Formal Connections between Template and Anchor Models via Approximate Simulation

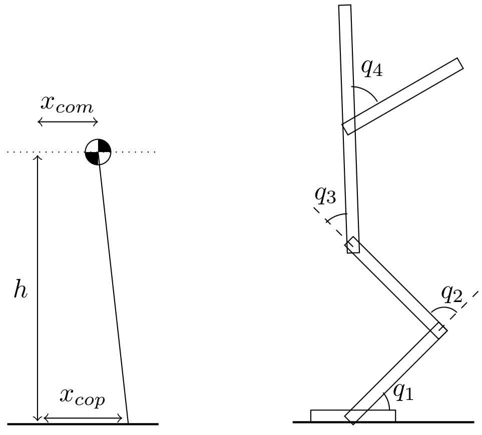
Vince Kurtz, Rafael Rodrigues da Silva, Patrick M. Wensing, and Hai Lin

***Abstract*—Reduced-order template models like the Linear Inverted Pendulum (LIP) and Spring-Loaded Inverted Pendulum (SLIP) are widely used tools for controlling high-dimensional humanoid robots. However, connections between templates and whole-body models have lacked formal underpinnings, preventing formal guarantees when it comes to integrated controller design. We take a small step towards addressing this gap by considering the notion of *approximate simulation*. Derived from simulation relations for discrete transition systems in formal methods, approximate similarity means that the outputs of two systems can remain ϵ-close. In this paper, we consider the case of controlling a balancer via planning with the LIP model. We show that the balancer approximately simulates the LIP and derive linear constraints that are sufficient conditions for maintaining ground contact. This allows for rapid planning and replanning with the template model by solving a quadratic program that enforces contact constraints in the full model. We demonstrate the efficacy of this planning and control paradigm in a simulated push recovery scenario for a planar 4-link balancer.**

I. Introduction

Template models like the Linear Inverted Pendulum (LIP), Spring-Loaded Inverted Pendulum (SLIP), and Compass-Gait Walker are widely used for analysis, planning, and control of locomotion and balance [1–6]. These models capture important properties of the full system, or anchor model, while also being simple enough to enable efficient planning. Often, the template model is used to generate a plan via Model Predictive Control (MPC) (e.g., [7, 8]) while the anchor model tracks this nominal trajectory, subject to contact constraints. The template-anchor control paradigm works well in practice, and is widely accepted in the robotics community [1]. Furthermore, it has been hypothesized that biological systems use templates as sensorimotor control targets: animals may regulate their center-of-mass (CoM) to follow a lower-order model [9].

However, a key weakness of the template-anchor paradigm is a lack of formal correctness. So far, formal connections between template and anchor models have not been identified. Beyond the insights that might come from identifying such connections, it is also difficult to provide provably correct whole-body controllers that track the template model while also accounting for constraints on contact forces, joint limits, joint torques, etc.



(a) Linear Inverted Pendulum (b) Planar Balancer

Fig. 1. Template (linear inverted pendulum) and anchor (balancer in single support) models considered in this paper. We show that the balancer approximately simulates the LIP, which allows us to bound the tracking error between the two models.

The state-of-the-art in this regard is to formulate the tracking problem as a Quadratic Program (QP), which partial-feedback linearizes the system dynamics while also encoding contact, friction, and torque constraints [10–12]. Such QPs can be solved at kHz rates for high-dimensional rigid-body models, and have been applied to great effect in the control of humanoid robots [13–15]. But this approach does not admit formal guarantees: we cannot ensure that the solver will always find a solution, or that the optimal solution will cause the anchor to converge to the template trajectory (though this property holds empirically under normal conditions with appropriately tuned controllers).

An alternative, more formal, approach is to restrict the anchor model to operate on a lower-dimensional manifold that is diffeomorphic to the template dynamics. This approach was taken by Poulakakis and Grizzle [16] in the context of an asymmetric hopper wherein the SLIP model was embedded as the Hybrid Zero Dynamics [17] of the monopod. This approach admits strong formal guarantees, but finding a whole-body controller that enforces this diffeomorphic relationship is difficult, especially for high-dimensional systems like humanoids. Furthermore, it is challenging to repurpose redundant degrees of freedom for other tasks (e.g., holding a cup of coffee while walking) under this approach.

In this work, we present an alternative method of providing

The partial support of the National Science Foundation (Grant No. ECCS-1253488, IIS-1724070, CNS-1830335, CMMI-1835186) is gratefully acknowledged.

V. Kurtz, R. Silva, and H. Lin are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 46556 USA. {vkurtz,rrodri17,hlin1}@nd.edu.

P.M. Wensing is with the Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN, 46556 USA. pwensing@nd.edu.

formal guarantees for control with template models. Specifically, we consider the paradigm of *approximate simulation* [18]. This concept is a generalization of simulation relations from formal methods [19] to continuous systems. If the anchor approximately simulates the template, then there exists a controller which tracks any trajectory of the template model with ϵ precision. Such a controller is known as an *interface*. For any approximate simulation relation, there is also a *simulation function*, a Lyapunov-like function that bounds the error between the two models.

The contributions of this paper are as follows. First, we show that classical whole-body controllers based on task-space feedback linearization function as an interface, which proves that a balancer in single support approximately simulates a LIP. However, this simulation relation ignores important constraints that exist in whole-body robot models (e.g., contact constraints, joint limits, torque limits, etc.). As a second contribution, we introduce a new MPC scheme that uses the simulation relationship to address contact wrench cone (CWC) [20] constraints during planning with the LIP. Inspired by centroidal momentum planning methods [21, 22], we address non-convexity in this MPC scheme by introducing a convex inner approximation to the CWC constraints. This inner approximation allows us to perform MPC for the LIP model by solving a QP that already accounts for contact constraints in the whole-body model, as shown in Figure 2. In contrast to existing work [23], this new method does not require an admissible region of the CoM to be specified a-priori.

The remainder of this paper is organized as follows: Section II presents a formal definition of approximate simulation and introduces relevant results for linear systems. Section III provides system definitions and a problem statement. Section IV presents our principal results, which are illustrated with a simulated push-recovery scenario in Section V. We provide a brief discussion in Section VI, and conclude with Section VII.

