EMBEDDED OPEN ARCHITECTURE ROBOTIC CONTROLLER FOR POSITION AND FORCE CONTROL

André Schneider de Oliveira, oliveira@emc.ufsc.br
Raul Guenther, guenther@emc.ufsc.br
Universidade Federal de Santa Catarina
Departamento de Engenharia Mecânica
Laboratório de Robótica
Caixa Postal 476
Campus Universitário – Trindade
88040 900 - Florianópolis, SC. Brasil

Abstract. *Generally the robotic controllers are developed for position control, without accomplishing integrally the requirements of tasks in which interactions with the environment occur. However, this is currently one of the main research areas in robotics. To consider this interaction the robot controller has to give priority to the force control time response, because in the instant of end-effector's contact with the surface, several forces act on the system. Depending on the speeds and the accelerations involved in the process, damages (errors) can occur. To avoid these effects, complacencies are inserted in tool or in surface of operation. This paper presents the development of an embedded open-architecture robotic controller for force control, which uses parallel and distributed processing techniques, and diminishes the necessity of compliance in system, supplying a processing in real-time of application and the total control of information. The system has a modulated structure, with a communication architecture derived from the standard IEEE 802.4: Token Bus, and uses a high-performance communication bus as the popular interfaces USB 2.0 and the industrial protocol CAN. The system is composed of eight processors operating simultaneously: seven digital signals controller (DSC) and a personal computer.*

Keywords*: Force control, Robotic controller, Open-architecture.*

1. INTRODUCTION

Current robotic applications are limited by the industry state of art of the manipulators control algorithms. The inclusion of force and vision feedbacks, the possibility of cooperation between two or more manipulators, the control of robots with irregular topology will certainly enlarge the industrial robotics applications. The development of control algorithms to this end brings the necessity of the use of controllers with open architecture.

As a matter of fact, rapid prototyping, i.e., the capability of designing and testing new control algorithms in short time and with limited costs, is becoming a fundamental issue in industrial robotic applications.

Notice that the "degree of openness" in a robot controller may vary from one system to the other. Usually the control of some components of the system (e.g., the power system, the low level control) are proprietary and cannot be modified by the user, others may be considered open (e.g., the communication interface, the higher level control), i.e., are based on standard hardware and software with open interface specifications.

Ford (1994) defines this "degree of openness" based on the controller access level concept. The author classifies the robotic controllers in three categories: proprietary, opened or hybrid. A proprietary is a closed system or an owner system, and in those systems it is really complicated or impossible to integrate external hardware. In hybrid systems the majority of control layers are accessible, however some aspects, as the control laws, are closed. In an open architecture all the aspects of the project can be modified, and it allows to add new hardware's, to modify the servo motors software structure, etc, These systems could be easily adapted to new situations.

Recently Lippielo et al. (2007) outlined that various open control architectures for industrial robots have already been developed by robot and control manufacturers as well as in research labs.

We mention additionally the open control architecture controller developed for position control in a constrained workspace for the Roboturb project robot in our Robotic Laboratory at Santa Catarina Federal University (Raposo, 2006). The Roboturb project continuity, which involves the force control necessity, motivates strongly the work presented in this paper.

Most of the existing robot control open architectures are based on a standard PC hardware and a standard operating system, because I/O boards and communication boards for robots have a higher cost in relation to the similar boards for PCs. Another reason is the lack of standardization of robot peripherals, with each manufacturer developing its own protocols and interfaces, forcing the users to buy all the components of a single manufacturer (Lages et al. (2003)). Additionally, a PC based controller can be integrated more easily with many commercially available add-on peripherals such as mass storage devices, Ethernet card and other I/O devices. So, the facility to integrate other functionalities is a strong reason to use a standard PC hardware in robot control open architectures.

Another reason is because the robot programming languages are, at low level, more similar to the Assembly languages than to the modern high level languages and this may difficult the implementations (Lages et al., 2003). In a PC based controller standard software development tools (e.g., Visual C++, Visual Basic or Delphi) can be used.

In this work we present the design and the development of a robotic controller with a totally open architecture built to control force and position. This architecture provides flexibility, the knowledge of all the control structures and allows the user to modify all the layers of the controller. The used controller conception aims to fulfill the following requirements: high capacity of processing, low cost, connectivity with other systems, availability for the remote access, easiness of maintenance, flexibility in the implementation, integration with a personal computer and programming in high level.

