

A Discrete Control Lyapunov Function for Exponential Orbital Stabilization of the Simplest Walker

Pranav A. Bhounsule and Ali Zamani

Robotics and Motion Laboratory,

Dept. of Mechanical Engineering, The University of Texas at San Antonio,

One UTSA Circle, San Antonio, TX 78249, USA.

Corresponding author: Email: pranav.bhounsule@utsa.edu

ABSTRACT

In this paper, we demonstrate the application of a discrete control Lyapunov function (DCLF) for exponential orbital stabilization of the simplest walking model supplemented with an actuator between the legs. The Lyapunov function is defined as the square of the difference between the actual and nominal velocity of the un-actuated stance leg at the mid-stance position (stance leg is normal to the ramp). The foot placement is controlled to ensure an exponential decay in the Lyapunov function. In essence, DCLF does foot placement control to regulate the mid-stance walking velocity between successive steps. The DCLF is able to enlarge the basin of attraction by an order of magnitude and to increase the average number of steps to failure by two orders of magnitude over passive dynamic walking. We compare DCLF with a one-step dead-beat controller (full correction of disturbance in a single step) and find that both controllers have similar robustness. The one-step dead-beat controller provides the fastest convergence to the limit cycle while using least amount of energy per unit step. However, the one-step dead-beat controller is more sensitive to modeling errors. We also compare the DCLF with an eigenvalue-based controller for the same rate of convergence. Both controllers yield identical robustness but the DCLF is more energy-efficient and requires lower maximum torque. Our results suggest that the DCLF controller with moderate rate of convergence provides good compromise between robustness, energy-efficiency, and sensitivity to modeling errors.

1 Introduction

Passive dynamic robots that walk downhill are highly energy-efficient because they rely only on their mass distribution and geometry [1–3]. These robots are gravity powered and do not use external control or power. Consequently, such robots

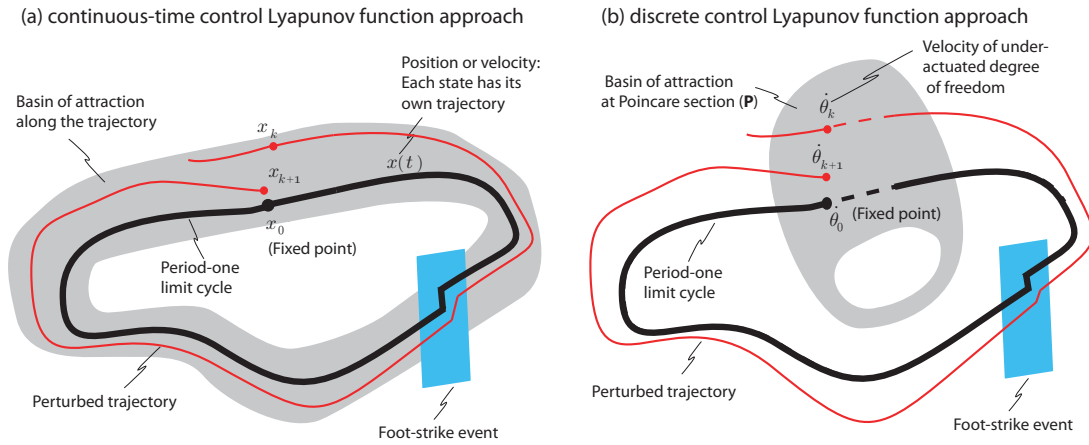


Fig. 1. Difference between continuous and discrete control Lyapunov function methodologies: (a) Continuous control Lyapunov function (CCLF) constructs a controller that keeps the trajectory within a tube *along the trajectory*. (b) Discrete control Lyapunov function (DCLF) constructs controller that keeps the trajectory in the basin of attraction only *at the Poincaré section*.

are highly energy-efficient. But no control means that these robots cannot correct their motion when disturbed, thus they fall down due to even the slightest disturbance [4, 5]. The lack of stability limits the practical use of such robots.

The compass gait model is a well studied model of walking [6]. The model consists of a point mass torso and two legs connected to each other by a pin joint. One leg pivots about the ground while the other leg, actuated by a hip actuator, swings about the grounded leg. The legs exchange their roles after the swinging leg contacts the ground. The model is challenging to control because of non-linearity and under-actuation (more degrees of freedom than actuators). Iida et al. [7] controlled the compass gait model by creating an open-loop hip actuator profile that effectively “phase locks” the swing leg to the gait cycle. Manchester et al. [8] used a closed-loop controller in which the swing leg is coupled to the motion of the stance leg using a high gain feedback controller. In this paper, we consider the simplest walking model, which is a special case of the compass gait model: the ratio of the leg to torso mass is zero. This feature decouples the swing leg from the stance leg. Consequently the hip actuator is not able to influence the motion of the stance leg in the swing phase of motion (that is, the system *cannot be stabilized*). Thus, a controller that does step-to-step or orbital stabilization (as opposed to local stabilization) is more appropriate for this model. Here, the hip actuator controls the step length in order to modulate the stance leg velocity between successive steps.

We develop a stabilizing controller using a discrete control Lyapunov function (DCLF). The Lyapunov function is chosen at the Poincaré section at mid-stance. The mid-stance is defined as the position when the stance leg (grounded leg) is normal to the ramp. Next, a control law is chosen that results in exponential stabilization of the system between steps at the Poincaré section. The key idea is to use the actuated degree of freedom (the swing leg) to control the un-actuated degree of freedom (the stance leg) between steps. Although the continuous control Lyapunov function (CCLF) [9] has been used for gait stabilization, the use of DCLF is novel in this area. The major difference between the two is that DCLF does orbital or step-to-step stabilization while CCLF does local stabilization (see Dingwell and Kang [10] for difference between orbital and local stabilization).

The organization of the paper is as follows. The background and related work is presented in Section 2. The details

about the simplest model including equations of motion are discussed in Section 3. The details about the DCLF technique are provided in Section 4. The results and the discussion are in Sections 5 and 6, respectively.

2 Background and Related Work

There are broadly two notations of bipedal robot stability: (1) stability is the ability to not fall down, and (2) stability is the ability to follow a given reference trajectory. Next, we discuss some metrics associated with these notions of stability.

Viability, mean first-passage time, and gait sensitivity norm are some metrics that quantify the robots ability to not fall down. Viability is the set of all states from which the robot can avoid falling down [11]. This is a computationally expensive metric to compute and almost intractable for high dimensional systems. The mean first-passage time is the number of steps the robot can take before falling down [12]. This definition is a probabilistic one and is evaluated by simulating the robot in a variety of different disturbances conditions and tends to be computationally expensive as the system dimension increases. The gait sensitivity norm (GSN) is the two norm of the ratio of a gait indicator (e.g., step time, velocity) to a disturbance (e.g., terrain height, push) [5]. This is easy to compute but is sensitive to the choice of a good gait indicator that correlates with falling. Moreover, as GSN is based on linearization, it works well only for small disturbances.

