Real-Time PID Motor Controller for a Walking Robot

Ko Ihara¹, Jason Cortell², Andy Ruina³

(1) Cornell University, Ithaca, NY 14853. e-mail: ki38@cornell.edu
(2) Cornell University, Ithaca, NY 14853. e-mail: jbc2@cornell.edu. Lab Manager.
(3) Cornell University, Ithaca, NY 14853. email: ruina@cornell.edu. Project Manager.

Abstract. A PWM-controlled H-bridge circuit board was designed and tested. A PID-based control firmware was programmed onto a digital signal processor (DSP) to control the output of a DC motor through the H-bridge board. Performance data was acquired by the DSP, and sent to a host computer via RS232 protocol. The performance of the controller was then evaluated on the host computer using the received data.

I. INTRODUCTION

The “four-legged bipedal” walking robot designed in the Human Power Laboratory takes advantage of gravity to realize a pendulum-like walking locomotion. While power-efficient in principle, the walking cycle of the current robot is inconsistent and unstable. A major factor that hinders a stable walking locomotion is the low refresh rate of the control variables in the microcontroller.

The main controller on the current robot uses a simple proportional-derivative (PD) control that refreshes at less than 100 Hz. An incentive is to switch the main controller to a faster digital signal processor, and to program a more sophisticated control algorithm. Another incentive towards the design for a more reliable and higher-performance walking robot is an upgrade of hobbyist-level components to high-performance off-the-shelf electronics.

II. ELECTRONICS ARCHITECTURE

The main components of the robotics architectures include the main processing unit, actuators, and feedback sensors. Actuators on the robot may include various types of servos and DC motors. To achieve the amount of force required for the legs to push off the ground, the only actuators used on the current robot are DC motors, driven by an off-chip H-bridge driver. The feedback sensors may include tactile switches, potentiometers, accelerometers, gyroscopes, and quadrature encoder (usually embedded inside a motor). Analog-to-digital conversion or quadrature-decoding of the feedback signals can be processed by the main processing unit, or an off-chip converters and decoders.

![Figure 1: Basic electronics architecture](image)

The signal processing can be handled by a single microcontroller or a digital signal processor (DSP). Ideally, the unit would include such on-chip peripherals as A/D and D/A converters, PWM generator, quadrature decoder, and duplex asynchronous receiver and transmitter. The main processing on the current robot is handled by the Innovation First controller, which is a Microchip PIC18F8530 microcontroller-based system that runs at a core clock frequency of 8 MHz. Unfortunately, this system suffers from a poor temporal resolution of the control: the main control program loop takes over 10 ms to complete, which limits the control refresh rate to under 100 Hz.

III. H-BRIDGE BOARD

The H-bridge driver used to drive the DC motors on the current robot is large, has significant on-resistance. A custom printed circuit board was designed around MT Microelectronics' VN92SP30-E H-bridge integrated circuit. The IC adjusts the amplitude and polarity of the output motor current based on the duty cycle of the input PWM signal, and the values of the two "direction" inputs. Some of the benefits from using VN92SP30 include small surface-mount package, high current capability, simple interface, low on-resistance, and built-in current measurement output. The PCB was designed with EAGLE Layout Editor [2].
IV. DSP-BASED CONTROLS

Core Specifications:

New Micros, Inc's IsoPod is a miniature PCB with a Freescale 56F805 DSP, other required components, and all the headers needed to access the pins on the DSP chip [3]. The on-board crystal oscillator provides an 8 MHz clock, which is multiplied to a 40 MHz core clock frequency by 56F805's on-chip phase-locked loop (PLL), enabled by the high-speed option in the firmware [4].

Programming Environment:
The control algorithm on the 56F805 is programmed in C language, in the Metrowerks CodeWarrior environment. The version specifically tailored for Freescale 56800/56800E family of DSP contains a "Processor Expert" feature, which abstracts an otherwise-necessary manual memory-mapping, and automatically generates various methods for the on-chip peripherals. The compiled program is transferred to the internal flash memory of 56F805 via parallel-JTAG connector.