II. Background

A. Approximate Simulation

Approximate simulation is formally defined in terms of two systems Σ_1 and Σ_2:

$$\Sigma_1 : \begin{cases} \dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, \mathbf{u}_1) \\ \mathbf{y}_1 = g_2(\mathbf{x}_1) \end{cases} , \quad \Sigma_2 : \begin{cases} \dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2, \mathbf{u}_2) \\ \mathbf{y}_2 = g_2(\mathbf{x}_2) \end{cases} , \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$ are the system states, $\mathbf{u}_i \in \mathbb{R}^{p_i}$ are the control inputs, and $\mathbf{y}_i \in \mathbb{R}^{m}$ are the system outputs. Note that the states may be different sizes but the outputs—which in our case correspond to the task-space—must be the same size. Without loss of generality, we consider Σ_1 to be the whole-body (anchor) model and Σ_2 to be the reduced-order (template) model.

Approximate simulation for continuous systems [18] is defined in terms of a Lyapunov-like simulation function $\mathcal{V}$ and an interface function $u_{\mathcal{V}}$:

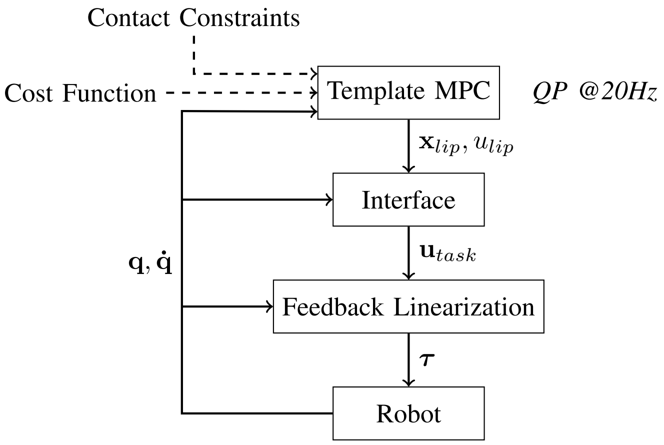


Fig. 2. Control flow for our approach. While most whole-body controllers account for contact constraints in the feedback linearization phase with a QP, we project contact constraints back to the template model using the approximate simulation relation. Note that $(\mathbf{q}, \dot{\mathbf{q}})$ are used only to compute the task state $\mathbf{x}_{task}$ in the Template MPC block.

Definition 1. *Let $\mathcal{V} : \mathbb{R}^{n_2} \times \mathbb{R}^{n_1} \to \mathbb{R}^{+}$ be a smooth function and $u_{\mathcal{V}} : \mathbb{R}^{p_2} \times \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{p_1}$ be a continuous function. $\mathcal{V}$ is a simulation function of Σ_2 by Σ_1 and $u_{\mathcal{V}}$ is an associated interface if there exists a class-κ function*[1] *γ such that for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$,*

$$\mathcal{V}(\mathbf{x}_1, \mathbf{x}_2) \geq \|g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)\| \tag{2}$$

and for all $\mathbf{u}_2 \in \mathbb{R}^{p_2}$ satisfying $\gamma(\|\mathbf{u}_2\|) < \mathcal{V}(\mathbf{x}_1, \mathbf{x}_2)$,

$$\frac{\partial \mathcal{V}}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, \mathbf{u}_2) + \frac{\partial \mathcal{V}}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_{\mathcal{V}}(\mathbf{u}_2, \mathbf{x}_1, \mathbf{x}_2)) < 0. \tag{3}$$

These conditions essentially state that when the interface is applied, the simulation function always bounds the output error (2) and decreases as long as $\mathbf{u}_2$ is not too large (3).

Definition 2 ([18]). *Σ_1 approximately simulates Σ_2 if and only if there exists a simulation function $\mathcal{V}$ of Σ_2 by Σ_1.*

If Σ_1 approximately simulates Σ_2, we can use the simulation function to bound the output error of the two systems by ϵ:

Theorem 1 ([24]). *Let $\mathcal{V}$ be a simulation function of Σ_2 by Σ_1 and $u_{\mathcal{V}}$ be an associated interface. Let $\mathbf{u}_2(t)$ be an admissible input of Σ_2 with associated state and output trajectories $\mathbf{x}_2(t)$ and $\mathbf{y}_2(t)$. Let $\mathbf{x}_1(t)$ be a state trajectory of Σ_1 satisfying*

$$\dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, u_{\mathcal{V}}(\mathbf{u}_2, \mathbf{x}_1, \mathbf{x}_2))$$

and $\mathbf{y}_1(t)$ be the associated output trajectory. Then

$$\|\mathbf{y}_1(t) - \mathbf{y}_2(t)\| \leq \epsilon \tag{4}$$

where

$$\epsilon = \max\left\{\mathcal{V}(\mathbf{x}_1(0), \mathbf{x}_2(0)), \gamma(\|\mathbf{u}_2\|_\infty)\right\}. \tag{5}$$

[1] A function $\gamma : \mathbb{R}^{+} \to \mathbb{R}^{+}$ is a class-κ function if it is continuous, strictly increasing, and $\gamma(0) = 0$.

Finding a simulation function for two arbitrary dynamical systems is a difficult and open problem, though some promising results with sum-of-squares programming exist [25, 26]. For linear systems, however, there are well-defined conditions for the existence of a simulation function [24]. We summarize these conditions in the following subsection.

B. Approximate Simulation for Linear Systems

Consider the case when both the template and the anchor are linear systems, i.e.,

$$\Sigma_i : \begin{cases} \dot{\mathbf{x}}_i = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i \mathbf{u}_i \\ \mathbf{y}_i = \mathbf{C}_i \mathbf{x}_i \end{cases}, \quad i = \{1, 2\}. \tag{6}$$

In this case, there are strong results regarding whether a simulation relation exists. First note the following Lemma:

Lemma 1 (Girard and Pappas [27])**.** *If the anchor system Σ_1 is stabilizable with feedback gain $\mathbf{K}$, i.e., $(\mathbf{A}_1 + \mathbf{B}_1\mathbf{K})$ is Hurwitz, then there exists a positive definite symmetric matrix $\mathbf{M}$ and positive scalar constant λ such that the following hold:*

$$\mathbf{M} \geq \mathbf{C}_1^T \mathbf{C}_1, \tag{7}$$

$$(\mathbf{A}_1 + \mathbf{B}_1\mathbf{K})^T \mathbf{M} + \mathbf{M}(\mathbf{A}_1 + \mathbf{B}_1\mathbf{K}) \leq -2\lambda\mathbf{M}. \tag{8}$$

Such an $\mathbf{M}$ can be used to show exponential convergence of $\mathbf{y}_1$ to zero with rate λ under the feedback $\mathbf{u}_1 = \mathbf{K}\mathbf{x}_1$. Note that these conditions are linear matrix inequalities in $\mathbf{M}$. This means that given $\mathbf{K}$ and λ, $\mathbf{M}$ can be computed using semidefinite programming.

