

RECONFIGURABLE ARCHITECTURE PROPOSAL APPLIED TO MOBILE ROBOTS

Carlos Raimundo Erig Lima
Laboratório de Inovação e Tecnologia em Sistemas Embarcados – Centro Federal de Educação Tecnológica do Paraná. Curitiba, Brasil. carlos@lit.cpdtt.cefetpr.br.

João Maurício Rosário
Labóratório de Automação Integrada e Robórica – Universidade Estadual de Campinas. Campinas, Brasil. rosario@fem.unicamp.br

Abstract. *Due to the fast innovation speed in the hardware and software technologies in the field of embedded systems, it becomes more and more necessary to develop applications based on methodologies, which take into account the easiness of future modifications, updates and improvements in the designed system. According to these requirements, this work presents a proposal of environment using reconfigurable computing applied to mobile robots prototype design. The developed software and hardware are structured in independent blocks, through open architecture implementation, allowing the easy expansion of the syste m, better adapting the mobile robot to the tasks associated to it.*

Keywords. *Open Control Architecture, Reconfigurable computing, Mobile Robots, Embedded Systems.*

1. Introduction

The mobile robots, as platforms for knowledge consolidation in several teaching and research areas, such as modelling, control, automation, power systems, sensors, data transmission, embedded electronics and software engineering, are more and more used in teaching and research institutions.
This work presents a proposal of environment to simplify the embedded systems prototypes design, in particular, mobile robots prototypes, using simple tools to control the several sensors and actuator usually present in this project class. Through the proposal a hierarchical architecture, distributing the several control actions in growing levels of complexity and of the use of resources of reconfigurable computing (Miyazaki, 1998; Compton, 2002) is possible the validation of this it sets an experimental mobile robot. The topic of open system architecture has been studied for a few years now by several institutions in the machine tool and production engineering fields, looking at all the aspects of modularity as well as the effects of the controller architecture and the communication network on the system performances (Pritschow *et al*., 1993; Koren *et al*., 1996). The objective of this reconfigurable architecture concept is thus to enable an easy and quick adaptation of a mobile robot structure to these technological evolutions, for a better portability and interchange ability of the final system. Through the division of the structure in small functional blocks, with very specific dedicated interfaces, the modularization of the project becomes efficient (Lima *et al*., 2000; Lima and Rosario, 2001). This allows first the best specification of the development tasks for a multidisciplinary team of people, then the adaptation of a particular block to a new technological evolution. The proposal and development of this open and generic architecture aims at supplying this need, having as an emphasis, the control structuring, the supervision and the transfer of information. Section 2 describes the general adopted structure and its different modules. Section 3 presents an application of this general structure for a practical mobile robot application. Section 4 presents an example of controller implementation using reconfigurable logic.

2. Proposed Environment

The proposed environment is a set of hardware and software modules, implemented with emphasis in reconfigurable logic, integrated to give support to development of mobile robots prototypes. The architecture, under the mobile robot control point of view, is organized in several independent blocks, connected like a hierarchical structure, as shown in Figure (1), where can be visualized three control levels. The level can be described as follows:

- **Supervisory control:** In this high control level, the supervision of one or more mobile robots can be carried out, through the execution of global control strategies. This level also allows managing mobile robot tasks, to establish corrections in the task realization according to the sensors data fusion information, or modify for example the required information for the local control level.

- **Embedded control:** in this level, the mobile robot embedded software processes control. The control strategies allow decision making to be performed at a local level, with occasional corrections from the supervisory control level. Without communication with the 'supervisory control level', the mobile robot only carries out actions based on obtained sensors data and on information previously stored in its memory. The transfer of data between this level and the supervisory level is achieved through the use of data structures implemented in a shared memory, which allows a better treatment of the transmitted and received data. Through the use of the same resource of shared memory, the tasks of this level can be divided among micro-processors and programmable logic devices (PLD).

- **Local control:** This is restricted to strategies of local control associated with the interfaces of the sensors and actuators (Borenstein *et al.*,1996). The strategies in this level can be implemented in reconfigurable logic, through PLD. The local controllers may be implemented under difference equations (coming from the so-called RST form), which appear to be a very general and useful structure in an open architecture environment, including classical PID controllers (Aström and Wittenmark, 1997) as well as more advanced control techniques such as predictive control for instance (Clarke and Tuffs, 1987; Soeterboek, 1992; Dumur and Boucher, 1994; Dumur and Boucher, 1996).

