Almost Driftless Navigation of 3D Limit-Cycle Walking Bipeds

Sushant Veer, Mohamad Shafiee Motahar, and Ioannis Poulakakis

Abstract— **This paper presents a method for navigating 3D dynamically walking bipedal robots amidst obstacles. Our framework relies on composing gait primitives corresponding to limit-cycle locomotion behaviors and it produces nominal motion plans that are compatible with the system's dynamics and can be tracked with high fidelity. The low-level controllers of the biped are designed within the Hybrid Zero Dynamics (HZD) framework. Exploiting the dimensional reduction afforded by HZD and properties of invariant sets of switching systems among multiple equilibria, we obtain polynomial approximations of a reduced order Poincaré map and of the net change of the center of mass location over a stride. These polynomials are then incorporated in a high-level Rapidly Exploring Random Tree (RRT) planner to generate nominal plans which are tracked by the biped with drastically low drifting errors, without adversely affecting the time for computation.**

I. Introduction

In the interest of planning speed, motion planners for robotic systems often assume a finite collection of actions that are composed to construct plans according to desired specifications [1]. In the context of dynamically walking[1] bipeds, these finite actions correspond to limit-cycle locomotion behaviors, the concatenation of which can generate more complex walking motion plans. However, these plans capture only a subset of the range of all possible behaviors exhibited by the system, thereby limiting the planner in its ability to construct paths that can be faithfully realized by the biped. As a result, a discrepancy between the suggested plan and the one followed by the system arises [2], [3]. This paper proposes a method to reduce this discrepancy by providing the planner access to suitable approximations of the underlying dynamics of the system.

The past two decades have seen a surge in the interest of planning motions with humanoid robots in high-dimensional environments; the book [4] contains a wide collection of algorithms for tasks involving whole-body motions of such systems. These algorithms are greatly facilitated by the analytical nature of the Zero Moment Point (ZMP) criterion of stability. Unlike ZMP-based walkers, much of the research on dynamically walking bipedal robots concentrates on the design of low-level control policies for stability and robustness, in isolation from high-level motion planning objectives. Various approaches can be found in the relevant literature; these include, for example, geometric reduction [5], virtual holonomic constraints [6], hybrid zero dynamics (HZD) [7], and partial hybrid zero dynamics [8]. A general framework for feedback motion planning has been proposed in [9] to enlarge the basin of attraction of desired goal regions in the state space, and can find application to limit-cycle walkers. Robustness to parameter uncertainty has been addressed in [10], [11], and to rough terrain in [12], [13]. Recently, limit-cycle walking under the influence of persistent exogenous force commands has been studied in [14], [15] for the purpose of collaborative human-robot object transportation [16].

Relative to ZMP-based walkers, motion planning to achieve high-level objectives—such as avoiding unsafe regions in the workspace—has received less attention in the context of dynamically walking bipeds. An instance of such tasks is footstep planning, in which a biped is required to carefully plan its steps to conform to terrain geometry. Exploiting the analytical tractability of HZD for planar bipeds, [17] generates feasible walking paths on known uneven terrain by using an energy-based planner to construct suitable sequences of motion primitives. Recently, [18] proposed an efficient method for safety-critical footstep planning that combines control barrier functions with control Lyapunov functions to provide guarantees of performance.

Beyond planar settings, navigation of 3D limit-cycle walkers in environments cluttered by obstacles with the purpose of reaching a desired goal location has only been treated in [2] and subsequently in [3]. The work in [2] introduced a framework for such tasks by formulating motion planning as a discrete switched system over a finite collection of gait primitives corresponding to nominal walking arcs. A rigorous analysis of the resulting switched system has been undertaken in [3], where an explicit expression of a dwell-time constraint that guarantees stable execution of a nominal plan has been obtained, together with analytical descriptions of compact regions in the state space in which the systems's state is guaranteed to remain as the plan is executed. However, both [2], [3] rely on a finite collection of *nominal* walking arcs passed as actions to the planner, which—as it constructs a plan—is ignorant of the evolution of the system in response to the planned actions. This causes the biped to drift as it executes the suggested plan, even in the absence of external perturbations. The objective of this paper is to provide a method for alleviating this drift.

This paper employs a fairly generic underactuated 3D bipedal robot model to extract a finite collection of limit-cycle walking behaviors. Formulating motion planning as a discrete switched system with multiple equilibria, and taking advantage of the dimensional reduction afforded by the HZD method, we provide explicit characterizations of low-

S. Veer, M. S. Motahar and I. Poulakakis are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA; e-mail: {veer, motahar, poulakas}@udel.edu.

This work is supported in part by NSF CAREER Award IIS-1350721 and by NRI-1327614.

[1] Here the term "dynamically walking" indicates limit-cycle walkers.

dimensional regions where the system's state is "trapped." Meshing these trapping regions, allows the representation of the actions available to the planner as polynomials of the biped's state, resulting in approximate walking arc *functions*. This way the dynamics of the biped can be accessed by the planner without adversely affecting the computational time. As a result, the suggested plans are compatible with the system's capabilities, and can be executed with minimal drift. To provide a sense of the improvement afforded by the proposed method, in an environment used in [2], [3], the final drift for an 85 stride long plan was just $3.40\mathrm{mm}$ compared to $2.59\mathrm{m}$ in [2] and $1.20\mathrm{m}$ in [3]. This drastic improvement is attributed to the planner's knowledge of the low-level dynamics of the system.

