PRE-PRINT

Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge

In revision for the International Journal of Robotics Research

Pranav A. Bhounsule[1,4], Jason Cortell[4], Anoop Grewal[4], Bram Hendriksen[2],

J.G. Daniël Karssen[2], Chandana Paul[3,4] and Andy Ruina[4]

[1]Disney Research, Pittsburgh   [2]Delft University of Technology

[3]Engineering and Applied Sciences, Harvard University

[4]Mechanical Engineering, Cornell University, ruina@cornell.edu

—Most recent modification on October 7, 2013—

Abstract

No legged walking robot yet approaches the high reliability and the low power usage of a walking person, even on flat ground. Here we describe a simple robot which makes a small progress towards that goal. Ranger is a knee-less 4-legged 'bipedal' robot which is energetically and computationally autonomous, except for radio controlled steering. Ranger walked 65.2 km in 186,076 steps in about 31 hours without being touched by a human with a total cost of transport [TCOT $\equiv P/mgv$] of 0.28, similar to human's TCOT of ≈ 0.3. The high reliability and low energy use were achieved by: 1) development of an accurate bench-test-based simulation; 2) development of an intuitively tuned nominal trajectory based on simple locomotion models; and 3) offline design of a simple reflex-based (that is, event-driven discrete feed-forward) stabilizing controller. Further, once we replaced the intuitively tuned nominal trajectory with a trajectory found from a numerical optimization, but still using event-based control, we could further reduce the TCOT to 0.19. At TCOT = 0.19, the robot's total power of 11.5 W is used by sensors, processors and communications (45%), motor dissipation ($\approx$34%) and positive mechanical work ($\approx$21%). Ranger's reliability and low energy use suggests that simplified implementation of offline trajectory optimization, stabilized by a low-bandwidth reflex-based controller, might lead to energy-effective reliable walking of more complex robots.

1   Introduction

Some walking robots can, in a given setting, walk reliably, without falling e.g., [13, 44]. Some robots have been relatively energy stingy, e.g., [9]. But none has had both the high reliability

1

and the low power usage (scaled for weight and speed) of a person walking on level ground. Here we describe a simple robot, Ranger, that attempts reliable walking on level ground with little energy use [43]. Ranger is essentially planar (rather than 3D); it has only 3 visible kinematic degrees of freedom (rather than the 10–30 DOFs of a more realistic humanoid), and all it can do is walk steadily and basically straight on level ground (as opposed to having a variety of gaits on rough terrain). But, before Ranger, no robot has been both efficient and reliable, even in this limited setting. Although admittedly limited in function, Ranger has walked farther, while using less energy per unit distance and weight, than any other legged robot. Here we explain Ranger's design and control with the hope that some of the ideas can be used in more versatile machines. We first briefly discuss some previous legged robots, their energy use, and some possible approaches to control.

1.1 Measuring energy effectiveness.

Energy effectiveness is measured by how little power is used. To compare different robots we use a measure that takes simple account of weight and speed.

$$\text{Cost of Transport (COT)} = \frac{\text{power consumption}}{\text{weight} \times \text{speed}} = \frac{P}{wv}.$$

COT (if based on weight $w = mg$ and not mass m) is dimensionless ($\text{W}/(\text{N m/s}) = 1$). The smaller the COT the more energy-effective. There are different COTs depending on what powers are included in P (e.g., 'total', 'locomotion', and 'mechanical' costs).

COT for humans. For a human walking, the *total* cost of transport, accounting for the full food energy used by a person as they walk, is about TCOT≈0.3, e.g., [1, 4, 15]. An often-reported *locomotion* cost of transport for people of 0.2 is based on subtracting the energy a person uses to stand still. For humans an estimated *mechanical* cost of transport (MCOT), based on the total positive work done by the muscles or actuators (and not subtracting out the negative work) is about MCOT $\approx 0.05 = 0.2 \times 25\%$ because muscles are about 25% efficient (while doing only positive work $\approx 75\%$ of the chemical energy used by muscles goes to heat[30]).

1.2 Passive dynamics: energy use and control

One approach to energy-effective control is based on passive dynamics, e.g., [31, 11, 24, 34]. A strictly passive-dynamic robot is a linkage with no sensors and no motors that can walk stably down a slight slope. As such a 'ramp walker' goes down a slope γ, gravitational potential energy is used up by friction and collisions. The first serious passive-dynamic robot was McGeer's "4-legged biped" with 4 side-by-side legs, each with a knee, and no upper body.

It is modeled as a two-legged machine living in 2 spatial dimensions [31]. Despite the non-anthropomorphic 4-legged layout, the gait of McGeer's 'biped' is inspirationally evocative of human walking. The simplicity, the human-like motion, and the low energy use of this and other passive-dynamic walkers, e.g., [11], have led to attempts at realizing near-to-passive dynamics on *level* ground by replacing gravity power with motor power.

Passive-dynamic energetics.   For a passive-dynamic robot the TCOT $= \sin\gamma \approx \gamma$, e.g., [18]; a machine that walks down a slope of $\gamma = 0.05$ ($\approx 3°$) has a TCOT $= \sin\gamma \approx 0.05$. For these robots the mechanical cost of transport (MCOT), the 'actuator' work per unit weight and distance, is the same as the TCOT because all of the gravitational energy is supplied as mechanical work. Typical passive-dynamic ramp walkers happen to use about the same amount of gravitational work as is performed by the muscles of a human walking on level ground (MCOT ≈ 0.05).

Powered passive-based robots.   The Cornell "Collins" powered biped [9, 10], with a passive hip, energetically passive knees (but for a controlled catch), and powered only at the ankles had a TCOT of 0.2, apparently lower than that of any motor-driven legged robot before or since, with the exception of Ranger described here. It is, however, a misconception to think that passive-based means maximally energy effective. In principle, a robot propelled using pre-emptive push off might use as little as 1/4 the power (75% less power) as one powered by gravity [27, 42, 49]. Even though passive-based robots are still far from the theoretic limits of energy stinginess, thus far passive-based robots have been less power consumptive than other powered walking robots.

Stability of passive-dynamic robots.   Despite the lack of control, passive walkers have some mechanisms that can contribute to stability: **1)** dissipation (e.g. the rimless wheel [6] dissipates more energy when it goes faster); **2)** the non-holonomic natured intermittent foot contact allows self-regulation by the natural dynamics of foot placement [41]; and **3)** the static stability of the splayed standing configuration that is intermittently visited in the walking cycle (as discussed in [8], that is, walking may have some aspect of the dynamics Shannon 'catch-and-throw' juggler [46]). That passive-dynamic robots can be stable has allowed the making of inspiring physical demonstrations.

One objective (independent of coordinate-system) measure of stability is the magnitude of the biggest eigenvalue of the Jacobian of the step-to-step map (McGeer's stride function, a Poincare map) at a periodic walking cycle [31, 50]. If the biggest (possibly complex) eigenvalue has magnitude less than one, then, after a small disturbance, in subsequent steps the motion decays exponentially back to the periodic motion. By this measure, if all eigenvalues are much less than one in magnitude (and thus close to the origin on the complex plane), then the decay is rapid and the robot is very stable.

3

Typical passive-dynamic walkers are only mildly stable at best by this measure, with their biggest eigenvalues rarely less than about 0.6 in magnitude. And passive-dynamic robots are definitely not intrinsically stable; unstable passive-dynamic designs are easy to generate, having, for example, the largest simulated eigenvalue much bigger than one, depending on the particulars of the mechanical design and the slope it is on [7, 19]. In the lab, the behavior of even the best passive-dynamic robots has been fussy. And, in our experience, a powered walker whose stability is based on passive dynamics, is inherits the fussiness of it's passive parent.

The problems with pursuing passive-dynamics as a basis for stability   There is no qualitative analytic theory of passive-dynamic stability; the eigenvalues of the characteristic equations are dependent on the physical parameters in too-complex a manner. And we know of no analytic recipes, or even promising numerical optimization approaches, for appreciably enhancing passive stability while preserving the visual appearance of gait. There is little evidence that passive strategies can have anywhere near the reliability needed for practical robotics or for explaining the balance stability of humans or other upright animals. That passive-dynamic robots can be self-stable at all has perhaps mis-inspired some of us into pursuing passive strategies for stabilizing motorized robots.

1.3   Feedback control

Most often robot balance is based on feedback control, with little concern for the passive orbital stability of the limit cycles of the uncontrolled mechanical system. Recovery from unbalance instead comes from calculated ankle torques reacting against flat feet (applying torques to move the robot's center of mass over the feet), calculated foot placement (essentially stepping in the direction of an undesired fall), or both. As evidenced by the difficulty humans have with balance on a taut wire, balance by means of moving upper body parts, unless based on a reaction wheel or a long pole, does not seem to be a needed central part of the control of robots that have the option recovering from disturbances by means of appropriate foot placement.

ZMP.   One class of controllers focuses on the position of the Zero Moment Point (ZMP), the point on the ground where the reaction force and couple have no horizontal moment component. In 2D, this is the point where the net reaction is a pure force with no couple (the so-called center of pressure, COP). In initial concept, ZMP controllers focus their attention on choosing ankle torques to keep the ZMP inside the foot contact polygon, thus keeping the foot flat on the ground [54, 55].

