Zero Dynamics, Pendulum Models, and Angular Momentum in Feedback Control of Bipedal Locomotion

Yukai Gong and Jessy Grizzle

Abstract—Low-dimensional models are ubiquitous in the bipedal robotics literature. On the one hand is the community of researchers that bases feedback control design on pendulum models selected to capture the center of mass dynamics of the robot during walking. On the other hand is the community that bases feedback control design on virtual constraints, which induce an exact low-dimensional model in the closed-loop system. In the first case, the low-dimensional model is valued for its physical insight and analytical tractability. In the second case, the low-dimensional model is integral to a rigorous analysis of the stability of walking gaits in the full-dimensional model of the robot. This paper seeks to clarify the commonalities and differences in the two perspectives for using low-dimensional models. In the process of doing so, we argue that angular momentum about the contact point is a better indicator of robot state than linear velocity. Concretely, we show that an approximate (pendulum and zero dynamics) model parameterized by angular momentum provides better predictions for foot placement on a physical robot (e.g., legs with mass) than does a related approximate model parameterized in terms of linear velocity. We implement an associated angular-momentum-based controller on Cassie, a 3D robot, and demonstrate high agility and robustness in experiments.

Index Terms—Bipedal robots, zero dynamics, pendulum models, angular momentum

I. INTRODUCTION

Models of realistic bipedal robots tend to be high-dimensional, hybrid, nonlinear systems. This paper is concerned with two major themes in the literature for “getting around” the analytical and computational obstructions posed by realistic models of bipeds.

On the one hand are the broadly used, simplified pendulum models [1]–[7] that provide a computationally attractive model for the center of mass dynamics of a robot. When used for control design, the fact that they ignore the remaining dynamics of the robot generally makes it impossible to prove stability properties of the closed-loop system. Despite the lack of analytical backing, the resulting controllers often work in practice when the center of mass is well regulated to match the assumptions underlying the model. Within this context, the dominant low-dimensional pendulum model by far is the so-called linear inverted pendulum model, or LIP model for short, which captures the center of mass dynamics of a real robot correctly when, throughout a step, the following conditions hold: (i) the center of mass (CoM) moves in a straight line; and (ii), the robot’s angular momentum about the center of mass ($L_c$) is zero (or constant). This latter condition can be met by designing a robot to have light legs, such as the Cassie robot by Agility Robotics [8], or by deliberately regulating $L_c$ to zero [9], [10]. When $L_c$ cannot be regulated to a small value, an MPC feedback control law based on the LIP model has been proposed to minimize zero moment point (ZMP) tracking error and CoM jerk [11]–[13]. The effects of $L_c$ can be compensated with ZMP, making a real robot’s CoM dynamics the same as those of a LIP [14]. Alternatively, $L_c$ can be approximately predicted and used for planning [15], [16].

On the other hand, the control-centric approach called the Hybrid Zero Dynamics provides a mathematically-rigorous gait design and stabilization method for realistic bipedal models [17]–[26] without restrictions on robot or gait design. In this approach, the links/joints of the robot are synchronized via the imposition of “virtual constraints”, meaning the constraints are achieved through the action of a feedback controller instead of contact forces. As opposed to physical constraints, virtual constraints can be re-programmed on the fly. Like physical constraints, imposing a set of virtual constraints results in a reduced-dimensional model. The term “zero dynamics” for this reduced dynamics comes from the original work of [27], [28]. The term “hybrid zero dynamics” or HZD comes from the extension of zero dynamics to (hybrid) robot models in [29]. A downside of this approach, however, has been that it lacked the “analytical tractability” provided by the pendulum models\textsuperscript{1}, and it requires non-trivial time to find optimal virtual constraints for a realistic model.

While CoM velocity is the most widely used variable “to

\textsuperscript{1}The approaches in [30]–[32] to build reduced-order models via embeddings is a step toward attaching physical significance to the zero dynamic models.

\begin{figure}[h]
\centering
\includegraphics[width=0.7\textwidth]{cassie.jpg}
\caption{Cassie Blue, by Agility Robotics, on the iconic University of Michigan Wave Field.}
\end{figure}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
Parameter & Value & Description \\
\hline
Mass & 10 kg & Mass of the robot \\
Height & 1.5 m & Height of the robot \\
Width & 0.5 m & Width of the robot \\
\hline
\end{tabular}
\caption{Robot Specifications}
\end{table}

\section{II. ZERO DYNAMICS}

Zero dynamics are a subset of hybrid zero dynamics and a first-order approximation of the robot’s dynamics. They are defined as the equations of motion that are zero when the system is in a steady state. In other words, zero dynamics are the equations that describe the system’s behavior when it is not changing in time.

\begin{equation}
\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})
\end{equation}

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control input, and $\mathbf{f}$ is the vector field that defines the dynamics of the system. The zero dynamics are given by the equation

\begin{equation}
\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z})
\end{equation}

where $\mathbf{z}$ is the vector of zero dynamics variables and $\mathbf{g}$ is the vector field that defines the zero dynamics.

\section{III. PENDULUM MODELS}

Pendulum models are a special case of zero dynamics in which the system is constrained to move in a plane. This allows for a simpler, more tractable model than the full-dimensional model. The pendulum model is given by

\begin{equation}
\ddot{\theta} + \frac{g}{L} \sin(\theta) = \frac{T}{I}
\end{equation}

where $\theta$ is the angular displacement, $g$ is the acceleration due to gravity, $L$ is the length of the pendulum, $T$ is the torque applied to the pendulum, and $I$ is the moment of inertia of the pendulum.

\section{IV. ANGULAR MOMENTUM}

Angular momentum is a measure of the system’s tendency to rotate about a fixed axis. In a bipedal robot, the angular momentum is given by

\begin{equation}
\mathbf{L} = I \omega + m \mathbf{r} \times \mathbf{v}
\end{equation}

where $I$ is the moment of inertia of the system, $\omega$ is the angular velocity, $m$ is the mass of the system, $\mathbf{r}$ is the position of the mass center, and $\mathbf{v}$ is the linear velocity of the system.

\section{V. VIRTUAL CONSTRAINTS}

Virtual constraints are a set of constraints that are imposed on the system in order to achieve a desired behavior. In the context of bipedal robots, virtual constraints are used to synchronize the robot’s movements and to achieve stability.

\begin{equation}
\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{g}(\mathbf{z})
\end{equation}

where $\mathbf{g}$ is the vector field of virtual constraints.
summarize the state” of a bipedal robot, angular momentum about the contact point has also been valued by multiple researchers. In [33], angular momentum is chosen to represent a biped’s state and it is regulated by stance ankle torque. In [34], the relative degree three property of angular momentum motivated its use as a state variable in the zero dynamics. In [35], [36], control laws for robots with an unactuated contact point were proposed to exponentially stabilize them about an equilibrium. In [23], angular momentum is explicitly used for designing nonholonomic virtual constraint. In [37], angular momentum is combined with the LIP model to yield a controller that stabilizes the transfer of angular momentum from one leg to the next through continuous-time (single-support-phase) control coupled with a hybrid model that captures impacts that occur at foot strike. In [38], the accuracy of the angular-momentum-based LIP model during the continuous phase is emphasized; as opposed to [37], angular momentum is allowed to passively evolve according to gravity during each single support phase, and foot placement is used to regulate the estimated angular momentum at the end of the ensuing step.

A. Objectives

The objectives of this paper are two-fold. Firstly, we seek to contribute insight on how pendulum models relate among one another and to the dynamics of a physical robot. We demonstrate that even when two pendulum models originate from the same (correct) dynamical principles, the approximations made in different coordinate representations lead to non-equivalent approximations of the dynamics of a (realistic) bipedal robot. Secondly, we seek a rapprochement of the most common pendulum models and the hybrid zero dynamics of a bipedal robot. Both of these objectives are addressed for planar robot models. The extension to 3D is not attempted here, primarily to keep the arguments as transparent as possible.

The first point that, approximations made in different coordinate representations lead to non-equivalent approximations of the dynamics of a real robot, is important in practice; hence we elaborate a bit more here, with details given in Sect. VI. Let’s only consider trajectories of a robot where the center of mass height is constant, and therefore, the velocity and acceleration of the center of mass height are both zero. In a realistic robot, the angular momentum about the center of mass, denoted by $L_c$, contributes to the longitudinal evolution of the center of mass, though it is routinely dropped in the most commonly used pendulum models. Can dropping $L_c$ have a larger effect in one simplified model than another?

In the standard 2D LIP model, the coordinates are taken as the horizontal position and velocity of the center of mass and the time derivative of $L_c$ is dropped from the differential equation for the velocity. It follows that the term being dropped is a high-pass filtered version of $L_c$, due to the derivative. Moreover, the derivative of $L_c$ is directly affected by the motor torques, which are typically “noisy” (have high variance) in a realistic robot. On the other hand, in a less frequently used representation of a 2D inverted pendulum [23], [37]–[39], the coordinates are taken as angular momentum about the contact point and the horizontal position of the center of mass, and $L_c$ is dropped from the differential equation for the position. In this model, $L_c$ (and not $\dot{L}_c$) shows up in the second derivative of the angular momentum about the contact point. It follows that variations in $L_c$ are low-pass filtered in the second representation as opposed to high-pass filtered in the first, and thus, speaking intuitively, neglecting $L_c$ should induce less approximation error in the second model. More quantitative results are shown in the main body of the paper.

In this paper, we focus on the underactuated single support phase dynamics and assume an instantaneous double support phase. The reader is referred to existing literature on how pendulum models [40], [41] and Hybrid Zero Dynamics [42]–[44] handle non-instantaneous double support phases; the topic is not discussed in this paper. We provide models for a robot with non-trivial feet. While most of the results are demonstrated for robots with point feet, we briefly show that the conclusions we obtained for robots with point feet still apply to robots with non-trivial feet. Further studies of how pendulum models and Hybrid Zero Dynamics handle non-trivial feet can be found in [11], [45]–[47].

B. Summary of Main Contributions

This paper makes the following contributions to pendulum models and zero dynamics:

- Starting from a common physically correct set of equations for real robots, we sequentially enumerate the approximations made to arrive at various reduced order pendulum models. This is done for bipedal robots with and without ankle torque.
- Several advantages of using angular momentum about the contact point as a key variable to summarize the dynamics of a bipedal robot are discussed and demonstrated.
- We show that a pendulum model parameterized by center of mass horizontal position and angular momentum about the contact point provides a higher fidelity representation of a physical robot than does the standard LIP model, which is a pendulum model parameterized by center of mass horizontal position and velocity. We identify the source of improvement as how the angular momentum about the center of mass is treated in the two models.
- For a set of non-holonomic virtual constraints, we derive the zero dynamics in a set of coordinates compatible with an angular-momentum-based pendulum model. We provide a precise sense in which the pendulum model is an approximation to the zero dynamics and demonstrate what this means for closed-loop stability of the full-order robot model.
- We formulate a foot placement strategy based on a high fidelity, one-step-ahead prediction of angular momentum about the contact point.
- We demonstrate that the resulting controller achieves highly dynamic gaits on a Cassie-series bipedal robot.

