

3D Dynamic Walking with Underactuated Humanoid Robots: A Direct Collocation Framework for Optimizing Hybrid Zero Dynamics

Ayonga Hereid¹, Eric A. Cousineau², Christian M. Hubicki¹ and Aaron D. Ames³

Abstract—Hybrid zero dynamics (HZD) has emerged as a popular framework for dynamic and underactuated bipedal walking, but has significant implementation difficulties when applied to the high degrees of freedom present in humanoid robots. The primary impediment is the process of gait design—it is difficult for optimizers to converge on a viable set of virtual constraints defining a gait. This paper presents a methodology that allows for the fast and reliable generation of efficient multi-contact robotic walking gaits through the framework of HZD, even in the presence of underactuation. To achieve this goal, we unify methods from trajectory optimization with the control framework of multi-domain hybrid zero dynamics. By formulating a novel optimization problem in the context of direct collocation and generating analytic Jacobians for the constraints, solving the resulting nonlinear program becomes tractable for large-scale nonlinear programming solvers, even for systems as high-dimensional as humanoid robots. We experimentally validated our methodology on the spring-legged prototype humanoid, DURUS, showing that the optimization approach yields dynamic and stable 3D walking gaits.

I. INTRODUCTION

Generating dynamic locomotion for humanoid robots is a challenging exercise, both analytically and computationally. The numerous linkages and degrees of freedom make the search space for gaits dauntingly large. To simplify this planning operation, humanoid control methods often restrict their available options by conforming the robot dynamics to a simplifying model. Notable examples include the linear inverted pendulum model (LIPM) [18], [26], the spring-loaded inverted pendulum (SLIP) [29], center-of-mass dynamics constrained to footstep plans [19], or more-heuristic “spring-mass” policies [22]. However, the formal foundations of these simplifications are possible primarily when the robot is “fully actuated,” i.e., robots with actuators and rigid connections at every joint. In this paper, we present a formal methodology for generating dynamic 3D walking gaits on humanoids that exploits the full-body dynamics of the system, even in the presence of underactuation. This is achieved without restricting the robot to reduced order models through the formulation of a novel large-scale optimization problem that utilizes hybrid zero dynamics.

Hybrid zero dynamics (HZD) [13], [30] is a formal framework for the design of nonlinear controllers that yield

This research is supported by DARPA grant D15AP00006 and NSF grants CPS-1239055 and NRI-1526519.

¹Ayonga Hereid and Christian M. Hubicki are with the Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, {ayonga27, chubicki6}@gatech.edu

²Eric A. Cousineau is with Mathworks Inc, eacousineau@tamu.edu

³Aaron D. Ames is with the Woodruff School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, ames@gatech.edu

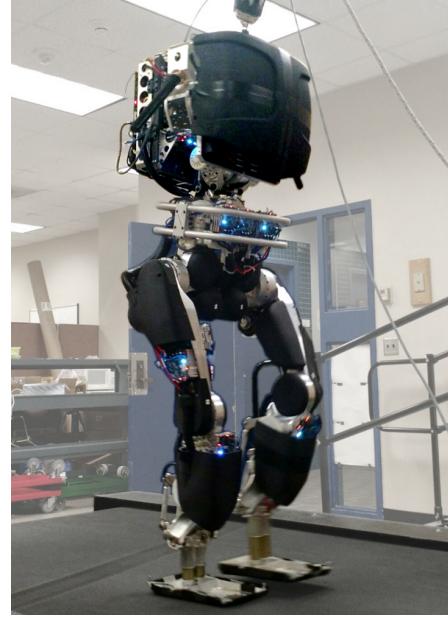


Fig. 1: DURUS is a 180-cm-tall spring-legged humanoid robot designed for efficient locomotion. The above experiment shows DURUS’ ability to walk dynamically on a treadmill with all three dimensions unrestricted as a consequence of the control framework presented in this paper (a video can be found at [1]).

dynamic bipedal locomotion even in the presence of underactuation and multi-contact foot behaviors, and has had numerous successful robot implementations [3], [7], [8], [25], [31]. HZD works by designing a set of *virtual constraints*, which are enforced via feedback control of the actuated degrees of freedom. If these constraints satisfy a hybrid invariance condition, i.e., invariance through impact, all of the stability properties of the high-dimensional system are effectively captured in a lower-dimensional representation. However, reliably generating bipedal gaits that satisfy these conditions becomes increasingly difficult as a robots’ degrees of freedom and underactuation grow.

The main contribution of this paper is to present a generalized gait generating framework for dynamic bipedal locomotion that scales effectively to 3D humanoid locomotion. Our approach unifies virtual constraint optimization with direct collocation formulations for multi-domain hybrid systems. While virtual constraints provide the formal guarantees of stability of the gait through hybrid zero dynamics, direct collocation provides scalability for optimizing large-scale dynamical systems through discretization and approximation

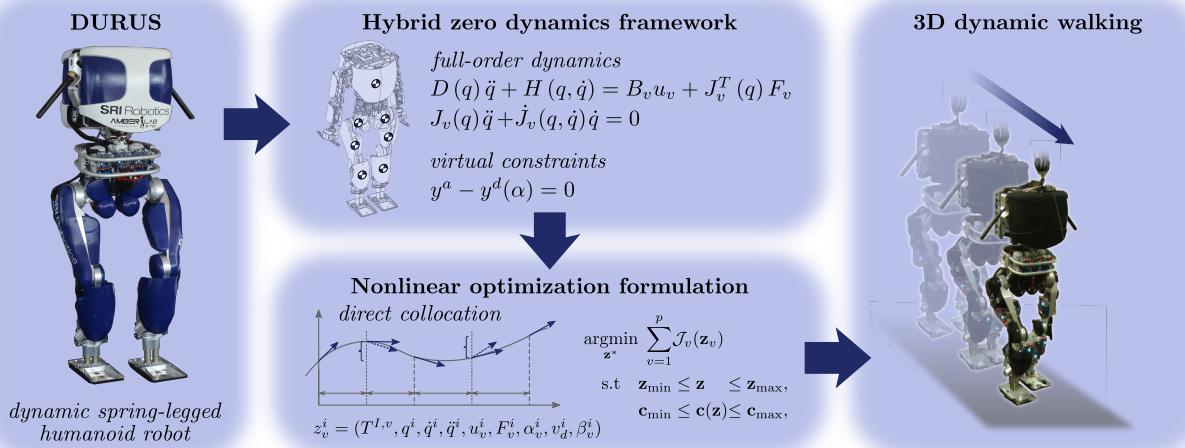


Fig. 2: Illustration of the process used to generate dynamic 3D walking with DURUS, an underactuated spring-legged humanoid robot. This direct collocation framework parses a multibody model of the robot and set of parameterized virtual constraints into a large and sparse nonlinear program (NLP) with upwards of 10,000 design variables and constraints. Large-scale algorithms can typically solve this NLP in under 10 minutes, thereby optimizing a dynamic gait for the 3D humanoid that exploits the full multi-body dynamics of the machine, even in the presence of underactuation.