With good foot placement, the reliance on ankle torques for balance can be minimized [51]. Robots that use ZMP control for walking, most famously Honda's ASIMO [44] series,

tend to walk with bent knees that allow the controllers to have authority over all the upper body degrees of freedom and usually have flat-bottomed feet. The TCOT of ASIMO in 2005 was estimated (from battery capacity, speed, weight and time) to be about 3.2 [9], about 10 times that of a human. It is not clear whether the relatively high energy use of ZMP-based robots thus far is from; i) the control strategy, which is not specialized just for steady walking, or, ii) from correlative features, such as highly gear-reduced motors, or, iii) merely from lack of attention to power minimization. There seems to be no fundamental reason ZMP robots cannot substantially reduce their power use.

Balance by foot placement. Some more-dynamic feedback-controlled robots have had balance control based almost entirely on foot placement, with little or no use of reaction torques acting on flat feet. The best known of these are from Marc Raibert's MIT lab and his company Boston Dynamics. The first versions of these robots were 2D single-leg hopping robots with control based on the observation that hop height, forward speed and body orientation could all be controlled by control of leg angle and leg length at appropriate flight or contact times [40]. These ideas were extended to 3D and multiple legs, e.g., [39]. Recently, balance based on foot placement has been used to make what seems to be a highly-reliable true 3D biped walker [13, 14]. Using the estimates that PETMAN and ATLAS weigh about 1000 N, move at about 2 m/s and consume about 10,000W (about 13.5 hp) of hydraulic pump power, they have a TCOT of about 5 (about 15 times that of a walking person). No-one has posited a fundamental reason that the energy use of such robots could not be substantially reduced.

Capture point and region. One approach to control using foot placement is to place the foot so that the robot's angles and rates are in, say, the '1-step capture region' of the robot's phase space. The 'one step capture region' is the set of states from which there is a known strategy by which the robot can come to a complete stop within 1 step [25, 38]. In this approach speed can be increased, say, by stepping slightly shy of the best point to stop the motion, thus inducing a larger next step [16]. The capture region can be enlarged by using ankle torques. Capture region techniques are model based, but to date have been based on point-mass models (e.g., the linear inverted pendulum) so as to reduce computational complexity. The capture region method is reported to be used on PETMAN and ATLAS.

Continuous stabilization of trajectories. A simulation-based approach is to first find a desirable trajectory, such as a limit cycle with low energy use, and then to create a controller that stabilizes that cycle. First one might imagine a locally linear controller. But a locally linear model does not provide a means to measure or expand the basins of attraction (the controllable regions). A recent approach is to find secure basins of attraction for each of a number of paths back to a desired trajectory. If these regions overlap, one can be assured

that a reasonable region in the phase space is controllable. This is being pursued with LQR (Linear Quadratic Regulator) trees [52]. Another approach is to use on-the-fly trajectory optimization and, between optimization cycles, to use a linear controller with known nonlinear properties. This has shown promise on a 2D robot [29]. The control we describe below also uses stabilization of trajectories on a pre-calculated limit cycle, but not continuous stabilization.

HZD. In Hybrid Zero Dynamics (HZD) [22] one angle, say the shank fore-aft angle, is regarded as effectively uncontrolled (there is no reaction torque at the ground). The first idea in HZD is to parameterize control using that angle, assumed monotonic in time, rather than time. The original HZD idea was to effectively eliminate internal degrees of freedom by tightly controlling them; they were all slaved to the motion of, say, the uncontrolled shank angle, according to functions (the slave joints' angles are functions of the free shank angle) whose specification *is* the control. These functions could be chosen to minimize this or that cost function (say, energy use, [5, 56]). In this HZD approach, the trajectory of the robot's internal configuration space is fixed, and the uncontrolled degrees of freedom evolve according the the laws of mechanics. This strict dimension-reduction version of HZD has some issues: **1)** it seems to depend on high bandwidth control of the slaved degrees of freedom. In our experience, such high-bandwidth, high-gain control tend to be energy consumptive, even when the pre-calculated mechanical work of the trajectory is small; **2)** The strict internal degrees of freedom trajectory tracking HZD depends on having a machine that is, after the HZD joint-position control is implemented, not compliant and thus perhaps not appropriately yielding to physical disturbances; and **3)** Such a stiff HZD machine is not, perhaps, satisfyingly biomimetic in that it imposes tight control of possibly unimportant degrees of freedom, thus violating biologically-relevant ideas associated with, say, the uncontrolled manifold or with optimal control [28, 53].

However the tight-trajectory tracking idea has been relaxed in the recent HZD implementations, and some internal degrees of freedom have been passive [47]. And in 3D there can be three or more external uncontrolled degrees of freedom, not just the single shank angle. Further relaxing the initial concept, the slaved trajectories need not be controlled with tight feedback, but can be pre-computed and generated with feed-forward torque profiles that are stabilized with low gain proportional control [35]. Finally, additional event-based feedback can be used to control the un-actuated degrees of freedom over the time scale of a whole step.

As modified to allow for compliance, more uncontrolled external degrees of freedom, feed-forward torques, and stabilizing feedback of the joint trajectories, and with the addition of intermittent event-based feedback, the HZD approach seems totally viable, as demonstrated by the fast and robust walking of the planar biped MABEL. And there seems to be no reason HZD control need be energy demanding. As thus evolved, there seems to be little
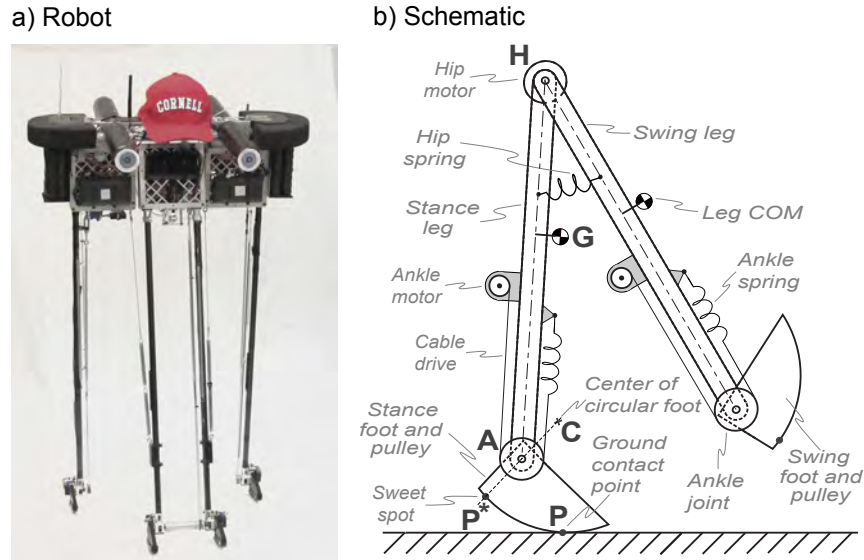
Figure 1: **(a)** Ranger. **(b)** 2D Schematic. The fore-aft 'eye' cylinders and the foam 'ears' are for cushioning in falls. The hat is decorative (hollow). There are two closed and rigid aluminum lace boxes connected by a hinge, conceptually shown as point H. The outer box, shaped like an upside-down U, is rigidly connected to the outer legs. The inner box, filling the space in the U, holds the inner legs (each of which can twist for steering). The boxes house all of the motors and gearing, pulleys for the ankle cable drives, and most of the electronics (on drawing (b) the hip and ankle motor locations are only schematic). The hip spring, which aids leg swing, is shown schematically as symmetric between the two legs (in the photograph the hip spring shows as a diagonal cable and spring from the outer boxes to the inner legs). The feet are shaped for toe-off propulsion, so that no torques are needed during single stance, and for ground clearance during swing (by rotating the toe up to the shank).

difference between an implementation of an HZD approach and a reflex-based controller as, for example, described below.

2   Ranger hardware

The Cornell Ranger is a four-legged knee-less biped (figure 1). It is about 1 m tall and has a total mass, including batteries, of 9.9 kg. It is autonomous in that all sensing and computation is on board, batteries are on board, and it has no booms, tethers or cable connections. It is not autonomous in that, at least as of this writing, it needs to be started manually, and the steering is done with a hobby-type radio control.

Hardware.   The outer pair of legs move together, acting as one leg, as do the inner pair. Each leg has an ankle joint and a foot but no knee joint. The inner feet are connected to

move together by means of a horizontal shaft, the outer feet with a long cable through the 'U' hip box. The robot has 3 main internal degrees of freedom (1 hip + 2 ankle-pairs). These are all powered by brushed DC motors (46 W nominal). In parallel with the hip motor is a hip spring that tends to align the legs. The two ankle motors are near the hip axis and actuate the ankles via one-way (toe-off) cable drives. Foot lifting, for ground clearance, is powered by a return spring on each ankle. A small fourth motor (1 W nominal) twists the inner legs about a vertical axis in order to steer the robot.

Electronics. The main control loop runs, with no supervisory operating system, on an ARM9 microcontroller. There are 4 ARM7 processors on custom boards, one each to monitor and control the inner ankles, the outer ankles, the hip and the steering. Two more ARM7 processors supervise the CAN communications (Controller Area Network), the Bluetooth data reporting, and the onboard data display and lights. The Inertial Measurement Unit (IMU) uses a proprietary microprocessor. The bus-based architecture with 8 processors was chosen to facilitate design evolution, to simplify overall wiring (e.g., so we could add a sensor without adding wires to the main processor) and to compartmentalize the control software (high-level on ARM9, low level on ARM7s). Sensors for each motor include an optical encoder, a voltage sensor, and a current sensor. In addition, each joint has an angle sensor using absolute encoders. Each foot has an optical strain gauge for measuring foot distortion (and hence foot contact). From the 3D IMU, Ranger's control only uses the sagittal plane angular rate sensor. The top-level control loop runs at 500 Hz on the ARM9 processor; data is sent to and from the satellite ARM7 processors once per loop execution; the motor current controllers, and their associated sensors operate at 2 kHz. The motors and electronics are powered by seven 25.9 V lithium-ion batteries with a total capacity of about 493 watt-hours $(1.77 \times 10^6 \text{J})$.