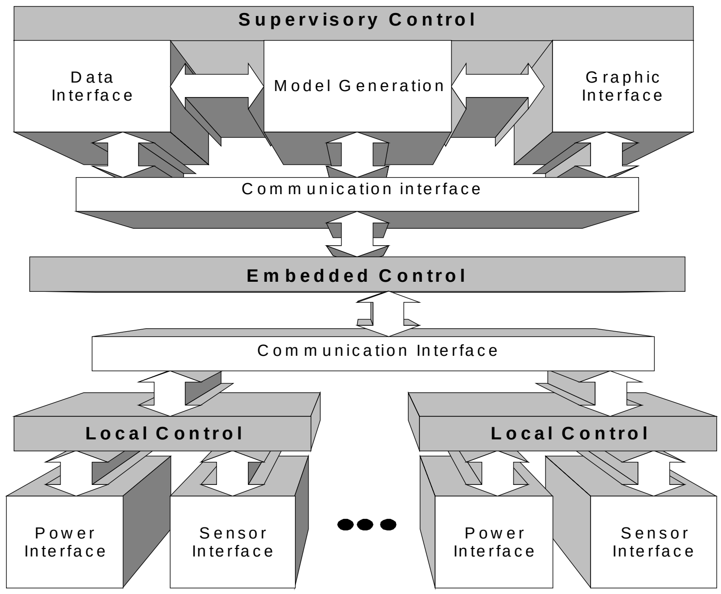

Figure 1. Hierarchical structure distributed in three control levels.

3. Practical Application

This previous architecture is in fact sufficiently generic to be used with several models of mobile robots (Farritor and Dubowsky, 1999; Iagnemma and Dubowsky, 2000). As an application of this architecture, a platform with a differential drive configuration is proposed here, as suggested in (Borenstein, 1995). The Figure (2) presents a simplified model of this platform and the real prototype developed. The encoders are directly coupled to the motor axis, allowing a larger resolution of measurements from odometers. The collision sensors are distributed to cover the probable areas of collision. A great number of sensors can be used however, but the following types have been considered in this prototype:

- Position sensor: Two encoders are coupled to the traction axis of the robot. They are responsible for the registration of the odometers as well as the determination of the displacement of the robot axis.
- Collision sensor: It is responsible for the distance to obstacles measured in front and on the side of the mobile robot.
- Compass sensor: It is responsible, with the information of the encoder, for the determination of the angular displacement of the robot.

The Figure (3) details the proposed architecture applied to prototype developed. The supervision level is built around a graphic interface under Windows System, in a PC microcomputer. The communication link between this level and the embedded level is implemented through a Radiometrix transceiver (Radiometrix, 2002) with data rate up to 32 kbits/s. In the embedded control level two main blocks are implemented: the command decoder block and the logic control block. The first block decodes commands received by the embedded communication interface, allowing different actions in function of received data. The second block generates control signs to actuators interfaces and receives signs from sensors interfaces located in local control level. Control strategies are implemented in logic control block. The local control level is present in Figure(4), which includes all required interfaces with the sensors and actuators, and the controller of each motor. In an open architecture point of view, the implementation of these local controllers must be general enough to enable an easy reconfiguration of the control law according to the user specifications.

To facilitate tests, implementation and future modifications, the use of PLD in the implementation of the interfaces and other logical blocks is greatly emphasized in our environment. In fact, development systems based on PLD present features adapted to this kind of systems. The motor control part, for example, can benefit of the characteristics of low consumption, high-speed operations, integration capacity, flexibility and simple programming (Miyazaki, 1998).

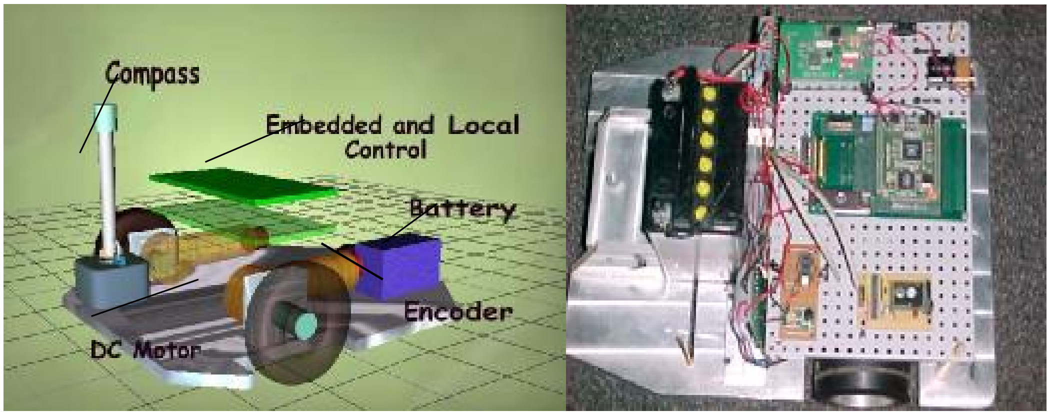

Figure 2. A model representation of mobile robot and the real prototype developed.

