Quadruped Robot

Michael Auchter David Baty
Mark Jacobson Tom Most

Requirements Statement

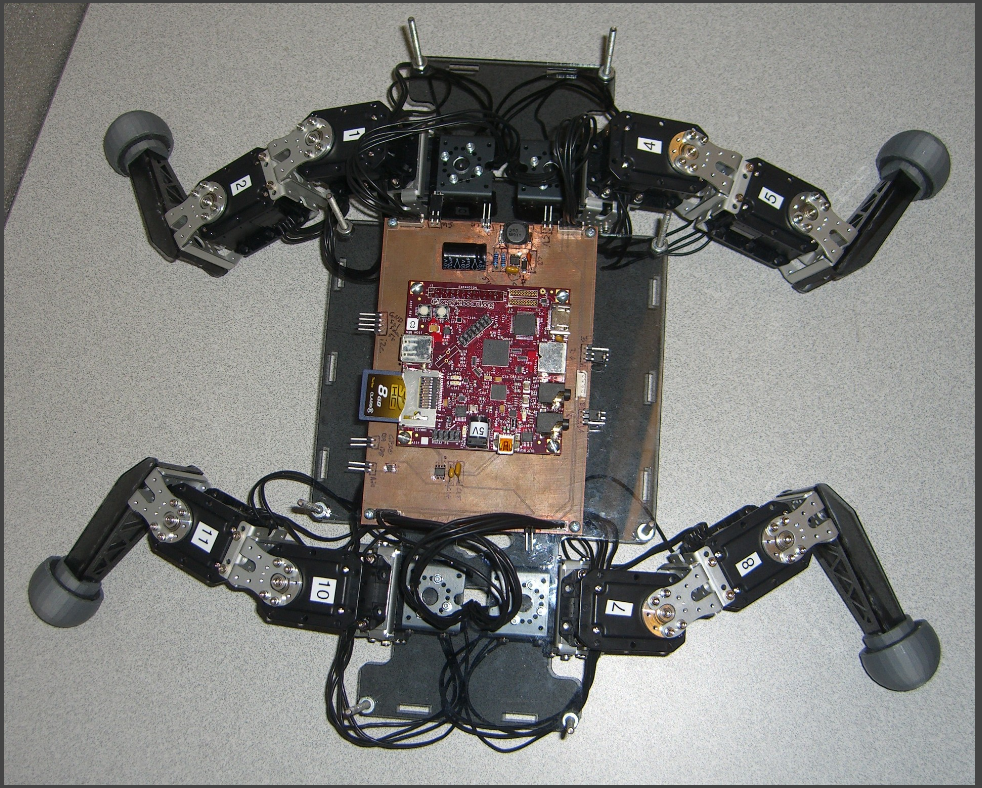
Produce a quadruped robot capable of navigating an environment by both traversing and avoiding obstacles. The robot's gait must be dynamically generated using an onboard BeagleBoard embedded Linux system, with the goal of participating in Texas Instruments' Analog Design Competition.

Starting Point

- Quadruped chassis developed last year in collaboration with Seoul National University of Technology
- Previous electronics were large and heavy, resulting in slow walking speed
- Time pressure limited sophistication of software

Robot Overview

- The robot is comprised of three main components
- Four legs, each made of three servo motors
- The BeagleBoard, the brain of the robot
- An interface board to connect the BeagleBoard to sensors, servos, and peripherals.



Why the BeagleBoard

- Previous design utilized two processors, interfaced over a CAN bus
  - dsPIC33 for motion control
  - TMS320C6x based DSP board for image processing
- This was unnecessarily complex, very expensive, and physically large
- Our solution is to utilize the Texas Instruments BeagleBoard

Why the BeagleBoard

- Attractive for a number of reasons
  - Physically small: 3" x 3.1"
  - OMAP3 SoC
    - 600MHz ARM-Cortex general purpose core
    - 430MHz TMS320 DSP core
  - Low power consumption (~2W)
  - Low cost ($150)
  - Fully supported under Linux
  - Large expansion header providing access to extra UARTs, SPI, I^2C, and GPIO

Current Status

- Hardware: Operational
  - Largely unchanged, with removal of turret and addition of grip to the feet
- Electronics: Rev. 1 interface board ready for manufacture
- Software: Porting recorded gait
  - Static walk complete
  - Dynamic walk in progress
  - Can walk forward, turn

Software

- Written in C using GLib
- Threading used for major subsystems
  - Gait generation & motor control
  - Wiimote communication
- Future tasks
  - Interfacing with sensors
  - Image processing

Kernel Modifications

- RS485 is a half-duplex bus, our code must manually enable the drive pin
- Originally done in userspace; context switch latency resulted in missed packets from the motors
- Moved this into the kernel; timing granularity was too coarse, still missed packets
- Rewrote to use hrtimers, consistently meet timing deadlines