The paper is organized as follows. First we describe the controller functional architecture in section 2. In section 3 we present the hardware development. The experimental set up is described in section 4, and in section 5 we outline some conclusions.

2. THE CONTROLLER FUNCTIONAL ARCHITECTURE

We designed the controller with a modulated structure, which communication architecture derived from the standard IEEE 802.4: Token Bus. The functional architecture is based on the ISO 7498-1 and it results in the layered architecture presented in Fig.1. This includes a task layer, an integration layer, a communication layer, an interface layer and a physical layer.

As we aim to develop a controller with a total open architecture, all the layers of the project, including the software processed for the computer and firmware processed by the digital signals controllers, are modifiable.

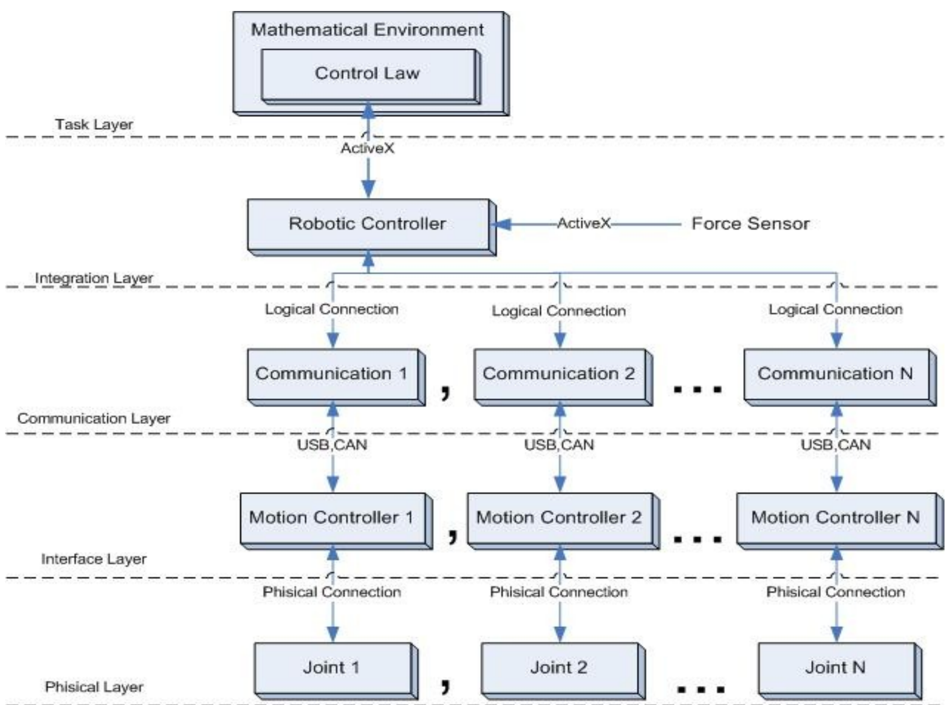

Figure 1. Functional layered architecture.

In the sequel we describe the main functions of each layer.

2.1. Task layer

The task layer has a mathematical environment prepared to make operations with matrices in which the control law is stored. The information proceeding from the n joints are available in matrices nx1 corresponding to the position vector q, and the velocity vector $\dot{q}$, where the lines represent the joints. The force sensor data are stored in a matrix 6x1 called $h = \begin{bmatrix} f_x & f_y & f_z & \mu_x & \mu_y & \mu_z \end{bmatrix}^T$, which contains forces and moments data. The information to be directed to the motors and encoders is stored in an nx3 control matrix u .In this layer the user develops the control laws of position and/or force of the manipulator and it is possible to carry through the task simulation.

2.2. Integration layer

In the integration layer the concatenation and the organization of all the information coming from the sensors and to be sent to the superior layer are done. In case of the inclusion of a new hardware to the system, it is necessary to add its control structure to this layer. This is carried through by a high-level application that manages the power unit and control unit. Preventing any irregular movements and danger situations and controlling the components of the lower level. In this software the controller's components can be activated or disabled independently.