of states and controls. Moreover, defect variables are introduced in the problem formulation to simplify the expression of constraints, enabling symbolic generation of an analytic Jacobian matrix of constraints. By carefully ordering optimization variables and constraints, a banded structure of the Jacobian matrix is realized, resulting in a fast and effective optimization tool for optimal HZD gait generation.

To demonstrate the effectiveness and reliability of this method, we then apply the formulation of multi-domain hybrid systems and virtual constraints based direct collocation optimization to a twenty-three degree-of-freedom humanoid robot, DURUS (see Fig. 1), to generate stable dynamic walking gaits in 3D. Finally, using the parameters of virtual constraints obtained from the optimization, sustained dynamic walking is realized experimentally on the humanoid robot DURUS.

II. HYBRID ZERO DYNAMICS AND BIPEDAL LOCOMOTION

Bipedal locomotion consists of phases of continuous dynamics (e.g., when the leg swings forward) and discrete dynamics (e.g., when the foot strikes the ground); formally modeling this interplay of continuous and discrete dynamics results in a hybrid system model of robotic walking [2], [13]. During a steady-state walking gait, the transitions between the continuous phases become ordered and periodic; this motivates the use of a multi-domain hybrid system with a predetermined ordering of phases (or domains) as represented by a *directed cycle*, i.e., a cyclic directed graph [24]. Formally, the definition of a *multi-domain hybrid control system* is given as a tuple (see [4] for a full definition),

$$\mathcal{H} = (\Gamma, \mathcal{D}, \mathcal{U}, S, \Delta, FG), \quad (1)$$

where $\Gamma = (V, E)$ is a directed cycle. In this section, we briefly introduce remaining elements of the hybrid system

model. For simplicity of notation, we specify $v \in V$ be an arbitrary vertex, v^+ be the subsequent vertex of v in the cycle, and $e = \{v \rightarrow v^+\}$ be the transition from v to v^+ for the remainder of the paper.

Holonomic Constraints. Given a robot model with coordinates $q \in \mathcal{Q}$, where $\mathcal{Q} \subset \mathbb{R}^n$ is the configuration space of the robot with n degrees of freedom, the dynamics of the system in a domain depends on both the Lagrangian of the model and the contact constraints. Any physical contact of the robot with the external environment introduces a *holonomic constraint*, $\eta_c(q)$. Let \mathcal{C}_v be a indexing set of all holonomic constraints defined on \mathcal{D}_v , we state the holonomic constraints of the domain as $\eta_v = \{\eta_c\}_{c \in \mathcal{C}_v} \equiv \text{constant}$ and the associated kinematic constraints as $J_v(q)\dot{q} = 0$, where $J_v(q)$ is the Jacobian matrix of η_v , i.e., $J_v(q) = \frac{\partial \eta_v}{\partial q}$.

Continuous Dynamics. With the mass, inertia and length properties of each link of a robot model, the equation of motion (EOM) for a given domain \mathcal{D}_v is determined by the classical Euler-Lagrange equation [13], [21]:

$$D(q)\ddot{q} + H(q, \dot{q}) = B_v u_v + J_v^T(q) F_v, \quad (2)$$

where $F_v : T\mathcal{Q} \times U_v \rightarrow \mathbb{R}^{n_v}$, with n_v the number of total holonomic constraints, is a vector of contact *wrenches* containing the constraint forces and/or moments (see [21]). To enforce the holonomic constraints, the second order differentiation of the constraints, η_v should be set to zero,

$$J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q} = 0. \quad (3)$$

The constrained dynamics of the system is determined by evaluating both (2) and (3) simultaneously.

Domains and Guards. It was shown in [13], [16] that forces and moments produced by holonomic constraints are limited in number. Specifically, we state these conditions in the form

of inequalities:

$$\nu_v(q)F_v(q, \dot{q}, u_v) \geq 0, \quad (4)$$

where $\nu_v(q)$ depends on the physical parameters of the system. Another class of constraints that determines the admissible configuration of the system are termed *unilateral constraints*, denoted by: $h_v(q) \geq 0$. Combining (4) and unilateral constraints (if present) together yields the domain of admissibility:

$$\mathcal{D}_v = \{(q, \dot{q}, u_v) \in T\mathcal{Q} \times U_v | A_v(q, \dot{q}, u_v) \geq 0\}, \quad (5)$$

for $v \in V$, where

$$A_v(q, \dot{q}, u_v) = \begin{bmatrix} \nu_v(q)F_v(q, \dot{q}, u_v) \\ h_v(q) \end{bmatrix} \geq 0, \quad (6)$$

defines the boundary of the domain manifold.

A guard S_e is a proper subset of the boundary of the domain, \mathcal{D}_v , determined by an edge condition associated with the transition from \mathcal{D}_v to the subsequent domain, \mathcal{D}_{v+} . Let $H_e(q, \dot{q}, u_v)$ be the appropriate elements from the vector in (6) corresponding to the edge condition, then the guard is defined as

$$S_e = \{(q, \dot{q}, u_v) \in T\mathcal{Q} \times U_v | H_e(q, \dot{q}, u_v) = 0, H_e(q, \dot{q}, u_v) < 0\}. \quad (7)$$

Discrete Dynamics. Associated with the guard S_e is a reset map Δ_e that maps the system states at the guard to the subsequent domain. Given pre-impact states (q^-, \dot{q}^-) on S_e , the post-impact states (q^+, \dot{q}^+) of \mathcal{D}_{v+} are computed using a reset map Δ_e by assuming a perfectly plastic impact (if an impact occurs) [12], [17]. Following the presentation in [13], configurations of the system are invariant through an impact, i.e., $q^+ = q^-$, but post-impact velocities need to satisfy the plastic impact equation:

$$\begin{bmatrix} D(q^-) & -J_{v+}^T(q^-) \\ J_{v+}(q^-) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ \delta F_v \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ 0 \end{bmatrix}, \quad (8)$$

where δF_v is a vector of impulsive contact wrenches.

