

FPGA Control of a Mobile Inverted Pendulum Robot

Bikram Adhikari, Deepak Gurung, Giresh Singh Kunwar, Prashanta Gyawali

Department of Electronics and Computer Engineering,
Institute of Engineering, Pulchowk Campus
Email: bikram.adhikari@ioe.edu.np

Abstract: The inverted pendulum is a classic problem in dynamics and control theory due to its inherently unstable nature. In the system tested, Field Programmable Gate Arrays (FPGAs) are used for the implementation of control and sensor fusion algorithms in the inertial navigation system of a Mobile Inverted Pendulum (MIP) robot. Additionally, the performance of digital PID control and Kalman filter algorithms are tested in this FPGA system. The test platform for tuning Kalman filter is designed using optical encoders as a standard reference. PWM signal generation and quadrature phase decoding of encoder pulses is accomplished using hardware description language in FPGA. The values from the inertial sensors and quadrature phase decoded values are fed into MicroBlaze, a 32-bit soft-core RISC processor, within the FPGA. The overall system demonstrates the use of low cost inertial sensors to balance a two wheeled robot. The system is presently able to balance on its own and it also serves as an extremely reconfigurable FPGA based platform to facilitate future modifications, updates and enhancements with more complex control and sensor fusion techniques.

Key Terms: Mobile Inverted Pendulum System, Inverted Pendulum Robot, Inertial Navigation System, FPGA, Kalman Filter, PID Control, Soft-core Processor

1. Introduction

The unstable nature of the inverted pendulum has drawn interests among both researchers of control engineering and robot enthusiasts. There has been significant research in mobile inverted pendulum systems. The human body itself is an example of a MIP system, controlling itself from falling while walking. Two-wheeled self balancing robots, derivations of the MIP system, are characterized by the ability to balance on their own. They possess unique turning and maneuvering abilities which make locomotion in various terrains, including sharp bends, simple and more efficient. Several mobile inverted pendulums have been built using different filtering and control algorithms [1] [2]. This specific implementation of an FPGA controlled MIP robot uses low cost inertial sensors along with Kalman filter and digital PID control algorithms.

2. System Overview

The robot framework (Figure 1), 50 cm in height, is comprised of adjustable plywood stacks used for placement of electronic components. Two motion sensors, an accelerometer and a gyroscope, are used to continuously calculate, via dead reckoning, the orientation of the MIP robot without the need for external references. These two sensors are low cost but their measurements are corrupted with noise. The orientation of the MIP robot must be sensed with higher precision to

prevent it from losing its balance. Hence, this necessitates the requirement of processing the noisy data before it is fed to the control algorithm.



Figure 1. Self-Balancing MIP Robot

The Kalman filter algorithm is one such algorithm that takes into consideration the noise associated with each of the sensors and estimates a best value closer to the actual value [3]. Among all the linear filters, the Kalman filter provides the best estimate and has the least mean square error. Furthermore, the Kalman filter is a recursive algorithm, requiring only knowledge of the previous state and the current measurement for estimation. This leads to lesser memory requirements. A PID control algorithm is implemented for the purpose of balancing the MIP robot. The output of the PID controller is fed into the motors in the form of pulse width modulated signal. Together, the Kalman filter algorithm and the PID control algorithm are implemented on a Xilinx FPGA.