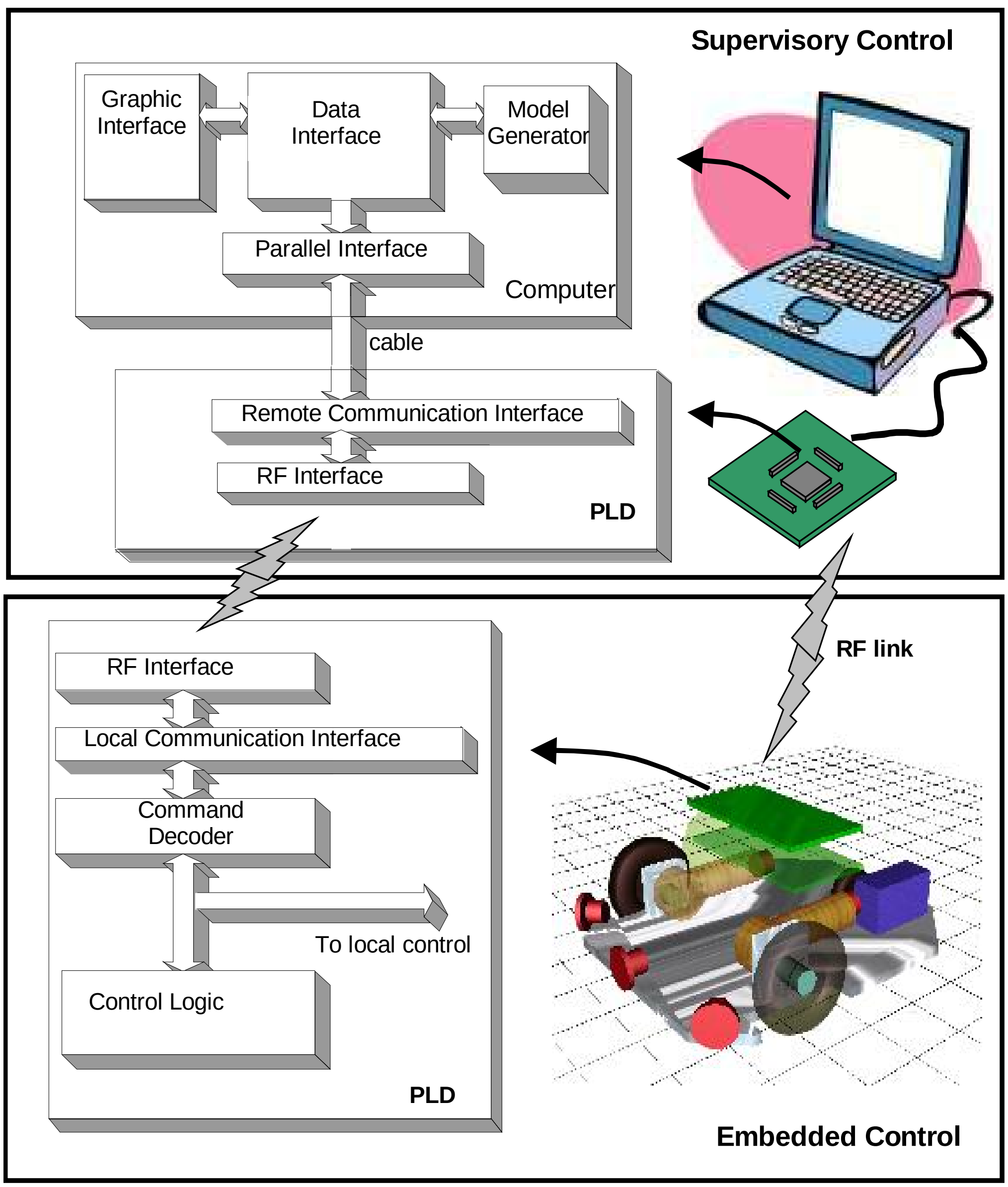

Figure 3. Block diagram representation of proposed architecture applied to prototype developed.

