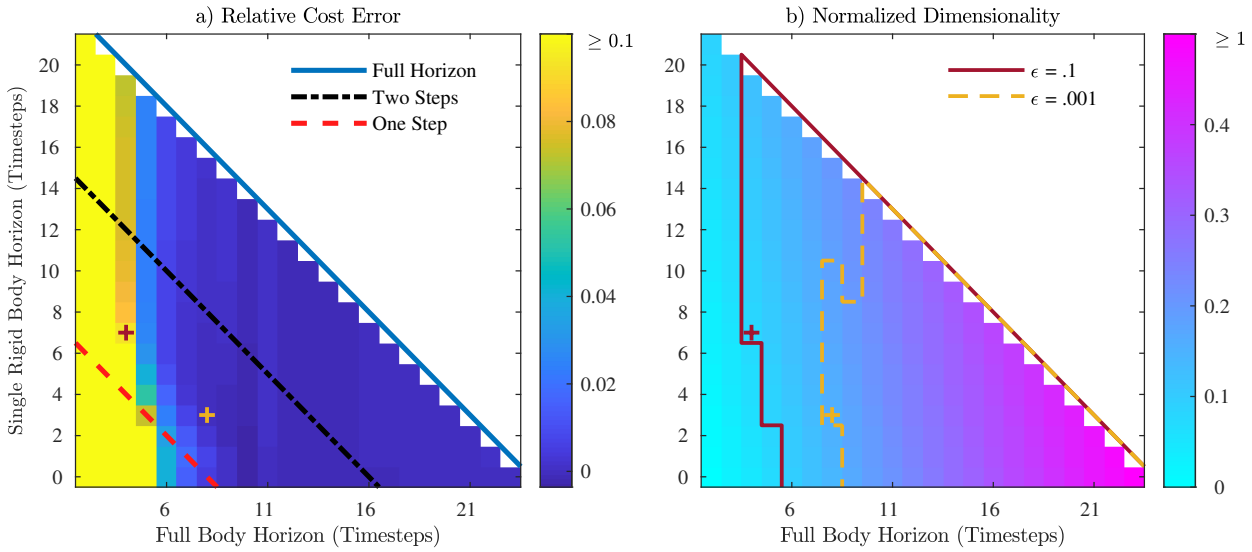

Fig. 2. a) Relative cost error and b) normalized dimensionality of ($\mathcal{MHPC}$) for all possible model schedules. The three diagonal lines in a) are isolines for the total horizon $N_{\text{full}} + N_{\text{SRB}}$ that highlight horizons of one walking step, two walking steps or the full horizon ($\hat{N} = 23$). In b), for each choice of ϵ, the lines delimit the feasible sets of model schedules from ($\mathcal{MS}$) and the "+" signs mark the optimal solution for each ϵ.

D. Optimal Model Schedules

The problem ($\mathcal{MS}$) solves for an optimal model schedule that minimizes the dimensionality of ($\mathcal{MHPC}$) while keeping the accumulated closed-loop cost of the robot close to optimal. Fig. 2b shows the dimensionality of each model schedule, which we wish to minimize. The boundaries delimit the feasible sets of model schedules for $\epsilon = 0.1$ and $\epsilon = 0.001$, as determined by the inequality constraint in ($\mathcal{MS}$). Within these sets, the optimal schedules are marked by "+" signs.

Table I gives the optimal model schedules and their dimensionality for $\epsilon \in \{0.1, 0.01, 0.001\}$. The solve times and the RCEs obtained using our TO-based framework is compared to the ones obtained in simulation. We find that the RCE from both TO and simulation monotonically decreases as dimensionality increases. This indicates improved performance with more complex model schedules in both simulation and TO-based approximations. On the other hand, while there is a relationship between solve times and dimensionality, the trends are not as clear. This may be because the ROM dynamics are enforced by introducing additional constraints rather than reducing the state-dimension (and thus the number of decision variables). Another factor could be the variability in solve times inherent to the general purpose non-linear solver. Regardless, this result highlights the need for a post-processing stage that removes the unnecessary decision variables before deploying the controller.