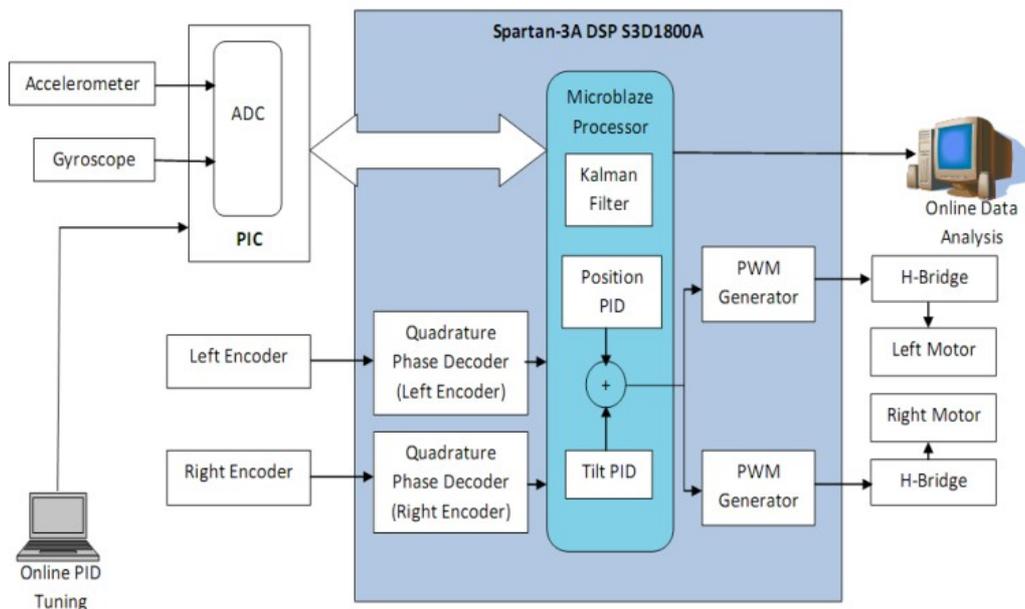


Figure 2. FPGA MIP Robot System Block Diagram

3. Xilinx FPGA and MicroBlaze Softcore Processor

VHDL is a general purpose programming language used to describe the behavior of digital systems ranging from simple logic gates to complete microprocessors and custom chips.[4] This language was used for hardware design and synthesis on the FPGA board.

MicroBlaze is a virtual microprocessor available for use with Xilinx FPGAs. It is operated with Xilinx EDK software that is built by combining blocks of code called cores inside the Xilinx FPGA. The main advantage is that the processors can be designed per requirement. The MicroBlaze processor has a 32-bit Harvard Reduced Instruction Set Computer (RISC) architecture optimized for implementation in Xilinx FPGAs with separate 32-bit instruction and data buses running at full speed to execute programs and access data simultaneously from both on chip and external memory.[5]

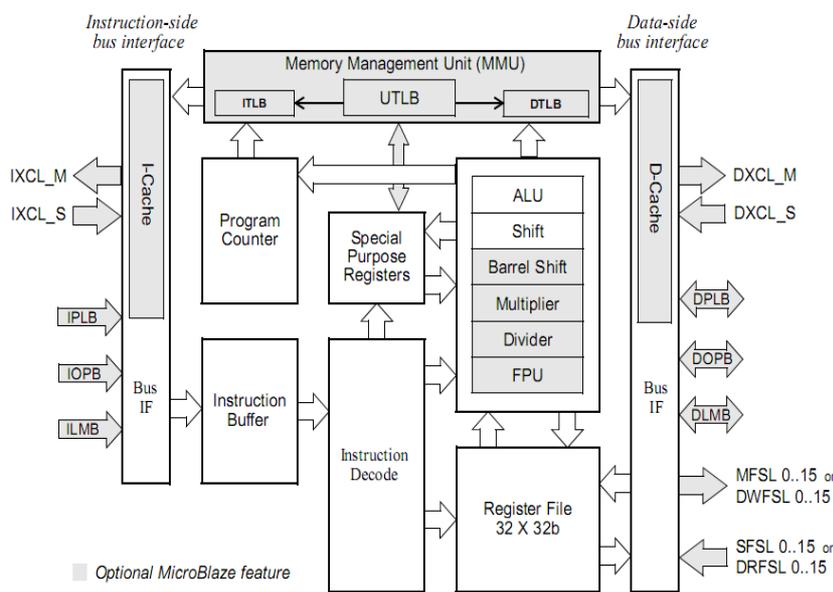


Figure 3. MicroBlaze Core Block Diagram [6]

In Figure 3 the items in the white are the backbone of the MicroBlaze architecture while the items in the gray are the optional features available, depending upon the exact needs of the target embedded application. Because MicroBlaze is a soft-core microprocessor, any optional features not used will not take up FPGAs resources.[5]

MicroBlaze has orthogonal instruction set characteristics. It has thirty two 32-bit general purpose registers and eighteen 32-bit special purpose registers depending upon configured options. The instruction execution is pipelined. For most instructions, each stage takes one clock cycle to complete. Consequently, the number of clock cycles necessary for an instruction to complete is equal to the number of pipeline stages [6].