Basin of attraction, the largest eigenvalue of the limit cycle, and Lyapunov function are some metrics that quantify the robots ability to follow a given reference trajectory. The basin of attraction is the set of all initial states that will converge to the given reference trajectory [4, 13]. This measure is computationally expensive even for the simplest system. The magnitude of the largest eigenvalue of the Poincaré map indicates how fast a perturbation in the state would converge back to the reference trajectory [1]. The closer the value is to zero, the faster will be the convergence, while values greater than 1 will diverge and lead to instability. This is a computationally simple measure but is only valid for small perturbations. A Lyapunov function is a positive definite function whose time derivative along any state trajectory of system decreases with time. It is non-trivial to find a Lyapunov function but the recent use of sum of squares optimization provides a generalizable numerical technique [14]. The issue with all these metrics is that they are used to check stability after the controller has been designed.

The eigenvalues of the Poincaré map can be controlled using feedback. We call this controller as an eigenvalue-based controller. The key idea is to linearize the Poincaré map with respect to the state and suitable control action (e.g., step length, push-off). Then pole placement is used to set the eigenvalues of the linearized equation. For example, McGeer [15] used the instantaneous push-off as the control action to stabilize a 2D model while Kuo [16] used lateral foot placement to control lateral stability of a 3D model of walking. We describe the eigenvalue-based controller in Sec. 4.5 and also compare it with our controller.

The control Lyapunov function provides a generalizable method to design a stable control law [17]. For example, Ames et al. [9] used continuous-time control Lyapunov functions (CCLF) to guarantee exponential stabilization of the hybrid zero dynamics of the system. A conceptualization of their methodology is shown in Fig. 1 (a). The thick black line shows the reference trajectory and the thin red line shows a perturbed trajectory. While the CCLF is chosen to keep the trajectory within the gray tube, the discontinuous foot-strike event at the section shown by blue rectangle, tends to push the system out

of the gray tube. However, by choosing suitable tuning parameters, it is possible to keep the perturbed trajectory within the gray tube in spite of the discontinuous foot-strike event.

We take an alternate approach by using a discrete-time control Lyapunov function (DCLF). This is shown in Fig. 1 (b). Here we try to keep the perturbed trajectory of the un-actuated degree of freedom (e.g., stance leg of bipedal robot) within the gray region *only* at the Poincaré section. This is done by using the actuated degrees of freedom, that are fully controllable in a step, to affect the un-actuated degree of freedom over a complete step (e.g., using foot placement). In this view, DCLF does not think of the foot-strike event as a disturbance, but rather an essential quantity to modulate the un-actuated degree of freedom. Another difference is that DCLF provides exponential orbital stability while CCLF can only provide asymptotic orbital stability, although CCLF is able to provide exponential stabilization of the hybrid zero dynamics.

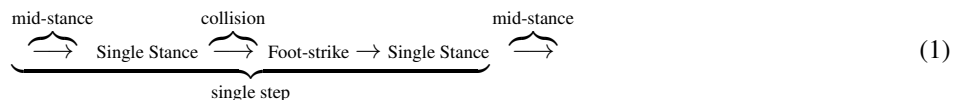
In this paper, we illustrate an application of DCLF using the simplest walking model [18], but with the addition of an actuator between the two legs. The system is nonlinear and under-actuated. Since the point mass torso is heavy compared to the legs, the swing leg motion cannot affect the stance leg motion in the swing phase, and the system is thus decoupled. However, the swing leg (actuated degree of freedom) motion can influence the stance leg (un-actuated degree of freedom) motion by appropriate foot-placement (e.g., big step reduces the stance leg velocity [19]), which is exploited in the DCLF approach.

3 Model

3.1 Model description

Fig. 2 shows a cartoon of the simplest walker [18]. The model has mass M at the hip and point mass m at each of the feet. Each leg has length ℓ . Gravity g points downwards. The leg in contact with the ramp is called the stance leg while the other leg is called the swing leg. The angle made by the stance leg with the normal to the ramp is θ and the angle made by the swing leg with the stance leg is ϕ . The hip torque is τ and the ramp slope is γ .

A single step of the walker is given below:



A single step consists of two phases; a single stance phase where the swing leg pivots about a stationary stance foot and foot-strike phase where there is support transfer and the legs exchange their roles. These phases are connected through two switching events; a mid-stance event where the stance leg is normal to the ramp and a collision phase where the leading leg touches the ground. Note that we have chosen a single step to start and end at mid-stance unlike the usual convention of using the instant just after foot-strike. The use of mid-stance as opposed to foot-strike for the Poincaré section will become clear in the section on Methods (see Sec. 4). Next, we describe the equations that describe the phases and events.

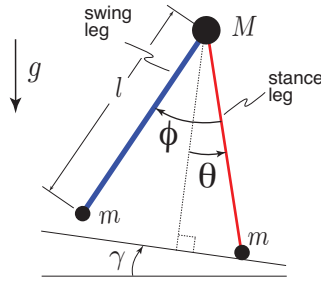


Fig. 2. Model: The simplest slope walking model analyzed by Garcia et al. [18].

3.1.1 Mid-stance event:

The mid-stance event is the logical condition that indicates the stance leg is normal to the ramp and is given by

$$\theta = 0. \quad (2)$$

3.1.2 Single stance phase (continuous dynamics):

In this phase of motion, the stance leg pivots and rotates about the stance foot contact point while the swing leg pivots and rotates about the hinge connecting the two legs. Our assumptions are: the stance foot does not slip (infinite friction between stance foot and the ground), there is no hip hinge friction, and foot scuffing is ignored. We obtain Eqs. (3) and (4) defined below by taking moments about the stance foot contact point and hip hinge respectively, and non-dimensionalizing time with $\sqrt{\ell/g}$ and applying the limit, $m/M \rightarrow 0$. In Eq. (4), τ is the non-dimensional torque obtained by dividing the dimensional torque by Mgl . The equations are:

$$\ddot{\theta} = \sin(\theta - \gamma), \quad (3)$$

$$\ddot{\phi} = \sin(\theta - \gamma) + \{\dot{\theta}^2 - \cos(\theta - \gamma)\} \sin(\phi) + \tau. \quad (4)$$

