A Topological Approach to Gait Generation for Biped Robots

Nelson Rosa Jr. and Kevin M. Lynch, Fellow, IEEE

Abstract—This paper describes a topological approach to generating families of open- and closed-loop walking gaits for underactuated 2D and 3D biped walkers subject to configuration inequality constraints, physical holonomic constraints (e.g., closed-loop linkages), and virtual holonomic constraints (user-defined constraints enforced through feedback control). Our method constructs implicitly-defined manifolds of feasible periodic gaits within a state-time-control space that parameterizes the biped’s hybrid trajectories. Since equilibrium configurations of the biped often belong to such manifolds, we use equilibria as “templates” from which to grow the gait families. Equilibria are reliable seeds for the construction of gait families, eliminating the need for random, intuited, or bio-inspired initial guesses at feasible trajectories in an optimization framework. We demonstrate the approach on several 2D and 3D biped walkers.

I. INTRODUCTION

A challenging problem in bipedal locomotion is the gait-generation problem: given a model of a bipedal robot, generate periodic gaits subject to the biped’s hybrid dynamics and other constraints. We present an approach to the gait-generation problem where equilibria of the biped are used as templates to find families of gaits. Under certain conditions, these equilibria can be continuously deformed into sets of walking gaits, including passive dynamic walking gaits (unactuated gaits where a biped walks downhill under the influence of gravity) and actuated gaits where the biped walks on flat ground or uphill.

In this paper, we assume the biped is physically symmetric about its sagittal plane, and we are interested in symmetric period-one gaits: periodic gaits where each step by the right leg is identical and the mirror image of steps by the left leg is enforced coupling between joints of the biped. Our goal is to find points in $S$ that correspond to period-one gaits.

To precisely define period-one gaits, we define the flow $\phi$ such that $\phi^\tau(x_0)$ is the biped’s state after time $\tau$ using the controls $\mu$ beginning from the state $x_0$. We define the coordinate-flip operator $\text{flip}: \mathcal{X} \rightarrow \mathcal{X}$ that maps a state of the biped to its symmetric state (i.e., the equivalent state when the other leg is taking a step). The flip operator satisfies $\text{flip}(\text{flip}(x_0)) = x_0$. With these definitions, a point $c = (x_0, \tau, \mu) \in S$ corresponds to a period-one gait if and only if $\phi^\tau(x_0) - \text{flip}(x_0) = 0$.

Said another way, the periodicity map $P: S \rightarrow \mathcal{X}$ is defined as

$$P(c) = \phi^\tau(x_0) - \text{flip}(x_0),$$

and the set of all period-one gaits, denoted $G$, is the set of all points $c \in S$ satisfying $P(c) = 0$, i.e., $G = P^{-1}(0)$. Since $P(c) = 0$ specifies $2n$ constraints on the $(2n + k + 1)$-dimensional space $S$, in general we would expect $G$ to be $(k + 1)$-dimensional ($\dim(S) - \dim(P)$).

The goal of our work is not to find a single period-one gait (a single point in $G$), but to map out a “large” continuous family of gaits $G_{\text{mapped}} \subset G \subset S$. The set of gaits $G_{\text{mapped}}$ may include walking downhill, walking uphill, and even hand-to-hand gibbon-like swinging gaits (brachiation) underneath a support. We construct these gait families through the continuous deformation of reference gaits in $G$. A continuous deformation from one gait to another defines a homotopy equivalence between two gaits, a type of topological equivalence [3]. In this respect, we consider our framework to be a topological approach to gait generation because we only consider the connectivity properties of gaits to each other in $G$. This is in contrast to other approaches which focus on other aspects such as the biped’s dynamics [4], [6], [7] to help simplify the search for periodic motions. A long-term goal is a full topological description of $G$ for a given state-time-control space $S$, but that is beyond the scope of this paper.

The standard approach to finding a single gait in $G$ is to formulate a non-convex optimization problem (OP) in the parameters $x_0$, $\tau$, and $\mu$. The convergence of non-convex OPs relies critically on the initial seed value [8], [9], which is typically chosen randomly or by applying domain-specific knowledge [9]–[12]. No general guidelines exist for generic $n$-degree-of-freedom bipeds.

In our framework, however, any one-footed rest state $x_{eq} = (x_{eq}, 0)$ which is also an equilibrium (i.e., $\phi^t(x_{eq}) = x_{eq}$ for some $\mu$ and all $t \geq 0$) is trivially an “equilibrium gait” $c_{eq} = (x_{eq}, \tau, \mu)$. The virtual “impact” after time $\tau$ does not change
the state of the biped, as the biped is at equilibrium, so the “hybrid” motion is trivially periodic.

The set of all such trivial, non-locomoting equilibrium period-one gaits is denoted \( E \), a subset of \( \mathcal{G} \). An equilibrium gait \( c_{eq} \) is often in the same connected component of \( \mathcal{G} \) as useful locomoting gaits, and this motivates the use of numerical continuation methods (NCMs) to generate this connected component starting from \( c_{eq} \). In particular, branches of locomoting gaits intersect an equilibrium branch containing \( c_{eq} \) on the connected component at critical values of the step duration and fixed values of \((x_{eq}, \mu)\) where the rank of the Jacobian of \( P \) at these values is not maximal. In other words, the easy-to-find equilibria are seeds, or “templates,” which are continuously deformed to generate \( \mathcal{G} \).

Having a continuous family of gaits \( \mathcal{G}_{\text{mapped}} \), instead of one or a small number of gaits, can be useful in a number of ways. First, some high-level walking motion planners rely on low-level gait-generation modules, or a pre-computed library of gaits, that can be applied on different terrains \cite{13,14,15,16}. A gait family \( \mathcal{G}_{\text{mapped}} \) constructed using our approach is a continuous version of a gait library. Second, a gait family \( \mathcal{G}_{\text{mapped}} \) allows the possibility of design of control laws that drive the biped to \( \mathcal{G}_{\text{mapped}} \) rather than to a single specific gait \( c \in \mathcal{G} \). In general, it is easier to design a controller to stabilize a manifold than to stabilize a point. Most importantly, \( \mathcal{G}_{\text{mapped}} \) provides a global view of the possible gaits of a biped robot for the given space of design and control parameters \( \mathcal{M} \).

A. Statement of Contributions

This paper describes a topological approach to generating families of walking gaits for 2D and 3D underactuated biped walkers with point, curved, or flat feet that are physically symmetric about their sagittal plane. We use NCMs to map out connected components of gaits in a state-time-control space \( S \). The biped may be subject to configuration inequality constraints, physical holonomic constraints (PHCs) such as closed chains, and virtual holonomic constraints (VHCs), i.e., user-defined constraints enforced through feedback control. Our main contributions are:

1) A **topological approach to the gait-generation problem**. We view gaits as points in a space \( S \) of parameterized trajectories, where we characterize a fundamental property of the periodic orbits of a biped’s hybrid dynamics: their connectivity to each other in \( S \) across variations in state, step duration, and design and control parameters. We take advantage of this connectivity to design algorithms to numerically construct families of gaits in \( S \).

2) The **use of equilibria to generate a continuum of walking gaits**. We prove that we can find families of locomoting gaits that transversally intersect a family of equilibrium gaits in \( E \) at points \( c_{eq} = (x_{eq}, \tau, \mu) \) for a given fixed pair \((x_{eq}, \mu)\). We provide an algorithm for determining the values of \( \tau \) where the intersections occur.

3) A **framework for generating open-loop periodic motions that satisfy the full hybrid dynamics**. We provide a systematic approach using known seed values and a conceptual model of the solution space to the challenging problem \cite{8} of generating open-loop periodic motions for the unactuated joints of a biped robot subject to PHCs and VHCs, including when all joints are unactuated (passive dynamic walking) and when a subset of joints track parameterized trajectories.

This paper builds on our conference paper \cite{17} and the abstract \cite{18}. In this previous work, we introduced the concept of using NCMs to generate gaits for bipeds, including those subject to virtual holonomic constraints. This paper extends our preliminary work in several important ways: 1) we provide a unified framework for generating gaits from equilibrium templates for 2D and 3D underactuated bipeds subject to configuration inequality constraints and physical and virtual holonomic constraints; 2) we provide a new algorithm to find specific types of gaits with desired properties (e.g., a gait that walks on level ground); and 3) we provide applications of the framework to finding gaits for simulated complex 3D bipeds such as Atlas and MARLO.

B. Related Work

Equilibria and numerical continuation methods have a strong history in generating gaits for unactuated biped walkers and brachiators \cite{19}. In particular, the use of equilibria for generating families of unactuated walking gaits can be found in past works studying simple two- and three-degree-of-freedom passive dynamic walking biped models \cite{2,3,20}. The solution families of walking gaits for these biped models converge to an equilibrium gait because the solution families exhibit a vanishing step size—as the biped’s walking slope approaches flat ground, the step size of the corresponding gait becomes shorter. In the limit, as the incline approaches level ground, the state of the biped must approach an equilibrium gait \cite{21}, a “gait” with zero step size. The work in \cite{22} explores this notion of finding periodic walking motions near equilibria for simple walking models with vanishing step sizes. The paper gives necessary conditions on the physical parameters of planar two- and three-link bipeds for walking at arbitrarily small but near-zero slopes.

We extend the work on unactuated, low-dimensional, planar bipeds with vanishing step sizes to include powered high-degree-of-freedom 2D and 3D bipeds. In our previous work \cite{17,23,24}, we used NCMs \cite{25} to generate families of open-loop walking and brachiating gaits that utilize the “natural” or full dynamics of the biped model. In particular, \cite{17} demonstrates that equilibria of representative point-foot planar bipeds can be continuously deformed into families of passive dynamic walking gaits. We extend this body of work to include closed-loop gaits for underactuated bipeds using the hybrid zero dynamics (HZD) framework \cite{4,26,27}.

The HZD framework is an experimentally-validated approach to generating stable walking gaits for underactuated bipedal robots subject to virtual constraints (constraints on the biped that are imposed using feedback control) \cite{11,28}.
The notion of virtual constraints, in particular virtual holonomic constraints, has been a useful concept in the design and control of bipedal walking gaits. We enforce VHCs using an HZD controller, which can provably impose the constraints under mild conditions [3]. Alternative control schemes for enforcing a set of VHCs also exist [30].

A common application of VHCs on a bipedal system is to couple the motion of a subset of joints on an underactuated robot so that they evolve with respect to a function of the biped’s configuration as opposed to time. The resulting motion is then synchronized to, for example, the motion of a biped’s center of mass projected onto its transverse plane when the constraints are properly enforced through feedback control. The net effect is that the biped’s joints move only if the center of mass moves, irrespective of time. In such a case, the motions are said to be self-clocking [28].

Given a biped subject to physical and virtual holonomic constraints, we generate gaits using NCMs, which originate from results in topology and differential geometry [31]. In this context, our application is similar to tracing the points on a surface represented as a set of equations that are continuously differentiable. Applications of continuation methods for generating dynamic motions can be found in [32], [33].

NCMs are also present in optimization solvers, which many gait-generation libraries rely on to generate gaits. NCMs are typically used to find feasible solutions (e.g., elastic mode in SNOPT [34]) or to solve a series of related optimization problems (e.g., interior-point methods [35], like IPOPT).

The standard approach to solving the gait generation problem is to formulate it as an optimization problem (OP) [7], [11]. [12]. The idea is to specify the decision variables, constraints, and objective function used in the optimization in such a way that the underlying solver (often SNOPT, IPOPT, or fmincon) can quickly and robustly converge from an arbitrary seed value [9], [11], [12]. Recent approaches use direct collocation methods as part of the problem formulation, where the biped’s equations of motion are discretized into a set of algebraic constraints using a low-order implicit Runge-Kutta scheme with fixed step size. A comparable optimization-based framework to our work is [11]. In [11], direct collocation methods are used to generate gaits for bipeds subject to VHCs using an HZD feedback controller to enforce the VHCs.

Our use of NCMs to find gaits differs from methods in the literature that rely on OPs in that these works attempt to find the “best” gait while we use NCMs to find many gaits without having to guess an initial seed value. However, with some effort it is possible to modify OPs to generate a continuum of gaits and NCMs to find optimal gaits.

C. Paper Outline

After covering mathematical preliminaries in Section II, we describe the gait space $G$ and how to generate gaits from equilibria using NCMs in Sections III [4] [5]. In Section IV, we give examples of generating gaits for the planar compass-gait walker and the 3D bipeds Atlas and MARLO. In Section V, we compare our approach to FROST [11].

We also provide downloadable material [56] consisting of
1) an MP4 video of walking animations of all biped models used in this paper,
2) a Mathematica v11.3.0 library of our framework and implementation details of the models, and
3) a Node.js v12.17.0 visualization library for animating and creating video clips of the gaits.

