Non-linear robust control for inverted-pendulum 2D walking

Matthew Kelly[1] and Andy Ruina[2]

***Abstract*— We present an approach to high-level control for bipedal walking exemplified with a 2D point-mass inextensible-legs inverted-pendulum model. Balance control authority here is only from step position and trailing-leg push-off, both of which are bounded to reflect actuator limits. The controller is defined implicitly as the solution of an optimization problem. The optimization robustly avoids falling for given bounded disturbances and errors and, given that, minimizes the number of steps to reach a given target speed. The optimization can be computed in advance and stored for interpolated real-time use online. The general form of the resulting optimized controller suggests a few simple principles for regulating walking speed: 1) The robot should take bigger steps when speeding up and should also take bigger steps when slowing down 2) push-off is useful for regulating small changes in speed, but it is fully saturated or inactive for larger changes in speed. While the numerically optimized model is simple, the approach should be applicable to, and we plan to use it for, control of bipedal robots in 3D with many degrees of freedom.**

I. INTRODUCTION

In the long run we would like to explain, and robotically reproduce, the efficiency, speed and versatility of human bipedal locomotion in various terrains. Here we work towards this goal by designing a controller for walking, focusing on the ability to avoid falling in level walking, considering various disturbances. Bipedal locomotion violates many common assumptions in basic classical control: it is nonlinear, non-holonomic [1], has discontinuities, and changes governing equations during the motion. Hence there is no generally-accepted classical-controls approach to stabilizing walking. Here we pursue a hierarchical controller strategy for walking control.

Most balance controllers use some sort of hierarchical structure, factoring the control problem for walking into two parts: a high-level controller concerned with overall robot position and balance, and a low-level controller that deals with the many internal degrees of freedom. The high-level controller may specify foot placement, leg length, and ground reaction forces and torques to achieve balance; while the low level controller determines individual motor commands to realize these sub-goals.

A few examples of robots that use a hierarchical approach include Atlas [2], Asimo [3], MABEL [4], and the Cornell Ranger [5], all of which use a balance controller based on one or another simple low-dimensional model. For instance, the capture point controller [6]–[9] uses the linear inverted pendulum (LIP) model to plan future foot step locations and a trajectory for the center of mass. These high-level commands are then realized by giving commands to individual motors based on inverse kinematics and dynamics of a full high-dimensional model of the robot.

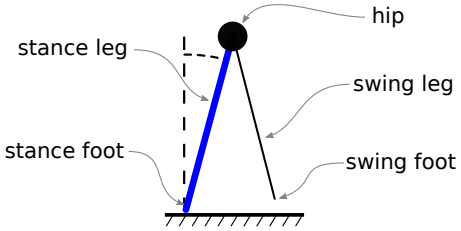

Fig. 1. **Point-mass walking model.** All mass is concentrated at a point at the hip. The legs are mass-less. There are two controls: The relative angle of the legs at heel-strike, and the size of the push-off impulse by the trailing leg just before heel-strike. Uncertainties in sensing, model, and actuation, as well as actuator limits on push-off impulse and step length, are discussed in the text.

Although these simple models are primarily used to make the locomotion balance problem tractable, they also seem to be genuinely good models for balance control [10]. Because of their utility, however motivated, most successful simple models are based on a point mass at or near the hip, light or mass-less legs, and actuation to control the time and location of foot falls and the ground reaction force. The models tend (sensibly, we think) to neglect the effects on balance of upper body motions and the details of leg swing between steps.

Within the class of point-mass models just mentioned, there are still choices. One common choice is to use the 'linear inverted-pendulum' (LIP) model, in which the height of the point mass is held constant (e.g. with 'capture point' and 'zero moment point' control). Here, however, we use an 'inverted-pendulum' (IP) model which has constant leg length instead of constant hip height. The IP model has not been as frequently used for walking balance control as the LIP model, in part because of the mathematical simplifications of the LIP (e.g. [11]).

