CLAWAR 2012 – Proceedings of the Fifteenth International Conference on
Climbing and Walking Robots and the Support Technologies for Mobile Machines,
Baltimore, MD, USA, 23 – 26 July 2012

DESIGN AND CONTROL OF RANGER: AN ENERGY-EFFICIENT, DYNAMIC WALKING ROBOT

Pranav A. Bhounsule, Jason Cortell and Andy Ruina

Biorobotics and Locomotion Laboratory,
306 Kimball Hall, Cornell University,
Ithaca, NY 14853, USA
E-mail: pab47@disneyresearch.com, jbc2@cornell.edu, ruina@cornell.edu

We present the design and control of an energy-efficient, knee-less, essentially planar, four-legged bipedal robot called Ranger. In separate trials, Ranger: 1) walked a 40.5 mile ultra-marathon on a single charge and without human touch, setting a robot distance record; and 2) walked stably at Total Cost Of Transport (TCOT= total energy used per unit weight per unit distance travelled) of 0.19, apparently less than that of any other legged robot to date. Key design features are: a light weight and high strength box body, low-inertia leg design for fast and efficient swing, foot actuation that combines toe-off and ground clearance, a steering mechanism that enables turning of this essentially planar robot, and a low-power modular networked electronics hardware system. The model-based control approach uses a simplified offline trajectory optimization with a reflex-based feedback controller for stabilization. Ranger's reasonable success suggests that these design and control ideas could be extended to the development of an energy-efficient higher degree of freedom, 3-D bipedal robot.

1. Introduction and Motivation

Present legged robots are either highly energy-efficient like passive dynamic walkers[5] but fall down frequently, or like PETMAN,[1] BigDog,[6] and ASIMO[8] are more robust but guzzle energy. For legged robots to be useful they will have to be energy-efficient as well as robust. Here we present some progress made in this direction: the design and control of a simple planar robot with just 4 degrees of freedom that is highly energy-efficient and reliable at the limited task of steady level walking. We believe that aspects of the design and control approach we present here can scale up to more complex machines and to more versatile tasks.

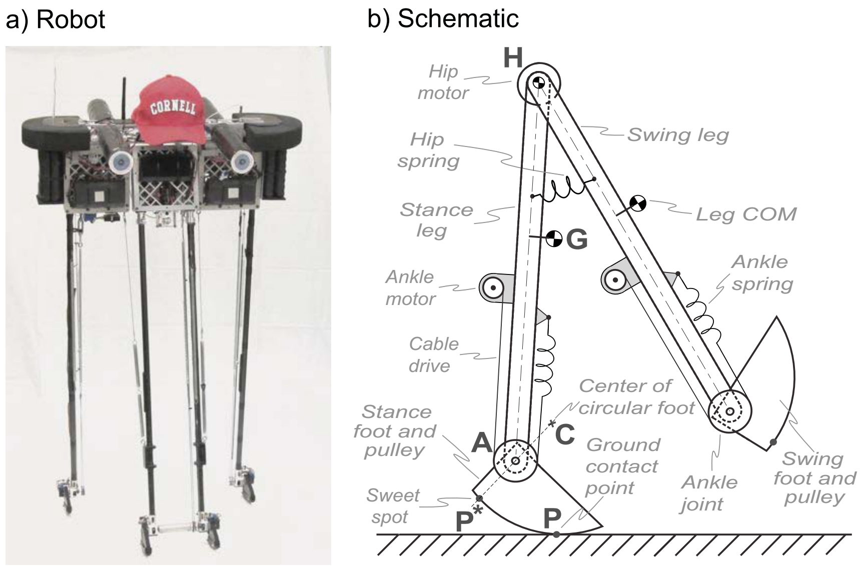

Fig. 1. **(a)** Ranger. **(b)** 2D schematic. The fore-aft cylinders with 'eyes' and the foam 'ears' (both visible in photo) are only for shock absorption in case of falls. The hat is decorative (hollow). The closed and rigid aluminum lace boxes, conceptually shown as point H, house all of the motors and gearing, various pulleys for the ankle cable drives, and most of the electronics (on the drawing the hip motor location is only schematic). There are two boxes connected by a hinge: an outer box, shaped like an upside-down U, rigidly connected to the outer legs, and an inner box, filling the space in the U, holding the inner legs (each of which can twist for steering). The hip spring, which aids leg swing, is shown schematically as symmetric between the two legs but shows as a diagonal cable and spring in the photograph. The feet are shaped so that toe-off is possible, so that no torques are needed during single stance, and so that ground clearance during swing can be achieved (by rotating the toe towards the hip).