Virtual Constraints. Analogous to holonomic constraints, virtual constraints (also termed *outputs* in the control literature [2]) are defined as a set of functions that modulate the behavior of a robot in order to achieve certain desired trajectories [30]. The term “virtual” comes from the fact that these constraints are enforced through feedback control instead of through physical constraints.

Virtual constraints are defined as the difference between the actual and desired outputs of the robot:

$$y_{1,v} = \dot{y}_{1,v}^a(q, \dot{q}) - y_{1,v}^d(\alpha_v), \quad (9)$$

$$y_{2,v} = y_{2,v}^a(q) - y_{2,v}^d(q, \alpha_v), \quad (10)$$

for $v \in V$, where $y_{1,v}$ and $y_{2,v}$ are relative degree 1 and (vector) relative degree 2 by definition (see [23] for the definition of relative degree), respectively. In this paper, we use the definition of outputs described in [2] (see **Definition I**). Specifically, we assume the desired velocity-modulating output to be a constant, i.e., $y_{1,v}^d(\alpha_v) = v_d \in \mathbb{R}$ and the

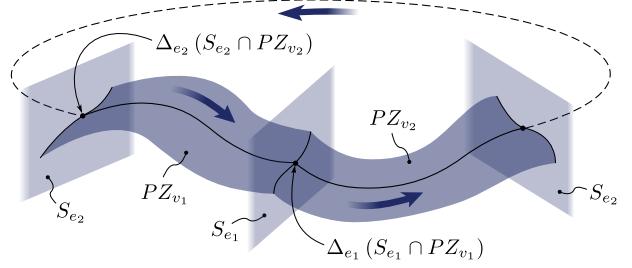


Fig. 3: Illustration of the PHZD periodic orbit in the case of a two-domain hybrid system.

desired position-modulating outputs are given in term of a Bézier polynomial of degree M , determined by $M + 1$ coefficients [31]:

$$y_2^d(\tau, \alpha_o) := \sum_{k=0}^M \alpha_o[k] \frac{M!}{k!(M-k)!} \tau^k (1-\tau)^{M-k}, \quad (11)$$

for all $o \in \mathcal{O}_v$ with \mathcal{O}_v be an indexing set, where α_o is a vector of Bézier polynomial coefficients, and τ is the state-based parameterization of time. The introduction of τ , which has to be monotonic over a gait cycle, is motivated by the desire to create an autonomous controller, which is more robust than non-autonomous controllers [30].

Partial Hybrid Zero Dynamics. With the goal of driving the virtual constraints $y_v = (y_{1,v}, y_{2,v}) \rightarrow 0$ exponentially, consider the feedback linearization control law described in Eq. 28 of [2]. Applying this control law yields linear output dynamics of the form:

$$\dot{y}_{1,v} = -\varepsilon y_{1,v}, \quad (12)$$

$$\dot{y}_{2,v} = -2\varepsilon \dot{y}_{2,v} - \varepsilon^2 y_{2,v}. \quad (13)$$

with $\varepsilon > 0$ and renders the *zero dynamics* submanifold invariant in each continuous domain. However, it is not necessarily invariant through discrete dynamics. In fact, enforcing impact invariance of the relative degree 1 output is too strong of a condition due to the velocity change at impact. Hence, we enforce conditions only related to the relative degree 2 virtual constraints, $y_{2,v}$, resulting in the *partial zero dynamics* surface (see [2]), given by:

$$\mathcal{PZ}_v = \{(q, \dot{q}) \in \mathcal{D}_v | y_{2,v} = 0, \dot{y}_{2,v} = 0\}. \quad (14)$$

Moreover, for any $e \in E$, the submanifold \mathcal{PZ}_v is called impact invariant, if there exist a set of parameters v_d and $\{\alpha_v\}_{v \in V}$, with $\alpha_v = (\alpha_o)_{o \in \mathcal{O}_v}$, so that

$$\Delta_e(x) \in \mathcal{PZ}_{v+}, \quad \forall x \in S_e \cap \mathcal{PZ}_v. \quad (15)$$

A manifold $\mathcal{PZ} = \bigcup_{v \in V} \mathcal{PZ}_v$ is called *hybrid invariant* if it is invariant over all domains of continuous dynamics and impact invariant through all discrete dynamics, i.e., solutions that start in \mathcal{PZ} remain in \mathcal{PZ} , even after impulse effects, see Fig. 3. If a feedback control law renders \mathcal{PZ} hybrid invariant, then we say that the multi-domain hybrid control system has a *partial hybrid zero dynamics* (PHZD), $\mathcal{H}|_{\mathcal{PZ}}$.

III. OPTIMIZATION VIA DIRECT COLLOCATION

The core contribution of this paper is a computational framework for generating walking gaits that meet the above conditions, and its subsequent implementation on the DURUS humanoid. Specifically, this requires determining a valid set of gait parameters, α_v and v_d , that both satisfies PHZD requirements and minimizes a cost—the result is a nonlinear program (NLP).

A straightforward and traditional approach to transcribing such a gait design optimization is via *direct single shooting* methods. This involves assigning each gait parameter a design variable, integrating the dynamics via standard time-marching numerical methods, evaluating the PHZD conditions as equality constraints, and employing an NLP solver to drive these conditions to zero. Variations of this approach have been successfully applied on fully actuated robots [3] and planar point-feet robots [30]. However, as robots are built with more actuated linkages or have higher degrees of underactuation, such NLP's become increasingly intractable for single-shooting methods. In the author's previous work, techniques from the trajectory optimization community were first utilized; specifically, direct multiple shooting methods [6] based on reduced-dimensional hybrid zero dynamics [15] were used to improve the reliability and speed of solving this PHZD NLP. But such multiple-shooting approaches also run into scalability issues with increasing degrees of freedom, due to the increased complexity of the equations describing the robot dynamics.