We use the IP model for control because it uses energy more effectively than the LIP model. More precisely, Srinivasan and Ruina [12], [13] used numerical trajectory optimization to find positive-work minimizing gaits on a general point-mass walker and found that the energy-minimal walking gait used constant-length legs: the IP model of walking. In this model the stance leg length remains constant, and the push-off at the end of the stance phase is (at least approximately) impulsive. This is a powered version of the

*This work is supported by the National Science Foundation (fellow ID: 2011116308) and the National Robotics Initiative (grant number 1317981)

[1]Matthew Kelly is a PhD student in Mechanical Engineering, Cornell University, Ithaca NY 14850, USA mpk72@cornell.edu

[2]Andy Ruina is a Professor of Mechanical Engineering, Cornell University, Ithaca NY 14850, USA ruina@cornell.edu

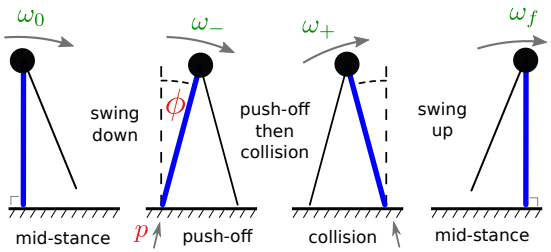

Fig. 2. **One walking step.** A step starts when the stance leg passes clockwise through the vertical orientation. The hip then falls to the right. Just before the swing leg hits the ground there is push-off from the ground on the stance leg. Immediately after push-off, the swing leg hits the ground and becomes the stance leg. The former stance leg leaves the ground and the new stance leg swings up to the start of the next step. In the leftmost and rightmost pictures, the angle of the swing leg is arbitrary (because it is fully controlled).

so-called 'Simplest Walker' [14]. The energy effectiveness of impulsive push-off and straight leg walking is also discussed in Kuo [15] for a similar model. Because of the IP's good energetics, it is the basis of our walking robots. In the design of the controller here, however, we do not explicitly consider energy, focusing instead on robustness to disturbances and quick return to a nominal gait.

Zaytsev [16] has introduced a simple walking controller for this IP model, based on finding simple control laws that maximize the distance from failure boundaries. Here, we extend Zaytsev's work by optimizing an arbitrary-form controller that is robust to disturbances and also stable. Additionally, the design process here is fast enough that it can be repeated to stabilize a range of speeds thus generating a full feedback policy for a range of walking-speed goals. Despite a difference of methods with Zaytsev, there are common results (discussed below). This may indicate that we are extracting features that are necessary aspects of any robust walking balance controller that is reasonably constrained by actuator and sensor limits.

II. WALKING MODEL

Our model consists of a point-mass hip on mass-less inextensible legs (Fig. 1). The model has two control inputs at each step: the angle of the stance leg (ϕ) at heel-strike, and the push-off impulse (p) that occurs immediately before heel-strike.

A step comprises four distinct phases, starting from a mid-stance (more or less a Poincaré Section) where the stance leg is vertical and rotating clockwise . These phases are illustrated in Fig. 2: a swing-down to the step angle (ϕ); followed by an impulsive push-off (p); then heel-strike and leg-switch; and finally a swing-up to mid-stance. Throughout the paper we measure walking speed in the various phases of the gait by the dimensionless (divide by $\sqrt{g/\ell}$) angular velocity ω of the stance leg.

A. Actuation

The step length is determined by the step angle: step length $= 2\ell \sin\phi$. We neglect swing-leg inertia: its motion does not affect the motion of the stance leg. The step angle is bounded (constrained) to mimic the joint limits of a real robot.

The impulsive push-off is meant to be a proxy for the energy injected into the walking motion by the extension of the trailing leg (ankle extension). The bounds on the push-off are a proxy for the maximum power available for leg extension.

B. Equations of Motion

The model is a simple inverted pendulum, but with impulses and resets at each step. The parameters m, g, and ℓ represent the robot's mass, gravitational acceleration, and leg length respectively. The control variables ϕ and p represent the step angle and push-off impulse respectively.

