

---

# Achieving Bipedal Running with RABBIT: Six Steps toward Infinity

B. Morris<sup>1</sup>, E.R. Westervelt<sup>2</sup>, C. Chevallereau<sup>3</sup>, G. Buche<sup>4</sup>, and  
J.W. Grizzle<sup>1</sup>

<sup>1</sup> Control Systems Laboratory, EECS Department, University of Michigan, Ann Arbor, Michigan 48109-2122, USA, {MorrisBJ,Grizzle}@umich.edu

<sup>2</sup> Locomotion and Biomechanics Laboratory, Department of Mechanical Engineering, The Ohio State University, Columbus, Ohio 43210, USA, Westervelt.4@osu.edu

<sup>3</sup> IRCCyN, Ecole Centrale de Nantes, UMR CNRS 6597, BP 92101, 1 rue de la Noë, 44321 Nantes, cedex 03, France, Christine.Chevallereau@ircyn.ec-nantes.fr

<sup>4</sup> LAG, Ecole Nationale d'Ingénieurs Electriciens de Grenoble, BP 46, 38402 St Martin d'Hères, France Gabriel.Buche@inpg.fr

**Summary.** This paper reviews and expands the class of hybrid zero dynamics (HZD) controllers that induce stable running in bipedal robots and discusses related experiments conducted in September 2004 in Grenoble, France. In these experiments, RABBIT, a five-link, four-actuator, planar bipedal robot, executed six consecutive running steps. These steps were notably human-like, having a long stride length (approx. 50 cm or 36% of body length), flight phases of significant duration (approx. 100 ms or 25% of the step duration), an upright posture, and an average forward rate of 0.6 m/s. A video is available at [8,18]. Validation of the theory's prediction that the running gait would be stable was not possible in the time available for the experiments. Unmodeled dynamic and geometric effects that contributed to the implementation difficulties are discussed.

## 1 Introduction

Designed and built as a platform to explore legged locomotion, RABBIT, a five-link, four-actuator, planar, bipedal robot [2, 3] (see Fig. 1(a)), has provided a means to test conceptually new approaches to underactuated, active, dynamic walking and running. Since March 2003, RABBIT has been used to experimentally verify a mathematical framework for the systematic design, analysis, and optimization of controllers that induce stable walking gaits in  $N$ -link planar bipedal robots with one degree of underactuation [9, 21]. Walking controllers designed within this framework act by enforcing *virtual constraints*, which are holonomic constraints on the robot's configuration that

are asymptotically imposed through feedback control. The existence and stability properties of resulting periodic walking motions are analyzed on the basis of the *hybrid zero dynamics* (HZD) of walking. The HZD of walking is a two-dimensional, invariant, sub-dynamic of the full hybrid walking model that arises from perfect enforcement of virtual constraints that satisfy certain boundary conditions.

The next challenge for RABBIT is to achieve a stable *running gait*, that is, a gait consisting of alternating phases of stance (one leg on the ground) and flight (no contact with the ground). Recent work in [5,6] extends the method of virtual constraints and the notion of an HZD to encompass stable running in robots such as RABBIT. The developed control strategy is hybrid with both continuous and event-based actions and leads to the deliberate creation of an *HZD of running*. As in walking, the HZD of running is a low-dimensional, invariant, sub-dynamic of the full hybrid running model that arises from perfect enforcement of virtual constraints that satisfy certain boundary conditions.

The present paper builds on the work of [6], establishing a new, larger class of controllers that induce stable running. These modified controllers are obtained by designing virtual constraints with fewer boundary conditions than required to achieve an HZD, yielding running motions that are simpler to design and control.

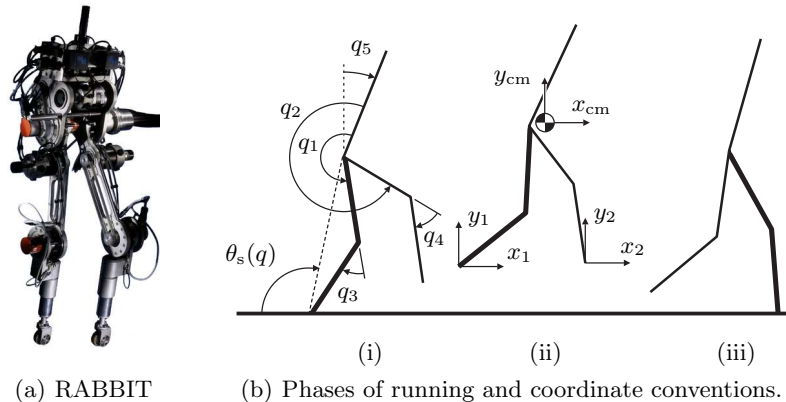
### 1.1 Related Work on Running Machines

Nearly twenty years ago, Raibert developed a series of 2D and 3D running machines that are more commonly called hoppers [17]. These hoppers employed a three part controller regulating hopping height, foot touchdown angle, and body attitude correction. The successful use of such a controller is tied to the morphology of the hopper: a point mass at the end of an actuated, compliant prismatic leg.

Recently, there have been a number of successful demonstrations of running in bipedal robots with revolute knees. In late 2003, both Iguana Robotics and Sony announced (separate) experimental demonstrations of running. Iguana Robotics' controller was based on central pattern generators (CPGs) and Sony's was based on the ZMP. In early 2004, running was announced for HRP-2LR [14] using a controller based on "resolved momentum". In December 2004, Honda's robot, ASIMO, achieved running at 3 km/h (0.83 m/s) with a 50 ms flight phase using a controller based on "posture control".

### 1.2 Outline

The remainder of the paper is a self-contained description of the theoretical development and hardware modifications leading up to the experiment in which RABBIT took six consecutive running steps. Echoing [6, Sects. III and



**Fig. 1.** RABBIT and the different phases of running with coordinate conventions labeled. In (b), the robot is shown (i) at the end of the stance phase; (ii) during flight; and (iii) at the beginning of the stance phase just after landing. To avoid clutter, the coordinate conventions have been spread out over the stance and flight phases. Angles are positive in the *counter clockwise* direction.

IV], Section 2 develops an open-loop model for RABBIT.<sup>5</sup> Section 3 discusses an extension to the control law given in [6]. Philosophy and motivation of the modified control law are given in Section 3.1 with a detailed development of the hybrid controller in Sections 3.2 to 3.6. The resulting closed-loop model of RABBIT is given in Section 3.7 with a discussion of theoretical stability in Section 3.8. Section 4 outlines a method for the design of stable gaits using constrained nonlinear optimization and includes a numerical example. Section 5 presents results from the first experimental implementation of running on RABBIT and a discussion outlining a number of possible reasons why stable running was not observed. Conclusions are drawn in Section 6.

