Spencer Burdette
March 9, 2007
sburd@gwu.edu

Self-Balancing Robot
Project Proposal

Abstract

This project will undertake the construction and implementation of a two-wheeled robot that is capable of balancing itself. The structural, mechanical, and electronic components of the bot will be assembled in a manner that produces an inherently unstable platform that is highly susceptible to tipping in one axis.

The wheels of the robot are capable of independent rotation in two directions, each driven by a servo motor. Information about the angle of the device relative to the ground (i.e. tilt) will be obtained from sensors on the device. The precise type of sensor that will be used is yet to be specified. The tilt sensor may be an accelerometer, gyroscopic sensor, or IR sensor (to measure distance to the ground). Information from the sensors will be fed back to the Z8, which will process the feedback using a crude proportional, integral, derivative (PID) algorithm to generate compensating position control signals to the servo motors in order to balance the device.

Strategy

The Zilog ZNEO Z16 development kit will serve as the microcontroller for the device. Specifically, timers on the board will be utilized to provide the pulse width modulated output necessary to control the servo motors. Feedback from the tilt sensor (or accelerometer, or gyroscope, or IR sensor) will be fed into an ADC converter on the board. The feedback and position control loop will be executed most likely at a time interval, driven by interrupts. In fact, the entire program will be interrupt driven if possible.

External to the Z16, the bot will require two servo motors, a motor control board to source the current for the motors, a tilt sensor (or collection of sensors), a portable battery pack, and a platform upon which to mount the components that will ultimately require balancing.

Physical Construction

The most difficult portion of the design task has been identifying a suitable physical platform for the device. The instructor has indicated on numerous occasions that this project is intended to be an exercise in programming the device, *not* an electrical engineering endeavor. Likewise, I am attempting to select a platform that requires as little assembly or as few modifications as possible, so as to not turn this project into a *mechanical* engineering project. As expected, locating a prefabricated two wheeled platform that is inherently <u>unstable</u> has been a difficult task. Aside from balance-related research projects, a robot prone to tipping has little practical value.

I have narrowed my choice of platforms down to the following three: A lego-based bot, the Nuts and Volts custom balancing bot, and budgetrobotics.com's scooter bot. I have provided reasons why I do not think that the first two platforms will be suitable, but have not yet ruled them out entirely.

Lego-based bot

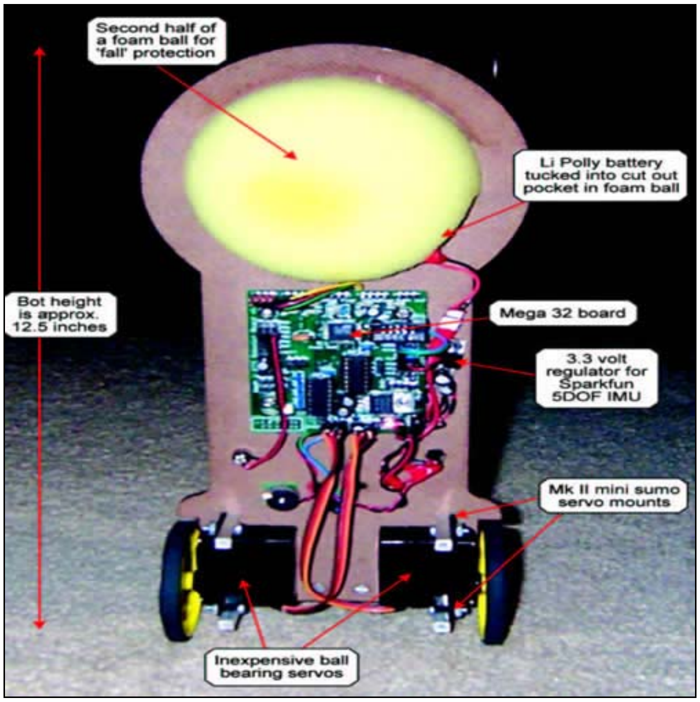
RidgeSoft.com has a complete instruction kit on constructing a self-balancing robot from legos. The problem I see with this solution is the kit utilizes the IntelliBrain controller with an expansion board. The IntelliBrain controller is programmed with a "true" Java development environment, complete with tutorials, example programs, and a rich suite of libraries. Call me a Java critic, but I feel that programming an embedded system with an object-oriented programming language such as Java is being treated just a bit too much with kid gloves. (Although I'm new to the embedded community, I imagine that there are developers who scoff at the fact that we're even able to use a cushy C-compiler rather than coding in assembly directly!).