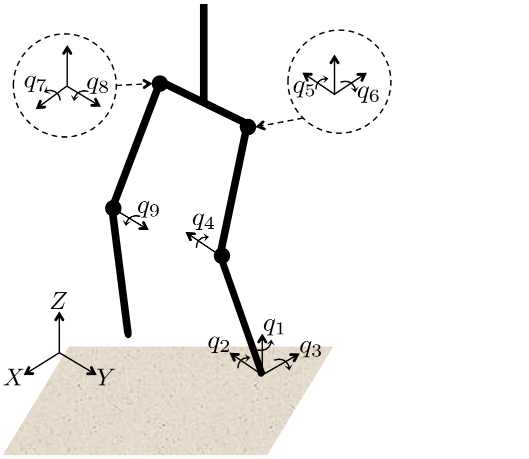

Fig. 1. Robot model with a choice of generalized coordinates when supported on left leg.

II. Modeling and Control

The 3D bipedal model considered here has two legs with knees, a non-trivial hip, and a torso; see Fig. 1. The thigh and the shin are connected at the knee with an one degree of freedom (DoF) revolute joint, and the hip and the thigh are connected with a two DoF revolute joint. The contact between the toe and the ground is modeled as a three DoF spherical joint. The model has nine DoFs $q := (q_1, ..., q_9)$ with seven actuators placed at all joints other than the yaw q_1 and pitch q_2 of the stance foot, thereby yielding 2 degrees of underactuation. The configuration space $\mathcal{Q}$ contains physically realizable configurations of the biped.

The non-zero hip width results in different dynamics for the left and right stance foot support; see [19, Section 2]. The rest of this section provides a model of the left foot stance phase. The model and the controller for the right leg support can be developed in a similar manner. Further details can be found in [19] so the exposition here will be terse.

Let $\hat{x} = (q^{\mathrm{T}}, \dot{q}^{\mathrm{T}})^{\mathrm{T}} \in \mathcal{TQ} := \{(q, \dot{q}) \mid q \in \mathcal{Q},\ \dot{q} \in \mathbb{R}^9\}$ be the state. For future use, note that $\hat{x} = (q_1, x^{\mathrm{T}})^{\mathrm{T}}$. The dynamics during the swing phase can be represented by

$$\dot{\hat{x}} = f(\hat{x}) + g(\hat{x})u \ , \tag{1}$$

where $u \in \mathbb{R}^7$ is the vector of control inputs. The vector fields f and g are defined accordingly.

The swing phase terminates when the toe of the swing leg reaches the ground. The set of states for which a valid impact of the swing toe with the ground occurs is represented by

$$\mathcal{S} := \{(q, \dot{q}) \in T\mathcal{Q} \mid \mathrm{p}^{\mathrm{v}}_{\mathrm{toe}}(q) = 0,\ \dot{\mathrm{p}}^{\mathrm{v}}_{\mathrm{toe}}(q, \dot{q}) < 0\} \ \ , \tag{2}$$

where $\mathrm{p}^{\mathrm{v}}_{\mathrm{toe}}$ denotes the height of the swing foot. The impact of the swing foot with the ground is assumed to be perfectly inelastic, i.e., the swing foot does not slip or rebound. The physics of the impact as well as the change of coordinates post impact are captured by the mapping $\Delta : \mathcal{S} \to \mathcal{TQ}$

$$\hat{x}^{+} = \Delta(\hat{x}^{-}) \ . \tag{3}$$

The controller to be used in the remainder of the paper is designed within the HZD framework. The following output is associated to the swing phase dynamics (1)

$$y = h(q) := q_{\mathrm{a}} - h_{\mathrm{d}}(\theta(q)) \ ,$$

where $\theta(q) := -q_2 - q_4/2$. We restrict our attention to gaits in which $\theta(q)$ increases monotonically during the step. The controlled joints are $q_{\mathrm{a}} := (q_3, ..., q_9)$. The desired evolution $h_{\mathrm{d}}(\theta)$ of the controlled joints is designed according to [19, Section 3], and the zero dynamics surface

$$\mathcal{Z} := \{(q, \dot{q}) \in \mathcal{TQ} \mid y = 0,\ \dot{y} = 0\} \tag{4}$$

is rendered invariant during the swing phase as shown in (4).

To ensure hybrid invariance, i.e., invariance under the action of Δ, an additional correction term, $h_{\mathrm{c}}(\theta, y_{\mathrm{i}}, \dot{y}_{\mathrm{i}})$, is added to the output

$$\tilde{y} = \tilde{h}(q, y_{\mathrm{i}}, \dot{y}_{\mathrm{i}}) := q_{\mathrm{a}} - h_{\mathrm{d}}(\theta) - h_{\mathrm{c}}(\theta, y_{\mathrm{i}}, \dot{y}_{\mathrm{i}}) \ \ , \tag{5}$$

which depends on the values y_{i} and $\dot{y}_{\mathrm{i}}$ at the beginning of the step of the output and its derivative, respectively. Details on h_{c} can be found in [19, Section 3.3]. We only mention that h_{c} vanishes by the middle of the step.