## 2 Modeling

### 2.1 Assumptions and Terminology

RABBIT is modeled as a planar robot with five rigid massive links connected by (four) actuated, frictionless, revolute joints. The model is subdivided into two legs with identical physical properties, and a torso. The legs each consist of two links, a thigh and a shank, and are connected to each other and the single-link torso at the hips. Let  $q_b := (q_1, q_2, q_3, q_4)'$  be the vector of actuated body coordinates,  $q_5$  be the absolute coordinate that gives the robot's absolute orientation, and  $x_{cm}$  and  $y_{cm}$  be the cartesian coordinates that give the

<sup>5</sup> In fact, the content of Section 2 is based entirely on [6, Sects. III and IV] and is included for completeness.

horizontal and vertical positions of the robot's center of mass. See Fig. 1(b) for a depiction of the robot's morphology and coordinate convention.

The robot is said to be in *flight phase* when neither leg is in contact with the ground, and in *stance phase* when one leg is in stationary contact with the ground. The point of contact is modeled as an ideal pivot. During stance, the leg contacting the ground is called the *stance leg* and the other is called the *swing leg*. The transition from stance to flight is called *takeoff* and the transition from flight to stance is called *landing*. In this context, (steady-state) *running* is defined as a sequence of alternating stance and flight phases that is symmetric with respect to the left and right legs stride-to-stride.<sup>6</sup>

## 2.2 Dynamics of Flight and Stance

In the flight phase, the robot has 7 DOF with generalized coordinates  $q_f := (q'_b, q_5, x_{cm}, y_{cm})'$ . The equations of motion for this phase may be derived using the method of Lagrange and written in the following form:

$$D_f(q_b)\ddot{q}_f + C_f(q_b, \dot{q}_f)\dot{q}_f + G_f(q_f) = B_f u. \quad (1)$$

Introducing the state vector  $x_f := (q'_f, \dot{q}'_f)'$ , the model (1) is expressed as

$$\dot{x}_f = f_f(x_f) + g_f(x_f)u. \quad (2)$$

The state space is taken as  $\mathcal{X}_f = T\mathcal{Q}_f = \{x_f = (q'_f, \dot{q}'_f)' \mid q_f \in \mathcal{Q}_f, \dot{q}_f \in \mathbb{R}^7\}$ , where the configuration space  $\mathcal{Q}_f$  is a simply-connected, open subset of  $\mathbb{R}^7$  corresponding to physically reasonable configurations of the robot.

In the stance phase, the stance leg end is fixed and, therefore,  $x_{cm}$  and  $y_{cm}$  are no longer independent coordinates. Accordingly, the robot in stance phase has 5 DOF with generalized coordinates  $q := (q'_b, q_5)'$ . Similar to the flight phase, the equations of motion may be written as

$$D_s(q_b)\ddot{q} + C_s(q_b, \dot{q})\dot{q} + G_s(q) = B_s u. \quad (3)$$

Note that (3) may be obtained by subjecting (1) to the constraint that one leg end is in contact with the ground. Choosing the state vector  $x_s := (q', \dot{q}')'$ , the model (3) may be expressed as

$$\dot{x}_s = f_s(x_s) + g_s(x_s)u. \quad (4)$$

The state space is taken as  $\mathcal{X}_s = T\mathcal{Q}_s = \{x_s = (q', \dot{q}')' \mid q \in \mathcal{Q}_s, \dot{q} \in \mathbb{R}^5\}$ , where the configuration space  $\mathcal{Q}_s$  is a simply-connected, open subset of  $\mathbb{R}^5$  corresponding to physically reasonable configurations of the robot.

<sup>6</sup> The chosen definition of running is fundamental to the following model and controller development. Other authors have defined running based on the motion of the center of mass or the reaction force profile on the stance leg, for example see [15].

### 2.3 Transitions

Landing, the transition from the flight phase to the stance phase, is modeled as a rigid impact. During this instantaneous event impulsive reaction forces from the ground bring the velocity of the touchdown foot to zero without causing it to rebound or slip. In addition, at the moment of landing, the robot's configuration remains unchanged, but joint velocities change instantaneously [12]. The post-impact joint velocities<sup>7</sup> are given by a function [6, Eq. (21)],

$$\dot{q}^+ = \tilde{\Delta}(q_f^-, \dot{q}_f^-). \quad (5)$$

Since the gait is assumed to be symmetric from stride to stride (with respect to the left and right legs) a state relabeling matrix  $R$  is used to swap leg definitions (redefine the coordinates) at landing. The flight-to-stance transition operator, including state relabeling, is therefore defined as

$$x_s^+ = \Delta_f^s(x_f^-) := \begin{bmatrix} R q^- \\ R \tilde{\Delta}(q_f^-, \dot{q}_f^-) \end{bmatrix}. \quad (6)$$

This transition operator is applied when the end of the advancing leg touches the ground, that is, when  $y_2 = 0$  (see Fig. 1(b)). Define the function,  $H_f^s : T\mathcal{Q}_f \rightarrow \mathbb{R}$  by  $H_f^s(x_f) := y_2$ , so that  $H_f^s(x_f) = 0$  characterizes the transition hypersurface surface within  $T\mathcal{Q}_f$ .

The transition from stance to flight is also modeled as an instantaneous event, but one on which positions and velocities are unchanged,

$$x_f^+ = \Delta_s^f(x_s^-) := \begin{bmatrix} \begin{bmatrix} q^- \\ f_{\text{cm}}(q^-) \end{bmatrix} \\ \begin{bmatrix} \dot{q}^- \\ \frac{\partial}{\partial q} f_{\text{cm}}(q^-) \dot{q}^- \end{bmatrix} \end{bmatrix}. \quad (7)$$

where  $f_{\text{cm}}(q) := (x_{\text{cm}}(q), y_{\text{cm}}(q))'$  gives the location of the center of mass. The transition from stance into flight is treated as a control decision because the flight phase is initiated (at will) by accelerating the stance leg off the ground. Thus, a description of the transition hypersurface,  $\mathcal{S}_s^f$ , as a level set of a function  $H_s^f(x_s) : T\mathcal{Q}_s \rightarrow \mathbb{R}$  is included as part of the control law.

---

<sup>7</sup> The terms  $x_f^- := (q_f^-, \dot{q}_f^-)'$  and  $x_s^+ := (q^+, \dot{q}^+)'$  refer to the system state just before and just after the landing event. The terms  $x_s^- := (q^-, \dot{q}^-)'$  and  $x_f^+ := (q_f^+, \dot{q}_f^+)'$  refer to the system state just before and just after the takeoff event. The addition of the superscript “\*” (such as  $x_f^{+*}$ ) indicates reference to the value *at steady-state*, i.e., on the periodic orbit.

