First Steps toward Formal Controller Synthesis for Bipedal Robots

Aaron D. Ames
Texas A&M University
aames@tamu.edu

Paulo Tabuada
University of California,
Los Angeles
tabuada@ee.ucla.edu

Bastian Schürmann
University of California,
Los Angeles
schuermann@ucla.edu

Wen-Loong Ma and
Shishir Kolathaya
Texas A&M University
{wenlongma,shishirny}
@tamu.edu

Matthias Rungger
TU München
matthias.rungger@tum.de

Jessy W. Grizzle
University of Michigan
grizzle@eecs.umich.edu

ABSTRACT

Bipedal robots are prime examples of complex cyber-physical systems (CPS). They exhibit many of the features that make the design and verification of CPS so difficult: hybrid dynamics, large continuous dynamics in each mode (e.g., 10 or more state variables), and nontrivial specifications involving nonlinear constraints on the state variables. In this paper, we propose a two-step approach to formally synthesize control software for bipedal robots so as to enforce specifications by design and thereby generate physically realizable stable walking. In the first step, we design outputs and classical controllers driving these outputs to zero. The resulting controlled system evolves on a lower dimensional manifold and is described by the hybrid zero dynamics governing the remaining degrees of freedom. In the second step, we construct an abstraction of the hybrid zero dynamics that is used to synthesize a controller enforcing the desired specifications to be satisfied on the full order model. Our two step approach is a systematic way to mitigate the curse of dimensionality that hampers the applicability of formal synthesis techniques to complex CPS. Our results are illustrated with simulations showing how the synthesized controller enforces all the desired specifications and offers improved performance with respect to a controller that was utilized to obtain walking experimentally on the bipedal robot AMBER 2.

1. INTRODUCTION

Legged robots are complex highly dynamic CPS. As a concrete example, consider MABEL shown in Fig. 1. This bipedal robot possesses highly nonlinear, non-order preserving, non-convex dynamics described by a hybrid model with 14 state variables and four actuators [23]. To enable MABEL to accept a set of high-level locomotion commands over a network, and successfully execute the commands while responding automatically and safely to significant uncertainty in the assumed profile of the environment, the finite state machine shown in Fig. 1 was designed [22]. It allowed MABEL to compose on the fly a set of low-level control algorithms executing a handful of motion primitives. A team of graduate students spent countless hours perfecting the transition conditions among the various nodes of the state machine. Each time a small change was made in one of the software or hardware components, such as adjusting a transition condition or adding a sensor, the entire state machine had to be completely retested, leading often to the redesign of other software components. *There is a pressing need to understand this, and more general CPSs, in a way that allows for the automatic synthesis of embedded control software that is provably correct by construction.*

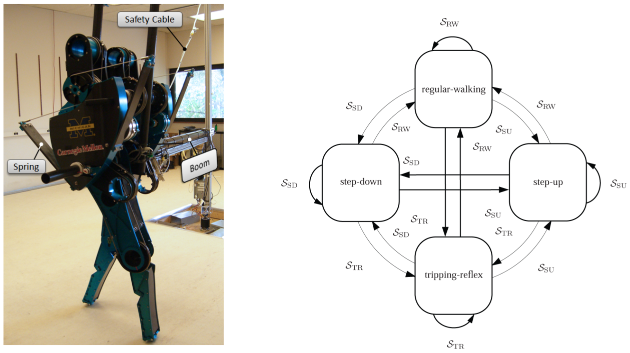


Figure 1: (a) The planar (2D) biped MABEL was developed for the study of highly dynamic locomotion. (b) High-level motion primitives in MABEL to allow walking over very rough ground without tripping. While tools for automatic low-level control algorithm synthesis are well developed, at the state machine level, all tuning was done by hand for lack of appropriate tools. "Probable correctness" was established through extensive simulation and experiments.

In this paper, we begin to lay the groundwork for this correct-by-construction control software design process in the context of highly dynamic systems. In particular, the specific bipedal robot that will be studied is AMBER 2 [18,

HSCC '15, April 14-16, 2015, Seattle, WA, USA
Copyright 2015 ACM 978-1-4503-3433-4/15/04$15.00
http://dx.doi.org/10.1145/2728606.2728611.

33] as shown in Fig. 2a. We consider a walking gait with the simplest discrete structure, resulting in a single-mode hybrid model with 12 state variables and 6 actuators. While we seek formal guarantees on the behavior of the 12-dimensional closed-loop system, we do not propose to perform formal synthesis on a model this large. Similarly to the work in [5, 6], we focus on the regulation of certain system outputs and use advanced nonlinear control methods to transform the highly complex dynamics to a simpler, more tractable system which is amenable to the correct-by-construction synthesis techniques. In contrast to [5, 6], where the authors exploit differential flatness to reduce the nonlinear synthesis problem to a controller design problem for a chain of integrators, our method applies to the aforementioned hybrid system with non-flat outputs. Specifically, in our approach the chain of integrators is forced to be at equilibrium and we apply the symbolic abstraction techniques to a hybrid system that lives on an attractive, hybrid-invariant, low-dimensional manifold which is "complementary" to the state space of the integrators [2, 30]. The low-dimensional hybrid subsystem is called the Hybrid Zero Dynamics (HZD), and its solutions can be used to reconstruct the solutions of the high-dimensional hybrid system. The end result is the ability to guarantee specifications on the full-order high-dimensional system via the reduced order representation encoded by the HZD.

There is a growing interest in the synthesis of correct-by-construction controllers for robotic applications as evidenced by the growing body of work on this topic [28, 7, 16, 8]. Although the techniques we employ for synthesis are based on the symbolic abstraction techniques described in [27], what sets the results in this paper apart from prior work is the complexity of the system being controlled. In particular, as previously mentioned, the hybrid model for AMBER 2 requires 12 state variables, and control synthesis for a model of this size is far larger than what has been previously reported in the literature. The key to scaling the symbolic controller synthesis techniques to this level of complexity is the new design flow based on the HZD. This is the main contribution of the paper as we believe that its applicability transcends the specific formal synthesis technique we employ and the robotics domain in which we develop the result.

The rest of the paper is laid out as follows: We begin by introducing hybrid systems in Section 2, with a special focus on the hybrid model of walking robots with a single domain. In addition, we introduce the notion of a stable walking gait (without requiring periodicity), and give conditions that ensure that these walking gaits are physically realizable. In Section 3, we introduce a means for dimension reduction through the use of a pre-feedback controller that renders a low-dimensional surface hybrid invariant yielding HZD. Importantly, for this initial result, the controller also results in linear dynamics on the hybrid zero dynamics surface and, ultimately, a two-dimensional linear hybrid system. We establish the first theoretic results of the paper in Section 3, wherein it is shown that solutions of the HZD lift to solutions of the full-order hybrid system and that the HZD manifold is stable as a set. Section 4 introduces the main results of the paper: a means for formally constructing controllers via the HZD to yield provably stable walking gaits that satisfy physical realizability constraints. The paper concludes with Section 5 where simulation results utilizing the correct-by-construction controller are presented.

2. BIPEDAL ROBOT MODELS, WALKING GAITS AND PHYSICAL CONSTRAINTS

In this section, we present a formalism for hybrid systems that is sufficient for modeling bipedal walking robots. After the introduction of these models, we present a definition of a walking gait for a bipedal robot along with associated physical realizability constraints. This will set the stage for formal controller synthesis for bipedal robots.

Hybrid Systems & Executions. We begin by introducing hybrid (control) systems (also referred to as systems with impulse effects [10, 11]). We consider hybrid systems with one domain because the specific biped models considered in this paper applies to flat-footed walking; for more complex foot behavior, more elaborate hybrid systems must be considered [9, 11, 12, 33].

A (simple) *hybrid control system* is a tuple,

$$\mathscr{HC} = (\mathcal{D}, U, S, \Delta, f, g), \tag{1}$$

where $\mathcal{D}$ is the *domain* with $\mathcal{D} \subseteq \mathbb{R}^n$ a smooth submanifold of the state space $\mathbb{R}^n$, $U \subseteq \mathbb{R}^m$ is the set of admissible controls, $S \subset \mathcal{D}$ is a proper subset of $\mathcal{D}$ called the *guard* or *switching surface*, $\Delta : S \to \mathcal{D}$ is a smooth map called the *reset map*, and (f, g) is a *control system* on $\mathcal{D}$, i.e., in coordinates: $\dot{x} = f(x) + g(x)u$ with $u \in U$. A *hybrid system* is a hybrid control system with $U = \{0\}$; a particular example would include a closed-loop hybrid system, meaning that a feedback controller has been applied, defining the inputs as functions of the state. In this case,

$$\mathscr{H} = (\mathcal{D}, S, \Delta, f),$$

where f is a *dynamical system* on $\mathcal{D} \subseteq \mathbb{R}^n$, i.e., $\dot{x} = f(x)$.