In this section, we present the main contribution of the paper—an optimization problem for multi-domain bipedal walking using the *direct collocation* method [27]. Motivated by the desire to provide a generalized gait optimization tool for HZD type bipedal walking, the discussion in this section is based on the general multi-domain hybrid system model presented in Sec. II.

A. General Description

In the case of local direct collocation, implicit Runge-Kutta methods—such as Hermite-Simpson and Trapezoidal method—are most commonly used. In this paper, we specifically employ the Hermite-Simpson (Separated) scheme so as to achieve better accuracy with fewer nodes (see [5]). Assuming $T_{I,v} > 0$ is the time at which the system reaches the guard associated with a given domain, \mathcal{D}_v with $v \in V$, the discretization of time is given as, $0 = t_0 < t_1 < t_2 < \dots < t_{N_v} = T_{I,v}$, with $N_v = 2(N_v^c - 1)$, where the even points are called *cardinal nodes*, the odd points are called *interior nodes* (see Fig. 4), and $N_v^c \in \mathbb{Z}$ represents the total number of cardinal nodes chosen per domain. There have to be two or more cardinal nodes defined per domain. The distribution of cardinal nodes within a domain could be arbitrary, however, a interior point has to be placed at the center of two adjacent cardinal nodes. Our formulation allows the cardinal nodes to be placed at the uniformly distributed points or the Chebyshev-Gauss-Lobatto (CGL) points, where the former provides simplicity in implementation, and the

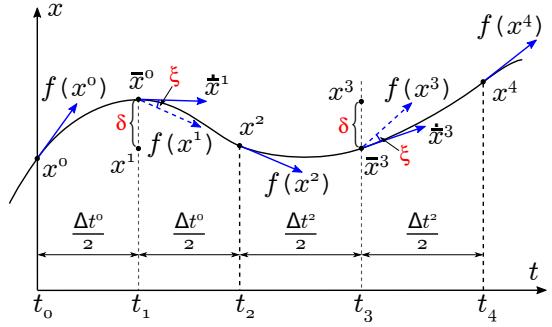


Fig. 4: Illustration of defect constraints and node distribution.

latter yields better accuracy due to the fact that more nodes are placed close to the two ends.

Given the discretization, the separated Hermite-Simpson scheme uses Hermite interpolation polynomials to represent the state trajectories within two neighboring cardinal nodes using the estimated states, $x^i = (q^i, \dot{q}^i)$, and their derivatives, \dot{x}^i , obtained by evaluating the system dynamic equation, where i is the numbering of the nodes. As illustrated in Fig. 4, there are two defect constraints defined at each interior point: 1) the difference between the estimated states from the optimizer and the interpolated states of that point using the approximated polynomial, and 2) the difference between the derivatives of states obtained through the system dynamic equation and the time derivatives of the approximated polynomial at that point [5], [14]. These constraints can be stated as,

$$x^i - \frac{1}{2}(x^{i+1} + x^{i-1}) - \frac{1}{8}\Delta t_v^i(\dot{x}^{i-1} - \dot{x}^{i+1}) = 0, \quad (C1)$$

$$x^{i+1} - x^{i-1} - \frac{1}{6}\Delta t_v^i(\dot{x}^{i-1} + 4\dot{x}^i + \dot{x}^{i+1}) = 0, \quad (C2)$$

with $\Delta t_v^i = t_{i+1} - t_{i-1}$, for $i \in \{1, 3, 5, \dots, N_v - 1\}$.

B. Introducing Defect Variables

Defect variables are sets of optimization variables that could have been determined by closed form functions, e.g., \dot{x}^i in (C1) and (C2). The idea of introducing defect variables in the NLP is that instead of computing these variables explicitly using relatively complicated functions, we impose implicit but equivalent equality constraints, which are often computationally easier in the optimization. For example, computing \dot{x}^i explicitly requires inverting the inertia matrix, however, the original formulation of system dynamics in (2) and (3) does not.

Introducing defect variables replaces these closed form equations with a set of equivalent, but less complicated equality constraints. With the discretization of a continuous domain, let

$$z_v^i = (T_{I,v}^i, q^i, \dot{q}^i, \ddot{q}^i, u_v^i, F_v^i, \alpha_v^i, v_d^i) \quad (16)$$

be a vector of optimization variables defined at each node $i \in \{0, 1, 2, \dots, N_v\}$. By introducing defect variables in this fashion, the first order derivatives in (C1) and (C2) are no longer obtained by evaluating the system dynamics

explicitly. Instead the evaluation at each node is replaced by the dynamics equation in (2) and the holonomic constraints equation in (3):

$$D(q^i)\ddot{q}^i + H(q^i, \dot{q}^i) - B_v u_v^i - J_v^T(q^i)F_v^i = 0, \quad (\text{C3})$$

$$J_v(q^i)\ddot{q}^i + \dot{J}_v(q^i, \dot{q}^i)\dot{q}^i = 0, \quad (\text{C4})$$

for $i \in \{0, 1, 2, \dots, N_v\}$ and $\forall v \in V$.

To determine the feedback controller through virtual constraints, the following constraints are imposed to ensure that the resulting solution of the system evolves as the linear output dynamics as in (12)-(13):

$$\dot{y}_{1,v}(q^i, \dot{q}^i, \ddot{q}^i, v_d^i) + \varepsilon y_{1,v}(q^i, \dot{q}^i, v_d^i) = 0, \quad (\text{C5})$$

$$\begin{aligned} \ddot{y}_{2,v}(q^i, \dot{q}^i, \ddot{q}^i, \alpha_v^i) + 2\varepsilon \dot{y}_{2,v}(q^i, \dot{q}^i, \alpha_v^i) \\ + \varepsilon^2 y_{2,v}(q^i, \alpha_v^i) = 0, \end{aligned} \quad (\text{C6})$$

for $i \in \{0, 1, 2, \dots, N_v\}$ and $\forall v \in V$. Notice that we define the time $T_{I,v}$, and the parameters α_v^i and v_d^i at each node albeit being constant throughout the domain. Defining variables in this fashion, along with arranging constraints and optimization variables of each node as a unit, realizes a computationally efficient band structure in the Jacobian matrix. Of course, this approach requires additional constraints to ensure the consistency of parameters:

$$\alpha_v^i - \alpha_v^{i+1} = 0, \quad (\text{C7})$$

$$v_d^i - v_d^{i+1} = 0, \quad (\text{C8})$$

$$T_{I,v}^i - T_{I,v}^{i+1} = 0, \quad (\text{C9})$$

for $i \in \{0, 1, 2, \dots, N_v - 1\}$ and $\forall v \in V$.