The last two contributions were introduced in [38] and are used here to demonstrate the utility of the proposed results. We will use both Rabbit [48] and Cassie Blue, shown in Fig. 1,
to illustrate the developments in the paper. Experiments will be conducted exclusively on Cassie.

Rabbit is a 2D biped with five links, four actuated joints, and a mass of 32 Kg; see Fig. 2. Each leg weighs 10 Kg; with 6.8 Kg on each thigh and 3.2 Kg on each shin. The Cassie robot designed and built by Agility Robotics weighs 32 Kg. It has 7 deg of freedom on each leg, 5 of which are actuated by motors and 2 are constrained by springs; see Fig. 2. A floating base model of Cassie has 20 degrees of freedom. Each foot of the robot is blade-shaped and provides 5 holonomic constraints when firmly in contact with the ground. Though each of Cassie’s legs has approximately 10 Kg of mass, most of the mass is concentrated on the upper part of the leg. In this regard, the mass distribution of Rabbit is more typical of current bipedal robots seen in labs, which is why we include the Rabbit model in the paper.

C. Organization
The remainder of the paper is organized as follows. Sect. II introduces the full hybrid model of a bipedal robot with instantaneous double support phases. It also provides the robot’s center of mass dynamics in different coordinates. Sect. III summarizes several desirable properties of angular momentum about contact point. Sect. IV goes through the approximations that must be made to the center of mass dynamics in order to arrive at the most popular pendulum models. The approximation errors of the resulting pendulum models are analyzed and compared. Sect. V uses a one-step-ahead prediction of angular momentum to decide foot placement. This provides a feedback controller based on predicted angular momentum that will stabilize a 3D pendulum. Sect. VI provides background on a feedback control paradigm based on virtual constraints and the attendant zero dynamics. One of the pendulum models is shown to be a clear approximation to the zero dynamics, and the implications for stability analysis are discussed and illustrated by numerical computation. In Sect. VII, we design the virtual constraints for a Cassie-series bipedal robot. In Sect. VIII, we provide our path to implementing the controller on Cassie Blue. Additional virtual constraints are required beyond a path for the swing foot, and we provide “an intuitive” method for their design.

Sect. IX shows the results of experiments. Conclusion are give in Sect. X.
B. Full Dimensional Hybrid Model

In the above, we implicitly assumed left-right symmetry in the robot so that we could avoid the use of two single-support models—one for each leg playing the role of the stance leg—by relabeling the robot’s coordinates at impact, thereby swapping their roles. Immediately after swapping, the former swing leg is in contact with the ground and is poised to take on the role of the stance leg. The result of the impact and the relabeling of the states provides an expression

\[ x^+ = \Delta(x^-) \]

where \( x^+ := (q^+; \dot{q}^+) \) (resp. \( x^- := (q^-; \dot{q}^-) \)) is the state value just after (resp. just before) impact and

\[ \Delta(x^-) := \begin{bmatrix} \Delta_q(q^-) \\ \Delta_\dot{q}(q^-) \end{bmatrix}. \]

A detailed derivation of the impact map is given in [17], showing that it is linear in the generalized velocities.

A hybrid model of walking is obtained by combining the single support model and the impact model to form a system with impulse effects [51]. A non-instantaneous double support phase can be added [42], [44], but we choose not to do so here. Even though the mechanical model of the robot is time-invariant, we will allow feedback controllers for (2) that are time varying. So that the hybrid model in closed loop can be analyzed with tools developed for time-invariant hybrid systems, we do the standard “trick” of adding time as a state variable via \( \dot{\tau} = 1 \). The guard condition (aka switching set) for terminating a step is

\[ S := \{ (q, \dot{q}) \in TQ \mid \dot{p}^e_{sw}(q) = 0, \dot{p}^e_{sw}(q, \dot{q}) < 0 \}, \]

where \( \dot{p}^e_{sw}(q) \) is the vertical height of the swing foot. It is noted that \( S \) is independent of time. Combining (2), (3) with the guard set and time gives the hybrid model

\[ \Sigma : \begin{cases} \dot{x} = f(x) + g(x)u & x^- \notin S \\ \dot{\tau} = 1 \\ x^+ = \Delta(x^-) \\ \tau^+ = 0. \end{cases} \]

It is emphasized that the guard condition for re-setting the “hybrid time variable”, \( \tau \), is determined by foot contact.

C. Center of Mass Dynamics in Single Support

While (2) is typically high dimensional and nonlinear, standard mechanics yields simpler equations for the evolution of the center of mass. For succinctness, we only consider the planar case and define the following variables:

- \((x_c, z_c)\) : CoM position in the frame of the contact point.
- \(v_c\) : CoM velocity in \(x\)-direction. The velocity in \(z\)-direction is denoted by \(\dot{z}_c\).
- \(L_c\) : \(y\)-component of Angular momentum about CoM.
- \(L\) : \(y\)-component of Angular momentum about contact point.
- \(u_a\) : ankle torque at the contact point.

In addition, we note the following (standard) result

\[ L = L_c + m \left[ \frac{x_c}{z_c} \land \frac{\dot{x}_c}{\dot{z}_c} \right] \]

where \(\land\) is the 2D version of cross product

\[ \begin{bmatrix} x_c \\ z_c \end{bmatrix} \land \begin{bmatrix} x_c \\ \dot{x}_c \\ \dot{z}_c \end{bmatrix} := \begin{bmatrix} x_c \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} x_c \\ \dot{x}_c \\ \dot{z}_c \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \]

We refer to (7) as the angular momentum transfer formula because it relates angular momentum determined about two different points.

In the following, we provide the CoM dynamics for two sets of coordinates

- \((x_c, v_c)\)
- \((x_c, L)\), and
- \((\theta_c, L)\),

where \(\theta_c := \text{atan}(x_c/z_c)\)

and we assume that \(z_c > 0\). We will subsequently dedicate Sect. IV to establishing connections between pendulum models and zero dynamics, which will allow the zero dynamics to be intuitively grounded in physics.

Case 1: \((x_c, v_c)\) Horizontal Position and Velocity Differentiating (7) and using \(v_c = \dot{x}_c\) results in

\[ \dot{x}_c = v_c \]

\[ \dot{v}_c = g \frac{x_c}{z_c} + \frac{\dot{x}_c}{z_c} - L_c \frac{mz_c + u_a}{mz_c}. \]

In general, \(z_c\) depends on \(q, \dot{z}_c\) and \(L_c\) depend on both \(q\) and \(\dot{q}\). While \(L_c\) and \(\dot{z}_c\) depend on \(q, \dot{q}\), and the motor torques \(u\), it is more typical to replace the motor torques by the ground reaction forces. In particular, one uses \(\ddot{z} = g - \frac{1}{m} \dot{F}_z\) and \(\dot{L}_c := \frac{d}{dt} L_c = x_c \dot{F}_x - z_c \dot{F}_z + u_a\), where \(F_x\) and \(F_z\) are the horizontal and vertical components of the ground reaction forces. In turn, the ground reaction forces can be expressed as functions of \(q, \dot{q}\), and the motor torques, \(u\).

Case 2: \((x_c, L)\) Angular Momentum and Horizontal Position: Manipulating (7) and using \(\dot{L} = mgx_c + u_a\) results in

\[ \dot{x}_c = \frac{L}{mz_c} + \frac{\dot{z}_c}{z_c} x_c - L_c \frac{mz_c}{mz_c} \]

\[ \dot{L} = mgx_c + u_a. \]

The remarks made above on \(z_c, \dot{z}_c\), and \(L_c\) apply here as well.

Case 3: \((\theta_c, L)\) Alternative absolute angle (cyclic variable): Differentiating (8) yields

\[ \dot{\theta}_c = \frac{L - L_c}{mr_\theta^2(q_b)}. \]

Combining (7) and (11) yields

\[ \dot{\theta}_c = \frac{L - L_c}{mr_\theta^2} \]

\[ \dot{L} = mg r_c \sin(\theta_c) + u_a. \]

It is remarked that the derivatives of the generalized coordinates only appear through \(L_c\). In the following, we will keep the discussion primarily focused on (10), but most of the results apply to (12) as well; see Appendix A.
III. Angular Momentum about Contact Point

In this paper, we are focusing on the angular momentum about the contact point, \( L \), as a replacement for the center of mass velocity, \( v_c \), which is used as an indicator of walking status in many other papers [7], [26], [52], [53]. Specific to this paper, \( L \) is also a state of the zero dynamics. Before we proceed to that, it is beneficial to explain why \( L \) can replace \( v_c \), summarize some general properties of \( L \), and highlight some of its advantages versus \( v_c \). More specific advantages of using \( L \) in the zero dynamics and the LIP model will be discussed in later sections.

We first need to answer why \( L \) can replace \( v_c \) as an indicator of walking. The relationship between angular momentum and linear momentum for a 3D bipedal robot is

\[
L = L_c + p_c \wedge mv_c, \tag{13}
\]

where \( L_c \) is the angular momentum about the center of mass, \( v_c \) is the linear velocity of the center of mass, \( m \) is the total mass of the robot, and \( p_c \) is the vector emanating from the contact point to the center of mass.

For a bipedal robot that is walking instead of doing somersaults, it is reasonable to focus on gaits where the angular momentum about the center of mass oscillate about zero (e.g., arms are not rotating as in a flywheel). The oscillating property of \( L_c \) is discussed in [54], [55]. When \( L \) oscillates about zero, (13) implies that the difference between \( L \) and \( p_c \wedge mv_c \) also oscillates about zero, which we will write as

\[
L - p_c \wedge mv_c = L_c \text{ oscillates about } 0. \tag{14}
\]

From (14), we see that we approximately obtain a desired linear velocity by regulating \( L \). Hence, in walking robots without a flywheel, one can replace the control of linear velocity with control of angular momentum about the contact point.

What are there advantages to using \( L \)?

(a) The first advantage of controlling \( L \) is that it provides a more comprehensive representation of current walking status because it is the sum of angular momentum about the center of mass, \( L_c \), and linear momentum, \( p_c \wedge mv_c \).

From (9), we see that there exists momentum transfer between these two quantities. If \( L_c \) increases, it must “take” some momentum away from \( v_c \), and vice versa. For normal bipedal walking, \( L_c \) oscillates about zero. \( L_c \) functions to store momentum [56], but importantly it can only store it for a short amount of time. When designing a foot placement strategy, it is important to take the “stored” momentum into account.