2.3. Communication layer

The communication layer controls the data transfer by managing the interface USB (Universal Serial 2,0 Bus) and the industrial protocol CAN (Campus Area Network), both high performance communication devices. The USB makes a system interconnection through a star form topology, which has the computer as a central knot. Each USB door supports up to 127 devices and, in this manner, it is possible to connect a great quantity of joints to the controller. The protocol CAN forms the bus between the secondary knots (motion controllers) and this structure results in the architecture derived from the standard IEEE 802.4: Token Bus shown in Fig. 2. The implementation of this bus is still being explored and intends to introduce the possibility of a joint to access information of another joint without passing through the central knot. This will increase the performance of the net and it gives the opportunity to an implementation of the system of control without the central knot: a totally embedded control.

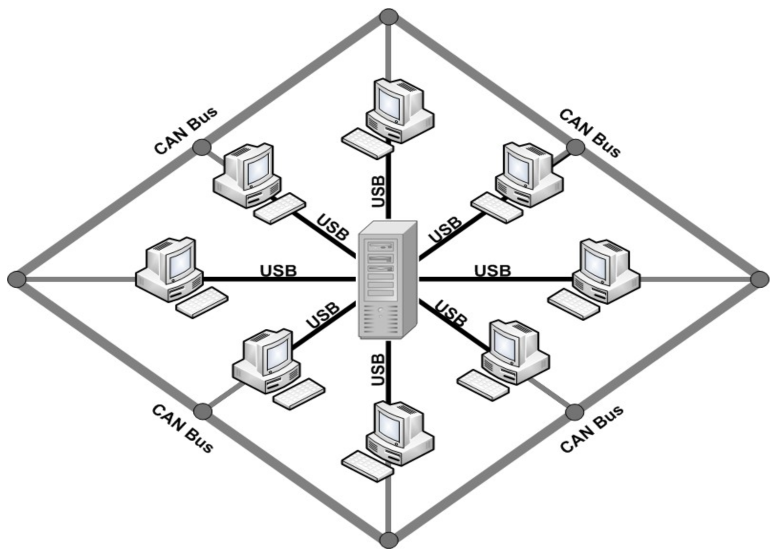

Figure 2. Communication net topology.

2.4. Interface layer

The interface layer comprises the embedded systems that carry out the control of the robotic joints, named motion controllers, whose block diagram is presented in Fig. 3.

As it could be observed in Fig.3, each of these motor digital controllers decodes the corresponding encoder signal and generates the modulation width pulse (PWM) to the control of the respective motor. Each of these systems has an optical isolated interface to prevent any inadequate return to the processor. It possesses a great amount of expansion doors, which allows the connection of other tools.

We developed the controller with a modular architecture to have an independent control for each joint and so, divide the mathematical complexity among the processors of the system. This results in a distributed processing organized by the central knot (computer), where the operations occur in parallel. This methodology facilitates the expansion and maintenance of the system.

Currently the system operates with a maximum tax of update of the signals of 1 ms, only for a convention of literature. In case of necessity this largeness can be diminished.

2.5. Physical layer

The most inferior layer, here denominated physical layer, is the power unit of the motors and the angular position sensors.

3. HARDWARE DEVELOPMENT

The adopted functional architecture hierarchical structure, together with its articulation into different modules, suggests a hardware implementation that exploits distributed computational resources interconnected by means of suitable communication channels (Oliveira and Andrade, 2006). Beside this, our hardware development considers the special requirement for robot controllers with force control presented in the sequel.

Special requirement for robot controllers that includes force control

Generally the robotic controllers are developed for applications that require only position control, and the robot end effector doesn't contact the workspace during its movement. In applications that need force control, the end effector contacts some surface in its workspace and this interaction generates contact forces that must be controlled in a way to fulfill the task correctly, without damaging both, robot tools and the working objects.

The contact force intensities, originated by tool movements commanded by the robot controller, depend on both, the tool rigidity and the object surface rigidity, and they must be also controlled. A small tool movement could originate large force intensities in case the tool and the object surface rigidity are large. It should be noted that by introducing compliance to the tool we generate a delay in the application of the forces and this could be unacceptable in some applications. Consequently, the system should have a small time response to these forces, to prevent tool, robot or object damages.

The use of high performance systems is a requisite of controllers for application of force control.

Development description

The system's hardware was developed and built using high performance and reliability, low cost and easiness to be found in the market components. Figure 3 shows the diagram of internal blocks used in the motion controllers.