Software. Most of the $\approx$ 10,000 lines of C and C++ code is for low-level measurement, low-level control and communications protocols. Control and estimation tasks are coordinated by a simple cooperative-multitasking scheduler, while low-level input-output, such as from motor encoders, uses processor interrupts. The main control logic and feedback, the primary topic here, is described further (below); it is a small part of the total software. For debugging and development, data is viewed and logged using a wireless Bluetooth system. During attempts at walking distance records autonomy is maintained by sending to the robot only steering commnds and requests for data (e.g., cumulative number of steps, battery voltage) but no walking control parameters.

More design details. As with any complex machine, success rests on a pyramid of details: the design of the foot shape to allow push-off, stance with low torque, and swing clearance; the design of the foot as a load cell (foot deformation is measured); a low-mass leg and

foot; design of the single-cable drive (which is also a series-elastic element which needs to have twice the stiffness for the inner leg-pair as for each of the outer legs); design of the body box for stiffness; selection of motors and gearing for torque and efficiency; design of the motor controllers for low dissipation; selection of energy-efficient sensors and microprocessors; determining overall state from sensor data, especially prediction of the ground collision time; design of a low-power leg-twist steering mechanism; hip-spring design and placement; and dozens of other issues like the selection of glues, the protocol for washing off flux to prevent corrosion, methods for joint alignment and needed drive-train compliance, cable tensioners, shock absorbers for fall protection, etc. Some of these details are described in the online Appendix, extension 1.
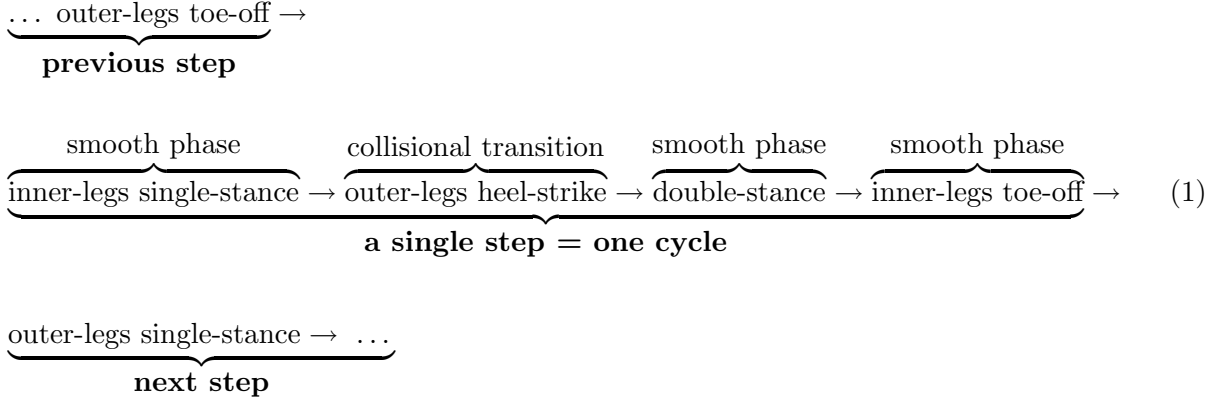
3 Model for simulation and control design

Our reflex-based control design was developed using a fast and accurate offline simulation of the 2D robot dynamics. Here we describe the model used.

Each 'leg' (one inner and one outer) is characterized by a mass, a center of mass (COM) location, and polar inertia about the COM. The rubber-bottomed feet are assumed to be rigid and massless with bottoms that make point contact with a rigid, flat and level ground (unless otherwise noted). The hip motor connects to the legs with gears so that transmission is modeled as not compliant (the hip angle does not have a degree of freedom independent from the hip motor). The hip spring, effectively in parallel with the hip motor, adds a centering torque, proportional to leg splay, to the hip motor torque. The hip motor ('reflected') inertia is neglected because, even after multiplying by the gear reduction squared ($66^2 = 4356$), the inertia of the hip motor is about 50 times less than that of the legs. In contrast, the ankle motors are in series with the elastic cable drive. So the ankle motors, with an associated non-negligible rotary inertia, each have a degree of freedom independent from the ankles. The ankle drive cables are modeled as linear springs, as are the return springs.

Phases of motion. The motion has two smooth phases: **1)** single-stance, when one foot is on the ground, and **2)** double-stance when both feet are on the ground. The two phases are separated by two instantaneous transitions: **a)** the heel-strike collision at the transition from single-stance to double-stance, and **b)** the (non-collisional) toe-off transition from double-

stance to single-stance. A single walking step consists of:

$$\underbrace{\ldots \text{ outer-legs toe-off} \rightarrow}_{\textbf{previous step}}$$

$$\underbrace{\overbrace{\text{inner-legs single-stance}}^{\text{smooth phase}} \rightarrow \overbrace{\text{outer-legs heel-strike}}^{\text{collisional transition}} \rightarrow \overbrace{\text{double-stance}}^{\text{smooth phase}} \rightarrow \overbrace{\text{inner-legs toe-off}}^{\text{smooth phase}} \rightarrow}_{\textbf{a single step = one cycle}} \quad (1)$$

$$\underbrace{\text{outer-legs single-stance} \rightarrow \ldots}_{\textbf{next step}}$$

Ground contact and collisions. When in contact with the ground the feet are assumed to roll without slip. The ground collisions are modeled as instantaneous with no bounce and no slip. The heel-strike collision is assumed to have no impulsive torques at the joints. The robot is taken as symmetric with respect to inner and outer legs, so only one step is needed to characterize a periodic gait. The two ankle motors do not participate in the heel-strike collision because the ankle motors are isolated by the ankle springs (Achilles tendons), so the ankle motor velocities are continuous in time. In contrast the hip motor has a collisional velocity discontinuity but, because its inertia is neglected, no collisional torque. Although we model the ground contact as a point contact between rigid feet and rigid ground, we add a small contact-damping couple between foot and ground, effectively a viscous rolling resistance, to damp out oscillatory rocking motions of the feet (observed in simulations if this damping is zero).

Degrees of freedom. Neglecting steering, the robot has five modeled internal degrees of freedom (one hip joint including the motor, two ankle joints, and two ankle motors). During single stance, there is one additional pose degree of freedom (rotation of the stance foot) making a total of six. During double stance there are only the five degrees of freedom. The accessible configuration space is 7 dimensional (position of contact foot, stance-foot angle, hip joint angle, two ankle joint angles, and two ankle motor angles). The robot is thus under actuated ($3 < 7$), even if neglecting position on the walking line ($3 < 6$), and even in double stance ($3 < 5$).

At any instant in time the robot is a 5 or 6 degree of freedom holonomic system. Due to intermittent contact the robot may be viewed as non-holonomic [41], but because we are not concerned with reaching absolute foot-strike targets, this non-holonomic aspect does not affect our control architecture.

Controllability. A linear model of the robot is locally controllable in a control-theory sense; given adequate motor torques any configuration could be achieved in finite time. But we have not investigated such controllability in detail because bounds on motor torque and joint angles make the robot, practically speaking, *not* controllable. For example, even with infinite motor torque, the robot could not balance vertically on two feet if disturbed more than about 3°; more than this would require legs to swing beyond joint limits. Thus, no attempt is made to continuously control all of the 5 or 6 degrees of freedom of the instantaneously holonomic system.

However, viewed as a discrete map, with consideration for only essential degrees of freedom, the system *is* controllable. That is, the robot actuation is sufficient to independently modify, from one mid-stance to the next, the stance leg angular rate, the swing-leg angle and the swing leg angular rate. The essential control design problem is thus to make use of this discrete controllability, given limited actuator torque, limited sensing and limited processing ability, and while minimizing energy use and complexity.

Governing equations. Using the assumptions described above, the governing equations follow from momentum and angular momentum balance applied to the robot and its subsystems:

$$\text{Single stance (continuous):} \quad \mathbf{M}_s(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_s(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{K}_s(\mathbf{q}) \qquad = \mathbf{T} \qquad (2)$$

Heel-strike (instantaneous):   Angles continuous but velocities jump

$$\mathbf{M}_c(\mathbf{q}^+)\dot{\mathbf{q}}^+ - \mathbf{M}_c(\mathbf{q}^-)\dot{\mathbf{q}}^- \qquad = \mathbf{J}_{\mathbf{c}}^{\mathbf{T}}\mathbf{P}^{\star} \qquad (3)$$

$$\text{Double stance (continuous):} \quad \mathbf{M}_d(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_d(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{K}_d(\mathbf{q}) \qquad = \mathbf{T} + \mathbf{J}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P} \qquad (4)$$

Toe-off (instantaneous):   Angles and velocities continuous   (5)

where each equation is a 6-component vector equation. The six elements of $\mathbf{q}$ are the three robot joint angles, the two ankle motor angles and the absolute angle of one foot. The six elements of $\mathbf{T}$ are the three motor torques (including the gear box friction), and three foot-related torques: two ankle torques and one ground-contact damping term at the reference-foot ground contact (see Appendix, extension 1 figure 16 and 17). The foot torques are determined by joint friction and by the ground-contact damping term. The $\mathbf{M}$s are the mass matrices (subscripts s, d and c are for the 'single', 'double' and 'collisional' phases), the $\mathbf{C}$s are the velocity squared terms (centrifugal and Coriolis) and the six components of the $\mathbf{K}$s include the gravity and spring related terms. The 2-element $\mathbf{P}$ and $\mathbf{P}^{\star}$ are trailing foot constraint force components during double stance and constraint impulse components during heel-strike respectively (in some formulations $\mathbf{P}$ and $\mathbf{P}^{\star}$ are Lagrange multipliers for constraints). Finally, the $6{\times}2$ matrices $\mathbf{J}_{\mathbf{c}}$ and $\mathbf{J}_{\mathbf{d}}$ are kinematically determined matrices

that show the effect of trailing-foot constraint force components on the internal degrees of freedom when the kinematic chain is closed (double stance). Appendix, extension 1 discusses the equations of motion in more detail.