When balancing on one foot for example, some researchers plan \( L_c \) and \( v_c \) separately [57], or use \( L_c \) as an input to regulate balance by waving the torso, arms, or swing leg [58], [59] or even a flywheel [60]. Here, instead of moving limbs to generate a certain value of \( L_c \), we view \( L_c \) as a result of the legs and torso moving to fulfill other tasks. In this paper, we observe \( L_c \) and take it into consideration through \( L \) and do not seek to regulate it directly as an independent quantity.

(b) Secondly, because \( \dot{L} = mgx_c + u_a \) depends only on the CoM position, it follows that \( L \) has relative degree three with respect to all inputs except the stance ankle torque, where it has relative degree one. Consequently, the evolution \( L \) is only weakly affected by motor torques of the body, that is \( u_b \), during a step. In Fig. 4 (a) and (d) and Fig. 5 (a) we see that the trajectory of \( L \) consistently has a convex shape when stance ankle torque is zero, irrespective of model or speed. We’ll see later the same property in experimental data.

(c) The discussion so far has focused on the single support phase of a walking gait. Bipedal walking is characterized by the transition between left and right legs as they alternately take on the role of stance leg (aka support leg) and swing leg (aka non-stance leg). In double support, the transfer of angular momentum between the two contact points satisfies

\[
L_2 = L_1 + p_{2\rightarrow1} \wedge mv_c, \tag{15}
\]

where \( p_{2\rightarrow1} \) is the vector from point 2 (the new stance leg position) to point 1 (the previous stance leg position). Hence, the change of angular momentum between two contact points depends only on the vector defined by the two contact points and the center of mass velocity. In particular, angular momentum about a given contact point is invariant under the impulsive force generated at that contact point. Consequently, we can easily determine the angular momentum about the new contact point by (15) when impact happens without resorting to approximating assumptions about the impact model. Moreover, if \( z \) is zero and the ground is level, then \( p_{2\rightarrow1} \wedge mv_c = 0 \), and hence \( L_2 = L_1 \). We note that pendulum models parameterized with CoM velocity often assume continuity at impact, which is not generally true for real robots.

Figure 4 shows the evolution of \( L, v_c, L_c \), and \( \dot{L} \) during a step for both Cassie and Rabbit, when walking speed is about 2 m/s, \( z \approx 0 \), and no stance ankle torque is applied. Figure 5 shows the evolution of \( L, v_c, L_c \) for Rabbit walking at a range of speeds from -1.8 m/s to 2.0 m/s. Figure 4 and 5 also show the continuity property of \( L \) at impact.
where \( v_p = \frac{L_c}{mH} \), which is more directly comparable to (16). In this paper we frequently plot \( L \) scaled by the coefficient \( \frac{1}{mH} \), so that it can be more directly compared to \( v_c \) (same units and similar magnitudes).

At this point, no approximations have been made and both models are valid everywhere that \( z_c(q) \equiv H \). Hence, the two models are still equivalent representations of the center of mass dynamics for all trajectories satisfying \( z_c(q) \equiv H \). We'll next argue that the models are not equivalent when it comes to approximations.

Dropping the \( \dot{L}_c \) term in (16) results in:

\[
\begin{align*}
\dot{x}_c &= v_c \\
\dot{v}_c &= \frac{g}{H} x_c - \frac{L_c}{mH} + \frac{u_a}{mH},
\end{align*}
\]

(19)

This is the well-known LIP model proposed by [61].

Dropping \( L_c \) in (17) results in

\[
\begin{align*}
\dot{x}_c &= \frac{L}{mH} \\
\dot{L} &= mgx_c + u_a,
\end{align*}
\]

(20)

which is used in [37], [38]. In [37], (20) is used instead of (19) so that the easy “update” property of \( L \) at impact can be used. In this paper, we demonstrate that during the continuous phase, the states of (20) much more accurately capture the evolution of \( (x_c, L) \) in a real robot than the states of (19) capture the evolution of \( (x_c, v_c) \). Moreover, we will make
the true trajectory of $L$ shows the prediction of $L$ of $L$ the values of $v$ $(20)$ are equivalent, because denote it by ALIP, where A stands for Angular Momentum.

Controller. To distinguish the model $(20)$ from $(19)$, we will use of this improved accuracy in the design of a feedback controller. To distinguish the model $(20)$ from $(19)$, we will use of this improved accuracy in the design of a feedback

For a robot with a point mass, the two models $(19)$ and $(20)$ are equivalent, because $L_c$ is then identically zero. For a real robot with $L_c$ and $L$ that are nonnegligible, however, we argue that $(20)$ is more accurate than $(19)$ primarily because of three properties,

(a) Relative Amplitude. Based on our observations, the ratio of $L_c/mg\dot{x}_c$ is much larger than $L_c/L$ over a wide range of walking velocities; thus the simplification $(20)$ introduces relatively less error than $(19)$.

(b) Relative degree. $L$ has relative degree two with respect to $L_c$ and three with respect to $\dot{L}_c$, whereas $v_c$ has relative degree one with respect to $L_c$. Because integration is a form of low-pass filtering, the lower relative degree makes $v_c$ more sensitive to the omission of the $L_c$ term.

(c) $L_c$ oscillates about zero. What makes $(20)$ even more accurate is that, based on our own observation and references $(54)$, $(55)$, the sagittal plane component of $L_c$ oscillates about zero for periodic and non-periodic gaits. The oscillation of $L_c$ results in the effect of $L_c$ on $x_c$ roughly averaging out to zero over a step.

In Fig. 6, we have used the models $(19)$ and $(20)$ to predict the values of $v_c$ and $L$ at the end of a step. We plot $\frac{L}{mH}$ instead of $L$ to make the scale and units comparable. The blue line is the true trajectory of $L$ (resp. $v_c$) during a step. The red line shows the prediction of $L$ (resp. $v_c$) at the end of each of step, at each moment throughout a step, based on the instantaneous values of $x_c$ and $L (v_c)$ at that moment. The red line would be perfectly flat if $(20)$ and $(19)$ perfectly captured the evolution of $L (v_c)$, respectively, in the full simulation model, and the flatter the estimate, the more faithful is the representation.

The prediction errors of $(19)$ and $(20)$ caused by neglecting $L_c$ and $\dot{L}_c$, respectively, satisfy

\[ \dot{x}_c = v_c \]  
\[ \dot{v}_c = g \frac{L_c}{m} - \frac{\dot{L}_c}{m} \frac{mH}{L_c} \]

and

\[ \dot{x}_c = \frac{L_c}{mH} - \frac{\dot{L}_c}{mH} \]

\[ \dot{L}_c = mgx_c, \]

where $(x_c, L_c)$ are the differences in the trajectories of $(19)$ and $(18)$; similarly, $(x_c, v_c)$ are the differences in the trajectories of $(20)$ and $(16)$. Direct solution of these two sets of differential equations for zero initial conditions leads to

\[ v_c(t_2, t_1) = e_1(t_2, t_1) \]
\[ e_2(t_2, t_1) + e_3(t_2, t_1) \]
\[ \frac{L_c(t_2, t_1)}{mH} = e_2(t_2, t_1), \]
where
\[
e_1(t_2, t_1) = -\frac{1}{mH} \int_{t_1}^{t_2} \cosh(\ell(t_2 - \tau))\dot{L}_c(\tau) \, d\tau
\]
\[
e_2(t_2, t_1) = -\frac{1}{mH} \int_{t_1}^{t_2} \ell \sinh(\ell(t_2 - \tau))L_c(\tau) \, d\tau
\]
\[
e_3(t_2, t_1) = -\frac{1}{mH} (L_c(t_2) - \cosh(\ell(t_2 - t_1))L_c(t_1)).
\]

Figure 7 shows the (relative) sizes of these error terms. If we view \(L_c\) as a disturbance and prediction error as an output in (21) and (22), we obtain the corresponding Laplace transforms and Bode plots shown in Fig. 8.

B. Simulation Comparison

We compare controllers designed on the basis of the ALIP and LIP models in simulation. The results shown in Fig 9 demonstrate the advantage of using ALIP over LIP for controller design. The initial hip velocity is set to 0.5 m/s and hip position is centered over the contact point. The goal of each controller is to regulate \(v_c\) (resp., \(L\)) to zero, with foot placement as the decision variable and step duration constant. In the plots, we observe that the ALIP-based controller regulates \(L\) closely to zero and thus has an average \(v_c\) close to zero, while the LIP-based controller is unable to regulate \(v_c\) effectively. The reason is that, at the end of a step, the linear momentum was transferred to centroidal angular momentum \(L_c\) due to the movement of Rabbit’s heavy legs (see Eqn (18)), resulting in a small \(v_c\), which misleads the LIP controller into choosing a small foot displacement. In the ALIP model, \(L\) is less affected by momentum transfer between \(v_c\) and \(L_c\) because \(L\) captures their sum, and thus the ALIP model suggests better foot placement. Though with a LIP controller it is possible to regulate velocity through ZMP (ankle torque) during continuous phase, we argue that with an ALIP controller, the capability of ZMP can be reserved for better purposes than compensating for model error.

C. Non-zero Ankle Torque

In previous subsections we have demonstrated the accuracy of pendulum model parameterized with \(L\) when ankle torque is zero. According to Eqn (10), the effect of \(L_c\) and \(u_a\) on the system are independent due to the superposition property. So if dropping \(L_c\) term has little effect on the model accuracy when \(u_a\) is zero, it should still has little effect on the model accuracy when \(u_a\) is non-zero. Though \(L_c\) trajectory itself will be changed when \(u_a\) is non-zero, its pattern is still similar. Here for completeness we run a simulation on Rabbit. The results are shown in Fig. 10

V. STABILIZING THE ALIP MODEL

In this section, we provide a means to regulate angular momentum about the contact point to approximately achieve a desired walking speed. Specifically, the ALIP model (20) is used to form a one-step ahead prediction of angular momentum, \(L\). In combination with the angular momentum transfer formula (15), a feedback law results for where to place the swing foot at the end of the current step so as to achieve a desired angular momentum at the end of the ensuing step. In robot locomotion, this is typically called “foot placement control” [7], [62]–[64].