II. PRELIMINARIES

In this section, we specify the biped’s hybrid dynamics, give the problem statement, state assumptions, and formally define the space of parameterized trajectories $S$, the gait space $G$, and the connected components of $G$.

A. The Hybrid Dynamics

The hybrid dynamics $\Sigma$ of an $n$-degree-of-freedom biped robot is the tuple $\Sigma = (X, f, \Delta, \phi)$, where

- $X$ is the robot’s state space;
- $f(x,u) \in T X$ describes the continuous dynamics, where $u \in \mathbb{R}^n$ is the robot controls;
- $\Delta : X \rightarrow X$ is a jump map to model instantaneous impacts; and
- $\phi : \mathbb{R} \times X \rightarrow \mathbb{R}$ is a switching function to indicate when a foot hits the ground. If $\phi(t, x) = 0$, then $t \in \mathbb{R}$ is a switching time, $x \in X$ is a pre-impact state, and the foot is in contact with the ground.

The motion of the biped can be subject to $n_p$ physical holonomic constraints and $n_v$ virtual holonomic constraints. The physical constraints, due to closed-loop linkages or kinematic constraints between the foot and the ground, for example, give rise to $n_p$ constraint forces. The virtual constraints are enforced using feedback control. We assume that the biped has $n_u$ ($n_u \geq n_v$) control inputs $u(t) \in \mathbb{R}^n$ to enforce the VHCs in the system.

The VHCs specify the configuration of certain degrees of freedom of the biped as a function of a phase variable $\theta$. In real-time control, the phase variable is often a function of the biped’s state $\mathbb{H}$ (e.g., the swing leg’s joints could be “clocked” by the angle from the stance foot to the hip), but to plan a single step of a gait, time suffices as a phase variable.

In this paper, a VHC takes the generic form

$$q_i(t) - b^d_i(\theta(t), a) = 0, \quad t \in [0, \tau],$$

where $\tau$ is the step duration, $\theta(t) = t/\tau \in [0, 1]$ is the phase variable, $q_i(t) \in \mathbb{R}$ is a joint angle (1 ≤ $i \leq n$), $b^d_i(\theta(t), a) \in \mathbb{R}$ is a Bézier polynomial of degree $d \in \mathbb{N}$, and $a \in \mathbb{R}^n$ is a vector of polynomial coefficients. Appendix A provides further modeling details, including how to enforce a set of VHCs.

B. The Space of Parameterized Trajectories

Given $\Sigma$, we are interested in hybrid trajectories that correspond to a step of a biped of the form

$$x(\tau) = \varphi^\mu_t(x_0) = \Delta(x_0; \mu) + \int_0^\tau f(\varphi^\mu_t(x_0), u; \mu) \, dt,$$
where \( x_0 \in \mathcal{X} \) is the pre-impact state at \( t = 0 \), \( \varphi^I_\mu(x_0) \in \mathcal{X} \) is the state of the robot at time \( t \), \( \varphi^I_\mu(x_0) \) is the next pre-impact state at \( t = \tau > 0 \in \mathbb{R} \), \( \mu \in \mathcal{M} \) is a vector of input parameters, and \( u(t) \in \mathbb{R}^n_u \) is a vector of control inputs that depend on \( \mu \). The parameters \( x_0, \tau, \mu \) define the space of parameterized trajectories.

**Remark 1.** The input parameters \( \mu \) can be used to specify design parameters of the biped such as the center of mass position of a link, leg length, spring coefficients, and moment of inertia. It can also be used to define control parameters, like feedback gains, magnitude of ankle push-off force, and spline coefficients.

**Definition 1.** A biped’s state-time-control space \( \mathcal{S} \) is a finite-dimensional vector space \( \mathcal{S} \subseteq \mathcal{X} \times \mathbb{R} \times \mathcal{M} \subseteq \mathbb{R}^{2n+1+k} \). A point \( c \in \mathcal{S} \), where \( c = (x_0, \tau, \mu) \), defines a hybrid trajectory \( x(t) \in \mathcal{X} \subseteq \mathbb{R}^n_p \) given input parameters \( \mu \in \mathcal{M} \subseteq \mathbb{R}^k \) starting from \( x_0 \in \mathcal{X} \) at switching time \( t = 0 \) until the next switching time \( \tau > 0 \).

Figure 1 shows how the parameters of the state-time-control space \( \mathcal{S} \) can affect the motion of an \( n \)-degree-of-freedom biped robot. As defined earlier, the set of all period-one gaits in \( \mathcal{S} \) is defined using the periodicity map \( P \) as

\[
\mathcal{G} = \{ c \in \mathcal{S} : P(c) = 0 \},
\]

i.e., the set of all points \( c = (x_0, \tau, \mu) \) satisfying \( \varphi^I_\mu(x_0) - \text{flip}(x_0) = 0 \). The set of equilibrium (stationary) gaits is defined as

\[
E = \{ c_{eq} = (x_{eq}, \tau, \mu) \in \mathcal{G} : f(x_{eq}, u(t); \mu) = 0 \forall t \in \mathbb{R} \},
\]

i.e., the set of all points \( c_{eq} \) satisfying \( P(c_{eq}) = 0 \) and \( \varphi^I_\mu(x_{eq}) = x_{eq} \) for all \( t \).

**C. The Connected Components of the Gait Space \( \mathcal{G} \)**

**Definition 2.** Let \( \mathcal{G} \) be the space of all gaits in \( \mathcal{S} \).

1) A path between two points \( a \) and \( b \) in \( \mathcal{G} \) is a continuous function \( p : [0, 1] \to \mathcal{G} \) such that \( p(0) = a \) and \( p(1) = b \).

2) A set \( X \subseteq \mathcal{G} \) is path-connected if for all \( a, b \in X \), there exists a path \( p : [0, 1] \to X \) with \( p(0) = a \) and \( p(1) = b \).

3) A set \( X \subseteq \mathcal{G} \) is a connected component of \( \mathcal{G} \) if \( X \) is path-connected and \( X \) is maximal with respect to inclusion.

**Theorem 1.** [37] Let \( D \) be an open set in \( \mathcal{G} \) and the periodicity map \( P : \mathcal{S} \to \mathbb{R}^{2n} \) be a class \( C^r \) differentiable function. If for every \( c \in D \), the Jacobian \( J(c) \in \mathbb{R}^{2n \times (2n+k+1)} \)

\[
J(c) = \frac{\partial P}{\partial c}(c) = \begin{bmatrix} \frac{\partial c}{\partial x}(c) - I_{2n}, & \frac{\partial \varphi}{\partial \tau}(c), & \frac{\partial \varphi}{\partial \mu}(c) \end{bmatrix}
\]

has maximal rank \( 2n \), then \( D \) is a \((k+1)\)-dimensional \((C^r \) differentiable) manifold in \( \mathcal{G} \).

For a point \( c \in \mathcal{G} \), we have from [25], [38]

\[
T_c \mathcal{G} = \text{Null}(J(c)),
\]

where \( \text{Null}(J(c)) \) is the null space of \( J \) from Equation 1 and \( T_c \mathcal{G} \) is the tangent space of \( \mathcal{G} \) at \( c \).

**Definition 3.** A point \( c \in P^{-1}(0) \) is a singular point of \( P \) if \( \text{rank}(J(c)) < 2n \). Points are regular if they are not singular.

The connected components of \( \mathcal{G} \) generally consist of submanifolds of \( \mathcal{S} \) glued together at singular points of the periodicity map \( P \).

**D. Problem Statement**

**Given**

- a hybrid model \( \Sigma = (\mathcal{X}, f, \Delta, \phi) \) of a biped,

- a finite-dimensional space \( \mathcal{S} \) of parameterized trajectories,

- an implicit description of the set of all gaits \( \mathcal{G} \subseteq \mathcal{S} \) as the points \( c \in P^{-1}(0) \), and

- a description of the set of equilibria \( E \subseteq \mathcal{G} \),

use NCMs to approximately trace the connected components of \( \mathcal{G} \) that contain \( E \). The constructed set is denoted \( \mathcal{G}_{\text{mapped}} \).

**E. Assumptions**

**Assumption 1.** Unless otherwise stated, we assume

A1 Bipeds are physically symmetric about their sagittal plane.

A2 Bipeds undergo exactly one collision per step, a plastic impact between the pre-impact swing leg and the ground. At impact, the pre-impact stance leg breaks contact with the ground, and there is no free-flight or double-support phase (the stance leg instantaneously changes at impact). No slipping occurs at contacts between a foot and the ground.

A3 The next foot hits the support surface after a specified period of time has elapsed, i.e., the impact is based on time, not state.

A4 Bipeds may be subject to physical and virtual holonomic constraints, but not nonholonomic constraints.

Assumptions A1–A2 allow us to take advantage of a biped’s symmetry to define a gait after one step with only one impact. Additional impacts would be needed to model the knees of a walker hitting a mechanical stop to prevent hyperextension [10], [40]. Assumption A2 also rules out heel-toe collisions for bipeds with non-point feet [7], [11]. This is a common assumption for point-, curved-, and flat-footed

---

**Fig. 1.** A generic \( n \)-degree-of-freedom biped model (left). We parameterize motions that satisfy the model’s hybrid dynamics (top right) with a pre-impact state \( x_0 \in \mathcal{X} \subseteq \mathbb{R}^{2n} \); a switching time \( \tau \in \mathbb{R} \), and a vector of input parameters \( \mu \in \mathcal{M} \subseteq \mathbb{R}^k \) (top and bottom right). We use the vector \( \mu \) to represent any parameter that is not a state or switching time. In this example, the vector \( \mu \) consists of \( k \) polynomial coefficients (bottom right) used to define the trajectory of joint \( q_i \) of the biped.
Given Assumption A3, the hybrid dynamics \( \Sigma = (\mathcal{X}, f, \Delta, \phi) \) with fixed switching times is

\[
\begin{align*}
\Sigma : \quad & \dot{x}(t) = f(x(t), u(t)) \quad t \neq k\tau, \\
& x(t^+) = \Delta(x(t^-)) \quad t = k\tau, \tag{5}
\end{align*}
\]

where \( x(t) \in \mathcal{X} \), \( f(x, u) \in T\mathcal{X} \), and \( \Delta(x) \in \mathcal{X} \) are the state of the robot at time \( t \), a vector field, and a jump map, respectively (see Equation (3)), \( x(t^-) \) and \( x(t^+) \) are pre- and post-impact states, respectively, and \( k \geq 0 \in \mathbb{Z} \) is the \( k \)th impact. The switching function \( \phi = \phi(t, x) = t - k\tau \).

Finally, Assumption A4 precludes the use of virtual non-holonomic constraints as discussed in [20].

### III. The Basic Gait-Generation Approach

We now present the core concepts and algorithms behind our framework. Specifically, we

1. describe the connected components of \( G \) that include equilibrium gaits (EGs) (Section III-A),
2. describe how to find paths from an EG \( c_{eq} \in E \subseteq G \) to a set of nonstationary gait trajectories (nontrivial periodic orbits of the hybrid dynamics) in \( G - E \) (Section III-B),
3. present a continuation method for tracing curves (one-dimensional manifolds) in \( G \) (Section III-C),
4. present an algorithm for constructing one-dimensional slices of \( G \) from EGs (Section III-D), and
5. give an illustrative example of the basic approach using the compass-gait walker (Section III-E).

### A. Connected Components of \( G \) Containing Equilibrium Gaits

The space of gaits \( G \) in the state-time-control space \( S \) may consist of multiple separate connected components [24]. We are particularly interested in those connected components that include EGs. A biped standing still on one foot is an example of an EG.

Figure 2 is a conceptual depiction of a connected component of an EG for an \( n \)-degree-of-freedom biped walker with one switching time and \( k = 1 \) design and control parameters. The dimension of \( S \) is \( 2n+k+1 = 2n+2 \) and the dimension of the manifolds in \( G \) are \( \dim(S) - 2n = k + 1 = 2 \)-dimensional. For illustrative purposes, we assume the connected component consists of manifolds of unactuated and actuated gait families. Examples of manifolds in the figure are the red-dashed and green curves (1D slices of a larger 2D manifold) and the green surface (a 2D manifold). Specifically,

- the green curves are 1D gait families of unactuated (passive dynamic walking) gaits,
- the green surface is a 2D branch of actuated gaits (we could, for example, expect \( p(1) \in \mathcal{X} \) to correspond to an actuated gait that walks on flat ground),
- the red dashed line consists of equilibrium gaits from which we would start to construct \( G_{\text{mapped}} \) (e.g., the EG \( p(0) \in E \)),
- the two singular points (black dots with thick red borders) divide the red dashed line into three distinct manifolds, and
- these singular points glue together the branches of unactuated and equilibrium gaits of \( X \).