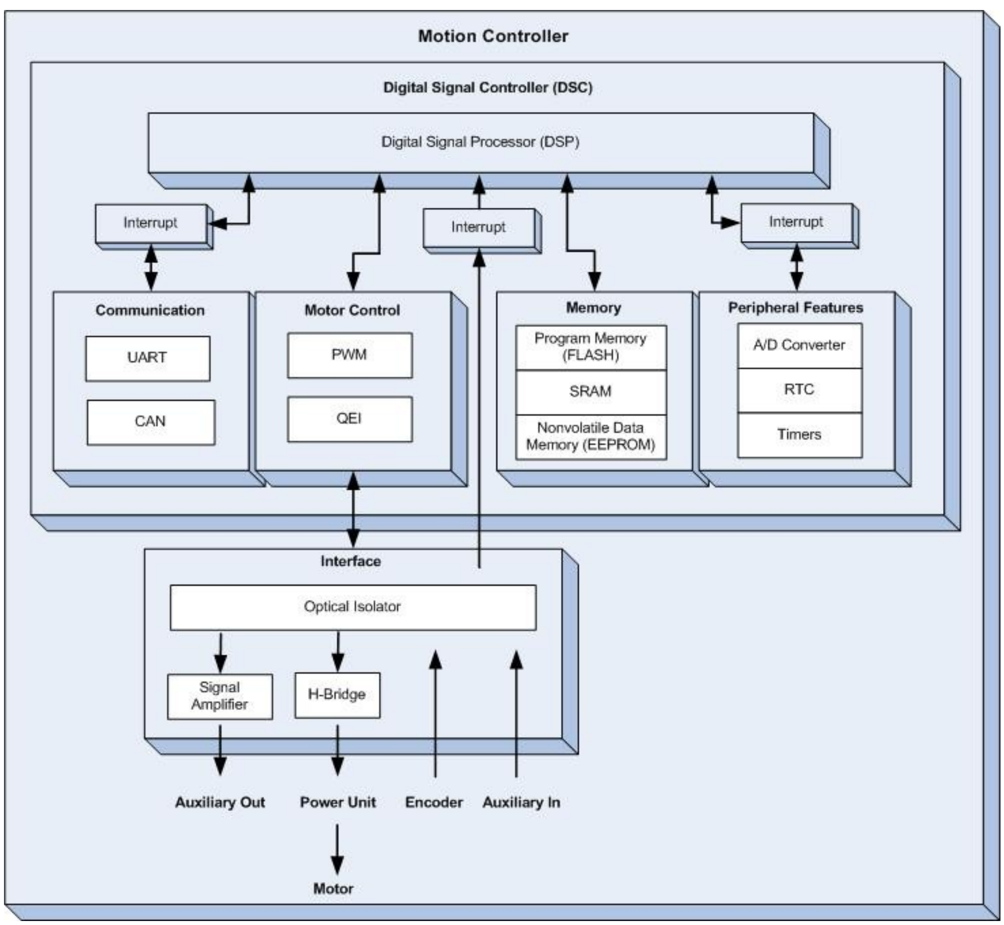

Figure 3. Motion controller block diagram.

The main component is a digital signal controller (DSC) produced by the Microchip Technology Inc. named dsPIC30F6010A. It operates with 16-bits, in a 120 MHz frequency with a package TQFP of 80 pins, and is one integrant of the family of the motors control. It possesses a great amount of well differentiated modules including an ample program memory with a 144K and a non-volatile memory with 4096 bytes for information storage. It has 16 ways for A/D conversions and the necessary modules of communication.

For the communication through USB we used a component which carries through the conversion of module UART for the bus. This component supports transference taxes up to 3 Megabaud and is manufactured by the FTDI (Future Technology Devices International Ltd). Moreover, it possesses other functionalities, including the generation of a digital external signal oscillator with changeable frequencies. Besides this, the same manufacturer produces available Royalty-Free drivers for many operational systems, for this form of implementation.

To implement the requirements for the physical layer defined by the ISO-11898, we connect the CAN industrial protocol to a transceiver of high speed, which supports until 1Mb/s.

The system firmware implementation uses the high level language C. This is completely modulated and organized in units, to facilitate modifications. All the modules operate with interruptions of the processor with distinct priorities, such that an operation of less priority doesn't delay a higher importance process.