To realize turning, the output function (5) is further modified as in [19, Section 4.3] by adding an extra term

$$\tilde{y}_p = \tilde{h}_p(q, y_i, \dot{y}_i) := q_{\mathrm{a}} - h_{\mathrm{d}}(\theta) - h_{\mathrm{c}}(\theta, y_{\mathrm{i}}, \dot{y}_{\mathrm{i}}) - h_{\mathrm{s}}(\theta, \beta_p) \ \ ,$$

where $h_{\mathrm{s}}(\theta, \beta_p)$ is a polynomial of θ with its coefficients dependent on β_p, $p \in \mathcal{P}$ where $\mathcal{P}$ is a finite index set representing different turning primitives. Further ahead we define gait primitives by choosing different β_p. The zero dynamic surface associated with a primitive $p \in \mathcal{P}$

$$\tilde{\mathcal{Z}}_p := \{(q, \dot{q}) \in \mathcal{TQ} \mid \tilde{y}_p = 0,\ \dot{\tilde{y}}_p = 0\} \tag{6}$$

is rendered hybrid invariant under the effect of the control law $u^*_p(x) = -L_g L_f \tilde{h}_p(x)^{-1} L^2_f \tilde{h}_p(x)$.

The Poincaré map $\hat{P}_p : \mathcal{S} \to \mathcal{S}$ [19, Section 3.2] in closed loop with $u^*_p(x)$, $p \in \mathcal{P}$, gives rise to the following discrete dynamical system

$$\hat{x}[k+1] = \hat{P}_p(\hat{x}[k]) \ . \tag{7}$$

The equivariance of $\hat{P}_p$ under yaw rotations q_1 established in [3, Proposition 1], allows us to write (7) as

$$q_1[k+1] = q_1[k] + P^{q_1}_p(x[k]) \ , \tag{8}$$

$$x[k+1] = P_p(x[k]) \ . \tag{9}$$

Let x_p^* be a locally exponentially stable fixed point of (9). A gait primitive can be defined as a pair $\mathcal{G}_p = \{P_p, x_p^*\}$, and it corresponds to a locally exponentially stable limit cycle associated with a particular walking gait of the biped.

III. Motion Planning As A Switched System

Switching between different gait primitives $\mathcal{G}_p$ is equivalent to switching among different systems P_p defined as in (9) with corresponding fixed points x_p^*. A motion plan consists of a concatenation of these gait primitives, and can be treated as a switching signal $\sigma : \mathbb{Z}_+ \to \mathcal{P}$ that maps the stride number k to the p-th Poincaré map, giving rise to the discrete switched system

$$x[k+1] = P_{\sigma(k)}(x[k]) \ . \tag{10}$$

It should be mentioned that (10) differs from the systems studied in [20], in that it switches among vector fields that do *not* share a common equilibrium point. Hence, under persistent switching, (10) will not converge to a single equilibrium. However, [3, Theorem 1] shows that if the frequency of the switching signal is sufficiently low, the solution of (10) remains in a compact set $\overline{\mathcal{M}}$ provided that suitable initial conditions are supplied. Furthermore, this compact set can be explicitly characterized as the union of sub-level sets of Lyapunov functions, and is essential in approximating the dynamics of the biped in a form suitable for planning purposes, as in Section IV-C below.

To construct $\overline{\mathcal{M}}$, let $\mathcal{D} \subset \mathcal{S}$ be an open connected set within which P_p is defined for all $p \in \mathcal{P}$. A continuous function $V_p : \mathcal{D} \to \mathbb{R}$ is an exponential Lyapunov function, if for all $x \in \mathcal{D}$

$$\chi_{p,1}(\|x - x_p^*\|) \leq V_p(x) \leq \chi_{p,2}(\|x - x_p^*\|) \ , \tag{11}$$

$$V_p(x[k+1]) \leq \epsilon V_p(x[k]) \ , \tag{12}$$

where $\chi_{p,1}$, $\chi_{p,2}$ are class-$\mathcal{K}$ functions [21, Section 4.4], and $0 < \epsilon < 1$. For each $p \in \mathcal{P}$, let $\mathcal{N}_p(\kappa) := \{x \in \mathcal{D} : V_p(x) \leq \kappa\}$ and let the union of these sets over all $p \in \mathcal{P}$ be $\mathcal{N}(\kappa) := \bigcup_{p \in \mathcal{P}} \mathcal{N}_p(\kappa)$. Next, define $\omega_p(\kappa) := \max_{x \in \mathcal{N}(\kappa)} V_p(x)$, and let $\omega_{\max}(\kappa)$ and $\omega_{\min}(\kappa)$ be the maximum and minimum of $\omega_p(\kappa)$ over the finite index set $\mathcal{P}$, respectively. Let $\mathcal{M}_p(\kappa) := \{x \in \mathcal{D} : V_p(x) \leq \omega_p(\kappa)\}$ and define

$$\overline{\mathcal{M}}(\kappa) := \bigcup_{p \in \mathcal{P}} \mathcal{M}_p(\kappa) \ , \quad \underline{\mathcal{M}}(\kappa) := \bigcap_{p \in \mathcal{P}} \mathcal{M}_p(\kappa) \ .$$

By construction $\mathcal{N}(\kappa) \subset \underline{\mathcal{M}}(\kappa)$ and $\overline{\mathcal{M}}(\kappa)$ is connected.