The connected component of an EG can be understood in terms of previous work with simple two-degree-of-freedom passive walkers [2], [3], [20]. While the hybrid dynamics in these works use state-based switching functions \( \phi = \phi(x) \), the search for an initial set of gaits reduces to finding explicit values of the step duration \( \tau \) that are roots of the linearized periodicity map \( P \) with respect to state \( x_{eg} \); the linearized map is derived as a closed-form, Taylor-series approximation of \( P \) about an equilibrium point \( x_{eq} \).

The connected component in Figure 2 gives a geometric interpretation of this approach. The search for roots of the linearized map is the same as the search being constrained to the red dashed line of \( X \) in a constant-control slice of \( \mathcal{S} \). The critical values of \( \tau \) that are roots of the linearized map correspond to singular EGs on the line. As can be seen in the figure, regular EGs cannot have critical values of \( \tau \) because nearby walking gaits do not exist when we only consider perturbations in state in a constant-control slice.

In this paper, we generalize (and formalize) this analytical approach to multi-degree-of-freedom bipeds. Given an EG, the task is to find paths from the EG in \( E \) to a set of gaits in \( G - E \) on the EG’s connected component. Across biped models, EGs have two useful properties that we take advantage of in the construction of \( G_{\text{mapped}} \), which we prove in Section III-B

P1 An EG in the gait space \( G \) is always a part of a continuum of EGs of the form \( \{x_{eq}, \tau, \mu\} \), where \( x_{eq} \) and \( \mu \) are fixed and \( \tau \) can take any positive value \((\tau > 0)\), and

P2 In a constant-control slice of \( \mathcal{S} \), the same continuum of EGs often intersects with a branch of walking gaits residing in the slice. The point of intersection can only happen at EGs that are singular points of \( P \).

For Property P1, the continuum of EGs only exists because we assume a robot’s swing foot can impact the ground at any time (Assumption A3). For the biped standing still on one
foot, Assumption A3 implies that the state $\varphi^T_\mu(x_{eq})$ at the end of the next step (the biped standing still on its opposite foot) does not change for any value of $\tau$ and fixed values of $x_{eq}$ and $\mu$. In the robot’s gait space, the resulting set of points $(x_{eq}, \tau, \mu)$ would correspond to the red dashed line of regular and singular EGs on the connected component in Figure 2.

Property P1 also places a constraint on an EG’s tangent space (Definition 3). For a regular EG in a constant-control slice, the constraint fully defines the EG’s 1D tangent space such that it can only be a part of a single branch.

**Definition 4.** A path-connected set of EGs that are regular points of the periodicity map $P$ form an equilibrium branch on the connected component. Equilibrium branches reside in constant-control slices of $S$.

Property P2 implies that this is not the case for singular EGs. At singular EGs in a constant-control slice, we can switch onto a non-equilibrium branch of gaits and, for example, trace a branch of walking gaits (dark green curves in Figure 2) for inclusion in $G_{\text{mapped}}$. These gaits can then be used to add gaits not in the constant-control slice (e.g., gaits on the 2D surface).

We specify a constant-control slice of $G$ with the map $M_0 : S \rightarrow \mathbb{R}^{2n+k}$

$$M_0(c) = [P^T(c), \Phi^T_\mu(c)]^T, \quad \Phi_0(c) = \mu - \mu_0,$$

where $G_0 = M_0^{-1}(0) \subset S$ is the set of gaits of $M_0$ and the $k$ auxiliary constraints $\Phi_0(c) = 0$ keep the controls constant at $\mu_0$. In other words, $G_0$ lives in a constant-control slice of $S$ (the red and dark green curves in Figure 2). The set of equilibrium gaits of $M_0$ is $E_0 = E \cap G_0$. We specify how to find singular EGs in $E_0$ in the next section. In general, we use the subscript 0, as in $M_0$, for variables related to a constant-control slice $G_0$ in $G$.

**Remark 2.** Under any state-based switching strategy, EGs are either isolated points in $G_0$, or not in $G_0$ at all, depending on whether conditions placed on the switching function $\phi$ allow for infinite impacts in zero time [4].

**Remark 3.** The locations of singular EGs on a connected component are determined by the kinematic and dynamic properties of the biped. Any changes to the biped model, for example, the addition of physical or virtual holonomic constraints, modifications to the control inputs $u$, or changing the biped’s total mass, can shift the singular points or change the number of singular points on the connected component.

### B. Detecting Singular Equilibrium Gaits

Given the characterization of equilibrium gaits in Section III-A, consider the set of EGs $\{c_{eq} = (x_{eq}, \tau, \mu) \mid \tau > 0\}$ for fixed control inputs $\mu$. For this set, the indicator function

$$I(\tau) = \det \left( \frac{\partial P}{\partial x_{eq}}(x_{eq}, \tau, \mu) \right)$$

(7)

can be used to identify singular EGs by searching along the switching-time axis for values of $\tau$ that make $I(\tau)$ zero.

Given the map $M_0$ (Equation 6) and the Jacobian of $M_0$,

$$J_0(c) = \begin{bmatrix} \frac{\partial P}{\partial x_{eq}}(c) \\ \frac{\partial P}{\partial \tau}(c) \\ \frac{\partial P}{\partial \mu}(c) \\ 0 \\ 0 \end{bmatrix},$$

the next two propositions prove that we can find a path from an equilibrium point $c_{eq} \in E_0$ to a set of gaits in $G_0 - E_0$ by searching for singular EGs in $E_0$. In the first proposition, we establish the existence of 1D equilibrium branches in $G_0$, which leads to a corollary that gives the condition for when an EG is a regular point of $M_0$.

**Proposition 1.** Given

1) a biped’s hybrid dynamics $\Sigma = (X, f, \Delta, \phi)$,
2) an equilibrium point $x_{eq} \in X$ of $f$,
3) a switching time $\tau_0 \in \mathbb{R}$, and
4) a vector of control parameters $\mu_0 \in \mathbb{R}^k$

such that $M_0(c_0) = 0$, where $c_0 = (x_{eq}, \tau_0, \mu_0) \in E_0$, if $c_0$ is a regular point of $G_0$, then there exists a unique curve $c:(-\delta, \delta) \rightarrow E_0$ of regular points contained in $E_0$ that passes through $c_0$ at $c(0) = c_0$ for some $\delta > 0$.

**Proof.** See Appendix B.

**Corollary 1.** If $c_0 \in E_0$, then $\frac{\partial P}{\partial \tau}(c_0) = 0$ and $J_0(c_0)$ has rank of at most $2n + k$. Furthermore, if $c_0 \in E_0$ is a regular point of $M_0$, then the submatrix

$$\begin{bmatrix} \frac{\partial P}{\partial x_{eq}}(c_0) \\ \frac{\partial P}{\partial \mu}(c_0) \\ 1_k \end{bmatrix} \in \mathbb{R}^{(2n+k) \times (2n+k)}$$

of the Jacobian $J_0$ of Equation (8) has full rank $2n + k$.

The next proposition states that $I(\tau)$ of Equation (7) can detect singular EGs in $E_0$.

**Proposition 2.** Assume there exists a path $p : [0,1] \rightarrow G_0$ such that $p(0) \in E_0$ and $p(1) \in G_0 - E_0$. If $p(0)$ is a regular point of $M_0$, then

1) the path $p$ contains at least one singular equilibrium point $p(s) \in E_0$, and
2) for each singular equilibrium point $p(s) \in E_0$ for $s \in (0,1)$, $\det \left( \frac{\partial P}{\partial x_{eq}}(p(s)) \right) = 0$.

**Proof.** See Appendix B.

After identifying a singular EG using the indicator function, the next step is switching onto a branch of gaits in $G_0 - E_0$. We can determine the correct branch for the case where the singular EG, say $c_0$, is isolated in $E_0$ and its tangent space $T_{c_0}G_0$ is two dimensional.

**Proposition 3.** If $c_0 \in E_0$ is an isolated singular point in $E_0$ and $\dim(T_{c_0}G_0) = 2$, then taking a step in the direction of the tangent vector in $T_{c_0}G_0$ orthogonal to the switching-time direction switches onto a branch of gaits in $G_0 - E_0$.

**Proof.** See Appendix B.

Given Propositions 1 and 2, we can automate the search for a singular EG along the switching-time axis (keeping the state and control constant) using Algorithm 1. Algorithm 1 finds simple roots of Equation (7) (i.e., if $I(\tau) = 0$, then $\frac{\partial I}{\partial \tau}(\tau) \neq 0$) in a given interval by applying the intermediate value theorem to first bracket a root and then switching to a root-finding algorithm to accurately find the root. The step size $h$ should be chosen with care to avoid skipping over multiple roots in a given subinterval (we use $3 \times 10^{-13}$). Alternative univariate...
Algorithm 1 Detecting singular equilibrium gaits

Require: an interval \([a, b] \subset \mathbb{R}\) and a step size \(h \in \mathbb{R}\).
1: Define functions \(c_{eq}(t) = (x_{eq}, \tilde{r}_0 + t, \mu_0)\) and \(\delta(t) = \det \left( \frac{\partial}{\partial t} (c_{eq}(t)) - \mathbb{I}_{2n} \right)\).
2: if \(\delta(t) \neq 0\) with \(t \in [a - h, a] \cup [b - h, b]\) then
3:   \(N = \left\lfloor \frac{b - a}{h} \right\rfloor\).
4:   \(t = a + i \times h\) for \(i = 1..N\) do
5:     \(\delta(t) > \delta(t - h) > 0\) if
6:     \(\delta(t) > 0\) if \(t \in [t - h, t]\) then
7:     \(\delta(t) = 0\) and \(\delta(t - h) < 0\) if
8:     \(\delta(t) < 0\) if \(t \in \mathbb{R}\) and \(\delta(t - h) > 0\) if
9:     \(\delta(t) = 0\) and \(\delta(t - h) < 0\) if
10: \(\delta(t) = 0\) and \(\delta(t - h) < 0\) if
11: \(\|\frac{\partial}{\partial s} (c_{eq}(t))\| = 1\) and
12: \(\|\frac{\partial}{\partial s} (c_{eq}(t))\| = 1\) and
13: \(\text{return} \text{ singular EGs and tangent vectors} (c_{eq}(t_0), \frac{dc_{eq}}{ds}(t_0))\) to satisfy periodic boundary conditions of Equation \(1\) at \(t = 0\) and \(t = \tau\).

Algorithm 2 Pseudo-arclength continuation method

Require: \(M : \mathbb{R}^{2n+k+1} \rightarrow \mathbb{R}^{2n+k}\) and step size \(h \in \mathbb{R}\).
1: function CMSTEP(c, \(\dot{c}, h\))
2: \(M(c) = 0, \frac{\partial M}{\partial c}(c)\dot{c} = 0\), and \(||\dot{c}|| = 1\)
3: Prediction Step:
4: \(z = c + \dot{c}h\)
5: Correction Step:
6: Solve for \(M(z) = 0\) and \(\dot{c}^T(z - c) = h\)
7: using Newton’s method
8: \text{return} \(z\)
9: end function
10: function CMCURVE(c0, \(\frac{dc_{eq}}{ds}, M\))
11: Set \(c(0) = c_0\) and \(\dot{c}(0) = \frac{dc_{eq}}{ds}\).
12: \(\text{for} \ i := 1..N \text{ do}\)
13: \(\dot{c}(i) = \text{CMSTEP}(c(i - 1), \dot{c}(i - 1), h)\)
14: \(\text{Set} \ \dot{c}(i) \ \text{such that} \ \frac{\partial M}{\partial c}(c(i))\dot{c}(i) = 0\) and \(||\dot{c}(i)|| = 1\)
15: \(\text{if} \ \dot{c}^T(i)|\dot{c}(i) - 1| < 0 \text{ then}\)
16: \(h = -h\)
17: \text{end if}\)
18: \text{end for}\)
19: \text{return the solution curve} \(c\)
20: \text{end function}\)

root-finding algorithms can be found in \([42]\). In the end, all singular EGs detected using Algorithm 1 are isolated singular tangency points that are orthogonal to the switching-time dimension, and the map \(M_{eq}\) serve as inputs to the numerical continuation method (NCM) of the next section.

Remark 4. Conditions for when a singular EG is isolated can be found in \([38]\). In Proposition 3 we assume isolated singular points with \(\dim(T_{c_{eq}}G_{eq}) = 2\) because it is the most common type of singular EG we encounter in practice.

