Cornell Ranger

DESIGN OBJECTIVES

Robot engineer Daniël Karssen shows off his ankle and foot. The beam arrangement of the foot allows it to flex slightly under load, and the resulting deflection is measured by a low-cost optical interrupter sensor (Fairchild H21A1) in the slot by his thumb.

Reliability:

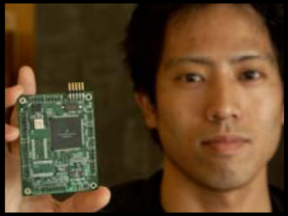
-Flip-up foot design, no torso, no knees, for minimal joint number
-Faulhaber precision DC motors and gearboxes
-High-flex 3M ribbon cable between inner and outer legs
-For power connection, fine-stranded robotic cable
-Optical sensing of foot contact; medical implant sensor cable
-D-shaped shaft/pulley connections, for positive grip
-Optical encoder position and velocity measurements
-Over-travel limit switches for feet
-Assembled with high-adhesion Metregrip 303 epoxy

Autonomous operation, with minimal power consumption and long battery life:

-Efficient Faulhaber permanent magnet DC motors
-ST Micro VNH2 integrated motor drivers, 0.019 ohms on
-Processing, memory, and I/O are largely on a single Freescale 56F8347 chip
-Switching voltage regulator for electronics: 2.6 watt total from battery.
-100 watt-hour max. (80 usable) 11.1 volt lithium-ion batteries
-Low power Radiotronix WI-232DTS radio transceiver modules

Minimal mass:

-Carbon-fiber leg tubes
-Sheet aluminum box structure, with laser-cut holes to reduce weight
-CNC-machined top truss to minimize overhung mass on outer legs
-Lightweight cable drive for the feet
-7075 aircraft aluminum for the feet
-Lightweight, high-energy lithium-ion batteries



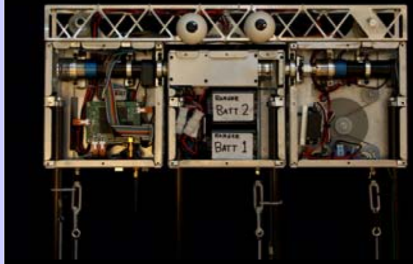
Designer Ko Ihara shows the 56F8347 microcontroller (MCU) board. It operates at 60 MHz, uses about 2 watts, and features a 3-axis accelerometer, three MEMS rate gyro axes, 32 12-bit analog inputs, a switching voltage regulator, and many pulse-width modulation (PWM), timer, optical encoder, and digital I/O channels. (Kionix X and Y gyros not shown in photo)

A Ranger-specific daughter board attaches to the top of the MCU board, providing a wireless transceiver module, three 10-amp VNH2 motor drivers, a watchdog timer for the motors (to protect the robot from software crashes), and wiring distribution.

Steering (at least a little):

A "4-leg" biped is not easily turned, but for long-distance walks we needed it to be able to make gentle turns (e.g., at the ends of a running track).
A single-cable system is used to drive the feet downwards; a pair of long, near-constant-force springs pull the feet upward when the motor unwinds the drive cable. This means that, for the outer feet, if the left cable is pulled a little more or less than the right cable, the feet will move accordingly, and the left-right symmetry of the robot will be broken. If, as a result, the right foot slips more than the left (or vice versa), the robot will turn.

Differential cable motion is driven by an eccentric pulley on a gear, powered by a standard radio-control servo motor and control unit.



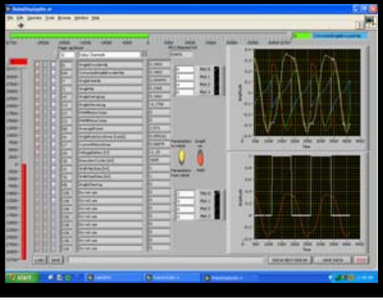
Visible above is the machined top truss, providing a rigid connection between the two outer boxes. Motors are along the hip rotation axis, driving (from left to right) the outer feet, the inner feet (motor is inside secondary support box), and the hip (through a flexible coupling). The steering motor, gears and pulleys are visible in the right box.