For the sake of simplicity, and without loss of generality for the formal results presented, we will consider infinite *solutions* (or *hybrid flows* or *executions*) of a hybrid system $\mathscr{H}$. Motivated by existing definitions [17, 30, 9, 15], we define a solution to a hybrid system $\mathscr{H}$ by the tuple:

$$\chi^{\mathscr{H}} = (\mathcal{I}, \mathcal{C}),$$

where $\mathcal{I} = \{I_i\}_{i \in \mathbb{N}}$ is a *hybrid interval* where $I_i = [\tau_i, \tau_{i+1}]$ with $\tau_i, \tau_{i+1} \in \mathbb{R}$ and $\tau_i \leq \tau_{i+1}$, and $\mathcal{C} = \{c_i\}_{i \in \mathbb{N}}$ is a collection of solutions to f, i.e., $\dot{c}_i(t) = f(c_i(t))$ for all $t \in I_i$. In addition, we require the following conditions to hold:

(i) $c_i(t) \in \mathcal{D}$ for all $t \in I_i$, $i \in \mathbb{N}$,

(ii) $\tau_{i+1} = \inf\{t \geq \tau_i : c_i(t) \in S\}$,

(iii) $\Delta(c_i(\tau_{i+1})) = c_{i+1}(\tau_{i+1})$.

The initial condition for a hybrid flow is $x_0 = c_0(\tau_0)$. When we wish to make explicit the initial condition of $\chi^{\mathscr{H}}$ we will write $\chi^{\mathscr{H}}(x_0)$.

Robotic Hybrid System Models. Utilizing the formulation of hybrid systems, we will now construct hybrid system models for bipedal robots. Specifically, we will consider a hybrid control system of the form:

$$\mathscr{HC}_R = (\mathcal{D}_R, U_R, S_R, \Delta_R, f_R, g_R). \tag{2}$$

The constructions of this section will be presented in the general case of a robot with a single discrete phase of walking, i.e., they will not be specific to the robot—AMBER 2—that will be considered in this paper. As a result they are applicable to both 2D and 3D robots in the case of full actuation, including humanoid robots. It is important to note that the constructions considered in this paper do not apply to robots with underactuation (since, in this case, there will not be actuation in the zero dynamics), yet future work will be devoted to considering this case as well.

Continuous Dynamics: Let $\mathcal{Q}_R$ be the configuration space of a robot with n degrees of freedom, i.e., $n = \dim(\mathcal{Q}_R)$, with coordinates $\theta \in \mathcal{Q}_R$. For the sake of definiteness, it may be necessary to choose $\mathcal{Q}_R$ to be a subset of the actual configuration space of the robot so that global coordinates can be defined, i.e., such that $\mathcal{Q}_R$ is embeddable in $\mathbb{R}^n$, or more simply $\mathcal{Q}_R \subset \mathbb{R}^n$. Calculating the mass and inertia properties of each link of the robot, coupled with the Euler-Lagrange equations [20, 26], yields the affine control systems (f_R, g_R):

$$f_R(\theta,\dot\theta)=\begin{bmatrix}\dot\theta\\ -D^{-1}(\theta)C(\theta,\dot\theta)\end{bmatrix},\quad g_R(\theta)=\begin{bmatrix}\mathbf{0}\\ D^{-1}(\theta)B\end{bmatrix}. \quad (3)$$

where D is the mass-inertia matrix, C contains the Coriolis/centrifugal effects and gravitational terms, and $B \in \mathbb{R}^{n\times n}$ is the actuation matrix and assumed to be nonsingular, i.e., there is one independent actuator for each degree of freedom. Such robot models are said to be fully actuated. Finally, since we are assuming full-actuation, the set of admissible values is given by $U_R \subseteq \mathbb{R}^n$.

Domain and Guard: The domain specifies the allowable configuration of the system as specified by a unilateral constraint function h; for the bipeds considered in this paper, this function specifies that the non-stance foot must be above the ground, i.e., h is the height of the non-stance foot and the system is subject to the unilateral constraint $h \geq 0$. Therefore, the domain $\mathcal{D}_R$ is given by:

$$\mathcal{D}_R = \left\{(\theta,\dot\theta) \in T\mathcal{Q}_R : h(\theta) \geq 0\right\}, \quad (4)$$

where $T\mathcal{Q}_R$ is the tangent bundle of $\mathcal{Q}_R$. The guard is just the boundary of the domain with the additional assumption that the unilateral constraint is decreasing:

$$S_R = \left\{(\theta,\dot\theta) \in T\mathcal{Q}_R : h(\theta) = 0 \text{ and } dh(\theta)\dot\theta < 0\right\}, \quad (5)$$

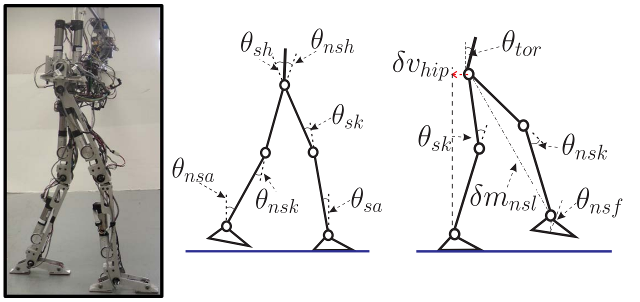
where $dh(\theta)$ is the Jacobian of h at θ.

Discrete Dynamics. The discrete dynamics of the robot determine how the velocities of the robot change when the foot impacts the ground, while simultaneously swapping the roles of the "stance" and "non-stance" legs. In particular, the reset map Δ_R is given by:

$$\Delta_R : S_R \to \mathcal{D}_R, \qquad \Delta_R(\theta,\dot\theta) = \begin{bmatrix}\Delta_\theta\theta\\ \Delta_{\dot\theta}(\theta)\dot\theta\end{bmatrix}, \quad (6)$$

where Δ_θ is the relabeling of the configuration variable associated with the stance and non-stance leg at impact. Here, $\Delta_{\dot\theta}$ determines the change in velocity due to impact (see [13, 11] for a detailed discussion).

Example 1. *The model for the bipedal robot AMBER 2 considered in this paper is a special case of* (2)*, with the parameters, e.g., masses, lengths and inertias, determined from a high-fidelity SolidWorks model. In particular, AMBER 2 is a 2D 7-link bipedal robot with feet (additional details regarding the development of the model can be found in [18]). For this initial study, a simplified flat-footed gait will be assumed, resulting in a 6 degree of freedom 12-state model. The coordinates of $\mathcal{Q}_R \subset \mathbb{R}^6$ are denoted by $\theta = (\theta_{sa}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk}, \theta_{nsa})^T$ where, as illustrated in Fig. 2b, θ_{sa} is the angle of the stance ankle, θ_{sk} is the angle of the stance knee, θ_{sh} is the angle of the torso with the stance thigh, θ_{nsh} is the angle of the non-stance thigh with the torso, and θ_{nsk} is the angle of the non-stance (or swing) knee, and θ_{nsa} is the angle of the non-stance ankle (or foot). Since AMBER 2 is fully actuated, based upon this choice of coordinates $B = I_6 \in \mathbb{R}^{6\times 6}$. Note that we will take $U_R = \mathbb{R}^6$ for AMBER 2, and impose physical constraints that will restrict the admissible inputs to a subset of U_R as will be discussed at the end of this section.*



(a) Sideview of AMBER 2. (b) Joint angles. (c) Outputs.

Figure 2: The bipedal robot AMBER 2 (a), the joint angles (b) and outputs (c).

Walking Gaits. To discuss physical constraints, it is necessary to consider solutions to the hybrid system obtained by applying feedback control to $\mathscr{HC}_R$. In particular, suppose a feedback controller $u(\theta,\dot\theta)$ is applied to (2) with the end result being a hybrid system $\mathscr{H}_R$. Let $\chi^{\mathscr{H}_R}(\theta_0,\dot\theta_0)$ be a solution to this system with initial condition $(\theta_0,\dot\theta_0)$ and $c_i(t) = (\theta_i(t),\dot\theta_i(t))$. In addition, let $p^x_{\text{com}}(\theta)$ be the horizontal position of the center of mass of the robot, $\dot p^x_{\text{com}}(\theta,\dot\theta)$ the velocity of the forward position of the center of mass, and $p^y_{\text{com}}(\theta)$ be the vertical position of the center of mass. Then we formalize the following notion of a *walking gait* for a bipedal robot.[1]

Definition 1. *A solution $\chi^{\mathscr{H}_R}(\theta^-,\dot\theta^-)$ to $\mathscr{H}_R$ is a* **walking gait** *if $(\theta^-,\dot\theta^-) \in S_R$ and there exist*[2] *constants $\tau_{\min} > 0$ and $p^{\min}_{\text{com}} > 0$ such that*

$$\tau_{i+1} - \tau_i > \tau_{\min} \qquad \text{(Dwell Time)}$$
$$\dot p^x_{\text{com}}(\theta_i(\tau_{i+1}),\dot\theta_i(\tau_{i+1})) > 0 \qquad \text{(Progress)}$$
$$\min_{t\in I_i} p^y_{\text{com}}(\theta_i(t)) > p^{\min}_{\text{com}} \qquad \text{(Upright)}$$

for all $i \in \mathbb{N}$. A walking gait is **stable** *if for all $\gamma > 0$ there exists a $\delta > 0$ such that all solutions $\chi^{\mathscr{H}_R}(\theta_0,\dot\theta_0)$ with $(\theta_0,\dot\theta_0) \in B_\delta(\theta^-,\dot\theta^-) \cap S_R$ satisfy:*

$$(\theta_i(\tau_{i+1}),\dot\theta_i(\tau_{i+1})) \in B_\gamma(\theta^-,\dot\theta^-) \cap S_R \qquad \text{(Stable)}$$

and are walking gaits.