3.1.3 Collision event:

The collision event is the logical condition that indicates the leading leg is about to make contact with the ground. We introduce a step down of height h at the collision. This is taken to be zero, except for testing the robustness of the control approach. The collision event is given by

$$\cos(\phi - \theta) - \cos(\theta) = h. \quad (5)$$

3.1.4 Foot-strike phase (discontinuous dynamics):

In this phase of motion, the legs exchange their roles, that is, the current swing leg becomes the new stance leg and the current stance leg becomes the new swing leg. There is an instantaneous plastic collision (no slip and no bounce) of the swing leg. The swapping of legs is expressed by Eqs. (6) and (7). The angular rates of the legs after support exchange are given by Eqs. (8) and (9) and are obtained by applying conservation of angular momentum about stance foot contact point and hip hinge respectively, followed by non-dimensionalizing time with $\sqrt{\ell/g}$ and applying the limit, $m/M \rightarrow 0$. In the algebraic equations below, the state variables before and after collision are denoted using the superscript $-$ and $+$ respectively.

$$\theta^+ = \theta^- - \phi^-, \quad (6)$$

$$\phi^+ = -\phi^-, \quad (7)$$

$$\dot{\theta}^+ = \dot{\theta}^- \cos \phi^-, \quad (8)$$

$$\dot{\phi}^+ = \dot{\theta}^- \cos \phi^- (1 - \cos \phi^-). \quad (9)$$

3.2 Failure modes:

There are two failure modes for the simplest walker and they are described below. These lead to two conditions on the state of the system and are checked at each integration step. Violation of any of these conditions is interpreted as system failure.

1. *Falling Backwards:* Falling backwards is detected when the angular velocity of the stance leg is positive (note that forward velocity is indicated by a negative angular velocity). Thus the condition for failure is: $\dot{\theta} \geq 0$.
2. *Flight phase:* Flight phase is detected when the vertical ground reaction force, $R_y = \cos(\theta - \gamma) - \dot{\theta}^2 \leq 0$. Thus the condition for failure is:

$$\dot{\theta}^2 - \cos(\theta - \gamma) \geq 0 \quad (10)$$

4 Methods

4.1 Overview of control technique

From the single stance phase Eqns. (3) and (4), we see that: (1) the hip torque can be used to control the swing leg and (2) the motion of the swing leg does not affect the motion of the stance leg. Thus, by the controllability definition [20], although the swing leg motion, $\phi(t)$, is fully controllable, the motion of the stance leg, $\theta(t)$, is not controllable *within a step*. However, we can find a function, F , that maps the mid-stance between consecutive steps and is indexed by step number, k .

Thus

$$\dot{\theta}_{k+1} = F(\dot{\theta}_k, \phi^-) \quad (11)$$

where ϕ^- is swing leg angle at foot-strike and is related to the step length. Given the measurement at step k , $\dot{\theta}_k$, the hip torque can be used to modulate ϕ^- to control $\dot{\theta}_{k+1}$. Thus, the stance leg is fully controllable *between steps*. Based on these observations, we use a hierarchical control approach: *the stance leg velocity is controlled between steps using foot placement obtained from DCLF, while the swing leg is controlled using a trajectory tracking controller based on the foot placement angle*.

4.2 Mid-stance to mid-stance map for stance leg:

In this section we present equations that can be used to numerically solve for the mid-stance to mid-stance map, F , given by Eq. (11).

We do an energy balance for the stance leg between the mid-stance and the instant just before foot-strike and then again from just after foot-strike to the next mid-stance to get Eqns. (12) and (13) respectively

$$\frac{(\dot{\theta}_k)^2}{2} + \cos \gamma = \frac{(\dot{\theta}^-)^2}{2} + \cos \left(\frac{\phi^-}{2} - \gamma \right), \quad (12)$$

$$\begin{aligned} \frac{(\dot{\theta}_{k+1})^2}{2} + \cos \gamma &= \frac{(\dot{\theta}^+)^2}{2} + \cos \left(\frac{\phi^+}{2} - \gamma \right), \\ &= \frac{(\dot{\theta}^- \cos \phi^-)^2}{2} + \cos \left(\frac{\phi^-}{2} + \gamma \right). \end{aligned} \quad (13)$$

To get the Eq. (13) we have used the foot-strike conditions given by Eqs. (7) and (8). We have also assumed that controller is not going to be aware of the step down disturbance, so we set $h = 0$ (see Eq. (5)). This leads to the condition $\phi^- = 2\theta^-$ and is used to write the cos expression on the right side of Eq. (13) in terms of ϕ^- .

4.3 Discrete control lyapunov function (DCLF)

DCLF was explained in the last paragraph of Sec. 2 and illustrated in Fig. 1. We provide mathematical details next.

First, we need a period-one limit cycle so that we are able to construct a DCLF to stabilize it. A period-one limit cycle is specified by setting $\dot{\theta}_{k+1} = \dot{\theta}_k = \dot{\theta}_0$ in Eq. (11) and solving for $\phi^- = \phi_0$ to get the following

$$\dot{\theta}_0 = F(\dot{\theta}_0, \phi_0) \quad (14)$$

Second, we define the DCLF as follows

$$V(\Delta\dot{\theta}_k) = \Delta\dot{\theta}_k^2 = (\dot{\theta}_k - \dot{\theta}_0)^2, \quad (15)$$

where $V(0) = 0$ and $V(\Delta\dot{\theta}_k) > 0$ at the mid-stance event, $\theta = 0$. For the system to be asymptotically stable, the following condition needs to be satisfied

$$V(\Delta\dot{\theta}_{k+1}) - V(\Delta\dot{\theta}_k) = (\dot{\theta}_{k+1} - \dot{\theta}_0)^2 - (\dot{\theta}_k - \dot{\theta}_0)^2 < 0 \quad (16)$$

However, we are interested in exponential stability so we set the following condition

$$V(\Delta\dot{\theta}_{k+1}) - V(\Delta\dot{\theta}_k) = -cV(\Delta\dot{\theta}_k), \quad (17)$$

where c is a user chosen positive constant such that, $0 < c < 1$. Thus the condition for exponential stability can be written as

$$\begin{aligned} \Delta\dot{\theta}_{k+1}^2 - (1-c)\Delta\dot{\theta}_k^2 &= 0 \\ \implies \left(F(\dot{\theta}_k, \phi^-) - \dot{\theta}_0\right)^2 - (1-c)\left(\dot{\theta}_k - \dot{\theta}_0\right)^2 &= 0. \end{aligned} \quad (18)$$

The stance leg velocity at mid-stance, $\dot{\theta}_k$, is measured and for given c and $\dot{\theta}_0$, the swing leg angle just before foot-strike, ϕ^- , is solved using the above equation. The choice of c determines the rate of decay of a perturbation in the mid-stance velocity; a larger value of c indicates faster convergence. When $c = 1$ there is full correction of disturbances in a single step, also known as one-step dead-beat control [20]. Dead-beat control gives the fastest convergence, ‘‘dead-beat’’ convergence [21].