\(L\) can also be controlled through vertical CoM velocity at the end of a step [37], adjusting the step duration [61], [65], ankle torque during continuous phase [33], which has a relative degree one relation to \(L\), or by other torques applied at the body coordinates during the continuous single support phase [35], [36], [66], which have a relative degree three relation to \(L\). We summarize these general methods\(^2\) to regulate \(L\) in Table I.

| Continuous Phase | Relative Degree Three | \(L_c, z_c\) |
|------------------|----------------------|-------------|
| Relative Degree One | \(u_a\) |

| Transition Event | Initial condition for next continuous phase | \(x(0), L(0)\) |
|------------------|-------------------------------------------|---------------|

| Phase duration | Step Time |
|----------------|-----------|

A. Gait assumptions

When designing the foot placement controller, we assume the gait of the robot is controlled such that:
(a) the height of the center of mass is constant, that is \(z_c \equiv H > 0\);
(b) each step has constant duration \(T > 0\); and
(c) a desired swing leg horizontal position, \(p_{SW}^{des}\) can be achieved at the end of the step.

We’ll explain how to accomplish these objectives via the method of virtual constraints in Sections VI and VII.

B. Notation

We distinguish among the following time instances when specifying the control variables.
- \(T\) is the step time.
- \(T_k\) is the time of the \(k\)th impact and thus equals \(kT\).
- \(T_k^-\) is the end time of step \(k\), so that
- \(T_k^+\) is the beginning time of step \(k + 1\) and \(T_{k+1}^-\) is the end time of step \(k + 1\).
- \((T_k^- - t) = (T - T_k^-)\) is the time until the end of step \(k\).

The superscripts + and – on \(T_k\) are necessary because of the (potential) jump in a trajectory’s values from the impact map; see [67]. As shown in Fig. 11, \(x(T_k^-)\) is the limit from the left of the model’s solution at the time of impact, in other words

\(^2\)With the exception of the method based on regulating \(L_c\), each of the methods listed in Table I can be applied to a pendulum model parameterized with \(L\). For humans, \(L_c\) is mainly utilized when we are about to fall off a support structure, such as when balancing on a tightrope. In bipedal robots without a flywheel, the effect of \(L_c\) on \(L\) is weak, due to multiple factors such as kinematic constraints, torque limits, and being relative degree three with respect to body torques. \(L_c\) is an ineffective means of regulating \(L\) for the same reasons that dropping \(L_c\) in the ALIP model does not introduce significant inaccuracy.
Fig. 6: Comparison of the ability to predict velocity vs angular momentum at the end of a step. The instantaneous values are shown in blue and the predicted value at the end of the step is shown in red, where a perfect prediction would be a flat line that intercepts the terminal point of the blue line. The most crucial decision in the control of a bipedal robot is where to place the next foot fall. In the standard LIP controller, the decision is based on predicting the longitudinal velocity of the center of mass. In Sect. V we use angular momentum about the contact point. We do this because on realistic bipeds, a LIP-style model provides a more accurate and reliable prediction of \( L \) than \( v_c \). The comparison is more significant on Rabbit, whose leg center of mass is further away from the overall center of mass.

Fig. 7: A plot of the error terms in (23) and (24) resulting from dropping \( \dot{L}_c \) and \( L_c \), respectively, for the Rabbit model walking at 2 m/s. The take-home message is that of the terms \( e_2(t_1, t_2) + e_3(t_1, t_2) \) in (23) comprising the velocity error of the LIP model, the term \( e_3(t_1, t_2) \) shown in the yellow line contributes by far the largest portion of the total error shown by the blue line. The error of the ALIP model, however, is given only by \( e_2(t_1, t_2) \), which results in the significantly reduced prediction error shown by the red line.

it’s value “just before” impact, while \( x(T_k^+) \) is the limit from the right of the model’s solution at the time of impact, in other words it’s value “just after” impact.

With this notation, the reset map for the ALIP becomes

\[
x_c(T_k^+) = p_{sw\rightarrow CoM}(T_k^-)
\]

\[
L(T_k^+) = L(T_k^-), \quad (25)
\]

after noting that (15) simplifies to \( L \) being constant across impacts when the ground is level and the vertical velocity of the center of mass is zero.

We remind the reader that

- \( p_{st\rightarrow CoM} \), \( p_{sw\rightarrow CoM} \) are the vectors emanating from stance/swing foot to the robot’s center of mass. The stance foot defines the current contact point, while the swing foot is defining the point of contact for the next impact and is therefore a control variable.

- Also, for the implementation of the control law on the 3D biped Cassie in Sect. VII, we need to distinguish between \( L^y \) and \( L^x \), the \( y \) and \( x \) components of the angular momentum (sagittal and frontal planes), respectively.

C. Foot placement in longitudinal direction

The control objective will be to place the swing foot at the end of the current step so as to achieve a desired value of angular momentum at the end of the ensuing step. The need to regulate the angular momentum one-step ahead of the current step, instead of during the current step, is because in (20) \( L \) is
Fig. 9: Simulation results of Rabbit with controllers based on LIP and ALIP, following an identical design philosophy, based on foot placement. The details of the controller are described in Sec V and Sec VII. The controller based on the ALIP model is much more effective in regulating velocity to zero.

Fig. 10: Comparison of the ability to predict velocity vs angular momentum at the end of a step in a model with ankle torque $u_a = 30 \sin(2\pi \tau / T)$, where $\tau$ varies from 0 to $T$ during a step. The instantaneous values are shown in blue and the predicted value at the end of the step is shown in red. Because ankle torque is an input, we assume its trajectory is known when making predictions. For comparison purposes with Fig. 6, the ankle torque is chosen to be sufficiently large so that gravity is no longer dominant in $\dot{L} = mgx_c + u_a$ and the trajectory of $L$ is no longer convex.

Passive without ankle torque, in other words, it is not affected by the control actions of the current step. The only way to act on its states is through the transition events.

The closed-form solution of (20) at time $T$ and initial time $t_0$ is

$$
\begin{bmatrix}
x_c(T) \\
L^y(T)
\end{bmatrix} = A(T - t_0) \begin{bmatrix}
x_c(t_0) \\
L^y(t_0)
\end{bmatrix},
$$

(26)

![Simulation results of Rabbit with controllers based on LIP and ALIP.](image)

![Comparison of the ability to predict velocity vs angular momentum at the end of a step in a model with ankle torque.](image)
which achieves
\[ (L_y^{\text{des}} - \hat{L}_y(T_{k+1}, t)) = \alpha (L_y^{\text{des}} - \hat{L}_y(T_k, t)) \] (33)
for \( \alpha \in [0, 1] \). Hence, for \( \alpha = 0 \), (32) reduces to (31).

D. Stability Analysis of the ALIP for \( L^{\text{des}} \)

Consider the ALIP model (20) with zero ankle torque and rest map (25). To compute the Poincaré map, we take the Poincaré section as \( S := \{(x_c, L, \tau) \mid \tau = 0^+\} \), which is the set of states just after impact. Computing (20) over one step and using swing foot position with respect to the center of mass, \( u_{fp} \), as an input, yields,
\[
\begin{bmatrix}
  x_c(T^+) \\
  L(T^+)
\end{bmatrix} =
\begin{bmatrix}
  0 & 0 & x_c(0^+) \\
  mH\ell \sinh(\ell T) & \cosh(\ell T) & L(0^+)
\end{bmatrix} +
\begin{bmatrix}
  0 \\
  1
\end{bmatrix} u_{fp}(T^-).
\] (34)

Next, applying the feedback law (32) with \( L^{\text{des}} \) a constant results in the Poincaré map being
\[
\begin{bmatrix}
  x_c(T^+) \\
  L(T^+)
\end{bmatrix} =
\begin{bmatrix}
  \alpha - \cosh(\ell T) & \alpha - \cosh(\ell T) \cosh(\ell T) \\
  mH\ell \sinh(\ell T) & \cosh(\ell T)
\end{bmatrix}
\begin{bmatrix}
  x_c(0^+) \\
  L(0^+)
\end{bmatrix}
\] +
\begin{bmatrix}
  1 - \alpha \\
  mH\ell \sinh(\ell T)
\end{bmatrix} L^{\text{des}}.
\] (35)

The Poincaré map has fixed point
\[
\begin{bmatrix}
  x_c^* \\
  L^*
\end{bmatrix} =
\begin{bmatrix}
  1 - \cosh(\ell T) \\
  mH\ell \sinh(\ell T)
\end{bmatrix} L^{\text{des}}
\] independent of \( \alpha \) and the eigenvalues of the Poincaré map are \( (\alpha, 0) \). Hence, for all \( 0 \leq \alpha < 1 \), the fixed point is exponentially stable and moreover, (35) is bounded-input bounded-state stable with respect to the command, \( L^{\text{des}} \).

E. Lateral Control and Turning

From (17), the time evolution of the angular momentum about the contact point is decoupled about the \( x- \) and \( y- \) axes. Therefore, once a desired angular momentum at the end of each step is given, Lateral Control is essentially identical to Longitudinal Control and (31) can be applied equally well in the lateral direction. The question becomes how to decide on \( L_x^{\text{des}}(T_{k+1}) \), since it cannot be simply set to zero for walking with a non-zero stance width.

For walking in place or walking with zero average lateral velocity, it is sufficient to obtain \( L_x^{\text{des}} \) from a periodically oscillating LIP model,
\[
L_x^{\text{des}}(T_{k+1}) = \pm \frac{1}{2} mH\ell \frac{\ell \sinh(\ell T)}{1 + \cosh(\ell T)},
\] (37)
where \( W \) is the desired step width. The sign is positive if next stance is left stance and negative if next stance is right stance. Lateral walking can be achieved by adding an offset to \( L_x^{\text{des}} \).

To enable turning, we assume a target direction is commanded and associate a frame to it by aligning the \( x- \) axis with the target direction while keeping the \( z- \) axis vertical. To achieve turning, we then define the desired angular momentum \( L_y^{\text{des}} \) and \( L_z^{\text{des}} \) in the new frame and use the hip yaw-motors to align the robot in that direction.

VI. PENDULUM MODELS, ZERO DYNAMICS, AND OVERALL SYSTEM STABILITY

This section establishes connections between the pendulum models of Sec. II-C and the swing phase zero dynamics as developed in [17], or more precisely, approximations of the zero dynamics. This is accomplished by analyzing how the zero dynamics are driven by the states of a bipedal robot’s full-order model and its feedback controller when the closed-loop system is evolving off the zero dynamics manifold. As a main contribution, the analysis will yield conditions under which the driving terms are small and hence do not adversely affect the stability predictions associated with the exact zero dynamics. A secondary contribution of the section will be a presentation of the swing phase zero dynamics for a more general set of “virtual constraints” than those developed in [17], [23], [68].

A. Intuitive Background

An initial sense of the meaning and mathematical foundation of the swing phase zero dynamics can be gained by considering a floating-base model of a bipedal robot, and then its pinned model, that is, the model with a point or link of the robot, such as a leg end or foot, constrained to maintain a constant position respect to the ground. The given contact constraint is holonomic and constant rank, and thus using Lagrange multipliers (from the principle of virtual work), a reduced-order model compatible with the (holonomic) contact constraint is easily computed. When computing the reduced-order model, no approximations are involved, and solutions of the reduced-order model are solutions of the original floating-base model, with inputs (ground reaction forces and moments) determined by the Lagrange multiplier.