[1] We define a ball of radius $\delta > 0$ around a point x by $B_\delta(x) = \{y \in \mathcal{D}_R : \|x - y\| < \delta\}$.

[2] These constants are practically defined by the user based upon the desired behavior of the walking gait.

Note that the given conditions for a walking gait are neither necessary nor sufficient for walking in general. Rather, they are used to single out "desired" features in a walking gait. In this case, the conditions ensure that there are no instantaneous transitions (e.g., no Zeno behavior [15]), guarantee that the center of mass makes forward progress via a velocity condition, and ensure that the robot is sufficiently upright (as defined by the user through $p_{\mathrm{com}}^{\min}$). Finally, note that due to the fact that we are only considering infinite solutions, inherent in this definition is the notion that the walking gait is indefinite.

Physical Specifications. The following physical specifications are required of a walking gait $\chi^{\mathscr{H}_R}$:

Torque Bounds: From the definition of a hybrid control system, any input belonging to U_R is allowed, yet we explicitly indicate the performance limitations of actuators through an additional physical constraint on a walking gait given by:

$$\sup_{t\in I_i,\ i\in\mathbb{N}} \|u(\theta_i(t),\dot\theta_i(t))\|_\infty < u_{\max}, \tag{C1}$$

where u is the control law that generated the gait and $u_{\max}$ is the maximum joint torque achievable by all of the actuators (this is assumed to be a uniform number for the sake of simplicity).

Velocity Bounds: As with torque bounds, we require that the maximum joint velocity stays under a maximum value, $\dot\theta_{\max}$, as specified by the actuators. In particular:

$$\sup_{t\in I_i,\ i\in\mathbb{N}} \|\dot\theta_i(t)\|_\infty < \dot\theta_{\max}. \tag{C2}$$

ZMP Constraints[3]*:* We require that the feet remain flat during a walking gait, which results in zero moment point (ZMP) type constraints [29, 11]. If $\eta_{st}(\theta)$ is the position and orientation of the stance foot with respect to a fixed inertial frame, then ZMP conditions are determined by viewing η_{st} as a holonomic constraint. In particular, if $J_{\eta_{st}}$ is the jacobian of η_{st}, then the forces and moments at the stance foot are given by:

$$F_{st}(\theta,\dot\theta,u) = \tag{7}$$
$$(J_{\eta_{st}}D(\theta)^{-1}J_{\eta_{st}}^T)^{-1}(J_{\eta_{st}}D(\theta)^{-1}(C(\theta,\dot\theta)-Bu)-\dot J_{\eta_{st}}\dot\theta),$$

where $F_{st}\in\mathbb{R}^3$ for 2D walking robots and $F_{st}\in\mathbb{R}^6$ for 3D walking robots. Conditions so that the foot does not roll during walking can be expressed through inequality constraints of the form:

$$\sup_{t\in I_i,\ i\in\mathbb{N}} A_{\mathrm{ZMP}}F_{st}(\theta_i(t),\dot\theta_i(t),u(\theta_i(t),\dot\theta_i(t)))<0, \tag{C3}$$

where A_{ZMP} depends on the physical parameters of the feet of the robot.

Foot Height Constraints: To achieve walking gaits on physical bipedal robots, it is important that the swing foot does not "scuff" before the end of a step. We introduce, therefore, foot scuffing constraints. Let $h(\theta)$ be the height (y position) of the non-stance (swing) foot, then we require that:

$$\tau_{i+1}=\inf\{t>\tau_i:\ h(\theta_i(t))=0\}. \tag{C4}$$

Note that this condition prevents the case when the foot "scuffs" the ground ($h(\theta)=0$) before reaching the guard (where $\dot h(\theta,\dot\theta)<0$). As a result, this is a stronger condition that the one imposed on solutions (and, specifically, (ii)). We also note that such a τ_{i+1} must exist due to the dwell time condition in Definition 1.

Definition 2. *A walking gait* $\chi^{\mathscr{H}_R}$ *is* **physically realizable** *if it satisfies constraints* (C1)-(C4)*.*

Example 2. *For AMBER 2, based upon the specifications of its actuators coupled with appropriate factors of safety,* $u_{\max}=8\ Nm$ *and* $\dot\theta_{\max}=5\ rad/s$*. For the ZMP constraints,* $F_{st}=(F_{st}^{fx},F_{st}^{fy},F_{st}^{mz})$ *(see [11]) since AMBER 2 is a 2D robot, and the ZMP constraints are determined by:*

$$A_{\mathrm{ZMP}}=\begin{bmatrix}0 & -l_h & -1\\ 0 & -l_t & 1\end{bmatrix},$$

where l_t *and* l_h *are the length to the toe from the ankle and heel from the ankle, respectively.*

3. DIMENSION REDUCTION THROUGH CONTROL

To achieve dimensionality reduction in bipedal robots, we will utilize the notion of a *virtual constraint* [2, 11, 30]. In particular, we will consider relative degree 2 outputs wherein the goal is to drive these outputs to zero. This will restrict the full-order dynamics of the robot to a low-dimensional surface—termed the *partial zero dynamics*—wherein the evolution of the system may be dictated by relative degree 1 outputs. Through pre-feedback control laws, the dynamics of the relative degree 1 outputs defines a 2-dimensional linear hybrid control system. We will demonstrate that solutions of this reduced-order hybrid system yield solutions to the full-order dynamics.

Virtual Constraints. Consider virtual constraints (or outputs) of the following form [2, 11]:

$$\begin{aligned} y_1(\theta,\dot\theta,v) &= y_{a,1}(\theta,\dot\theta)-v, & (8)\\ y_2(\theta,\alpha) &= y_{a,2}(\theta)-y_{d,2}(\rho(\theta),\alpha), & (9)\end{aligned}$$

where y_1 and y_2 will be chosen so that they are relative degree 1 and (vector) relative degree 2, respectively. In this case, $y_{a,1}(\theta,\dot\theta)$ is the "actual" velocity-based output and v is the "desired" velocity. In the following, we will view v as the control input to the system (after pre-feedback), and we synthesize v through formal methods. Similarly, $y_{a,2}(\theta)$ is the actual vector of outputs that modulate the posture of the robot and $y_{d,2}(\rho(\theta),\alpha)$ gives the desired evolution of the associated configuration variables as dictated by parameters α in the desired evolution of the virtual constraints and a parameterization of time $\rho(\theta)$.

For the sake of simplicity, we will assume that the virtual constraints have a linear structure, namely

$$\begin{aligned} y_{a,1}(\theta,\dot\theta) &= c\dot\theta\\ y_{a,2}(\theta) &= H\theta\end{aligned} \qquad \text{s.t.} \qquad \mathrm{rank}\left(\begin{bmatrix}c\\ H\end{bmatrix}\right)=n, \tag{10}$$

and that $\rho(\theta)=c\theta-c\theta^+$, where θ^+ is the configuration at the beginning of a step, and will be specified later. The goal is to drive both the velocity and posture modulating outputs to zero, i.e., $y_1\to 0$ and $y_2\to 0$.

Example 3. *In the case of AMBER 2, the virtual constraints considered in this paper are illustrated in Fig. 2c.*

[3] Note that we could also consider friction conditions in a similar fashion but will omit doing so for brevity.

In particular, as discussed in [18], $y_{a,1}$ is the linearized velocity of the hip, v is the desired velocity of the hip, $y_{a,2}$ consists of a vector of configuration based functions and $y_{d,2}$ is the time solution to a linear mass-spring-damper system parameterized by the linearized position of the hip.

Pre-Feedback Control. To set the stage for obtaining the reduced order dynamics that will be used to synthesize controllers for the system, we begin by applying a "pre-feedback" controller based upon feedback linearization [25] (note that this controller is a slight modification of the controller presented in [2, 3]). With the goal of driving $y_2 \to 0$ and shaping the dynamics of y_1 to be that of a linear system with control input v, consider the feedback controller:

$$u_{\mathrm{FB}}^{(\alpha,\varepsilon)}(\theta,\dot\theta) = -\mathcal{A}^{-1}(\theta,\dot\theta)\left(\begin{bmatrix} 0 \\ L_{f_R}L_{f_R}y_2(\theta)\end{bmatrix}\right. \tag{11}$$
$$\left. + \begin{bmatrix} L_{f_R}y_{a,1}(\theta,\dot\theta) \\ 2\frac{1}{\varepsilon}L_{f_R}y_2(\theta,\dot\theta)\end{bmatrix} + \begin{bmatrix} \frac{1}{\varepsilon}y_1(\theta,\dot\theta) \\ \frac{1}{\varepsilon^2}y_2(\theta,\alpha)\end{bmatrix}\right),$$

with control gain $\varepsilon > 0$ and decoupling matrix:

$$\mathcal{A}(\theta,\dot\theta) = \begin{bmatrix} L_{g_R}y_{a,1}(\theta,\dot\theta) \\ L_{g_R}L_{f_R}y_2(\theta,\dot\theta,\alpha)\end{bmatrix}. \tag{12}$$

Here L denotes the Lie derivative [25], and we assume that the decoupling matrix is invertible. It follows that $u_{\mathrm{FB}}^{(\alpha,\varepsilon)}(\theta,\dot\theta)$ results in dynamics on the outputs given by:

$$\dot y_{a,1} = -\frac{1}{\varepsilon}y_1 \tag{13}$$
$$\ddot y_2 = -2\frac{1}{\varepsilon}\dot y_2 - \frac{1}{\varepsilon^2}y_2 \tag{14}$$

and therefore, for a control gain $\varepsilon > 0$, the control law $u_{\mathrm{FA}}^{(\alpha,\varepsilon)}$ renders the outputs exponentially stable [25]. That is, in the case when v is a constant (and hence $\dot y_{a,1} = \dot y_1$), the virtual constraints $y_1 \to 0$ and $y_2 \to 0$ exponentially at a rate of $\frac{1}{\varepsilon}$.