3.1 Motor and gearbox model.

We bench-tested out the motor and gear box in positive and negative work regimes, measuring voltage, current, angular velocity and torque. There are two parts to the resulting motor and gearbox model. A power equation and a torque equation. We summarize these next.

The power equation gives the power consumption (P) as a function of the current (I) and the speed (ω).

$$P = \{IR + V_c \, \mathrm{sgn}(I) + GK\omega\} \, I \tag{6}$$

where the signum function is defined as $\mathrm{sgn}(I) \equiv I/|I|$. The gear ratio is G, the motor resistance is R, motor torque constant is K and a contact resistant modeled as a contact voltage drop V_c. This power equation includes two non-standard terms: **1)** we found the non-standard contact voltage drop V_c [23], which changes signs when the current is reversed. This term has the biggest effect when the total motor voltage is low; **2)** the winding resistance, R, was 1.3 Ω and nearly a factor of two higher than that reported in the specification sheet.

The torque equation gives the output shaft torque (T) as a function of the current and shaft speed.

$$T = GKI - G^2 J_m \dot{\omega} - T_f(I, \omega) \tag{7}$$

where the motor inertia is J_m and T_f is the friction torque and is given next.

$$
\begin{aligned}
|T_f(I, \omega)| \quad &\leq \quad C_0 + \mu GK|I| &&\text{if } \omega = 0 \\
T_f(I, \omega) \quad &= \quad C_1\omega + C_0 \, \mathrm{sgn}(\omega) + \mu \, \mathrm{sgn}(\omega)GK|I| &&\text{if } \omega \neq 0
\end{aligned} \tag{8}
$$

The viscous friction is given by $C_1\omega$ and the coulomb friction by $C_0 \, \mathrm{sgn}(\omega)$. A friction that scales approximately with torque, apparently due to load-dependent friction in the gears, is given by $\mu \, \mathrm{sgn}(\omega)GK|I|$ (Note that $|I|$ correlates with the magnitude of net torque and $\mathrm{sgn}(\omega)$ makes this a resistance-to-motion term).

Our current-dependent friction term is similar in effect to the load dependent friction term [12]. To make the solutions unique in numerical implementation at $\omega = 0$ and to ensure smoothness in the solution for better convergence of numeric optimizations, we replace the

sgn (sign) function with a hyperbolic tangent function. The values we used in simulation were: $J_m = 1.6 \times 10^{-6}\ kg - m^2$, $K = 0.018\,\mathrm{N\,m/A}$, $G = 66$ (hip), $G = 34$ (ankle, gear reduction is 43, taking account of pulley radii the ratio is 34), $C_1 = 0\ \mathrm{N\,m\,s/rad}$, $C_0 = 0.01$ N m and $\mu = 0.1$. In this $C_1 = 0$ approximation the torque is mis-estimated by 2% or less. See the Appendix, extension 1 for more motor-modeling details.

Later in this paper we describe a further motor model feature (frequency dependent inductance) that was discovered by comparing the robot simulations with robot data.

4   Overall control approach: sparse state machine

Our overall control design approach is similar to that of Miura and Shimoyama [32] who seem to have just followed Formalsky [17] in discussing walking as a Poincaré map. Their controller was made up of two parts: an open loop time-based trajectory planner and a feedback controller to stabilize the nominal trajectory. Their stabilizing linear feedback controller used the measurements at the beginning of the step to drive the robot state at the end of the step to its nominal value. That is, instead of tracking a trajectory in the gait cycle, their controller, active throughout a step, only tried to regulate the state at the end of the step. The gains for the linear controller are calculated by doing a step-to-step eigenvalue calculation. Where the HZD and ZMP approaches constrain out most degrees of freedom at all times, the Miura and Shimoyama approach only worries about them once per step. We add two small changes to this control idea: **1)** use of optimization of some performance metric to generate the nominal trajectory; and **2)** allowing control at multiple times during a step, instead of just once per step, using different control actions and control goals in each interval.

Our approach is also similar to that used by Pratt [36] [37]. And, as mentioned, despite the difference in the evolution of design and controller ideas, there seems to be a convergence between approaches used here and in recent implementations of HZD.

Goals for the control law.   The architecture was chosen so as to have various general features: **1)** it should allow simple implementation of simple controllers such as the Collins one-sensor-measurement-per-step controller [10]; **2)** it should be able to implement intuitive control constructs (e.g., of the Raibert hopper type); **3)** it should gracefully handle sensor delays; **4)** it should be able to come arbitrarily close to any continuous non-linear multi-variable feedback policy; **5)** it should be of a form so that it has a relatively simple expression for a controller that is good enough. Although energy minimization is a central concern, in practice it is subservient to the overall control architecture ('the control law'), so we discuss it later.

General description. The control law we chose to implement is reflex-based: there are triggers (thresholds in dynamic variables or in elapsed time) and responses (motor programs). It is intermittently feed-forward in that there is only minimal stabilizing feedback (e.g., simple local motor-control feedback) during the motor programs that run between triggers. The control does discrete trajectory tracking, but the discrete control is *essentially* discrete, that is, it is not based on an approximation of a continuous controller. The discrete controller is also not impulsive (and can be smooth). There is no tight control over any aspect of the robot configuration or balance. Most of the rest of the paper describes the details of the control law design and implementation. This necessarily includes some machine-specific descriptions. Those features of the approach which we feel are generalizable to more complex robots are summarized at the end.

Concurrent augmented state machines. Each joint motor is controlled by its own concurrently running augmented finite state machine which traverses a set of logical states. The *logical* state describes which (one of a few available) continuous controls is active. This logical state is 'augmented' by the *dynamical* state which describes the measured and estimated aspects of the robot (angles, angular rates, elapsed time since last transition, binary variables for whether each foot is on the ground or not, etc.). Each joint's state machine has access to the full dynamical state and is also informed by global commands (such as for steering, starting, shutdown). For each joint, a gait cycle (2 steps) traverses a circle of states. The transitions are triggered by events which are thresholds in the dynamic state or time (e.g., 'change logical states when ankle angle has reached 0.126 rad). The individual state machines do not communicate with each other explicitly but are synchronized through the shared data, and shared dynamic state estimation. Within one logical state there is a tight (2 kHz) feedback loop that controls, say, the current to the inner ankle (ia).

Although the architecture allows a given motor current to be any function f of the full dynamical state, we only use simple functions with dependence on only some dynamical-state variables; at any instant most joints are in a 1-degree-of-freedom impedance control mode, as expressed by

$$I_{ia} = f(\text{full dynamical state of the robot})$$
$$= A + C_1\theta_{ia} + C_2\dot{\theta}_{ia} + D \tag{9}$$

where the constants A, C_1, C_2 and D vary from one logical state to another. The redundant constant D expresses a dependence of the within-state control on the dynamical state at the start of the state. In most cases the within-state control is fully local, with no feedback from other joints. However, couplings more general than equation 9 are allowed and sometimes used. For example, during single stance the ankle joint current is based on the absolute angle of the foot (which is a state estimation based on data from several sensors). And one of the

dynamical states is the binary variable saying whether the robot is, overall, going too fast or too slow (used to select which actuator should be used to make the correction, described more below).

We call this control architecture "reflex"-based because the change of state is triggered by an event. Because each state has its own motor program, in effect motor programs are reflex actions triggered by events. We feel this is similar to the philosophy of the control of human walking used in the simulations of Geyer [20]. If the triggers are defined to occur at many small changes in dynamical state, say the passage of 10 ms, and the changes of the within-state parameters are small, and if complex forms of f are allowed within a state, then the machine can be a close approximation of continuous gain scheduling. Similarly, if the functions f within a state are allowed to be arbitrarily complicated, then the control can be arbitrarily close to any non-linear state feedback controller. Thus the architecture can span a range of control approaches.

The parameters for this controller (thresholds, numbers of states, gains within a state) are developed by a sequence of optimizations and human decisions: fine-grid trajectory optimization; coarse-grid trajectory definition and optimization; and stabilizing controller definition and optimization. These are described in the sections below for the case of steady forwards walking. For the ultra-marathon walk, the resulting controller had this form

Hip logical states. The hip has six logical states per two-step cycle, three states for each step (also see [2]).

   a) **Double stance.** Starts at heel-strike; ends at toe-off. Hip current = 0.

   b) **Pre-mid swing.** Starts at toe-off; ends when the stance leg is vertical (mid-stance). Hip current = constant+gain × hip velocity.

   c) **Post mid-swing.** Starts at mid-stance; ends at heel-strike. Hip current = 0 (altered by the discrete controller, see equation 10 and nearby equations and text).

   d,e,f) **Repeat.** The same 3 states are visited again but with the roles of the legs reversed.