The step starts with the stance leg vertical and rotating with speed (ω_k) towards heel-strike. The angular rotation rate immediately before push-off can be obtained via conservation of energy:

$$\omega^- = \sqrt{(\omega_k)^2 + \frac{2g}{\ell}(1-\cos\phi)}. \tag{1}$$

Next, a push-off impulse is applied to the point-mass along the stance leg, changing the hip's velocity vector. After that, the swing leg becomes the new stance leg as it collides with the ground, exerting another impulse on the point-mass hip. The composition of these two collisions, governed by angular momentum balance about the new stance foot, yields the rotational speed of the new stance leg (ω^+),

$$\omega^+ = (\omega^-)\left(\cos^2\phi - \sin^2\phi\right) + \frac{2p}{ml}\cos\phi\sin\phi. \tag{2}$$

After the two collisions, the stance leg swings up to the next mid-stance, again ruled by conservation of energy,

$$\omega_{k+1} = \sqrt{(\omega^+)^2 - \frac{2g}{\ell}(1-\cos\phi)}. \tag{3}$$

C. Feasibility conditions

A walking step is considered successful (i.e. the robot did not fail) if two conditions are met: first, the stance leg is always in compression (this is also a no-flight condition), and second, the speed after the heel-strike must be sufficient to continue on to the next step without falling over backwards. These conditions are expressed with the four inequality constraints:

$$\omega^- < \sqrt{\frac{g}{\ell}\cos\phi} \tag{4}$$

$$0 < (2m\ell)(\omega^-)\cos\phi\sin\phi - p\left(\cos^2\phi - \sin^2\phi\right)2\ell \tag{5}$$

$$\sqrt{\frac{2g}{\ell}(1-\cos\phi)} < \omega^+ < \sqrt{\frac{g}{\ell}\cos\phi}. \tag{6}$$

The first inequality (4) is a restriction on the speed before the push-off impulse. The second inequality (5) is a restriction on the collision impulse - the collision cannot pull the walker towards the ground. The final inequalities (6) are restrictions on the lower and upper bounds on the speed after collision, preventing both falling over backwards and flight.

D. Disturbance Model

In this paper we claim that the controller is robust to errors in modeling (7), actuation (8)(9), and sensing (10). Each type of error is modeled as a perturbation:

$$\ell := \ell + \delta_\ell, \qquad |\delta_\ell| \leq \Delta_\ell \tag{7}$$

$$p := p + \delta_p, \qquad |\delta_p| \leq \Delta_p \tag{8}$$

$$\phi := \phi + \delta_\phi, \qquad |\delta_\phi| \leq \Delta_\phi \tag{9}$$

$$\omega_k := \omega_k + \delta_\omega, \qquad |\delta_\omega| \leq \Delta_\omega \tag{10}$$

We represent the vector of disturbances:

$$\boldsymbol{\delta} = [\delta_\ell, \delta_p, \delta_\phi, \delta_\omega], \quad \boldsymbol{\delta} \in \mathcal{D} \tag{11}$$

The set of disturbances $\mathcal{D}$ may be thought of as a four-dimensional hyper-rectangle. We define a set of maximal disturbances $\mathcal{D}_{\max}$ that correspond to the 2^4 corners[1]of this hyper-rectangle.

$$\mathcal{D}_{\max} = \{[\pm\Delta_\ell, \pm\Delta_p, \pm\Delta_\phi, \pm\Delta_\omega]\} \tag{12}$$

We use the full set of disturbances ($\mathcal{D}$) for doing controller verification and testing, and the smaller sub-set of maximal disturbances ($\mathcal{D}_{\max}$) for controller design and optimization.

III. CONTROLLER DESIGN

Our controller is a function that maps the estimated mid-stance[2] speed ($\hat{\omega}_k$) to a desired push-off impulse (p) and step angle (ϕ). When implemented on a real robot, the push-off impulse is related to how much energy the extension of the trailing leg adds to the robot before the next step. The step angle corresponds to a target foot-placement location on the ground.

$$\{p, \phi\} = K(\hat{\omega}_k) \tag{13}$$

The controller K aims to stabilize the walking gait to a user-specified target speed (ω^*). Here, we outline the design process for a single given target speed. The method is then repeated to find controllers for a variety of speeds (see results section).