We can now state the following Theorem:

Theorem 2 (Girard and Pappas [24])**.** *Assume that Σ_1 is stabilizable with feedback gain $\mathbf{K}$ and that there exist matrices $\mathbf{P}$ and $\mathbf{Q}$ such that the following conditions hold:*

$$\mathbf{P}\mathbf{A}_2 = \mathbf{A}_1\mathbf{P} + \mathbf{B}_1\mathbf{Q}, \tag{9}$$

$$\mathbf{C}_2 = \mathbf{C}_1\mathbf{P}. \tag{10}$$

Then a simulation function of Σ_2 by Σ_1 is given by

$$\mathcal{V}(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{P}\mathbf{x}_2)^T \mathbf{M} (\mathbf{x}_1 - \mathbf{P}\mathbf{x}_2)}, \tag{11}$$

an associated interface is

$$u_{\mathcal{V}} = \mathbf{R}\mathbf{u}_2 + \mathbf{Q}\mathbf{x}_2 + \mathbf{K}(\mathbf{x}_1 - \mathbf{P}\mathbf{x}_2), \tag{12}$$

and the class-κ function γ is given by

$$\gamma(\nu) = \frac{\|\sqrt{\mathbf{M}}(\mathbf{B}_1\mathbf{R} - \mathbf{P}\mathbf{B}_2)\|}{\lambda}\nu, \tag{13}$$

where $\mathbf{R}$ is an arbitrary matrix of proper dimensions.

The matrix $\mathbf{R}$ acts as a "feedforward" mapping from $\mathbf{u}_2$ to $\mathbf{u}_1$. While the simulation relation holds for any $\mathbf{R}$ of proper dimensions, choosing $\mathbf{R}$ to minimize (13) is a logical choice, as this tightens the error bound ϵ (5).

III. Problem Formulation

A. System Definitions

In this paper, we consider controlling a balancer in single support using the LIP model as a template.

An example of a balancer used as the anchor model is shown in Figure 1b. We assume that all joints are actuated and that the system is mounted to a single foot in contact. By modeling the balancer as a kinematic tree, we can write the equations of motion with minimal coordinates in "manipulator" form:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_g = \boldsymbol{\tau} \tag{14}$$

where $\mathbf{q}$ are joint angles, $\mathbf{H}(\mathbf{q})$ is the mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ accounts for Coriolis and centripetal terms, $\boldsymbol{\tau}_g$ is torque due to gravity, and $\boldsymbol{\tau}$ are applied torques. The gravitational vector is denoted $\mathbf{g}$ and has magnitude g. The total mass of the robot is m. Note that (14) is valid only if contact constraints are not violated.

We consider the CoM position $\mathbf{p}_G \in \mathbb{R}^3$ and centroidal momentum $\mathbf{h}_G = [\mathbf{k}_G^T \; \mathbf{l}_G^T]^T \in \mathbb{R}^6$ as the task-space of the anchor model:

$$\mathbf{x}_{task} = \begin{bmatrix} \mathbf{p}_G^T & \mathbf{h}_G^T \end{bmatrix}^T,$$

where $\mathbf{k}_G$ is the angular momentum about the CoM and $\mathbf{l}_G$ is the net linear momentum. The task-space dynamics can be computed in terms of the centroidal dynamics [28, 29]:

$$\dot{\mathbf{p}}_G = \frac{1}{m}\mathbf{l}_G, \tag{15}$$

$$\mathbf{h}_G = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}}, \tag{16}$$

$$\dot{\mathbf{h}}_G = \mathbf{A}_G(\mathbf{q})\,\ddot{\mathbf{q}} + \dot{\mathbf{A}}_G(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}}. \tag{17}$$

The LIP model [3], shown in Figure 1a, is constrained to the (x, z) plane and governed by the horizontal position of the CoM x_{com} and the center of pressure x_{cop}. The vertical position of the CoM is fixed at height h at all times. The dynamics of the LIP model are given by

$$\ddot{x}_{com} = \omega^2(x_{com} - x_{cop}), \tag{18}$$

where $\omega = \sqrt{\frac{g}{h}}$ is the natural frequency of the LIP.

For easier comparison with the task-space of the anchor model, we can write the LIP dynamics as a linear system

$$\dot{\mathbf{x}}_{lip} = \mathbf{A}_{lip}\mathbf{x}_{lip} + \mathbf{B}_{lip}u_{lip}, \tag{19}$$

where

$$\mathbf{x}_{lip} = \begin{bmatrix} x_{com} & 0 & h & \vdots & 0 & 0 & 0 & \vdots & m\dot{x}_{com} & 0 & 0 \end{bmatrix}^T$$

expresses the CoM position and spatial momentum of the LIP model and $u_{lip} = x_{cop}$ is the x-position of the center of pressure. While it may seem odd to include static elements like angular momentum in the LIP model, this particular definition of $\mathbf{x}_{lip}$ will allow us to draw a connection between an interface that certifies approximate simulation and the PD control law often used to track template models.

B. Problem Statement

As per Section II-A, finding an interface and a simulation function allows us to bound the output error between two systems. With this in mind, our primary goal is to certify that the balancer approximately simulates the LIP.

More formally, we define the output of the template (LIP) model to be the full system state, i.e.,

$$\mathbf{y}_{lip} = \mathbf{x}_{lip},$$

and the output of the anchor model to be the task-space (position and spatial momentum of the CoM):

$$\mathbf{y}_{task} = \mathbf{x}_{task} = \begin{bmatrix}\mathbf{p}_G^T & \mathbf{h}_G^T\end{bmatrix}^T.$$

Our goal is to find a simulation function and an interface that certify that the anchor model (14) approximately simulates the template model (19).

Finding such an interface and simulation function would allow us to guarantee that the CoM of the balancer tracks the LIP's CoM and bound the associated tracking error, subject to constraints on contact, torques, joint limits, etc. This leads us to our secondary goal, which is to use the simulation relation to project contact constraints for the balancer to constraints on the LIP model. That way, when we plan with the LIP model, we can be sure that the balancer will maintain ground contact.