Inner ankle logical states. Unlike the case for the hip, all 6 logical states need description because the foot has both stance and swing roles in a two-step gait cycle (also see [2]).

   i) **Inner-leg single stance.** Starts at outer leg toe-off; ends when outer ankle reaches, from above, a prescribed height above the floor. Absolute foot angle, not ankle joint angle, is controlled (with the standard impedance controller, equation 9).

   ii) **Inner-leg push-off.** Starts when the outer (swing) foot is a critical distance above the ground; ends at outer-leg heel-strike. It is the same controller structure as inner-leg single stance (above), but with a different target absolute angle and different

compliance (and with damping of zero). This is the nominal control. It is altered by the discrete controller (see equation 10 and nearby equations and text).

iii) Double-stance after inner-leg single stance. Starts at outer-leg heel-strike; ends at inner-leg toe-off. Ankle current is zero (and foot tends to lift because of tensioning spring).

iv) Inner-leg flip-up. Starts at toe-off; ends at mid-stance. A compliant controller relaxes towards the foot being at its uppermost position (the ankle joint has a loosely p-d controlled target angle).

v) Inner-leg flip-down. Starts when stance leg is at a critical angle; ends at inner-leg heel-strike. Compliant controller tries to put foot at a specified absolute angle.

vi) Double stance after heel-strike. Starts at inner-leg heel-strike; ends at outer-leg toe-off. Compliant control aims for target absolute angle.

The outer ankle has an identical (mirrored) set of states. There are 6 logical states for a 2-step cycle.

Selection of parameters to design nominal trajectory. A given candidate set of controller parameters (gains and set points) was evaluated as follows, using simulation. Using the candidate control-parameter set, the robot is simulated in a root-finding context that finds, if the root finding converges, periodic walking motions (the roots are of the Poincare map minus the identity). Then the parameters are adjusted, by human intuition, to lead to desirable walking features, such as these: the motors are far from saturation, the step length is not too large, the final foot velocity is mostly vertically down (and not scuffing), the motions are reasonably stable (even without the feedback layer below), etc.

5   Reflex feedback: discrete intermittent feedback control

Next, to stabilize the gait we wrap a linear feedback controller around this nominal trajectory. If the control output is U (e.g., a set prescribed motor currents), then we decompose the control into the sum of two parts: a trajectory generator part (described above) and a stabilizing controller part (described here). In its simplest form the stabilizing part is D in equation 9. The stabilizing controller is designed to be have output close to zero on the nominal trajectory, designed above.

The feedback controller works to keep the robot near its nominal trajectory, at least loosely. Commonly, feedback trajectory tracking uses a high bandwidth continuous control

a) Trajectory without stabilization   b) Stabilizing controller   c) Trajectory with stabilization
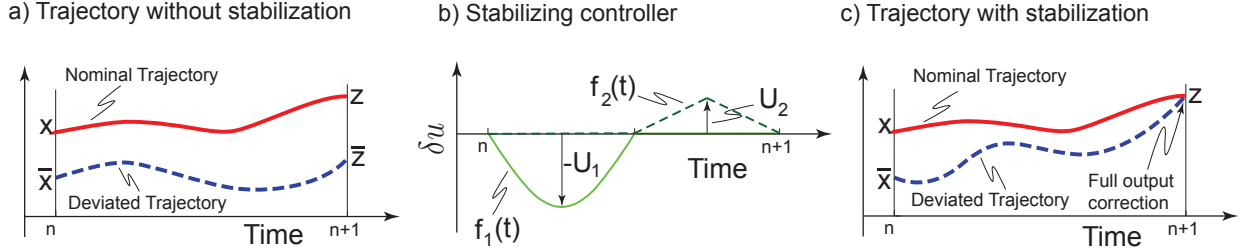
Figure 2: **Reflex control:** A schematic example. **(a)** Shows the nominal (solid red = periodic optimal) and deviated (dashed blue = disturbed by modeling errors, sensor errors or physical disturbances) trajectory for some dynamic variable x of interest which is measured at the start of a continuous interval, namely at section n. This is a generalized state in that it may contain redundant information such as average speed over the whole previous step. The trajectory shown is what would happen without the stabilizing feedback controller. The goal of the stabilizing controller is to reduce the output variable error at the end of the next interval. **(b)** In this schematic example the feedback motor program has two control actions: a sinusoid for the first half cycle and a hat function for the second half of the cycle. The amplitudes U_1 and U_2 of the two functions are chosen at the start of the interval depending on the error $(x - \bar{x})$. **(c)** Shows the new deviated trajectory after switching on the feedback control. By a proper choice of the amplitudes U_1 and U_2, selected at the start of the finite interval, deviations might be, for example, be fully corrected in one step giving a 'deadbeat' controller, as shown here.

along the trajectory. Our reflex based feedback on the other hand is low bandwidth controller that uses measurements at discrete locations in the trajectory to develop intermittent controllers that slowly bring the robot back to the nominal trajectory.

Our reflex feedback architecture is illustrated in figure 2. In this discrete control, we try to track some key variables (the outputs z_{n+1}) at key points of the trajectory (the sections $n, n+1$), using state estimates only at those points (the measurement x_n). The time interval between the measurements n and $n+1$ is typically on the order of the characteristic time scale of the system (say, leg swing time) and not the shortest time our computational speed allows. We linearize the equations of motion and the output as follows

$$\delta x_{n+1} = \mathbf{A}\delta x_n + \mathbf{B}\delta U_n \tag{10}$$

$$\delta z_{n+1} = \mathbf{C}\delta x_n + \mathbf{D}\delta U_n. \tag{11}$$

Using the equations 10 and 11, we set up a discrete linear quadratic regulator (DLQR)

that minimizes the cost J_{dlqr} defined as

$$
\begin{aligned}
J_{\text{dlqr}} &= \sum_{n=0}^{n=\infty} \left(\delta z_{n+1}{}^T \mathbf{R}_{zz} \delta z_{n+1} + \delta U_n{}^T \mathbf{R}_{UU} \delta U_n \right) \\
&= \sum_{n=0}^{n=\infty} \left(\delta x_n{}^T \mathbf{C^T R_{zz} C} \delta x_n + 2\delta x_n{}^T \mathbf{D^T R_{zz} C} \delta u_n + \delta U_n{}^T \{\mathbf{D^T R_{zz} D + R_{UU}}\} \delta U_n \right)
\end{aligned} \quad (12)
$$

where $\mathbf{R}_{zz}$ and $\mathbf{R}_{UU}$ are diagonal matrices that weight the different components of δz_{n+1} and δU_n. The weights $\mathbf{R}_{zz}$ and $\mathbf{R}_{UU}$ are design parameters picked to give reasonably fast return to nominal trajectories (more than 50% return within one step) but without unduly high gains (which tend to lead to motor currents that are beyond safety limits).

The solution to equation 12 gives the linear controller, $U_n = -\mathbf{K}\delta x_n$. The gain $\mathbf{K}$ is obtained by solving the Ricatti equation [33] which we do using the MATLAB control system toolbox (DLQR).

The success of this feedback control method depends on suitable section n, suitable measurement variables x, output variables z and actuator profiles U that have relatively independent effects on target variables. We discuss these next. Also, see [3] for three more applications of this method.

Mid-stance dynamic state evaluation. The angular speed of the stance leg at mid-stance is key for evaluating stability: the robot has to make it to that point at a positive, but not-too-fast, speed. It is also a good time for state estimation because it is not near in time to collisional vibrations which contaminate the dynamic-state estimation. Further, at that instant, there is still time to make foot-placement decisions. Thus we use the 'mid-stance' event as a key time for evaluating the dynamical state. This contrasts with typical numerical simulations of passive-dynamic robots that take (Poincaré) sections just before or just after heel-strike, a natural transition time because it coincides with a dynamical collision event. The top-dead-center (mid-stance) position is sufficiently useful, however, that it is worth introducing an artificial logical-state boundary there.

How can the robot fall down? Given that we do all we can to preclude tripping with our foot-lifting plan, the only possible falls are falling forwards and falling backwards. Because leg-swing speed is limited, if the robot is going too fast the foot will not swing forwards enough and the robot will fall forwards. On the other hand, if the robot is going too slowly the stance leg will not make it over the top-dead-center position and it will fall backwards. Thus the essential control is thus to get the center-of-mass through the top-dead-center position at a positive speed that is close to nominal. It must speed up or slow down the walking, as needed, to keep this top-dead-center speed positive but small enough.

18

Push-off and step-length regulation. There are two main ways to regulate the overall center-of-mass speed of this robot. One is regulation of the ankle push-off: the size and timing of the push-off affect the robot speed (bigger push-off leads to bigger speed of the next step). The other is by changing step length (increasing step length decreases speed of the next step). For simple walking models, it is known that the energy lost during heel-strike collision scales with the square of step length at a given forward speed [18, 19, 26, 42, 48]. Increasing the step length increases the collisional loss and slows the robot. No other motor actions (leg swinging, foot motions) have a significant effect on robot speed.

However, push-off control and step length control each have limitations. To decrease speed, push-off can only be decreased to zero. Thus push-off regulation has limited effectiveness for slowing the robot. On the other hand, decreasing the step length is also bounded in effectiveness because the minimum step length is zero. And, more practically, the step time has to be long enough for the foot to flip up and flip down, so very short steps are not possible. To circumvent these limitations we use each mode where it is most effective. If the robot is going too fast we use increased step length to slow the robot. If it is going too slow we use increased push-off to increase speed. In this way, we expect our quasi-linear controller to have a larger controllable region.

