An FPGA-Based Robotics Platform: Sensing and Control Aspects

Marcel D. A. Siqueira, Júlio Conforto and Marcos B. R. Vallim
Centro de Experimentação Ninho de Pardais
Federal Technological University of Paraná - UTFPR
Cornélio Procópio - PR - Brazil
marceldanilo@gmail.com, julio.conforto@hotmail.com, mvallim@utfpr.edu.br

ABSTRACT

This paper presents the development of three modules of a robotics platform, for processing, sensing and control. The platform includes three main modules: mechanical module, software module and hardware module, being the last one the subject of this paper. The hardware is based on FPGA and described through VHDL, aiming flexibility of prototyping and feasibility of structural changes. We have used a Cyclone II FPGA to base the prototype and a Nios II soft processor to run application algorithms. In order to read analog signals, we have used a PWM reading technique. Based on a Lego NXT motor a mathematical model was calculated and projected PI control modules.

Keywords

FPGA, Nios II, Robotics Platform.

1. INTRODUCTION

Robotics market provides from simple and flexible robotics kits as Lego® Robotics and Fischertechnik® to other more expensive and sophisticated, as Festo® robots [4]. Spread the use of these kits to public schools is a challenge given the fact the its high cost keeps them out of the reality of Brazilian public education [6].

The uses of robotics is seen as an incentive for students to follow science and technological careers. With this concept, the Ninho de Pardais project has the development of a robotics kit as one of its main goals. This project is supported by the Financier of Studies and Projects (FINEP), and is formed by high school and engineering students, professors and a psychopedagogical team [12].

The development of the kit is intended to provide robotics in public schools, while it can be also used as a development platform on research. The kit is designed by electrical and mechanical engineering students, all of them guided by a team of engineering and co-related areas professors.

With a DE2 development board, the platform development makes usage of FPGA (Field-Programmable Gate Array) prototyping, serving two different needs. The use of FPGA allows to find a light structure for a microcontroller, and therefore manufacture the robotic platforms for educational purposes with cost lower than commercial kits. Within the research scope, the usage of FPGA also is interesting because it gives freedom to develop digital circuits dedicated to the robot applications, not limiting the platform performance to the fixed structure of a microcontroller.

This paper presents aspects of the project of the hardware module for the robotics platform proposed for research and educational usage. In the form of a modular kit, it is designed as a tool for robotics workshops with high school students.

The works is presented under sensing and control aspects. Initially, we present the hardware module, showing its processing, input and output levels. Following the same order, the results are presented based on the structure proposed. Finally, the conclusions are made, considered the acknowledgment of contributors and the references are shown.

2. HARDWARE MODULE

The hardware module of the platform was divided into three parts: the processing module, sensing module and motors control module, being the first fully implemented in FPGA and the others with hardware parts divided into internal and external to the FPGA. Figure 1 shows this modular structure. The reconfigurable hardware parts are described through VHDL (VHSIC Hardware Description Language, VHSIC - Very High Speed Integrated Circuits). This section is devoted to show the processing module, the sensing technique and the motor control module, respectively.

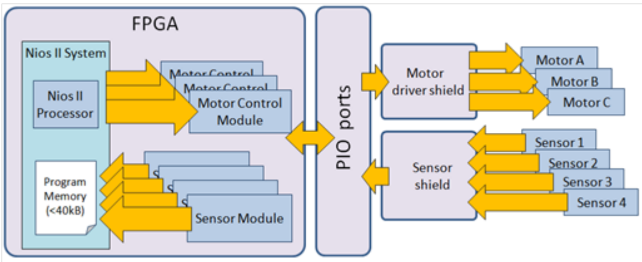

Figure 1: Hardware structure of the platform

2.1 Processing Module

In order to process sensor signals and control motors based on a user algorithm, a processor was employed. The processing module comprises a system with a traditional arrangement of components similarly to what is within a microcontroller.

The virtual processor Nios II is a component that can be parameterized and generated by the software Qsys [1]. It

SForum 2012 - Student Forum on Microelectronics
This work has been developed by the first author(s) in the scope of the undergraduate studies

is a 32-bit RISC general-purpose core processor. A Nios II system is equivalent to a microcontroller, which contains a processor, memory and a set of peripherals [2].

The Qsys contains a graphical interface through which allows to choose the desired components for a Nios system, such as processor units, audio and video interface, serial and parallel ports, etc. Once defined the set of components, Qsys generates a Nios system in the form of logical component, which can be instantiated in a design of reconfigurable logic in Verilog, VHDL or graphical blocks. Hence, the generated Nios system is instantiated in a Quartus II project, a development environment for reconfigurable logic circuits.