## 2.4 A Hybrid, Open-Loop Model of Running

The stance and flight dynamic models may be represented compactly, along with their transition models, as a discrete-event system with two charts (terminology taken from [10]). This open-loop hybrid model is specified by charts  $\Sigma_f$  and  $\Sigma_s$  where for  $(i, j) \in \{(f, s), (s, f)\}$ ,  $\Sigma_i = \{\mathcal{X}_i, \mathcal{F}_i, \mathcal{S}_i^j, \mathcal{T}_i^j\}$ ,

1.  $\mathcal{X}_i$  is a *state manifold*, which is 10 dimensional in stance and 14 dimensional in flight;
2.  $\mathcal{F}_i$  is a *flow on the state manifold*, a differential equation describing the in-phase motion on  $\mathcal{X}_i$ ;
3.  $\mathcal{S}_i^j$  is a *switching hypersurface*, a hypersurface of  $\mathcal{X}_i$  corresponding to transition from one state manifold to another; and
4.  $\mathcal{T}_i^j$  is a *transition map* giving initial conditions for the next continuous phase.

In this notation, the open-loop hybrid model is

$$\Sigma_f : \begin{cases} \mathcal{X}_f = T\mathcal{Q}_f \\ \mathcal{F}_f : (\dot{x}_f) = f_f(x_f) + g_f(x_f)u \\ \mathcal{S}_f^s = \{x_f \in T\mathcal{Q}_f \mid H_f^s(x_f) = 0\} \\ \mathcal{T}_f^s : x_s^+ = \Delta_f^s(x_f^-) \end{cases} \quad (8a)$$

$$\Sigma_s : \begin{cases} \mathcal{X}_s = T\mathcal{Q}_s \\ \mathcal{F}_s : (\dot{x}_s) = f_s(x_s) + g_s(x_s)u \\ \mathcal{S}_s^f = \{x_s \in T\mathcal{Q}_s \mid H_s^f(x_s) = 0\} \\ \mathcal{T}_s^f : x_f^+ = \Delta_s^f(x_s^-). \end{cases} \quad (8b)$$

## 3 Control Methodology

### 3.1 Summary and Philosophy

The overall philosophy of HZD control is to use the freedom available in feedback design to achieve a parameterized family of closed-loop systems whose stability analysis is analytically tractable. This allows the use of numerical optimization to search among the family of closed-loop systems to find those that yield a desired behavior, such as stable running at a pre-determined speed, with upper bounds on peak actuator power and the coefficient of static friction between the leg end and ground.

Parameterization is achieved through the use of virtual constraints in both the stance and flight phases. Perfect enforcement of virtual constraints results in low-dimensional surfaces that are invariant under the differential equations

of the closed-loop model<sup>8</sup> and are also invariant under the transition maps.<sup>9</sup> To achieve the invariance at landing, a deadbeat action is incorporated in the flight phase controller that steers the robot to land in a pre-determined configuration, while respecting conservation of angular momentum about the robot’s center of mass. This hybrid controller creates a one DOF HZD that allows the stability of a running motion to be analyzed in closed form on the basis of a one-dimensional Poincaré map.

In the first running experiment attempted on RABBIT, there was not sufficient time<sup>10</sup> to implement completely the controller of [6]. The controller that was implemented used virtual constraints in the both the stance and flight phases, but the deadbeat action of the flight phase controller was not implemented to regulate the final configuration of the robot at touchdown. Instead, to account for the changing configuration of the robot at touchdown, the transition controller of [20] was adopted.

In the next sections, the controller that was used in the running experiment is detailed. This controller does not create a one DOF HZD, and thus the stability analysis of [6] must be modified; the key points of the analysis are highlighted in Section 3.8.

### 3.2 Preliminaries on virtual constraints

Since RABBIT has four independent actuators (two at the hips and two at the knees), four virtual constraints may be imposed in both the stance and flight phases. To define them, consider a function pair  $\{\theta(q), h_d(\theta)\}$ , where  $\theta : \mathcal{Q} \rightarrow \mathbb{R}$  is a monotonic scalar function of the configuration variables, and  $h_d : \mathbb{R} \rightarrow \mathbb{R}^4$  is a function giving the desired configuration of the actuated joints as a function of  $\theta$ . The virtual constraints are expressed as outputs of (8),

$$y = h(q) := q_b - h_d \circ \theta(q). \quad (9)$$

The output (9) is zeroed by the action of a state feedback controller. The design of such a controller is a well-understood, standard problem of nonlinear control [13].

For purposes of design, the virtual constraints are parameterized [6]. For notational convenience, the stance phase and flight phase virtual constraints will be parameterized separately by  $a_s$  and  $a_f$ , respectively. These parameter sets, which lie in the parameter spaces  $\mathcal{A}_s := \mathbb{R}^{n_s}$  and  $\mathcal{A}_f := \mathbb{R}^{n_f}$ , may be

<sup>8</sup> “Invariant” in this sense means that if the differential equation is initialized on the constraint surface, then its solution remains on the constraint surface until a transition occurs.

<sup>9</sup> “Invariant” in this sense means that if the solution is on the flight phase (resp. stance phase) constraint surface at touchdown (resp. takeoff), then after transition the solution will be contained in the stance phase (resp. flight phase) constraint surface.

<sup>10</sup> A total of two weeks were available to perform the experiments.

updated at takeoff and landing events but are otherwise constant. With this notation, the virtual constraints for stance and flight are, respectively,<sup>11</sup>

$$y = q_b - h_{d,s}[a_s](\theta_s(q)) \quad (10a)$$

$$y = q_b - h_{d,f}[a_f](\theta_f[a_f](q_f)). \quad (10b)$$

### 3.3 Stance phase control

The controller for the stance phase acts by updating the parameters  $a_s$  and by enforcing the virtual constraints (10a). As a result of enforcing the virtual constraints, in stance phase, the robot behaves as an unactuated 1 DOF system whose properties may be tuned by choosing different constraint parameters. Apart from different boundary conditions on the virtual constraints, this control is identical to the walking controllers developed in [19, 21]. The set of parameters  $a_s$  is detailed next.