Furthermore, to satisfy the *partial hybrid zero dynamics* (PHZD) constraints, relative degree 2 outputs $y_{2,v}$ and their first-order derivatives $\dot{y}_{2,v}$ should be zero at the beginning of each continuous domain. In other words, the following equality constraints should be imposed at the first node of each domain $v \in V$:

$$y_{2,v}(q^0, \alpha_v^0) = 0, \quad (\text{C10})$$

$$\dot{y}_{2,v}(q^0, \dot{q}^0, \alpha_v^0) = 0. \quad (\text{C11})$$

Thus, any parameters set $\{\alpha_v\}_{v \in V}$ and v_d that satisfy (C1)-(C11), as well as other admissible constraints described later, guarantee the system has *partial hybrid zero dynamics*.

C. Admissible Constraints

To complete the optimization problem for multi-domain bipedal walking under the hybrid system model framework, we also have to consider other admissible constraints, including path and terminal constraints.

Domain of Admissibility. According to (5), the domain of admissibility constraints are enforced at each node (including interior nodes):

$$\nu_v(q^i)F_v^i \geq 0, \quad (\text{C12})$$

$$h_v(q^i) \geq 0, \quad (\text{C13})$$

for $i \in \{0, 1, 2, \dots, N_v\}$ and $\forall v \in V$. It can be noted that the dependence of contact wrenches on q , \dot{q} , and u_v

is no longer required here, as F_v^i is explicitly defined as optimization variables.

Guard Condition. We impose the guard condition in (7) at the last node of each domain, i.e.,

$$H_e(q^N, \dot{q}^N, F_v^N) = 0, \quad (\text{C14})$$

$$\dot{H}_e(q^N, \dot{q}^N) < 0, \quad (\text{C15})$$

$\forall v \in V$ with e is the subsequent edge of a vertex v . We also replace u_v with F_v for the reason that the guard condition is often defined as either a function of robot configuration or contact wrenches.

Reset Map. Let \mathcal{R} be the relabeling matrix if there is a coordinate change and an identity matrix if there is no coordinates change. We impose the state continuity constraints between two neighboring domains as,

$$\mathcal{R}q^{0_{v+}} - q^{N_v} = 0, \quad (\text{C16})$$

$$J_{v+}(q^{N_v})\mathcal{R}\dot{q}^{0_{v+}} = 0, \quad (\text{C17})$$

$$D(q^{N_v})(\mathcal{R}\dot{q}^{0_{v+}} - \dot{q}^{N_v}) - J_{v+}^T(q^{N_v})\delta F_v^{N_v} = 0, \quad (\text{C18})$$

$\forall v \in V$. Here, we use the notation \square^{0_v} and \square^{N_v} to represent variables defined at the first and the last node of the domain \mathcal{D}_v , respectively. Following the idea of introducing defect variables, we include the impact wrenches δF_v in the NLP variables vector to express constraints in the simplest form possible.

Holonomic Constraints. We add the desired holonomic constraints $\bar{\eta}_v = \{\bar{\eta}_c\}_{c \in \mathcal{C}_v}$ as a set of augmented parameters in the optimization variables. Then, the following constraint is imposed at the first node of each domain $v \in V$:

$$\eta_v(q^{0_v}) - \bar{\eta}_v^{0_v} = 0. \quad (\text{C19})$$

Parameter Consistency. The desired velocity v_d should be the same for each domain. We state the parameter consistency constraints in the following form:

$$v_d^{N_v} - v_d^{0_{v+}} = 0, \quad (\text{C20})$$

for each domain \mathcal{D}_v except the last domain in the graph. If the virtual constraint is present in neighboring domains, then α_v continuity is enforced via,

$$\left\{ \alpha_o^{N_v} - \alpha_o^{0_{v+}} \right\}_{o \in (\mathcal{O}_v \cap \mathcal{O}_{v+})} = 0, \quad (\text{C21})$$

and if holonomic constraints are present in neighboring domains,

$$\left\{ \bar{\eta}_c^{0_v} - \bar{\eta}_c^{0_{v+}} \right\}_{c \in (\mathcal{C}_v \cap \mathcal{C}_{v+})} = 0, \quad (\text{C22})$$

for each domain \mathcal{D}_v except the last domain in the graph.

D. Problem Formulation of Direct Collocation Optimization

We now formally define the nonlinear program where we let $\mathbf{z} = \{\mathbf{z}_v\}_{v \in V}$ be a vector of all optimization variables, where $\mathbf{z}_v = (\bar{\eta}_v^0, z_v^0, z_v^1, \dots, z_v^{N_v}, \delta F_v^{N_v})$ is a vector of