The module of motion controller is also composed by a 16 bits PWM generator and by a module to read the quadrature encoder (named QEI), which we extend for 32 bits. Connected to it there is an optical decoupling barrier and an H-bridge for the control of the power unit that supports 100V and 8A. There are also amplified auxiliary output channels, which operate until 100V and 6A. To protect the system, the encoder entrances and the auxiliary inputs also have been connected to optical decoupling barrier. Internally there is still a great amount of resources that had not been used and that can be useful in future upgrades of the system.

The resulting motion controller layout is shown Fig. 4.

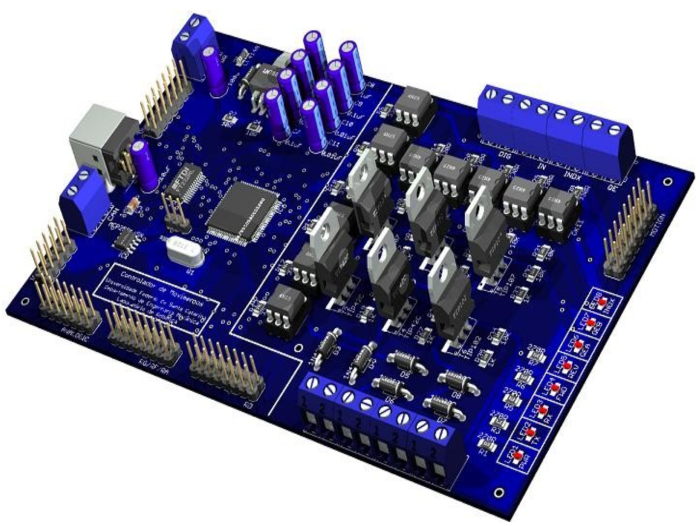

Figure 4. Motion controller layout.

4. EXPERIMENTAL SETUP

We are using the six degrees of freedom industrial robot, REIS Rv15, shown in (Fig. 5), and a force sensor manufactured by the JR3 Inc. to validate experimentally the developed controller. At the same time this robot is being retrofitted using this controller.

4.1. REIS Rv15 robot

The REIS Rv15 robot has six rotating joints acted by electric motors and the angular positions measurement are done using incremental optical encoders. It is a manipulator with a topology that is very common in industry applications, which constitutes an anthropomorphous arm (joints 1, 2 and 3) with a spherical wrist (joints 4, 5 and 6).

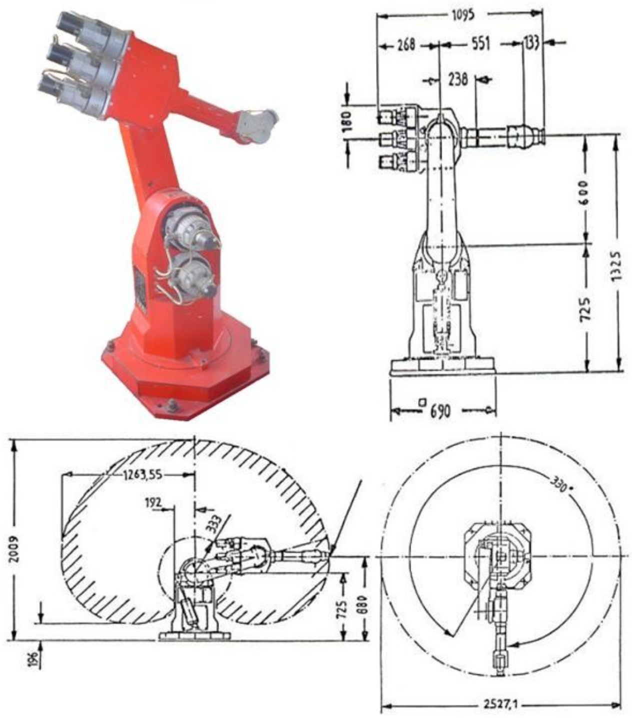

Figure 5. Robot REIS Rv15 (dimensions in millimeters).

4.2. JR3 Force sensor

We use a force sensor which supports loads until 200N, attached to the robot as shown in Fig. 6.

Figure 6. Force sensor and end-effector.

5. CONTROLLER IMPLEMENTATION

The Fig. 7 presents a complete diagram of the embedded open architecture robotic controller for a manipulator with six degrees of freedom, containing it data flow and the systems interconnections.

The control system was validated with implementation of impedance control law (Fig.8), based on Sciavicco and Siciliano (2004), defined in the following equation:

$$M_d \ddot{\tilde{x}} + K_d \dot{\tilde{x}} + K_P \tilde{x} = h_A \quad (1)$$

Where, M_d is mass matrix, K_d is damping matrix, K_p is stiffness matrix and h_A is interaction force.