2. Ranger

The Cornell Ranger is a four-legged knee-less biped (figure 1 a). 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 so far, it needs to be started manually, and steering is done with a model plane type radio control.

Hardware. Leg swing is powered by a DC brushed hip motor. Each of the ankle pairs (one inner pair and one outer pair) has a motor along the hip axis that actuates the ankle via a one-way (toe-off) cable drive. Foot lifting, for ground clearance, is powered by a return spring on each ankle. A fourth motor twists the inner legs about a vertical axis with each step in order to steer the robot; the amount and direction of steering is governed by the remote control. The motors and electronics are powered by seven 25.9 V lithium-ion batteries with a total capacity of about 493 watt-hours.

Electronics. The main control loop runs, with no supervisory operating system, on an ARM9 microcontroller. Four ARM7 processors on custom boards monitor and control the three main joints (the outer ankles are counted as a single joint) and the steering. Two more ARM7 processors supervise the on-board communications network (CAN), the Bluetooth data reporting, and the onboard data display and lights. The Inertial Measurement Unit (IMU) also contains a proprietary microprocessor board. The multi-processor bus-based architecture (CAN) was chosen to facilitate design evolution, to simplify overall wiring (e.g., so a new sensor could be added without new wiring into 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 absolute angle sensor. 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 on the ARM7 processors.

Software. The total custom control code is about 10,000 lines of C and C++ code, the bulk of which is associated with low-level measurement, low-level control and communications protocols. The main control loop is based on hierarchical concurrent finite state machines. 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.

For debugging and development, walking data is viewed and logged via wireless system. Although control parameters can be adjusted wirelessly mid-walk, during attempts at walking distance records autonomy is maintained by sending to the robot only steering signals and requests for data (e.g., cumulative number of steps, battery voltage) and not sending any walking control nor any changes of walking control parameters.

3. Control

Our model-based robot control algorithm is presented next. We summarize the key aspects of the control. For more details please see.[3,4]

Model for simulation. The 2D model (figure 1 b) consists of two rigid legs characterized by a mass, center of mass and polar moment of inertia about the center of mass. The feet are massless and round and are assumed to roll without slip when on the flat, level ground. The robot is powered by three motors; one for the hip and one for each ankle. The ankle motors are connected to the feet through a linear, torsional spring ('series elastic actuation'). Also, there is a zero-free-length hip spring pulling the legs to their parallel position. We assume a walking step to consist of the following concatenation of phases: single stance → heel-strike → double stance. The foot-to-ground collisions are assumed to be instantaneous, hard, with no slip and bounce, and with continuity in the position by discontinuity in the velocities. The power to the DC motor model is the sum of two parts: mechanical work, and electrical dissipation in the motor resistance and brush contact. The output torque from the DC motor is proportional to the current to the motor minus the frictional torque, which we found to be load-dependent.

Energy-optimal trajectory control (Fine grid). We try to find that periodic motion which minimizes the Total Cost Of Transport (TCOT), defined as the power used per unit weight per unit speed. The optimization variables are the time to take one step, all initial angles, and velocities in single stance. The control functions to be optimized are motor currents, which are parameterized as piecewise linear functions of time. We also enforced various constraints: current should be within actuator bounds, tensional contact with the ground is forbidden, and the swinging leg's foot should have sufficient ground clearance.

Simplifying the optimal trajectory. In order to implement the energy-optimal trajectory control solution presented above we proceed in two stages; first, we simplify the description of the near-optimal trajectory, and then we add a reflexive (event-based) feedback. These are discussed in the next two sections.

(i) Optimizing the coarse-grid description. First, we use a simpler coarse-grid representation that has few parameters. We did this by looking at the general shape of the fine-grid optimization and choosing intervals where we could use a simpler controller (see figure 2). The coarse-grid form is chosen manually so as to best catch the features of the fine-grid optimization but with fewer terms. Our goal is to achieve a gait with close to the same TCOT, informally doing a simultaneous optimization of both energy use and simplicity.