The stance phase parameter vector,  $a_s$ , may be expressed as the vector

$$a_s := (a'_{s,0}, a'_{s,1}, \dots, a'_{s,m_s-1}, a'_{s,m_s}, \theta_s^-, \theta_s^+)' \quad (11)$$

where  $m_s \geq 3$ ,  $a_{s,i} \in \mathbb{R}^4$  for  $i \in \{0, 1, \dots, m_s - 1, m_s\}$ , and  $\theta_s^-, \theta_s^+ \in \mathbb{R}$ . Note that  $n_s = 4(m_s + 1) + 2$ . The terms  $\theta_s^-$  and  $\theta_s^+$  are the values of the function  $\theta_s(q)$  evaluated at the end and the beginning of the stance phase. In [19, 21],  $h_d$  is expressed in terms of Bézier polynomials. Here, a slightly different class of polynomials is used. These polynomials satisfy the following:<sup>12</sup>

$$\begin{aligned} h_{d,s}[a_s](\theta_s^+) &= a_{s,0} & \frac{d}{d\theta_s} h_{d,s}[a_s](\theta_s^-) &= a_{s,m_s-1} \\ \frac{d}{d\theta_s} h_{d,s}[a_s](\theta_s^+) &= a_{s,1} & h_{d,s}[a_s](\theta_s^-) &= a_{s,m_s}. \end{aligned} \quad (12)$$

The stance phase virtual constraints are imposed on the dynamics by using a control  $u_s : \mathcal{X}_s \rightarrow \mathbb{R}^4$  that drives (10a) to zero *in finite time*. The specific assumptions are as in [9, 21].

### 3.4 Flight phase control

The development of the flight phase controller is similar to that of the stance phase controller. The key difference is the choice of  $\theta_f$  in (10b) to be a function of the position of the center of mass. The flight phase parameter vector,  $a_f$ , is defined as

$$a_f := (a'_{f,0}, a'_{f,1}, \dots, a'_{f,m_f-1}, a'_{f,m_f}, x_{cm,f}^+, \dot{x}_{cm,f}^+, T_f)' \quad (13)$$

<sup>11</sup> Terms that are constant during the continuous phases of motion, and potentially updated at phase transitions, will be considered *parameters* and enclosed in square brackets.

<sup>12</sup> In fact, any class of smooth functions satisfying these properties may be used to define virtual constraints.



where  $m_f \geq 3$ ,  $a_{f,i} \in \mathbb{R}^4$  for  $i \in \{0, 1, \dots, m_f-1, m_f\}$ , and  $x_{\text{cm},f}^+, \dot{x}_{\text{cm},f}^+, T_f \in \mathbb{R}$ . Note that  $n_f = 4(m_f + 1) + 3$ . The terms  $x_{\text{cm},f}^+$ ,  $\dot{x}_{\text{cm},f}^+$ , and  $T_f$  are, respectively, the horizontal position of the center of mass at the beginning of the flight phase, the horizontal velocity of the center of mass at the beginning of the flight phase, and the estimated<sup>13</sup> duration of the flight phase. The flight phase virtual constraints (10b) are given by

$$\theta_f[a_f](q_f) := \frac{1}{T_f} \left( \frac{x_{\text{cm}} - x_{\text{cm},f}^+}{\dot{x}_{\text{cm},f}^+} \right), \quad (14)$$

and  $h_{d,f}[a_f]$ , which, as in the stance phase, is a smooth, vector-valued function that satisfies

$$\begin{aligned} h_{d,f}[a_f](0) &= a_{f,0} & \frac{d}{d\theta_f} h_{d,f}[a_f](1) &= a_{f,m_f-1} \\ \frac{d}{d\theta_f} h_{d,f}[a_f](0) &= a_{f,1} & h_{d,f}[a_f](1) &= a_{f,m_f}. \end{aligned} \quad (15)$$

For a given stride, let  $t_f$  denote elapsed time within the flight phase. By conservation of linear momentum,  $\dot{x}_{\text{cm},f}^+$  is constant during flight, which implies  $t_f = (x_{\text{cm}} - x_{\text{cm},f}^+)/\dot{x}_{\text{cm},f}^+$ . Therefore, under the given assumptions,  $\theta_f = t_f/T_f$  is a valid substitute for (14), and for this reason, the given flight phase virtual constraints are said to be *time scaled*. Flight phase virtual constraints are enforced using any smooth state feedback controller  $u_f : \mathcal{X}_f \rightarrow \mathbb{R}^4$  that drives (10b) to zero exponentially quickly.

Note that finite-time convergence is not used in the flight phase. To perform reduced-order stability analysis similar to that outlined in Section 3.8, a finite-time converging controller must be used in either the stance or the flight phase. A finite-time controller is used in the stance phase to render the stance phase constraint surface finite-time attractive so that the analysis of running will be similar to that of walking [9]. For further discussion of this point, refer to Section 3.8.

### 3.5 Transition control: landing

In the event that landing occurs with the state of the robot not satisfying the virtual constraints, the control parameters of the subsequent stance phase,  $a_s$ , are updated to ensure that the configuration of the robot satisfies  $q_b - h_{d,s}[a_s](\theta_s^+) = 0$ . The parameter updates are governed by the differentiable function  $w_f^s : \mathcal{S}_f^s \rightarrow \mathcal{A}_s$ , such that for  $a_s = w_f^s(x_f^-)$ ,

<sup>13</sup> Calculation of  $T_f$  requires the height of the center of mass at landing,  $y_{\text{cm},f}^-$  to be known *a priori*, which is only possible if the virtual constraints are exactly enforced throughout the flight phase.

$$\begin{aligned}
a_{s,0} &= q_{b,s}^+ \\
a_{s,1} &= a_{s,1}^* & \theta_s^+ &= \theta_s(q^+) \\
&\vdots & \theta_s^- &= \theta_s^{-*} \\
a_{s,m_s-1} &= a_{s,m_s-1}^* \\
a_{s,m_s} &= a_{s,m_s}^*.
\end{aligned} \tag{16}$$

In the above,  $q^+$  is calculated using  $\Delta_f^s(x_f^{-*})$ , and the terms  $\theta_s^{-*}$  and  $a_{s,i}^* \in \mathbb{R}^4$ ,  $i \in \{1, \dots, m_s - 1, m_s\}$  are constant parameters chosen during design.

If the stance phase finite-time controller can satisfy the virtual constraints (10a) before the liftoff event occurs, and the parameter updates obey (16), then the stance phase will terminate with  $q_b - h_{d,s}[a_s](\theta_s^-) = 0$ , or equivalently, with  $q^- = q^{-*}$ .