C. Tracing Branches with Numerical Continuation Methods

Continuation methods are useful numerical tools for tracing the level set of a continuously differentiable function. While multi-dimensional continuation methods exist \([25], [38], [43]\), we present an NCM for tracing one-dimensional manifolds. The map \(M_{eq}\) is as smooth as the map \(M_{eq}\) is implicitly defined such that every point on the plane as the initial guess using an Euler-like root-finding algorithm for use in lines 6–7 of Algorithm 2. The tangent space of \(M_{eq}\) is the polynomial coefficient vector of the Bézier polynomials of Equation \(1\) and \(\mu = [\ldots, a^T, \ldots]^T\), then \(x_0\) and \(a\) have Geometrically, Algorithm 2 defines a hyperplane a distance \(h\) away from the current point \(c(s_i)\) on the curve where the search for the next point on the curve \(c(s_{i+1})\) takes place; the hyperplane is normal to the tangent \(\frac{dc}{ds}(s_i)\) at \(c(s_i)\). The algorithm’s prediction step (line 4 of the algorithm) selects a point on the plane as the initial guess using an Euler-like integration step and then a root-finding method iteratively refines the guess until the point is on the curve.

In order to define the hyperplane, we can compute a tangent to the curve \(\frac{dc}{ds}(s_i)\) at \(c(s_i)\) by solving for \(\frac{dc}{ds}(c(s_i))\frac{dc}{ds}(s_i) = 0\) (line 13 of Algorithm 2). In other words, the tangent \(\frac{dc}{ds}(s_i)\) is in the null space of the Jacobian of the map \(M\)

\[ T_{c(s)}M^{-1}(0) = \text{Null} \left( \frac{\partial M}{\partial c}(c(s)) \right) \]

where \(T_{c(s)}M^{-1}(0)\) is the tangent space of \(M^{-1}(0)\) at \(c(s)\). An arclength parameterization of the curve leads to the constraint \(||\frac{dc}{ds}(s)|| = 1\) (see \([38]\)). If the point \(c(s)\) is a regular point of \(M\), then \(\dim(T_{c(s)}M^{-1}(0)) = 1\) (\(= \dim(S) - (2n + k)\) constraints).

Algorithm 3 is the projected Newton’s method \([45], [46]\), a root-finding algorithm for use in lines 9–11 of Algorithm 2. The projected Newton’s method is a variant of Newton’s method that imposes box constraints \(L \leq c \leq U\) on the values of \(c \in S\), where \(L, U \in S \cup \{\pm \infty\}\) specify the lower- and upper-bounds of \(c\), respectively, and the relational operators are applied elementwise. An application of the projected Newton’s
method is modeling inequality constraints, like the swing leg of a biped staying above or on the walking surface, as equality constraints (see Section V for an example).

Remark 5. Part of the input to Algorithm 2 is a vector \( \frac{dc}{ds}(s_i) \) tangent to the initial point \( c_0 \). If the manifold \( M^{-1}(0) \) is a one-dimensional differentiable manifold, then the tangent space is one-dimensional and there is no need to pass \( \frac{dc}{ds} \) as an argument; the algorithm can compute the tangent internally. However, we use NCMs to generate branches of connected components starting from a singular point, in which case we do need to specify the tangent vector as the null space has dimension greater than one [38]. If we do not specify the tangent at a singular point, the behavior of the algorithm is implementation dependent.

D. From Equilibria to One-Dimensional Sets of Walking Gaits

Given an EG \( c_{eq} \in E_0 \), Algorithm 2 constructs \( G_{mapped} \) in a constant-control slice \( M_0(c) = 0 \) in \( S \). For simplicity, we define \( G_{mapped} \) as a two-dimensional array of gaits, but other data structures can be used. Most of the details of the algorithm are covered in Sections III-B and III-C (e.g., lines 2 and 8). In particular, \( N \) new gaits are added to \( G_{mapped} \) when we successfully return from calls to Algorithm 2 (line 8). In the event that Algorithm 2 is not able to find isolated singular EGs, an error message is printed along with a plot of the indicator function \( I \) (lines 11-12). Further analysis of the plot provides potential directions for improving the model. We list an informal list of steps in Appendix C.

E. An Illustrative Example of the Basic Gait-Generation Approach Using the Compass-Gait Walker

We present an example of extending equilibria to periodic orbits for the passive compass-gait walker [2]. The compass-gait walker is a common two-link model. The model (Figure 4(a)) consists of two legs each with a point mass \( m \) and length \( a+b \). The biped also has a large point mass \( m_H \) at the hip. We use the same values for the physical parameters as in [2] with \( \frac{m_H}{m} = 2 \), \( \frac{b}{a} = 1 \), and \( g = 9.81 \text{ m/s}^2 \). The state of the robot is \( x = [q_1, q_2, q_1', q_2']^T \in \mathbb{R}^4 \), representing the two leg angles and their velocities. With this minimal set of coordinates, we can directly compare our results to those in [2]. The angle of the walking surface \( \sigma \in \mathbb{R} \) is implicitly defined according to the position of the swing leg’s foot at \( t = 0 \): \( \sigma = \frac{1}{2}(q_1(0) + q_2(0)) \). The biped has no motors \( (n_a = 0) \), no VHCs \( (n_v = 0) \), and two PHCs \( (n_p = 2) \) representing the no-slip contact conditions between the stance foot and the ground. We do not place constraints on the resulting ground reaction force arising from the PHCs.

After using this data to derive the biped’s hybrid dynamics, the goal is to find period-one walking gaits in a five-dimensional state-time space \( S \). A point \( c \in S \) consists of the pair \( (x_0, \tau) \), where \( x_0 \in \mathcal{X} \) is a pre-impact state and \( \tau \in \mathbb{R} \) is a switching time (in seconds). There are no control parameters \( \mu \) \((k = 0)\). Given the five parameters that define \( S \) and the four periodicity constraints of the periodicity map \( P \), we expect to find one-dimensional manifolds of gaits in \( S \). The search for a walking gait starting on a manifold of EGs is a two-step

Algorithm 3 Projected Newton’s method

Require: \( r : S \rightarrow \mathbb{R}^m \), where \( m \leq \dim(S) \).
1: function PROJNEWT(c, L, U)
2: \( z = c \)
3: repeat
4: Compute Newton Step:
5: \( d = \frac{dc}{ds}(z)^\dagger r(z) \), where \([;]^\dagger\) is the Moore-Penrose inverse
6: \( \Delta = z - d \)
7: Project onto Box Constraints:
8: for \( 1 \leq i \leq (2n_k + k + 1) \) do
9: \( \text{if } \Delta[i] \leq L[i] \) then
10: Set to lower bound: \( z[i] = L[i] \)
11: \( \text{else if } \Delta[i] \geq U[i] \) then
12: Set to upper bound: \( z[i] = U[i] \)
13: \( \text{else} \)
14: Take Newton step: \( z[i] = \Delta[i] \)
15: end if
16: end for
17: until a stopping criterion is met
18: return \( z \)
19: end function

Algorithm 4 Constructing \( G_{mapped} \) from equilibria

Require: \( c_{eq} \in E_0 \).
1: Search for Singular Equilibrium Gaits:
2: Call Algorithm 1 with a search interval of \( \tau \in [a, b] \).
3: Store singular EGs and their tangent vectors in arrays
4: \( A \) and \( \bar{A} \), respectively.
5: Generate Curves in \( G - E \):
6: if \( |A| > 0 \) then
7: \( \text{for } i := 1..|A| \) do
8: \( G_{mapped}[i-1][0..N] = \text{CMCURE}(A[i], \bar{A}[i], M_0) \)
9: end for
10: else
11: Print “No isolated singular equilibrium gaits found.”
12: Print plot of Indicator Function \( I(\tau) \) for \( \tau \in [a, b] \).
13: end if
14: return \( G_{mapped} \)
process:
1) Identify a subset of EGs of interest in \( E \subseteq \mathcal{G} \).
2) Choose a \( x_{eq} \in E \) and call Algorithm 1 which
   a) calls Algorithm 2 to find all singular EGs in a closed
      interval of switching times \( \tau \in [a, b] \subseteq \mathbb{R} \) where \( 0 \leq a < b \), then
   b) calls Algorithm 2 with the map \( M_0 = P \), a singular
      EG \( x_{eq} \), and the correct tangent vector \( \frac{dx_{eq}}{d\tau}(s) \in T_{x_{eq}} \mathbb{R}^3(0) \).

The first step for the compass gait is straightforward. The
biped’s state space \( \mathcal{X} \) has four one-stance-foot equilibrium
points, but only two equilibria in \( \mathcal{X} \) correspond to fixed points
of \( \mathcal{G} \) because of the flip operator. These are \( x_{eq} = [0, 0, 0, 0]^T \)
(standing on a surface) and \( x_{eq} = [\pi, \pi, 0, 0]^T \) (hanging below
a surface), as shown in Figure 4(b)–(c). These points define
of \( \mathcal{G} \) at \( \tau = 0 \), \( \tau = 0.62 \) s and \( \tau = 0.68 \) s for the compass
gait correspond to the start of the “short” and “long” solution
branches of walking gaits as reported in [2]. The (symmetric)
green branches that extend from each singular point contain
gaits that are mirror images of each other, i.e., if one branch
has state \( x_{eq} \), the other branch has state \(-x_{eq} \); the sign indicates
whether the gait walks downhill to the left or right of its initial
stance.

In order to generate the gaits in Figure 5(a), we do not
constrain the ground reaction force generated during a step
between the stance foot and ground [47]. For gaits located on
the thicker portions of the green curves, the ground reaction
force pushes and pulls the stance foot during a step.

In Figure 5(b)–(g), we see the existence of gaits in \( \mathcal{G}_{mapped} \)
over a range of slopes that include walking and overhand
brachiating gaits. Each of the gaits in Figure 5(b)–(g) can
continuously be deformed into each other and are part of
the same connected component of the EG \( (x_{eq}, 0) \) depicted
in Figure 4(b).

IV. Extensions to the Approach

In the previous section, we focused on constant-control
slices in \( \mathcal{G} \). We now present extensions to the approach for
1) constructing multi-dimensional manifolds in \( \mathcal{G}_{mapped} \), from
EGs (Section IV-A), and
2) searching the manifolds of \( \mathcal{G} \) for gaits with desired
properties, for example, specific walking speeds or values
for \( x_{eq} \), \( \tau \), or \( \mu \) (Section IV-B).

These extensions expand our work beyond passive dynamic
walkers. In particular, for biped models that cannot balance on
one foot, EGs for use as input to Algorithm 1 may not exist
or may be difficult to find. In order to handle these types of
models, we add control parameters that continuously modify
the physical parameters of the biped model. For example, in
Section V, we introduce a control parameter \( \omega \in [0, 1] \) that
parameterizes a family of MARLO biped models with different
hip widths and center of mass positions such that \( \omega = 0 \)
corresponds to a planarized version of MARLO and \( \omega = 1 \)
corresponds to the 3D model used in [26]. The purpose of
parameters such as \( \omega \) is to start with a model that has a simple
set of EGs that can be used as input to Algorithm 1 (e.g., the
EGs for MARLO at \( \omega = 0 \)) and to then eventually connect
these gaits to a family of walking gaits of the desired biped
model (for MARLO, walking gaits with control parameter
\( \omega = 1 \)). Given that the number of control parameters \( k \) will
necessarily be greater than zero, this motivates the use of
algorithms that can search higher-dimensional manifolds for
desired gaits.

A. Constructing Multi-Dimensional Manifolds

Algorithm 5 is an example of a higher-dimensional con-
tinuation method. It uses the map \( M_0 \) and a collection of \( k \)
Fig. 5. (a) A continuous set of unactuated periodic motions $G_{\text{mapped}}$ that satisfy the compass gait’s hybrid dynamics as points in a parameter space $S$ projected onto a slope-switching-time ($\sigma$-$\tau$) plane (red and green curves); the slope $\sigma$ is the biped’s walking surface. The plot consists of two singular equilibrium gaits (black dots with thick red borders), three equilibrium branches (red curves separated by the two singular points), and four 1D manifolds of walking and brachiating gaits (green curves). Specifically, each of the two leftmost green curves are linear interpolations of 250 gaits computed with Algorithm 4 (the maximum number of gaits we have Algorithm 4 compute), the two rightmost green curves consist of the first 113 points we computed on each branch, and the red line is added after the fact to represent the equilibrium branches. The thicker regions of the green curves correspond to walking gaits, where the ground pushes and pulls the foot of the robot in order to maintain the no-slip constraint. The callout labels in the plot correspond to the animated trajectories in (b)–(g); the images in the plot are the pre-impact configurations of the biped at $t = 0$. (b)–(g) The motion of the gaits depicted in (a) with respect to (absolute) time $t \geq 0$. Gaits (b), (c), (f), and (g) locomote from right to left and (e) goes from left to right. Because of the biped’s symmetry, every gait in (a) has a mirrored version of itself about the $\tau$ axis. In other words, trajectories in (b)–(g) that locomote on a slope $\sigma$ have mirrored trajectories that walk or brachiate downhill on a slope of $-\sigma$. 