For Nios II system programming, the adopted tool was the Nios II Software Build Tools (SBT), which contains an Eclipse-based environment for Nios II systems programming and debugging. The environment combines compiler for C and C++, and implements all the steps of communicating between the DE2 and the PC.

2.2 Sensing Module

Any robot needs sensors to deduce what is happening in its world intending to be able to react to various situations [5]. Both the FPGA and the development board have no analog/digital converters (ADC), so it was necessary a strategy for reading analog sensor signals.

The adopted idea is the same used to drive the motors with PWM signal, but in the reverse flow. As the FPGA can only read the high and low logic levels, it is proposed that an analog signal modulates a PWM wave, thus it can be read by the FPGA.

A light sensor, for example, can provide a variable voltage level corresponding to the luminous intensity. This voltage level can modulate the duty cycle of a PWM wave. Finally, the FPGA can read the PWM wave signal modulated according to the light sensor. This idea was adopted, as Figure 2 shows. In this illustration, a variable signal modulates a PWM waveform. The strategy for the modulation is based in the comparison of the voltage generated by the sensor transducer to a triangular wave pattern. Adjusted with infinite gain, a comparator generates a square wave with variable duty cycle, a PWM wave. After decoupling voltage, the FPGA may interpret the analog signal.

The FPGA is able to read cyclically profile a PWM wave profile at high speed and controlled time. The beginning of a read cycle occurs on a positive edge. At this moment, the process checks the wave's logic state and increments a variable 'high' until the wave toggles to low level. Thus, the module switches the increment to a variable 'low', until the moment it detects a new rising edge. At this point, the process has read a full period of the PWM signal, and a mathematical operation of division sets its duty cycle.

2.3 Control Module

The DC motors are often used in robotics platforms. In this work, we developed a speed and position controller based on a Lego Mindstorms NXT® motor. This motor is made in its internal structure by a gear box, an incremental encoder and a DC motor [9], with saturation limit of 1 A to supply current and average voltage excursion between 0 and 9 V.

Within this context, we designed a power driver to control three motors. Its structure enables the motors to operate in the four quadrants and allows input and output signals isolation and conditioning.

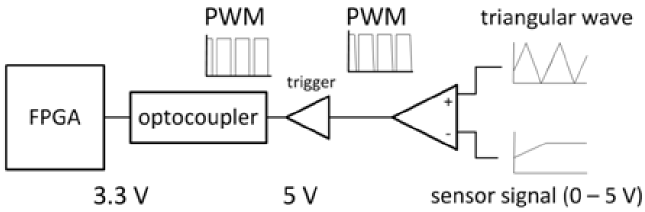

Figure 2: Analog to PWM signal conversion process

Table 1: Lego NXT® Motor parameters

Parameter	Symbol	Value
Armature resistance	R_a	7.6 Ω
Armature inductance	L_a	0.00488 H
Constant V/ω	K_m	0.3233 Nm/A
Back-efm	K_b	0.4908 Vs/rad
Shaft inertia moment	J	0.075 kgm^2
Viscous friction coefficient	B	0.00192 kgm^2/s

2.4 Modeling for the Lego Mindstorms NXT® Motor

For purposes of modeling and control, the equations are represented in the frequency domain, applying the Laplace transform [3].

In order to control the motor position and speed, we considered the parameters of the mathematical model of the LEGO motor suggested by [8], which are shown in table 1.

The mathematical model of the DC motor to the position system is deduced by means of a linear approximation of the actual engine [10]. Both equation 1 and 2 represent the model of the DC motor.

Through the derivation of equation 1, the equation 2 was found as a representation for the speed modeling.

$$G(s) = \frac{\theta(s)}{V_a(s)} = \frac{K_m}{s[(R_a + L_a s)(Js + B) + K_m K_b]} \quad (1)$$

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_m}{(R_a + L_a s)(J.s + B) + K_m.K_b} \quad (2)$$

Both models use the voltage $Va(s)$ as input. The output of the first and second model are given by the angular position $\theta(s)$ and the angular speed $\omega(s)$, respectively.

As reference for the system model are used position (radians)

Table 2: Comparison of theoretical to experimental values read by the sensing module

duty cycle	theoretical values	real values
0%	0	0
20%	205	204
40%	409	411
60%	614	613
80%	818	817
100%	1023	1023

and angular speed (rad/sec) values, so these units must be converted to voltage values. To convert the speed value, the voltage is divided by the angular speed, and for position, is made by the ratio of the voltage with 2π.