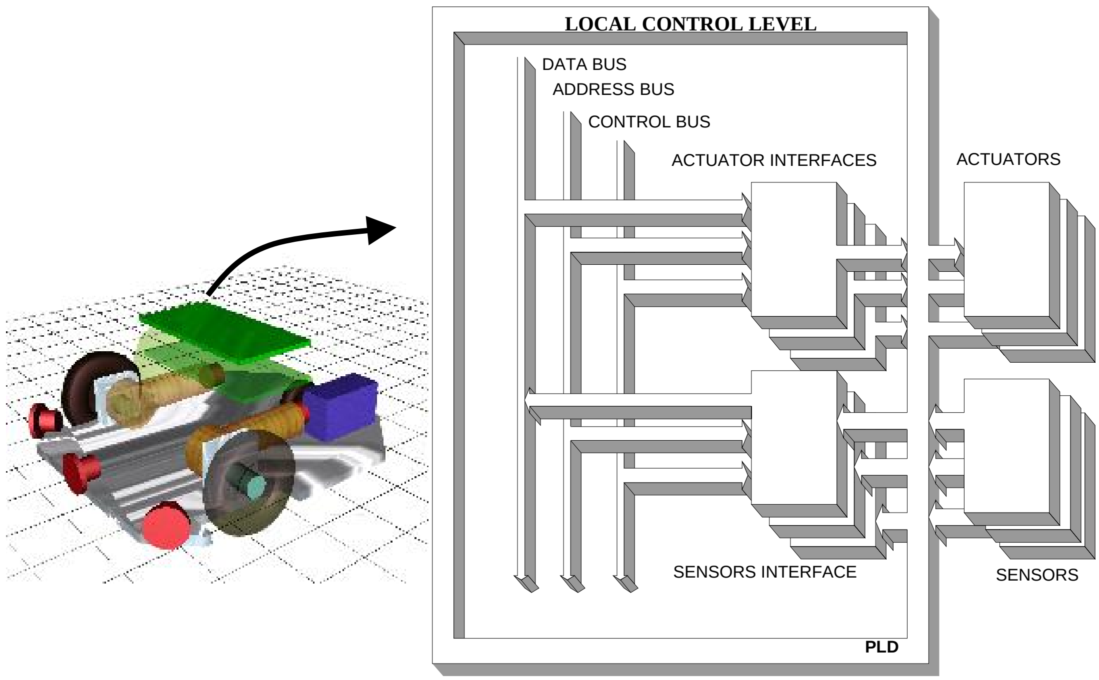

Figure 4. Local control level representation in prototype developed.

The synthesis of the logical circuits in PLD is made through CAD (Computer Aid Design) systems, allowing the use of different project interfaces. Examples of interfaces languages used are: graphical (through schematic), VHDL (VHSIC Hardware Description Language) and AHDL (Altera Hardware Description Language). Several CAD systems and PLD devices vendors disputes the world market (Altera, 2002; Atmel, 2002; Actel, 2002; Accolade, 2002; Mentor, 2002; Synopsys, 2002). This prototype uses graphical interface and VHDL interface, taking advantages of each method. All blocks were projected using Altera MAX+PLUS II development tool (Altera, 2002), under Altera University Program. Examples using the described interfaces will be presented in the next sections.

An implementation with graphical language can be visualized in figure (5).