TABLE I
COMPARISON OF TO SOLUTIONS WITH SIMULATION RESULTS

Relative Cost Error Bound ϵ		0.1	0.01	0.001
Optimal Schedule ($N_{\text{full}}, N_{\text{SRB}}$)		(4,7)	(6,5)	(8,3)
Normalized Dimensionality		0.15	0.23	0.31
Relative Cost Error	TO	0.094	0.0098	-0.0003
	Sim	0.0419	0.0106	0.0085
Mean Solve Time (ms)	TO	378.2	135.4	164.4
	Sim	153.8	170.6	211.3

While we formulate ($\mathcal{MHPC}$) with a total of 23 timesteps, the proposed method finds optimal model schedules that substantially truncate the total horizon by making use of the void model. We solved ($\mathcal{MS}$) for 100 values of ϵ ranging from 0.01% to 3%, and the optimal model schedules always give total horizons $N_{\text{full}} + N_{\text{SRB}}$ between 10 and 14 timesteps, corresponding to between 1.1 and 1.6 walking steps. This result is in line with previous work where it was shown that planning beyond two walking steps does not provide a significant advantage [29].

E. Relative Cost Errors from TO and Simulation

The RCE obtained with the TO-based approximations varies similarly to the RCE from the simulations, even if they use different integration schemes and simulation frequencies. To illustrate this, we normalize both RCEs from TO and simulation by their respective maxima, and compute their difference. As shown in Fig. 3, the difference is close to zero for most model schedules, indicating that the shapes of each RCE closely match. Therefore, the closed-loop performance of ($\mathcal{MHPC}$) is well approximated by the TO-based approach. This result motivates the use of the proposed framework to first identify only a few model schedules before validating them in simulation.

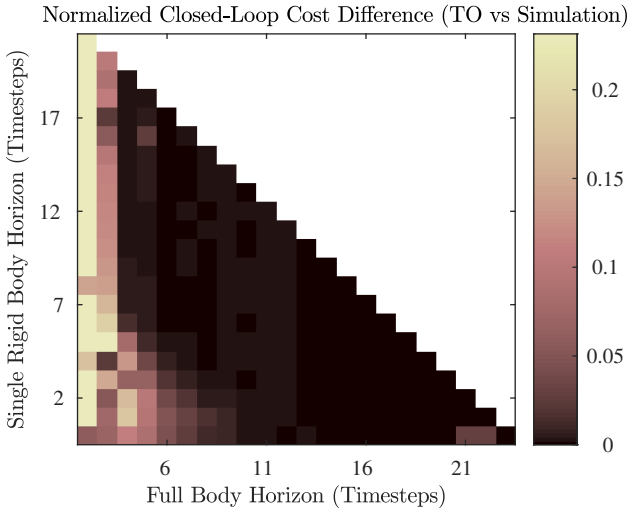

Fig. 3. Normalized error of accumulated cost between the approximate closed-loop ($\mathcal{MHPC}$) and simulated ($\mathcal{MHPC}$) trajectories for all model schedule possibilities. The absence of gradient for the error in most of the schedules indicate that the approximation is a good proxy for running ($\mathcal{MHPC}$) in a more accurate simulation.

VI. CONCLUSION AND OUTLOOK

We proposed a practical framework to optimize the model schedule for the planning horizon of MHPC. The method minimizes the dimensionality of the MHPC controller while keeping the closed-loop cost incurred by the robot close to optimal. We leverage trajectory optimization to estimate the closed-loop cost accumulated by the robot under a MHPC controller. The approach only requires the equations of motion of the full-body model and constraints for each ROM. We validated the approach with a proof of concept for a planar, 13-DoF humanoid robot, where we solved for optimal horizons of full-body and single rigid body models. We show that the shape of the accumulated cost estimated from TO matches the one obtained from higher-fidelity simulations.

A limitation of this work is that smaller problems did not consistently lead to faster solve times, the main cause likely being that we were not truly removing the decision variables from the MHPC optimizations, but rather adding constraints to the full-body model. This highlights the need to remove the decision variables in a post-processing stage.