### 3.6 Transition control: takeoff

At takeoff, the parameters of the flight phase virtual constraints,  $a_f$ , are updated so that the duration of the planned motion of the robot is equal to the estimated flight time. Parameter updates are governed by a continuously differentiable function  $w_s^f : \mathcal{S}_s^f \rightarrow \mathcal{A}_f$ , such that for  $a_f = w_s^f(x_s^-)$ ,

$$\begin{aligned}
a_{f,0} &= a_{f,0}^* \\
a_{f,1} &= a_{f,1}^* & x_{\text{cm},f}^+ &= (f_{\text{cm}}(q^-))_1 \\
&\vdots & \dot{x}_{\text{cm},f}^+ &= \left( \frac{\partial f_{\text{cm}}}{\partial q}(q^-) \dot{q}^- \right)_1 \\
a_{f,m_f-1} &= a_{f,m_f-1}^* \\
a_{f,m_f} &= a_{f,m_f}^* \\
T_f &= \frac{\dot{y}_{\text{cm},f}^+}{g} + \frac{\sqrt{(\dot{y}_{\text{cm},f}^+)^2 - 2g(y_{\text{cm},f}^{-*} - y_{\text{cm},f}^+)}}{g}.
\end{aligned} \tag{17}$$

where,  $g$  is the magnitude of acceleration of gravity and  $y_{\text{cm},f}^{-*}$  is the height of the center of mass at the end of the flight phase on the limit cycle. The terms  $a_{f,i}^* \in \mathbb{R}^4$ ,  $i \in \{0, 1, \dots, m_f - 1, m_f\}$  are parameters chosen during design. Initiation of the takeoff event is a control decision, designated to occur when  $\theta_s(q) = \theta_s^-$ . In the open loop model (8), the switching hypersurface is therefore given by  $\mathcal{S}_s^f = \{(x_s, a_s) \in \mathcal{X}_s \times \mathcal{A}_s \mid H_s^f(x_s, a_s) = 0\}$  where  $H_s^f(x_s, a_s) := \theta_s(q) - \theta_s^-$ .

### 3.7 Resulting closed-loop model of running

To form the closed-loop model of running, the state space of the open-loop model, (8), is enlarged to include the parameters of the flight and stance phases. Define the augmented state spaces  $\tilde{\mathcal{X}}_f := T\mathcal{Q}_f \times \mathcal{A}_f$  and  $\tilde{\mathcal{X}}_s := T\mathcal{Q}_s \times \mathcal{A}_s$

with elements given by  $\bar{x}_f := (q'_f, \dot{q}'_f, a'_f)'$  and  $\bar{x}_s := (q', \dot{q}', a'_s)'$ . The closed-loop dynamics may then be written as

$$\bar{f}_f(\bar{x}_f) := \begin{bmatrix} f_f(x_f) + g_f(x_f)u_f(x_f) \\ 0_{n_f \times 1} \end{bmatrix} \quad (18a)$$

$$\bar{f}_s(\bar{x}_s) := \begin{bmatrix} f_s(x_s) + g_s(x_s)u_s(x_s) \\ 0_{n_s \times 1} \end{bmatrix}. \quad (18b)$$

The zero vectors reflect that the virtual constraint parameters do not change during the continuous phases of running. The impact maps, in which the parameters are updated, are modified to include the parameter update laws,  $w_f^s$  and  $w_s^f$ :

$$\bar{\Delta}_f^s(\bar{x}_f^-) := \begin{bmatrix} \Delta_f^s(x_f^-) \\ w_f^s(x_f^-) \end{bmatrix} \quad (19a)$$

$$\bar{\Delta}_s^f(\bar{x}_s^-) := \begin{bmatrix} \Delta_s^f(x_s^-) \\ w_s^f(x_s^-) \end{bmatrix}. \quad (19b)$$

The closed-loop hybrid model is then

$$\Sigma_{\text{cl},f} : \begin{cases} \bar{\mathcal{X}}_f = T\mathcal{Q}_f \times \mathcal{A}_f \\ \bar{\mathcal{F}}_f : (\dot{\bar{x}}_f) = \bar{f}_f(\bar{x}_f) \\ \bar{S}_f^s = \{(x_f, a_f) \in \bar{\mathcal{X}}_f \mid H_f^s(x_f) = 0\} \\ \bar{\mathcal{T}}_f^s : \bar{x}_f^+ = \bar{\Delta}_f^s(\bar{x}_f^-) \end{cases} \quad (20a)$$

$$\Sigma_{\text{cl},s} : \begin{cases} \bar{\mathcal{X}}_s = T\mathcal{Q}_s \times \mathcal{A}_s \\ \bar{\mathcal{F}}_s : (\dot{\bar{x}}_s) = \bar{f}_s(\bar{x}_s) \\ \bar{S}_s^f = \{(x_s, a_s) \in \bar{\mathcal{X}}_s \mid H_s^f(x_s, a_s) = 0\} \\ \bar{\mathcal{T}}_s^f : \bar{x}_s^+ = \bar{\Delta}_s^f(\bar{x}_s^-). \end{cases} \quad (20b)$$

### 3.8 Existence and stability of periodic orbits

Analysis on running shows that asymptotically stable orbits of the HZD are asymptotically stabilizable orbits of the full-order hybrid model (equivalently, stabilizable running gaits) [6]. Using the idea of a *restricted Poincaré return map*—the Poincaré return map associated with the HZD—explicit criteria are given for determining the existence and stability of periodic orbits.

The Poincaré return map is a well-known tool for determining the existence of periodic orbits and their stability properties; for its use in hybrid systems, see for example [7, 9, 11, 16]. Its application to periodic orbits of (20) can be carried out using the results in [9] and a construction presented in [6, Thm. 1].

Since through development of these ideas would consume more space than is available, the main ideas are only sketched.

The first step is to construct a system with impulse effects (that is, a single-chart hybrid model) that has the same Poincaré map as (20). Following [6, Eq. (62)], define

$$\Sigma_{\text{cl}} : \begin{cases} \dot{\bar{x}}(t) = \bar{f}_s(\bar{x}(t)) & x^-(t) \notin \bar{S} \\ \bar{x}^+(t) = \bar{\Delta}(\bar{x}^-(t)) & \bar{x}^-(t) \in \bar{S}, \end{cases} \quad (21)$$

where  $\bar{S} := \bar{S}_s^f$ ,  $\bar{\Delta} := \bar{\Delta}_f^s \circ P_f$ , and  $P_f$  is the flow of the closed-loop flight phase model (see [6, Eq. (59)]). In words, this system consists of the differential equation of the closed-loop stance phase model of (20) and a generalized impact map  $\bar{\Delta}$  that includes the transition map from stance to flight, the flight phase dynamics, and the impact map from flight to stance. The generalized impact map is the result of event-based sampling the solution of (20) from takeoff to landing.

Because the virtual constraints in the stance phase are achieved with a continuous finite-time controller [1], the reduction technique of [9, Thm. 2] is applicable. Because the parameter updates in the stance phase can be computed in terms of the state of the robot at takeoff, the analysis of periodic orbits can be reduced to the computation of a one-dimensional restricted Poincaré map,  $\rho$ , having  $\mathcal{S}_s^f$  as a Poincaré section.