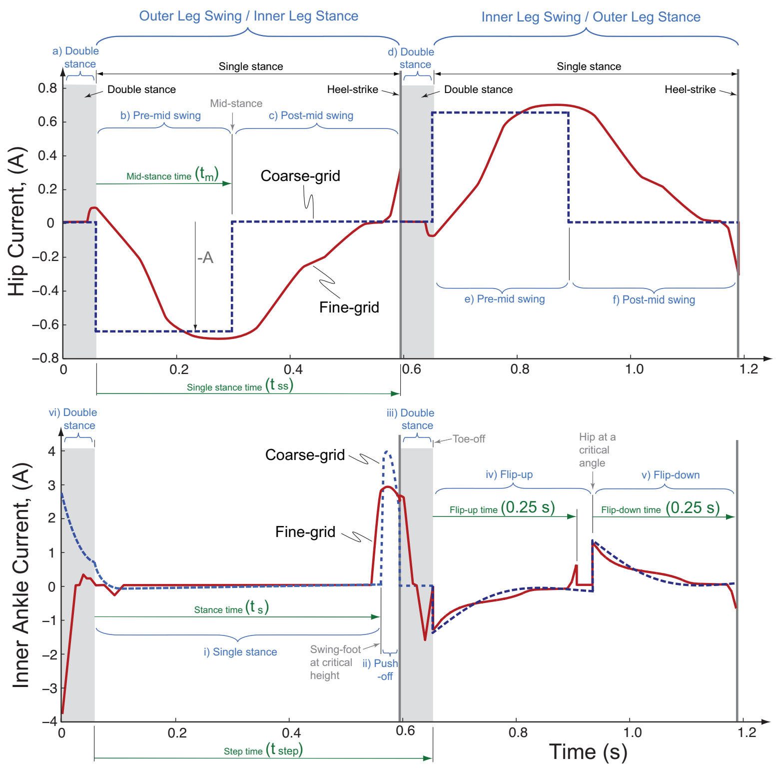

Fig. 2. Numerically, the fine-grid optimum (solid red) is obtained by solving the energy-optimal control problem (see energy-optimal trajectory control section). The current obtained was a piecewise linear function of time involving about 126 parameters, with TCOT = 0.167. The coarse-grid optimal (dashed blue) motor currents are the simplified version of the fine-grid solution (see optimizing the coarse-grid description above). Here, we first divide an entire step into phases (e.g., pre-mid swing, foot flip-up). Next, in each phase we use a compliant controller that involves selecting a constant, a stiffness, and a damping term (i.e. $I = I_0 + K\theta + C\dot{\theta}$, where I is current, θ is angle and I_0, K, C are constants). Our coarse-grid description has only 15 parameters and a TCOT = 0.18. Thus, in going from a fine grid to a coarse grid we have reduced the parameters by almost a factor of 8, but have increased the cost by only 14%.

(ii) 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., current), then we decompose the control into two parts: a trajectory generator part (coarse-grid optimization described above) and a stabilizing controller part (described here).

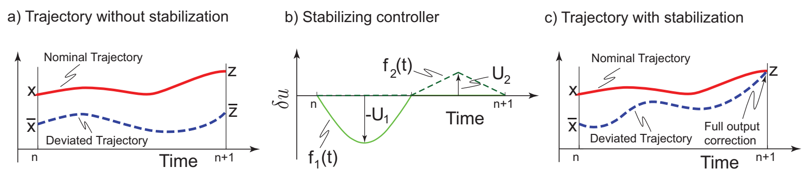

Fig. 3. A schematic example of our reflex-based discrete intermittent feedback controller

We illustrate the idea with a schematic example. Consider the nominal trajectory of a second-order system shown as a continuous red color line in figure 3a. Let n and $n+1$ be instances of time at which we are taking measurements from sensors. Let us assume that we take two measurements, $x = [x_1 \;\; x_2]'$ at time n (e.g., a position and velocity). We are interested in regulating two outputs, z_1 and z_2 at time $n+1$.

Suppose that due to an external disturbance the system deviates from its nominal trajectory, and thus follows the deviated trajectory shown as a dashed blue line in figure 3a. Now, the sensors read $\bar{x}$ at time n. In the absence of any feedback correction, the output values would become $\bar{z} = [\bar{z}_1 \;\; \bar{z}_2]'$.

The stabilizing controller measures deviations at time n ($\delta x_n = x - \bar{x}$) and uses actuation to minimize the deviations in output variables ($\delta z_{n+1} = z - \bar{z}$). For illustration we choose two control actions, $\delta u_n = [U_1 f_1(t) \;\; U_2 f_2(t)]'$, a half sinusoid and a hat function, each active for half the time between time $n+1$ and n. This is shown in figure 3b. We adjust the amplitudes of the two control functions U_1 and U_2, based on measured deviations δx_n, to regulate the deviated outputs δz_n. For example, with a proper selection of the amplitudes of the two functions it is possible to fully correct the deviations in the output variables, as shown in figure 3c.