Partial Hybrid Zero Dynamics. While the introduced controller drives $y_1 \to 0$ and $y_2 \to 0$, we want to be able to modulate the relative 1 degree output (through v), while forcing the relative 2 degree output to remain zero and hence form an invariant surface. This motivates the introduction of the *partial zero dynamics surface* [2]:

$$\mathbf{PZ}_\alpha = \{(\theta,\dot\theta)\in T\mathcal{Q}_R : y_2(\theta,\alpha)=0,\ \ L_{f_R}y_2(\theta,\dot\theta,\alpha)=0\}. \tag{15}$$

We say that the hybrid system (2) has *partial hybrid zero dynamics (PHZD)*[4] if:

$$\Delta_R(S_R\cap\mathbf{PZ}_\alpha)\subset\mathbf{PZ}_\alpha. \tag{PHZD}$$

In particular, we can choose the parameters α so that this holds through an optimization of the form:

$$\alpha^* = \underset{\alpha\in\mathbb{R}^{5(n-1)}}{\operatorname{argmin}}\ \mathrm{Cost}_{\mathrm{HD}}(\alpha) \tag{16}$$
$$\text{s.t.}\quad \Delta_R(S_R\cap\mathbf{PZ}_\alpha)\subset\mathbf{PZ}_\alpha. \tag{17}$$

Note that the cost can be chosen based upon the specific objective of interest, i.e., minimizing the cost of transport, and that this optimization can be stated only in terms of the parameters α through the constructions introduced in [2].

The notion of PHZD allows for the construction of a hybrid system model for the reduced order dynamics defined by the surface (15). In particular, we reformulate the constructions in [30] in a way applicable to full-actuation [2]. Because of the specific form of $y_{a,1}$ in (10) due to the linear output assumption, we begin by picking the following coordinates for the partial zero dynamics surface:

$$z_1 = c\theta \tag{18}$$
$$z_2 = y_{a,1}(\theta,\dot\theta) = c\dot\theta$$

where $c\in\mathbb{R}^{1\times n}$ as introduced in (10). As a result of the fact that we have full actuation and have completely linearized the dynamics with (11), the relative degree 1 output (8) evolves according to (13). Therefore, the partial hybrid zero dynamics evolve according to the linear ODE:

$$\dot z_1 = z_2 \tag{19}$$
$$\dot z_2 = -\frac{1}{\varepsilon}(z_2 - v).$$

where $v\in\mathbb{R}$ is viewed as a control input. The end result is, therefore, a linear control system:

$$\dot z = A_{\mathbf{PZ}}z + B_{\mathbf{PZ}}v \tag{20}$$

with $A_{\mathbf{PZ}}$ and $B_{\mathbf{PZ}}$ obtained from (19).

Impact Configurations. It is important to note that the proper choice of parameters α that determine the partial zero dynamics surface determine the configuration of the robot at impact (foot strike). In particular, the configuration of the robot at impact, θ^-, is determined by the following requirement:

$$\theta^- = \theta\quad \text{s.t.}\quad \begin{bmatrix} y_2(\theta,\alpha) \\ h(\theta)\end{bmatrix} = \begin{bmatrix} 0 \\ 0\end{bmatrix}. \tag{21}$$

By virtue of the form of the relabeling matrix, Δ_θ, it follows that if $h(\theta^-)=0$ then $h(\theta^+)=0$ for $\theta^+ = \Delta_\theta\theta^-$. Moreover, from the fact that $y_2(\theta,\alpha)\in\mathbb{R}^{n-1}$, for the proper choice of relative degree 2 outputs and parameters α, (21) has at least two solutions for $\theta\in\mathcal{Q}_R$[5]. Since the additional solutions are points where the foot scuffs the ground, i.e., points where $h \geq 0$, we make the following assumption:

Assumption 1. *Let α be parameters solving the optimization problem* (16) *and therefore guarantee* (PHZD). *Furthermore, assume that θ^- and $\theta^+ = \Delta_\theta\theta^-$ are the only two points in $\mathcal{Q}_R$ satisfying* (21).

Reduced Order Hybrid Dynamics. The advantage of the partial zero dynamics representation is that it yields a reduced-order hybrid system representation that dictates the behavior of the full order dynamics of the system. We will explicitly construct this hybrid system, and establish properties of its solutions relative to solutions of the full-order hybrid system.

Pick, once and for all, parameters α solving the optimization problem (16) and a point θ^- satisfying Assumption 1 with $\theta^+ = \Delta_\theta\theta^-$. We can therefore compute $z_1^- = c\theta^-$ and $z_1^+ = c\theta^+$. From this, since (PHZD) is satisfied, the discrete

[4] This formulation is based upon the notion of *hybrid zero dynamics* for underactuated bipedal robots [31].

[5] Note that, due to the trigonometric functions that yield h, it may be necessary to consider subsets of the configuration space $\mathcal{Q}_R$ that limit the number of solutions to (21).

change in z_1 and z_2 can be determined via [2, 31]:

$$\begin{aligned} z_1^+ &= c\Delta_\theta \theta^- \\ z_2^+ &= \Delta_{\mathbf{PZ}}(\theta^-) z_2^- \end{aligned} \tag{22}$$

where θ^- is a point that is chosen *a priori* and

$$\Delta_{\mathbf{PZ}}(\theta^-) := c\Delta_{\dot{\theta}}(\theta^-)\Psi_{\mathbf{PZ}}(c\theta^-). \tag{23}$$

This defines a *linear* 2-dimensional hybrid control system:

$$\mathscr{HC}_{\mathbf{PZ}} = (\mathcal{D}_{\mathbf{PZ}}, U_{\mathbf{PZ}}, S_{\mathbf{PZ}}, \Delta_{\mathbf{PZ}}, f_{\mathbf{PZ}}, g_{\mathbf{PZ}}). \tag{24}$$

where the domain and guard are given by:

$$\mathcal{D}_{\mathbf{PZ}} = \{z \in \mathbb{R}^2 : z_1^+ \leq z_1 \leq z_1^-\}, \tag{25}$$

$$S_{\mathbf{PZ}} = \{z \in \mathbb{R}^2 : z_1 = z_1^-\}. \tag{26}$$

We will not (initially) restrict the control input v and, therefore, $U_{\mathbf{PZ}} = \mathbb{R}$. The reset map, $\Delta_{\mathbf{PZ}}$, is a linear transformation as given by (22). Finally, the control system has $f_{\mathbf{PZ}}(z) = A_{\mathbf{PZ}} z$ and $g_{\mathbf{PZ}}(z) = B_{\mathbf{PZ}}$.

PHZD Reconstruction. We can use the relationship between the reduced order PHZD and the full-order dynamics, as afforded by the feedback control law, to reconstruct the full-order state of the system. Note that since z_1 is directly related to the parameterization of time, we can write $y_{d,2}(z_1) = y_{d,2}(\rho(\theta), \alpha)$ wherein it was assumed that we are working with a fixed parameter set α. Therefore, defining

$$\begin{aligned} \Phi_{\mathbf{PZ}}(z_1) &= \begin{bmatrix} c \\ H \end{bmatrix}^{-1} \begin{pmatrix} z_1 \\ y_{d,2}(z_1) \end{pmatrix} \\ \Psi_{\mathbf{PZ}}(z_1) &= \begin{bmatrix} c \\ H \end{bmatrix}^{-1} \begin{pmatrix} 1 \\ \frac{\partial y_{d,2}(z_1)}{\partial z_1} \end{pmatrix} \end{aligned} \tag{27}$$

it follows that:

$$\begin{aligned} \vartheta_r(z) &:= \Phi_{\mathbf{PZ}}(z_1) \\ \dot{\vartheta}_r(z) &:= \Psi_{\mathbf{PZ}}(z_1) z_2 \end{aligned} \quad \Rightarrow \quad (\vartheta_r(z), \dot{\vartheta}_r(z)) \in \mathbf{PZ}_\alpha \tag{28}$$

with $z = (z_1, z_2)^T$. Note that if we pick coordinates $\eta = (y_2, \dot{y}_2)$, since y_2 is a relative degree 2 output it follows that there is a diffeomorphism $\Pi : (\theta, \dot{\theta}) \to (\eta, z)$. We also note that there is the canonical embedding $\iota_{\mathbf{PZ}} : \mathcal{D}_{\mathbf{PZ}} \to \mathcal{D}_R$ given by $\iota_{\mathbf{PZ}}(z) = \Pi^{-1}(0, z)$.