Interface Board, Rev. 0

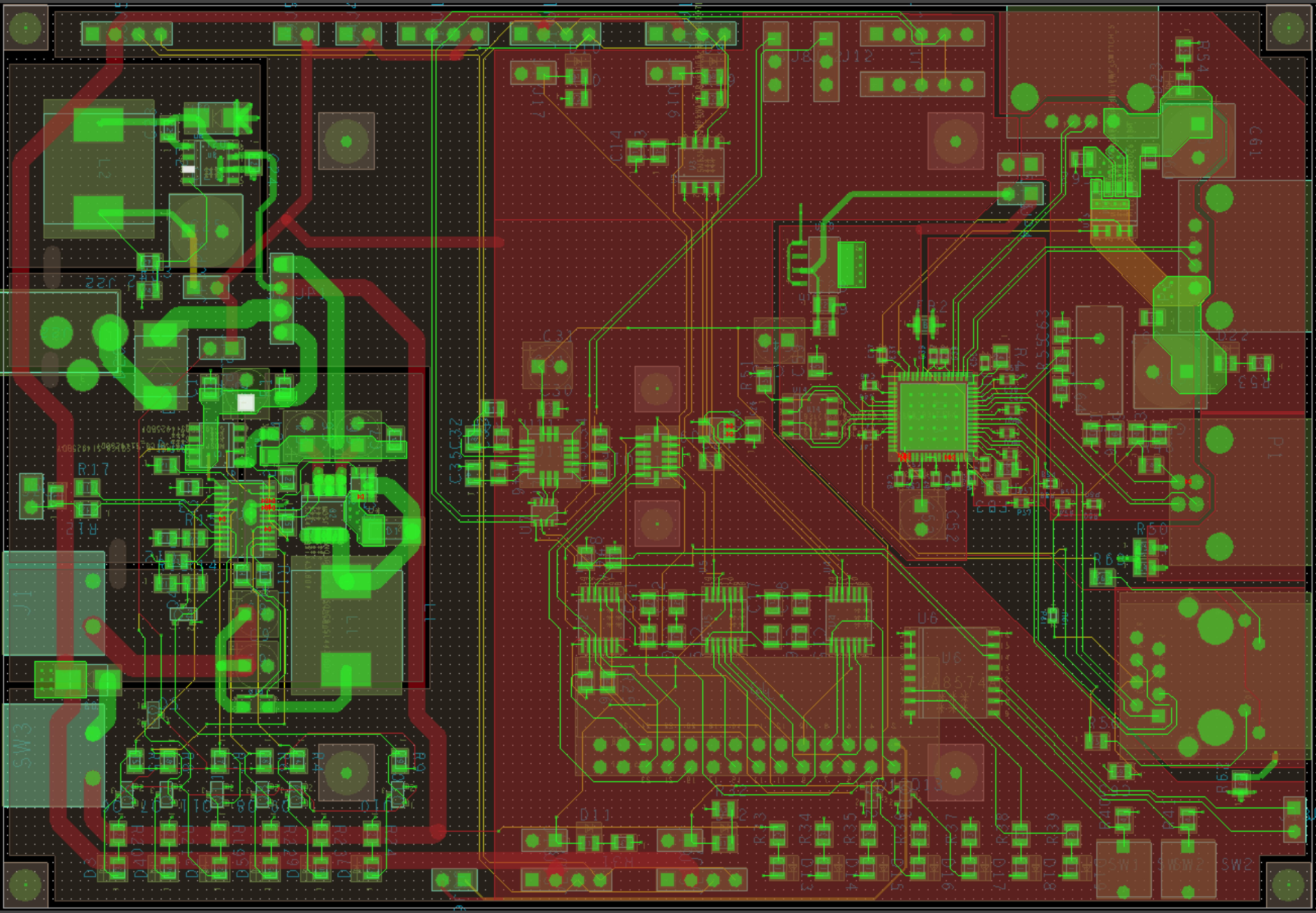
- PCB to interface BeagleBoard with the rest of the robot
- Components:
  - Switching regulator to reduce 14.8V input to 5V for the BeagleBoard
  - Level-shifters to allow interfacing with 3.3V/5V peripherals (I2C, GPIO)
  - RS485 transceiver to communicate with servos

Interface Board, Rev. 0.5

- A daughter board with two I^2C devices
  - IO Expander for foot-switches
  - 3-Axis accelerometer
- Allowed us to begin work on software and verify correct operation
- Low-level driver code complete

Interface Board, Rev. 1

- The next iteration of the interface board will have all the functionality of the current board, with some additional features
  - On-board charging and protection circuit for Li-ion battery pack
  - Triple-axis accelerometer/gyroscope
  - Connectors for foot contact sensors
  - Support for sonar and infrared sensors
  - Support for USB peripherals
  - On-board USB hub and USB to ethernet adapter
  - 4 layer board for USB, ethernet and routing



Dynamic Walking

- Major components
  - Forward kinematics
  - Inverse kinematics
  - Center-of-gravity (CoG) calculation
- Use this information to generate a gait

Kinematics

- Forward: Given joint angles, where is the foot?
  - Easily solved using matrices
- Inverse: Given Cartesian coordinates for the foot, what joint angles place the foot there?
  - Many possible solutions, not necessarily closed form
  - Chosen solution: neural network
  - Feedforward neural network with 3-10-10-3 topology

Gait Generation

- Challenges
  - Neural network does not provide perfect answers (some degree of error)
  - Accurately measuring CoG
- Basic strategy:
  - Position CoG in side of "stability triangle" composed of stationary legs
  - Move a single foot forward at a time
  - This is a "stable" gait

Known Issues

- USB/UART conflicts
    - High amounts of USB traffic seems to negatively impact UART communication

Any Questions?