## 4 Design of Running Motions with Optimization

### 4.1 Optimization parameters

In designing a gait, a numerical routine is used to search the parameter spaces  $\mathcal{A}_s$  and  $\mathcal{A}_f$  for a set of parameters that results in a desirable gait (periodic orbit of (20)). Common requirements on the gait are achieved by incorporating constraints into the numerical search. Such constraints address actuator limits, allowable joint space, and unilateral ground-contact forces. The constraints also ensure steady-state running at a certain speed and overall efficiency of the gait. For the experiments reported here, the gait was designed using an optimization approach that combined the ideas of [4] and [21]; the optimization was performed directly on the parameters of the virtual constraints in order to *simultaneously* determine a periodic running motion and a controller that achieves it. This is in contrast with the approach of [6] where virtual constraints are designed by regression against optimal, pre-computed, periodic trajectories.

Virtual constraints are assumed to be identically satisfied on the periodic orbit, which has two consequences: First, the integration of the closed-loop system dynamics can be performed using the stance and flight phase zero dynamics (see [6] for details), resulting in short computation times; and second,

the virtual constraint parameters,  $a_s$  and  $a_f$ , are not independent. Once the independent parameters have been identified, standard numerical optimization routines may be used to search for desirable gaits. The implementation of such a procedure is outlined in the following subsections.

## 4.2 Boundary conditions of the virtual constraints

Periodicity at takeoff and landing results in constraints between the virtual constraint parameter vectors  $a_s$  and  $a_f$ . Given the state corresponding to the end of the limit-cycle stance phase,  $x_s^{-*} = (q^{-*}, \dot{q}^{-*})$ , the state at the beginning of the subsequent flight phase may be computed as  $x_f^{+*} = (q_f^{+*}, \dot{q}_f^{+*}) = \Delta_s^f(x_s^{-*})$ . For both  $x_s^{-*}$  and  $x_f^{+*}$  to satisfy the virtual constraints of their respective phases, the following relations must hold,

$$\begin{aligned} a_{s,m_s-1}^* &= \dot{q}_{b,s}^{-*} / \dot{\theta}_s^{-*} & a_{f,0}^* &= q_{b,f}^{+*} \\ a_{s,m_s}^* &= q_{b,s}^{-*} & a_{f,1}^* &= \dot{q}_{b,f}^{+*} T_f^*, \end{aligned} \quad (22)$$

which are derived by applying (12), (14), (15), and (17) to (10). These are the boundary conditions associated with the liftoff event of the periodic orbit. To find the boundary conditions associated with the landing event of the periodic orbit, consider the state of the robot corresponding to the beginning of the stance phase,  $x_s^{+*} = (q^{+*}, \dot{q}^{+*})$ . This can be related to the state corresponding to the end of the previous flight phase by inverting the landing map  $x_f^{-*} = (q_f^{-*}, \dot{q}_f^{-*}) = (\Delta_f^s)^{-1}(x_s^{+*})$ , yielding the following additional design constraints,

$$\begin{aligned} a_{s,0}^* &= q_{b,s}^{+*} & a_{f,m_f-1}^* &= \dot{q}_{b,f}^{-*} T_f^* \\ a_{s,1}^* &= \dot{q}_{b,s}^{+*} / \dot{\theta}_s^{+*} & a_{f,m_f}^* &= q_{b,f}^{-*}. \end{aligned} \quad (23)$$

The controller presented in this paper requires fewer boundary conditions than the HZD of running presented in [6]. To design virtual constraints for an HZD of running, boundary conditions are associated not only with the periodic orbit, but also with the the constraint surfaces of stance and flight. The additional boundary conditions associated with takeoff are satisfied by (22). The additional boundary conditions of landing are more difficult to meet because of conservation of angular momentum in the flight phase. The main theoretical result of this paper is that invariance of the flight and stance phase constraint surfaces over the landing event is not a necessary condition for achieving provably stable running. As noted earlier, relaxing this condition makes running motions significantly easier to design.

## 4.3 Optimization algorithm details

Trial gaits for the running experiments were generated using the constrained nonlinear optimization routine `fmincon` of MATLAB's Optimization Toolbox.

Three quantities are involved in optimization,  $J$ , a scalar cost function to be minimized on the periodic orbit,  $EQ$ , a vector of equality constraints, and  $INEQ$ , a vector of inequality constraints. The following is a description of the optimization procedure that was implemented. The independent and dependent terms<sup>14</sup> of optimization are given in Table 1. Note that when the optimizer terminates with the constraints satisfied,  $x_s^{+*}$  will be point on a closed-loop periodic orbit and the virtual constraints will be given by (11) and (13).

### Algorithm

1. Select  $x_s^{+*} = (q^{+*}, \dot{q}^{+*})$ , the state corresponding to the beginning of the stance phase.
2. Calculate  $\theta_s^{+*}$  by (16) and  $a_{s,0}^*, a_{s,1}^*$  by (23).
3. Select  $a_{s,2}^*, \dots, a_{s,m_s}^*$ , and  $\theta_s^{+*}$  to complete the stance phase parameter vector  $a_s$ .
4. Using parameters  $a_s$  and the initial condition  $x_s^{+*}$ , integrate the equations of motion of stance and apply the stance-to-flight transition operator,  $\Delta_s^f$ , to obtain  $x_f^{+*} = (q_f^{+*}, \dot{q}_f^{+*})$ .
5. Calculate  $a_{f,0}^*, a_{f,1}^*$  by (22);  $a_{f,m_f-1}^*, a_{f,m_f}^*$  by (23); and  $x_{cm,f}^{+*}$ ,  $\dot{x}_{cm,f}^{+*}$ , and  $T_f^*$  by (17).
6. Select  $a_{f,2}^*, \dots, a_{f,m_f-2}^*$  to complete the flight phase parameter vector  $a_f$ .
7. Using parameters  $a_f$ , and initial condition  $x_f^{+*}$ , integrate the equations of motion of flight through the landing event to obtain  $x_s^+$ .
8. Evaluate  $J$ ,  $EQ$ , and  $INEQ$ .
9. Iterate Steps 1 to 8 until  $J$  is (approximately) minimized, each entry of  $EQ$  is zero, and each entry of  $INEQ$  is less than zero.

### 4.4 An Example Running Motion

A sample running gait designed by the above algorithm is now presented. A stick diagram of this motion is given in Fig. 2(a). The stability analysis outlined in Section 3.8 was applied to the resulting running motion. Fig. 2(b) gives the restricted Poincaré map, which indicates that the motion is locally exponentially stable. The gait was designed to minimize electrical energy per distance traveled, with the following constraints:

#### Equality constraints, $EQ$

- error associated with finding a fixed point  $\|x_s^+ - x_s^{+*}\|$
- deviation from the desired running rate
- continuity of required frictional forces on takeoff and landing

<sup>14</sup> “Terms” is used to describe those variables used in optimization; these are different from the *parameters* of the virtual constraints.