Virtual constraints are relations (i.e., constraints) on the state variables of a robot’s model that are achieved through the action of actuators and feedback control instead of physical contact forces. They are called virtual because they can be re-programmed on the fly without modifying any physical connections among the links of the robot or its environment. We use virtual constraints to synchronize the evolution of a robot’s links, so as to create exponentially stable motions. Like physical constraints, under certain regularity conditions, they induce an exact low-dimensional invariant model, called the zero dynamics, due to the highly influential paper [28].

Each virtual constraint imposes a relation between joint variables, and by differentiation with respect to time, a relation between joint velocities. As a consequence, for the virtual constraints studied in this paper, the dimension of the zero dynamics is the number of states in the robot’s (pinned) model minus twice the number of virtual constraints (which can be at most the number of independent actuators). As explained in [69], the computation of the motor torques to impose virtual constraints parallels the Jacobian computations for the ground reaction forces in a pinned model.

B. Allowing Non-holonomic, Time-varying Virtual Constraints

In this section, we choose \( L \) as one of the states of the zero dynamics. So that fully actuated and underactuated biped
models can be addressed simultaneously, we suppose that the torque distribution matrix \( B(q) \) in (1) can be split so that
\[
B(q)u =: B_a(q)u_a + B_b(q)u_b,
\]
where \( u_a \) is the torque affecting the stance ankle as in Sect. II-C and \( u_b \in \mathbb{R}^n \) are actuators affecting the body coordinates, \( q_b \). When the robot is underactuated, \( B_a(q) \) is an empty column vector.

We define \( n \) virtual constraints as an output zeroing problem of the form
\[
y = h(q, L, \tau) = h_0(q) - h_d(x_c, L, \tau)
\]
where \( \tau \) captures time dependence. As in Sect. II-C, we use \( L \) instead of other functions of \( \dot{q} \) because \( L \) has relative degree three with respect to all actuators except stance ankle torque, while \( \dot{q} \) has relative degree one. Hence, the relative degree of \( y \) is determined by \( q \) once \( u_a \) is fixed. Indeed, while \( u_b \) is used for imposing the virtual constraints, \( u_a \) can be used for shaping the evolution of \( x_c \) and \( L \) directly. We assume a feedback law, \( u_a \), of the form
\[
u_a = \alpha(x_c, L, \tau)
\]
and note that \( u_a \) should respect relevant ankle torque limits and ZMP constraints when \( y = 0 \).

Following [17], [27], [28], we make the following specific regularity assumptions for the virtual constraints:

**A1:** \( h \) is at least twice continuously differentiable and \( u_a \) is at least once differentiable.

**A2:** The virtual constraints (39) are designed to identically vanish on a desired nominal solution (gait) \((\bar{q}(t), \dot{q}(t), \bar{u}(t))\) of the dynamical model (6) with \( \tau(t) = t \), where the solution meets relevant constraints on motor torque, motor power, ground reaction forces, and work space. To be clear, \( y \) vanishing means
\[
h_0(\dot{q}) - h_d(x_c, \bar{L}, \tau) \equiv 0
\]
for \( 0 \leq t \leq T \), where \( \bar{L}(t) \) is the angular momentum about the contact point, evaluated along the trajectory.

**A3:** The decoupling matrix
\[
A(q) := \frac{\partial h(q, L, t)}{\partial \dot{q}} D^{-1}(q) B_a(q)
\]
is square and invertible along the nominal trajectory, so that, from [27] and [70], by treating \( u_a \) as a known signal, there exists a feedback controller of the form
\[
u_a =: \gamma(q, \dot{q}, \tau) + \gamma_a(q, L, \tau) u_a
\]
resulting in the closed-loop dynamics
\[
\ddot{y} + K_d \dot{y} + K_p y = 0,
\]
with \( K_d > 0 \) and \( K_p > 0 \) positive definite.

**A4:** The function
\[
\begin{bmatrix} y \\ \dot{y} \\ x_c \\ L \end{bmatrix}
\]
is full rank and injective in an open neighborhood of the nominal solution \((\bar{q}(t), \dot{q}(t)) \forall t \).

From [17], [27], [28], the above assumptions imply that \((y, \dot{y}, x_c, L)\) is a valid set of coordinates for the full-order swing phase model (2). In particular,

1) there exists an invertible differentiable function \( \Phi \) such that
\[
\begin{bmatrix} q \\ \dot{q} \\ \tau \end{bmatrix} = \Phi(y, \dot{y}, x_c, L, \tau), \quad \text{and}
\]

2) the swing phase zero dynamics, that is, the dynamics of the robot compatible with \( y = 0 \), exists and can be parameterized by \((x_c, L)\).

### C. Zero Dynamics and Approximate Zero Dynamics

From Assumptions A1-A4, it follows that the swing phase zero dynamics exists and for \( \xi = (x_c, L, \tau) \) can be expressed as
\[
\dot{\xi} = f_{\text{zero}}(\xi),
\]
when \( y = 0 \). As with the popular pendulum models, the dimension of (47) is low, it has two states plus time. Different that the pendulum models, (47) is exact. Moreover, tools are known for relating periodic orbits of the hybrid version of (47) to corresponding orbits in the full-order model (6), including their stability properties; see Sect. VI-D.

On the basis of (10) evaluated at (46), the zero dynamics (47) can be written more explicitly as
\[
\dot{x}_c = \frac{L}{m z_c(x_c, L, \tau)} + \frac{\dot{z}_c(x_c, L, \tau)}{z_c(x_c, L, \tau)} x_c - \frac{L_c(x_c, L, \tau)}{m z_c(x_c, L, \tau)}
\]
\[
\dot{\tau} = 1.
\]
The state \( \dot{\tau} = 1 \) is included in (48) because, at hybrid transitions, \( \tau \) is reset to zero, that is, \( \tau^+ := 0 \). As discussed above, this reduced-order model is exact along all trajectories of the full-order model for which \( y = 0 \).

If one of the virtual constraints in (39) is \( z_c \equiv -H \), that is, the center of mass height is regulated to a constant, then the zero dynamics (exactly) simplifies to
\[
\dot{x}_c = \frac{L}{m H} - \frac{L_c(x_c, L, \tau)}{m H}
\]
\[
\dot{\tau} = 1.
\]

This model is nonlinear and time-varying through \( L_c \) and possibly, the feedback control policy chosen for the stance ankle torque, \( u_a \). We’ve argued in Sec. IV that \( L_c \) can be dropped from the model. Doing so results in the ALIP model, (20). Hence, the ALIP model is an approximate swing phase zero dynamics when the center of mass height is controlled to a constant.

### D. Consequences for Closed-loop Stability of the Full-order Model

When the foot placement policy (32) is applied to (49) with the rest map (25), the resulting closed-loop system is a (small) perturbation of a hybrid system that possesses a family of
exponentially stable periodic orbits parameterized by $L^{\text{des}}$. If the virtual constraints in (39) are hybrid invariant for constant $L^{\text{des}}$ [17], [73], [74], then

- (49) with impact map (25) is the hybrid zero dynamics, and
- an exponentially stable periodic solution of the hybrid zero dynamics is also an exponentially stable solution of the full order closed-loop system for appropriate choices of the feedback gains $K_p$ and $K_d$ in (44).

Consequently, the closed-loop system would possess a family of exponentially stable periodic orbits parameterized by $L^{\text{des}}$. If the virtual constraints are not hybrid invariant, then (49) with (25) does not form a hybrid zero dynamics in the sense of [17], but rather a limit restriction dynamics [75, pp. 102]. Moreover, via the Brouwer Fixed Point Theorem, reference [75, Theorem 6, pp. 105] shows each exponentially stable periodic solution of the limit restriction dynamics corresponds to an exponentially stable periodic solution of the full model for appropriate design of the feedback gains in (44).

To illustrate the correspondence between exponentially stable motions of the ALIP and the full-order model, we turn to the Rabbit model controlled via virtual constraints that implement the foot placement control law (32), the center of mass at a constant height, the torso upright, and adequate foot clearance. We then numerically estimate the Jacobian of the Poincaré map for the closed-loop full-order model and compare its dominant eigenvalues to the dominant eigenvalue of the closed-loop ALIP model; see (35).

In Table II, for various values of $\alpha$ in the step placement feedback controller, we show the dominant eigenvalue from the ALIP model and the dominant eigenvalue from the numerically estimated Poincaré map. We see that the dominant eigenvalue of the full-order closed-loop system corresponds to the dominant eigenvalue of the ALIP model for $0 \leq \alpha \leq 0.9$. The remaining eigenvalues of the full model are (very) small due to the gains chosen in (44). In fact, the zero dynamics captures the “weakly actuated”, slow part of the full-order model that is evolving under the influence of gravity.

### Remark:
We numerically obtained the Jacobian of the Poincaré map for Rabbit with the foot placement controller by the method symmetric differences; $\delta$ deviations were applied on ten states (Rabbit has 5 degree of freedom) and we measured the corresponding responses after two steps. The $\delta$ was chosen from the set $\{\pm 0.05, \pm 0.1, \pm 0.2, \pm 0.3\}$; see Table III.

| $\delta$ | $\alpha = \pm 0.05$ | $\delta = \pm 0.1$ | $\delta = \pm 0.2$ | $\delta = \pm 0.3$ |
|----------|----------------|----------------|----------------|----------------|
| 0.9      | 0.981          | 0.780          | 0.783          | 0.781          |
| 0.8      | 0.602          | 0.602          | 0.601          | 0.599          |
| 0.7      | 0.442          | 0.444          | 0.441          | 0.439          |
| 0.6      | 0.299          | 0.301          | 0.300          | 0.299          |
| 0.5      | 0.170          | 0.167          | 0.168          | 0.164          |
| 0.4      | 0.054          | 0.053          | 0.052          | 0.051          |
| 0.3      | 0.014          | 0.014          | 0.012          | 0.011          |

### Table III: Numerical support for estimating the Jacobian of the two-step Poincaré map. The dominant eigenvalue of Rabbit model is insensitive to the perturbation used in estimating the Jacobian.

#### E. Non-periodic Walking

The desired angular momentum, $L^{\text{des}}$, determines the fixed point of the Poincaré map and hence the walking speed of the robot. While varying $L^{\text{des}}$ causes the walking speed to change, the analysis of the controller has only been presented for a constant value of $L^{\text{des}}$. Reference [76] analyzes gait transitions in the formalism of the hybrid zero dynamics when $L^{\text{des}}$ is switched “infrequently”, meaning the closed-loop system is moving from a neighborhood of one periodic orbit to another. References [77], [78] generalize tools from Input-to-State Stability (ISS) of ODEs to the case of hybrid models. These results apply to time-varying $L^{\text{des}}$. The experimental work reported in Sec. IX includes examples of rapidly varying $L^{\text{des}}$, turn direction, and ground height.