For the solution to remain in $\overline{\mathcal{M}}(\kappa)$, a constraint on the switching frequency is required. To express this constraint, let N_d be the dwell time of the switching signal $\sigma(k)$ defined as the least number of strides between any two consecutive switches. Then, the following theorem provides a lower bound on the dwell time, which guarantees that a solution starting in $\underline{\mathcal{M}}(\kappa)$ will stay in $\overline{\mathcal{M}}(\kappa)$.

Theorem 1: Consider (10) and assume that for each $p = \sigma(k) \in \mathcal{P}$ there exists a function $V_p : \mathcal{D} \to \mathbb{R}$ that satisfies (11) and (12). Let $\mu(\kappa) > 1$ be such that

$$\frac{V_{p_i}(x)}{V_{p_j}(x)} \leq \mu(\kappa), \ \forall p_i, p_j \in \mathcal{P}, \ \forall x \in \mathcal{D} \setminus \mathcal{N}(\kappa) \ .$$

Assume further that the dwell time $N_\mathrm{d} \in \mathbb{Z}_+$ of σ satisfies

$$N_\mathrm{d} \geq \frac{\log\left(\mu(\kappa)\frac{\omega_{\max}(\kappa)}{\omega_{\min}(\kappa)}\right)}{\log(1/\epsilon)} \ . \tag{13}$$

Then, for every initial condition in the set $\underline{\mathcal{M}}(\kappa)$, the solution of (10) remains in $\overline{\mathcal{M}}(\kappa)$.
A proof of Theorem 1 can be found in [3].

In what follows, we particularize these constructions to plan motions for 3D bipedal robots controlled using HZD. We will focus on the implications of Theorem 1 in facilitating the computation of approximate state update maps to be used as actions by the planning algorithm, thereby informing the planner of the underlying dynamics.

IV. Generating Walking Arcs

In this section we exploit the dimensional reduction afforded by HZD and discuss walking arcs that are provided to the planner as actions available for planning.

A. HZD Reduction

We switch between different gait primitives at the beginning of the stride. The turn-inducing output $h_\mathrm{s}(\theta, \beta_p)$ and the correction $h_\mathrm{c}(\theta, y_\mathrm{i}, \dot{y}_\mathrm{i})$ are chosen such that they vanish by the end of the stride [19, Section 4.3]. This ensures that, despite switching, the state always evolves on the corresponding $\tilde{\mathcal{Z}}_p$ of (6), and that by the end of the stride $\mathcal{S} \cap \tilde{\mathcal{Z}}_p = \mathcal{S} \cap \mathcal{Z}$, where $\mathcal{Z}$ is defined in (4) and does not depend on p. This property greatly facilitates planning by allowing us to limit our attention on the 3 dimensional surface $\mathcal{S} \cap \mathcal{Z}$ instead of the 17-dimensional $\mathcal{S}$. The reduced Poincaré map is then defined as the restriction $\hat{\rho}_p := \hat{P}_p|_{\mathcal{S} \cap \mathcal{Z}}$.

With $\hat{z} = (q_1, \dot{q}_1, \dot{\theta})^\mathrm{T}$ serving as a valid set of coordinates on $\mathcal{S} \cap \mathcal{Z}$ and by following arguments similar to [3, Proposition 1], the restricted Poincaré map takes the form

$$\begin{bmatrix} q_1[k+1] \\ z[k+1] \end{bmatrix} = \begin{bmatrix} q_1[k] + \rho_p^{q_1}(z[k]) \\ \rho_p(z[k]) \end{bmatrix} =: \hat{\rho}_p(\hat{z}[k]) \ , \tag{14}$$

where $z = (\dot{q}_1, \dot{\theta})^\mathrm{T}$.

The dimensional reduction discussed above allows the definition of low-dimensional gait primitives $\mathcal{R}_p = \{\rho_p, z_p^*\}$, where z_p^* is a locally exponentially stable fixed point of

$$z[k+1] = \rho_p(z[k]) \ . \tag{15}$$

B. Nominal Walking Arcs

The dimensionally reduced gait primitives $\mathcal{R}_p$ can be used to extract basic planning actions in the form of *nominal walking arcs* (NWAs). Each NWA corresponds to a path realized in the biped's workspace as it follows a limit-cycle gait primitive $\mathcal{R}_p$. NWAs are made available to the planning algorithm in a discrete form, effectively capturing the net change in the Cartesian position of the center of mass (CoM) and in the heading direction of the biped as it moves along a limit-cycle gait primitive $\mathcal{R}_p$.