Third, we need to define a controller for the hip torque based on the computed swing leg angle at foot-strike. We use 2 third-order, time-based trajectories for the swing leg; one for the instant from mid-stance to foot-strike and another one from instant after foot-strike to mid-stance. Each of the third order polynomials has four coefficients that are computed based on the initial and final values of the position and velocity which are completely known. One also needs the time from mid-stance to instant just before foot-strike ($T_{\text{mid-fs}}$) and from instant after foot-strike to mid-stance ($T_{\text{fs-mid}}$). These can be computed as

follows

$$T_{\text{mid-fs}} = \int_{\theta=\frac{\phi^-}{2}}^{\theta=0} \frac{d\theta}{\sqrt{(\dot{\theta}_k)^2 + 2(\cos\gamma - \cos(\theta - \gamma))}} \quad (19)$$

$$T_{\text{fs-mid}} = \int_0^{\theta=\frac{\phi^-}{2}} \frac{d\theta}{\sqrt{(\dot{\theta}^+)^2 + 2(\cos(\frac{\phi^-}{2} - \gamma) - \cos(\theta - \gamma))}} \quad (20)$$

The time-based trajectory is open-loop but we have a proportional derivative controller on the position at the end of the time-based trajectory. The controller comes into effect only when the time from mid-stance to foot-strike is greater than the predicted time using Eq. (19) (e.g., during a step-down disturbance).

4.4 Why use mid-stance position for Poincaré map?

We use mid-stance position, defined as $\theta = 0$, for the Poincaré section. In general, any section such as $\theta = \text{constant}$ will work just as well. This particular choice of Poincaré section enables us to choose V to be a function of only the stance leg velocity, (e.g., $V(\Delta\dot{\theta}_k) = \Delta\dot{\theta}_k^2$). This allows us to do a hierarchical control: the swing leg is controlled in the continuous sense to affect the stance leg velocity in the discrete sense or between steps. On the other hand, a section before or after foot-strike would be a function of stance leg and swing leg angle, $\cos(\phi - \theta) - \cos(\theta) = 0$. In this case, V needs to be a function of stance leg and swing leg position and velocity, that is, $V(\Delta\theta_k, \Delta\dot{\theta}_k, \Delta\phi_k, \Delta\dot{\phi}_k)$. This complicates the controller design. Moreover, in a practical sense, the instant just before or after foot-strike is most prone to modeling errors (e.g., mis-estimation of ground height) and noise (e.g., two sensors, stance and swing leg angles, are needed and each of them would have their own noise levels). Perhaps the best choice for the Poincaré section is the instant after foot-strike but following the decay of vibrations due to foot-strike because that instant gives the swing leg the entire step for adjusting the step length.

4.5 Eigenvalue-based controller

The eigenvalue-based controller imparts orbital stability in the linearized sense. The Poincaré map is linearized based on the state and suitable control action. Then the eigenvalues are modulated/placed using the control action. We describe the mathematical details next.

First, we linearize the Poincaré map given by Eq. (11) about the passive limit cycle:

$$\Delta\dot{\theta}_{k+1} = A\Delta\dot{\theta}_k + B\Delta\phi^-, \quad (21)$$

where $A = \frac{\partial\dot{\theta}_{k+1}}{\partial\dot{\theta}_k}$ and $B = \frac{\partial\dot{\theta}_{k+1}}{\partial\phi^-}$.

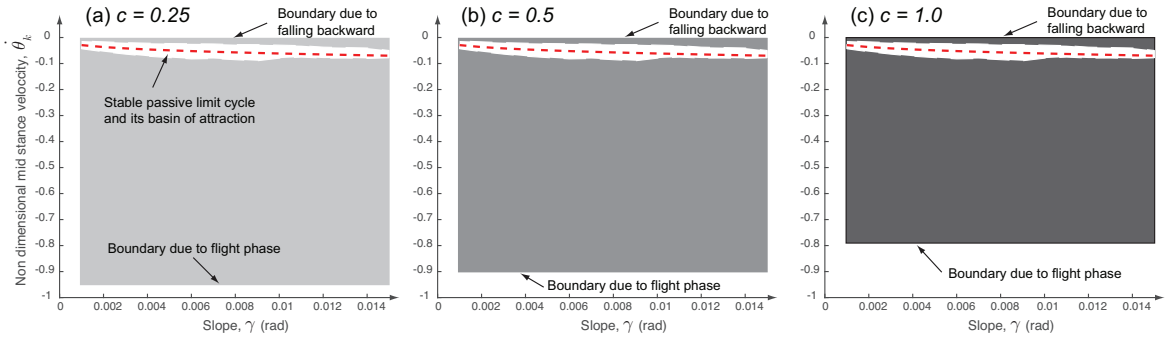


Fig. 3. Basin of attraction: Uncontrolled/passive dynamic case (white region) and DCLF (gray region includes the white region for the uncontrolled case) for three values of c . The red dashed line shows the passive limit cycle.

Next, we assume a linear controller, $\Delta\phi^- = -K\Delta\theta_k$ and substitute it in Eq. (21) to obtain

$$\Delta\dot{\theta}_{k+1} - (A - BK)\Delta\dot{\theta}_k = 0. \quad (22)$$

Finally, we compare Eq. 22 with Eq. 18 to compute the gain K . We have

$$A - BK = \sqrt{1-c} \quad \implies K = \frac{A - \sqrt{1-c}}{B}. \quad (23)$$

This allows use to compare the DCLF controller with the eigenvalue-based controller in an objective way. Note that A , B , and K are scalars in our problem and we have used the positive root from Eq. 18 (that is, $\sqrt{1-c}$) to ensure monotonic decay in the perturbed state, $\Delta\dot{\theta}_{k+1}$.