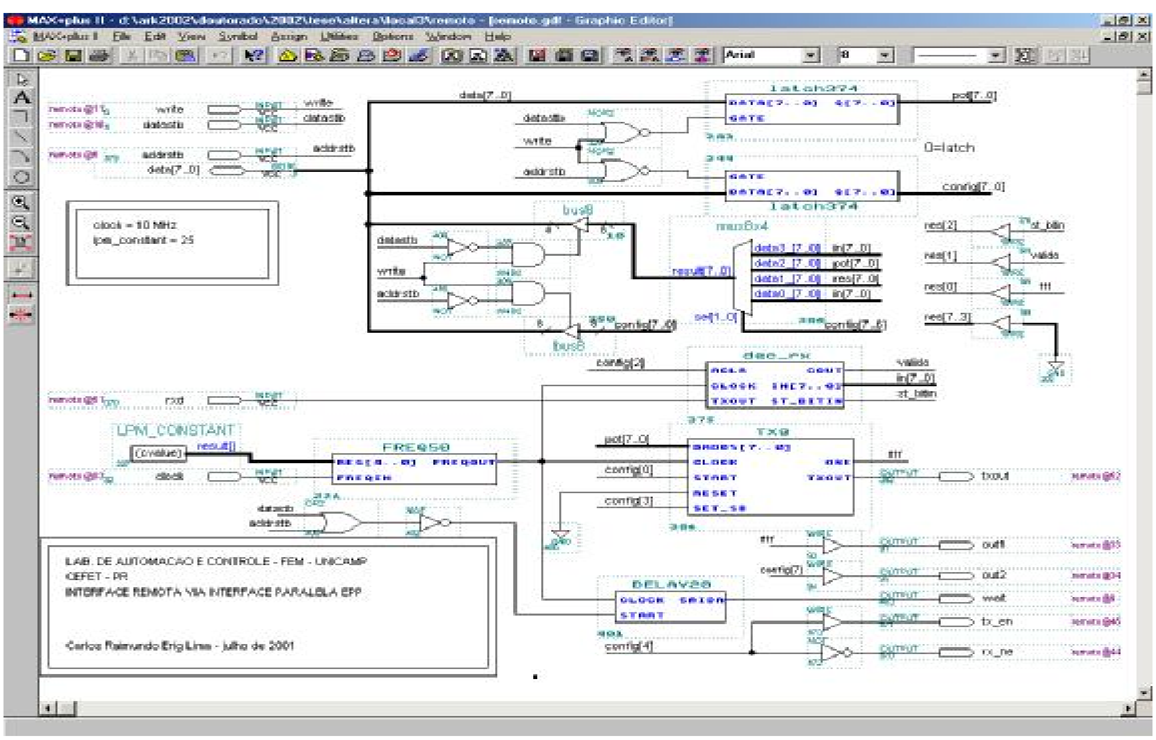

Figure 5. Block implementation with graphical language.

4. Using Reconfigurable Logic to Implement Controllers

An alternative to controllers implemented by software is reconfigurable logic. The controller proposed has as objective the control of mobile robot DC motors. This programmable controller is capable to process the digital signs originating from an encoder coupled to the motor (sign ENCODER) and of a digital sign of control representative of a trajectory (sign TRAJETORY). In PLD, a PID digital controller is implemented. This PID has its gain parameters fitting through external programming. The controller's output is a digital sign for the PWM potency block. Digital-analog or analog-digital converters are not necessary in the control loop.

The implemented controller can be visualized in the diagram of blocks presented in the Figure 6. In this figure, the control of an only motor is represented, but the synchronized control of whole motors can be made easily through the same PLD.

Four main blocks are observed:

- Error Detecting Block: This block is used for the comparison of the signs ENCODER and TRAJETORY, allowing the generation of a proportional binary word to the error among the periods of the signs.
- PID Controller Block: it implements a PID digital controller, using the gain parameters contained in the control registers.
- Control Register Block: it implements the control registers, responsible for the programming of parameters in PLD.
- Power Interface Block: it converts the binary word supplied by PID controller in a pattern of digital signs to control of the PWM potency block.

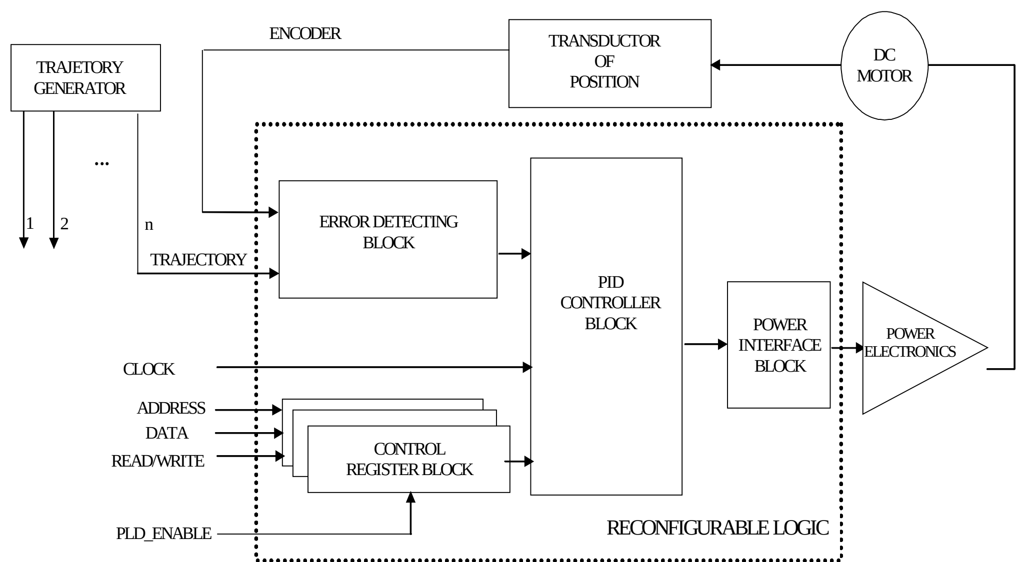

Figure 6. Diagram of blocks of the system implemented in PLD.