Key Properties. We now establish the key properties of hybrid systems obtained through PHZD as they relate to the full order dynamics of hybrid systems. First, suppose that we have a feedback control law $v(z)$ that is applied to the hybrid control system $\mathscr{HC}_{\mathbf{PZ}}$ with the end result being a hybrid system $\mathscr{H}_{\mathbf{PZ}}$. The application of this control law in (11) via y_1 (which depends on $v(z)$) to $\mathscr{HC}_R$, yields a hybrid system $\mathscr{H}_R$.

Theorem 1. *Let $\chi^{\mathscr{H}_{\mathbf{PZ}}} = (\mathcal{I}, \mathcal{Z})$, with $\mathcal{Z} = \{z_i\}_{i \in \mathbb{N}}$, be a solution to $\mathscr{H}_{\mathbf{PZ}}$ with $\tau_{i+1} - \tau_i \geq \tau_{\min}$. If $\chi^{\mathscr{H}_R} = (\mathcal{I}, \mathcal{C}_r)$, with $\mathcal{C}_r = \{c_i^r\}_{i \in \mathbb{N}}$ where $c_i^r(t) = (\vartheta_r(z_i(t)), \dot{\vartheta}_r(z_i(t)))$, satisfies* (Progress), (Upright), *and* (C1)-(C3), *then $\chi^{\mathscr{H}_R}$ is a physically realizable walking gait of $\mathscr{H}_R$.*

Proof. We need only verify that if $\chi^{\mathscr{H}_{\mathbf{PZ}}}$ is a solution to $\mathscr{H}_{\mathbf{PZ}}$ then $\chi^{\mathscr{H}_R}$ is a solution to $\mathscr{H}_R$; the remaining statements then follow from the fact that (Progress), (Upright), and (C1)-(C3) are satisfied for the reconstructed solution: $c_i^r(t) = (\vartheta_r(z_i(t)), \dot{\vartheta}_r(z_i(t)))$. To establish that $\chi^{\mathscr{H}_R}$ is a solution to $\mathscr{H}_R$, we must verify both the continuous and discrete conditions on a solution to a hybrid system.

The continuous conditions on $\chi^{\mathscr{H}_R}$ are given by the requirement that $\dot{c}_i^r(t) = f_{\mathrm{cl}}(c_i^r(t))$, where f_{cl} is the closed-loop dynamics obtained by applying $v(z)$ to (3) via (19) and (11). Since the initial condition $(\vartheta_r(z_0(0)), \dot{\vartheta}_r(z_0(0))) \in \mathbf{PZ}_\alpha$, and because the dynamics in the η and z coordinates evolve in a decoupled fashion according to (13) and (14), we need only verify that $y_2(\vartheta_r(z_i(t))) = 0$ and $\dot{y}_2(\vartheta_r(z_i(t)), \dot{\vartheta}_r(z_i(t))) = 0$ for all $t \in I_i$ and $i \in \mathbb{N}$. This follows from the construction of ϑ_r and $\dot{\vartheta}_r$ and, specifically, (27) and (28).

The discrete conditions on $\chi^{\mathscr{H}_R}$ are given by **(i)**, **(ii)** and **(iii)** as stated in Section 2. Condition **(i)** is satisfied by Assumption 1 (which also implies (C4)), i.e., the boundary of the domain $\mathcal{D}_R$ cannot be reached until the guard S_R is reached, and the configuration in which the guard is reached is given by θ^-. Similarly, **(ii)** is satisfied again because the first configuration where the guard is reached is θ^-, $z_1^- = c\theta^-$ and τ_{i+1} satisfies **(ii)** for $\chi^{\mathscr{H}_{\mathbf{PZ}}}$. Finally, **(iii)** is satisfied again through the formulation of the reset map (23). □

The next result is an extension of Theorem 1 that will allow for the stability of walking gaits to be established in Section 4. To establish this, we will utilize a notion of distance between a set and solutions. In particular, given a set S and an execution $\chi^{\mathscr{H}_R}$:

$$\mathrm{dist}(\chi^{\mathscr{H}_R}, S) = \inf_{t \in I_i, i \in \mathbb{N}, x \in S} \|c_i(t) - x\|. \tag{29}$$

Theorem 2. *Let $P \subset \mathcal{D}_{\mathbf{PZ}}$ be an invariant set of $\mathscr{HC}_{\mathbf{PZ}}$ under a feedback control law $v(z)$. Then $\iota_{\mathbf{PZ}}(P) \subset \mathbf{PZ}_\alpha$, and for any $\gamma > 0$ there exists a $\delta > 0$ such that for $\|\eta_0\| < \delta$ any walking gait $\chi^{\mathscr{H}_R}(\eta_0, z_0)$ with $z_0 \in P$ satisfies:*

$$\mathrm{dist}(\chi^{\mathscr{H}_R}, \iota_{\mathbf{PZ}}(P)) < \gamma.$$

Proof. To establish this result, we note that

$$\mathrm{dist}(\chi^{\mathscr{H}_R}, \iota_{\mathbf{PZ}}(P)) \leq \inf_{x \in \iota_{\mathbf{PZ}}(P)} \|c_i(t) - x\| \tag{30}$$

for arbitrary $t \in I_i$ and $i \in \mathbb{N}$. Moreover, since $\iota_{\mathbf{PZ}}$ is the canonical embedding $\iota_{\mathbf{PZ}}(z) = \Pi^{-1}(0, z)$ and because the dynamics of the system evolve in a decoupled fashion according to (13) and (14), it follows that the only nontrivial component of the distance will be contributions from the y_2 and $\dot{y}_2$ dynamics. Formally, writing $(\eta_i(t), z_i(t)) = \Pi(c_i(t)) = \Pi(\theta_i(t), \dot{\theta}_i(t))$, it follows that

$$\begin{aligned} \inf_{x \in \iota_{\mathbf{PZ}}(P)} \|c_i(t) - x\| &\leq \|\eta_i(t)\| = \left\| \begin{matrix} y_2^i(t) \\ \dot{y}_2^i(t) \end{matrix} \right\| \\ &\leq \underbrace{\frac{\lambda_1}{\varepsilon} e^{-\frac{\lambda_2}{\varepsilon}\tau_{\min}}}_{\beta(\varepsilon)} \left\| \begin{matrix} y_2^i(\tau_i) \\ \dot{y}_2^i(\tau_i) \end{matrix} \right\| \end{aligned} \tag{31}$$

for some constants $\lambda_1, \lambda_2 > 0$, where the second inequality follows from [4] since y_2 and $\dot{y}_2$ evolve according to the linear system (14), picking $t = \tau_{i+1}$, and utilizing the dwell time assumption: $\tau_{i+1} - \tau_i > \tau_{\min}$.

To understand the role of the discrete dynamics, we note that in the coordinates (η, z) we can decompose the reset map as follows: $\Delta_R \circ \Pi^{-1}(\eta, z) = (\Delta_\eta(\eta, z), \Delta_z(\eta, z))$. The assumption of partial hybrid zero dynamics (PHZD) implies that $\Delta_\eta(0, z) = 0$. By assumption Δ_η is Lipschitz continuous with Lipschitz constant L_{Δ_η}, wherein:

$$\|\Delta_\eta(\eta, z)\| = \|\Delta_\eta(\eta, z) - \Delta_\eta(0, z)\| \leq L_{\Delta_\eta}\|\eta\|.$$

Combining this with (30) and (31) implies that:

$$\mathrm{dist}(\chi^{\mathscr{H}_R}, \iota_{\mathbf{PZ}}(P)) \leq \beta(\varepsilon)^{i+1} L^i_{\Delta_\eta} \delta \tag{32}$$

for any $i \in \mathbb{N}$. Since $\beta(\varepsilon) \to 0$ as $\varepsilon \to 0$, there exists an $\overline{\varepsilon} > 0$ such that $\beta(\varepsilon) < \frac{1}{L_{\Delta_\eta}}$ for all $0 < \varepsilon < \overline{\varepsilon}$. Therefore, picking $\delta < \gamma L_{\Delta_\eta}$ yields the desired result. □

4. ABSTRACTION BASED CONTROLLER SYNTHESIS

In this section, we show how to synthesize an abstraction based controller for the linear hybrid system $\mathscr{HC}_{\mathbf{PZ}}$, defined in (24), enforcing the desired specifications by construction. The techniques to be employed are described in [32] (see also [27] for an introduction to abstraction based controller synthesis) and were developed for discrete-time systems. Hence, we start by defining discrete-time executions for a hybrid system $\mathscr{H}$.