Too fast. In particular, if the robot's mid-stance velocity is greater than the nominal, we alter the foot placement while maintaining the nominal push-off. We try to regulate the stance leg angular rate z_{lr} at the next midstance as well as the downward falling rate z_{ar} of the foot just before the next heel-strike (to prevent mis-timing the next push-off). Thus the regulated variables are $z^f = [z_{lr} \ z_{ar}]'$. The three state variables in δx_n are the stance leg velocity, swing leg angle and swing velocity at mid-stance. The two control variables in U_n^f are constant amplitudes of the hip motors. The first amplitude acts for 0.15 seconds starting from mid-stance phase and the second amplitude lasts for 0.15 seconds after the first one ends. Relative to the nominal trajectory we have

$$\delta x_{n+1} = \mathbf{A}^f \delta x_n + \mathbf{B}^f \delta U_n^f \ , \qquad \delta z_{n+1}^f = \mathbf{C}^f \delta x_n + \mathbf{D}^f \delta U_n^f \qquad (13)$$

Using equations 12, 13 we set up a DLQR problem for determining the 2×3 gain matrix $\mathbf{K}$, which permits calculation of the U values from the x values. In the DLQR we use the weights $\mathbf{R}_{zz}^f = \text{diag}\{1/\sigma_{sr}^2, 1/\sigma_{lr}^2\}$ and $\mathbf{R}_{UU}^f = \rho_{hip}\text{diag}\{1,1\}$, where $\sigma_{ar} = 0.2$ is a user-selected characteristic foot falling rate deviation, $\sigma_{lr} = 0.2$ is the user-selected characteristic deviation in the leg rate (both in consistent units), and $\rho_{hip} = 0.5$ weights the effort by the hip actuators.

Too slow. If the robot's mid-stance velocity is less than the nominal then we increase push-off while maintaining the same step length. So we try to regulate, back to nominal, the values of three dynamic state variables: the stance leg-rate z_{lr} at the next step, the downward velocity z_{ar} of the ankle just before the next heel-strike, and the step length

z_{sl} to the upcoming heel-strike. These three quantities $z^s = [z_{lr} \ z_{ar} \ z_{sl}]'$ are affected by three actions (δU_n): two constant-in-time hip torques (same as the one used for the too fast case above) and the reference angle of the proportional-derivate control on the foot during push-off. The discrete linear equations have the standard form:

$$\delta x_{n+1} = \mathbf{A}^s \delta x_n + \mathbf{B}^s \delta U_n^s \ , \qquad\qquad \delta z_{n+1}^s = \mathbf{C}^s \delta x_n + \mathbf{D}^s \delta U_n^s. \qquad (14)$$

Again using equations 12 and 14 we set up a DLQR problem to determine a 3×3 gain matrix $\mathbf{K}$. We choose $\mathbf{R}_{zz}^s = \mathrm{diag}\{1/\sigma_{sr}^2, 1/\sigma_{ar}^2, 1/\sigma_{sl}^2\}$ and $\mathbf{R}_{UU}^s = \mathrm{diag}\{\rho_{hip}, \rho_{hip}, \rho_{ankle}\}$, where $\sigma_{sr} = 0.2$ is the user-specified characteristic stance leg angular rate, $\sigma_{ar} = 0.2$ is the characteristic falling rate of the foot just before heel-strike, $\sigma_{sl} = 0.2$ is the characteristic step length, and $\rho_{hip} = 0.5$ weights the hip actuator effort relative to the deviation of outputs while $\rho_{ankle} = 2$ weights the feet actuators efforts relative to the deviation of the output variables.

In summary, the stabilizing (discrete trajectory tracking) controller has 15 gains. Six for 'too fast' (two actuations × three sensors) and nine for 'too slow' (three actuations × three sensors).

6   Results

6.1   Simulation validation

Using an earlier (pre-DLQR) heuristic controller, which we hand-tuned on the machine, we compared simulation with the actual robot motion. We had satisfying agreement for the motion. But we had poor agreement with energy use.

On three different occasions this discrepancy revealed errors in our electronics. First we found that our PWM (pulse width modulation) controller did not have adequately high frequency to simulate the DC supply used in our motor bench testing. However a discrepancy remained even with a higher frequency PWM signal (20 kHz increased to 100 kHz). We then discovered that the motor data-sheet inductance value was valid only at low frequencies, and was too small at our higher PWM frequency. In particular at 1 kHz our measurements agreed with the data sheet: 95 μH. But this dropped to 35 μH at 20 kHz and 20 μH at 100 kHz. To reduce these losses, a 47 μH inductor was added in series with each motor, more than tripling the effective inductance from 20 μH to about 67 μH at 100 kHz. Finally, we observed a power spike in the foot-flip control that was not in the simulation. This turned out to be due to a transient in the low-level (ARM7 based) current controller due to a poor choice of gains in that inner (2 kHz) control loop.

Although we cannot give precise attributions for the reasons for the reduction in energy

use by a factor of 8, as the robot developed (from TCOT = 1.6 in 2006 to TCOT = 0.19 in 2011 [43]), just these simulation-discovered electronics refinements might be responsible for up to a factor of 4.

6.2 Long distance walking record

We set ourselves the goal of making Ranger walk a marathon distance of 26.2 miles or 42.2 kilometers, without falling down, without stopping, without recharging, without on-the-fly tuning and without human touch.

On 1-2 May 2011, using a intuitively tuned nominal trajectory stabilized by reflex-based feedback (all tuned offline as described above), Ranger walked 40.5 miles or 65 kilometers, non-stop, and on a single battery charge. Ranger took 186,076 steps at a leisurely pace of 2.12 kilometers per hour (1.32 mph) to set this distance record. The total energy consumption was 493 watt-hours (a nickel's worth[1]). For this ultra-marathon Ranger had a TCOT of 0.28. Table 3 gives more data about this walk.

6.3 Energy-efficiency record

After we achieved the ultra-marathon walk, we tried implementing an optimization based nominal trajectory. To do this, in simulation we solved an energy-optimal control problem with TCOT as the cost metric. We did two optimization sequentially as follows. Details of step 1 and 2 below are in [2].

1. **Fine grid optimization:** The goal of this optimization calculation is to figure out the nature of the optimal solution and cost by doing a reasonable fine grid optimization. Our grid here is time based. The control current is parameterized as a piecewise linear function of time on in each small time interval.

2. **Coarse grid optimization:** The goal of this optimization is to figure out a sparse representation of the control parameterization that captures the optimal solution. Based on the results of the fine grid optimization above, we re-paramterized the control using a sparse state machine (see Section 4). The parameters of the state machine (amplitudes, gains, set points), were then found by re-running the energy-optimization in terms of control parameters.

3. **Experimental verification:** Next, we tried implementing the coarse-grid control representation on the physical robot. Unsurprisingly we found that in this nearly open-loop mode the robot could not walk reliably; it always fell within a few steps. A

---

[1]Electricity averaged 11.2 cents per kilo-watt-hour in United States in 2011. Ranger's 493 watt-hours of energy use would cost 5.5 cents ($0.493 \times 11.2 = 0.55$).

Figure 3: **Ranger's ultra-marathon walk.** On 1-2 May 2011 [43], Ranger walked non-stop for 40.5 miles (65 km) on Cornell's Barton Hall track without recharging or being touched by a human. Some of the crew that worked on Ranger are shown walking behind Ranger during the 65 km walk. The walk lasted for 30 hr 49 min. The robot speed was 1.32mph or 2.12kmph. Total energy usage was 16W of which 11.3 W was used by motors and the rest 4.7 W was used by the computers and sensors. The TCOT was 0.28. This was further reduced to 0.19 after we did energy optimization to compute the nominal trajectory.

> look at the sensor outputs revealed that the outer legs swung slower than the inner ones. This was confirmed by further physical measurements that showed that the outer legs have a higher moment of inertia than the inner legs. Note that the same commands are given to the outer and inner legs. The result was that the outer legs did not swing fast enough to break the forward fall of the robot. Next, we switched our discrete intermittent feedback controller on and tried walking the robot. The feedback controller compensated for the slow speed of the outer hips and the robot was able to walk many steps.

Figure 4 compares the joint angle, motor current and motor power predicted by the open loop coarse grid optimization (solid blue) with those obtained on the physical robot with feedback (red dashed). Note that that which we call 'open loop' does have some fast impedance-control and current-control loops. The spikes in figure d and g are from the feedback controller compensating for the robot asymmetry. Of course, we could have tried to learn this asymmetry, or precompensate for it, but we chose to let the feedback control do what it is supposed to do, compensate for errors, including model errors.

With this control the robot's measured total power of 11.5 W was used by sensors, processors and communications (45%), motor dissipation (34%) and positive mechanical work (21%). In this final implementation the TCOT was 0.19 (compared to a human's TCOT of about 0.3) or, subtracting the electrical overhead, the COT was about 0.11 (compared to a COT for humans of about 0.2 when the resting metabolic costs are subtracted). *mechanical* cost of transport, based on positive motor work, was MCOT = 0.04 (compared to MCOT ≈ 0.05 for the Collins robot and for humans). For comparison, recall that the TCOT of most other legged robots is well over 1, at least 3 times more.

The energetic cost of control.   In going from the fine-grid optimal trajectory control of a simulated robot to our coarse-grid control with stabilizing control of a physical robot, we have reduced the parameters from 126 to 30 and added gait reliability, but at the cost of increasing the TCOT from 0.167 to 0.19, a 14% increase. That is, the incremental cost of simplifying the trajectory description over the fine-grid calculated trajectory was about 8%, as calculated in simulation. The cost of stabilization was about 6%, as calculated by comparing the physical robot with the simulation. The latter is a less reliable estimate because it also includes modeling errors. Nonetheless, for this simple walking task the energetic cost of simplifying the controller is small. And the energetic cost of adding stability seems to be not far from the theoretical prediction that stability (in the absence of disturbances) should have no cost.