4. Measuring Orientation of the MIP Robot

The MIP robot must be able to detect its tilt. Since the robot is in motion, inertial sensors can be used to capture the dynamics of the robot. Accelerometer MMA7260 and Gyroscope ENC-03R are the two inertial sensors used.

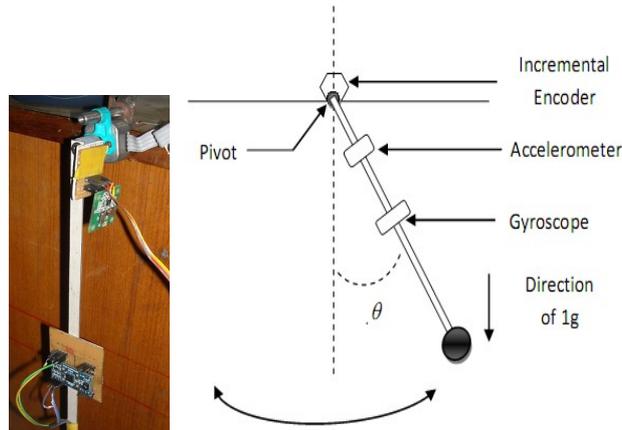


Figure 4. Sensor Calibration Platform

The platform shown in Figure 4 was designed to validate the accuracy of the inertial sensors in measuring tilt. The pendulum is allowed to swing freely under the influence of gravity. As time progresses, the tilt measured using the gyroscope deviates from actual tilt. The accelerometer shows the tilt near the actual value, but is corrupted with high frequency noise.

5. Extended Kalman Filter

The Linear Kalman filter provides an optimal estimate in terms of reduced least square error. In an Extended Kalman filter, the non-linear system's process model is approximated to its linear version by first order approximation around the current state estimate.

Process model:

The process model is a matrix function that estimates the dynamics of the MIP robot. The gyroscope captures the dynamics of the MIP robot and is incorporated in the process model.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \delta \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{\theta}^{gyro} + w \quad (1)$$

Measurement model:

The measurement model relates the output with the state vector.

$$\begin{bmatrix} \theta^{accel} \end{bmatrix} = [1 \ 0] \begin{bmatrix} \theta \\ \delta \end{bmatrix} + v \quad (2)$$

The standard Extended Kalman filter algorithm [7] to estimate the state of MIP robot as described in equation (1) and (2) is:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \dots\dots\dots (3a)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h_k(\hat{x}_{k|k-1}, u_k, u_{EXT})) \dots\dots\dots (3b)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \dots\dots\dots (3c)$$

$$\hat{x}_{k+1|k} = f_k(\hat{x}_{k|k}, u_k, u_{EXT}) \dots\dots\dots (3d)$$

$$P_{k+1|k} = \frac{\partial f_k(\hat{x}_{k|k}, u_k, u_{EXT})}{\partial \hat{x}_{k|k}} P_{k|k} + \frac{\partial f_k(\hat{x}_{k|k}, u_k, u_{EXT})}{\partial \hat{x}_{k|k}} Q_k \dots\dots\dots (3e)$$