5 Results

All computations are done using MATLAB. We used *ode113* for integrating the equations of motion and *fsolve* to find the limit cycle. To find the passive limit cycle, we chose $\gamma = 0.009$ and set the controller off ($\tau = 0$). Our passive limit cycle has the fixed point; $\theta_0 = 0$, $\dot{\theta}_0 = -0.0593$, $\phi_0 = -0.0532$, and $\dot{\phi}_0 = -0.3397$. The swing leg position and velocity just before foot-strike are; $\phi^- = -0.4006$ and $\dot{\phi}^- = 3.5 \times 10^{-4} \sim 0$. We used central differencing with step size of 10^{-5} to compute $A = 0.3711$ and $B = -0.4026$. We also found the Jacobian of the Poincaré section using central difference. The largest eigenvalue of the Jacobian is 0.5891, which is less than 1, indicating that the passive limit cycle is stable

5.1 Stability

We use the basin of attraction metric to compare the stability of the uncontrolled case (passive dynamic walking) with our control approach. The basin of attraction is defined as the set of initial conditions that converge to the limit cycle as time goes to infinity [13]. To compute the basin of attraction we proceed as follows. We perturb the mid-stance velocity

from its nominal value and do forward simulations of the walker for 50 steps. We find the upper and lower limits of the mid-stance velocity that allows the walker to complete 50 steps without falling down. The limits give the boundary of the basin of attraction. Note that 50 steps are large enough to be able to see the effect of the perturbation but small enough that we are able to obtain results in a short period of time. We repeat the above procedure for different ramp slopes and with and without control.

Figure 3 shows the basin of attraction for the simplest walker with and without control as a function of ramp slope. The white region shows the basin of attraction for the passive or uncontrolled case. The gray region (including the white region) shows the basin of attraction for three values of c . The basin of attraction is substantially improved using our control approach. We compute the area of the basin of attraction for the uncontrolled case and the three control cases with different c values using numerical quadrature. Then we find the ratio of areas for a specific c value to the uncontrolled case. We found ratios of 19.56, 18.60, 16.4 for $c = 0.25$, $c = 0.5$, $c = 1$ respectively. This indicates that our controller was able to increase the region of attraction by an order of magnitude. Further, we note that $c = 1$ has the smallest basin of attraction and this increases as c decreases. This can be explained as follows: A larger c value corresponds to a faster convergence to the limit cycle. For example, when $c = 1$, the convergence is in a single step. This is achieved by taking a bigger than nominal step as reduction in velocity between steps is directly proportional to the step length [19, 22]. But a bigger step will lead to flight phase at a lower velocity (see Eq. (10)). As c decreases, the controller chooses shorter steps, leading to higher velocities for flight phase. Consequently, as c decreases, the basin of attraction increases.

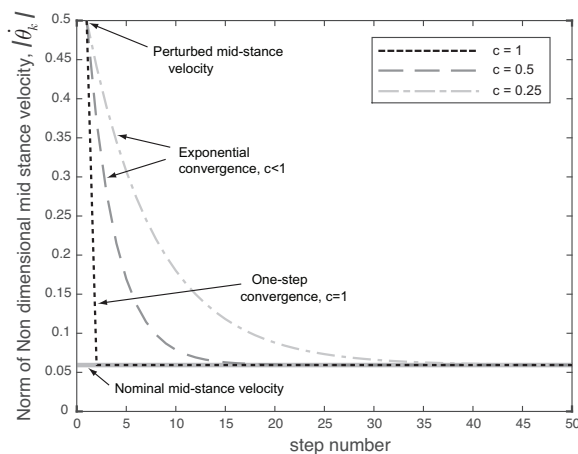


Fig. 4. Rate of stabilization: The convergence is exponential for $0 < c < 1$ and “dead-beat” for $c = 1$ for state perturbation $\hat{\theta} = -0.5$ for $\gamma = 0.009$.

Next, we demonstrate the exponential stabilization provided by the DCLF. We perturb the mid-stance velocity to $\hat{\theta}_k = -0.5$ at a slope of $\gamma = 0.009$ and plot the mid-stance velocity as a function of step number to obtain Fig. 4. For $0 < c < 1$ there is exponential stabilization (dashed and dash-dot lines) and a larger value of c gives faster convergence to the nominal mid-stance velocity. However, $c = 1$ (dotted line) leads to the condition $\hat{\theta}_{k+1} = \hat{\theta}_0$ (see Eqs. (15) and (17)) or a “dead-beat” convergence in a single step. This is faster than exponential convergence.

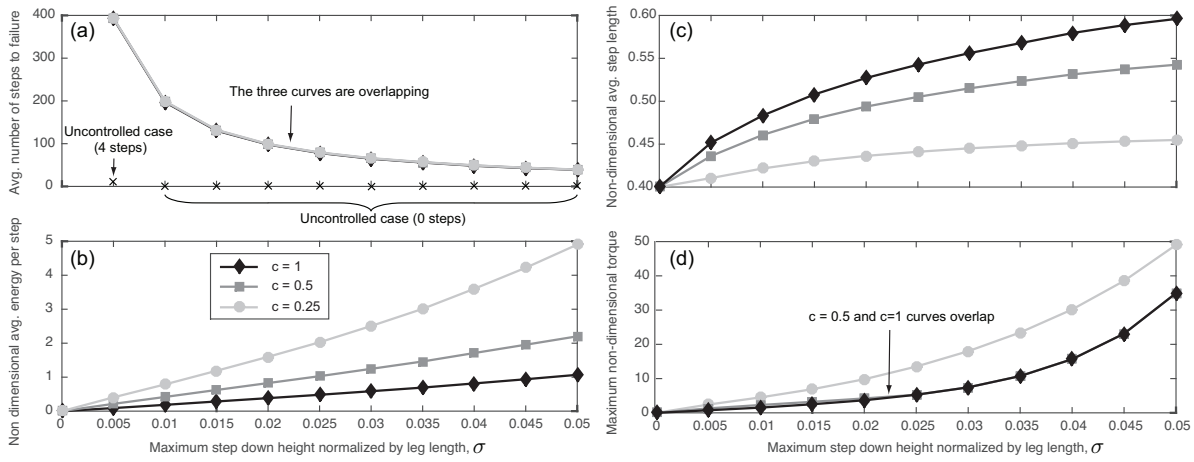


Fig. 5. Robustness to step-down disturbance for different c values: (a) Average number of steps to failure, (b) average energy (absolute value of mechanical work done by the hip actuator) used per step, (c) average step length, and (d) maximum torque needed.

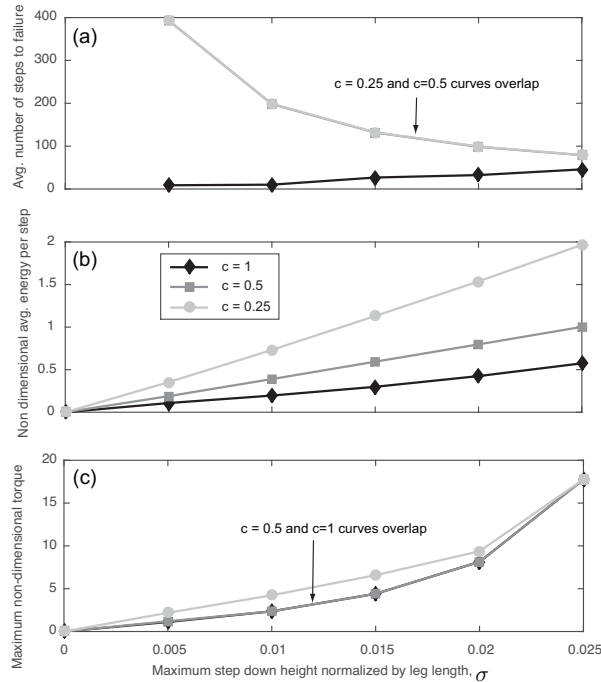


Fig. 6. Sensitivity of DCLF to modeling errors: (a) Average number of steps to failure, (b) average energy (absolute value of mechanical work done by the hip actuator) used per step, (c) maximum torque needed. The slope, γ , is 1.1 of its actual value.