IV. Theoretical Results

A. Approximate Simulation for the LIP and Balancer

In order to harness the results presented in Section II-B, we take a task-space feedback linearization of the balancer. To do so, we define a task-space interia matrix

$$\mathbf{\Lambda} = (\mathbf{A}_G \mathbf{H}^{-1} \mathbf{A}_G^T)^{-1} \tag{20}$$

and apply torques $\boldsymbol{\tau}$ such that

$$\boldsymbol{\tau} = \mathbf{A}_G^T \mathbf{\Lambda}(\mathbf{u}_{task} - \dot{\mathbf{A}}_G \dot{\mathbf{q}} + \mathbf{A}_G \mathbf{H}^{-1}(\mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g)), \tag{21}$$

where $\mathbf{u}_{task} = \dot{\mathbf{h}}_G$ is a virtual control. This gives rise to the task-space dynamics

$$\dot{\mathbf{x}}_{task} = \mathbf{A}_{task}\mathbf{x}_{task} + \mathbf{B}_{task}\mathbf{u}_{task}, \tag{22}$$

where

$$\mathbf{A}_{task} = \begin{bmatrix}\mathbf{0}_{3\times 6} & (1/m)\mathbf{I}_{3\times 3} \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 3}\end{bmatrix}, \quad \mathbf{B}_{task} = \begin{bmatrix}\mathbf{0}_{3\times 6} \\ \mathbf{I}_{6\times 6}\end{bmatrix}.$$

In the case where $dim(\mathbf{q}) > dim(\mathbf{x}_{task})$, we can resolve redundancies in the standard manner via the null-space projector N, such that applying $\boldsymbol{\tau} + N\boldsymbol{\tau}_0$ has the same effect on the task-space as applying $\boldsymbol{\tau}$ alone [30]. This allows us to design $\boldsymbol{\tau}_0$ to achieve secondary control objectives like reducing extraneous motion or controlling a certain limb. In our example, we designed $\boldsymbol{\tau}_0$ to regulate the balancer to the nominal pose shown in Figure 1b.

Now we have a linear system that describes the evolution of the planar balancer's CoM (22), and another linear system describing the evolution of the LIP's CoM (19). Taking the outputs as described above, i.e., $\mathbf{C}_{task} = \mathbf{C}_{lip} = \mathbf{I}_{9\times 9}$, we can use the results from Section II-B to find a simulation function and an associated interface.

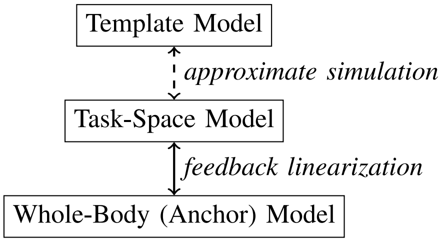


Fig. 3. Heirarchy of models used in our approach. We show that the whole body (anchor) model simulates the template model via a task-space feedback linearization.

First, we find a feedback control gain matrix $\mathbf{K}$ to stabilize the centroidal dynamics (22). There are many techniques to choose such a gain: LQR can be used to find a $\mathbf{K}$ that is optimal with respect to a certain cost functional. Given $\mathbf{K}$, we can use semi-definite programming to find $\mathbf{M}$ following Lemma 1.

Then, following Theorem 2, we define the matrices $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$ as follows:

$$\mathbf{P} = \mathbf{I}_{9\times 9}, \quad \mathbf{Q} = \begin{bmatrix}\begin{bmatrix}\mathbf{0}_{3\times 1} \\ \omega^2 \\ \mathbf{0}_{2\times 1}\end{bmatrix} & \mathbf{0}_{6\times 8}\end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix}\mathbf{0}_{3\times 1} \\ -\omega^2 \\ \mathbf{0}_{2\times 1}\end{bmatrix}.$$

This choice of $\mathbf{P}$ and $\mathbf{Q}$ satisfies the conditions of Theorem 2. This particular choice of $\mathbf{R}$ not only minimizes (13), but also establishes a connection between the interface and the PD controller often used in practice.

To see this connection, consider a robot with mass $m = 1$. A controller for the feedback-linearized anchor model is often designed as follows:

$$\mathbf{u}_{task} = \begin{bmatrix}-\mathbf{K}_{ang}\mathbf{k}_G \\ \ddot{\mathbf{p}}_G\end{bmatrix}, \tag{23}$$

$$\ddot{\mathbf{p}}_G = \begin{bmatrix}\ddot{x}_{com} \\ \mathbf{0}_{2\times 1}\end{bmatrix} + \mathbf{K}_D(\dot{\mathbf{p}}_{lip} - \dot{\mathbf{p}}_G) + \mathbf{K}_P(\mathbf{p}_{lip} - \mathbf{p}_G), \tag{24}$$

where $\mathbf{p}_{lip}, \dot{\mathbf{p}}_{lip}$ are the CoM positions and velocities of the LIP, $\mathbf{K}_{ang}$ is an angular momentum damping term, and $\mathbf{K}_P$ and $\mathbf{K}_D$ are matrices of tuned control gains. Recalling that $\ddot{x}_{com} = \omega^2(x_{com} - u_{lip})$, we can rewrite the first term of (23) as

$$\begin{aligned}\begin{bmatrix}\mathbf{0}_{3\times 1} \\ \ddot{x}_{com} \\ \mathbf{0}_{2\times 1}\end{bmatrix} &= \begin{bmatrix}\mathbf{0} \\ \omega^2(x_{com} - u_{lip}) \\ \mathbf{0}\end{bmatrix} \\ &= \mathbf{R}u_{lip} + \mathbf{Q}\mathbf{x}_{lip}.\end{aligned}$$

Similarly, we can use the fact that $\mathbf{x}_{lip} = [\mathbf{p}_{lip}^T \ \mathbf{0}_{3\times 1}^T \ \dot{\mathbf{p}}_{lip}^T]$ and $\mathbf{x}_{task} = [\mathbf{p}_G^T \ \mathbf{k}_G^T \ \dot{\mathbf{p}}_G^T]$ to rearrange the PD gain terms as

$$\begin{bmatrix}\mathbf{K}_{ang}\mathbf{k}_G \\ \mathbf{K}_D(\dot{\mathbf{p}}_{lip} - \dot{\mathbf{p}}_G) + \mathbf{K}_P(\mathbf{p}_{lip} - \mathbf{p}_G)\end{bmatrix} = \mathbf{K}(\mathbf{x}_{task} - \mathbf{x}_{lip}),$$

recovering the familiar form of the interface

$$\mathbf{u}_{task} = \mathbf{R}u_{lip} + \mathbf{Q}\mathbf{x}_{lip} + \mathbf{K}(\mathbf{x}_{task} - \mathbf{x}_{lip}). \tag{25}$$

Thus the commonly used task-space PD controller is in fact a special case of an interface that admits an approximate simulation relation.