**Table 1.** Independent and dependent terms used in optimization. The choice of the independent terms is non-unique and depends on the specific optimization procedure. The parameters below correspond to algorithm in Section 4.3, which is one straightforward method to ensure the boundary conditions of the virtual constraints are met.

Terms of Optimization	
Independent	Dependent
$x_s^{+*} \in \mathbb{R}^{10}$	$\theta_s^{+*} \in \mathbb{R}$
$a_{s,2}^*, \dots, a_{s,m_s}^* \in \mathbb{R}^4$	$a_{s,0}^*, a_{s,1}^* \in \mathbb{R}^4$
$\theta_s^{-*} \in \mathbb{R}$	$x_f^{+*} \in \mathbb{R}^{14}$
$a_{f,2}^*, \dots, a_{f,m_f-2}^* \in \mathbb{R}^4$	$a_{f,0}^*, a_{f,1}^* \in \mathbb{R}^4$
	$a_{f,m_f-1}^*, a_{f,m_f}^* \in \mathbb{R}^4$
	$x_{cm,f}^{+*}, \dot{x}_{cm,f}^{+*}, T_f^* \in \mathbb{R}$
	$x_s^+ \in \mathbb{R}^{10}$

### Inequality constraints, *INEQ*

- magnitude of the required torque at each joint less than 100 Nm
- knee angles to lie in  $(0^\circ, -70^\circ)$  and hip angles to lie in  $(130^\circ, 250^\circ)$  (see Fig. 1(b) for measurement conventions)
- minimum height of the swing foot during stance greater than 7 cm
- required coefficient of friction of the stance phase less than 0.7
- flight time greater than or equal to 25% of total gait duration
- landing foot impacts the ground at an angle of approach less than  $45^\circ$  from vertical
- joint angular velocities less than 5 rad/s

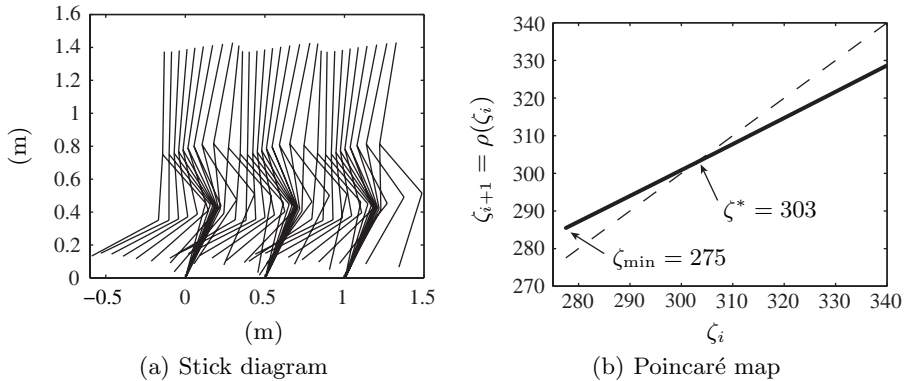
## 5 Experiment

### 5.1 Hardware Modifications to RABBIT

Prior to the experiment reported here, only walking experiments had been performed with RABBIT. To prepare for the task of running, the following four hardware modifications were made.

The first modification was the inclusion of prosthetic shock absorbers in the shanks. It was speculated that with shock absorbers the landing would cause less wear and tear on the harmonic drive gear reducers that form RABBIT’s hip and knee joints. The inclusion of shock absorbers added approximately 5 cm to length of each shank.

The second modification was the installation of force sensitive resistors into RABBIT’s point feet. These devices allowed for more accurate measurement of the touchdown time than did the previously installed mechanical contact



**Fig. 2.** Stick diagram and Poincaré map for the example running motion (rate 0.58 m/s). Poincaré map constructed by evaluating  $\zeta := (\sigma_{s,1}^-)^2/2$  at the end of successive stance phases, where  $\sigma_{s,1}^-$  is the angular momentum about the stance leg end just before liftoff. The fixed point,  $\zeta^* = 303$ , is located at the intersection of  $\rho$  and the identity map  $\zeta_i = \zeta_{i+1}$ , and corresponds to an equilibrium running rate of 0.58 m/s. The slope of the graph at  $\zeta^*$  is  $d\rho/d\zeta \approx 0.67$ , indicating exponential stability.

switches. Since these sensors suffer from significant drift, their signals were numerically differentiated to make easier the detection of impact events.

The last two modifications were the bolting of aluminum u-channel stock along each thigh and the widening of the hips. Both of these changes were made to help prevent flexing of the legs in the frontal plane. Significant flexing was witnessed during the first several experimental trials of running. This problem was more pronounced in running than in walking because of the greater impact forces associated with landing. On several occasions RABBIT “tripped itself” during a stance phase of running when the swing leg passed by the stance leg (the legs knocked against each other). Ironically, RABBIT was intentionally designed to have its legs close together to better approximate a planar biped.

## 5.2 Result: Six Running Steps

After completing hardware modifications and successfully reproducing previous walking experiments, running experiments were conducted. One such experiment (the implementation of the example running motion of Section 4.4) resulted in six human-like<sup>15</sup> running steps.

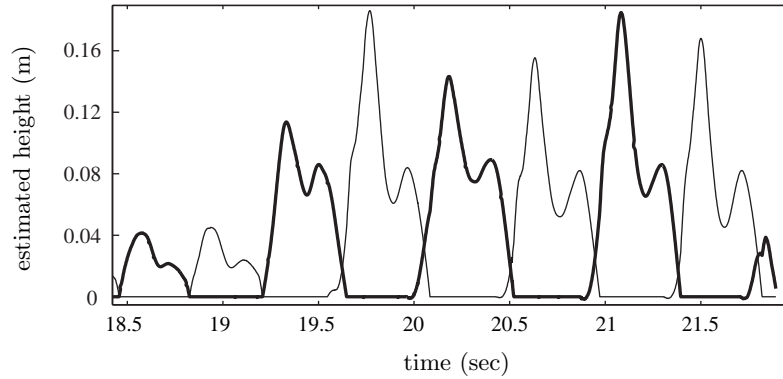
For this experiment, motion was initiated by an experimenter who pushed the robot forward, into the basin of attraction of a walking controller that induced walking with an average forward walking rate of 0.8 m/s. RABBIT

<sup>15</sup> A human-like gait is considered to be characterized by an upright posture, a torso leaning slightly forward, and a long step length.



then achieved stable walking, followed by a transition to running in a single step, followed by 6 running steps. After the sixth step, the experiment was terminated by the control software when the tracking error limit of 0.3 radians was exceeded for the stance knee angle. Examination of collected data suggests that tracking error resulted from actuator saturation. Data also shows the back leg extremely close to the ground at the moment the experiment was terminated, suggesting the back leg may have, in fact, struck the ground contributing additional tracking error.