5.2 Robustness

We evaluate robustness by computing the average number of steps that the controller can withstand without the flight phase or falling backward on uneven terrain chosen from a random distribution with maximum height of σ [12]. Note that σ can be interpreted as a maximum step down normalized against the leg length. We create 10 terrains with 400 steps each, selected from a random distribution with maximum height σ . For each terrain, we do forward simulations of the system for a given value of c . We evaluate the average number of steps and average energy used per step for the 10 terrains for different values of σ and c .

Figure 5 (a) shows results for robustness as a function of σ for different values of c for a slope of $\gamma = 0.009$. The average number of steps is infinity for $\sigma = 0$ which corresponds to no disturbance. As σ increases, the average number of steps decreases steadily as shown in the figure. The average number of steps to failure is almost the same for each value of c . The average number of steps to failure is 393 at $\sigma = 0.005$ and decreases to 40 at $\sigma = 0.05$. We found that for very low values of c , $0 < c \leq 0.005$, the robustness dropped appreciably with increase in step height. We also did the robustness test without any control. We observed that the average number of steps to failure is 4 at $\sigma = 0.005$ and 0 at $\sigma = 0.05$.

Figure 5 (b) demonstrates the average energy used per step as a function of σ . We obtain the energy usage by integrating the absolute value of mechanical work done by the hip actuator as the robot walks on the terrain and dividing it by the total number of successful steps. For a given controller specified by c , the average energy per step increases with σ . This is because a larger σ leads to a larger deviation from the limit cycle and consequently more energy is needed to get back to the nominal trajectory. For a given σ , the average energy per step is the lowest for $c = 1$ and increases as c decreases. This can be explained as follows: larger value of c implies a faster convergence to the limit cycle and hence lower energy usage since the limit cycle is passive (energy usage of zero).

Figure 5 (c) illustrates the average step length, the control strategy, as a function of σ . Each data point for a particular σ and c was obtained by averaging the step length for the 10 simulation runs. For a given maximum step down, σ , the largest average step length is for $c = 1$ and decreases as c decreases. This is because larger c values correspond a greater decrease of the DCLF, which is possible by taking a bigger step length. Also, the average step length increases as the step down, σ , increases. Since a larger step down leads to a larger deviated mid-stance velocity, the average step length needs to consequently increase to regulate the mid-stance velocity at the subsequent step.

Finally, Fig. 5 (d) depicts the maximum torque needed as a function of σ . This was obtained by searching for maximum torque across the 10 simulations runs for a given σ . The maximum torque is almost the same for $c = 0.5$ and $c = 1$ but is greater for $c = 0.25$ for a given σ . This indicates that low values of c require more torque, consequently, a larger actuator for stabilization. The maximum torque increases as the maximum step down increases for a given c because bigger step down requires bigger step length and hence, higher torque.

Figure 6 compares the sensitivity of the DCLF controller to modeling errors. To generate these simulations, we provided the DCLF controller with a slope of $\gamma = 0.01$, which is 10% greater than the actual value of $\gamma = 0.009$, a modeling error. The robustness measured by the average number of steps to failure for $c = 0.25$ and $c = 0.5$ is virtually unchanged (compare Fig. 5 (a) with Fig. 6 (a)). However, the robustness decreases substantially for $c = 1$. This can be explained as follows: a larger than actual γ in the model, leads to a bigger step and to a greater energy lose during collision, and consequently the model falls backwards. The average number of steps walked before failure for $c = 1$ is about 9 at $\sigma = 0.005$ and it increases to about 40 at $\sigma = 0.025$. This is because an increase in step down disturbance nullifies the effect of this particular modeling error: bigger than actual γ in the model leads to a bigger than the actual step but this regulates the walking speed better as the step down disturbance is increased. Fig. 6 (b) shows that $c = 1$ is more energy-efficient, followed by $c = 0.5$ and $c = 0.25$. The maximum torques for $c = 1$ and $c = 0.5$ are the same while those for $c = 0.25$ are much higher.

Figure 7 compares the DCLF controller with the eigenvalue-based controller for $c = 0.9$, which corresponds to an

eigenvalue of 0.32. Fig 7 (a) demonstrates that both the controllers have similar robustness. However, the DCLF controller is more energy-efficient (see Fig. 7 (b)) and requires a lower maximum torque 7 (d). The main difference between the two controllers is that eigenvalue-based controller uses a linearization while DCLF is based on the actual model, which is non-linear. In this case, the linearization leads the eigenvalue-based controller to over correct by taking longer than ideal steps as shown in Fig. 7 (c).

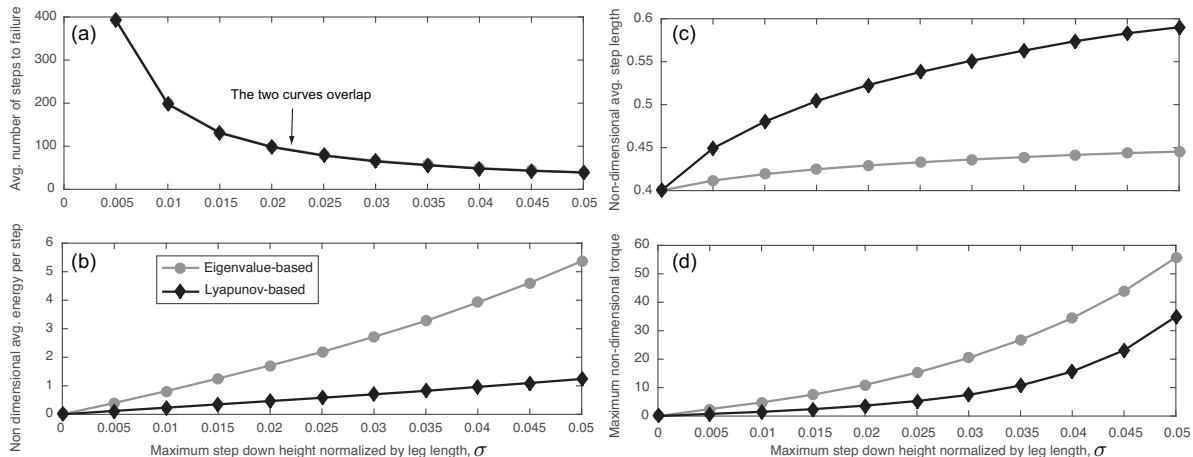


Fig. 7. Robustness to step-down disturbance comparing Eigenvalue-based with Lyapunov based stability: (a) Average number of steps to failure, (b) average energy (absolute value of mechanical work done by the hip actuator) used per step, (c) average step length, and (d) maximum torque needed. All plots are for $c = 0.9$ which corresponds to an eigenvalue of $\sqrt{1 - c} = 0.32$.