Additionally, the number of possible model schedules was sufficiently small to solve ($\mathcal{MHPC}$) using an exhaustive search. In future work, we plan to make use of more efficient algorithms to scale the approach to a wider variety of reduced order models that arise when simplifying the whole-body dynamics equations. One possible approach is to solve ($\mathcal{MS}$) using gradient-free algorithms.

REFERENCES

[1] M. Zhong, M. Johnson, Y. Tassa, T. Erez, and E. Todorov, "Value function approximation and model predictive control," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2013.

[2] A. Parag, S. Kleff, L. Saci, N. Mansard, and O. Stasse, "Value learning from trajectory optimization and Sobolev descent: A step toward reinforcement learning with superlinear convergence properties," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[3] B. Landry, H. Dai, and M. Pavone, "SEAGuL: Sample Efficient Adversarially Guided Learning of Value Functions," in *Proceedings of the Conference on Learning for Dynamics and Control (L4DC)*, 2021.

[4] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control," in *IEEE/International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[5] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3D-SLIP model," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[6] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Footstep Placement," *Advanced Robotics*, 2010.

[7] H. Li, R. J. Frei, and P. M. Wensing, "Model Hierarchy Predictive Control of Robotic Systems," *IEEE Robotics and Automation Letters*, 2021.

[8] J. Wang, S. Kim, S. Vijayakumar, and S. Tonneau, "Multi-Fidelity Receding Horizon Planning for Multi-Contact Locomotion," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2021.

[9] C. Khazoom and S. Kim, "Humanoid Arm Motion Planning for Improved Disturbance Recovery Using Model Hierarchy Predictive Control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[10] V. A. Laurense and J. C. Gerdes, "Long-Horizon Vehicle Motion Planning and Control Through Serially Cascaded Model Complexity," *IEEE Transactions on Control Systems Technology*, 2022.

[11] M. Posa and R. Tedrake, "Direct Trajectory Optimization of Rigid Body Dynamical Systems through Contact," in *Algorithmic Foundations of Robotics*, Springer, 2013.

[12] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao, "Momentum-Aware Trajectory Optimization and Control for Agile Quadrupedal Locomotion," *IEEE Robotics and Automation Letters*, 2022.

[13] R. Full and D. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, 1999.

[14] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, 1989.

[15] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE/ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[16] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, 2012.

[17] I. Poulakakis and J. W. Grizzle, "The Spring Loaded Inverted Pendulum as the Hybrid Zero Dynamics of an Asymmetric Hopper," *IEEE Transactions on Automatic Control*, 2009.

[18] X. Xiong and A. D. Ames, "Dynamic and Versatile Humanoid Walking via Embedding 3D Actuated SLIP Model With Hybrid LIP Based Stepping," *IEEE Robotics and Automation Letters*, 2020.

[19] S. Faraji and A. J. Ijspeert, "3LP: A linear 3D-walking model including torso and swing dynamics," *The International Journal of Robotics Research*, 2017.

[20] T. Apgar, P. Clary, K. Green, A. Fern, and J. Hurst, "Fast Online Trajectory Optimization for the Bipedal Robot Cassie," in *Robotics: Science and Systems (RSS)*, 2018.

[21] Y. Ding, C. Khazoom, M. Chignoli, and S. Kim, "Orientation-aware model predictive control with footstep adaptation for dynamic humanoid walking," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2022.

[22] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control," *arXiv:1909.06586 [cs]*, 2019.

[23] Y.-M. Chen and M. Posa, "Optimal Reduced-order Modeling of Bipedal Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[24] A. Jain, L. Chan, D. S. Brown, and A. D. Dragan, "Optimal cost design for model predictive control," in *Proceedings of the Conference on Learning for Dynamics and Control (L4DC)*, 2021.

[25] D.-S. Chen, R. G. Batson, and Y. Dang, *Applied integer programming: modeling and solution*. John Wiley & Sons, 2011.

[26] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2021.

[27] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2019.

[28] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*. 2006.

[29] P. Zaytsev, S. J. Hasaneini, and A. Ruina, "Two steps is enough: No need to plan far ahead for walking balance," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.