#### F. Varying Center of Mass Height

We have seen that the difference between the zero dynamics of a real robot and a pendulum model is the term related to $L_c$. In previous sections, we have shown that the $L_c$ term has very little effect on the $L$ dynamics when $z_c$ is constant. This observation can be extended to the case when $z_c$ is not constant but virtually constrained by $(x_c, L, \tau)$.

In Fig. 12, we illustrate that when the $z_c$ is a function of time, the pendulum dynamics can still be used to predict accurately the zero dynamics of Rabbit.

### VII. Integrating Virtual Constraints and Angular-Momentum-based Foot Placement

In this section we generate virtual constraints for a 3D robot such as Cassie. As in [79], we leave the stance toe passive.

---

4Hybrid invariance means that if $y$ and $\dot{y}$ are zero before the impact, they will also be zero after the impact. References [71], [72] show how to systematically modify a given set of virtual constraints to achieve hybrid invariance.
Consequently, there are nine (9) control variables, listed below from the top of the robot to the end of the swing leg,

\[
\begin{bmatrix}
\text{torso pitch} \\
\text{torso roll} \\
\text{stance hip yaw} \\
\text{swing hip yaw} \\
\text{\( p^{\text{st} \rightarrow \text{CoM}} \)} \\
\text{\( p^{\text{sw} \rightarrow \text{CoM}} \)} \\
\text{\( p^{\text{sw} \rightarrow \text{CoM}} \)} \\
\text{\( \text{swing toe absolute pitch} \)} \\
\end{bmatrix}
\]

For later use, we denote the value of \( h_0 \) at the beginning of the current step by \( h_0(T_{k-1}^+) \). When referring to individual components, we’ll use \( h_{03}(T_{k-1}^+) \), for example.

We first discuss variables that are constant. The reference values for torso pitch, torso roll, and swing toe absolute pitch are constant and zero, while the reference for \( p^{\text{st} \rightarrow \text{CoM}} \) which sets the height of the CoM with respect to the ground, is constant and equal to \( H \).

We next introduce a phase variable

\[
s := \frac{t - T_{k-1}^+}{T}
\]

that will be used to define quantities that vary throughout the step to create “leg pumping” and “leg swinging”. The reference trajectories of \( p^{\text{sw} \rightarrow \text{CoM}} \) and \( p^{\text{sw} \rightarrow \text{CoM}} \) are defined such that:

- at the beginning of a step, their reference value is their actual position;
- the reference value at the end of the step implements the foot placement strategy in (31); and
- in between a half-period cosine curve is used to connect them, which is similar to the trajectory of an ordinary (non-inverted) pendulum.

The reference trajectory of \( p^{\text{sw} \rightarrow \text{CoM}} \) assumes the ground is flat and the control is perfect:

- at mid stance, the height of the foot above the ground is given by \( z_{\text{CL}} \), for the desired vertical clearance.

When implemented with an Input-Output Linearizing Controller\(^5\) so that \( h_0 \) tracks \( h_d \), the above control policy allows Cassie to move in 3D in simulation.

VIII. PRACTICAL IMPLEMENTATION ON CASSIE

This section resolves several issues that prevent the basic controller from being implemented on Cassie Blue.

A. IMU and EKF

In a real robot, an IMU and an EKF are needed to estimate the linear position and rotation matrix at a fixed point on the robot, along with their derivatives. Cassie uses a VectorNav IMU. We used the Contact-aided Invariant EKF developed in [81], [82] to estimate the torso velocity. With these signals in hand, we could estimate angular momentum about the contact point.

B. Filter for Angular Momentum

Angular Momentum about the contact toe could be computed directly from estimated \([q, \dot{q}]\), but it is noisy. We used a Kalman Filter to improve the estimation. The models we used are

\[\text{The required kinematic and dynamics functions are generated with FROST [80].}\]

\[\text{Fig. 12: Trajectory of } L \text{ and its prediction in a simulation of Rabbit. The instantaneous values are shown in blue and the predicted value at the end of the step is shown in red. In the prediction of } L, \text{ the virtual constraint on center of mass height for the model model and for Rabbit are set to } z_c = 0.6 + 0.05 \sin(\frac{T}{2\pi} - \frac{\pi}{2}) + 0.05, \text{ where } T \text{ is the step time. Large } \Delta z_c \text{ oscillations often occur in running. Here, we modify the ground model to pin the stance foot to the ground, so that we can impose a non-trivial } z_c \text{ oscillation in periodic walking.}\]

\[\text{Fig. 13: Block diagram of the implemented controller.}\]

\[\text{The reference trajectories for the stance hip and swing hip yaw angles are simple straight lines connecting their initial actual position and their desired final positions. For walking in a straight line, the desired final position is zero. To include turning, the final value has to be adjusted. Suppose that a turn angle of } \Delta D^\text{des}_k \text{ radians is desired. One half of this value is given to each yaw joint:}\]

\[\begin{align*}
&+\frac{1}{2} \Delta D^\text{des}_k \rightarrow \text{swing hip yaw; and} \\
&-\frac{1}{2} \Delta D^\text{des}_k \rightarrow \text{stance hip yaw}
\end{align*}\]

\[\text{The signs may vary with the convention used on other robots. The final result for Cassie Blue is}\]

\[
h_d(s) :=
\begin{bmatrix}
0 \\
0 \\
(1 - s)h_{03}(T_{k-1}^+) + s(-\frac{3}{2}(\Delta D_k)) \\
(1 - s)h_{04}(T_{k-1}^+) + s(\frac{1}{2}(\Delta D_k)) \\
H \\
\left[(1 + \cos(\pi s))h_{06}(T_{k-1}^+) + (1 - \cos(\pi s))p^{\text{sw} \rightarrow \text{CoM}}(T_k^-) \right] \\
\left[(1 + \cos(\pi s))h_{07}(T_{k-1}^+) + (1 - \cos(\pi s))p^{\text{sw} \rightarrow \text{CoM}}(T_k^-) \right] \\
4z_{\text{CL}}(s - 0.5)^2 + (H - z_{\text{CL}}), \\
0
\end{bmatrix}
\]

\[\text{When implemented with an Input-Output Linearizing Controller}\(^5\) \text{ so that } h_0 \text{ tracks } h_d, \text{ the above control policy allows Cassie to move in 3D in simulation.}\]
Prediction: \( L^y(k) = A L^y(k - 1) + B u(k) + \delta \)  

Correction: \( L^y_{\text{obs}}(k) = CL^y(k) + \epsilon \)  

where \( A = B = C = 1 \), \( u(k) = (mqx_w(k) + u_a(k)) \Delta T \). The update formula for angular momentum is

\[
L^y(k) = (I - K(k)C)(AL^y(k - 1) + B u(k)) + K(k)L^y_{\text{obs}}(k)
\]

The Kalman Gain \( K(k) \) is obtained following the algorithm described in [83, Sec 3.2].

C. Inverse Kinematics

Input-Output Linearization does not work well in experiments [79], [84], [85]. To use a passivity-based controller for tracking that is inspired by [70], we need to convert the reference trajectories for the variables in (50) to reference trajectories for Cassie’s actuated joints,

\[
q^{\text{act}} = \begin{bmatrix}
\text{torso pitch} \\
\text{torso roll} \\
\text{stance hip yaw} \\
\text{swing hip yaw} \\
\text{stance knee pitch} \\
\text{swing hip roll} \\
\text{swing hip pitch} \\
\text{swing knee pitch} \\
\text{swing toe pitch}
\end{bmatrix}
\]

Iterative inverse kinematics is used to convert the controlled variables in (50) to the actuated joints.

D. Passivity-based Controller

We adapt the passivity-based controller developed in [70] to achieve joint-level tracking. This method takes the full-order model of robot into consideration and, on a perfect model, will drive the virtual constraints asymptotically to zero. The derivation is given in the Appendix.

E. Springs

On the swing leg, the spring deflection is small and thus we are able to assume the leg to be rigid. On the stance leg, the spring deflection is non-negligible and hence requires compensation. While there are encoders on both sides of the spring to measure its deflection, direct use of this leads to oscillations. The deflection of the spring is instead estimated through a simplified model.

F. COM Velocity in the Vertical Direction

When Cassie’s walking speed exceeds one meter per second, the assumption that \( \dot{v}_{\text{CoM}} \approx 0 \) breaks down due to spring and imperfect low level control, and (29) is no longer valid. Hence, we use

\[
L^y(T^k_k) = L^y(T^k_k) + mv^z(T^k_k)(p^z_{\text{sw} \rightarrow \text{CoM}}(T^k_k) - p^z_{\text{st} \rightarrow \text{CoM}}(T^k_k)).
\]

From this, the foot placement is updated to

\[
p^z_{\text{sw} \rightarrow \text{CoM}}(T^k_k) = m(H \ell \sinh(ET) - v^z_{\text{CoM}}) \cosh(ET) - \frac{(L^y(T^k_k) + mv^z_{\text{CoM}}(T^k_k)p^z_{\text{st} \rightarrow \text{CoM}}(T^k_k)) \cosh(ET)}{m(H \ell \sinh(ET) - v^z_{\text{CoM}}) \cosh(ET)}.
\]

Remark: In our experiments \( v^z_{\text{CoM}} \) becomes negative at the end of a step when the robot is walking fast. If we still use (31) to decide foot placement, which is based on the reset map (29), in the lateral direction \( L^x(T^k_k) \) will be overestimated. This in turn leads to the lateral foot placement being commanded further from the body than it should be. At the end of the next step, the magnitude of \( L^x \) will be larger than expected, requires even further lateral foot placement from the body. The final phenomenon is abnormally large step width.

IX. EXPERIMENTAL RESULTS

The controller was implemented on Cassie Blue. The closed-loop system consisting of robot and controller was evaluated in a number of situations that are itemized below.