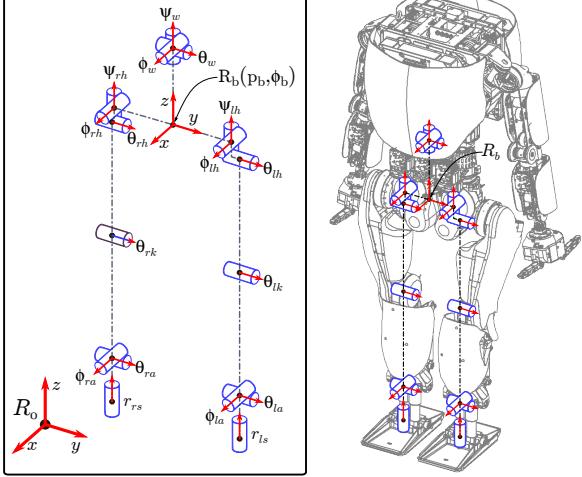


Fig. 5: The coordinates of DURUS robot, where R_0 is the inertial frame, R_b is the robot base frame located at the center of hip with p_b , ϕ_b is the position and orientation of R_b . ψ_w , ϕ_w , and θ_w are the waist yaw, roll, and pitch angles, ψ_{lh} , ϕ_{lh} , θ_{lh} , θ_{lk} , θ_{la} , ϕ_{la} , and r_{ls} are the left hip yaw, hip roll, hip pitch, knee pitch, ankle pitch, ankle roll angles, and spring deflection, respectively, and ψ_{rh} , ϕ_{rh} , θ_{rh} , θ_{rk} , θ_{ra} , ϕ_{ra} , and r_{rs} are the right hip yaw, hip roll, hip pitch, knee pitch, ankle pitch, ankle roll angles, and spring deflection, respectively.

optimization variables defined on domain \mathcal{D}_v with z_v^i given in (16). Then we state the optimization problem as,

$$\underset{\mathbf{z}^*}{\operatorname{argmin}} \Phi(\mathbf{z}) \quad (17)$$

$$\text{s.t. } \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}, \quad (18)$$

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{z}) \leq \mathbf{c}_{\max}, \quad (19)$$

where $\Phi(\mathbf{z})$ is the cost function, and $\mathbf{c}(\mathbf{z})$ is a vector of functions defined in (C1)-(C22) organized in the order of nodes, \mathbf{z}_{\min} , \mathbf{c}_{\min} and \mathbf{z}_{\max} , \mathbf{c}_{\max} are the vectors containing the minimum and maximum values of optimization variables and constraints, respectively.

The formulation of direct collocation method significantly increases the number of constraints and optimization variables, leading to a *large sparse* nonlinear optimization problem. Yet, the Jacobian matrix is very sparse—the density of the matrix is less than 1% in many cases. This feature allows the problem to be solved efficiently using appropriate large sparse NLP solvers such as IPOPT [28], SNOPT [11], etc. Additionally, in this formulation, the Jacobian matrix of the constraints and objective with respect to the design variables can be obtained symbolically, and evaluated in closed form (as opposed to computationally-taxing finite differencing methods). This analytical convenience allows for reasonable computation times for such 10,000+ design variable problems.

IV. APPLICATION TO THE DURUS HUMANOID

DURUS is a three-dimensional humanoid robot designed and built by SRI International to implement efficient and dynamic locomotion. DURUS consists of fifteen actuated

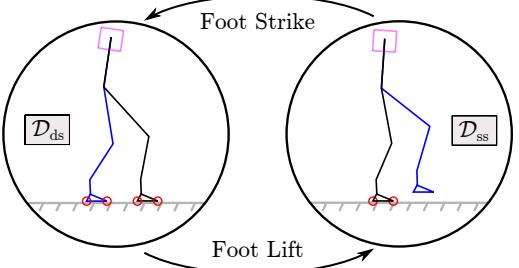


Fig. 6: Domains graph of 3D walking, where red circles represent foot contact points.

joints throughout the body driven by either ultra-efficient cycloid drives or harmonic drives. The design of the passive spring at the end of each leg, which is often vertical to the foot plate, helps reduce energy loss by absorbing the kinetic energy during the foot impact. To model the robot, we assume that the base frame, R_b , of the robot is located at the center of the hip. As illustrated in Fig. 5, the kinematic tree of body coordinates consists of three branches: waist joints, $q_w = [\psi_w, \phi_w, \theta_w]^T$, left leg joints, $q_l = [\psi_{lh}, \phi_{lh}, \theta_{lh}, \theta_{lk}, \theta_{la}, \phi_{la}, r_{ls}]^T$, and right leg joints, $q_r = [\psi_{rh}, \phi_{rh}, \theta_{rh}, \theta_{rk}, \theta_{ra}, \phi_{ra}, r_{rs}]^T$, respectively. Specifically, we model the passive spring on both legs as a prismatic joint initially, and then apply a “feedback controller” on this joint by ensuring that the “feedback controller” actually equals to the spring force associated with the deflection of this joint. In addition, we incorporate the reflected inertia of actuators as a decoupled addition to the inertia matrix as in [10].

Due to the existence of the passive springs, the system is no longer rigid, thus the impact of the non-stance foot is not guaranteed (and is unlikely) to cause the trailing foot to break contact. Therefore, the hybrid system model of the flat-foot walking of 3D humanoid robot consisting of two domains: a *double-support*, \mathcal{D}_{ds} , and a *single-support* domain, \mathcal{D}_{ss} , (see Fig. 6). Accordingly, we define that a transition from double-support to single-support takes place when the normal force on non-stance foot reaches zero, and a transition from single-support to double support domain occurs when the non-stance foot strikes the ground. The continuous and discrete dynamics of the system are (2), (3), and (8). To ensure the foot on the ground remains flat, the associated contact wrenches should satisfy: the positive normal force, non-slipping condition, and ZMP constraints [13]. Incorporating these constraints in the form given in (4), the domain of admissibility conditions for both domains are determined along with additional unilateral constraints.

Virtual Constraints. With the hybrid system model of 3D flat-foot walking in hand, now we design virtual constraints for each domain based on the formulation in Sec. II. Inspired by [3], we pick the linearized hip position, $\delta p_{\text{hip}}(q) = L_a \theta_{ra} + (L_a + L_c) \theta_{rk} + (L_a + L_c + L_t) \theta_{rh}$, as the velocity-modulating output $y_{1,v}^a(q)$ for both domain $v \in \{\text{ds}, \text{ss}\}$, where L_a , L_c , and L_t are the length of ankle, calf, and thigh link of the robot, respectively.