To represent NWAs in a form suitable for planning purposes note that by (14) the change $(\delta X_p, \delta Y_p)$ in the CoM's Cartesian position over a stride is independent of q_1 when it is expressed in a frame of reference attached to the stance toe and aligned with the current heading direction. Hence, $(\delta X_p, \delta Y_p)$ depends only on the state z, and so does the change $\delta q_{1,p}$ in the heading angle. Given an initial state z, computing $\{\delta q_{1,p}(z), \delta X_p(z), \delta Y_p(z)\}$ for the p-th gait primitive would require evaluating the corresponding Poincaré map ρ_p at z as in (15). This process involves the numerical integration of the hybrid zero dynamics, which despite its reduced dimension, would slow down the planning process considerably. This can be avoided by admitting a set of actions available to the planner that corresponds to a discrete set of fixed points z_p^*; namely

$$\mathcal{A}_p^*(z_p^*) := \{\delta q_{1,p}(z_p^*), \delta X_p(z_p^*), \delta Y_p(z_p^*)\} \ . \tag{16}$$

In other words, the actions (16) provided to the planner can be thought of as output functions of the discrete system (15) *evaluated* at the corresponding fixed point z_p^*; see [2, Table I,II] and [3, Table I].

The fact that the planning algorithm has available a discrete collection of actions $\{\mathcal{A}_p^*(z_p^*), p \in \mathcal{P}\}$ significantly reduces the time required to compute feasible plans. However, the actual state of the system z may not coincide with the corresponding fixed point z_p^* when the planner requires the execution of an action $\mathcal{A}_p^*(z_p^*)$, thereby causing the system to drift from the suggested plan. This problem arises even under nominal conditions—i.e., in the absence of externally applied perturbations—due to the concatenation of the actions suggested by the planner. For example, consider the situation depicted in Fig. 2 that involves the execution of two actions. If the state z_{a} at the beginning of the execution coincides with the fixed point z_1^* corresponding to the intended action $\mathcal{A}_1^*(z_1^*)$, then at the end of the execution the state $z_{\mathrm{b}} = z_1^*$ under nominal conditions. Subsequent application of a different action $\mathcal{A}_2^*(z_2^*)$ assumes that the system starts at state z_2^* which does not agree with z_{b}, thereby resulting in an error at the end of the execution. In effect, this error arises because the system (15) switches among different equilibrium points.

This error accumulates over the steps, resulting in possibly large drifts by the end of the plan depending on the environment [2]. The explicit characterization of the "trapping" regions $\overline{\mathcal{M}}(\kappa)$ provided by Theorem 1 can help reduce this drift, as can be seen in [3]. Indeed, by selecting a smaller κ, the size of $\overline{\mathcal{M}}(\kappa)$ can be reduced, forcing the state z to evolve nearer to the fixed points z_p^*. On the other hand, reducing the value of κ requires an increase in the dwell time N_{d}. Intuitively, to avoid large excursions in the evolution of the state z, the system would have to wait long enough on a specific gait primitive before it is allowed to switch to a new one. Clearly, this strategy greatly reduces the flexibility of the planner to generate motion plans in environments with "tight" workspaces.

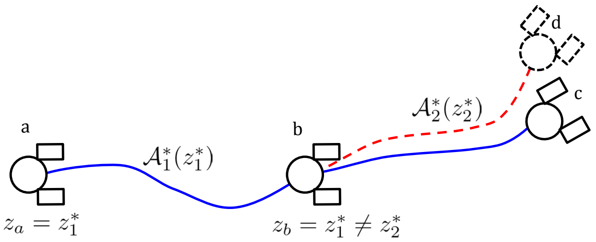

Fig. 2. Illustration comparing actual path taken by the biped (blue) and the path predicted using NWA (dashed red).

C. Approximate Walking Arc Functions

One way to eliminate the drift associated with NWAs and the corresponding actions would be to allow the planner to compute $(\delta X_p, \delta Y_p)$ and $\delta q_{1,p}$ at a point z, which may not be the fixed point of the intended action. As a result, the planner needs to keep track of the low level state z, which can be achieved by augmenting the set of actions with the Poincaré map corresponding to the intended gait primitive, forming *walking arc functions* (WAFs); i.e.,

$$\mathcal{A}_p(z) := \{\rho_p(z), \delta q_{1,p}(z), \delta X_p(z), \delta Y_p(z)\} \ . \tag{17}$$

In more detail, if the action $\mathcal{A}_p$ is implemented at step k, the planner computes $(\delta X_p, \delta Y_p)$ and $\delta q_{1,p}$ based on the state $z[k]$ to advance the position and orientation of the biped, and computes the prediction $\rho_p(z[k])$ to be used at the *next step*[2].

In the absence of perturbations, planning with WAFs instead of NWAs results in motion plans realized by the biped with no drift. However, this is achieved at the expense of computing the Poincaré map, which, as was mentioned above, entails significant computational burden. In what follows, we will exploit Theorem 1 and properties of the associated set constructions to obtain *approximate walking arc functions* (AWAF), which will then be used to provide actions to the planner as

$$\tilde{\mathcal{A}}_p(z) := \{\tilde{\rho}_p(z), \delta\tilde{q}_{1,p}(z), \delta\tilde{X}_p(z), \delta\tilde{Y}_p(z)\} \ , \tag{18}$$

where the symbol " ˜ " represents a suitable approximation of the corresponding function. The approximate actions $\tilde{\mathcal{A}}_p(z)$ can be computed efficiently at any point z in their domain, thus drastically decreasing the error associated to the actions $\mathcal{A}_p^*(z_p^*)$ based on NWAs.