- Walking in a straight line on flat ground. Cassie could walk in place and walk stably for speeds ranging from zero to 2.1 m/s.
- Diagonal Walking. Cassie is able to walk simultaneously forward and sideways on grass, at roughly 1 m/s in each direction.
- Sharp turn. While walking at roughly 1 m/s, Cassie Blue effected a 90° turn in six steps, without slowing down.
- Rejecting the classical kick to the base of the hips. Cassie was able to remain upright under “moderate” kicks in the longitudinal direction. The disturbance rejection in the lateral direction is not as robust as the longitudinal, which is mainly caused by Cassie’s physical design: small hip roll motor position limits.
- Finally we address walking on rough ground. Cassie Blue was tested on the iconic Wave Field of the University of Michigan North Campus. The foot clearance was increased from 10 cm to 20 cm to handle the highly undulating terrain. Cassie is able to walk through the “valley” between the large humps with ease at a walking pace of roughly 0.75 m/s, without falling in all tests. The row of ridges running east to west in the Wave Field are roughly 60 cm high, with a sinusoidal structure. We estimate the maximum slope to be 40 degrees. Cassie is able to cross several of the large humps in a row, but also fell multiple times. On a more gentle, straight grassy slope of roughly 22 degrees near the laboratory, Cassie can walk up it with no difficulty with 20cm foot clearance.

The experimental data is analyzed in Fig. 14 and 15. The figures support the advantages of using \( L \) to indicate robot status and the accuracy of ALIP model, as discussed in Sec III and IV.

X. CONCLUSIONS

We established connections between various approximate pendulum models that are commonly used for heuristic controller design and those that are more common in the feedback
control literature where formal stability guarantees are the norm. The paper clarified commonalities and differences in the two perspectives for using low-dimensional models. In the process of doing so, we argued that models based on angular momentum about the contact point provide more accurate representations of robot state than models based on linear velocity. Specifically, we showed that an approximate (pendulum or zero dynamics) model parameterized by angular momentum provides better predictions for foot placement on
a physical robot (e.g., legs with mass) than does a related approximate model parameterized in terms of linear velocity. While ankle torque, if available, can achieve small changes in angular momentum during a step, the limitations of foot roll prevent ankle torque from achieving large changes in angular momentum during a step. For this reason, we focused our analysis on regulating angular momentum about the contact point step-to-step and not within a step. We implemented a one-step-ahead angular-momentum-based controller on Cassie, a 3D robot, and demonstrated high agility and robustness in experiments. Using our new controller, Cassie was able to accomplish a wide range of tasks with nothing more than common sense task-based tuning: a higher step frequency to walk at 2.1 m/s and extra foot clearance to walk over slopes exceeding 22 degrees. Moreover, in the current implementation, there is no optimization of trajectories used in the implementation on Cassie. The robot’s performance is currently limited by the hand-designed trajectories leading to joint-limit violations and foot slippage. These limitations will be mitigated by incorporating optimization and or constraints via MPC.

In addition to the foot placement scheme described in this paper, the ALIP model can be combined with many other control methods that have been implemented on the LIP model: it can be used with ZMP in an MPC scheme as in [11]–[13], or used with Capture Point as in [4], [5], or used with an estimated $L_c$ [15], [16]. As has already been done in [37], $L$ can be regulated by controlling the center of mass velocity just before impact, as well.

APPENDIX

A. Constant Pendulum Length

Suppose that one component of the virtual constraints in (39) is $r_c(q) = R$, where $R$ is a constant. Then $y \equiv 0$ yields $r_c = R$, simplifying (12) to

$$\dot{\theta}_c = \frac{L - L_c}{mR^2} \dot{\theta}_c + u_a.$$  \hspace{1cm} (58)

At this point, no approximations have been made and the models is valid everywhere that $r_c(q) \equiv R$. An interesting aspect of this pendulum model is that it does not depend on $R$, and thus imperfections in a achieving the virtual constraint $r_c = R$ have a smaller effect here than in (17), where $\dot{z}_c$ would appear when $z_c \neq H$, or in (16), where both $\dot{z}_c$ and $\ddot{z}_c$ would appear.

As with (17), the model (58) is driven by the strongly actuated states $q_b, \dot{q}_b$ through $L_c$ and the same discussion applies. Dropping $L_c$ in (10) results in

$$\dot{\theta}_c = \frac{L}{mR^2} \dot{\theta}_c + u_a,$$  \hspace{1cm} (59)

which is nonlinear in $\theta_c$. However, for $R = 1$ and a step length of 60 cm, $\max \theta_c \approx \pi/6$, and for 70 cm, $\max \theta_c \approx \pi/4$, giving simple bounds on the approximation error,

$$\frac{1}{\pi/6} \int_0^{\pi/6} (\theta - \sin(\theta))d\theta < 0.006$$

$$\frac{1}{\pi/4} \int_0^{\pi/4} (\theta - \sin(\theta))d\theta < 0.02.$$  

Moreover, if desired, one can chose $K$ to set

$$\frac{1}{\theta_{\max}} \int_0^{\theta_{\max}} (K\theta - \sin(\theta))d\theta = 0.$$  

For $\theta_{\max} = \pi/4$, the value is $K \approx 0.95$. While a linear approximation is useful for having a closed-form solution, numerically integrating the nonlinear model (59) in real time is certainly feasible.

The discussion on the approximate zero dynamics can be repeated here. The associated impact map is nonlinear and can be linearized about a nominal solution.

B. Passivity-based Input-Output Stabilization

The material presented here adapts the original work in [70] to a floating-base model

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + Js^T \tau_s + J_g(q)^T \tau_g,$$  \hspace{1cm} (60)

with $u$ the vector of motor torques, $\tau_s$ the spring torques, and $\tau_g$ the contact wrench. During the single support phase, the blade-shape foot on Cassie provides five holonomic constraints, leaving only foot roll free. To simplify the problem, we also assume the springs are rigid, adding two constraint on each leg. These constraints leave the original 20-degree-of-freedom floating base model with 11 degrees of freedom.

The constraints mentioned above can be written as

$$\begin{cases}
J_s\ddot{q} = 0 \\
J_g(q)\ddot{q} + J_g(q)\dot{q} = 0.
\end{cases}$$  \hspace{1cm} (61)

Combining (60) and (61) yields the full model for Cassie in single support,

$$\begin{bmatrix}
D & -J_g^T \\
J_s & 0
\end{bmatrix}
\begin{bmatrix}
\ddot{q} \\
\tau_s
\end{bmatrix} +
\begin{bmatrix}
H \\
0
\end{bmatrix}
\begin{bmatrix}
\dot{q} \\
\dot{\tau}_g
\end{bmatrix} =
\begin{bmatrix}
B \\
0
\end{bmatrix}
u.$$  \hspace{1cm} (62)

For simplicity we assume that the components of $q$ have already been ordered such that $q = [q_c, q_u]^T$, where $q_c$ are the coordinates chosen to be controlled and $q_u$ are the free coordinates. Define $\lambda = [q_u, \tau_s, \tau_g]^T$ and partition (62) as

$$\begin{bmatrix}
D_{11}\ddot{q}_c + D_{12}\lambda + H_1 = \tilde{B}_1u \\
D_{21}\ddot{q}_c + D_{22}\lambda + H_2 = \tilde{B}_2u
\end{bmatrix}$$  \hspace{1cm} (63)

The vector $\lambda$ can be eliminated from these equations, resulting in

$$D\ddot{q}_c + H = Bu,$$  \hspace{1cm} (64)

where

$$\begin{bmatrix}
\tilde{D} \\
\tilde{H} \\
\tilde{B}
\end{bmatrix} =
\begin{bmatrix}
D_{11} - D_{12}\tilde{D}_{22}^{-1}\tilde{D}_{21} \\
H_1 - \tilde{D}_{12}\tilde{D}_{22}^{-1}H_2 \\
\tilde{B}_1 - \tilde{D}_{12}\tilde{D}_{22}^{-1}\tilde{B}_2
\end{bmatrix}.$$
We note that $\bar{D}_{22}$ being invertible and $\bar{D}$ being positive definite both follow from $\bar{D}$ being positive definite. Later, we need $\bar{B}$ to be invertible; this is an assumption similar to that in (42). Equation (64) is what we will focus on from here on.

For the Passivity-Based controller, the error dynamic for $y := q_e - q_r$ and is designed to be [88]

$$D\dot{y} + (C + k_d)\dot{y} + k_p y = 0,$$

(65)

where $C$ is the Coriolis/centrifugal matrix in $\bar{H}$ and it is chosen such that $\bar{D} = C + GT$. From (64) and (65), we have

$$u = \bar{B}^{-1}(\bar{D}\dot{q}_r + \bar{H}) - \bar{B}^{-1}(k_p y + (C + k_d)\dot{y}).$$

(66)

Compared with a standard Input-Output Linearization controller, whose error dynamics and command torque are

$$\dot{y} + k_d \dot{y} + k_p y = 0,$$

(67)

the passivity-based controller induces less cancellation of the robot’s dynamics, and if $k_p$ and $k_d$ are chosen to be diagonal matrices, the tracking errors are approximately decoupled because, for Cassie, $\bar{B}^{-1}$ is close to diagonal. This controller provides improved tracking performance over the straight-up PD implementation in [79].

Acknowledgment

Toyota Research Institute provided funds to support this work. Funding for J. Grizzle was in part provided by NSF Award No. 1808051. The authors thank Omar Harib and Jiunn-Kai Huang for their assistance in the experiments.

References

[1] H. Miura and I. Shimoyama. Dynamic walk of a biped. International Journal of Robotics Research, 3(2):60–74, 1984.

[2] Shunji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180), volume 1, pages 239–246, IEEE, 2001.

[3] R. Blickhan. The spring–mass model for running and hopping. Journal of Biomechanics, 22(11-12):1217–1227, 1989.

[4] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In 2006 6th IEEE-RAS international conference on humanoid robots, pages 207–207. IEEE, 2006.

[5] Johannes Englsberger, Christian Ott, Máximo A. Roa, Alin Albu-Schäffer, and Gerhard Hirzinger. Biped walking control based on capture point dynamics. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4420–4427. IEEE, 2011.

[6] Ting Wang and Christine Chevallereau. Stability analysis and time-varying walking control for an under-actuated planar biped robot. Robotics and Autonomous Systems, 59(6):444 – 456, 2011.

[7] Xiaobin Xiong and Aaron D. Ames. Orbit characterization, stabilization and composition on 3d underactuated bipedal walking via hybrid passive linear inverted pendulum model. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4644–4651, IEEE, 2019.

[8] Agility Robotics. Robots. https://www.agilityrobotics.com/robots/cassie, 2021.

[9] A. Goswami and V. Kullem. Rate of change of angular momentum and balance maintenance of biped robots. In Proc. of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, volume 4, pages 3783–3790, 2004.

[10] M. Popovic, A. Höfmann, and H. Her. Zero spin angular momentum control: definition and applicability. In 4th IEEE-RAS International Conference on Humanoid Robots, 2004., volume 1, pages 478–493 Vol. 1, 2004.
[31] Ayonga Hereid, Shishir Kolathaya, Mikhail S. Jones, Johnathan Van Why, Jonathan W. Hurst, and Aaron 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, HSCC '14, pages 263–272, New York, NY, USA, 2014. ACM.