(a)

(b)

(c)

(d)

(e)

(f)

(g)
Algorithm 5 Constructing multi-dimensional manifolds

Require: singular equilibrium gait $c_{eq} \in E_0$.
1: function MULTI-DIM($d$) 
2: if $d = 0$ then 
3: return \{ $c_{eq}$ \} 
4: else 
5: $G_{d-1} = \text{multi-dim}(d-1)$ 
6: return $\bigcup_{g \in G_{d-1}} \text{CMCURVE}(g, \frac{\partial p}{\partial s}, M_{d-1})$ 
7: end if 
8: end function

additional maps $M_1, \ldots, M_i, \ldots, M_k$ such that the level set $M_i(c) = 0$ defines a constant slice in $\mathcal{S}$ where the switching time $t$ and all but the $i^{th}$ control parameter are fixed:

$$M_i(c) = [P^T(c), \Phi_i^T(c)]^T,$$

$$\Phi_i(c) = [\tau - t, \mu_1 - v_1, \ldots, \mu_{i-1} - v_{i-1},$$

$$\mu_i + v_i, \ldots, \mu_k - v_k]^T,$$

where $c = (x_0, \tau, \mu) \in \mathcal{S}$ is the input; $1 \leq i \leq k$ is the $i^{th}$ control parameter in $\mu$ that varies throughout the continuation; $P$ is the periodicity map; $\Phi_i : \mathcal{S} \rightarrow \mathbb{R}^k$ defines the slice in $\mathcal{S}$; $t \in \mathbb{R}$ is a switching time; and $\mu_j$ and $v_j$ are the $j^{th}$ element ($j \in [1,k] - \{i\} \in \mathbb{N}$) of the control parameter vectors $\mu$ and $v$, respectively, such that $\mu_j$ is held fixed at the value $v_j$.

The algorithm generalizes Algorithm 4 by using $M_0$ and the $k$ maps of Equation (11) to recursively construct a $(k+1)$-dimensional manifold. The algorithm recurses on the dimension $d$ ($0 \leq d \leq k+1$) of the manifold. The base case of $d = 0$ returns a singular EG, which is the seed value for constructing a curve for the case of $d = 1$. The recursive step generates gaits for a $d$-dimensional manifold using the gaits on a $(d-1)$-dimensional submanifold as seed values.

While this algorithm allows for the control space to vary throughout the continuation, it becomes impractical for large $k$. The algorithm is a brute-force approach to searching higher-dimensional manifolds for desired gaits as we have to continue to run the algorithm until it happens to come across a gait we are interested in.

B. Finding Desired Gaits Using the Global Homotopy Map

In this section, we give an algorithm for searching $(k+1)$-dimensional gait manifolds for gaits with desired properties, such as gaits for walking on flat ground. A core part of the algorithm is the use of the global homotopy map (GHM) \cite{38, 48} to find these gaits. The GHM continuously deforms gaits found using our previous map $M_0$ into gaits that satisfy the periodicity constraints of the map $P$, and up to $k$ additional constraints, by varying all parameters in $\mathcal{S}$ at the same time. The GHM $G : \mathcal{S} \rightarrow \mathbb{R}^{n_h}$ is

$$G(c) = H(c) - pH(a), \quad a \in G_{\text{mapped}} - H^{-1}(0),$$

$$p(c) = (H^T(a)H(c))/(H^T(a)H(a)).$$

The map $H : \mathcal{S} \rightarrow \mathbb{R}^{n_h}$ ($n_h \leq k$) is similar to $\Phi$ of Equation 10 with the exception that $H$ can specify fewer than $k$ constraints. The gait $a \in G_{\text{mapped}} - H^{-1}(0)$ serves as a template motion that we attempt to continuously deform into a desired gait—a to-be-determined point $c \in \mathcal{S}$ satisfying $H(c) = 0$. The parameter $p(c) \in [0,1]$ is the homotopy parameter that continuously deforms the reference gait $a$ into a gait on the $H(c) = 0$ slice in $\mathcal{S}$. Two important properties of the homotopy parameter are the following:

1) for $c = a$, we have $p(a) = 1$, which makes $a$ a trivial root of $G$ (i.e., $G(a) = H(a) - p(a)H(a) = H(a) - H(a) = 0$), and,

2) for points $c_0 \in H^{-1}(0)$, we have $p(c_0) = 0$ and $G(c_0) = H(c_0) - p(c_0)H(a) = H(c_0)$ thus making $c_0$ roots of $p$ and $G$ as well.

The GHM is a type of auxiliary function that is meant to be used to query the gait space. As an illustrative example, let $\sigma(c) \in \mathbb{R}$ compute the incline of a planar biped’s walking surface and $\nu(c) \in \mathbb{R}$ compute the biped’s average walking velocity, then the structure of the query is: Given the manifold in $G$ that contains the gait $a$, find a gait $c_0$ that walks on flat ground ($\sigma(c_0) = 0$) at 0.7 m/s. The constraint function $H = [\sigma(c), \nu(c) - 0.7]^T = 0$ is used to encode the query as a set of equality constraints. We now integrate the GHM map into the rest of our framework.

Let $n_G = 2n + n_h$ be the number of total constraints and $k_G = k - n_h$ be the number of expected freedoms (as singularities on a connected component can cause the number of freedoms to increase). Given the GHM, define the map $M_a : G \rightarrow \mathbb{R}^{n_G}$ as

$$M_a(c) = [P^T(c), G^T(c)]^T.$$

Algorithm 6 A modified continuation method for use with the map $M_a$ of Equation (13)

Require: $M_a : \mathbb{R}^{2n+k+1} \rightarrow \mathbb{R}^{n_G}$, $a \in M_a^{-1}(0)$, $\alpha \in \mathbb{R}$, and $\beta \in \mathbb{R}$.
1: function GHM($c$) 
2: Assume: $M_a(c) = 0$ 
3: Define merit function: $f(x) = \frac{1}{2}p(x)^2$. 
4: Newton Step: 
5: Solve for $\frac{\partial f}{\partial s}$ such that $\frac{\partial M_a}{\partial c}(c)\frac{\partial c}{\partial s} = 0$ and 
6: $\frac{\partial f}{\partial s} = I_{k_G}$, where $I_{k_G}$ is a $k_G \times k_G$ identity matrix. 
7: $\Delta s = -\left[\frac{\partial f}{\partial c}(c)\frac{\partial c}{\partial s}\right]^T \frac{\partial f}{\partial c}(c)$ 
8: Perform Line Search: 
9: Set $m = 0$ and $\hat{c} = \frac{\partial f}{\partial c}(c)\Delta s$. 
10: repeat 
11: Set $\lambda = \beta^m$, $m = m + 1$, and $h = \lambda \Delta s$ 
12: $z = \text{CMSTEP}(c, \hat{c}, h)$ (see Algorithm 2) 
13: until $f(z) - f(c) \leq -\alpha \lambda \frac{\partial f}{\partial c}(c, \hat{c}) \Delta s$ 
14: return $z$ 
15: end function 
16: Generate Curve: 
17: Set $c[0] = a$. 
18: for $i := 1,..N$ do 
19: $c[i] = \text{GHM}(c[i-1])$ 
20: end for 
21: return the solution curve $c$
The space $M_{-1}(0)$ is comprised of $k_G$-dimensional manifolds in $\mathcal{G}$. The goal is to find a path from a known gait $a \in \mathcal{G}_{\text{mapped}}$ to a gait $c \in \mathcal{G}$ such that $H(c) = 0$. From Equation (12), this is equivalent to finding a root of $p$. In order to find a root of $p$, we modify Algorithm 2 so that we simultaneously generate $\mathcal{G}_{\text{mapped}}$ and search for a gait in $\mathcal{G}$ that is a root of $p$. The modified algorithm is summarized in Algorithm 6.

**Proposition 4.** Given a point $a \in \mathcal{G}_{\text{mapped}}$, $-H^{-1}(0)$ sufficiently close to a root of $p$ and the map $M_a$, if at every iteration $i$ ($1 \leq i \leq N$) of Algorithm 6 the tangent vector $\frac{\partial c}{\partial s}(s_i) \in T_{c(s_i)}M_{-1}(0)$ is chosen such that

$$\frac{dc}{ds}(s_i) = -\frac{\partial c}{\partial s}(s_i) \left[ \frac{\partial p^T}{\partial c} (c(s_i)) \frac{\partial c}{\partial s}(s_i) \right]^T p(c(s_i)),$$

where $[\cdot]^T$ is the Moore-Penrose inverse and $\frac{\partial c}{\partial s}(s_i) \in \mathbb{R}^{(2n+k+1) \times k_G}$ is a matrix whose $k_G$ columns are basis vectors for $T_{c(s_i)}M_{-1}(0)$ at $c \in M_{-1}(0)$, then tracing the vector field $\frac{dc}{ds}(s_i)$ starting from a simultaneously a one-dimensional curve of fixed points $c(s_i) \in M_{-1}(0) \subseteq \mathcal{G}$ and a sequence of Newton iterates that converges to a root of $p$.

**Proof.** We prove the proposition in two steps. First, we derive the direction of a Newton step $\Delta s \in \mathbb{R}^{k_G}$ for numerically solving $p(c(s)) = 0$, where $s \in \mathbb{R}^{k_G}$ is some parameterization of the manifold $M_a(c(s)) = 0$. We then project $\Delta s$ onto the tangent space $T_{c(s_i)}M_{-1}(0)$ using $\frac{\partial c}{\partial s}(s_i)$.

As the system of equations $p(c(s)) = 0$ is not square, we use the Moore-Penrose inverse to define a Newton step $\Delta s$ from a Taylor approximation of $p$ about a root; that is from $p(c(s) + \Delta s) \approx p(c(s)) + \frac{\partial p^T}{\partial c} (c(s)) \frac{\partial c}{\partial s}(s) \Delta s = 0$, we get $\Delta s = -\left[ \frac{\partial p^T}{\partial c} (c(s)) \frac{\partial c}{\partial s}(s) \right]^T p(c(s))$.

At iteration $i$ of Algorithm 6, we project the Newton step onto the tangent space $T_{c(s_i)}M_{-1}(0)$ at $c(s_i) \in M_{-1}(0)$, which yields the tangent vector

$$\frac{dc}{ds}(s_i) = -\frac{\partial c}{\partial s}(s_i) \left[ \frac{\partial p^T}{\partial c} (c(s_i)) \frac{\partial c}{\partial s}(s_i) \right]^T p(c(s_i)) = -\Delta s.$$

This choice leads to a curve being traced in $M_{-1}(0)$ with points on the curve that converge to a root of $p$. □

Algorithm 6 contains the function GHM for computing a new point $z \in M_{-1}(0)$ from $c \in M_{-1}(0)$ based on Proposition 2. In order to ensure that we are making progress towards a root, we take a step of magnitude $h$ in the direction of $\frac{\partial c}{\partial s}(s) = \frac{\partial c}{\partial s}(s)\Delta s$ based on an Armijo line search using the merit function $f(c(s)) = \frac{1}{2}p(c(s))^2$. This leads to an adaptive step size strategy for $h$. Typical values used for $\alpha$ and $\beta$ in Algorithm 6 are $10^{-4}$ and 0.5, respectively.

**Remark 7.** For biped models with sufficient control authority, we can seed the gait $a$ of Algorithm 6 with an EG that is a regular point of $P$. A biped has sufficient control authority if any regular EG has a local neighborhood of gaits in $\mathcal{G} - E$ on the $(k + 1)$-dimensional manifold it is on. If not, we would have to search for singular EGs as outlined in Section III-B.

**Remark 8.** In Section IV, we discussed using the control parameters to define a family of 2D and 3D biped models. An important application of the GHM is taking the parameterized model and, for example, continuously deforming a gait for a planar version of the model (where the EGs are trivial to specify) into gaits for a 3D version of the biped model.

**V. Examples**

We have applied our framework to several bipeds taken from the literature (Figure 6) to confirm its wide applicability. The biped models range from planar passive dynamic walkers to high-degree-of-freedom actuated 3D humanoids. The details of these bipeds can be found in the Multimedia Material. In this section, we expand on using equilibria to generate actuated gait for the compass gait [2], MARLO [29] (Figure 6(f)), and Atlas [54] (Figure 6(g)) bipeds.

