13

Rapid Prototyping for Mobile Robots Embedded Control Systems

Leonimer Flavio de Melo, Jose Fernando Mangili Junior
and Jose Augusto Coeve Florino
State University of Londrina
Brazil

1. Introduction

One of the main motivations of this work is to propitiate a virtual environment that facilitates development of archetypes of embedded systems, emphasizing implementation of tools that allow the simulation of kinematic, dynamic and control conditions, with real time monitoring of all important points of the system. In this way, the proposal of a virtual simulator of mobile robotic systems is presented together with techniques of rapid prototyping.
The use of the rapid prototyping technique in mobile robotic systems differs from the traditional target used in mechanics engineering and enters in a new field of research and development for projects of mobile robots mechatronics systems. In this way, the rapid prototyping of these systems is associated not only with the project of the physical system, but mainly with the experimental implementations in the fields of hardware and software of the robotic system.
In the context of this work, the rapid prototyping is the technology that allows, in conjunction with a simulation virtual environment, the development of a mobile robots controller project. After tested and validated in the simulator, the control system is programmed in the control onboard memory of the mobile robot. In this way, a economy of time and material are obtained, validating first all the model virtually for later operating the physical implementation of the system.
It is fundamental that the architecture of hardware of the considered system be opened and flexible in the way of effecting the necessary modifications for system optimization. A proposal of open architecture system was presented in Melo *et al.* (2005) and summarized in this work. The software of the embedded control system of the mobile robot, in the context of the rapid prototyping, can be elaborated in simulators and have all the parameters tested for adjustments that makes necessary in accordance with the physical system to be implemented, the hardware architecture, the actuators and the sensors. In this way, in the context of this work, the rapid prototyping is then the methodology that allows the creation of a virtual environment of simulation for the project of a controller for mobile robots. After being tested and validated in the simulator, the control system is programmed in the control board memory of the mobile robot. In this way, an economy of time and material are obtained, firstly validating all the model virtually and later operating the physical implementation of the system (Dudek & Jekin, 2000).
The choice of DSP TMS320C6474 multicore digital signal processor as main onboard manager device falls into two main factors. First, its great information and instructions processing

capacity, operating at 1,2 GHz clock frequency and executing up to 10 billion information per seconds in its maximum performance, makes the system sufficiently efficient for implementations of complex software techniques required in the robotic navigation systems. Second because it has one specific MatLab tool box that can be used for its real time programming with *hardware-in-the-Loop* (HIL) techniques, that it is one of the simulation techniques utilized in the rapid prototyping systems for embedded mobile robotic controllers. This new technique of simulation, HIL (previously only available in the aerospace and aeronautical industry), can be used for the development and establishment of parameters of embedded mobile robotic controllers (Ledin, 2001).

The development of control systems for independent mobile robots has appeared as a great challenge for researchers until current days. Different platforms for project of control system for independent mobile robots have being used in diverse research areas. For many years the researchers have been designing control systems that present an intelligent behavior in controlled environments, with ideal situations, but that normally does not keep the same performance in real world. Innumerable systems of control exist to be used in real world, but generally these systems are limited and they do not present an independent or intelligent behavior.

For mobile robotics systems, so many possible applications exist, for example in transport, monitoring, inspection, cleanness of houses, space exploration, aid physical deficient, among others (Braunl, 2008). However, independent mobile robots had not yet caused much impact in domestic or industrial applications, mainly had the lack of a system with robust, trustworthy and flexible control that it would allow these robots operated in dynamic environments, less structuralized, and inhabited by human beings. The development of a mobile robotic model system with open architecture and flexible control, with robust control system, that incorporates the most modern embedded hardware technology and that makes possible the operation of a mobile robotic systems in a real world environment is one of the motivations of this work.

The locomotion planning, under some types of restrictions, is a very vast field of research in area of mobile robotics (Graf, 2001). The basic planning of trajectory for mobile robots imply the determination of a way in space-C (*configuration space*) between an initial configuration of robot and a final configuration, in such a way that robot does not collide with no obstacle in the environment, and that the planned movement is consistent with kinematic restrictions of the vehicle (Beeson *et al.*, 2007). In this context, one of the boarded points in this work was development of a trajectory calculator for mobile robots.