We discretize the range of possible input mid-stance speeds $\Omega_I = [0, \omega_{\max}]$. The output of the controller at each of these grid-points is computed by solving an optimization problem: p and ϕ are the controls that stabilize the mid-stance speed ($\omega_\infty \to \omega^*$) in the fewest number of steps while preventing falls despite all possible bounded disturbances. At run-time the controller is evaluated via linear interpolation over this grid.

We express the design requirements (stability and robustness) as constraints in the optimization problem that defines the controller. Thus, if the optimization returns a feasible solution, then the controller will satisfy the design requirements precisely for every grid point in the controller.

[1]For example, $[+\Delta_\ell, +\Delta_p, -\Delta_\phi, +\Delta_\omega] \in \mathcal{D}_{\max}$

[2]Mid-stance is the point during a step when the stance leg is vertical.

A. Asymptotic Stability

We want the controller to bring the robot towards the desired walking speed (ω^*), starting from any mid-stance speed ω_k drawn from the set of allowed initial speeds Ω_I. This goal can be expressed as the need for reduction at each step k of a discrete Lyapunov function (V), with V defined as the mid-stance speed-error squared:

$$V(\omega_k) = (\omega^* - \omega_k)^2 \tag{14}$$

The controller is asymptotically stable if the Lyapunov function decreases at each successive step:

$$V(\omega_{k+1}) < V(\omega_k) \quad \forall \omega_k \in \Omega_I \tag{15}$$

This condition (15) is imposed as a constraint in the controller design, thus any controller will be asymptotically stable, *in the absence of disturbances*.

B. Robust Stability

Asymptotic stability in the absence of disturbances is good, but we would also like to show that the controller is still stable given any disturbance $\boldsymbol{\delta} \in \mathcal{D}$. In this case, it is not possible to show convergence to a point, but we can (and do) show convergence to some finite set Ω_G that contains the target mid-stance speed.

For this aspect of the controller, we separate the controller design and verification. For the controller design we find the controls that minimize the Lyapunov function (14) over the finite the set of maximal disturbances $\mathcal{D}_{\max}$. The set $\mathcal{D}_{\max}$ is a proxy for the disturbance that precisely maximizes the Lyapunov function. Note that ω_i is the next mid-stance speed, subject to disturbance $\boldsymbol{\delta}_i$. The controls (p, ϕ) and initial state (ω_k) are held constant over each step in the sum:

$$f(p, \phi) = \sum_{\boldsymbol{\delta}_i \in \mathcal{D}_{\max}} (\omega^* - \omega_i)^2 \tag{16}$$

Once the controller has been designed, we do stability verification by running an additional optimization to find the precise disturbance that maximizes the Lyapunov function at the next step $\forall \omega_k \in \Omega_I$. The result of this optimization is the size of the goal set Ω_G that satisfies (17) and (18).

$$V(\omega_{k+1}) < V(\omega_k) \qquad \forall \omega \in \Omega_I - \Omega_g \quad \forall \boldsymbol{\delta} \in \mathcal{D} \tag{17}$$

$$V(\omega_{k+1}) \leq V(\omega_k) \qquad \forall \omega_{k+1} \in \Omega_G \quad \forall \omega_k \in \delta\Omega_G \quad \forall \boldsymbol{\delta} \in \mathcal{D} \tag{18}$$

The first of these equations (17) shows that the controller is asymptotically stable to Ω_G, even in the presence of disturbances. The second equation (18) shows that once the walking speed is inside the boundary of the goal set ($\delta\Omega_G$), that there is no disturbance that can push it out.

C. Fall Prevention

In addition to reaching the target walking speed, we would like that any execution of the controller avoid falling. This is accomplished by adding the constraints (1)–(3) to the optimization problem, and requiring that they hold for $\forall \boldsymbol{\delta} \in \mathcal{D}_{\max}$.

D. Implementation

The constraints for the optimization problem require solving total of 17 simulated walking steps: one nominal step (without perturbation), and 2^4 perturbed steps (one for each disturbance in $\mathcal{D}_{\max}$). All of these simulated steps start from the same initial state and use the same control.