Regardless of whether the stabilizing gain matrix $\mathbf{K}$ is generated as a PD controller or via techniques like LQR, the resulting simulation function takes the form

$$\mathcal{V}(\mathbf{x}_{task}, \mathbf{x}_{lip}) = \sqrt{(\mathbf{x}_{task} - \mathbf{x}_{lip})^T \mathbf{M} (\mathbf{x}_{task} - \mathbf{x}_{lip})}, \quad (26)$$

where $\mathbf{M}$ can be computed with semi-definite programming as per Lemma 1.

Recall from Theorem 1 that the simulation function $\mathcal{V}$ always bounds the output error, which in this case is $\|\mathbf{x}_{task} - \mathbf{x}_{lip}\|$. Furthermore, $\mathcal{V}$ always decreases, except when $\gamma(\|u_{lip}\|) \geq \mathcal{V}(\mathbf{x}_{task}, \mathbf{x}_{lip})$. From Theorem 2, we have

$$\begin{aligned} \gamma(\nu) &= \frac{\|\sqrt{\mathbf{M}}(\mathbf{B}_{task}\mathbf{R} - \mathbf{P}\mathbf{B}_{lip})\|}{\lambda}\nu \\ &= \frac{\|\sqrt{\mathbf{M}}(0)\|}{\lambda}\nu = 0. \end{aligned}$$

In other words, $\mathcal{V}$ is decreasing along all trajectories of the LIP model, regardless of the value of u_{lip}.

B. Projecting Contact Constraints to the Template

When we use this simulation relation to control the anchor system, we do have some restrictions. Intuitively, the CoM cannot follow arbitary trajectories: at the very least, it cannot reside beyond the limits imposed by a fixed ground contact. As an initial step toward handling such constraints, we show that contact constraints on the anchor model can be reformulated as linear constraints for MPC planning with the template model.

The Contact Wrench Cone (CWC) contact constraint states that all contact forces must remain in cones defined by a Coulomb friction model [20]. In terms of the spatial force expressed at the ground frame $\mathbf{f}_0$, the CWC can be expressed as

$$CWC = \left\{ \mathbf{f}_0 \mid \mathbf{f}_0 = \sum_j \begin{bmatrix} \mathbf{S}(\mathbf{p}_{c_j}) \\ \mathbf{I} \end{bmatrix} \boldsymbol{f}_{c_j}, \left\| \begin{bmatrix} f^x_{c_j} \\ f^y_{c_j} \end{bmatrix} \right\| \leq \mu f^z_{c_j} \right\}, \quad (27)$$

where c_j are ground contacts, $\mathbf{p}_{c_j}$ are their associated positions in the 0 frame, $\boldsymbol{f}_{c_j} \in \mathbb{R}^3$ are ground contact forces, μ is the coefficient of friction, and $\mathbf{S}(\cdot)$ is the skew-symmetric cross product matrix.

We can express the CWC constraint in terms of $\mathbf{u}_{task} = \dot{\mathbf{h}}_G$:

$${}^0\mathbf{X}^*_G \left(\mathbf{u}_{task} - \begin{bmatrix} \mathbf{0} \\ m\mathbf{g} \end{bmatrix} \right) \in CWC,$$

where ${}^0\mathbf{X}^*_G$ is the spatial force transform from the CoM frame $\{G\}$ to the ground frame $\{0\}$ given by

$${}^0\mathbf{X}^*_G = \begin{bmatrix} \mathbf{I} & \mathbf{S}(\mathbf{p}_G) \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

If we consider friction pyramids as inner approximations of friction cones, we can derive a polytopic under-approximation of the CWC [31, 32], i.e., $\mathbf{A}\mathbf{f}_0 \leq 0$. In this case, the CWC criterion can be written as a bilinear constraint on $\mathbf{x}_{task}$, $\mathbf{u}_{task}$

$$\mathbf{A}^0\mathbf{X}^*_G(\mathbf{u}_{task} - \begin{bmatrix} \mathbf{0} \\ m\mathbf{g} \end{bmatrix}) \leq 0, \quad (28)$$

where the bilinearity arises due to the dependence of ${}^0\mathbf{X}^*_G$ on a cross-product term of $\mathbf{p}_G$.

However, if we constrain the CoM acceleration $\dot{\mathbf{l}}_G$, we can formulate the CWC criterion as a linear constraint on $\mathbf{u}_{task}$ and $\mathbf{x}_{task}$ [22]. This is shown by the following Theorem:

Theorem 3. *If the CoM acceleration $\dot{\mathbf{l}}_G$ of the anchor model is constrained by $\|\dot{\mathbf{l}}_G\|_\infty \leq \dot{l}_{max}$, then there exists a linear constraint $\mathbf{A}_{cwc} \begin{bmatrix} \mathbf{x}_{task} \\ \mathbf{u}_{task} \end{bmatrix} \leq \mathbf{b}_{cwc}$ that is a sufficient condition for the CWC contact criterion.*

Proof. We will prove by construction. First, recall that $\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b} = -\mathbf{S}(\mathbf{b})\mathbf{a}$. With this in mind, the CWC criterion (28) can be written as

$$\mathbf{A}^0\mathbf{X}^*_G\dot{\mathbf{h}}_G \leq \mathbf{A}^0\mathbf{X}^*_G \begin{bmatrix} \mathbf{0} \\ m\mathbf{g} \end{bmatrix} \quad (29)$$

$$\mathbf{A} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \dot{\mathbf{k}}_G + \mathbf{A} \begin{bmatrix} \mathbf{S}(\mathbf{p}_G) \\ \mathbf{I} \end{bmatrix} \dot{\mathbf{l}}_G \leq \mathbf{A} \begin{bmatrix} \mathbf{I} & \mathbf{S}(\mathbf{p}_G) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ m\mathbf{g} \end{bmatrix} \quad (30)$$

$$\mathbf{A}\dot{\mathbf{h}}_G + \mathbf{A} \begin{bmatrix} \mathbf{S}(m\mathbf{g}) - \mathbf{S}(\dot{\mathbf{l}}_G) \\ \mathbf{0} \end{bmatrix} \mathbf{p}_G \leq \mathbf{A} \begin{bmatrix} \mathbf{0} \\ m\mathbf{g} \end{bmatrix} \quad (31)$$

Noting that $m\mathbf{g}$ and $\mathbf{A}$ are fixed for a given ground contact, the only remaining nonconvexity is from the $\mathbf{S}(\dot{\mathbf{l}}_G)$ term.