In addition, the ZNEO controller has a different footprint than the IntelliBrain controller, which would clearly change the dimensions of each of the lego blocks. Clearly, I could scale the assembly to the appropriate size if I had the selection of parts physically before me. However, I feel that choosing a collection of the correct size blocks from an online vendor would prove to be a difficult task. If necessary, I can visit a local retailer of Lego Mindstorm components (such as Toys R Us) to attempt to select the proper parts. A picture of the assembled bot with the Intellibrain controller is below:

Lego BalanceBot. Photo from http://www.ridgesoft.com/buildingbots.htm. For an indication of scale, the topmost component is an enclosure for 6 AA batteries.

Nuts and Volts Custom Balance Bot

A two part article in the hobbyist magazine "Nuts and Volts" describes the construction and implementation of a self balancing robot. The local hobbyist book store (Reiters) did not carry back orders of the periodical; however I was able to identify the issues that contain the article on the Nuts and Volts website. I did not order the back issues, since I did not feel that the information warranted the shipping lead time. (Assuming I undertake solution 3). I was, however, able to obtain the following picture and description of the device:

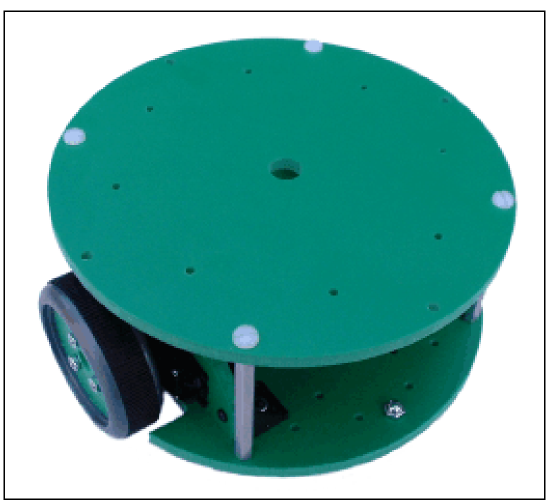


Custom self-balancing robot described in the December 2006 issue of Nuts and Volts. Photo from http://www.nutsvolts.com/toc_Pages/dec06toc.htm.

From the above diagram, the frame of the balance bot is clearly custom made. While admittedly it does not appear all that complicated to reproduce (and access to the magazine articles themselves would likely yield step-by-step instructions), I believe a more suitable, pre-fabricated platform exists. This alternative, however, remains attractive if the proposed solution below does not develop.

Budget Robotics Scooterbot Platform

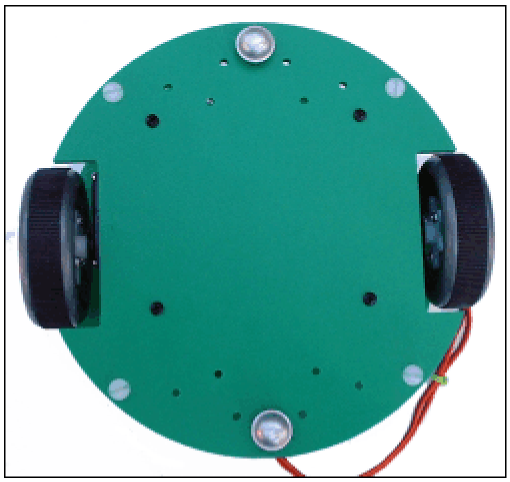
Budgetrobotics.com offers a barebones, wheeled platform capable of supporting a reasonable sized controller and collection of sensors. The base device is shown below:



ScooterBot. Photo from http://www.budgetrobotics.com/shop/?shop=1&cat=35&cart=403971. The diameter of the circular deck is 7”, and the diameter of each wheel is 2”.

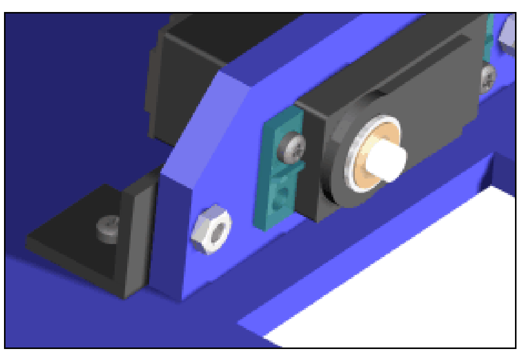
The platform can be extended vertically with riser hardware and an additional deck. One slight complication is that the scooterbot is intended to be a stable platform; whereas a self-balancing robot requires a platform subject to tipping. I have developed a few potential solutions to this problem:

- Forward and aft stability of the device is provided by "adjustable height-balancing skids". It appears that these skids can be adjusted so as to allow tipping of the device in either direction. In fact, these skids can hopefully be used as training wheels of sorts, and be shortened as the balance control of the device evolves. Ultimately, they can probably be removed from the base altogether. A better view of the skids can be seen in the bottom view of the Scooterbot below:



Bottom view of the Scooterbot. Photo from http://www.budgetrobotics.com/shop/?shop=1&cat=35&cart=403971.