An alternative for a digital PID equations is to use libraries built for algebraic manipulation in VHDL. Several works can aid this project class (Klotchkov and Pedersen, 1996; Kolling *et al.*, 1998; Samet *et al.*, 1998). A digital PID controller can be represented by the following expression (Samet *et al.*, 1998):

$$u[k] = u[k-1] + e[k](kp+kd+ki) + e[k-1](ki-kp-2kd)+ e[k-2]kd. \quad (1)$$

The equation (1) is represented in Figure (6) in block diagram form and in Figure (7) in graphical language. The register error block stores values of e[k], e[k-1] and e[k-2], and makes shift operations (e[k-1] = e[k] and e[k-2] = e[k-1]). The output register block stores u[k] and u[k-1]. A example of the PID Controller simulation Interface under Altera MAX+PLUS II development tool can be visualized in figure (7).

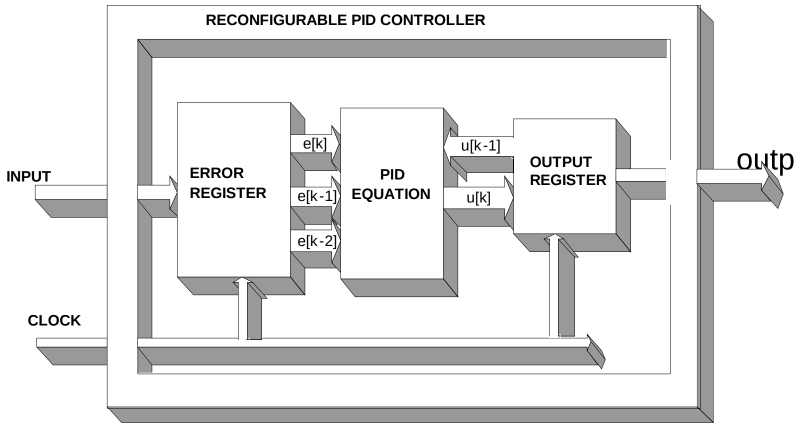

Figure 6. Digital PID controller block implemented in reconfigurable logic.