We pick the following position-modulating outputs for the

double-support domain:

- stance knee pitch: $y_{2,\text{skp}}^a = \theta_{rk},$
- stance torso pitch: $y_{2,\text{stp}}^a = -\theta_{ra} - \theta_{rk} - \theta_{rh},$
- stance ankle roll: $y_{2,\text{sar}}^a = \phi_{ra},$
- stance torso roll: $y_{2,\text{str}}^a = -\phi_{ra} - \phi_{rh},$
- stance hip yaw: $y_{2,\text{shy}}^a = \psi_{rh},$
- waist roll: $y_{2,\text{wr}}^a = \phi_w,$
- waist pitch: $y_{2,\text{wp}}^a = \theta_w,$
- waist yaw: $y_{2,\text{wy}}^a = \psi_w,$
- non-stance knee pitch: $y_{2,\text{nskp}}^a = \theta_{lk},$

and define five more outputs for the single-support domain:

- non-stance slope:

$$y_{2,\text{nsl}}^a = -\theta_{ra} - \theta_{rk} - \theta_{rh} + \frac{L_c}{L_c + L_t} \theta_{lk} + \theta_{lh},$$

- non-stance leg roll: $y_{2,n}^a = \phi_{ra} - \phi_{la},$
- non-stance foot roll: $y_{2,\text{nsfr}}^a = p_{nsf^1}^z(q) - p_{nsf^0}^z(q),$
- non-stance foot pitch: $y_{2,\text{nsfp}}^a = p_{nst}^z(q) - p_{nsh}^z(q),$
- non-stance foot yaw: $y_{2,\text{nsfy}}^a = p_{nst}^y(q) - p_{nsh}^y(q),$

with $(p_{\square}^x(q), p_{\square}^y(q), p_{\square}^z(q))$ the Cartesian positions of a point indicated by the subscript.

Gait Generation. To achieve efficient walking, we set the objective function to minimize the mechanical cost of transport of the walking gait:

$$\Phi(\mathbf{z}) := \sum_{v \in V} \frac{1}{mgd(\bar{\eta}_v)} \left(\sum_{i=1}^{N_v-1} \left(\frac{\|P_v(u_v^i, \dot{q}^i)\| \cdot \Delta t_i}{T_{I,v}^i} \right) \right), \quad (20)$$

where mg is the robot weight, $d(\bar{\eta}_v)$ is the distance traveled during a gait which could be determined from the desired holonomic constraints, and $P_v(u_v^i, \dot{q}^i)$ is the total power consumed (assuming no power-regeneration) computed at each interior node. Constraints can be formulated as in Sec. III with proper upper and lower boundaries imposed on both optimization variables and constraints according to physical limitation of the robot hardware. Additionally, constraints were added to keep the non-stance foot flat with respect to the ground, to ensure flat foot contact. We assigned 10 and 20 cardinal nodes for the double-support and single-support domains, respectively.

The stability of a walking gait is evaluated *a posteriori* by computing the magnitude of the eigenvalues of the Jacobian of the Poincaré return map [20], which must all be less than one to certify exponential stability of the periodic orbit. Numerically evaluating the Jacobian of the return map in this simulation revealed a maximum eigenvalue magnitude of 0.24, suggesting exponential stability.

Typical Optimization Performance. The formulated optimization problems were solved using IPOPT with linear solver ma57 on a laptop computer with an Intel Core i7-3820QM processor (2.7 GHz) and 12 GB of RAM. The NLP's were solved to a constraint tolerance less than 10^{-12} in 910 seconds using a random initial guess. Seeding the optimizer with a previous gait as an initial guess reduced solving time to 380 seconds.

Experimental Setup. During the experiments, the robot walks on a $5' \times 8'$ large treadmill platform. A slack safety

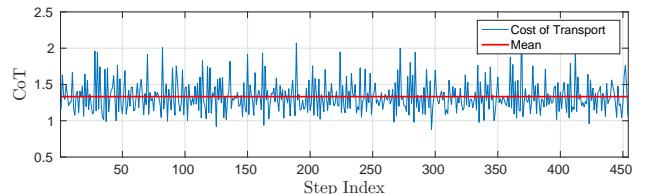


Fig. 7: Cost of transport (CoT) for a walking experiment over 450 steps with a mean cost of transport of 1.33.

tether is connected to the robot to catch DURUS in case of a fall. Otherwise, the robot is capable of fully untethered locomotion, operating with an on-board processor and battery pack installed in the torso.

3D Walking Experiment. To execute 3D walking, DURUS was programmed to follow the optimized outputs with respect to time using PD position control for all actuators. Fig. 8a shows the phase plots for actuated joints compared to the optimized gait when simulated using a MATLAB variable time step integrator (`ode45`). The resulting control yielded stable 3D walking on DURUS, as depicted in walking tiles in Fig. 8b. DURUS walked unassisted for over 30 minutes on many occasions at an average speed of $0.23m/s$, based on treadmill data (slightly slower than the $0.27m/s$ predicted by the simulator). Measurements of the electrical cost of transport (CoT)¹ of one trial of these experiments is shown in see Fig. 7, where the CoT averages to 1.33 over 450 steps.

V. CONCLUSION

The authors presented a generalized optimization framework for generating formally stable 3D locomotion on underactuated humanoid robots. It does so by solving a nonlinear program that converges in under ten minutes on a laptop computer. By building upon the theoretical foundation of HZD, this method optimizes the interactions of the full multi-domain multibody dynamics of humanoid system models, without conforming motions to simpler more-tractable dynamics. Further, this method produced 3D dynamic walking on the spring-legged humanoid, DURUS. By optimizing for efficient locomotion, we achieved an average cost of transport of 1.33, significantly lower than that reported by other human-scale humanoid robots [9]. The authors believe that this result encourages robot designs that don't shy away from underactuation to simplify control, and signals that HZD approaches can now tackle the complexity of humanoid locomotion.

ACKNOWLEDGMENT

The authors would like to thank the researchers and technicians at SRI International, other members of the AMBER Lab, and our collaborators in Dr. Jonathan Hurst's Dynamic Robotics Laboratory at Oregon State University. We also thank Mikhail Jones for guidance on large-scale optimization formulations.

¹The calculation of the cost of transport of a walking gait considers all powers consumed, including the electrical power of central control computer, motor drivers and controllers, motors and sensors.