For each of these simulated walking steps, the intermediate speeds ω^-, ω^+ and next step speed ω_{k+1} are passed as decision variables in the optimization. This allows the dynamics (1)–(3) and no-falling constraints (4)–(6) to be expressed as simple non-linear functions of the decision variables. The asymptotic stability condition (15) and objective function (16) are both quadratic in the decision variables. Posing each optimization problem in this way makes it easy to compute gradients and solve quickly using standard non-linear constrained optimization packages.

IV. RESULTS

In this section we present an optimal controller, as designed using the framework presented in this paper.

A. Design Parameters

All parameters in this paper (Table I) have values that roughly match the Cornell Ranger [17].

B. Optimization

We used Matlab's [18] FMINCON optimization software to solve all optimization problems presented here.

The optimization problem that defines the controller is evaluated for each point on a grid. The results here were computed using a grid of 50 points. Similar results are obtained for grids with fewer points, although there is a slight degradation in performance due to interpolation errors at runtime.

The results presented here took approximately 62 seconds[3] to generate in Matlab. Three controllers were generated (including verification), with 50 grid-points each, giving a average time of 0.4 seconds per grid-point.

TABLE I
PARAMETERS

Symbol	Name	Value
m	mass	$9.91\ kg$
g	gravity	$9.81\ m/s^2$
ℓ	leg length	$0.96\ m$
$p_{\max}$	max push-off impulse	$12.2\ kgm/s$
$\phi_{\max}$	max stance angle	$30^\circ = 0.52$ rad
$\omega_{\max}$	max mid-stance speed	2.56 rad$/s$
Δ_ℓ	leg length error bound	$\pm\ 0.05\ \ell$
Δ_p	push-off error bound	$\pm\ 0.05\ p_{\max}$
Δ_ϕ	step length error bound	$\pm\ 0.05\ \phi_{\max}$
Δ_ω	mid-stance speed error bound	$\pm\ 0.05\ \omega_{\max}$

C. Optimized Push-Off Controller

The optimized push-off controller is shown in Fig. 3. It has a relatively simple form: big push-off to increase speed, and no push-off to slow down. For fast walking, the push-off is saturated at the maximum value for much of the domain.

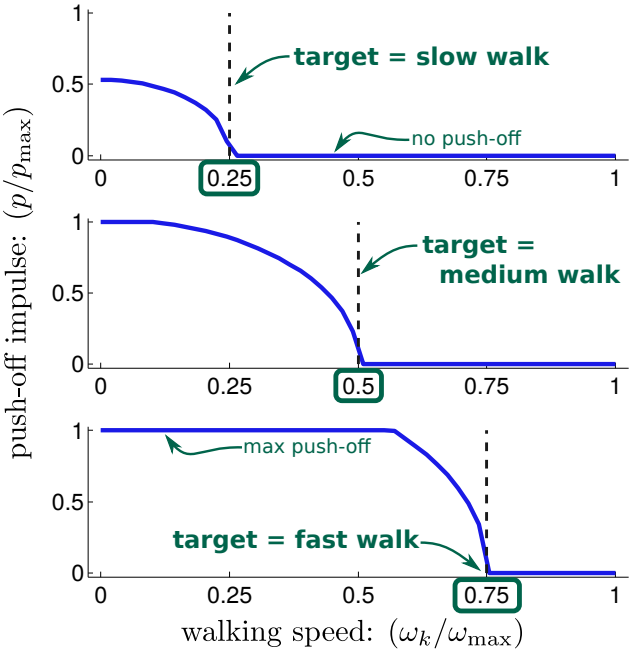

Fig. 3. **Optimized Push-Off Controller.** This figure shows the optimal push-off controller for three different walking speeds. The general trend is simple: too fast $\rightarrow$ no push-off, too slow $\rightarrow$ big push-off. Notice that the actuation is saturated for much of the domain for the *fast* controller.