To obtain valid approximations to be used in (18), polynomial functions will be fitted to the values $\rho_p(z)$, $\delta q_{1,p}(z)$, $\delta X_p(z)$, and $\delta Y_p(z)$ of the actions in (17) at points z of an appropriately dense mesh of the set $\mathcal{S} \cap \mathcal{Z}$ over which (15) evolves. Clearly, meshing $\mathcal{S} \cap \mathcal{Z}$ can be tedious. However, Theorem 1 significantly simplifies the process, for it establishes conditions under which the evolution of the system is confined in the subset $\overline{\mathcal{M}}(\kappa) \subset \mathcal{S} \cap \mathcal{Z}$. Thus, it allows one to restrict the mesh to the region $\overline{\mathcal{M}}(\kappa)$, which in addition admits an explicit characterization.

In more detail, to apply Theorem 1, we use quadratic functions $V_p(z) = z^{\mathrm{T}} S_p z$, where S_p is the solution of the discrete Lyapunov equation of the linearization of (15) about a fixed point z_p^*. Sums-of-squares (SOS) programming [9] is then employed to estimate the region over which V_p is a valid Lyapunov function according to (11)-(12). The details can be found in [3, Section V-C] and are not given here for

[2] Note that advancement of the state is necessary since the planner works offline to generate *a priori* motion plans.

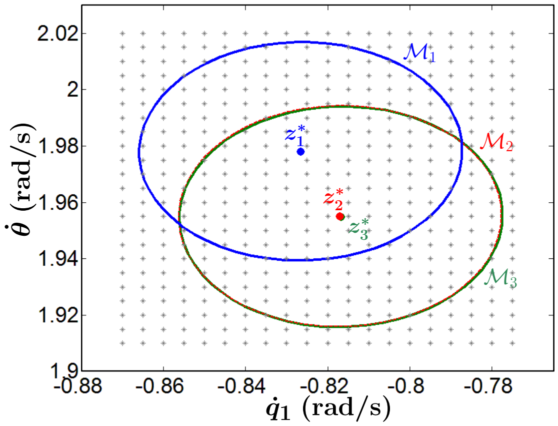

Fig. 3. Mesh used for obtaining AWAF for a collection of gait primitives. Grey stars denote the nodes of the mesh. The evolution of (15) is trapped in $\overline{\mathcal{M}}(\kappa)$, which is the union of the sub-level sets of the corresponding Lyapunov functions, and lies entirely in $\mathcal{B}_R$ defined by (19). Note that $\mathcal{M}_2$ and $\mathcal{M}_3$ corresponding to the fixed points z_2^* and z_3^* almost overlap.

space limitations. Figure 3 shows the sub-level sets of V_p associated with three fixed points, the union of which is the set $\overline{\mathcal{M}}(\kappa)$ required by Theorem 1. If the initial condition is in the intersection of the sub-level sets[3], i.e. $z[0] \in \underline{\mathcal{M}}(\kappa)$, selecting N_d according to (13) ensures that the system never escapes from $\overline{\mathcal{M}}(\kappa)$. To construct a mesh that covers $\overline{\mathcal{M}}(\kappa)$, let $\bar{z}^*$ be the centroid of the fixed points z_p^* for all $p \in \mathcal{P}$ and choose $R > 0$ large enough so that

$$\mathcal{B}_R(\bar{z}^*) := \{z \in \mathcal{S} \cap \mathcal{Z} \;:\; \|z - \bar{z}^*\|_\infty \leq R\} \;, \qquad (19)$$

includes the set $\overline{\mathcal{M}}(\kappa)$. An appropriately dense mesh size on $\mathcal{B}_R$ can then be selected and the nodes of the mesh are extracted. The functions ρ_p, $\delta q_{1,p}$, δX_p, and δY_p can then be computed at each node of the mesh and fitted with polynomials of sufficiently high degree using least square minimization, resulting in the approximations used in (18).

V. Minimizing Drift through AWAFs

This section evaluates the performance of AWAFs in comparison to NWAs. The AWAFs are subsequently used to navigate the 3D biped in environments cluttered by obstacles.

A. Comparing AWAFs versus NWAs

For concreteness, we consider a set of three gait primitives $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$, where $\mathcal{R}_1$ corresponds to the biped walking straight, and $\mathcal{R}_2$ and $\mathcal{R}_3$ to the biped turning by 45^o and -45^o, respectively. As described in Section IV-C, applying Theorem 1 with SOS-verified Lyapunov functions it is found that for $\kappa = 0.0002$ any plan satisfying $N_\text{d} \geq 1$ will not result in loss of stability. The set $\overline{\mathcal{M}}(\kappa)$ can then be computed and $R = 0.05$ is chosen to construct the set $\mathcal{B}_R$ which is discretized using a mesh of 400 nodes, as in Fig. 3.