2.5 PI Controller

In order to perform the robot movements with better accuracy a PI controller was chosen, which reduces the error in steady state, characterized as a phase lag compensator [10].

In a PI controller, the relationship between the output $U(s)$ and the input $E(s)$ is given by the transfer function of equation 3.

$$\frac{U(s)}{E(s)} = K_p + \frac{K_p}{T_i s} \tag{3}$$

where K_p is the proportional gain and T_i is the integral time. The parameters K_p and T_i are called tuning parameters of the controller [10].

3. RESULTS

This section shows results on hardware development following the same order used on methodology presentation.

3.1 Validation of Sensing Modules

For validation of sensing technique we built an approach with a light sensor modulating a PWM wave at frequency of 230 Hz. As a sample reading time is equal to a single period of the PWM signal, the sampling time was 4.34 ms.

At testes stage, the slow edges of the PWM signal generated reading errors. To fix it, a trigger was added to decrease the rising and falling times of edges of the PWM-modulated wave. The slowest time fell down from 164.0 μs to 3.024 μs, improving greatly the reading done by the FPGA.

The test based on these parameters is summarized in Table 2, comparing the theoretical and practical measurements of PWM signals performed by the FPGA. The value of the duty cycle was varied from 0 to 100% and converted into a 10-bit binary array, which was connected to a Nios system's parallel port, read through the Nios II SBT console. For applications in educational robotics, where sensor values are usually in the range 0-100, the results are satisfactory. Hence the proposal to read analog signals with the FPGA via PWM wave was well succeeded.

3.2 Implementation of PI Controllers

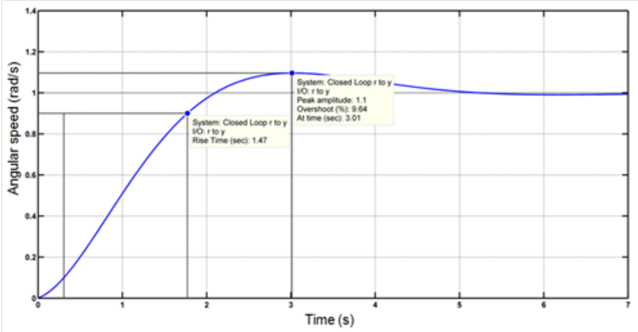

Figure 3: Step response of control action for angular speed

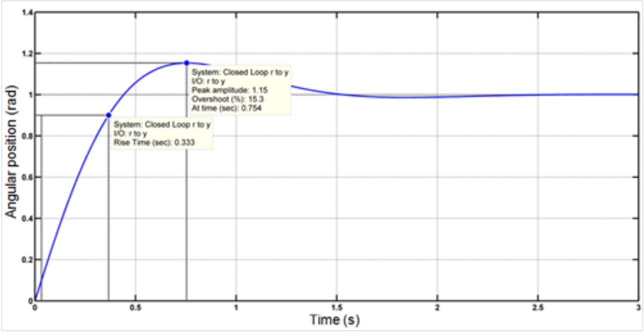

Figure 4: Step response of control action for angular position

To estimate the parameters of the proposed controller, we choose the root locus method through the determination of the parameters K_p and T_i. The root locus method is chosen because it presents good analysis for slow response systems. In addition, this method provides visualization of response in the time domain [10].

Using the mathematical model and the speed controller tuned to gains $K_p = 0.120$ and $T_i = 0.012$, there was obtained a settling time of 3.3 seconds and overshoot of 9.64% to a step input. Figure 3 shows the response to a unit step of the compensated system.

For position analysis, the mathematical model and the position controller tuned to the gains $K_p = 0.850$ and $T_i = 18.88$, we found a settling time of 1.4 seconds and about overshoot of 45.4% to a step input. Figure 4 shows the response to an unitary step of the compensated system.

In order to implemented the control action on digital hardware, it is required to discretize the controller model [7]. The discretization of the PI controller for this proposal was made applying the trapezoidal approximation for the integrative portion.