Let $t_s \in \mathbb{R}^+$ be a sampling time and let $\chi^{\mathscr{H}} = (\mathcal{I}, \mathcal{C})$ be a hybrid execution of hybrid system $\mathscr{H}$. A discrete-time hybrid execution of $\mathscr{H}$ with sampling time t_s, denoted by $\chi^{\mathscr{H}}_d = (\mathcal{I}_d, \mathcal{C}_d)$, is given by a collection of time intervals $\mathcal{I}_d = \{I_{d,i}\}_{i \in \mathbb{N}}$ where $I_{d,i} = \{\tau_i, \tau_i + t_s, \tau_i + 2t_s, \ldots, \tau_{i+1}\}$, and by a collection of functions $\mathcal{C}_d = \{c_{d,i}\}_{i\in\mathbb{N}}$ where each $c_{d,i} : I_{d,i} \to \mathcal{D}$ is given by the restriction of c_i to the set $I_{d,i}$.

The starting point for abstraction based controller synthesis is the construction of a finite-state abstraction $S(\mathscr{HC}_{\mathbf{PZ}})$ of $\mathscr{HC}_{\mathbf{PZ}}$ by following the methods in [32]. This abstraction comes equipped with an ε-approximate alternating simulation relation from $S(\mathscr{HC}_{\mathbf{PZ}})$ to $\mathscr{HC}_{\mathbf{PZ}}$ guaranteeing that any controller synthesized for $S(\mathscr{HC}_{\mathbf{PZ}})$ can be refined to a controller for $\mathscr{HC}_{\mathbf{PZ}}$ resulting in the same closed-loop behavior up to an error of $\varepsilon \in \mathbb{R}^+$. In other words, let us denote by $S(\mathscr{H}_{\mathbf{PZ}})$ the hybrid system resulting from composing $S(\mathscr{HC}_{\mathbf{PZ}})$ with a controller and let us denote by $\mathscr{H}_{\mathbf{PZ}}$ the hybrid system resulting from composing $\mathscr{HC}_{\mathbf{PZ}}$ with the refined controller. Then, for every discrete-time hybrid execution $\chi_d^{S(\mathscr{H}_{\mathbf{PZ}})}$ any corresponding discrete-time hybrid execution $\chi_d^{\mathscr{H}_{\mathbf{PZ}}}$ satisfies:

$$\mathrm{dist}\left(\chi_d^{S(\mathscr{H}_{\mathbf{PZ}})}, \chi_d^{\mathscr{H}_{\mathbf{PZ}}}\right) \leq \varepsilon.$$

Moreover, ε is a design parameter that can be made as small as desired, at the expense of a larger finite-state abstraction.

Convexity of Reachable Sets. The key technical assumption required for the results in [32] is the possibility of computing an over-approximation of the reachable set of $\mathscr{HC}_{\mathbf{PZ}}$. Hence, we describe in this section how this can be efficiently done. We start by recalling a few notions.

A vector $x \in \mathbb{R}^n$ is a convex combination of m vectors $x_1, \ldots, x_m \in \mathbb{R}^n$ if x can be written as

$$x = \sum_{i=1}^m \lambda_i x_i, \quad \lambda_i \geq 0, \quad \sum_{i=1}^m \lambda_i = 1. \tag{33}$$

A set is convex if it contains the convex combination of its elements. A point $x \in B$ is called an extreme point of a compact convex set $B \subset \mathbb{R}^n$ if it cannot be represented by a convex combination of any two points $x_1, x_2 \in B$, with $x_1 \neq x$ and $x_2 \neq x$. The convex hull of a set $B \subseteq \mathbb{R}^n$ is the set of all convex combinations of points in B and is denoted $\mathbf{conv}(B)$. It follows that any compact convex set is the convex hull of its extreme points.

Definition 3. *The set reached by the trajectories of (19) from $B \subseteq \mathcal{D}_{\mathbf{PZ}}$ in time $t_s \in \mathbb{R}_0^+$ under constant input v is denoted by $\mathcal{R}_v^{t_s}(B)$ and defined as:*

$$\mathcal{R}_v^{t_s}(B) = \left\{ z' \in \mathbb{R}^2 \mid z(0) \in B \wedge z(t_s) = z' \right\},$$

where $z(t)$ is a solution to (19) with the constant input v and initial condition $z(0)$. Moreover, we define $\mathcal{R}_v(B)$ by:

$$\mathcal{R}_v(B) = \cup_{t_s \in \mathbb{R}_0^+} \mathcal{R}_v^{t_s}(B).$$

To simplify notation we write $\mathcal{R}_v(x)$ rather than $\mathcal{R}_v(\{x\})$ when B is the singleton $\{x\}$.

We now show that the intersection of $\mathcal{R}_v(B)$ with the guard set $S_{\mathbf{PZ}}$ is a convex set.

Theorem 3. *Consider the convex set $B \subseteq \mathbb{R}^2$ defined by:*

$$B = [a_1, b_1] \times [a_2, b_2],$$

with $a_1, a_2, b_1, b_2 \in \mathbb{R}$, $a_1 < b_1$, and $a_2 < b_2$. Denote by $\hat{z}_1, \ldots, \hat{z}_4 \in B$ the extreme points of B. If the linear dynamics (19) satisfies $\dot{z}_1 \geq c$ on $\mathcal{D}_{\mathbf{PZ}}$ for some $c > 0$, then:

$$\mathcal{R}_v(B) \cap S_{\mathbf{PZ}} = \mathbf{conv}\{\mathcal{R}_v(\hat{z}_1) \cap S_{\mathbf{PZ}}, \ldots, \mathcal{R}_v(\hat{z}_4) \cap S_{\mathbf{PZ}}\}.$$

Proof. In this proof we denote by $F_t : \mathbb{R}^2 \to \mathbb{R}^2$ flow of the linear system (19) with constant input v, i.e., if $z(t)$ is the solution of (19) with initial condition z' and constant input v then $F_t(z') = z(t)$. We will also say that (19) is transversal to the boundary of B at $z' \in B$ when $\dot{z}_1 \neq 0$ for z' belonging to the vertical boundaries and when $\dot{z}_2 \neq 0$ for z' belonging to the horizontal boundaries.

We first state and prove two facts.

Fact 1: *If z belongs to the boundary of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$, then z is the image under F_t of a boundary point of B for some $t \in \mathbb{R}$.*

It suffices to show that F_t maps interior points of B to interior points of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$. But this follows directly from the fact that F_{-t}, the inverse of F_t, exists and is continuous. The inverse image of an open set by a continuous map is an open set. Hence, let $O' \subseteq B$ be an open set containing a point z' in the interior of B and let t_s satisfy $F_{t_s}(z') = z \in S_{\mathbf{PZ}}$. Since $\dot{z}_1 \geq c > 0$ and the guard is given by $z_1 = z_1^-$, t_s exists. Then the set $O = F_{-t_s}(O')$ is an open set containing the point z in $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$. Since $O \subseteq \mathcal{R}_v(B)$ and $O \cap S_{\mathbf{PZ}}$ is open in the topology induced on $S_{\mathbf{PZ}}$ by the standard topology in $\mathbb{R}^2$, z is an interior point of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$.

Fact 2: *If z belongs to the boundary of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$, then z cannot be the image under F_t of a point in the boundary of B where* (19) *is transversal to the boundary.*

We will show that if z' belongs to the boundary of B and (19) is transversal to the boundary at z', then the flow takes z' to an interior point of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$. Let O' be an open set in the boundary of B containing z'. Consider the set $B' = B \cup F_t(O')$ for sufficiently small t (t is positive if the vector field points to the outside of B and negative otherwise). It is clear that $\mathcal{R}_v(B) = \mathcal{R}_v(B')$. Moreover, we can now take an open (in $\mathbb{R}^2$) subset O of B' containing z'. Applying the argument used to prove Fact 1, we see that F_t takes z' into an interior point of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$.

Proof of Theorem 3: Facts 1 and 2 tell us that boundary points of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$ are the image under the flow of extreme points of B or of boundary points of B where (19) is not transversal. The assumption $\dot{z}_1 \geq c > 0$ implies that the transversality condition can only fail on the horizontal boundaries. Moreover, the dynamics of z_2 given by (19) shows that if (19) is not transversal at a point on a horizontal boundary then all the points in that horizontal boundary are on the same trajectory. We thus conclude that boundary points of $\mathcal{R}_v(B) \cap S_{\mathbf{PZ}}$ are the image under the flow of extreme points of B. Let now $\gamma : [0,1] \to \mathbb{R}^2$ be a continuous curve contained in B and joining the extreme point $\hat{z}_1$ to the extreme point $\hat{z}_2$, i.e., $\gamma(0) = \hat{z}_1$, $\gamma(1) = \hat{z}_2$, and $\gamma(r) \in B$ for $r \in [0,1]$. By continuity of the map $F_t \circ \gamma$ we have $\mathcal{R}_v(\cup_{r\in[0,1]}\{\gamma(r)\}) \cap S_{\mathbf{PZ}} = \mathbf{conv}(\mathcal{R}_v(\hat{z}_1), \mathcal{R}_v(\hat{z}_2))$. Since this argument does not depend on the choice of extreme points, the result follows. $\square$

The abstraction techniques in [32] require the over-approximation of $\mathcal{R}_v^{t_s}(B)$ when the guard is not reached in t_s units of time. In this case the linearity of (19) implies that $\mathcal{R}_v^{t_s}(B)$ is a convex set and can be computed as:

$$\mathbf{conv}(\mathcal{R}_v^{t_s}(\hat{z}_1), \ldots, \mathcal{R}_v^{t_s}(\hat{z}_4)).$$

If the guard can be reached in t_s units of time or less, then we need to over-approximate the set of points that can be reached up to the time the guard is hit and immediately after the reset. This set can be over-approximated by:

$$\begin{aligned}&\left(\mathbf{conv}(\mathcal{R}_v^{t_s}(\hat{z}_1), \ldots, \mathcal{R}_v^{t_s}(\hat{z}_4)) \cap \mathcal{D}_{\mathbf{PZ}}\right)\\ &\cup\Delta_{\mathbf{PZ}}\left(\mathbf{conv}\{\mathcal{R}_v(\hat{z}_1) \cap S_{\mathbf{PZ}}, \ldots, \mathcal{R}_v(\hat{z}_4) \cap S_{\mathbf{PZ}}\}\right). \quad (34)\end{aligned}$$

Once again, all the sets are convex and can thus be efficiently computed since we only have to perform numerical simulations for the vertices of B.