In deriving the approximate actions used in (18), it is convenient to replace $(\delta X_p(z), \delta Y_p(z))$ with their polar counterparts $l_p(z) = \|(\delta X_p(z), \delta Y_p(z))\|$ and $s_p(z) = \arctan(\delta Y_p(z)/\delta X_p(z))$. Then, for each primitive $\mathcal{R}_p$, the functions $\rho_p(z)$, $\delta q_{1,p}(z)$, $l_p(z)$, and $s_p(z)$, are computed at the nodes of the mesh, and their approximations $\tilde{\rho}_p(z)$, $\delta\tilde{q}_{1,p}(z)$, $\tilde{l}_p(z)$, and $\tilde{s}_p(z)$ are derived as polynomials of degrees 3, 6, 3, and 5, respectively. It is worth noting that it takes around 30 s to compute one step of (15) by numerical integration of the actual dynamics whereas the polynomial computation requires fractions of milliseconds.

We can now examine the final drift error accumulated by concatenating actions in $\mathcal{A}_p^*(z_p^*)$ of (16) and $\tilde{\mathcal{A}}_p(z)$ of (18) corresponding to NWAs and AWAFs, respectively. To do this 50 sequences of actions are generated, each composed by 100 actions randomly chosen from $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ with equal likelihood. The actual dynamics (15) of the model is used to obtain the actual position and orientation of the biped after the execution of each sequence. The error between the final position of the biped and of the one resulting from the actions corresponding to NWAs and AWAFs is recorded in Table I. Clearly, the average drift recorded using the AWAFs is drastically smaller than that obtained by NWAs. Figure 4 shows error accumulation for one of the 50 sample sets, demonstrating that AWAFs significantly outperform NWAs.

TABLE I

Final Drift Error for NWAs and AWAFs

Method	Mean [m]	Std. Dev. [m]	Min [m]	Max [m]
NWA	0.6658	0.4692	0.0444	2.1335
AWAF	0.0047	0.0026	0.0005	0.0104

B. Application to Motion planning

This section uses the actions (18) corresponding to AWAFs in conjunction with a high-level planning algorithm to construct feasible paths taking the biped from an initial position to a desired final one while avoiding any obstacles on the way. In what follows, (X, Y) represent the coordinates of the CoM of the biped in the global frame and Θ the heading angle of the biped with respect to the global positive X-axis. A Rapidly Exploring Random Tree (RRT) [1, Section 5.5] is employed to find feasible paths. Each node of the associated tree holds information about (X, Y, Θ), the low-level state of

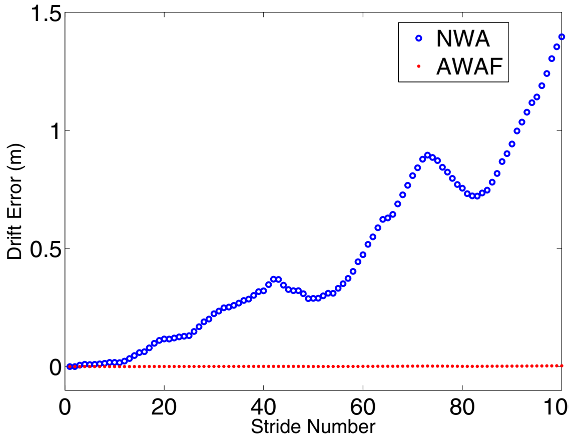

Fig. 4. Comparison of drift error between NWA (blue) and AWAF (red)

[3] Since nominal plans are generated, it can always be imposed that the initial state of (15) corresponds to a fixed point which belongs in $\underline{\mathcal{M}}(\kappa)$.

the biped z, the index of its parent node, and the primitive index $p \in \mathcal{P}$ that was applied on the parent node. In addition, the number of strides n since the last primitive switch on the path is available in the tree to check the dwell time condition (13) which the planner must respect.

The tree node k such that (X_k, Y_k) has the least euclidean distance from a randomly chosen point (X_r, Y_r) in the free space among all other nodes, is expanded for each $p \in \mathcal{P}$ to obtain the successive node as

$$\begin{aligned} X^p_{k+1} &= X_k + \tilde{l}_p(z_k)\cos(\Theta_k + \tilde{s}_p(z_k)) \ , \\ Y^p_{k+1} &= Y_k + \tilde{l}_p(z_k)\sin(\Theta_k + \tilde{s}_p(z_k)) \ , \\ \Theta^p_{k+1} &= \Theta_k + \delta\tilde{q}_{1,p}(z_k) \ , \\ z^p_{k+1} &= \tilde{\rho}_p(z_k) \ . \end{aligned}$$

If (X^p_{k+1}, Y^p_{k+1}) is not in the free space, then the corresponding node is pruned. The tree generation continues until (X^p_{k+1}, Y^p_{k+1}) is within a distance $\gamma > 0$ of the goal position or until a maximum number N of iterations is reached.

To demonstrate the power of AWAFs as actions available to the planning algorithm, the environment of Fig. 5 is borrowed from [2], [3]. Both methods in [2] and [3] were successful in generating feasible paths with final drift errors (assuming the same initial and goal positions) being 2.59m in [2] and 1.20m in [3]. Using AWAFs, Fig. 5 shows that despite the length of the path, the biped tracks the nominal plan almost perfectly, resulting in a final drift error of just 3.40mm! Note that, given that the step length of the biped is approximately 0.34m, the error for a plan with 85 strides, i.e. 170 steps, is about $1/100$ of the step length.