Note that the left hand side of (31) is bilinear in $[\mathbf{x}^T_{task} \;\; \mathbf{u}^T_{task}]^T$ but linear in $\dot{\mathbf{l}}_G$ alone. Furthermore, the constraint $\|\dot{\mathbf{l}}_G\|_\infty \leq \dot{l}_{max}$ defines a polytope (specifically, a cube in $\mathbb{R}^3$). Linear functions constrained to polytopes have extrema at the vertices, so we can create a linear inner approximation of (31) by enforcing (31) for $\dot{\mathbf{l}}_G$ evaluated at all 8 corners of the cube defined by $\|\dot{\mathbf{l}}_G\|_\infty \leq \dot{l}_{max}$.

This new constraint is linear in the position of the CoM $\mathbf{p}_G$ and the time derivative of the centroidal momentum $\dot{\mathbf{h}}_G$, allowing us to write it as a constraint of the form

$$\mathbf{A}_{cwc} \begin{bmatrix} \mathbf{x}_{task} \\ \mathbf{u}_{task} \end{bmatrix} \leq \mathbf{b}_{cwc}. \quad (32)$$

□

This linear encoding of the CWC criterion allows us to perform MPC for the template model by solving a QP. To account for contact constraints, we consider the anchor variables $\mathbf{x}_{task}, \mathbf{u}_{task}$ as additional optimization variables, and enforce the interface (25) as a constraint. We then

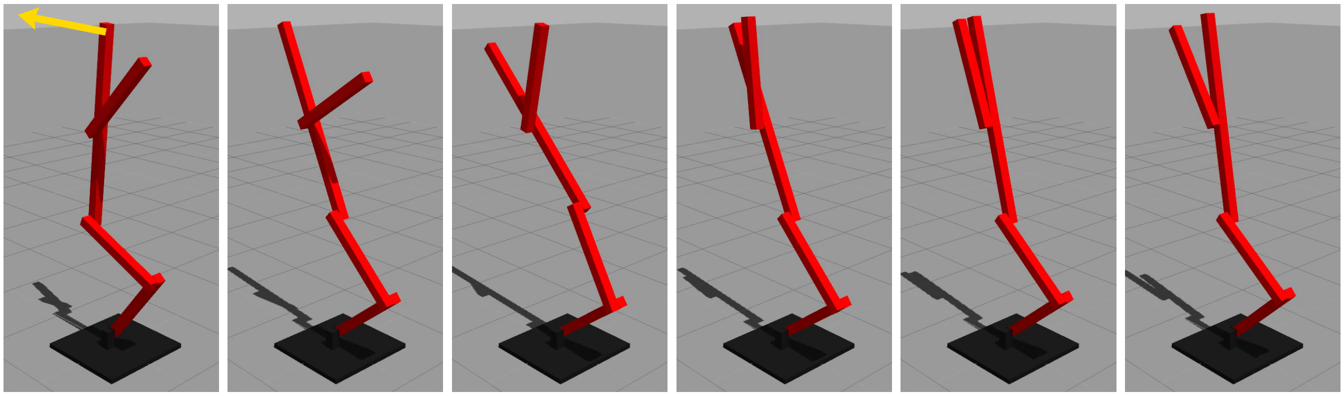


Fig. 4. Snapshots taken at 1Hz as the multi-link balancer recovers from an initial push via planning with the LIP model. Using the approximate simulation relation between the balancer and the LIP allows us to guarantee that the balancer will track the LIP trajectory and maintain ground contact.

perform MPC using a simple forward Euler direct collocation scheme as follows:

$$\min \sum_{t=1}^{N-1} \|\mathbf{x}_{lip}^t\|^2_{\mathbf{Q}_{mpc}} + \|u_{lip}^t\|^2_{R_{mpc}} + \|\mathbf{x}_{lip}^N\|_{\mathbf{Q}_f} \tag{33}$$

$$\text{s.t. } \mathbf{x}_{lip}^0, \mathbf{x}_{task}^0 \text{ given} \tag{34}$$

$$\mathbf{x}_{lip}^{t+1} = \mathbf{x}_{lip}^t + (\mathbf{A}_{lip}\mathbf{x}_{lip} + \mathbf{B}_{lip}u_{lip})dt \tag{35}$$

$$\mathbf{x}_{task}^{t+1} = \mathbf{x}_{task}^t + (\mathbf{A}_{task}\mathbf{x}_{task} + \mathbf{B}_{task}\mathbf{u}_{task})dt \tag{36}$$

$$\mathbf{u}_{task}^t = \mathbf{R}u_{lip}^t + \mathbf{Q}\mathbf{x}_{lip}^t + \mathbf{K}(\mathbf{x}_{task}^t - \mathbf{x}_{lip}^t) \tag{37}$$

$$\mathbf{A}_{cwc} \begin{bmatrix} \mathbf{x}_{task}^t \\ \mathbf{u}_{task}^t \end{bmatrix} \leq \mathbf{b}_{cwc} \tag{38}$$

$$\|\dot{\mathbf{l}}_G\|_\infty \leq \dot{l}_{max}, \tag{39}$$

where (34) fixes the initial conditions, (35-36) enforce forward Euler dynamic constraints, (37) enforces the feasibility of the interface, and (38-39) ensures contact constraints are met in the anchor model.