Walking Gait Generation. One of the main advantages of abstraction based control is the possibility to enforce the specifications by construction. In our case, for a stable robot walking gait, there are seven specifications that have to be satisfied: the (Dwell Time), (Progress) and (Upright) constraints in Definition 1, as well as the physical requirements (C1), (C2), (C3), and (C4). For our system, the constraints (Dwell Time), (Upright), and (C4) are enforced by the choice of output functions (8),(9) and the feedback linearizing controller. The (Progress) constraint is automatically satisfied by the dynamics since $\dot{z}_1 = z_2$ and $\mathcal{D}_{\mathbf{PZ}}$ only includes points where z_2 is strictly positive. Therefore, we only have to cater to the physical constraints (C1)-(C3).

We thus synthesize a controller forcing the closed-loop trajectories to remain in the set $P = P_1 \cap P_2 \cap P_3$ for all time where each set P_i describes the constraint (Ci) in the PHZD:

$$P_1 = \{z \in \mathcal{D}_{\mathbf{PZ}} : |u(\vartheta_r(z), \dot{\vartheta}_r(z))| < u_{\max}\}$$

$$P_2 = \{z \in \mathcal{D}_{\mathbf{PZ}} : |\dot{\vartheta}_r(z)| < \dot{\theta}_{\max}\}$$

$$P_3 = \{z \in \mathcal{D}_{\mathbf{PZ}} : A_{\mathrm{ZMP}} F_{st}(\vartheta_r(z), \dot{\vartheta}_r(z), u(\vartheta_r(z), \dot{\vartheta}_r(z))) < 0\}.$$

The over-approximation of the reachable sets based on Theorem 3 has been implemented in the tool Pessoa, see [24]. In order to compute the finite-state abstraction we restrict the set of states to the operating region $Z^{abstr} = [-0.12, 0.13] \times [0, 0.8]$ and the input space to $U^{abstr} = [0.02, 0.8]$. Note that Z^{abstr} contains z_1^- and z_1^+. The abstraction is then computed by dividing the state space Z^{abstr} into boxes B of length 0.005, by quantizing the input space U^{abstr} with a resolution $\mu = 0.005$, and time discretization of $t_s = 0.05$. See [32] for the definition of these parameters and their relation to the finite-state abstraction. For these parameters, the state space is covered by 8211 boxes and we consider 157 different input values. The abstraction can be computed in about four hours on a computer with a 2.4 GHz dual core processor. A controller for this abstraction is found after 4.5 seconds.

Main Result. Recall that we denote by $\mathscr{H}_{\mathbf{PZ}}$ the hybrid system obtained by composing $\mathscr{HC}_{\mathbf{PZ}}$ with the refined controller. This composition restricts the behavior of $\mathscr{HC}_{\mathbf{PZ}}$ in two different ways: by restricting the available inputs, and by restricting the initial conditions. The set of initial conditions is denoted by $\mathcal{D}_{\mathbf{PZ}}^{init}$ and is a subset of $\mathcal{D}_{\mathbf{PZ}}$. It then follows from Proposition 9.4 in [27] that the hybrid executions of $\mathscr{H}_{\mathbf{PZ}}$ starting in $\mathcal{D}_{\mathbf{PZ}}^{init}$ remain in:

$$P^{\varepsilon} = \{z \in \mathcal{D}_{\mathbf{PZ}} \mid \|z - z'\| \leq \varepsilon \text{ for some } z' \in P\},$$

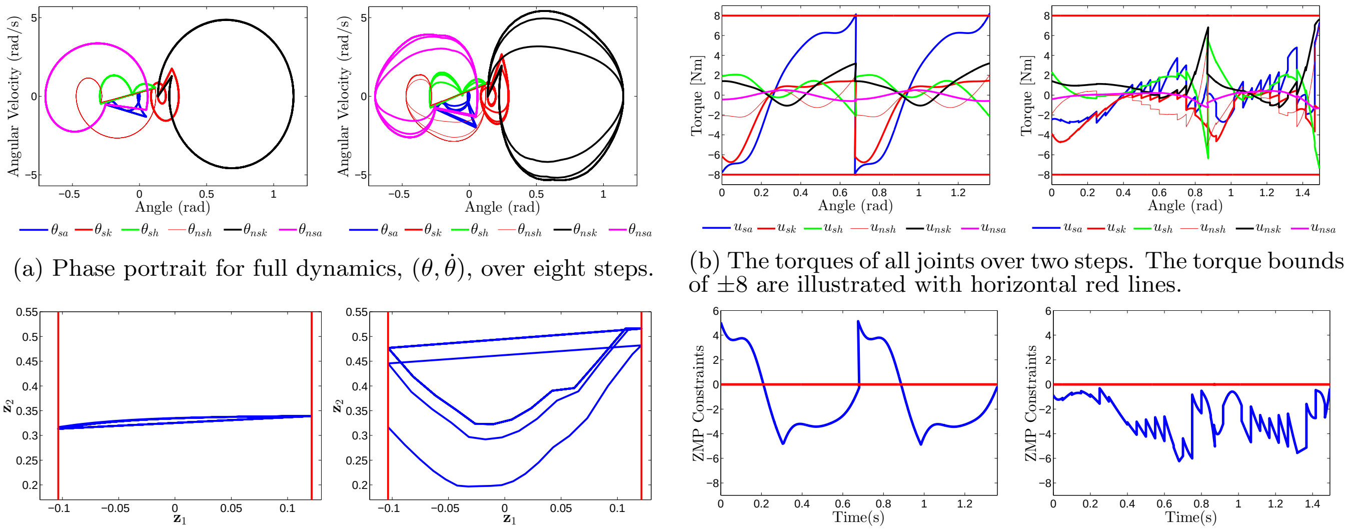
for all time. By bounding the inter-sample behavior using a standard Lyapunov-type argument [19, 21, 14] we conclude that the (continuous-time) hybrid executions of $\mathscr{H}_{\mathbf{PZ}}$ remain in $P^{\delta(\varepsilon,t_s)}$ for all time where δ is a continuous and increasing function of ε and t_s satisfying $\delta(\varepsilon, 0) = 0$. Therefore, we can always find a subset $\overline{P}$ of P and a choice of t_s and ε so that $\overline{P}^{\delta(\varepsilon,t_s)} \subseteq P$. By synthesizing a controller enforcing the stricter constraints defined by $\overline{P}$ we then guarantee that executions remain in P as desired. This discussion, when coupled with Theorem 1 and 2, can be summarized in the following result.

Theorem 4. *Let $\mathscr{H}_{\mathbf{PZ}}$ be the hybrid system resulting from composing $\mathscr{HC}_{\mathbf{PZ}}$ with the controller obtained by refining the controller synthesized for the finite-state abstraction of $\mathscr{HC}_{\mathbf{PZ}}$. Any hybrid execution of $\mathscr{H}_{\mathbf{PZ}}$ starting in $\mathcal{D}_{\mathbf{PZ}}^{init}$ remains in P for all time. Moreover, any hybrid execution of $\mathscr{H}_R$ with initial condition in $\iota_{\mathbf{PZ}}(\mathcal{D}_{\mathbf{PZ}}^{init})$ is a physically realizable stable walking gait.*

5. SIMULATION RESULTS

In this section, we present simulation results for AMBER 2 and compare them against simulation results for a nominal controller that was used to realize walking experimentally on AMBER 2 [18] (a video of the walking gait on the robot can be seen at [1]). In particular, the parameters α solving (16) are taken from [18]; in that paper, a constant control input, $v = v^*$, was utilized to achieve the walking gait. This is in contrast to the state-dependent $v(z)$ chosen by the abstraction based controller as constructed in the previous section. The simulation results for the constant v controller and the formally synthesized controller are shown in Fig. 3. The resulting walking gait obtained via the abstraction based controller is illustrated in Fig. 4b.