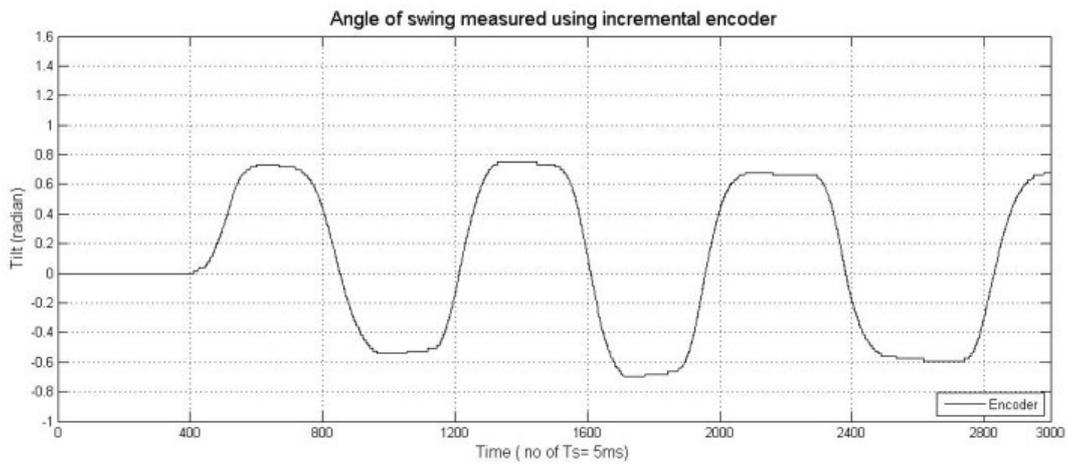


Figure 5. Tilt measure given by incremental encoder

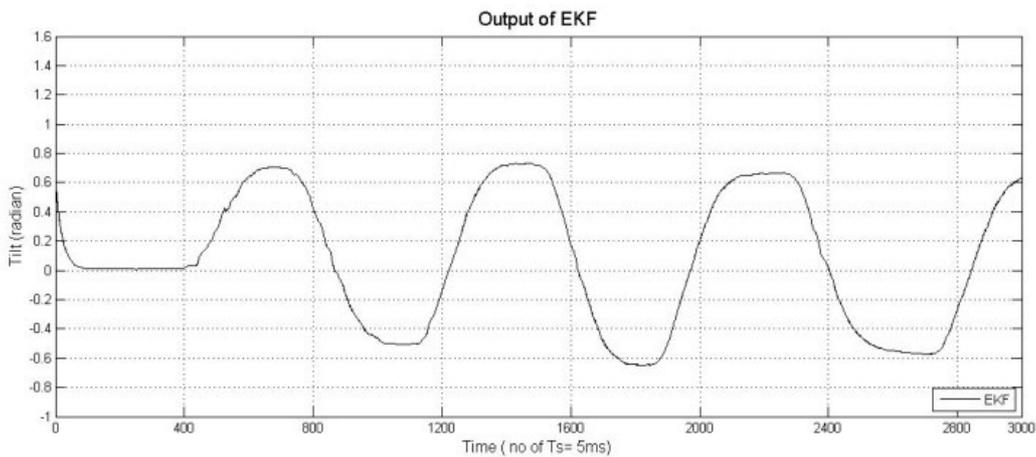


Figure 6. Tilt measure given by Extended Kalman Filter

Figure 5 and 6 verifies that the tilt measured using the incremental encoder and Kalman filter are similar.

PID Control

MIP is a Single-Input Multiple-Output (SIMO) system where a single input force controls both the position and tilt. This model uses two PID controllers connected in parallel. The sum of the outputs of the controllers is the input to the system.

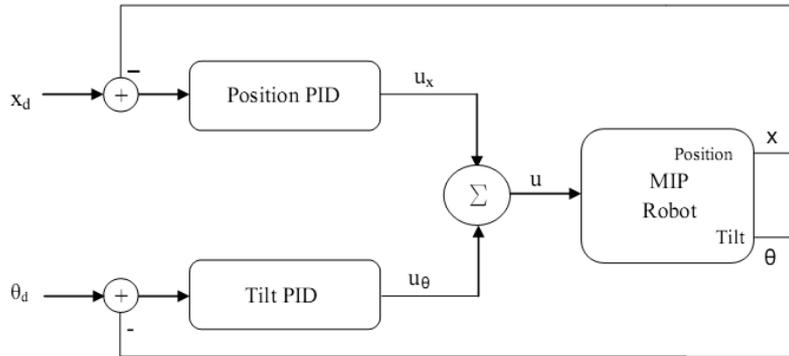


Figure 7. MIP Robot as a SIMO System

There are various structures of digital PID control algorithms. The three basic forms are:

$$u_k = K_p \left[e_k + \frac{T}{T_i} \sum_{j=0}^k e_j + \frac{T_d}{T} (e_k - e_{k-1}) \right]$$

The digital incremental form of PID control is preferred in industrial applications. One advantage with the incremental algorithm is that most of the computations are done using increments only [8]. With this approach it is easy to avoid windup, thus simplifying the use of the algorithm in digital processors. Some special features of this controller are [9]:

- i. Bumpless transfer
- ii. Separate input and output limits, including rate and absolute limits
- iii. Integrator anti-windup (also called a reset-limiter)
- iv. A dead-band on controller output.