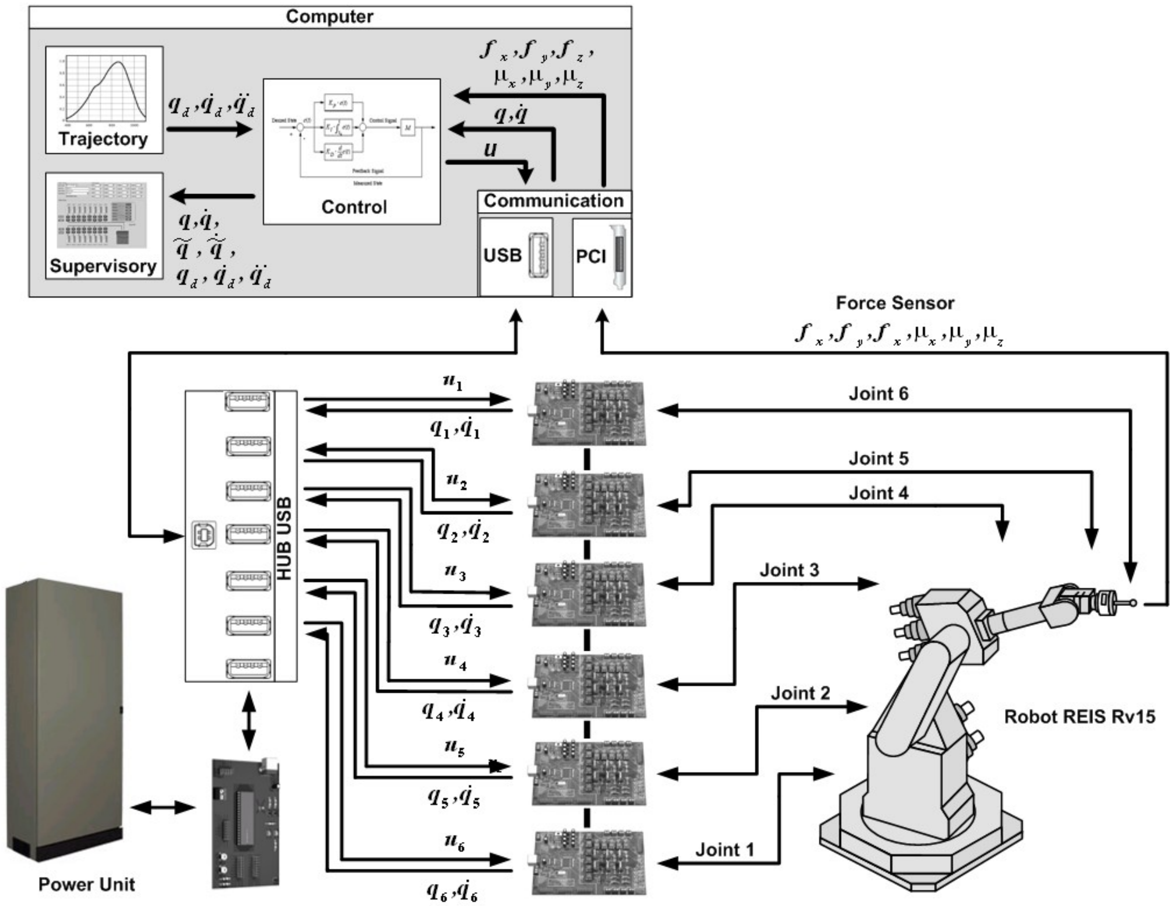

Figure 7. Open architecture robotic controller complete diagram.