VI. Conclusion

This paper presents a method to reduce the drifting error observed during the execution of a motion plan by a dynamically walking 3D bipedal robot. Our approach exploits the dimensional reduction afforded by HZD to obtain actions for the planner that approximate the evolution of the state of the biped during the construction of a plan. These actions are used in an RRT-based planner to generate nominal motion plans, which can be faithfully realized by the biped, all the while stable execution is guaranteed in the absence of exogenous disturbances. In future work, we intend to generate motions plans robust to disturbances and validate our method experimentally. The overall goal of this work is to facilitate motion planning for high-dimensional systems with complex low-level dynamic behaviors using established planning algorithms, and to apply these methods in the context of dynamically walking bipedal robots.

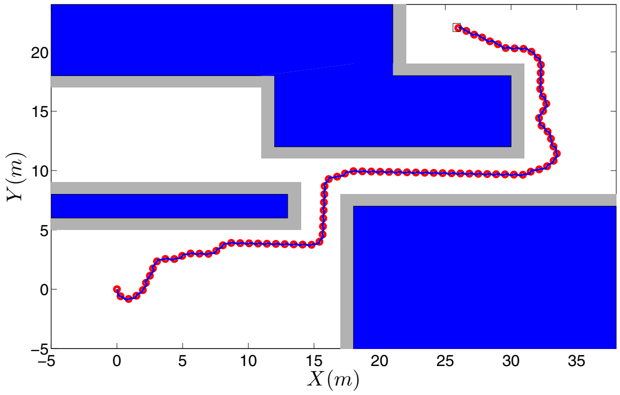

Fig. 5. The nominal plan generated using AWAFs (red circles) and the simulated trajectory of the CoM of the biped (blue line). The initial and goal locations are $(0\ \text{m}, 0\ \text{m})$ and $(25\ \text{m}, 22\ \text{m})$ and the initial heading direction is -90^o. The biped takes 85 strides to reach the goal (marked by black rectangle) and the final drift error is 3.4 mm.

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, 2006.

[2] 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, 2012.

[3] M. S. Motahar, S. Veer, and I. Poulakakis, "Composing limit cycles for motion planning of 3D bipedal walkers," in *Proc. of IEEE Conference on Decision and Control*, Las Vegas, December 2016, pp. 6368–6374.

[4] K. Harada, E. Yoshida, and K. Yokoi, Eds., *Motion Planning for Humanoid Robots*. Springer-Verlag, 2010.

[5] R. D. Gregg and M. W. Spong, "Reduction-based control of three-dimensional bipedal walking robots," *Int. J. of Robotics Research*, vol. 29, no. 6, p. 680702, May 2009.

[6] L. B. Freidovich, U. Mettin, A. S. Shiriaev, and M. W. Spong, "A passive 2-dof walker: Hunting for gaits using virtual holonomic constraints," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1202–1208, October 2009.

[7] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Boca Raton, FL: CRC Press, 2007.

[8] A. D. Ames, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, May 2014.

[9] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *Int. J. of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[10] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs," in *Proc. of Robotics: Science and Systems*, 2015.

[11] S. Kolathaya and A. D. Ames, "Parameter sensitivity and boundedness of robotic hybrid periodic orbits," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 377–382, 2015.

[12] C. O. Saglam and K. Byl, "Switching policies for metastable walking," in *Proc. of IEEE Conference on Decision and Control*, Florence, Italy, December 2013, pp. 977–983.

[13] 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," *Int. J. of Robotics Research*, vol. 35, no. 8, pp. 977–999, July 2016.

[14] S. Veer, M. S. Motahar, and I. Poulakakis, "On the adaptation of dynamic walking to persistent external forcing using hybrid zero dynamics control," in *Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Hamburg, September 2015, pp. 997–1003.

[15] ——, "Local input-to-state stability of dynamic walking under persistent external excitation using hybrid zero dynamics," in *Proc. of the American Control Conference*, Boston, July 2016, pp. 4801–4806.

[16] M. S. Motahar, S. Veer, J. Huang, and I. Poulakakis, "Integrating dynamic walking and arm impedance control for cooperative transportation," in *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Hamburg, September 2015, pp. 1004–1010.

[17] I. R. Manchester and J. Umenberger, "Real-time planning with primitives for dynamic walking over uneven terrain," in *Proc. of IEEE Int. Conference on Robotics and Automation*, 2014, pp. 4639–4646.

[18] Q. Nguyen and K. Sreenath, "Safety-critical control for dynamical bipedal walking with precise footstep placement," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 147–154, 2015.

[19] C.-L. Shih, J. Grizzle, and C. Chevallereau, "From stable walking to steering of a 3d bipedal robot with passive point feet," *Robotica*, vol. 30, no. 07, pp. 1119–1130, 2012.

[20] D. Liberzon, *Switching in Systems and Control*. Birkhäuser, 2003.

[21] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice hall, 2002.