As mentioned before, there are 157 possible inputs in the discrete abstraction. Depending on the state there might exist several inputs which satisfy all constraints (see Fig. 4a where Z^{abstr} is shown together with the number of available inputs at each point in Z^{abstr}). When this is the case, the controller uses this degree of freedom by selecting the v that minimizes the cost of transport at this state. For the trajectory shown in the plots, the maximal number of available inputs at a state is 71, and the state with the minimal number of inputs has 5 available inputs (see Fig. 4a). In the state space of the abstraction, many states exists for which no input enforcing the specifications exists at all; however these states are not part of $\mathcal{D}_{\mathbf{PZ}}^{init}$ and are never reached.



(a) Phase portrait for full dynamics, $(\theta, \dot{\theta})$, over eight steps.

(b) The torques of all joints over two steps. The torque bounds of ±8 are illustrated with horizontal red lines.

(c) Phase portrait for the zero dynamics (z_1, z_2) over eight steps. The reset set is represented by the left vertical line and the guard set represented by the right vertical line.

(d) The ZMP constraint function $A_{\text{ZMP}} F_{st}$ over two steps. The upper bound of this constraint is equal to zero as illustrated by the horizontal red line.

Figure 3: Plots demonstrating the different specifications for the walking gait obtained from a constant $v = v^*$ as generated in [18] (left figure in each subplot) and the state-dependent $v(z)$ provided by the abstraction based controller (right figure in each subplot).

Fig. 3a and Fig. 3c show eight steps of AMBER 2 in the $(\theta, \dot{\theta})$ plane, and in the (z_1, z_2) plane, respectively. For the constant input $v = v^*$, the initial condition is chosen so that the angles and zero dynamics evolve in a periodic manner. Therefore, one cannot distinguish different steps. The trajectory of AMBER 2 when controlled with the abstraction based controller starts with the same initial conditions. We see that it also converges after three steps to a periodic behavior. This happens even though we did not include this requirement in the specifications. In Fig. 3c, we see that due to the varying v, z_2 changes much more in the case of the abstraction based controller than in the case of constant $v = v^*$. We also note that the constraint (C2) is satisfied since the magnitude of the angular velocities never exceed 5 rad/s. Fig. 3b and Fig. 3d show the satisfaction of the torque constraint and the ZMP constraint (C1) and (C3), respectively. AMBER 2 with constant input $v = v^*$ is not able to satisfy the ZMP constraint, and it even violates the torque constraints at the end of the steps. The abstraction based controller enforces these constraints even during the inter sampling times. These simulation results indicate that the abstraction based controller is able to generate a physically realizable stable walking gait.

Acknowledgments

This research is supported by NSF CPS Awards 1239055, 1239037 and 1239085.

6. REFERENCES

[1] AMBER-Lab. Human-inspired robotic walking with AMBER 2.0. `http://youtu.be/d6oM5sLI9vA`.

[2] A. D. Ames. Human-inspired control of bipedal walking robots. *Automatic Control, IEEE Transactions on*, 59(5):1115–1130, May 2014.

[3] A. D. Ames, E. A. Cousineau, and M. J. Powell. Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 135–144. ACM, 2012.

[4] A. D. Ames, K. Galloway, K. Sreenath, and J. Grizzle. Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *Automatic Control, IEEE Transactions on*, 59(4):876–891, 2014.

[5] A. Colombo and D. Del Vecchio. Supervisory control of differentially flat systems based on abstraction. In *50th IEEE Conference on Decision and Control and European Control Conference.*, pages 6134–6139. IEEE, 2011.

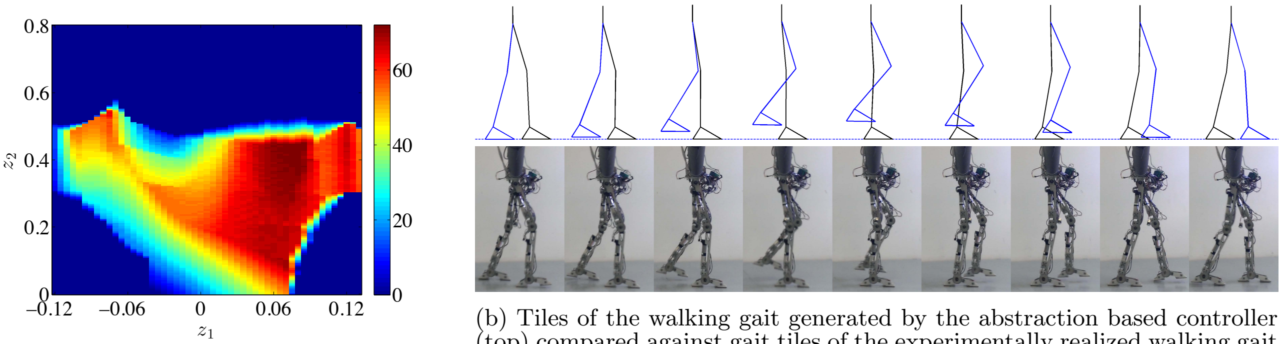
[6] A. Colombo and A. Girard. An approximate abstraction approach to safety control of differentially flat systems. In *European Control Conference*, pages 4226–4231. IEEE, 2013.

[7] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343–352, Feb. 2009.

[8] J. Fu, H. G. Tanner, and J. Heinz. Concurrent multi-agent systems with temporal logic objectives: Game theoretic analysis and planning through negotiation. *IET Control Theory and Applications*, 9(3):465–474, 2015.

[9] R. Goebel, R. Sanfelice, and A. Teel. Hybrid dynamical systems. *IEEE Contr. Syst. Mag.*, 29(2):28–93, Apr. 2009.

[10] J. W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE TAC*, 46(1):51–64, Jan. 2001.



(a) Number of available control inputs.

(b) Tiles of the walking gait generated by the abstraction based controller (top) compared against gait tiles of the experimentally realized walking gait (bottom) obtained from a constant $v = v^*$ controller [18] (video at [1]).

Figure 4: The domain of the abstraction based controller with the number of available inputs (a) together with tiles of the resulting walking gait (b) as compared against the experimentally realized walking gait with a constant input $v = v^*$.

[11] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet. 3D bipedal robotic walking: models, feedback control, and open problems. In *8th IFAC Symposium on Nonlinear Control Systems*, 2010.

[12] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. D. Ames. Dynamic multi-domain bipedal walking with ATRIAS through SLIP based human-inspired control. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 263–272. ACM, 2014.

[13] Y. Hürmüzlü and D. B. Marghitu. Rigid body collions of planar kinematic chains with multiple contact points. *Intl. J. of Robotics Research*, 13(1):82–92, Feb. 1994.

[14] C. M. Kellett, H. Shim, and A. R. Teel. Further results on robustness of (possibly discontinuous) sample and hold feedback. *IEEE Transactions on Automatic Control*, 49(7):1081–1089, Jan. 2004.

[15] A. Lamperski and A. Ames. Lyapunov theory for Zeno stability. *Automatic Control, IEEE Transactions on*, 58(1):100–112, Jan. 2013.

[16] J. Liu, N. Ozay, U. Topcu, and R. M. Murray. Synthesis of reactive switching protocols from temporal logic specifications. *IEEE Transactions on Automatic Control*, 58(7):1771–1785, July 2013.

[17] J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. S. Sastry. Dynamical properties of hybrid automata. *Automatic Control, IEEE Transactions on*, 48(1):2–17, 2003.

[18] W.-L. Ma, H.-H. Zhao, S. Kolathaya, and A. D. Ames. Human-inspired walking via unified PD and impedance control. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5088–5094, May 2014.

[19] M. Mazo Jr, A. Anta, and P. Tabuada. An ISS self-triggered implementation of linear controllers. *Automatica*, 46(8):1310–1314, 2010.

[20] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, Mar. 1994.

[21] D. Nesic and A. R. Teel. Input-output stability properties of networked control systems. *IEEE Transactions on Automatic Control*, 49(10):1650–1667, Jan. 2004.

[22] H. Park, A. Ramezani, and J. W. Grizzle. A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking. *IEEE Trans. Robot.*, 29(2):331–345, 2013.

[23] H. Park, K. Sreenath, J. W. Hurst, and J. W. Grizzle. Identification of a bipedal robot with a compliant drivetrain: Parameter estimation for control design. *IEEE Contr. Syst. Mag.*, 31(2):63–88, Apr. 2011.

[24] P. Roy, P. Tabuada, and R. Majumdar. Pessoa 2.0: A controller synthesis tool for cyber-physical systems. In *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, pages 315–316, 2011.

[25] S. S. Sastry. *Nonlinear Systems: Analysis, Stability and Control*. Springer, New York, June 1999.

[26] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*, volume 3. Wiley, 2006.

[27] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.

[28] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus. Optimality and robustness in multi-robot path planning with temporal logic constraints. *The International Journal of Robotics Research (IJRR)*, 32(8):889–911, July 2013.

[29] M. Vukobratović and B. Borovac. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

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

[31] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE TAC*, 48(1):42–56, 2003.

[32] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, July 2012.

[33] H.-H. Zhao, W.-L. Ma, M. Zeagler, and A. Ames. Human-inspired multi-contact locomotion with AMBER2. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 199–210, 2014.