The TCOT of the physical robot of 0.19 makes our robot probably slightly more energy-effective than the Collins walker, which had a measured TCOT of about 0.2.

Robustness.   One issue noted in testing made us change in our description of the swing-foot logical-state transition. The fine-grid optimization is always time-based. In the conversion to the coarse-grid optimization we thus had a parameter for the time when push-off should start. From theory we know that the timing of push-off has a huge effect; a factor of 4 difference in simple models, depending on the exact timing [27, 42]. Thus we originally used as a trigger the time until the extrapolated time of heel-strike (based on velocity and height of the foot). Because the optimal gait had a nearly grazing collision, on the robot this led to a high sensitivity to sensor errors and to ground height fluctuations; a small ground-height change would substantially change the time of push-off relative to heel-strike and thus dramatically affect the speed of the next step. Thus to increase the robustness we changed the push-off transition to be at a critical height of the swing foot. When this was given as a parameter to the optimization the optimization chose a gait with swing-leg retraction prior to heel-strike and thus a higher vertical velocity of the swing foot before heel-strike. Note that this robustness from leg retraction due to improved state estimation is unrelated to the stability that leg retraction can provide in open loop control, as described elsewhere[45, 57].

The COT=0.28 controller was robust enough to work reliably on a running track where the maximum slopes were about 1° and maximum step-to-step variation was a few mm,
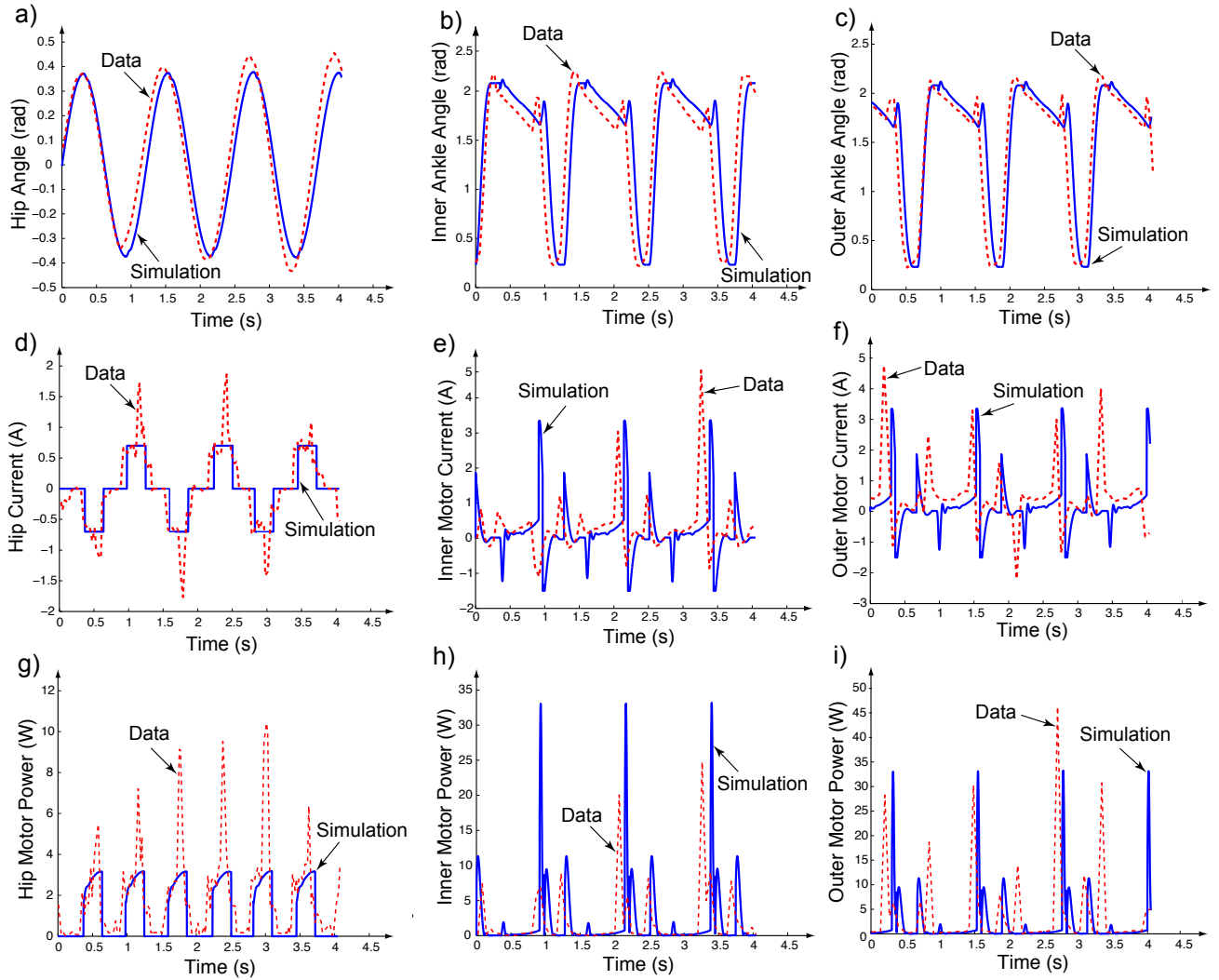
Figure 4: We compare the results of a forward simulation of the robot with experimental data for the same controller. The simulation (solid blue) is periodic, the data (dashed red) is not. The biggest discrepancies are the spikes in hip current and hip power. These are from the stabilizing controller attempting to compensate for differences between model and machine. The horizontal offsets visible above are because the step period of the machine does not exactly match the period of the model.

but not much more. The COT=0.19 controller was only robust enough for a nominally flat linoleum floor.

7   Discussion and conclusion

Our overall design and control approach can be summarized as:

- Design and build a modelable robot. This precludes flat feet, which can have indeterminate collisions, and high-gain control, which can excite poorly characterized vibration modes. Similarly, friction and play need to be minimized, as both are hard to model.

- Make and verify a high-fidelity simulation. This requires that all robot parts be well-characterized, particularly the motors and transmissions, and checking a simulated walking cycle with a real walking cycle.

- Make a fine-grid trajectory optimization (in our case, minimizing TCOT). This optimization includes some events necessitated by discontinuities in the model description (when contacts are made or broken).

- By eye, extract features of the fine-grid optimization to define a coarse-grid parameterization of the trajectories. The discretization of the coarse-grid description defines new events and new logical states. Then, offline, use this coarse-grid description for a new trajectory optimization. Tune the description to minimize complexity and maximize closeness of the objective to the fine-grid objective.

- Define a plausible reflex-based (discrete) controller with a manageably small number of free parameters. These measure the dynamical state at logical state transitions and use the values to adjust the parameters in the motor program in the next logical state. Tune the free parameters of the feedback, in simulation, to maximize simplicity and minimize gains and errors as desired, using, e.g., DLQR.

- Implement the controller on the robot, comparing simulation with experiment so as to check for physical machine, simulation or optimization problems.

In this list, there is no part that we claim as novel. But this is the first time they have been combined to realize energy efficient and realiable walking on a legged robot to the best of our knowledge. The benefit of this approach, using discrete (reflex) control, as summarized above is, that it allows:

- Low gains are possible because extended time is available to make corrections. High gains demand higher machine stiffness to prevent excitation of control oscillations or higher vibration modes.

- Use of a manageably low number of parameters in the trajectory and stabilizing control, with a small resultant energy penalty (compared to the best physically obtainable optimal-energy gait).

- Ability to be progressively refined, because of the small number of free control parameters, so as to better approximate full state feedback control.

While we made a machine that was reliable for its purpose, the biggest defect in the approach, as implemented so far, is that it has no means of systematically maximizing the sizes of allowable disturbances and modeling errors. That is, so far the non-linear system is only used in the trajectory generation but not in the stabilization. Thus the approach does not include a way to systematically, even if by hand, maximizing the non-linear robustness (the tolerance of disturbances and of sensor and modeling errors).

Three uses of events. Discrete events have been used for three purposes in the controller design. First, the fine-grid optimization needs to change its form when the governing equations change form. This occurs when contacts are made or broken. Second, the coarse-grid optimization introduces new logical state transitions (at the grid points) and new logical states (between grid points). Finally, the discrete feedback uses the event transitions (we happen to use the same events as for the course-grid optimization) as the coarse-grid optimization does, and then uses the dynamical state at that time to adjust the parameters used during the time span of the next logical state.

Four kinds of feedback. We think of the primary stabilizing feedback as the adjustment to parameters at state transitions. However, in reality there are three other forms of feedback. First, within each logical state the motors are run by simple continuous-time (approximately, actually a 2 kHz digital control) compliant controllers. Second and innermost, each motor has an inner feedback loop, constantly adjusting the PWM signal to impose a specified motor current. Finally, and independently of the explicit feedback just described, having sensor-based state transitions of any kind is a form of feedback.

How low can you go? On an absolute scale is TCOT = 0.19 low? It's about 2/3 that of a person (TCOT≈0.3) and a fifth or less that that of most robots (TCOT>1). But it's more than that of a Toyota Prius (1400 kg and 4.6 liters/100km $\implies$ TCOT≈ .15) and infinitely more than the theoretical lower limit for walking machines of zero (TCOT = 0.00) [21]. For a machine with Ranger's kinematics and mass distribution a reduction of a factor of at least 4 (to near the MCOT of 0.04) is possible by reducing the electronics overhead from about 5 W to near zero and improving the motor average efficiency from about 65% to near 90%. More realistically, TCOT = 0.1 might be achieved with the same overall design and current technology, by careful selection and design of all electrical components.