All bipeds are modeled as kinematic trees with floating bases attached to their pelvis [55]. If the biped is planar the floating base has three configuration variables, otherwise the floating base has six configuration variables. Unless otherwise noted,

1. physical quantities are measured in SI units (i.e., meters, kilograms, seconds),
2. the gaits reported in this paper have a normed error between two consecutive pre-impact states $x_0 = (q, \dot{q})$ of less than $10^{-8}$ (with $q$ having units of radians and $\dot{q}$ radians per second),
3. the search window for singular equilibrium gait for Algorithm 1 is $\tau \in [0.1, 1]$ divided into 100 steps,
4. the step size $h$ of Algorithm 3 is $\pm \frac{1}{2}$ in order to trace both sides of a curve. We attempt to generate $N = 250$ gait per function call,
5. no constraints are placed on the ground reaction force,
6. joints that do not appear as free variables in the periodicity map $P$ have been solved for explicitly because they appear linearly in a VHC or PHC:
   * for joints subject to VHCs, the feedback control laws are written so that the joint trajectories are periodic and a function of the free parameters in $S$;
   * for joints subject to PHCs, especially the Cartesian floating base joints, the values of these joints can be determined from the other parameters in the PHCs,
7. boundary Bézier polynomial coefficients in a VHC are written in terms of $x_0$ and satisfy periodicity constraints,
8. the $x$, $y$, and $z$ axes of the world and local frames of our 3D biped models are labeled with blue, red, and green arrows, respectively, and
9. gaits are computed on a 2.7 GHz Intel Core i7-4800MQ CPU laptop running 64-bit Ubuntu 18.04 LTS.

The Multimedia Material contains an implementation of our framework as a Mathematica library. Given model-specific
information, like the biped’s PHCs and VHCs, the code is capable of finding entire families of walking gaits using only equilibria of the biped models. The code computes items like the map $P$, the flow $\phi$, and their derivatives from the model.

A. Extending Passive Gaits into a Family of Actuated Gaits

If we add a motor at the hip joint of the compass-gait robot as described in Section III-E, the state-time space $S$ can be augmented with control dimensions. In general, increasing the dimension of $S$ will also increase the dimension of $G$. The original $G$ is just the $\mu = 0$ slice of the extended $G$.

We use the motor to drive leg 2 relative to leg 1 with the torque $u_0(t) = \mu_0 \sin(\omega t)$ [56], where the amplitude $\mu_0 \in \mathbb{R}$ is a control parameter. The angular frequency is fixed at $\omega = 2\pi$ to keep this example low dimensional. The control dimension is one ($k = 1$) and we have added an actuator ($n_u = 1$) to the model. Design parameters could also be defined. For example, a parameter defining the curvature of the feet [50] or position of the center of mass [20] could be added. In this example, however, we only add the control parameter $\mu_0$. Figure 7(a) shows an actuated gait in the extended gait space. The value of $\mu_0$ for this step is $-5.34$ N m/(kg m$^2$) and becomes $5.34$ N m/(kg m$^2$) at the start of the next step.

As masses and lengths are scaled in our reference model [2], this corresponds to a maximum output torque of about $\mu_0 = 1.34$ N m.

In this six-dimensional state-time-control space $S$ consisting of points $(x_0, \tau, \mu_0)$, we have two-dimensional gait manifolds (the six parameters minus the four periodicity constraints). The passive gait of the previous section are now a slice of this higher-dimensional gait space $G \subset \mathbb{R}^6$, where the control parameter $\mu_0$ is zero (Figure 7(b)).

Algorithm 5 is used to construct the surface of Figure 7(b). The algorithm first uses the map $M_0$ of Section III-B to generate a set of passive gaits using the seed values $(x_{eq}, 0, 0, 0)$ (i.e., the singular points of the original $G$ space mapped to the extended $G$ space). Our Mathematica implementation of the algorithm took 7.2 minutes to compute the 828 passive gaits in Figure 7(b). Algorithm 5 then uses every gait found in $M_0^{-1}(0)$ as seed values to Algorithm 2 using the map $M_1 : S \rightarrow \mathbb{R}^{2n+k}$ of Equation (11), which holds $\tau$ constant and allows $\mu_0$ to vary during the continuation. The library took 1.8 hours to compute the 18,400 actuated gaits in Figure 7(b).

This higher-dimensional example shows that we can grow $G$ mapped from an EG to a set of passive gaits to an even larger
set of actuated gaits by adding extra control parameters to the state-time-control space $S$. We use this notion of growing a manifold from lower-dimensional slices for our 3D bipeds as well. In this case, we generate gaits for a planar or simplified 3D version of the biped and use these walking gaits to generate gaits for the full 3D model.

### B. Generating Gaits for a Flat-footed Walking Biped with Arms

We have tested our technique on Boston Dynamics’ Atlas, a 3D flat-footed walking biped (Figure 8(a)). The use of flat-footed walking constraints is inspired by the reduced models of [11], [12], which only consider the legs of Atlas as their 3D flat-footed walking biped (Figure 8(a)). The use of flat-footed walking constraints is inspired by the reduced models of [11], [12], which only consider the legs of Atlas as their 3D flat-footed walking biped (Figure 8(a)).

We have tested our technique on Boston Dynamics’ Atlas, a 3D flat-footed walking biped (Figure 8(a)). The use of flat-footed walking constraints is inspired by the reduced models of [11], [12], which only consider the legs of Atlas as their 3D flat-footed walking biped (Figure 8(a)).

The model we use for Atlas is from a DARPA Robotics Challenge “.cfg” file available online [57]. This source file generates the biped’s URDF file for use in ROS and Gazebo. We use the full model with 3D dynamics and a 6D floating base at the pelvis. The only changes to the model are:

1) zeroing the $y$ position of the center of mass of two links (see Multimedia Material) to make the model physically symmetric (Assumption A1), and

2) defining the biped’s home position with the arms pointing downwards (Figure 8(a)), so that $x_{eq} = 0 \in \mathcal{X}$ can be an equilibrium point.

In the end, our model of Atlas has 34 degrees of freedom when unconstrained (Figure 8(a)). The model has 15 VHCs and 15 actuators. However, the VHCs and actuators are not all active during a step. When a VHC is not active, an actuator is considered off and the joint attached to the inactive actuator is unactuated during the step.

During a step, the biped walks in its sagittal plane and is subject to six PHCs that fix the stance foot to the ground ($n_p = 6$), 13 active VHCs ($n_v = 13$), and 13 active actuators to enforce the VHCs ($n_a = 13$). Atlas’ remaining 15 internal joints that are not subject to VHCs as well as the 6 joints of the floating base are unactuated.

The set of active VHCs depends on whether the left or right leg is the stance leg. At the post-impact time $t = 0^+$, we assume the left leg is the stance leg. The configurations and velocities of the joints about the $x$ and $z$ axes at the hips, $y$ axis at the neck, $x$ and $y$ axes at the ankles, $y$ axis at the right elbow, and $x$ axis at the wrists are subject to VHCs that force the joints to track third-order Bézier polynomials.

The configuration of the left knee joint is subject to the final VHC that keeps the relative knee angle locked at zero degrees throughout the motion post-impact. Further details, like the active set of VHCs when the right leg is the stance leg, can be found in the Multimedia Material.

From this data, the biped’s pre-impact state is $x_0 \in \mathcal{X} \subset \mathbb{R}^{68}$ ($n = 34$). In our code, we use the PHCs and VHCs to reduce the number of independent states in $x_0$ down to 33 state variables and define $x_0$ as a function of a reduced state vector $\bar{x}_0 \in \mathbb{R}^{33}$ such that $x_0 = x_0(\bar{x}_0)$.

In addition to the states, we define two control parameters $\omega_1, \omega_2 \in [0, 1]$, making $\mu = [\omega_1, \omega_2]^T \in \mathcal{M}$. These dimensionless control parameters modify the biped’s physical parameters to create a 3D model that can balance on one foot at $x_0 = 0$. The parameter $\omega_1$ affects every link’s center of mass position and relative distance from the pelvis along the link’s $x$ axis (i.e., if the distance of link $i$ from the pelvis along the $x$ axis was $\delta x_i$, then the model would contain the product $\omega_1 \delta x_i$). The parameter $\omega_2$ affects the center of mass position along the $y$ axis of every link’s center of mass. When both parameters are zero, they, along with the active
VHCs, eliminate the non-zero moments about the joints due to the gravity vector making the configuration depicted in Figure 8(a) an equilibrium for the model. A value of one for each parameter gives the original biped model.

In total, the biped has a 71-dimensional state-time-control space $S$ (68 states, 2 control parameters, and 1 switching time). We define $M_0$ of Equation (6) so that $\omega_1 = \omega_2 = 0$ throughout the continuation making $\mu_0 = 0$ in $M_0$. At these fixed values of the control parameters, we generate gaits for Atlas using the same process as we did with the compass gait.

For the parameterized model at $\omega_1 = \omega_2 = 0$, a singular EG found in the constant-control slice occurs at $\tau = 0.396$ s. Algorithm 4 performs a NCM using the map $M_0$ starting at the singular EG at $\tau = 0.396$ s. Our Mathematica code took 1.5 hours to compute 250 gaits. We then apply Algorithm 6 to find a gait with $\omega_1 = \omega_2 = 1$. The library took 1.8 hours to find a gait with the desired values and computed 48 gaits in the process. The desired gait from this continuation is shown in Figure 8(g). The biped is shown taking two steps.

C. Incorporating Inequality Constraints

Our final example gives an in-depth overview of using the GHM and Algorithm 6. In this example, we use the University of Michigan’s MARLO [26, 29] (Figure 8(b)) to demonstrate how to incorporate inequality constraints into a continuation. The biped is part of a line of ATRIAS bipeds developed at Oregon State University [58]. The hybrid dynamics of the model is detailed in [29]. We do not model the series-elastic actuators, but do take into account the mass of the actuators. The physical parameters for our model are taken from the source code in [59], which is used in [26].

MARLO has 16 degrees-of-freedom (DOFs) when no constraints are applied. The biped walks in its sagittal plane with gravity pointing downward. Referring to Figure 8(b), the joints of the biped are a 6-DOF floating base (where $q_1$–$q_3$ are the roll, pitch, and yaw angles, respectively), two hip joints for out-of-plane leg rotation $(q_4$–$q_6$), and eight joints for the two four-bar mechanisms serving as legs for the biped $(q_7$–$q_{13})$. When constraints are applied, MARLO has seven PHCs ($n_p = 7$). Three of the constraints keep the stance foot (a point) stationary and the other four constraints are the four-bar linkage constraints on each leg:

$$q_6 + q_{10} - q_7 = 0, \quad q_7 + q_{11} - q_6 = 0, \quad q_8 + q_{12} - q_9 = 0, \quad q_9 + q_{14} - q_8 = 0.$$  

Given these constraints, we can describe a pre-impact state $x_0 \in X$ of the biped using 18 numbers, the nine joint angles $q_1$–$q_9$ and their respective velocities ($n_q = 9$).

The biped has six actuators that drive the robot’s leg joints $q_4$–$q_9$ ($n_u = 6$) and is subject to six virtual holonomic constraints ($n_v = 6$). When the left leg is in stance, the constraints are

$$q_4 - b_1^4(\theta, a) = 0, \quad q_5 - b_1^5(\theta, a) = 0, \quad q_6 - b_1^6(\theta, a) = 0, \quad q_{10} - b_{10}^4(\theta, a) = 0, \quad q_{11} - b_{10}^5(\theta, a) = 0, \quad q_{12} - b_{10}^6(\theta, a) = 0,$$

and similarly when the right leg is in stance.

During a step, the VHCs force the hip, stance thigh, and lower leg to track third-order Bézier polynomials and the swing thigh and lower leg to track fourth-order Bézier polynomials. The two fourth-order polynomials $b_2^4(\theta, a)$ and $b_2^4(\theta, a)$ each have a free coefficient that is not determined by the periodicity boundary constraints. The two free coefficients, say $\alpha_1, \alpha_2 \in \mathbb{R}$, correspond to control dimensions in $S$. The continuation will determine values for these coefficients for each gait generated.

Overall, there are six control parameters $\mu = [\omega, s_1, s_2, s_3, \alpha_1, \alpha_2]^T \in \mathcal{M} \subseteq \mathbb{R}^6$ (k = 6), explained below. The dimensionless parameter $\omega \in \mathbb{R}$ continuously deforms the physical parameters of the biped from a planar model into a 3D model by controlling the hip width and the position of the center of mass of each link on the biped (in other words, if the hip width is defined by the physical parameter $\ell_{\text{hip}}$, then the model would have its hip width defined as the product $\omega \ell_{\text{hip}}$). Figure 9 depicts how $\omega$ affects the biped’s hip width; it does not show the effects on the center of mass of each link. At $\omega = 0$, the biped has zero hip width and all of the center of masses are projected onto their respective links. When $\omega = 1$, the original values for the biped’s hip width and link center of masses are restored.

Finally, the vector $\mu$ has three slack variables $s_1, s_2, s_3 \in \mathbb{R}$ with lower bounds $s_1, s_2, s_3 \geq 0$; there are no upper bounds on the variables. These constraints are treated as box constraints [44]. We use the slack variables $s_1$ and $s_2$ so that we only search for gaits where the knee joints $q_{10}$ and $q_{12}$ are nonnegative pre-impact. This is sufficient to avoid knee hyperextension as the biped takes its step ($q_{10}(t), q_{12}(t) \leq 0$ for all $t$). The third slack variable $s_3$ is used to make the biped walk from left to right by imposing a forward velocity constraint on the biped’s floating base $q_b \in \mathbb{R}$ coordinate.