The equations for the digital incremental form of a PID controller for position and tilt control are:

$$\Delta u_x[k] = kp_x (e_x[k] - e_x[k-1]) + ki_x e_x[k] + kd_x (e_x[k] - 2e_x[k-1] + e_x[k-2]) \tag{5a}$$

$$\Delta u_\theta[k] = kp_\theta (e_\theta[k] - e_\theta[k-1]) + ki_\theta e_\theta[k] + kd_\theta (e_\theta[k] - 2e_\theta[k-1] + e_\theta[k-2]) \tag{5b}$$

$$u[k] = u[k-1] + \Delta u_\theta[k] + \Delta u_x[k] \tag{5c}$$

where, $e_\theta = \theta_d - \theta$ and $e_x = x_d - x$

The controller gains are adjusted by an iterative trial and error method.

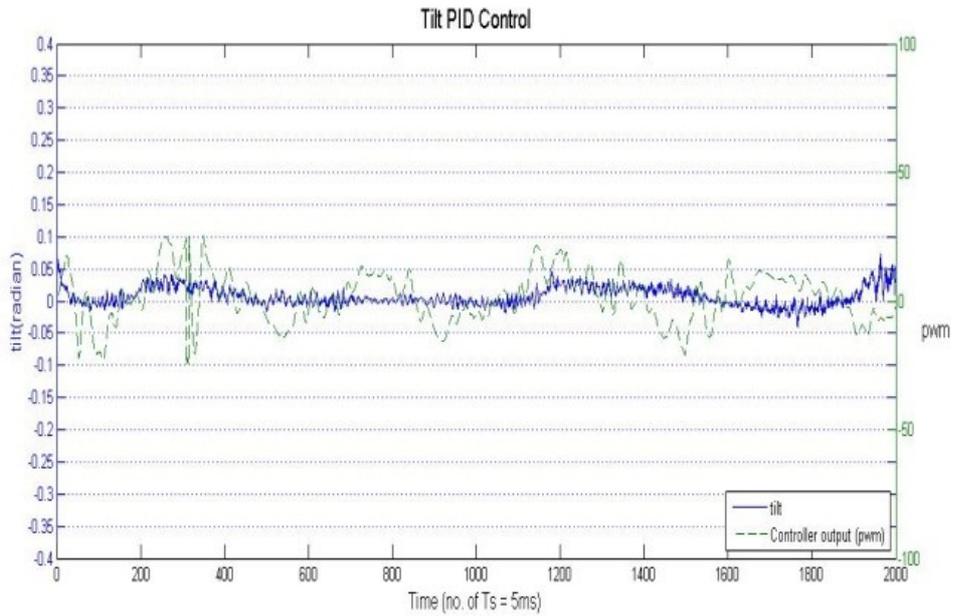


Figure 8. PID Control Velocity Response

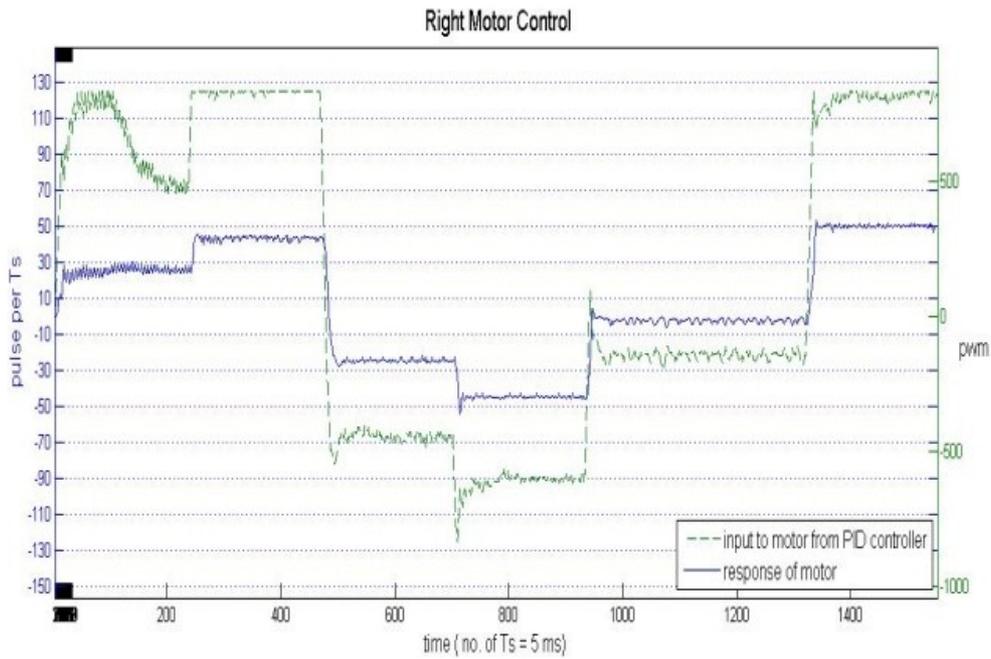


Figure 9. PID Tilt Control Response

Results



Figure 10. Balancing the MIP Robot

The methods and equipment successfully demonstrated balancing of a MIP robot (Figure 10). Figure 6 shows the results from the Extended Kalman Filter implemented in the MicroBlaze soft-core processor. In Figure 8, the response of the PID controller for a dc motor in FPGA is shown. It can be observed that for a large variation in loading applied to the motor, the response of the controller maintains nearly constant motor speed. In Figure 9, the response of the PID tilt controller is shown. The robot at its equilibrium position swings within a vertical deviation of ± 0.05 radians.

6. Conclusions

Hardware and software embedded into a single chip was used to balance a two wheeled inverted pendulum robot. The overall system demonstrated the successful use of low cost inertial sensors. The system is presently able to balance on its own. The complete system is an extremely reconfigurable FPGA based platform to facilitate future work involving more complex control and sensor fusion techniques. It also provides control systems engineers and students of the Institute of Engineering, Pulchowk Campus a basic test platform to implement different types of control algorithms.