General-Purpose Ports:

56F805 has 32 general-purpose digital I/O ports, some of which are shared with internal peripherals such as A/D converter and asynchronous serial module. To control a single DC motor through an H-bridge, two digital ports are used to specify the polarity of the H-bridge output.

PWM Generator:

56F805's six independent PWM generators allows an independent control over six separate DC motors, each controlled through an H-bridge board. To control a single DC motor, one of the six on-board PWM module was configured to a PWM frequency of 20 kHz, with a duty cycle adjustable from 0.025 µs to 49.975 µs by 0.025 µs.

Quadrature Decoder:
The quadrature signal generated by the DC motor's embedded encoder is decoded by 56F805's on-chip quadrature decoder. 2048 (2^11) discrete "positions" are detected per single revolution of the motor axle, yielding a spatial resolution of 5.7 positions/degree. There are only 2 independent decoders on the chip, so an off-chip quadrature decoder may be necessary to process all quadrature feedback signals.

Analog-to-Digital Converter
An alternative to using quadrature signals to obtain the angular position of the motor axle is to couple a potentiometer wiper with the motor. The
angular position can be obtained by converting the variable analog wiper voltage. This solution is used by PIC18F8330 on the current robot, but not used yet on the 56F805.

Asynchronous Communication

56F805 has two sets of independent duplex asynchronous serial transmitter and receiver. The baud rate is adjusted with a prescalar constant, and can vary between 305 and 2,500,000 bps. Decreasing the prescalar value can increase the baud-rate, but the mismatch from the conventional RS232 baud-rates would increase. 56F805’s asynchronous peripheral is configured to the baud-rate of 227,272 bps, close to the 230400 bps rate on the conventional RS232 standard. This semester, only the transmitter was used to send quadrature decoder information back to the host computer for analysis. In the future, receiver will be implemented to enable various controls, and on-the-fly adjustment of key variables from a host computer.

PID Control

A traditional continuous time-domain PID control equation is shown below:

\[
Input = K_p \text{Error} + K_D \frac{\partial \text{Error}}{\partial t} + K_I \int \text{Error}
\]

Equation 1: Continuous time-domain PID

Equation 1 was modified into a discrete-time PID equation to be programmed into 56F805:

\[
Input = K_p \text{Err}[i] + K_D (\text{Err}[i] - \text{Err}[i - 1]) + K_I \sum \text{Err}[i]
\]

Equation 2: Discrete time-domain PID

V. EXPERIMENT SETUP

To test the performance of the PID control on 56F805, two methods were programmed to generate a time-based target positions, using the on-chip timer. The experiment uses the angular position feedback from the DC motor’s embedded quadrature decoder to calculate the error from the generated target positions.

In the first test case, the target toggles between the positions of 0 (0 degree) and 2024 (360 degrees) every 500 ms. In the second test case, the target position changes linearly back and forth between 0 and 2024, creating a “zigzag” profile with a period of 500 ms. 56F805’s on-chip asynchronous transmitter was used to send back the measured position and target data every 1 ms. The hardware setup is illustrated in Figure 5.

Figure 5: PID controller test circuit

VI. RESULTS

Figure 6 shows the time-based performance of the PID control for a “step” target profile. As shown in the figure, the system has a rise and fall times of approximately 50 ms to make a 360- or -360-degree turn. The maximum overshoot is less than 1%.

Figure 6: Result from "step" target profile

The PID control’s performance is even better when it is programmed to follow a smoother target profile. For a periodic “zigzag” target profile, the temporal delay of the measured position from the target position is under 10 ms.
VII. CONCLUSION

This semester’s work demonstrates a completion of easy-to-program PID control algorithm on a DSP, which will be used to develop a new generation of walking robots in the spring semester. The project team looks forward to incorporating new gyroscope MEMS sensors to improve the control algorithm for better stability of the walking locomotion.

VIII. REFERENCES