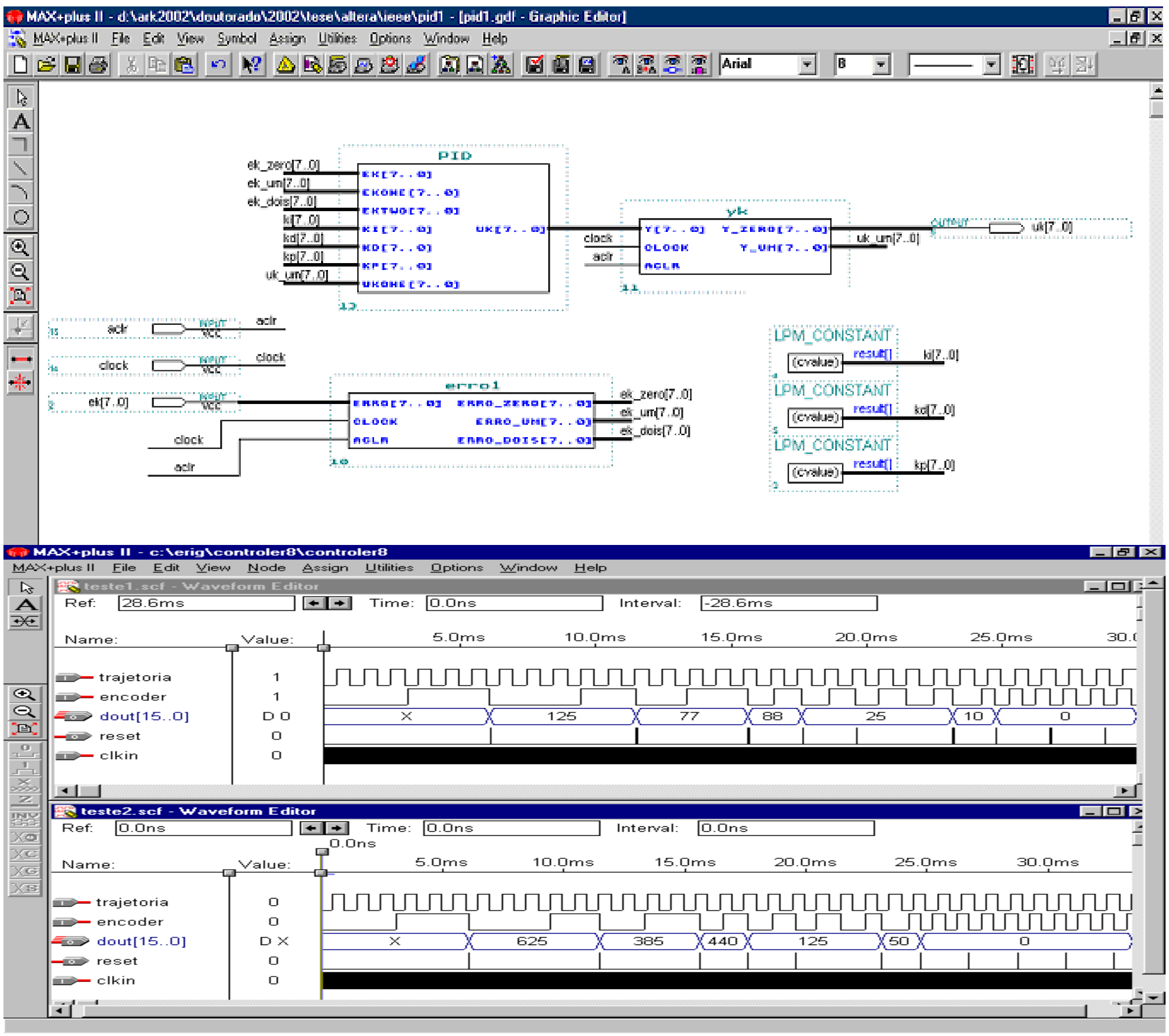

Figure 7. Equation (1) graphical language implementation and the PID Controller simulation Interface.

The “PID” block is implemented in VHDL language using dedicated libraries (Altera, 2002). This VHDL language exemple is presented in Figure (8).

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_signed.all;

ENTITY pid IS
        PORT (ek : IN std_logic_vector (7 DOWNTO 0);
                ekone : IN std_logic_vector (7 DOWNTO 0);
                ektwo : IN std_logic_vector (7 DOWNTO 0);
                ki : IN std_logic_vector (7 DOWNTO 0);
                kd : IN std_logic_vector (7 DOWNTO 0);
                kp : IN std_logic_vector (7 DOWNTO 0);
                ukone : IN std_logic_vector (7 DOWNTO 0);
                uk : OUT std_logic_vector (7 DOWNTO 0));

END pid;

ARCHITECTURE piddig OF pid IS

BEGIN

        uk <= ukone + ek * kp + ek * ki + ek * kd + ekone * kp - ekone * ki - ekone * kd + ektwo * kd;