Remark 1. *The constraints (38-39) represent an inner approximation of the CWC, and as such, involve a tradeoff with the parameter $\dot{l}_{max}$. If $\dot{l}_{max}$ is too high, the intersection of constraints defined by (31) will be negligible or empty. On the other hand, if $\dot{l}_{max}$ is too small, $\mathbf{u}_{task}$ may not be able to meet the interface constraint (37). Similarly, while we can find simulation relations with arbitrarily high decay rates λ, the resulting large $\mathbf{K}$ may cause a conflict between the interface constraint (37) and the CWC constraints (38-39).*

These constraints allow us to plan using the (lower-order) LIP model in an MPC fashion. The fact that the constraints are linear means that planning is as simple as solving a QP, for which many fast solvers exist. Then, when the anchor model tracks the nominal LIP trajectory using the interface (25), tracking is guaranteed to be ϵ-close by Theorem 1, as long as additional constraints on torques, joint limits, self-collisions, etc. are met.

V. Simulation Results

As an example, we control the balancer shown in Figure 1. We assume the balancer is constrained to the (x, z) plane, so $\mathbf{x}_{task} \in \mathbb{R}^5$, $\mathbf{u}_{task} \in \mathbb{R}^3$. All links except the "torso" are 1m long, uniform density, and have mass of 1kg. The foot was a square platform with length 1m and mass 5kg. The coefficient of friction was 0.3. The torso link has length 2m and mass 2kg. For the LIP model, we assume a height of $h = 1.75$m.

We simulated the balancer in Gazebo [33], interfacing with Matlab via ROS [34]. We used Casadi [35] and the qpOASES solver [36] to solve (33). All computation was performed on a laptop with an Intel i7 processor and 32GB RAM.

For the interface, we chose $\mathbf{K}$ by solving the following infinite horizon LQR problem:

$$\min_{\mathbf{u}_{task}} \int_0^\infty (\mathbf{x}_{task}^T\mathbf{x}_{task} + 0.01\mathbf{u}_{task}^T\mathbf{u}_{task})dt$$
$$\text{s.t. } \dot{\mathbf{x}}_{task} = \mathbf{A}_{task}\mathbf{x}_{task} + \mathbf{B}_{task}\mathbf{u}_{task}.$$

We then set $\lambda = 0.1$ and solved the SDP described in Section II-B to find $\mathbf{M}$. To linearize the CWC constraint, we chose $\dot{l}_{max} = 5N$.

We compared this approach with a standard QP for tracking the template model as per [10–12]:

$$\min \|\mathbf{J}_{com}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{com}\dot{\mathbf{q}} - \mathbf{u}_{com}\|_2^2 + w\|\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^{des}\|_2^2 \tag{40}$$

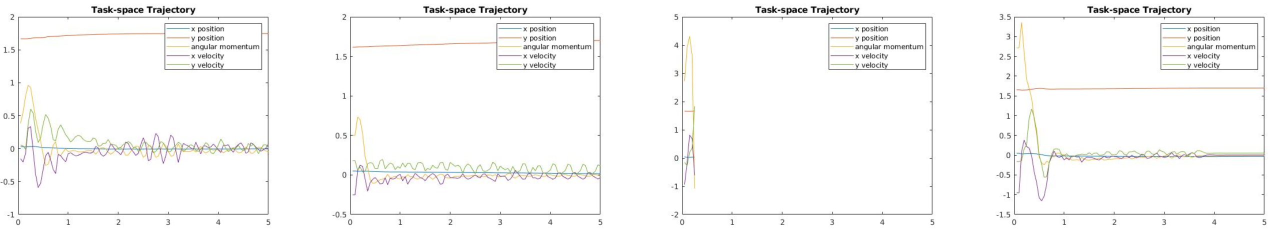
$$\text{s.t. } \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_g = \boldsymbol{\tau} \tag{41}$$

$$\mathbf{f}_0 \in CWC \tag{42}$$

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{max}, \tag{43}$$

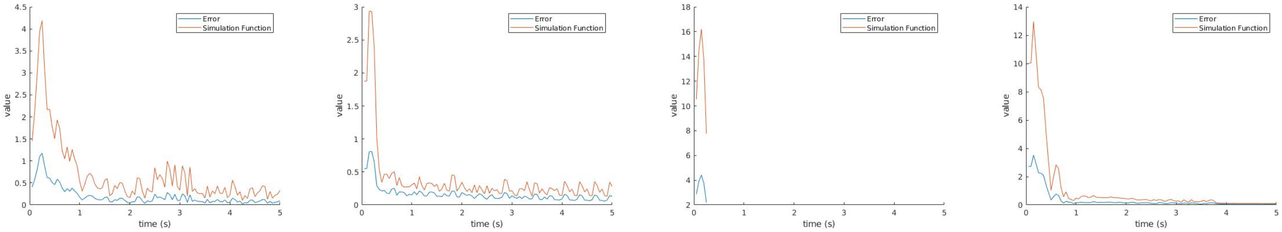
where $\mathbf{J}_{com}$ is the CoM jacobian and $\mathbf{u}_{com}$ is a desired CoM acceleration, which tracks a nominal template trajectory (see Section IV-A). The secondary objective, weighted by $w = 0.1$, is determined by a desired joint-space acceleration $\ddot{\mathbf{q}}^{des}$, which regulates the robot to the static position shown in Figure 1b.

Starting from an initial balanced state, a force was applied for 10ms to the top of the torso link in the $-x$ direction to simulate a push. For our approach, we performed MPC for the template model (33) with $\mathbf{Q}_{mpc} = \text{diag}([10\ 0\ 0\ 10\ 0])$, $R_{mpc} = 5.0$, and $\mathbf{Q}_f = 100\mathbf{Q}_{mpc}$. We applied commands and solved this MPC problem at 20Hz ($dt = 0.05$) with horizon $N = 5$. For the traditional QP approach, we computed a nominal control for the template model via LQR and solved (40) at 20Hz as well.



(a) QP Approach, 20N Push (b) Our Approach, 20N Push (c) QP Approach, 100N Push (d) Our Approach, 100N Push

Fig. 5. task-space trajectories for the push recovery scenario. A high angular momentum disturbance in the 100N scenario causes the traditional QP approach to fail, while our approximate simulation-based controller successfully balances the robot.



(a) QP Approach, 20N Push (b) Our Approach, 20N Push (c) QP Approach, 100N Push (d) Our Approach, 100N Push

Fig. 6. Output error and the simulation function (26) over time for the push-recovery scenario. For all scenarios, the simulation function bounds the output error. Using our approach, the simulation function is nonincreasing (apart from time-discretization error) after the push.