We also impose integral inequality constraints. Let $p(t) = p(x(t)) \in \mathbb{R}^3$ represent the location of the swing foot in space, and $\text{dist}(p(t)) \in \mathbb{R}$ be a distance function that is positive when the swing foot is above the surface, zero when the swing foot is on the surface, and negative when the swing foot is below the surface. The goal is for the swing foot to never go below the walking surface. To avoid foot penetration with the ground, we require $\text{dist}(p(t)) \geq 0$ for all $t \in \mathbb{R}$. This is equivalent to finding the zeros of $\int_0^T \text{dist}(p(t)) \, dt$, where $[x^-]$ returns $x$ if $x \leq 0$ and zero otherwise.

Given the model and its PHCs and VHCs, the resulting state-time-control space $S$ is 25-dimensional (18 state variables, six design and control parameters, and one switching time). The biped’s periodicity map $P$ is

$$P(c) = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} q_1(\tau^-) - q_1(0^-) \\ q_2(\tau^-) - q_2(0^-) \\ q_3(\tau^-) - q_3(0^-) \end{bmatrix}, \quad Q_2 = \begin{bmatrix} q_1(\tau^-) - q_1(0^-) \\ q_2(\tau^-) - q_2(0^-) \\ q_3(\tau^-) - q_3(0^-) \end{bmatrix},$$

which states that the roll, pitch, and yaw angles of the floating base have to be periodic. These are the only angles that are unactuated; all other angles are subject to VHCs which we can design to satisfy the periodicity condition of the (virtually) constrained joints [4]. We note that while we do not write periodicity constraints in $P$ for joint angles $q_1$–$q_9$ and their velocities, they are still free parameters in $S$ that affect the motion of the base and the four-bar linkage. The numerical continuation determines their values for each gait generated.
The manifolds in the gait space $\mathcal{G}$ are 19-dimensional (25 state-time-control variables minus six periodicity constraints). We cannot apply Algorithm 1 because $\frac{\partial P}{\partial \omega}(c) \in \mathbb{R}^{6 \times 18}$ is not a square matrix. For MARLO, we demonstrate the utility of searching for locomoting gaits on a high-dimensional manifold using a GHM. As an additional benefit of the GHM, we can search for locomoting gaits on a high-dimensional manifold and Algorithm 6. The Mathematica code found a desired gait after 42 minutes and computed 58 gaits.

VI. COMPARISON TO FROST WITH IPOPT

We have demonstrated our approach using the examples in Section VI and the bipeds in Figure 6. In this section, we informally compare the numerical continuation approach to FROST [60] using IPOPT. FROST is a Matlab/Mathematica-based library for modeling and simulating hybrid mechanical systems and transcribing them to nonlinear programming (NLP) problems. The FROST library ships with IPOPT as the underlying solver for the NLP. IPOPT uses interior point methods (a form of numerical continuation [44]) to minimize a cost function subject to box constraints. We view FROST as a library interface that simplifies the process of generating an NLP for bipedal robots (similar to our library) and IPOPT as a suite of algorithms that performs the search for an optimal gait (like our algorithms).

Our library has several features in common with FROST. For example, they both use Mathematica (MMA) to model and simulate rigid-body mechanical systems subject to impacts, PHCs, and VHCs; compute derivatives; and compile MMA expressions into C code.

In terms of distinct features, FROST can output its C code to be used in other environments such as Matlab. A unique feature of our library is the ability to differentiate and compile user-defined algorithms written in a subset of MMA into C executables, including the Recursive Newton-Euler and Composite Rigid Body algorithms [55]. This approach scales better than FROST’s approach, which symbolically computes the Euler-Lagrange equations [39]. As an example, on our test computer, FROST crashes after more than an hour of computing the mass matrix of the Atlas model in Section V-B while our approach can compile the hybrid equations of motion and their linearized dynamics to C in under 10 minutes. However, a limitation of our current implementation is MMA’s memory management of its C code, which precludes compiling higher-order derivatives.

Fig. 9. The effect of the parameter $\omega$ on the hip width of MARLO. The parameter also affects the position of the center of mass of each link (not shown).
A. Generating Gaits for RABBIT

For comparison, we ported the FROST model of RABBIT (see Figure 1 and the corresponding NLP constraints into our framework. Briefly, the NLP locally minimizes the time-integral of summed and squared input torques subject to a desired average forward velocity and constraints on friction at the stance foot and the minimum step height at the middle of the step. See the Multimedia Material for details.

In order to represent the NLP, FROST uses direct collocation and generates 1270 variables and 1463 constraints to solve the problem at 21 nodes, where the time steps of the integration occur over the first 20 nodes and the last node is where the collision occurs. The total time for NLP compilation to C and solving using the GHM algorithm (Algorithm 6) is approximately 213 seconds. The focus of this paper is to show that entire families of biped using FROST.

We present a robust method for generating large families of walking gaits for high-degree-of-freedom bipeds using PHCs and VHCs (n_p, n_v ≥ 0). For a biped’s continuous dynamic regime, we assume that the state of the biped x = (q, ˙q) ∈ X is known and that the biped is subject to a set of physical and virtual holonomic constraints h_p(q) = 0 ∈ R^n_p and h_v(q) = 0 ∈ R^n_v, respectively. The state x, accelerations ¨q ∈ R^n, constraint forces λ ∈ R^n_p, and control inputs u ∈ R^n_u (n_u ≥ n_v) of the continuous dynamics satisfy

\[
\begin{aligned}
M(q) ˙q + b(q, ˙q) &= J_p^T(λ) + B_v(q)u, \\
J_p(q) ˙q + J_v(q) ˙q &= -v(q, ˙q),
\end{aligned}
\]

where M(q) ∈ R^{n × n} is the mass matrix, b(q, ˙q) ∈ R^n is a vector of the centrifugal, Coriolis, and gravitational forces, B_v(q) ∈ R^n_p × n_u is a transmission matrix, J_p(q) = ∂h_p(q) / ∂q ∈ R^n_p × n and J_v(q) = ∂h_v(q) / ∂q ∈ R^n_v × n are the constraint Jacobian for the physical and virtual constraints, respectively, and v(q, ˙q) ∈ R^n_v is a linear feedback controller for stabilizing the virtual constraints. An example PD control law for v(q, ˙q) used in (4), (11) is

\[
v(q, ˙q) = \frac{1}{\epsilon} K_D J_v(q) ˙q + \frac{1}{\epsilon^2} K_P h_v(q),
\]

where h_v(q) ∈ R^n_v are VHCs, K_P, K_D ∈ R^n_v × n_v are positive-definite matrices and ϵ ∈ R is a positive scalar tuning parameter which can speed up convergence to the origin, h_v(q) = J_v(q) ˙q = 0.

Given Equation (14), the vector field f is then f(x, u) = ( ˙q, ˙q), where ˙q is the solution to

\[
\begin{bmatrix}
M(q) & -J_p^T(λ) & -B(q) \\
J_p(q) & 0 & 0 \\
J_v(q) & 0 & 0
\end{bmatrix}
\begin{bmatrix}
q \\
λ \\
u
\end{bmatrix} = \begin{bmatrix}
b(q, ˙q) \\
0 \\
v(q, ˙q) + J_v(q) ˙q
\end{bmatrix},
\]

which is a linear system of equations with n + n_p + n_v equations in n + n_p + n_v unknowns. A solution exists with a generalized right inverse (we use the Moore-Penrose inverse) as long as the matrix on the left-hand side has maximal rank n + n_p + n_v.

Remark 9. The differential form of the virtual constraints J_v(q) ˙q + J_v(q) ˙q = -v(q, ˙q) can model virtual holonomic and nonholonomic constraints. The same is true of the differential form of the physical constraints.

Remark 10. A linear stabilizing feedback controller v_p(q, ˙q) can also be defined for the n_p physical constraints such that J_p(q) ˙q + J_p(q) ˙q = -v_p(q, ˙q) in order to reduce constraint
violations during simulations due to numerical rounding
errors. For example, we can implement Baumgarte’s constraint
stabilization technique \cite{62} using the form of the feedback law
for \(v(q, \dot{q})\). Other options also exist \cite{64}.

Remark 11. The solution to \(u(t)\) of Equation \(14\) is equivalent
to the solution of a stabilizing feedback linearizing control
law used to enforce VHCs in the Hybrid Zero Dynamics
framework \cite{4} with \(v(q, \dot{q})\) as the linear control law.

B. The Jump Map \(\Delta\)

We model collisions as a set of impulsive algebraic equa-
tions, namely the impulse-momentum equations in generalized
coordinates along with a set of plastic impact equations needed
to uniquely solve for the impulse and post-impact velocities.
During a collision at time \(t \in \mathbb{R}\), we assume the pre-impact
state of the biped \(x(t^-) = (q, \dot{q}) \in \mathcal{X}\) is known. The pre-
impact state \(x(t^-)\), post-impact state \(x(t^+) = (q^+, \dot{q}^+) \in \mathcal{X}\),
and impulses \(\tau \in \mathbb{R}^{n_t}\) of the impulse equations satisfy
\[
q^+ = q, \quad M(q)(\dot{q}^- - \dot{q}) = J^T q \in \mathcal{X}, \quad J(q)\dot{q}^- = 0,
\]
where \(J_c \in \mathbb{R}^{n_c \times n_q}\) is the constraint Jacobian that maps the post-
impact velocities to contact velocities.

The jump map \(\Delta\) is then \(\Delta(x) = (q, \dot{q}^+)\), where \(\dot{q}^+\) is the solution to
\[
\begin{bmatrix}
M(q) & -J^T(q) \\
J_c(q) & 0
\end{bmatrix}
\begin{bmatrix}
\dot{q}^+ \\
\dot{q}_c
\end{bmatrix}
= \begin{bmatrix}
M(q)\dot{q} \\
0
\end{bmatrix},
\]
which is a linear system of equations with \(n + n_c\) equations
in \(n + n_q\) unknowns. A unique solution exists as long as \(J_c\)
has maximal rank \(n_c\).

For most bipeds, \(J_c = J_p\). In general, any \(J_c\) is fine as long
as the set of jump map constraints are more restrictive than
the PHC constraints, i.e., \(\{x \in \mathcal{X} : J_c(q)\dot{q} + J_p(q)\dot{q} = 0\}\).

APPENDIX B
PROOFS OF THE EQUILIBRIUM BRANCHES OF \(M_0\)

Proof of Proposition 7 It follows from the implicit function
theorem (IFT, \cite{37}) that there exists a unique curve \(c\) passing
through \(c_0\). What remains to be shown is that the points on
the curve are all in \(E_0\). From the IFT, we conclude that in
an open neighborhood \(A \subseteq \mathbb{R}\) containing \(c_0\) and an open
neighborhood \(B \subseteq \mathbb{R}^{2n+k}\) containing the pair \((x_{eq}, \mu_0)\) that
for each \(\tau(s) \in A\) there exists a unique \(g(\tau(s)) \in B\) such that
\(g(\tau) = (x_0(\tau, \mu_0(\tau)), c(s) = (x_0(\tau(s)), \tau(s), \mu_0(\tau(s)))\) and
\(M_0(c(s)) = 0\) for some parameterization of \(s \in (-\delta, \delta)\).

We determine \(g\) from the Jacobian \(J_0\) of \(M_0\)
\[
J_0(c) = \begin{bmatrix}
\frac{\partial \tau}{\partial \tau} (c) \\
\frac{\partial \mu}{\partial \tau} (c)
\end{bmatrix} = \begin{bmatrix}
\frac{\partial \tau}{\partial x_{eq}} (c) & \frac{\partial \tau}{\partial u} (c) \\
0 & 0 & \frac{\partial P}{\partial \mu} (c)
\end{bmatrix}.
\]
(15)

Given that \(x_{eq}\) is an equilibrium point, we have at \(J_0(c_0)\)
\[
\frac{\partial P}{\partial \tau} (p(s_0)) = f(x_{eq}, u(\tau(s_0))) \frac{\partial \tau}{\partial \mu} (p(s_0)) = 0.
\]
Additionally, because \(c_0\) is a regular point of \(M_0\), the sub-matrix
\[
\bar{J} = \begin{bmatrix}
\frac{\partial P}{\partial x_{eq}} (c_0) & \frac{\partial P}{\partial u} (c_0) \\
0 & \frac{\partial P}{\partial \mu} (c_0)
\end{bmatrix}_k
\]
must be full rank, \(\det(J) \neq 0\), or else \(J_0\) cannot have maximal
rank and \(c_0\) would be a singular point, which it is not. From
these two facts, the IFT states that \(g\) must be the solution to the
IVP
\[
g(\tau_0) = (x_{eq}, \mu_0),
\]
\[
\frac{\partial g}{\partial \tau} (\tau_0) = -\begin{bmatrix}
\frac{\partial P}{\partial x_{eq}} (p(s_0)) & \frac{\partial P}{\partial u} (p(s_0)) \\
0 & 0
\end{bmatrix}^{-1} \begin{bmatrix}
\frac{\partial P}{\partial x_{eq}} (p(s_0)) \\
0
\end{bmatrix} = 0,
\]
which gives \(g(\tau(s)) = (x_{eq}, \mu_0)\) for all \(\tau(s) \in \mathbb{R}\).