A plot of estimated<sup>16</sup> foot height is given in Fig. 3. Average stride duration for the steps was 431 ms. Flight times, observed as those portions of Fig. 3 where neither leg is at zero height, lasted an average of 107 ms (25% of the stride). Videos of the experiment and many additional data plots are available at [8, 18].



**Fig. 3.** Estimated height of the point feet. Flight phases occur when neither foot is at zero height.

### 5.3 Discussion

Several problematic issues related to RABBIT’s hardware did not appear until running was attempted. (For a discussion of general implementation issues of walking including unmodeled effects of the boom, gear reducers, and an uneven walking surface see [19].) Future running experiments—whether on RABBIT or another, similar mechanism—should take into account the following issues.

<sup>16</sup> When RABBIT is in flight, there is no accurate way to determine hip height. A sensor was mounted to record boom pitch angle, but due to flexing of the boom, these data were inaccurate. During the stance phase, this problem is not an issue since the lower foot may always be assumed to be on the floor.

### Boom dynamics

The perturbing effects of the boom were found to be much more significant during flight phases than during stance phases. When RABBIT is modeled as a planar system, an analysis of the three-dimensional mechanics shows that the contribution of the boom to the center of mass dynamics is significant. Specifically,  $q_5$  is no longer, in general, a cyclic variable during flight. However, if boom masses are appropriately distributed, the parabolic motion of the center of mass, as modeled in a planar system, is recovered. Unfortunately, this special mass distribution was impossible because RABBIT does not have a counterweight system.

### Walking surface

The walking surface was also a source of problems. This surface—consisting of rubber over elevated plywood supported on the edges by a wood frame—was originally built to provide a uniform, level surface. Although the surface appears uniform, walking experiments demonstrated otherwise. It was found that the surface has “fast” and “slow” areas corresponding to varying floor stiffness and coefficient of friction.

### Limited joint space

For safety, RABBIT’s joints have hard stops that limit its joint space, which, for example, prevent the shank from contacting the thigh. Although the available joint space was sufficient for walking, it became a significantly limiting factor in the design of running gaits. These hard stops prevented the swing leg from being folded close to the hip, which is a natural and desirable motion that minimizes the leg’s rotational inertia.

## 6 Conclusion

A novel approach to the control of running in planar bipeds and its first experimental implementation on RABBIT were presented. The control law is hybrid, consisting of continuous actions in the stance and flight phases, and discrete actions at the transitions between these phases. In the stance and flight phases, the controller coordinates the relative motions of the robot’s links by imposing virtual constraints at the actuated joints. At the transition from stance to flight, the controller adjusts the virtual constraints for the flight phase as a function of estimated flight duration to ensure that the former swing leg is advanced properly to take up its role as the next stance leg. At the transition from flight to stance, the controller updates the virtual constraints of the stance phase to account for the orientation of the robot at landing. For the nominal periodic running motion, the parameters of the virtual constraints

are determined by numerical optimization in order to meet actuator power limits, friction bounds, joint limits, etc. For running experiments, RABBIT's mechanical and electrical systems were modified: shock absorbers were added to the shanks; the ground contact sensors were improved; the stiffnesses of legs in the frontal plane were increased; and the hips were widened.

The main result of the experiment was the physical realization of six consecutive running steps with a human-like gait. Although continuous, stable running was not achieved, the authors are confident that this will soon change.

## Acknowledgments

The work of J.W. Grizzle and B. Morris was supported by NSF grant ECS-0322395. The work of E.R. Westervelt was supported by NSF grant CMS-0408348.

## References

1. S. P. Bhat and D. S. Bernstein. Finite-time stability of continuous autonomous systems. *SIAM J. Contr. Optim.*, 38:751–66, 2000.
2. G. Buche. ROBEA Home Page. <http://robot-rabbit.lag.ensieg.inpg.fr/English/>.
3. C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. Canudas, and J. W. Grizzle. RABBIT: a testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, October 2003.
4. C. Chevallereau and Y. Aoustin. Optimal reference trajectories for walking and running of a biped robot. *Robotica*, 19(5):557–69, September 2001.
5. C. Chevallereau, E. R. Westervelt, and J. W. Grizzle. Asymptotic stabilization of a five-link, four-actuator, planar bipedal runner. In *Proc. of the 2004 IEEE International Conference on Decision and Control, Nassau, Bahamas*, 2004.
6. C. Chevallereau, E. R. Westervelt, and J. W. Grizzle. Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. *International Journal of Robotics Research*, 24:431–464, 2005.
7. B. Espiau and A. Goswami. Compass gait revisited. In *Proc. of the IFAC Symposium on Robot Control, Capri, Italy*, pages 839–846, September 1994.
8. J. W. Grizzle. Jessy Grizzle's publications. <http://www.eecs.umich.edu/~grizzle/papers/robotics.html>.
9. J. W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on Automatic Control*, 46:51–64, January 2001.
10. J. Guckenheimer and S. Johnson. Planar hybrid systems. In *Hybrid Systems II, Lecture Notes in Computer Science*, pages 203–25. Springer-Verlag, 1995.
11. I. A. Hiskens. Stability of hybrid limit cycles: application to the compass gait biped robot. In *Proc. of the 40th IEEE Conf. Dec. and Control, Orlando, FL*, December 2001.

- 20 B. Morris, E.R. Westervelt, C. Chevallereau, G. Buche, and J.W. Grizzle
12. Y. Hürmüzlü and D. B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *International Journal of Robotics Research*, 13(1):82–92, 1994.
  13. A. Isidori. *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin, third edition, 1995.
  14. S. Kajita, T. Nagasaki, K. Kaneko, K. Yokoi, and K. Tanie. A hop towards running humanoid biped. In *Proc. of the 2004 IEEE International Conference on Robotics and Automation*, 2004.
  15. T. A. McMahon, G. Valiant, and E. C. Frederick. Groucho running. *Journal of Applied Physiology*, 62(6):2326–37, June 1987.
  16. S. G. Nersesov, V. Chellaboin, and W. M. Haddad. A generalization of Poincaré’s theorem to hybrid and impulsive dynamical systems. *Int. J. Hybrid Systems*, 2:35–51, 2002.
  17. M. H. Raibert. *Legged robots that balance*. MIT Press, Cambridge, MA, 1986.
  18. E. R. Westervelt. Eric Westervelt’s publications. <http://www.mecheng.osu.edu/~westerve/publications>.
  19. E. R. Westervelt, G. Buche, and J. W. Grizzle. Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds. *International Journal of Robotics Research*, 23(6):559–82, 2004.
  20. E. R. Westervelt, J. W. Grizzle, and C. Canudas. Switching and PI control of walking motions of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(2):308–12, February 2003.
  21. E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, January 2003.