$$s = \frac{2(z-1)}{T_s(1+z)} \tag{4}$$

Replacing the term of equation 4 into equation 3, the equa-

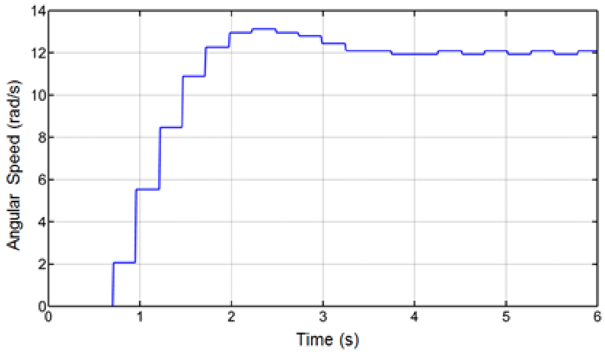

Figure 5: Real step response of control action for angular speed

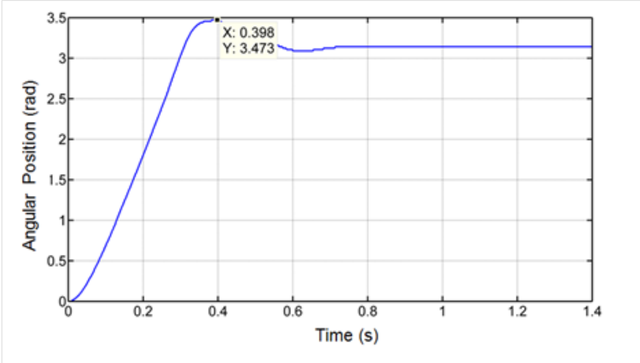

Figure 6: Real step response of control action for angular position

tion 5 is generated, with the transfer function in the complex domain.

$$\frac{U(z)}{E(z)} = K_p\left(1 + \frac{T_s(1+z^{-1})}{2T_i(1-z^{-1})}\right) \quad (5)$$

The difference equation is formulated in equation 6.

$$u[n] = u[n-1] + a(e[n] - e[n-1]) + \frac{b}{2}(e[n] + e[n-1]) \quad (6)$$

where $a = K_p$, $b = K_p(T_s/T_i)$.

This equation of the controller has three terms: the previous signal from the actuator, the proportional gain multiplying the difference between the current and previous error and the average of the integrative gain multiplying the sum of the current and the previous error.

The sampling time (T_s) used for position and speed systems were respectively 50 ms and 250 ms.

To position control the calculated values for the constants a and b were 0.850 and 0.95 respectively. In the speed controlling the calculated values for the constants a and b were 0.120 and 0.300 respectively.

With these values, the controller was implemented in FPGA. Figure 5 shows the real response of the speed controller. The settling time of the real system was 2.7 seconds, while the real maximum overshoot was 9.41%.

Figure 6 shows the step response for the angular position, where the settling time was 2.0 seconds and overshoot 5.37%.

4. CONCLUSION

This paper presented the methods and achieved results about the development of a FPGA-based robotics platform. For development usage the results for the PI control module are satisfactory. However the sensing module must be improved intending its application for some systems. The Nios II using proved be suitable and easy to use, but requires the license purchase if used in commercial products, therefore the project must include the migration to open source processors, as [11] suggests.

5. ACKNOWLEDGMENTS

This work is supported by the Financier of Studies and Projects - FINEP (Ref. 4971/2006), Federal Technological University of Paraná (UTFPR), and National Council for Scientific and Technological Development - CNPq Brazil.

6. REFERENCES

[1] Altera. *Quartus II Handbook Version 12.0.*

[2] Altera. *Nios II Processor Reference Handbook*, 2009.

[3] R. C. Dorf and R. H. Bishop. *Sistemas de Controle Modernos.* LTC, 8th edition, 2011.

[4] S. Enderle. The qfix robot kits. In *1st International Conference on Research and Education in Robotics*, pages 84–95, May 2008.

[5] P. O. F. G. Martin and R. Sargent. *Robot Builders Guide.* MIT Press, 1992.

[6] D. A. M. Filho and P. C. Goncalves. Robótica educacional de baixo custo: uma realidade para as escolas brasileiras. In *XIV Workshop sobre Informatica na Escola - WIE*, Jul 2008.

[7] J. P. G. Franklin and M. Workman. *Digital control of dynamic systems.* Addison-Wesley, addison-wesley world student series edition, 2010.

[8] J. Goncalves, J. Lima, P. Malheiros, and P. Costa. Sensor and actuator stochastic modeling of the lego mindstorms nxt educational kit. *10th International Conference on Mobile Robots and Competitions*, 1:35–39, 2010.

[9] Y. Kim. Control systems lab using a lego mindstorms nxt motor system. *IEEE Trans. on Education*, 54:452–461, 2011.

[10] K. Ogata. *Engenharia de Controle Moderno.* Pearson Prentice Hall, 5th edition, 2010.

[11] OpenCores. Open source for hardware?

[12] M. B. R. Vallim, A. Herden, R. Gallo, L. R. Cardoso, and L. C. Bitencourt. Incentivando carreiras na Área tecnológica através da robotica educacional. In *37 Congresso Brasileiro de Ensino de Engenharia - COBENGE*, 2009.