D. Optimized Step-Length Controller

The optimized step-length controller is shown in Fig. 4. The general trend is that you should take small steps when you are near the target speed, and bigger steps otherwise. In the absence of push-off ($p = 0$), taking bigger steps increases the energy lost due to collision, and will slow the walking gait. If the push-off is non-zero, then it is scaled by a term that increases with the step angle. In effect, taking bigger steps allows the push-off impulse to be more effective. This explains the general trend: if a large push-off is used, then that term dominates the collision losses, and the walking gait speeds up; if a small push-off is used, then the collision losses dominate and the walker slows.

E. Stability and Robustness

Given the optimal controller shown in Figs. 3 and 4, it is possible to compute the closed-loop dynamics of the system, mapping the mid-stance speed from one step to the next. Figure 5 shows this so-called one-step map for an intermediate speed walking gait. The horizontal axis shows

[3]Processor: Intel Core i5-3570K @ 3.40GHz x 4

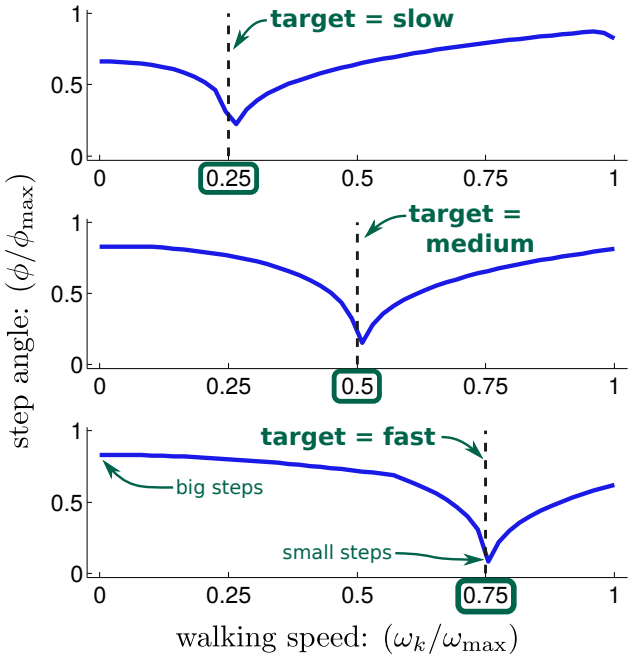

Fig. 4. **Optimized Step-Length Controller.** This figure shows the optimal step-length controller, for three different walking speeds. The general strategy is to use small steps for the nominal walking speed, and take larger steps otherwise. The slight corner in the slow- and medium-speed walking controllers for high speeds is cause by the no flight constraint (4).