6 Discussion

We have presented the discrete control Lyapunov function (DCLF) to exponentially stabilize the simplest walking model with a hip actuator. Our control approach is able to enlarge the basin of attraction by an order of magnitude and increase the average number of steps to failure by two orders of magnitude over the passive dynamic walking. We compared the DCLF controller with a one-step dead-beat controller and eigenvalue-based controller. We found that all three controllers had similar robustness. However, the dead-beat controller was the most energy-efficient and required lower maximum torque followed by the DCLF controller and finally the eigenvalue-based controller.

Theoretically, one-step dead-beat stabilization is better than exponential stabilization in terms of convergence rate and final value achieved. While one-step dead-beat stabilization converges fully to the nominal value in a single step, there will always be some finite (usually very small) error between the actual and nominal state for exponential stabilization. Also, the one-step dead-beat stabilization leads to lower energy usage and lower maximum torque than exponential stabilization suggesting its superior performance. However, the robustness of the one-step dead-beat stabilization reduces quickly in the presence of modeling errors. Thus, in practice, an exponential stabilized controller is preferred.

The most common stability metric used in legged locomotion is the maximum eigenvalue of Jacobian of the Poincaré map [1]. This is a linear stability measure that holds true for small state perturbations only. On the other hand, our measure

is a non-linear measure that works for large state perturbations. Some authors have optimized the maximum eigenvalues to create stable walking gaits [23, 24]. The issue with this approach is that the maximum eigenvalue sometimes is a non-smooth function of the robot and control parameters [6], which will lead to numerical issues when used in conjunction with gradient-based optimization methods. In contrast, our stability measure is a function of the velocity of the under-actuated degree of freedom only. Although velocity is discontinuous at foot-strike, the optimization can be made smooth by using multiple shooting method [25]. This makes it possible to apply gradient-based methods which lead to faster convergence to the optimal solution for smooth problems. Note that multiple shooting method allows one to handle discontinuities in the physics of the problem (e.g., due to foot-strike), and not in the control parameters.

The eigenvalues of the Jacobian of the Poincaré map can be manipulated using a feedback controller as was done in this paper. We used pole placement to place the eigenvalues but one can also use a discrete linear quadratic regulator, DLQR (e.g., [26]). In the feedback approach, eigenvalues of the closed-loop system are manipulated using pole placement or DLQR while in the optimization approach discussed in the previous paragraph, the eigenvalues of the open-loop system are modified. The feedback control method does not require smoothness of the eigenvalues and is computationally faster than the optimization-based method. However, since the the eigenvalue-based feedback controller is based on linearization of the system, it usually performs poorly in the presence of large perturbations (disturbances, sensor noise, and modeling errors) as we have seen in this paper.

One advantage of the DCLF is that we are able to construct a controller for a given stability bound during the design phase. This is in contrast to the more common approach of checking stability after control design has taken place, which is time consuming and provides limited stability guarantees. Specifically, in our approach Eq. (17) gives the stability condition and can be incorporated easily into the control design framework (e.g., optimization). One can incorporate stability in the controller design by maximizing the mean first-passage time using a stochastic optimization [12]. Another approach is to find a heuristic control strategy that imparts gait stability such as swing leg retraction (swing leg moves backwards just before foot-strike) [27]. But recent results have shown that under certain conditions (big slopes and dead-beat stabilization), swing-leg protraction can also impart gait stability [28].

The DCLF approach uses a single measurement, the stance leg velocity at mid-stance, to adjust the foot step location. A practical way to achieve this would be to place an inertial measurement unit (IMU) on the torso. The IMU serves two purposes: one, it detects mid-stance event using the accelerometer; and two, the forward velocity using the gyroscope. Another feature of DCLF is that it is predictive, that is, the control technique is able to calculate the desired foot placement at mid-step, giving ample time for the hip actuator to perform the required control action.

Our control approach has several limitations. We use a single measurement at mid-stance that determines the foot placement. Thus our method is sensitive to measurement noise and model parameters. Our method works best when the sequence of events is: disturbance followed by measurement (at mid-stance) followed by control action (at foot-strike). However, our approach might show poor performance when the sequence of events is: measurement followed by disturbance followed by control action. Because in this case, the measurement cannot act on the disturbance immediately but needs to wait for one complete step to be able to do any corrections. The sensitivity can be tackled by incorporating a robust control

law under bounded uncertainty [29]. This model is incapable of walking on level ground because energy lost to collision during foot placement cannot be recovered. Recently, we have extended these results to walking on level ground by adding ankle push-off control to the model [30]. DCLF can also be extended to 3D by using a decoupled approach where the step length and step width are independently controlled to stabilize walking motion in the fore-aft and side-to-side direction respectively. Additional degrees of freedom (e.g., torso, knees) provide additional control actions that can be exploited for stabilization (e.g., [16]). We ignore actuator limits in our calculations which could substantially limit the basin of attraction of the controller. We also limit ourselves to walking solutions and assume controller failure when there is a flight phase. The latter two limitations of actuator limits and running solutions can easily be incorporated and will be basis for future work.

Acknowledgment

This work was supported by NSF grant IIS 1566463. The authors would like to thank Ahmad F. Taha for helpful discussions.