END piddig;
```

Figure 8. Block PID VHDL language implementation.

5. Conclusions

Among all fields related to the complete achievement of an embedded project, hardware technologies and software have rapidly improved. That is particularly true for the evolution of technologies of motors, sensors, microprocessors, communication interfaces and power interfaces. From this, the idea is then to elaborate open structures, which may adapt very easily to the developments of all these technologies. The consequence of this requirement is the design of small independent modules, with communication interfaces, included within an open architecture oriented structure.

The main objective of this work was to propose a environment to aid a generic robotic mobile system project, seeking to obtain a support tool for under-graduation and graduation activities. This came from encountering the growing need to propose to graduated students researches that integrate the knowledge acquired in several disciplines and improve teamwork results. Thus, in graduate courses, this environment became an alternative support for several types of research, such as validation of control and supervision strategies, model generation and data transmission protocols, among others. Another objective was to gather knowledge in the mobile robotic area, aiming at presenting practical solutions for industrial problems, such as maintenance, supervision and transport of materials.

The use of a hierarchical architecture of control and reconfigurable logic allows a better division of the tasks associated with the mobile robot. Especially, the concept of a layer of supervisory control allows the use of test and maintenance tools, with time decrease of project and better error analysis and correction.

Another aspect to be considered is the use of the radio communication among control layers, minimizing or eliminating cables connected to mobile robot, facilitating the remote supervision, even under outdoor operation. This implementation has been realized merging knowledge acquired in multiple areas, and appears to be a very promising design strategy for a better reconfigurability and portability of systems. If a better maintenance of the system is possible, then the cost of robots may decrease in a considerable way.

Another objective is to gather knowledge in the mobile robotics area, aiming at presenting practical solutions for industrial problems, such as maintenance, supervision and transport of materials. Some promising aspects of this architecture are thus:

- Flexibility, as there is a great variety of possible configurations in the implementation of solutions for several problems associated with mobile robots.
- Great capacity of memory storage, which allows implementation of sailing strategies for maps.
- Possibility of modification of control strategies during operation of the mobile robot.

- The open architecture of this platform enables the use for educational activities.

Future research and further activities must be carried out in the following directions:

- Optimization of the current prototype in term of design and modularization.
- Validation of this open architecture approach on other kinds of prototypes.
- Proposal for an operating system dedicated to this flexible architecture.
- Development of a minimum prototype for low cost applications

6. Acknowledgement

This paper was developed with the support of the "Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq", Brazil, through a collaboration work between the 'Control Department' of Supélec, France, the Laboratory of Automation and Robotics of Unicamp, Brazil and Laboratory for Embedded Systems Innovation and Technology of Federal Center of Technological Education – Paraná – Brazil.

7. References

Accolade Design Automation, Inc. Disponível na internet via www, URL: http://www.acc-eda.com, 2002.

Actel Corporation, URL: http://www.actel.com, 2002.

Altera Corporation, URL: http://www.altera.com, 2002.

Aström, K.J. and B. Wittenmark (1997). Computer Controlled Systems. 3rd Edition, Prentice Hall, New Jersey.

ATI Technologies INC., URL: http://www.ati.com, 2002.

Atmel Corporation, URL: http://www.atmel.com, 2002.

Borenstein, J. (1995). Control and Kinematic Design of Multi-Degree-of-Freedom Mobile Robots with Compliant Linkage. In: IEEE Transactions on Robotics and Automation, Vol. 11, n° 1, pp. 21-35.

Borenstein, J., Everett H. R. and L. Feng (1996). Sensors and Methods for Mobile Robot Positioning. University of Michigan, Ann Arbor.

Boucher, P. and D. Dumur (1996). La Commande Prédictive, Technip Editions, Paris.

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987). Generalized Predictive Control: Part I The Basic Algorithm, Part II: Extensions and Interpretation. In: Automatica, Vol. 23:2, pp. 137-160.

Compton, Katherine. Reconfigurable Computing: A Survey of Systems and Software, ACM Computing Surveys, pp.171-210, vol 34, n. 2, 2002.

Dumur, D. and P. Boucher (1994). New Predictive Techniques. Control Axis Solutions. Invited Session 'Applications of M.B.P.C.'. In: Proceedings of the 3rd IEEE Conference on Control Applications, vol. 3, pp. 1663-1668.

Farritor, S., Dubowsky, S. On Modular Design of Field Robotic Systems. Journal of of Autonomous Robots, June 1999.

General Purpose Board – 80EC186, Laboratory for Embedded Systems Innovation and Technology - CEFET-PR, 1999.

Iagnemma, K., and Dubowsky, S., Vehicle Wheel-Ground Contact Angle Estimation: With Application to Mobile Robot Traction Control. Proceedings 7th International Symposium On Advances in Robot Kinematics, Piran-Portoroz, Slovenia, June 2000.

Klotchkov, I. V.; Pedersen, S. "A Codesign Case Study: Implementing Arithmetic Functions in FPGA´s". IEEE 1996.

Kollig, P.; Al-Hashimi, B.M.; Abbott, K.M. "Efficient Scheduling of Behavioural Descriptions in High-Level Synthesis". IEEE Proceedings- Computers and Digital Techniques, pp. 75 – 82, 1997.

Koren, Y., Z.J. Pasek, A. Galip Ulsoy and U. Benchetrit (1996). Real-Time Open Control Architectures and System Performance. In: Annals of the CIRP, Vol. 45/1, pp. 377-380.

Lima, C. R. E.; Rosário, J.M., "Utilização de Circuitos Lógicos Programáveis para Controle de Atuadores em Robôs Móveis", Revista Robótica, Portugal, 2001.

Lima, R. C. E, Silva; N. C., Rosário, J. M., 2000, "A Proposal of Flexible Architecture for Mobile Robotics", In Mechatronics 2000 – The 7th Forum International Conference and Mechatronics Education Workshop, 6-8 September 2000. Atlanta, 6p.

Mentor Graphics Corporation. Site URL: http://www.mentor.com, 2002.

Miyazaki, T. "Reconfigurable Systems: a Survey". IEEE Proceedings of the Design Automation Conference 1998, ASP-DAC '98, pp. 447 - 452 , 1998.

Pritschow, G., Ch. Daniel, G. Junghans and W. Sperling (1993). Open System Controllers – A Challenge for the Future of the Machine Tool Industry. In: Annals of the CIRP, Vol. 42/1, pp. 449-452.

Radiometrix Ltd, "Low Power UHF Data Tranceiver Module", site URL: http://www.radiometrix.co.uk, 2002.

Renaux, D. P. B. – PET – A Small Real-Time Support System for Microcontrollers without Virtual Memory. Laboratory for Embedded Systems Innovation and Technology - CEFET-PR, 1996.

Samet, L. Masmoudi; N. Kharrat, M.W. Kamoun, L. "A digital PID Controller for Real Time and Multi Loop Control: a Comparative Study". IEEE International Conference on Electronics, Circuits and Systems, pp. 291 – 296, vol.1, 1998.

Soeterboek, R. (1992). Predictive Control – A Unified Approach. Prentice Hall.

Synopsys, Inc., URL: http://www.synopysis.com, 2002.