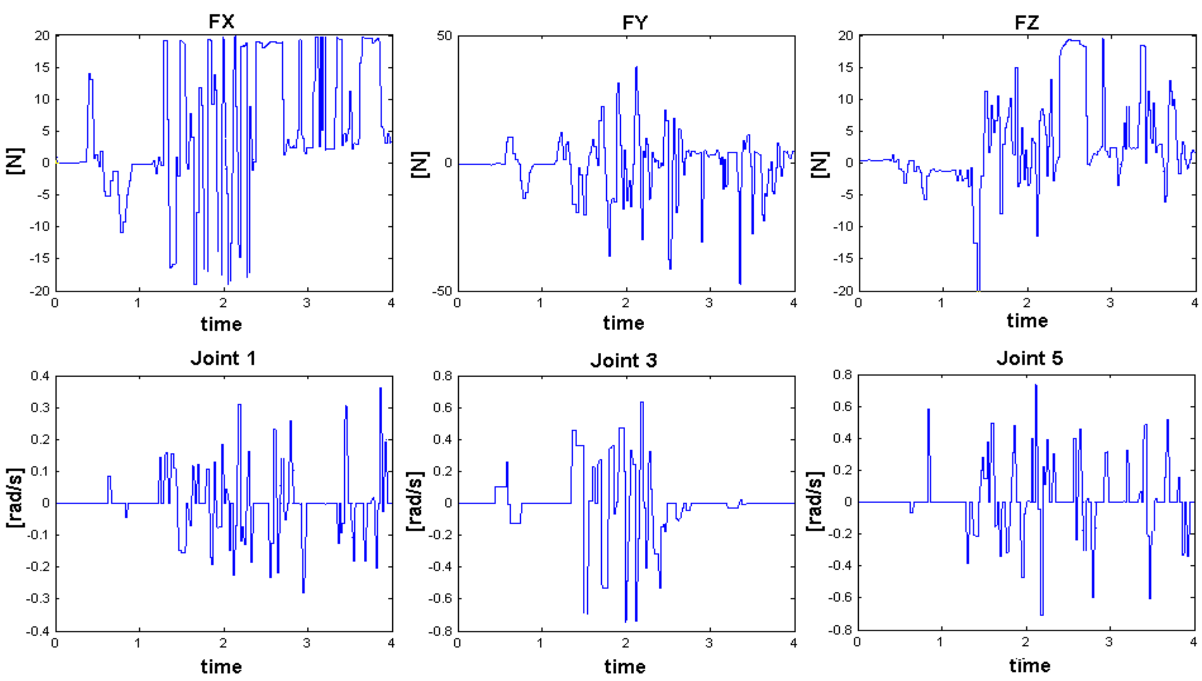

Figure 8. Impedance control.

6. CONCLUSION

This work presents the design and the hardware development of an embedded open architecture robotic controller aiming to position and force control.

The resulting controller has the following properties: high capacity of processing, low cost, connectivity with other systems, availability for the remote access, easiness of maintenance, flexibility in the implementation, integration with a personal computer and programming in high level.

Retrofitting a REIS Rv15 robot with a JR3 force sensor attached to its end has been performed to experimentally validate the presented controller with implementation of impedance control law.

7. ACKNOWLEDGEMENTS

This work was partially supported by 'Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ)' Brazil.

8. REFERENCES

Ford, W., 1994, "What is an Open Architecture Robot Controller?", 9th IEEE Int. Symp. On Intelligent Control, pp. 27-32.

IS Association, 1990, "IEEE Standard 802.4: Information processing systems-Local area networks – Token passing bus access method and physical layer specifications ".

ISO, 1994, "ISO/IEC Standard 7498-1: Information Technology – Open Systems Interconnection – Basic reference model".

ISO, 11898, "ISO Standard: 11898: Road Vehicles, Interchange of digital information - Controller Area Network (CAN) for high speed communications".

Lages, W.F., Henriques, R.V.B., Bracarense, A. Q., 2003, "Arquitetura Aberta para Retrofitting de Robôs" (in portuguese), Manet Notes Workshop, Bragança Paulista, São Paulo, Brazil.

Lippiello, V., Villani, L., Siciliano, B., 2007, "An open architecture for sensory feedback control of a dual-arm industrial robotic cell", Industrial Robot: An International Journal, Vol. 34 No.1, pp. 46-53.

Oliveira, A.S. and Andrade, F.S., 2006. "Sistemas Embarcados : Hardware e Firmware na Prática" (in portuguese). Ed. Érica, São Paulo, Brazil.

Raposo, E., 2006, "Um Modelo Aberto para Robôs", PhD. Thesis (in portuguese), Universidade Federal de Santa Catarina, Department of Mechanical Engineering, SC, Brazil.

Sciavicco, L., Siciliano, B., 2004, "Modeling and control of robot manipulators", The McGraw-Hill Companies.

9. RESPONSIBILITY NOTICE

The author is the only responsible for the printed material included in this paper.