References

- [1] McGeer, T., 1990. "Passive dynamic walking". *The International Journal of Robotics Research*, **9**(2), pp. 62–82.
- [2] Owaki, D., Koyama, M., Yamaguchi, S., Kubo, S., and Ishiguro, A., 2010. "A two-dimensional passive dynamic running biped with knees". In International Conference on Robotics and Automation, Anchorage, Alaska, USA, pp. 5237–5242.
- [3] Steinkamp, P., 2017. "A statically-unstable passive hopper: design evolution". *Journal of Mechanisms and Robotics*, **9**(1), pp. 011006–011006–7.
- [4] Schwab, A., and Wisse, M., 2001. "Basin of attraction of the simplest walking model". In Proceedings of the ASME design engineering technical conference, Vol. 6, pp. 531–539.
- [5] Hobbelen, D., and Wisse, M., 2007. "A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm". *IEEE Transactions on robotics*, **23**(6), pp. 1213–1224.
- [6] Goswami, A., Thuilot, B., and Espiau, B., 1998. "A study of the passive gait of a compass-like biped robot: symmetry and chaos". *The International Journal of Robotics Research*, **17**(12), pp. 1282–1301.
- [7] Iida, F., and Tedrake, R., 2010. "Minimalistic control of biped walking in rough terrain". *Autonomous Robots*, **28**(3), pp. 355–368.
- [8] Manchester, I., Mettin, U., Iida, F., and Tedrake, R., 2011. "Stable dynamic walking over uneven terrain". *The International Journal of Robotics Research*, **30**(3), pp. 265–279.
- [9] Ames, A. D., Galloway, K., Sreenath, K., and Grizzle, J. W., 2014. "Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics". *IEEE Transactions on Automatic Control*, **59**(4), pp. 876–891.
- [10] Dingwell, J. B., and Kang, H. G., 2007. "Differences between local and orbital dynamic stability during human walking". *Journal of biomechanical engineering*, **129**(4), pp. 586–593.

- [11] Wieber, P.-B., 2008. “Viability and predictive control for safe locomotion”. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 1103–1108.
- [12] Byl, K., and Tedrake, R., 2009. “Metastable walking machines”. *The International Journal of Robotics Research*, **28**(8), pp. 1040–1064.
- [13] Strogatz, S., 1994. *Nonlinear dynamics and chaos*. Addison-Wesley Reading.
- [14] Tedrake, R., Manchester, I., Tobenkin, M., and Roberts, J., 2010. “Lqr-trees: Feedback motion planning via sums-of-squares verification”. *The International Journal of Robotics Research*, **29**(8), pp. 1038–1052.
- [15] McGeer, T., 1993. “Dynamics and control of bipedal locomotion”. *Journal of Theoretical Biology*, **163**(3), pp. 277–314.
- [16] Kuo, A. D., 1999. “Stabilization of lateral motion in passive dynamic walking”. *The International journal of robotics research*, **18**(9), pp. 917–930.
- [17] Slotine, J.-J. E., Li, W., et al., 1991. *Applied nonlinear control*. prentice-Hall Englewood Cliffs, NJ.
- [18] Garcia, M., Chatterjee, A., Ruina, A., and Coleman, M., 1998. “The simplest walking model: Stability, complexity, and scaling”. *ASME J. of Biomech. Eng.*, **120**, pp. 281–288.
- [19] Bhounsule, P. A., 2015. “Control of a compass gait walker based on energy regulation using ankle push-off and foot placement”. *Robotica*, **33**(06), pp. 1314–1324.
- [20] Antsaklis, P., and Michel, A., 2006. *Linear systems*. Birkhauser.
- [21] Gauthier, G., and Boulet, B., 2005. “Convergence analysis of terminal ilc in the z domain”. In Proceedings of the American Control Conference, Vol. 1, p. 184.
- [22] Ruina, A., Bertram, J., and Srinivasan, M., 2005. “A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition”. *Journal of theoretical biology*, **237**(2), pp. 170–192.
- [23] Mombaur, K., Longman, R., Bock, H., and Schlöder, J., 2005. “Open-loop stable running”. *Robotica*, **23**(01), pp. 21–33.
- [24] Chevallereau, C., Grizzle, J., and Shih, C., 2009. “Asymptotically stable walking of a five-link underactuated 3-d bipedal robot”. *IEEE Transactions on Robotics*, **25**(1), pp. 37–50.
- [25] Betts, J., 1998. “Survey of numerical methods for trajectory optimization”. *Journal of Guidance control and dynamics*, **21**(2), pp. 193–207.
- [26] Bhounsule, P. A., Ruina, A., and Stiesberg, G., 2015. “Discrete-decision continuous-actuation control: balance of an inverted pendulum and pumping a pendulum swing”. *Journal of Dynamic Systems, Measurement, and Control*, **137**(5), p. 051012.
- [27] Wisse, M., Atkeson, C., and Kloimwieder, D., 2005. “Swing leg retraction helps biped walking stability”. In Proceedings of 2005 IEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan.
- [28] Bhounsule, P. A., and Zamani, A., 2017. “Stable bipedal walking with a swing-leg protraction strategy”. *Journal of Biomechanics*, **51**, pp. 123–127.
- [29] Kelly, M., and Ruina, A., 2015. “Non-linear robust control for inverted-pendulum 2d walking”. In 2015 IEEE Interna-

tional Conference on Robotics and Automation (ICRA), IEEE, pp. 4353–4358.

- [30] Zamani, A., and Bhounsule, P. A., 2017 (to appear). “Foot placement and ankle push-off control for the orbital stabilization of bipedal robots”. In 2017 IEEE International Conference on Intelligent Robots and Systems (IROS).

List of Figures

1 Difference between continuous and discrete control Lyapunov function methodologies: (a) Continuous control Lyapunov function (CCLF) constructs a controller that keeps the trajectory within a tube *along the trajectory*. (b) Discrete control Lyapunov function (DCLF) constructs controller that keeps the trajectory in the basin of attraction only *at the Poincaré section*. 2

2 Model: The simplest slope walking model analyzed by Garcia et al. [18]. 5

3 Basin of attraction: Uncontrolled/passive dynamic case (white region) and DCLF (gray region includes the white region for the uncontrolled case) for three values of c . The red dashed line shows the passive limit cycle. 10

4 Rate of stabilization: The convergence is exponential for $0 < c < 1$ and “dead-beat” for $c = 1$ for state perturbation $\dot{\theta} = -0.5$ for $\gamma = 0.009$ 11

5 Robustness to step-down disturbance for different c values: (a) Average number of steps to failure, (b) average energy (absolute value of mechanical work done by the hip actuator) used per step, (c) average step length, and (d) maximum torque needed. 12

6 Sensitivity of DCLF to modeling errors: (a) Average number of steps to failure, (b) average energy (absolute value of mechanical work done by the hip actuator) used per step, (c) maximum torque needed. The slope, γ , is 1.1 of its actual value. 12

7 Robustness to step-down disturbance comparing Eigenvalue-based with Lyapunov based stability: (a) Average number of steps to failure, (b) average energy (absolute value of mechanical work done by the hip actuator) used per step, (c) average step length, and (d) maximum torque needed. All plots are for $c = 0.9$ which corresponds to an eigenvalue of $\sqrt{1 - c} = 0.32$ 14