- Another possibility is to increase the ground clearance of the base itself. This can be achieved by fixing the servo mounts to the *bottom* of the base, rather than the top. The picture below illustrates the proper orientation of the mount. If the entire base were flipped, the base would ride nearer the top of the wheel, rather than the bottom, providing an extra centimeter or so of ground clearance for the base.

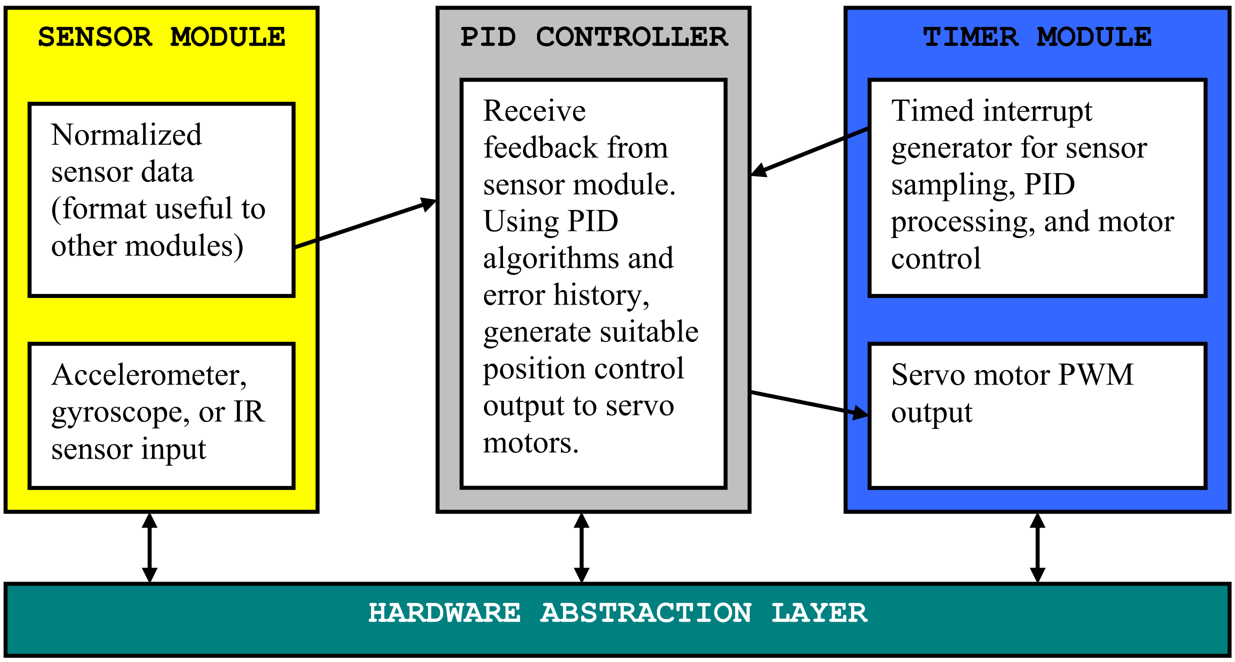


Servo motor assembly. Photo from http://www.budgetrobotics.com/shop/?shop=1&cat=35&cart=403971.

- As a last resort, I can use a band saw to trim some of the area off of the base platform. The base platform can be trimmed to serve as nothing but a mounting point for the servo motors and for the higher level decks.

Software Modules

The following is a preliminary set of software modules:



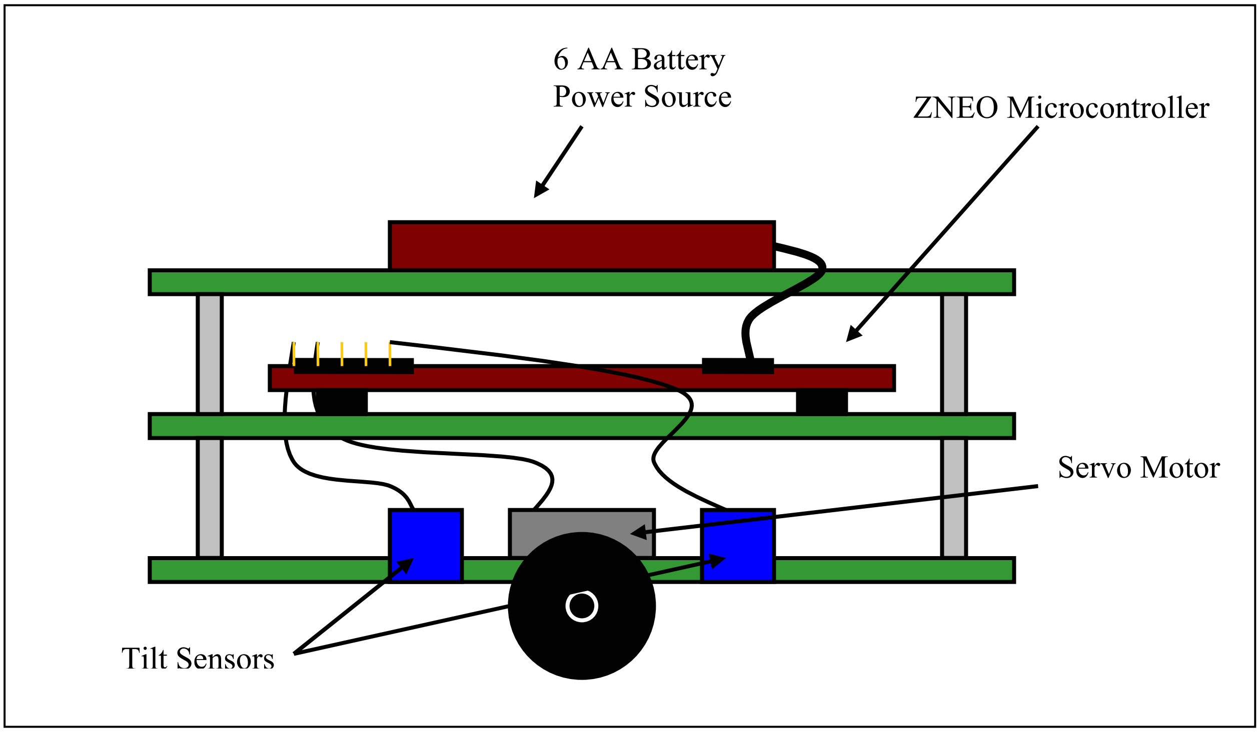
Preliminary Software Block Diagram

The hardware abstraction layer provides a C-application-like interface to the other layers by handling all of the low level details of bit manipulations and register assignments that are particular to the hardware components. The sensor module gathers data from the tilt sensor (be it an accelerometer, gyroscope, etc.) and stores it in a device-neutral format that is available to other modules. The timer module maintains a timer that generates the servo motor control signals and a timer that generates the periodic interrupt at which sensor sampling and feedback processing occurs. The PID controller performs calculations on the current (and past) sensor input and generates position control signals for the servo motor.

Hardware Components

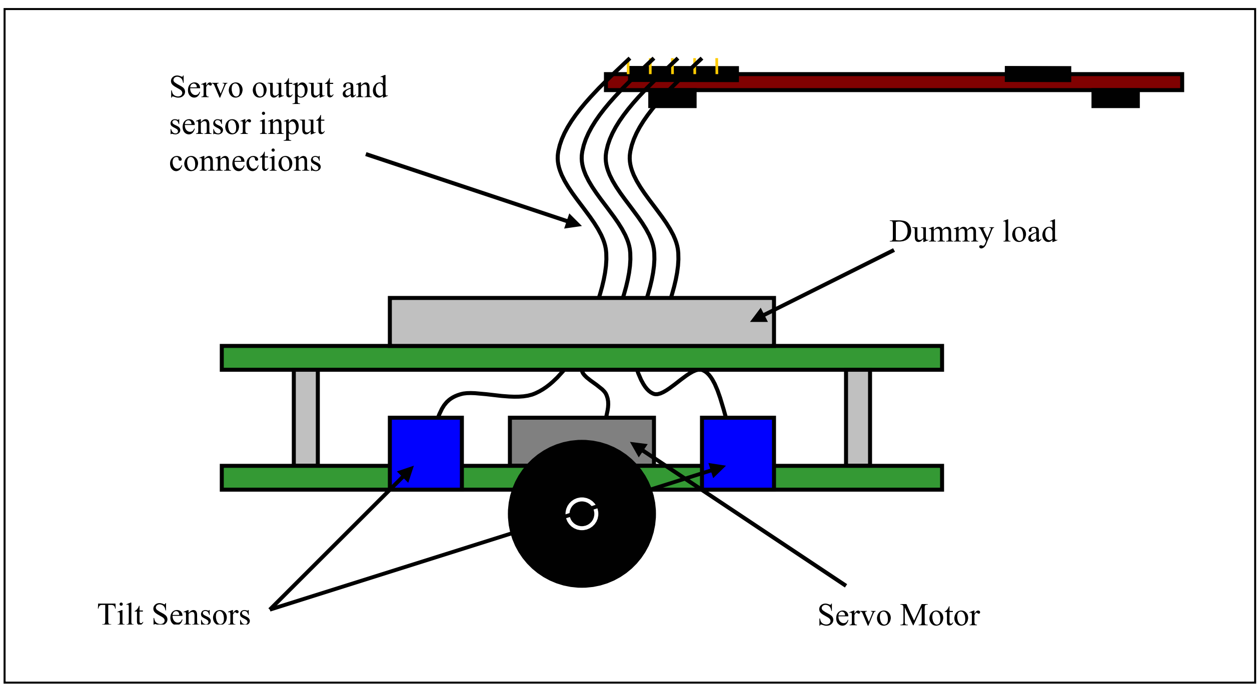
There are two distinct approaches that I have identified regarding the hardware configuration. The first is a design in which <u>all</u> components (controller, sensors, servo motors, and power) are mounted onboard the bot. The power supply (which will most likely be a bank of AA batteries) will reside on the top platform and will serve as the destabilizing load. The second approach mounts only the servo motors and sensors to the bot (in addition to a dummy "load" for mass). The Zilog controller and power source will be located external to the robot, and will connect to the device through a series of cables. The two physical design approaches are illustrated below.