Mass distribution for efficient walking:

Experience with a previous robot suggested that asymmetry in mass properties between the inner and outer leg pairs led to an asymmetrical gait that was less efficient and more complex to control. Further, we aimed to minimize the mass of the feet and legs, allowing us to swing them faster for a given motor torque.
-All three drive motors are mounted along the hip axis, to minimize the effects of the weight imbalance resulting from having two motors mounted in the outside leg boxes, and one in the inner box.
-CNC-machined top truss to minimize overhung mass on outer legs
-Lightweight cable drive for the feet
-7075 aircraft aluminum for the feet

Wireless data logging and parameter adjustment:

Arrays of data and control parameters in the robot microcontroller are matched to similar arrays in a LabView program running on a laptop, and synchronized by a wireless communication protocol. Control parameters (up to 96) on the robot can be changed while it is walking, and the user can select 16 of 128 data variables for logging and graphing.



Jason Cortell

Cornell University

Dynamic Walking III
June 2007

Graduate student Gregg Stiesberg walks with Ranger during a record distance attempt at Cornell's Barton Hall indoor track. We entered the Ranger in an American Cancer Society Relay for Life this spring in honor of Gregg, who is battling cancer. He is progressing well and we hope to have him back at Cornell in December 2007.

Cornell Ranger was built over the past year in Andy Ruina's Biorobotics and Locomotion Lab at Cornell University, with the aid of several dozen students. TU Delft student Daniël Karssen is due special credit for his many hundreds of hours on Ranger during fall of 2006, culminating in its 1 km walk in early December.

Funding was provided by Cornell's College of Engineering, and a grant from the National Science Foundation.

CONTROLLER OPERATION

Discrete-time feedback loops:

We are in the process of experimenting with several feedback loops intended to allow the robot to continue walking despite internal or external disturbances, including bumps, wind, hills, backlash, and sensor noise. Examples include:

Step size: Larger steps are taken when the stance leg angular rate, sampled at midstance (fully upright, legs together) begins to exceed a preset value.

Pushoff angle: The angle of pushoff, and thus the pushoff energy, is decreased if the stance leg angular rate, sampled just before heelstrike, exceeds a given parameter value.

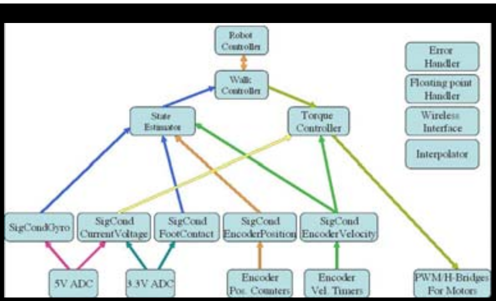


Diagram of the control modules in Ranger. The bottom row modules are implemented in electronics; all of the upper modules are written in C, using CodeWarrior from Freescale/MetroWerks.

Walk Controller Operation:

Three parallel state machines are used, one for the hip control, one for the swing feet, and one for the stance feet. (Not shown – an experimental additional state to swing the leg back to catch the robot if it begins to tilt backward.)

Stance Feet

Post-heelstrike: The stance feet are held at the same preset angle as they were pre-heelstrike (hold down). Transition to the next state occurs when the swing feet go to the hold up state.

Pushoff-prepare: The stance feet go to a new angle, in preparation for pushoff. Transition to the next state occurs at heelstrike.

Swing Feet

Pushoff: A predetermined torque is applied to the swing foot until it reaches the commanded angle. When the applied torque drops to a negligible level (as a result of successful pushoff), the foot state machine goes to the next state.

Hold up: The foot is flipped up out of the way, and held via position control. Transition to the next state occurs when the leg is within a fixed angle of its target position, most of the way through the swing.

Hold down - swing: The foot is flipped back down to a preset angle, ready for heel strike, the next transition condition.

Hip

Start: Hip position is controlled to stay fixed at where it was at heelstrike. The hip state machine goes to the next state after the foot has finished pushing off, and has lifted far enough up to clear the floor.

Swing: A preset torque is applied to swing the leg forward. Transition to the next state occurs when a preset angular velocity is achieved.

Free swing: Zero torque is applied. Transition occurs when the leg swing reaches the start brake angle.

Brake: A reverse torque is applied, based on leg angular velocity at the start of the state. Transition occurs when the leg stops, or reaches the desired swing angle.

Hold: Hip position is held in place until heelstrike (actually a little higher, to take out slack in the gears and controller).

Heelstrike (swing leg contacts ground)

All three state machines undergo a state transition at this point, marking the start of another step. It is at this point that the former stance leg and feet become the new swing leg and feet, and vice versa (rather confusingly!).