For a $20N$ push, both the QP approach and our approximate simulation approach successfully returned the balancer to an upright position. For a $100N$ push, however, the traditional QP (40) became infeasible after 5 timesteps, causing the robot to fall down. Our approach successfully recovered from the $100N$ push.

Task-space trajectories for these simulations are shown in Figure 5, and the associated errors and simulation functions are shown in Figure 6. For all the scenarios, the simulation function (26) bounds the output error $\|\mathbf{x}_{task} - \mathbf{x}_{lip}\|$. However, this bound is tighter when using our approach, and apart from some noise due to time discretization, decreases over time. Snapshots of our approach recovering from a 100N push are shown in Figure 4.

VI. Discussion

Our primary result is showing a formal connection between template and anchor models, namely that a balancer in single support approximately simulates a LIP. The associated interface, interestingly, is a generalization of the partial feedback linearization-based PD controller often used to track template models. Controlling the anchor model with this interface brings us closer to providing formal guarantees regarding tracking performance. Specifically, we can compute a simulation function which bounds the output error between the two models as long as constraints on contacts, torques, joint limits, etc. are not violated.

As a secondary result, we derived linear constraints that are sufficient conditions for maintaining ground contact. This allows for MPC planning with the template model by solving a QP, for which there are many fast solvers. While our simulation demonstrated solving this QP with a relatively short horizon ($N = 5$), we expect that further optimization of the code and conversion to C/C++ will enable planning with longer horizons.

Using this approach for control of a planar balancer enabled recovery from a large-magnitude push, which a standard QP controller was unable to recover from. While additional tuning of this controller and running it at a higher rate would likely improve performance, the fact that our controller explicitly accounts for angular momentum and contact constraints in the template planning phase suggests that it is more robust to angular momentum disturbances. Furthermore, this explicit accounting for angular momentum results in non-preprogrammed behavior of the "arm" link: the arm swings upward after the push, seemingly in at attempt to regulate angular momentum.

Finally, the particular interface (25) that we propose certifies approximate simulation, but approximate simulation is a relationship between models. This means that there may be other control policies that also provide guaranteed tracking performance. An important open question is whether other control approaches, such as the QP constraint approach commonly used in whole-body control, can also be shown to provide such provably correct tracking of the template model.

VII. Conclusion

We explored approximate simulation as a means of providing formal connections between template and anchor models. We showed that a balancer in single support approximately simulates a linear inverted pendulum and derived the associated interface. We found that this interface is a generalization of the PD controller that is commonly used to track template models. As a secondary result, we derived linear constraints that are sufficient conditions for maintaining ground contact. These allow for rapid planning and replanning in the template

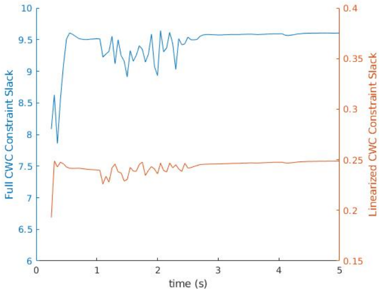


Fig. 7. Slack for the full CWC constraint (28) and the linearized CWC constraint (38) following a 20N push.

model by solving a quadratic program. In a simulated push-recovery scenario for a planar balancer, our approximate simulation-based controller recovered from a large push disturbance that a conventional whole-body controller failed to recover from. Future work will extend these results to account for joint and torque limits, self-collisions, and multi-contact scenarios.

References

[1] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2016, pp. 1203–1234.

[2] H. Geyer and U. Saranli, "Gait based on the spring-loaded inverted pendulum," in *Humanoid Robotics: A Reference*, A. Goswami and P. Vadakkepat, Eds. Springer Netherlands, 2018, pp. 1–25.

[3] 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/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2001, pp. 239–246.

[4] M.-Y. Chen and K. Byl, "Analysis and control techniques for the compass gait with a torso walking on stochastically rough terrain," in *American Control Conference (ACC)*, 2012, pp. 3451–3458.

[5] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3D-SLIP model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5134–5140.

[6] M. Posa, T. Koolen, and R. Tedrake, "Balancing and step recovery capturability via sums-of-squares optimization," in *Robotics: Science and Systems*, 2017.

[7] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2003, pp. 1620–1626.

[8] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 137–142.

[9] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[10] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 981–988.

[11] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[12] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3103–3109.

[13] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[14] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the DARPA robotics challenge," *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.

[15] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 295–302.

[16] 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*, vol. 54, no. 8, pp. 1779–1793, 2009.

[17] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE transactions on automatic control*, vol. 48, no. 1, pp. 42–56, 2003.

[18] A. Girard and G. J. Pappas, "Approximate bisimulation: A bridge between computer science and control theory," *European Journal of Control*, vol. 17, no. 5-6, pp. 568–578, 2011.

[19] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT Press, 2008.

[20] J.-S. Pang and J. Trinkle, "Stability characterizations of rigid body contact problems with coulomb friction," *ZAMM-Journal of Applied Mathematics and Mechanics*, vol. 80, no. 10, pp. 643–663, 2000.

[21] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 579–586.

[22] H. Audren and A. Kheddar, "3-D robust stability polyhedron in multicontact," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 388–403, 2018.

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

[24] A. Girard and G. J. Pappas, "Hierarchical control system design using approximate simulation," *Automatica*, vol. 45, no. 2, pp. 566–571, 2009.

[25] ——, "Approximate bisimulations for nonlinear dynamical systems," in *IEEE Conference on Decision and Control*, 2005, pp. 684–689.

[26] A. Murthy, M. A. Islam, S. A. Smolka, and R. Grosu, "Computing bisimulation functions using sos optimization and δ-decidability over the reals," in *International Conference on Hybrid Systems: Computation and Control*. ACM, 2015, pp. 78–87.

[27] A. Girard and G. J. Pappas, "Approximate bisimulation relations for constrained linear systems," *Automatica*, vol. 43, no. 8, pp. 1307–1317, 2007.

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

[29] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550039, 2016.

[30] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[31] S. Caron, Q.-C. Pham, and Y. Nakamura, "Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5107–5112.

[32] ——, "Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics." in *Robotics: Science and Systems*, 2015.

[33] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2004, pp. 2149–2154.

[34] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source robot operating system," in *IEEE Intl. Conf. on Robotics and Automation Workshop on Open Source Robotics*, 2009.

[35] 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, no. 1, pp. 1–36, 2019.

[36] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.