To finish describing the curve \(c\), we now determine an
expression for \(\tau = \tau(s)\) valid for \(s \in (-\delta, \delta)\). Given
an arclength parameterization of the curve \(c\), the tangent to the
curve \(c\) at \(c_0\) is \(\frac{\partial P}{\partial \tau} (0)\) and it is the only vector in the tangent
space \(T_{c_0}M_0\). As the tangent space is equal to the null space
of \(J_0(c_0)\), we have
\[
\text{Null}(J_0(c_0)) = \text{Null} \left( \begin{bmatrix}
\frac{\partial P}{\partial x_{eq}} (c) & 0 \\
0 & 0
\end{bmatrix} \right) = \{0_{2n}, 1, 0_k\}_k^T,
\]
where \(0_k\) and \(0_{2n}\) are vectors with \(k\) and \(2n\) zeros, respectively.
Hence, \(\frac{\partial P}{\partial \tau} (s) = 1\) and for \(\tau(0) = \tau_0\), \(\tau(s) = \tau_0 + s\).
Therefore, \(c(s) = (x_{eq}, \tau_0 + s, \mu_0)\) for \(s \in (-\delta, \delta)\), which
are all in \(E_0\) and by the IFT are the only points in a neighborhood
containing \(c_0\) that are in \(G_0\).

Proof of Proposition 2 We prove the first claim by showing
that the path \(p\) can never have points in \(G_0 - E_0\) if EGs on
the path are regular points of \(M_0\). The second claim is proven
through a direct computation.

Assume there exists a path \(p\) with \(p(0) \in E_0\) and \(p(1) \in G_0 - E_0\) such that all EGs in \(p\) are regular points of \(M_0\). Then
by Proposition 1 because \(c_0 \in E_0\) is a regular point of \(M_0\), any
path \(p : [0, 1] \to G_0\) starting at \(c_0\) must coincide locally with the
unique curve \(c : (-\delta, \delta) \to E_0\). As the functions \(p\) and \(c\)
coincide but have different domains, assume that \(s \in [0, s_\delta]\)
continuously maps to \(\alpha(s) \in (-\delta, \delta)\) such that for \(s = 0\) we
have \(\alpha(0) = 0\).

Consider \(p\) at \(s = s_\delta\). Because\(p\) is continuous \(p(s_\delta)\) must
be equal to the left-sided limit of \(p \leq s_\delta\), that is, \(p(s_\delta) = \lim_{s \to s_\delta} (x_{eq}, \tau_0 + \alpha(s), \mu_0) = (x_{eq}, \tau_0 + \alpha(s_\delta), \mu_0)\). Therefore,
\(p(s_\delta)\) is an EG in \(E_0\), but by assumption it is also a regular
point of \(M_0\), so we can apply Proposition 1 and conclude
that the interval over which \(p\) and \(c\) have to coincide extends
beyond \(s \in (0, s_\delta)\) for the path \(p\).

In other words, the unique curve \(c\) passing through \(c(0)\)
can be extended past the open interval \((-\delta, \delta)\). In fact, there
is no finite value of \(\delta\) that can contain the entire interval of \(c\) as its limit points \(c(\delta) = p(s_\delta) \in E_0\) are regular points
of \(M_0\). The curve can always be extended and as the curve \(c\) is
unique, the path \(p\) has no choice but to follow it. The path \(p\)
however is finite with its points determined by \(c\) over its finite
interval, but, for \(s \in [0, 1], p(s) \in E_0\). This contradicts our
main assumption that \(p(1) \in G_0 - E_0\) given that all EGs are
regular points of \(M_0\).

Therefore, there must exist at least one singular EG on the
path \(p\). In order for a point \(p(s) \in E_0\) in the path \(p\) to be
singular, we need the submatrix $J$ of the Jacobian $J_0$ evaluated at $p(s)$

$$J(p(s)) = \begin{bmatrix} \frac{\partial P}{\partial x} (p(s)) & \frac{\partial P}{\partial \mu} (p(s)) \\ 0 & 1_k \end{bmatrix}$$

to not be invertible so that the IFT (and Proposition 1) does not apply. For any $p(s)$ that is a singular EG of $M_0$, this can only happen if

$$\det \left( \begin{bmatrix} \frac{\partial P}{\partial x} (c_s) & \frac{\partial P}{\partial \mu} (c_s) \\ 0 & 1_k \end{bmatrix} \right) = \det \left( \begin{bmatrix} \frac{\partial P}{\partial x_0} (c_s) \\ 0 & 1_k \end{bmatrix} \right) = 0.$$

Proof of Proposition 3. Given that $c_0$ is an isolated singular point in $E_0$, we must have that there are exactly $\dim(T_{c_0}G_0) = 2$ curves that intersect transversally at $c_0$. Otherwise, we would conclude that $c_0$ is not isolated or $\dim(T_{c_0}G_0) \neq 2$, which would be a contradiction. Hence, it suffices to study the tangent space $T_{c_0}G_0$ to locally determine the gaits on each curve near $c_0$.

From Equation (4), a basis for the tangent space $T_{c_0}G_0$ is $T_{c_0}G_0 = \text{Null}(J_0(c_0))$, where $J_0$ is the Jacobian of $M_0$ (Equation (8)). Because $c_0$ is a singular EG, we have from Corollary 1 and Proposition 2 that $\partial P / \partial x = 0$ and $\det(\partial P / \partial x_0) = 0$. This implies that the null space of $J_0(c_0)$ has at least two linearly independent tangent vectors and, by assumption of the above proposition, it has exactly two. As $\partial P / \partial x_0 = 0$, the Jacobian $J_0$ evaluated at $c_0$ has the form

$$J_0(c_0) = \begin{bmatrix} \frac{\partial P}{\partial x_0} (c_0) & \frac{\partial P}{\partial \mu} (c_0) \\ 0 & 1_k \end{bmatrix}.$$ 

Assuming coordinates $(x_0, \tau, \mu)$, for a basis for the null space is $\text{Null}(J_0(c_0)) = \{e_0, 0g_0\}$, where $e_0 = [0, 1, 0]^T \in \mathbb{R}^{2n+1}$, $g_0$ satisfies $1_k^T g_0 = 0$, and $0g_0$ is a row vector of zeros. The tangent vector $e_0$ is in $TE_0$. A curve $c_1$ with tangent vector $\frac{dc_1}{ds}(0) = e_0$ at $c_1(0) = c_0$ can only have points of the form $c_1(s) = (x_{eq}, \tau_0 + s, \mu_0)$ for $s \in [-\epsilon, \epsilon]$. The tangent vector $0g_0$ is orthogonal to $e_0$, hence a curve $c_2$ with $c_2(0) = c_0$ and $\frac{dc_2}{ds}(0) = g_0$ has gaits $c_2(s)$ in $G_0 - E_0$ for small values of $s \neq 0$.

APPENDIX C

DEBUGGING ERROR MESSAGES FROM ALGORITHM

We present a list of potential sources of errors and solutions in the Multimedia Material when a call to Algorithm results in an error state. The issues covered are when

1) the plot of $I(\tau)$ is the constant zero line $I(\tau) = 0$, and
2) the plot has no zero crossings $I(\tau) \neq 0$.

REFERENCES

[1] R. D. Gregg, Y. Y. Dhaher, A. Degani, and K. M. Lynch, “On the mechanics of functional asymmetry in bipedal walking,” IEEE Transactions on Biomedical Engineering, vol. 59, no. 5, pp. 1310–1318, May 2012.

[2] A. Goswami, B. Thuilot, and B. Espiau, “A study of the passive gait of a compass-like biped robot: Symmetry and chaos,” Int. J. Robotics Research, vol. 17, no. 12, pp. 1282–1301, 1998.

[3] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman, “The simplest walking model: Stability, complexity, and scaling,” ASME J. Biomechanical Engineering, vol. 120, no. 2, pp. 281–288, 1998.

[4] 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.

[5] M. D. Crossley, Essential Topology. Springer-Verlag, 2005.

[6] A. Goswami, “Postural stability of biped robots and the foot-rotation indicator (FRI) point,” Int. J. Robotics Research, vol. 18, no. 6, pp. 533–539, 1999.

[7] G. Bessonnet, P. Seguin, and P. Sardain, “A parametric optimization approach to walking pattern synthesis,” The International Journal of Robotics Research, vol. 24, no. 7, pp. 523–536, Jul 2005. [Online]. Available: [http://dx.doi.org/10.1177/0278364905055377]

[8] J. W. Grizzle, C. Chevallereau, R. W. Sinner, and A. D. Ames, “Models, feedback control, and open problems of 3D bipedal robotic walking,” Automatica, vol. 50, no. 8, pp. 1955–1998, Aug 2014.

[9] W. Xi, Y. Yesilevskiy, and C. D. Remy, “Selecting gaits for economical locomotion of legged robots,” The International Journal of Robotics Research, Nov 2015.

[10] V. E. H. Chen, “Passive dynamic walking with knees: A point foot model,” Master’s thesis, Massachusetts Institute of Technology, 2007.

[11] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, “Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization,” IEEE Transactions on Robotics, vol. 34, no. 2, pp. 370–387, apr 2018.

[12] M. Posa, S. Kuindersma, and R. Tedrake, “Optimization and stabilization of trajectories for constrained dynamical systems,” in Proceedings of the International Conference on Robotics and Automation (ICRA), May 2016.

[13] R. D. Gregg, A. K. Tilton, S. Candido, T. Bretl, and M. W. Spong, “Control and planning of 3-d dynamic walking with asymptotically stable gait primitives,” IEEE Transactions on Robotics, vol. 28, no. 6, pp. 1415–1423, dec 2012.

[14] M. S. Motahar, S. Veer, and I. Poulikakis, “Composing limit cycles for motion planning of 3d bipedal walkers,” in 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, dec 2016.

[15] C. Liu, C. G. Atkeson, and J. Su, “Biped walking control using a trajectory library,” Robotica, vol. 31, no. 2, pp. 311–322, May 2012.

[16] C. O. Saglam and K. Byl, “Robust policies via meshing for metastable rough terrain walking,” in Proceedings of Robotics: Science and Systems, Berkeley, USA, July 2014.

[17] N. Rosa and K. M. Lynch, “Extending equilibria to periodic orbits for walkers using continuation methods,” 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sep 2014.

[18] ——, “Using equilibria and virtual holonomic constraints to generate families of walking gaits,” in Dynamic Walking Conference, Mariehamn, Finland, Jun. 2017.

[19] M. W. Gomes, “Collisionless rigid body locomotion models and physically based homotopy methods for finding periodic motions in high degree of freedom models,” Ph.D. dissertation, Cornell University, 2005.

[20] T. McGeer, “Passive dynamic walking,” Int. J. Robotics Research, vol. 9, no. 2, pp. 62–82, 1990.

[21] A. Chatterjee and M. Garcia, “Small slope implies low speed for McGeer’s passive walking machines,” Dynamics and Stability of Systems, vol. 15, no. 2, pp. 139–157, 2000.

[22] M. Garcia, A. Chatterjee, and A. Ruina, “Efficiency, speed, and scaling of two-dimensional passive-dynamic walking,” Dynamics and Stability of Systems, vol. 15, no. 2, pp. 75–99, Jun 2000.

[23] N. Rosa, A. Barber, R. D. Gregg, and K. M. Lynch, “Stable open-loop brachiation on a vertical wall,” in IEEE International Conference on Robotics and Automation, May 2012, pp. 1193–1198.

[24] N. Rosa and K. Lynch, “The passive dynamics of walking and brachiating robots: Results on the topology and stability of passive gaits,” in Nature-Inspired Mobile Robotics: Proceedings of the 16th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, 2013.

[25] B. Krauskopf, H. Osinga, and J. Galan-Viquez, Numerical Continuation Methods for Dynamical Systems: Path Following and Boundary Value Problems. Springer, 2007.

[26] B. Griffin and J. Grizzle, “Nonholonomic virtual constraints for dynamic walking,” 2015 54th IEEE Conference on Decision and Control (CDC), Dec 2015. [Online]. Available: [http://dx.doi.org/10.1109/CDC.2015.7402850]

[27] K. A. Hamed, B. G. Buss, and J. W. Grizzle, “Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations,” The International Journal of Robotics Research, vol. 35, no. 8, pp. 977–999, Jul 2016.