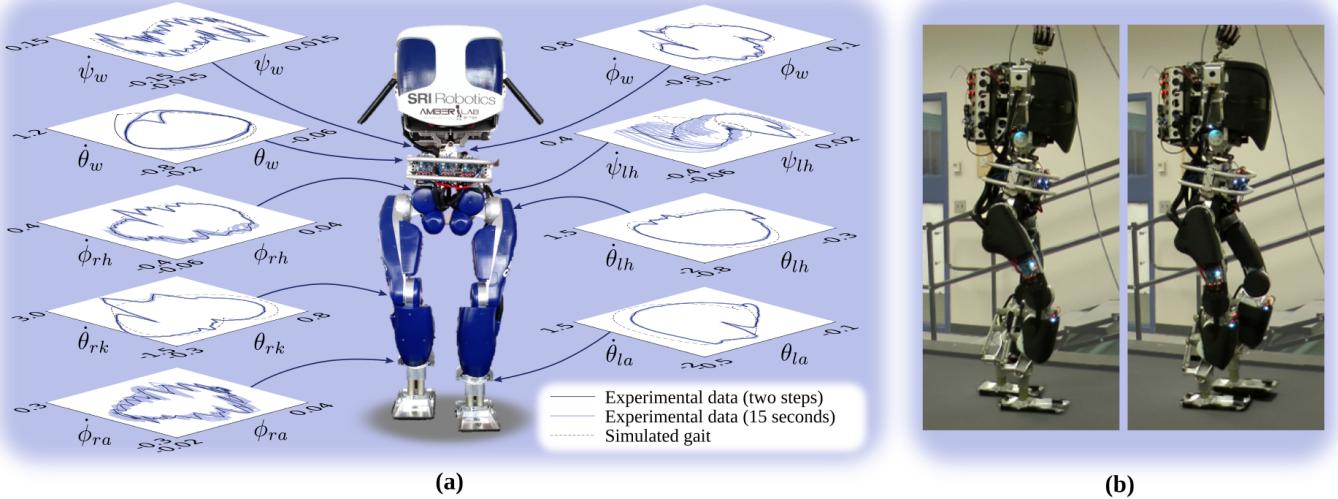


Fig. 8: Experimental results of DURUS walking in 3D at $0.23m/s$. **a)** Periodic orbits from each joint in experiment and overlaid on the simulated gait (units: rad and rad/s ; symmetric joints omitted for clarity). **b)** Tiled still images from the walking experiment.

REFERENCES

- [1] Dynamics walking on DURUS (a.k.a. PROXI) at the 2015 DRC Finals: <https://youtu.be/a-R4H8-8074>.
- [2] A. D. Ames. Human-inspired control of bipedal walking robots. *IEEE Transactions on Automatic Control*, 59(5):1115–1130, May 2014.
- [3] A. D. Ames, E. A. Cousineau, and M. J. Powell. Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 135–144. ACM, 2012.
- [4] A. D. Ames, R. Vasudevan, and R. Bajcsy. Human-data based cost of bipedal robotic walking. In *Proceedings of the 14th international conference on Hybrid Systems: Computation and Control (HSCC)*, pages 153–162. ACM, 2011.
- [5] J. T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*, volume 19. Siam, 2010.
- [6] H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. Sonderforschungsbereich 72, Approximation u. Optimierung, Univ. Bonn, 1983.
- [7] C. Chevallereau, J.W. Grizzle, and C-L. Shih. Asymptotically stable walking of a five-link underactuated 3D bipedal robot. *IEEE Transactions on Robotics*, 25(1):37–50, February 2009.
- [8] C. Chevallereau et al. RABBIT: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, Oct 2003.
- [9] S. H. Collins and A. Ruina. A Bipedal Walking Robot with Efficient and Human-Like Gait. In *IEEE Conference on Robotics and Automation*, number April, pages 1983 – 1988, 2005.
- [10] E. A. Cousineau and A. D. Ames. Realizing underactuated bipedal walking with torque controllers via the ideal model resolved motion method. In *IEEE International Conference on Robotics and Automation*, 2015.
- [11] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM journal on optimization*, 12(4):979–1006, 2002.
- [12] Ch. Glocker and F. Pfeiffer. Dynamical systems with unilateral contacts. *Nonlinear Dynamics*, 3(4):245–259, 1992.
- [13] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames. Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica*, 50(8):1955 – 1988, 2014.
- [14] C. R Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.
- [15] A. Hereid, C. M. Hubicki, E. A. Cousineau, J. W. Hurst, and A. D. Ames. Hybrid zero dynamics based multiple shooting optimization with applications to robotic walking. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.
- [16] Y. Hurmuzlu, F. Génot, and B. Brogliato. Modeling, stability and control of biped robots—a general framework. *Automatica*, 40(10):1647–1664, 2004.
- [17] Y. Hurmuzlu and D. B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *The International Journal of Robotics Research*, 13(1):82–92, 1994.
- [18] S. Kajita et al. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1620–1626. IEEE, 2003.
- [19] S. Kuindersma et al. Optimization-based Locomotion Planning , Estimation , and Control Design for the Atlas Humanoid Robot. *Autonomous Robots*, (0929-5593):1–27, 2015.
- [20] B. Morris and J.W. Grizzle. Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots. *IEEE Transactions on Automatic Control*, 54(8):1751–1764, Aug 2009.
- [21] R. M. Murray, Z. Li, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [22] S. Rezazadeh et al. Spring-mass Walking with ATRIAS in 3D: Robust Gait Control Spanning Zero to 4.3 KPH on a Heavily Underactuated Bipedal Robot. In *Proceedings of the ASME 2015 Dynamic Systems and Control Conference*, 2015.
- [23] S. Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer New York, 1999.
- [24] R. W. Sinnet, M. J. Powell, R. P. Shah, and A. D. Ames. A human-inspired hybrid control approach to bipedal robotic walking. In *18th IFAC World Congress*, pages 6904–11, 2011.
- [25] K. Sreenath, H. Park, I. Pouliquen, and J. W. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.
- [26] B. J. Stephens and C. G. Atkeson. Dynamic Balance Force Control for compliant humanoid robots. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1248–1255, October 2010.
- [27] O. Von Stryk. *Numerical solution of optimal control problems by direct collocation*. Springer, 1993.
- [28] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.
- [29] P. M. Wensing and D. E. Orin. High-speed humanoid running through control with a 3D-SLIP model. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5134–5140, November 2013.
- [30] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton, 2007.
- [31] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, 2003.