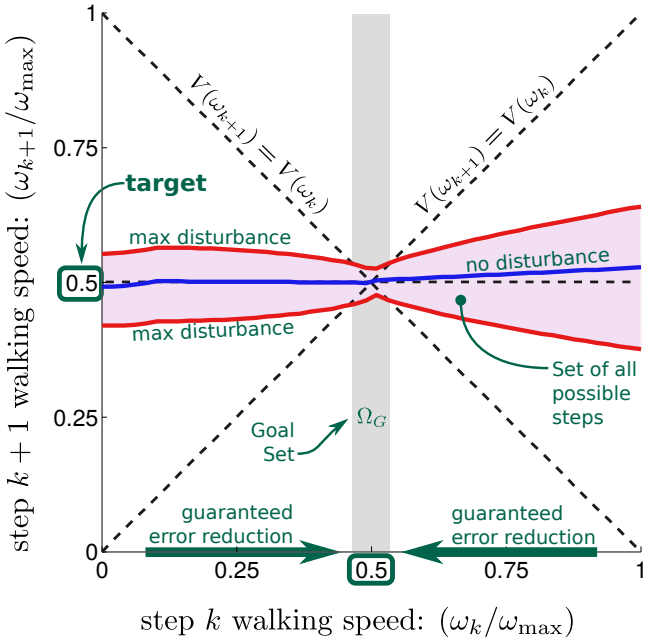

Fig. 5. **One-step speed map.** This figure shows the closed-loop dynamics for the medium-speed ($\omega^* = 0.5\,\omega_{\max}$) walking controller. The horizontal axis gives the mid-stance speed for step k, and the vertical axis gives the mid-stance speed for step $k+1$ that is achieved by the robust controller. The horizontal shaded region show all possible steps that occur. The *no disturbance* line shows the behavior of the controller in the absence of disturbances. The boundary of the shaded region, marked with *max disturbance*, shows maximum possible speed error at the next step due to a disturbance. The dashed lines show the points where the speed error is unchanged from one step to the next. Notice that for most of the domain of the controller, there is a large reduction in error from one step to the next, despite disturbances. For example, if $\omega_k = 0.75\,\omega_{\max}$, then $\omega_{k+1} \in [0.4\,\omega_{\max}, 0.6\,\omega_{\max}]$. If there were no disturbances, then ω_{k+1} would be $0.51\,\omega_{\max}$. The vertical shaded region shows the goal set Ω_G that satisfies (17) and (18).

the mid-stance speed at step k, and the vertical axis shows the mid-stance speed the next step $(k+1)$. Any given point on the horizontal axis maps to a set of points on the vertical axis, corresponding to the set of reachable speeds given any allowed disturbance.

The purpose of this figure is to visualize the stability of the controller. The diagonal dashed lines show the points where the Lyapunov function (14) is unchanged from one step to the next. The entire horizontal axis is the set of initial speeds Ω_I, and there is a thin vertical shaded region that shows the goal set Ω_G. For initial speeds that are outside of the goal set, the Lyapunov function is always decreasing from one step to the next. For initial speeds inside the goal set, the Lyapunov function (error squared) might increase, but the next step speed will never leave the goal set.

In the absence of disturbances the controller rejects nearly all speed error in a single step, as shown by the nearly horizontal line labeled *no disturbance*. In the presence of disturbances the controller is still quite stable, reaching the goal set in two or three steps even with the worst possible disturbances.

F. Simulation Test

As one final check, we simulated the closed loop system for a total of 10^6 steps, where the robot was subject to random disturbances, uniformly drawn from $\mathcal{D}$[4]. We ran 10^3 simulations, each consisting of 10^3 steps and starting from a randomly chosen initial condition uniformly drawn from Ω_I. In all cases the mid-stance speed stabilized to the target speed within a few steps, and the mid-stance speed remained within the goal set (Ω_G) on all subsequent steps. That is, despite the disturbances applied at each step, the controller was able to prevent falls in all cases.

V. Discussion

We have presented a controller for a simple model of walking that is maximally robust, by our measures. This controller can regulate a desired walking speed while preventing falls due to reasonable errors in the model, sensors, and actuation. The controller avoids falls for all reasonable values of disturbed and desired walking speeds. That is, assuming we consider only forwards walking, the basin of attraction of this controller is maximal with the given noise.

The approach here was inspired by robust control: the controller should be robust (never fail) for any bounded disturbance, and the walking speed should converge to some goal set. The controller was designed using non-linear

[4]The perturbation on leg length was randomly drawn once at the beginning of each of the 10^3 simulations, and then held constant for all 10^3 steps in each simulation. This was done to more closely mimic a robot that had a modeling error. The other disturbances were randomly drawn on every single step.

optimization, where these robustness and stability requirements were enforced as constraints on the optimization. Although the resulting controller is correct by construction, we demonstrated the performance of the controller using massive simulation.

We designed a controller that could robustly stabilize slow, moderate, and fast walking gaits for the inverted pendulum model of walking, using parameters based on our robot, the Cornell Ranger. An interesting and perhaps general result is this: To increase speed, the controller takes large steps and uses large push-off. To maintain speed, it takes small steps and uses intermediate values of push-off. To decrease speed, the controller takes large steps with no push-off. This general trend is observed across all walking speeds. Although only implemented here on a simple 2D model, we believe the approach will be useful for a more complex robot in 3D.

VI. FUTURE WORK

We plan to develop this controller for use on a 2D robot, Cornell Ranger, and then later, for a 3-D robot with many degrees of freedom.