[32] Yu-Ming Chen and Michael Posa. Optimal reduced-order modeling of bipedal locomotion. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 8753–8760. IEEE, 2020.

[33] A. Sano and J. Furusho. Realization of natural dynamic walking using the angular momentum information. In Proc. of the 1990 IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 1990, pages 1476–1480.

[34] E. Westervelt, W. J. Grizzle, and D.E. Koditschek. Hybrid zero dynamics of planar biped walkers. IEEE Transactions on Automatic Control, 48(1):42–56, January 2003.

[35] J. W. Grizzle, Claude H. Moog, and Christine Chevallereau. Nonlinear control of mechanical systems with an unactuated cyclic variable. IEEE Transactions on Automatic Control, 50(5):559–576, 2005.

[36] Morteza Azad and Roy Featherstone. Angular momentum based controller for balancing an inverted double pendulum. In Romanya 19–Robot Design, Dynamics and Control, pages 251–258. Springer, 2013.

[37] Matthew J Powell and Aaron D Ames. Mechanics-based control of underactuated 3d robotic walking: Dynamic gait generation under torque constraints. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 555–560. IEEE, 2016.

[38] Yûkai Gong and Jessy Grizzle. One-step ahead prediction of angular momentum about the contact point for control of bipedal locomotion: Validation in a lip-inspired controller. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 2832–2838. IEEE, 2021.

[39] Brent Griffin and Jessy Grizzle. Nonholonomic virtual constraints and gait optimization for robust walking control. The International Journal of Robotics Research, page 0278364917708249, 2016.

[40] Shuuji Kajita, Fumio Kanohe, Kenji Kaneko, Kiyoshi Fujiwara, Ken-suke Harada, Kazuhito Yokoi, and Hiroshi Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In ICRA, volume 3, pages 1620–1626, 2003.

[41] Xiaobin Xiong, Jenna Reher, and Aaron D Ames. Global position control on underactuated bipedal robots: Step-to-step dynamics approximation for step planning. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 2825–2831. IEEE, 2021.

[42] Kaveh Akbari Hamed, Naser Sadati, William A Gruver, and Guy A Dumont. Stabilization of periodic orbits for planar walking with instantaneous double-support phase. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 42(3):685–706, 2011.

[43] Woong Ma, Kaveh Akbari Hamed, and Aaron D Ames. First steps towards full model based motion planning and control of quadrupeds: A hybrid zero dynamics approach. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5498–5503. IEEE, 2019.

[44] Jacob P Reher, Ayonga Hereid, Shishir Kolathaya, Christian M Hubicki, and Aaron D Ames. Algorithmic foundations of realizing multi-contact locomotion on the human robot durus. In Proc. 12th Adv. Robot. Algorithmic Found. Robot, 13:400–415, 2020.

[45] Takahide Kamioka, Hirokiyo Kaneko, Toru Takenaka, and Takahide Yoshiike. Simultaneous optimization of zmp and footsteps based on the analytical solution of divergent component of motion. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1763–1770, 2018.

[46] J. H. Choi and J. W. Grizzle. Planar bipedal walking with foot rotation. In Proceedings of the American Control Conference, pages 4909–4916, Portland, OR, 2005.

[47] Victor Paredes and Ayonga Hereid. Dynamic locomotion of a lower-limb exoskeleton through virtual constraints based zmp regulation. In Dynamic Systems and Control Conference, volume 84270, page V011T14A001. American Society of Mechanical Engineers, 2020.

[48] C. Chevallereau, G. Abba, Y. Asoustin, F. Plestan, E. R. Westervelt, C. Canudas-de-Wit, and J. W. Grizzle. RABBIT: A testbed for advanced control theory. IEEE Control Systems Magazine, 23(5):57–79, October 2003.

[49] Y. Hurmuzlu and T.H. Chang. Rigid body collisions of a special class of planar kinematic chains. IEEE Transactions on Systems, Man, and Cybernetics, 22(5):964–971, 1992.

[50] Y. Hurmuzlu. Dynamics of bipedal gait - part 1: objective functions and the contact event of a planar five-link biped. Journal of Applied Mechanics, 60:331–336, June 1993.

[51] 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(1):64, January 2001.

[52] Shuju Kajita and Kumi Tan. Study of dynamic biped locomotion on rugged terrain-theory and basic experiment. In Advanced Robotics, 1991.’Robots in Unstructured Environments’, 91 ICAR, Fifth International Conference on, pages 741–746. IEEE, 1991.

[53] Omar Harib, Ayonga Hereid, Ayush Agrawal, Thomas Gurtner, Sylvain Finet, Guilhem Boeris, Alexits Duburcq, M. Eva Mungai, Matthieu Maselle, Aaron D. Ames, Koushil Sreenath, and Jessy Grizzle. Feedback control of an exoskeleton for paraplegics: Toward robustly stable hands-free dynamic walking. arXiv preprint arXiv:1802.08322 [cs.RO], 2018.

[54] H-M Maas, S W Lipfert, M Gross, J Rumel, and A Seyfarth. Upright human gait did not provide a major mechanical challenge for our ancestors. Nature Communications, 1(6):70, January 2010.

[55] P-B Wieber. Helonomy and nonholonomy in the dynamics of articulated motion. In Fast motions in biomechanics and robotics, pages 411–425. Springer, 2006.

[56] Jerry E Pratt and Russ Tedrake. Velocity-based stability margins for fast bipedal walking. In Fast Motions in Biomechanics and Robotics, pages 299–324. Springer, 2006.

[57] Hongkai Dai and Russ Tedrake. Planning robust walking motion on uneven terrain via convex optimization. In Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on, pages 579–586. IEEE, 2016.

[58] Andreas Hofmann, Marko Popovic, and Hugh Herr. Exploiting angular momentum to enhance bipedal center-of-mass mass. In 2009 IEEE International Conference on Robotics and Automation, pages 4423–4429. IEEE, 2009.

[59] Ambarish Goswami and Vinutha Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04, volume 4, pages 3785–3790. IEEE, 2004.

[60] Xiaobin Xiong and Aaron D Ames. Sequential motion planning for bipedal somersault via flywheel slip and momentum transmission with task space control. arXiv preprint arXiv:2008.02432, 2020.

[61] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on, pages 1405–1411 vol.2, 1991.

[62] M. H. Raibert. Hopping in legged systems—modeling and simulation for the two-dimensional one-legged case. IEEE Transactions on Systems, Man and Cybernetics, 14(3):451–63, June 1984.

[63] J. Pratt, J. Carriñ, S. Drakanov, and A. Goswami. Capture Point: A Step toward Humanoid Push Recovery. 2006 6th IEEE-RAS International Conference on Humanoids, 2006.

[64] Xingye Da, Omar Harib, Ross Hartley, Brent Griffin, and Jessy W Grizzle. From 2D design of underactuated bipedal gaits to 3D implementation: Walking with speed tracking. IEEE Access, 4:3469–3478, 2016.

[65] Majid Khadiv, Alexander Herzog, S Ali A Moosavian, and Ludovic Righetti. Walking control based on step timing adaptation. IEEE Transactions on Robotics, 36(3):629–643, 2020.

[66] Carlos Gonzalez, Victor Barasuol, Marco Frigerio, Roy Featherstone, Darwin G Caldwell, and Claudia Semini. Line walking and balancing for legged robots with point feet. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3649–3656. IEEE, 2020.

[67] Eric R Westervelt, Christine Chevallereau, Jun Ho Choi, Benjamin Morris, and Jessy W Grizzle. Feedback control of dynamic bipedal robot locomotion. CRC press, 2007.

[68] Matthew J Powell, Ayonga Hereid, and Aaron D Ames. Speed regulation in 3d robotic walking through motion transitions between human-inspired partial hybrid zero dynamics. In 2013 IEEE international Symposium on Robotics and automation, pages 4803–4810. IEEE, 2013.

[69] Jessy W Grizzle and Christine Chevallereau. Virtual constraints and hybrid zero dynamics for realizing underactuated bipedal locomotion. In Bruno Siciliano and Oussama Khatib, editors, Springer handbook of robotics. Springer, 2016.

[70] Hamid Sadeghian, Christian Ott, Gianluca Garofalo, and Gordon Cheng. Task-based control of a biped robot in a hybrid hybrid zero dynamics approach. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 4096–4101. IEEE, 2017.
[71] B. Morris and J. W. Grizzle. Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots. IEEE Transactions on Automatic Control, 54(8):1751–1764, August 2009.

[72] C. Chevallereau, J. W. Grizzle, and C. L. Shih. Asymptotically stable walking of a five-link underactuated 3D bipedal robot. IEEE Transactions on Robotics, 25(1):37–50, February 2009.

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

[74] William Yang and Michael Posa. Impact invariant control with applications to bipedal locomotion. arXiv preprint arXiv:2103.06907, 2021.

[75] Giuseppe Viola. Control of underactuated mechanical systems via passivity-based and geometric techniques. , 2008.

[76] Eric R Westervelt, Jessy W Grizzle, and C Canudas De Wit. Switching and pi control of walking motions of planar biped walkers. IEEE Transactions on automatic control, 48(2):308–312, 2003.

[77] Shishir Kolathaya and Aaron D Ames. Input-to-state safety with control barrier functions. IEEE control systems letters, 3(1):108–113, 2018.

[78] Sushant Veer, Ioannis Poulakakis, et al. Input-to-state stability of periodic orbits of systems with impulse effects via poincaré analysis. IEEE Transactions on Automatic Control, 64(11):4583–4598, 2019.

[79] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J. Huang, and J. Grizzle. Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway. In 2019 American Control Conference (ACC), pages 4559–4566, 2019.

[80] Hereid Ayonga. Fast Robot Optimization and Simulation Toolkit (FROST). https://ayonga.github.io/frost-dev/.

[81] Ross Hartley, Maani Ghaffari, Ryan M Eustice, and Jessy W Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. The International Journal of Robotics Research, 39(4):402–430, 2020.

[82] Ross Hartley. Invariant-EKF. https://github.com/RossHartley/invariant-ekf.

[83] Sebastian Thrun. Probabilistic robotics. Communications of the ACM, 45(3):52–57, 2002.

[84] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle. Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs. IEEE Access, 3:323–332, 2015.

[85] 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, 24(6):559–582, June 2004.

[86] Experiment Video Compilation. https://youtu.be/o8msaembEu8.

[87] Michigan Robotics: Dynamic Legged Locomotion Lab Youtube Channel. https://www.youtube.com/channel/UCMfDV8rkWhUwnTAYq0Q. Accessed: 2021-05-09.

[88] Christian Ott. Cartesian impedance control of redundant and flexible-joint robots. Springer, 2008.