Acknowledgement

Special thanks to the Robotics and Automation Research Unit, Robotics Club, Pulchowk Campus, and IOE for providing a research laboratory for this project. Special thanks to Kapil Pudasaini for designing the mechanical framework of the robot. Additional thanks to project supervisors Dr. Shashidhar Ram Joshi and Dr. Arbind Kumar Mishra for their useful comments and help in the successfully completion this project.

REFERENCES

- [1] F. Grasser, A. Darrigo, S. Colombi, and A. Rufer, "JOE: A mobile inverted pendulum", *IEEE Transaction on Industrial Electronics*, Vol. 49, No.1, pp. 107-114, 2002
- [2] Richi Chi Oii, —Balancing a Two Wheeled Autonomous Robot, thesis, University of Western Australia, 2003
- [3] Kalman R. E., "A New Approach to Linear Filtering and Prediction Problems", *Transaction of the ASME—Journal of Basic Engineering*, pp. 35-45, March 1960
- [4] http://www.xilinx.com/ise/embedded/edk_pstudio.htm
- [5] Jesman R., Vallina F.M., Sanjie J., *MicroBlaze tutorial Creating a Simple Embedded System And Adding Custom Peripherals Using Xilinx EDK Software Tools*
- [6] *MicroBlaze Processor Reference Guide Embedded Development Kit EDK 11.4 UG081(V 10.3)*
- [7] Brown, R.G., *Introduction to Random signal Analysis and Kalman Filtering*, John Wiley and Sons, 1983
- [8] Astrom, K. J., Online-<http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf> *Control System Design*, 2002
- [9] http://csd.newcastle.edu.au/book_slds_download/Chapter6.ppt