One limitation of our model is that we restricted it to forward motion without a flight phase. These restrictions make calculations easier, but are overly conservative with respect to the real limits on falling; a small flight phase, or a step backwards does not necessarily lead to a fall. With a more general model that allowed some flight and backstepping, we could make a robust controller with even larger allowed bounds on the various disturbances.

The extension from a 2-D robot to a 3-D robot will raise the mid stance state from 1 number to 2 (or 3 if heading is to be stabilized). The actuation will go from 2 controls to 3 (push off, step length and steering angle). However, the addition of more internal degrees of freedom does not change the form of this high-level controller. Rather the output of a controller of the type developed here, will serve as input to the micro-management of the various joints so as to achieve the desired push off and foot placement.

ACKNOWLEDGMENT

This research is supported by the National Science Foundation: Graduate Research Fellowship Program (fellow ID: 2011116308) and the National Robotics Initiative (grant number: 1317981).

Thanks to Petr Zaytzev for his insights about the nature of the control laws for the simplest walker, and to Anoop Grewal for his work on the simplest walker model.

REFERENCES

[1] A. Ruina, "Nonholonomic stability aspects of piecewise holonomic systems," *Reports on Mathematical Physics*, vol. 42, no. 1-2, pp. 91–100, 1998.

[2] S. Kuindersma, F. Permenter, and R. Tedrake, "An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion," in *International Conference on Robotics and Automation*, 2014.

[3] P. R. D. Honda, "Asimo Technical Report, Tech. Rep. September, 2007.

[4] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A Compliant Hybrid Zero Dynamics Controller for Stable, Efficient and Fast Bipedal Walking on MABEL," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, Sept. 2010. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364910379882

[5] P. a. Bhounsule, J. Cortell, a. Grewal, B. Hendriksen, J. G. D. Karssen, C. Paul, and a. Ruina, "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1305–1321, June 2014. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364914527485

[6] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 200–207, Dec. 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4115602

[7] T. Koolen, T. de Boer, J. Rebula, a. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, July 2012. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364912452673

[8] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, Aug. 2012. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364912452762

[9] J. Englsberger, C. Ott, M. A. Roa, A. Albu-sch, and G. Hirzinger, "Bipedal walking control based on Capture Point dynamics," in *International Conference on Intelligent Robots and Systems*, San Francisco, 2011, pp. 4420–4427.

[10] A. Ruina and A. Kuo, "Some things that we think we know about human and robotic walking," in *Dynamic Walking Conference*, 2014.

[11] S. Kajita and K. Tan, "Study of Dynamic Biped Locomotion on Rugged Terrain - Derivation and Application of the Linear Inverted Pendulum Mode," in *International Conference on Robotics and Automation*, no. April, 1991, pp. 1405–1411.

[12] M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running." *Nature*, vol. 439, no. 7072, pp. 72–5, Jan. 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16155564

[13] ——, "Idealized walking and running gaits minimize work," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2086, pp. 2429–2446, Oct. 2007. [Online]. Available: http://rspa.royalsocietypublishing.org/content/463/2086/2429.short

[14] M. Garcia, a. Chatterjee, a. Ruina, and M. Coleman, "The simplest walking model: stability, complexity, and scaling." *Journal of biomechanical engineering*, vol. 120, no. 2, pp. 281–8, Apr. 1998. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10412391

[15] A. D. Kuo, "Energetics of Actively Powered Locomotion Using the Simplest Walking Model," *Journal of Biomechanical Engineering*, vol. 124, no. 1, p. 113, 2002.

[16] P. Zaytzev, "The viability and controllability approach to robustness in bipedal locomotion: application to simple models," Ph.D. dissertation, Cornell University, 2015.

[17] P. a. Bhounsule, J. Cortell, A. Grewal, B. Hendriksen, J. G. D. Karssen, C. Paul, and A. Ruina, "MULTIMEDIA EXTENSION # 1 International Journal of Robotics Research Low-bandwidth reflex-based control for lower power walking : 65 km on a single battery charge," *International Journal of Robotics Research*, 2014.

[18] MATLAB, *version 8.3.0 (r2014a)*. Natick, Massachusetts: The MathWorks Inc., 2014. [Online]. Available: http://www.mathworks.com/