Extensions and improvements. Ways to make the robot better include using more efficient electronics and adding knees so that energy can be saved using the natural dynamics

to provide foot clearance. Although this machine was only designed to walk well, its mechanics should also be capable of other simple behaviors (e.g., starting, stopping, walking backwards, balancing on two feet) and higher (meta) states for different behaviors (to switch between, say, walking forwards and backwards). While we had relative success with brushed DC motors, a major modeling issue was the brush contact resistance; the issue would be eliminated by using brushless motors. This would also improve motor efficiency.

Most importantly, the control approach also can be further developed. There are several steps in our controller design that depended on human insight. Some of these could, perhaps, be automated. These include the selection of the coarse-grid parameterization and the architecture of the discrete stabilizing controller. Because the overall control architecture is perhaps more appropriate to motor control than, say, a more general neural network, it may thus be more manageable (i.e., require fewer free parameters for a given quality of control) for optimization in simulation or for offline or online learning. Although the system we have is reliable in a limited context, we have yet to determine how robust it can be made, going beyond linear control approaches, to disturbances and terrain variation.

Appendix: Index to Multimedia Extensions

The multimedia extensions are at http://www.ijrr.org.

Table of multimedia extensions:

Extension	Type	Description
1	Text supplement	Appendix on hardware, modeling and control
2	Video	Ranger's 65 km walk
3	Video	Simulation validation

References

[1] E. Atzler and R. Herbst. Arbeitsphysiologische studien. *Pflügers Archive European Journal of Physiology*, 215(1):291–328, 1927.

[2] Pranav A. Bhounsule and Andy Ruina. Controller design for a bipedal walking robot using simple approximations to energy-optimal solution. *ASME Journal of Mechanisms and Robotics, (in preparation)*, 2013.

[3] Pranav A. Bhounsule and Andy Ruina. Three applications of intermittent control using event-based feedback: Dead-beat control of a time delayed inverted pendulum, pumping a pendulum swing and swing-up and balance of a pendulum. *International Journal of Control, (in preparation)*, 2013.

[4] AC Bobbert. Energy expenditure in level and grade walking. *Journal of Applied Physiology*, 15(6):1015–1021, 1960.

[5] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, and J. W. Grizzle. Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5):57–79, 2003.

[6] M.J. Coleman. *A stability study of a three-dimensional passive-dynamic model of human gait*. PhD thesis, Cornell University, 1998.

[7] M.J. Coleman, M. Garcia, K. Mombaur, and A. Ruina. Prediction of stable walking for a toy that cannot stand. *Physical Review E*, 64(2):022901, 2001.

[8] M.J. Coleman and A. Ruina. An uncontrolled walking toy that cannot stand still. *Physical Review Letters*, 80(16):3658–3661, 1998.

[9] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, 2005.

[10] S.H. Collins and A. Ruina. A bipedal walking robot with efficient and human-like gait. In *Proceeding of 2005 International Conference on Robotics and Automation, Barcelona, Spain*, 2005.

[11] S.H. Collins, M. Wisse, and A. Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.

[12] P.E. Dupont. The effect of friction on the forward dynamics problem. *The International Journal of Robotics Research*, 12(2):1442–1447, 1993.

[13] Boston Dynamics. Petman - bigdog gets a big brother. [available online]. `http://www.bostondynamics.com/robot_petman.html`, April 2012.

[14] Boston Dynamics. Atlas - the agile anthropomorphic robot. [available online]. `http://www.bostondynamics.com/robot_Atlas.html`, September 2013.

[15] H. Elftman et al. Biomechanics of muscle with particular application to studies of gait. *Journal of bone and joint surgery (american volume)*, 48(2):363–377, 1966.

[16] J. Englsberger, C. Ott, M.A. Roa, A. Albu-Schaffer, and G. Hirzinger. Bipedal walking control based on capture point dynamics. In *International Conference on Intelligent Robots and Systems (IROS), 2011*, pages 4420–4427. IEEE, 2011.

[17] A Formalsky. Motion of anthropomorphic mechanism under impulsive control. *Proceedings of Institute of Mechanics, Lomonosov Moscow State University*, pages 17–34, 1978.

[18] M. Garcia, A. Chatterjee, and A. Ruina. Efficiency, speed, and scaling of two-dimensional passive-dynamic walking. *Dynamics and Stability of Systems*, 15(2):75–99, 2000.

[19] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman. The simplest walking model: Stability, complexity, and scaling. *ASME J. of Biomech. Eng.*, 120:281–288, 1998.

[20] H. Geyer and H. Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(3):263–273, 2010.

[21] M. Gomes and A. Ruina. Walking model with no energy cost. *Physical Review E*, 83(3):032901, 2011.

[22] 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):51–64, 2001.

[23] E. Holm. Contribution to the theory of the contact between a carbon brush and a copper collector ring. *Journal of Applied Physics*, 28:1171–1176, 1957.

[24] Y. Ikemata, A. Sano, and H. Fujimoto. A physical principle of gait generation and its stabilization derived from mechanism of fixed point. In *International Conference on Robotics and Automation, Orlando, Florida, USA*, pages 836–841, 2006.

[25] Twan Koolen, Tomas De Boer, John Rebula, Ambarish Goswami, and Jerry E. Pratt. Capturability-based analysis and control of legged locomotion. part 1: Theory and

application to three simple gait models. *International Journal of Robotics Research, to appear*, 2012.

[26] A.D. Kuo. A simple model of bipedal walking predicts the preferred speed–step length relationship. *Journal of biomechanical engineering*, 123:264–269, 2001.

[27] A.D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124:113–120, 2002.

[28] M.L. Latash, J.P. Scholz, and G. Schöner. Motor control strategies revealed in the structure of motor variability. *Exercise and sport sciences reviews*, 30(1):26–31, 2002.

[29] I.R. Manchester, U. Mettin, F. Iida, and R. Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[30] R. Margaria. Positive and negative work performances and their efficiencies in human locomotion. *European Journal of Applied Physiology and Occupational Physiology*, 25(4):339–351, 1968.

[31] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

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

[33] K. Ogata. *Discrete-Time Control Systems*. Prentice Hall, 1995.

[34] D. Owaki, M. Koyama, S. Yamaguchi, S. Kubo, and A. Ishiguro. A two-dimensional passive dynamic running biped with knees. In *International Conference on Robotics and Automation, Anchorage, Alaska, USA*, pages 5237–5242, 2010.

[35] I. Poulakakis and JW Grizzle. Monopedal running control: Slip embedding and virtual constraint controllers. In *International Conference on Intelligent Robots and Systems*, pages 323–330. IEEE, 2007.

[36] J. Pratt and G. Pratt. Intuitive control of a planar bipedal walking robot. In *International Conference on Robotics and Automation*, volume 3, pages 2014–2021. IEEE, 1998.

[37] J.E. Pratt and G.A. Pratt. Exploiting natural dynamics in the control of a planar bipedal walking robot. In *Proceedings of the 36 Annual Allerton Conference on Communication, Control, and Computing*, pages 739–748, 1998.

[38] Jerry Pratt, Twan Koolen, Tomas De Boer, John Rebula, Sebastien Cotton, John Carff, Matthew Johnson, and Peter Neuhaus. Capturability-based analysis and control of legged locomotion. part 2: Application to m2v2, a lower-body humanoid. *International Journal of Robotics Research, to appear*, 2012.

[39] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, et al. Bigdog, the rough-terrain quadruped robot. *Proceedings of the 17th International Federation of Automatic Control*, 2008.

[40] M.H. Raibert. *Legged robots that balance.* MIT press Cambridge, MA, 1986.

[41] A. Ruina. Nonholonomic stability aspects of piecewise holonomic systems. *Reports on mathematical physics*, 42(1-2):91–100, 1998.

[42] A. Ruina, J.E.A. Bertram, and M. Srinivasan. 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):170–192, 2005.

[43] A. Ruina et al. Cornell ranger 2011, 4-legged bipedal robot. [available online]. `http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/Ranger2011/`. Or Google search for: cornell ranger, April 2012.

[44] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent asimo: System overview and integration. In *Proc. of International Conference on Intelligent Robots and Systems, Lausanne, Switzerland*, volume 3, pages 2478–2483, 2002.

[45] A. Seyfarth, H. Geyer, and H. Herr. Swing-leg retraction: a simple control model for stable running. *Journal of Experimental Biology*, 206(15):2547–2555, 2003.

[46] C.E. Shannon. Scientific aspects of juggling, 1993.

[47] 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*, 30(9):1170–1193, 2011.

[48] M. Srinivasan. *Why walk and run: energetic costs and energetic optimality in simple mechanics-based models of a bipedal animal.* PhD thesis, Cornell University, 2006.

[49] M. Srinivasan and A. Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72–75, 2005.

[50] S.H. Strogatz. *Nonlinear dynamics and chaos.* Addison-Wesley Reading, 1994.

[51] T. Sugihara and Y. Nakamura. Whole-body cooperative balancing of humanoid robot using cog jacobian. In *International Conference on Intelligent Robots and Systems*, volume 3, pages 2575–2580. IEEE, 2002.

[52] R. Tedrake, I.R. Manchester, M. Tobenkin, and J.W. Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052, 2010.

[53] E. Todorov and M.I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.

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

[55] M. Vukobratovic and D. Juricic. Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*, 16(1):1–6, 1969.

[56] E.R. Westervelt. *Toward a coherent framework for the control of planar biped locomotion.* PhD thesis, University of Michigan, 2003.

[57] M. Wisse, C.G. Atkeson, and D.K. Kloimwieder. Swing leg retraction helps biped walking stability. In *Proceedings of 2005 IEEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan*, 2005.