In order to compute the amplitudes U_1 and U_2 we use a linear controller as follows: We linearize the system about the nominal trajectory (actually, we only linearize the section to section map). The sensitivities of the dynamic state to the previous state and the controls $\delta U_n = [U_1 \;\; U_2]'$ are: $\mathbf{A} = \partial x_{n+1}/\partial x_n$, $\mathbf{B} = \partial x_{n+1}/\partial U_n$, $\mathbf{C} = \partial z_{n+1}/\partial x_n$ and $\mathbf{D} = \partial z_{n+1}/\partial U_n$.

Thus we have $\delta x_{n+1} = \mathbf{A}\delta x_n + \mathbf{B}\delta U_n$ and $\delta z_{n+1} = \mathbf{C}\delta x_n + \mathbf{D}\delta U_n$. Using the above two equations, we set up a discrete linear quadratic regulator (DLQR) that is to find δU_n from δx_n. Our resulting linear controller has the form $\delta U_n = -\mathbf{K}\delta x_n$.

On Ranger. In order to implement the stabilizing controller described above, we used the following heuristics: we measure the robot state x_n during mid-stance, i.e., when the stance leg is vertical; then we try to regulate the robot velocity at the next mid-stance z_{n+1}. Our control actions U_n are as follows: If the robot velocity at mid-stance is slower than the nominal, then the robot has lower energy than the nominal at mid-stance and so we inject energy by pushing off harder (similar to walking uphill). If the robot velocity at mid-stance is faster than nominal, then the robot has higher energy than the nominal at mid-stance and so we extract out energy by increasing the step length (similar to walking downhill). We use the linearized version of the equation presented above to do the stabilizing control.

4. Results

(i) Energy-efficiency record. Table 1 lists the various gait parameters and energetics for the fine-grid trajectory control problem, the coarse-grid control representation and the experimental robot data.

For the energy-optimal trajectory control problem we had 126 parameters in all and a TCOT of 0.167. With our coarse-grid representation we reduced the parameters to 15 while increasing the TCOT to 0.18, a 7% increase in cost. However, the coarse-grid representation could not realize stable walking and had to turn on the stabilizing controller. We see that in going from the optimal trajectory control of the robot to our coarse-grid control with stabilizing control, 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. The TCOT of 0.19 makes our robot perhaps slightly more energy-effective than the Collins walker, which had a TCOT of about 0.2.[2]

(ii) Long distance walking record. The broad goal of the Ranger project[7] is to develop a reliable robot capable of walking long distances on minimal amounts of energy. 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, and without recharging.

On 1-2 May 2011, before we had optimized the energy-effective controller presented earlier in this paper, Ranger walked 40.5 miles or 65 kilometers, non-stop, and on a single battery charge. Ranger took 186,076 steps

Gait Parameter	Fine-grid	Coarse-grid	Experiment
Total COT	0.167	0.180	0.190
Motor COT	0.087	0.100	0.110
Overhead COT	0.083	0.080	0.080
Hip COT	0.019	0.018	0.030
Ankle COT (push-off)	0.029	0.052	0.046
Ankle COT (foot-flip)	0.039	0.029	0.034
Step Length	0.38	0.39	0.38
Step Velocity	0.64	0.66	0.62
Step Time	0.60	0.60	0.61
Double Stance (%)	9.5	5.0	3.0
Control Parameters	126	15	30

at a leisurely pace of 2.12 kilometers per hour or 1.32 miles per hour to set this distance record. The total energy consumption for Ranger for this walk was 493 watt-hours and it had a TCOT of 0.28 (as noted, this was later reduced to TCOT = 0.19).

References

1. Boston Dynamics. PETMAN - BigDog gets a big brother. [available online]. `http://www.bostondynamics.com/robot_petman.html`, April 2012.
2. 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.
3. P. A. Bhounsule, J. Cortell, B. Hendriksen, J. G. D. Karssen, C. Paul, A. Ruina, A robot that can walk far using little energy, *The International Journal of Robotics Research (in preparation).*
4. P. A. Bhounsule, *A controller design framework for bipedal robots*, PhD Thesis, Cornell University, 2012.
5. T. McGeer, Passive dynamic walking, *The International Journal of Robotics Research*, 9(2):62–82, 1990.
6. M. Raibert, K. Blankespoor, G. Nelson, R Playter and others, *Proceedings of the 17th International Federation of Automatic Control*, 2008.
7. 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/`, April 2012.
8. 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*, 3:2478–2483, 2002.