Self Contained Balancing Robot Design



Side view of the self-contained balancing robot design.

Balancing Robot with External Controller Design



Side view of the balancing robot with external controller design.

Unknowns

1. Until further information about the scooter bot is obtained, it is not 100% clear whether it will be a suitable platform for a balancing project. For example, it may be determined that the scooter bot is intended to be a rock stable base, and modifications to inhibit stability will be difficult. This will hopefully be resolved by a phone conversation with a representative from budgetrobotics.com prior to initiating an order.

2. A portable power source for the Zneo board (such as a bank of AA batteries) needs to be identified and purchased. A more thorough internet search needs to be conducted, particularly on the Zilog manufacturer site itself.

3. The precise type of tilt sensor needs to be identified and purchased. The lego robot project mentioned above utilizes an IR sensor to measure the distance to the ground, as does another balancing robot project that was found on a university website. IR sensors need to be briefly researched and a suitable part purchased.

4. A suitable servo motor control board needs to be identified. The instructor has mentioned several in past conversations, and this can be resolved either by a quick email exchange with the instructor or via an internet search and inspection of the data sheets.

5. I was initially worried about the implementation of the proportional, integral, derivative (PID) controller, but after extensive research on the internet I have come across numerous examples and tutorials of embedded PID implementations.

Implementation Plan

The most critical part of the project has progressed from a software issue (i.e. concerns about programming the PID controller) to a mechanical issue – obtaining a suitable balancing platform. The ZNEO controller board has been ordered and is scheduled to arrive on March 18th. The platform itself needs to be ordered, and ideally a conversation with a representative from budgetrobotics.com will answer my questions about the stability of the scooterbot.

The project can progress loosely as follows:

1. Obtain ZNEO microcontroller, scooterbot platform, tilt sensor, battery pack (or other mobile power source), and servo motor control board.

2. Construct scooterbot and ensure that it is susceptible to tipping when an appropriately weighted load is attached. Make any modifications to the base platform as necessary.

3. Experiment with tilt sensor to understand its behavioral characteristics and develop an API that will shape its outputs into usable data.

4. Experiment with the servo motors to understand their operating characteristics. Develop an API that will position the wheels in a simple and reliable manner.

5. Assemble the components in either the “self-contained” or “external controller” configuration as illustrated above.

6. Protect the device from damage during tipping, either by extending the height-balancing skid plates, or affixing some type of temporary "training wheels" type supports. This protection will gradually be minimized (and hopefully altogether removed) as the control logic evolves.

7. Perform simple experiments of driving the servo motors based on input from the tilt sensors. Vary the sampling rate and develop a minimum and maximum interval at which the board can receive feedback, process the data, and send output to servos to compensate.

8. Enhance the feedback loop experiment from step 7 above to incorporate additional factors. Ideally, this step will iterate numerous times to evolve the controller to the point that the robot can balance itself.

Resources

The equipment for the project will be independently purchased.

- Zilog ZNEO Z16 microcontroller (mouser.com)
- Scooterbot wheeled platform with additional deck (budgetrobotics.com)
- Tilt sensor. Most likely an IR sensor (TBD)
- Servo motor control board (TBD)
- Portable battery pack (TBD)