2. The mobile robot platform

Platforms for knowledge consolidation could be used in several educational and research areas, such as modeling, control, automation, power systems, sensors, transmission of data, embedded electronics and software engineering. In fact, the use of mobile robots as a basis for knowledge consolidation has been successfully adopted in many educational and research institutions mainly because they appear to be a quite attractive low cost solution that allow integration of several important areas of knowledge (Braunl, 2008). Mobile robots also become a better solution for practical problems in modern society. These appointments show a large applicability of mobile robots and a crescent request in modern world (Beeson *et al.*, 2007). One of the proposals of this project is developing a generic open system for control a mobile robot and supplies this need.

The system emphasizes control structure, supervision and transfer of information. In a context of educational and research aims the project aspects and integration solution compose the desired know-how acquired during development of system, which certainly would not to be approached if a commercial mobile robot was acquired.

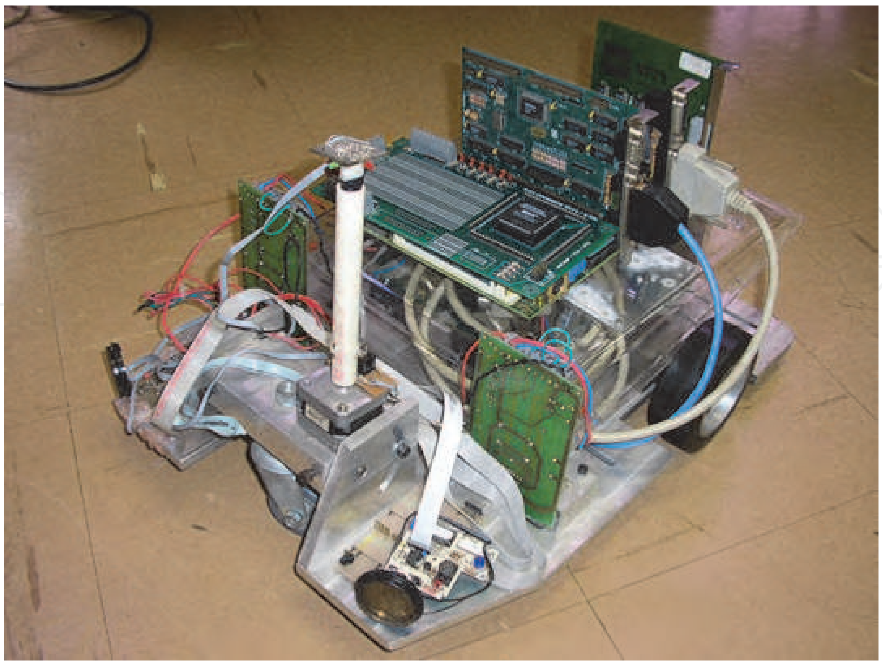


Fig. 1. Mobile robot platform prototype.

An embedded processor, with a dedicated control software, to be used on a platform mobile robotics, is considered (Siciliano *et al.*, 2009). In addition to this platform another one, a commercial platform, coupled to a communication net, is analyzed. The set of platforms, whose objectives are making use of existing communication interfaces and providing an embedded user interface alternative in mobile robot, allows creation of a powerful link with external world. The objective of this platform is to make use of the existing communication interfaces, as well as to provide an embedded user interface alternative in mobile robot. Another aspect considered, is flexibility of hardware project, which allows the expansion of mobile robot facilities. New sensor combinations should be used. Different supervision and control models should equally be used to carry out the mobile robot tasks.
This paper presents the implementation of a virtual environment for project simulation and conception of supervision and control systems for mobile robots and focus on the study of the mobile robot platform, with differential driving wheels mounted on the same axis and a free castor front wheel, whose prototype used to validate the proposal system is depicted in Fig. 1 and Fig. 2 which illustrate the elements of platform.

3. The control architecture system

The control architecture system can be visualized at a logical level in the blocks diagram in Fig. 3.
The system was divided into three control levels, organized in the form of different degrees of control strategies. The levels can be described as:

- **Supervisory control level**: This represents a high level of control. In this level it was possible to carry out the supervision of one or more mobile robots, through the execution of global control strategies.

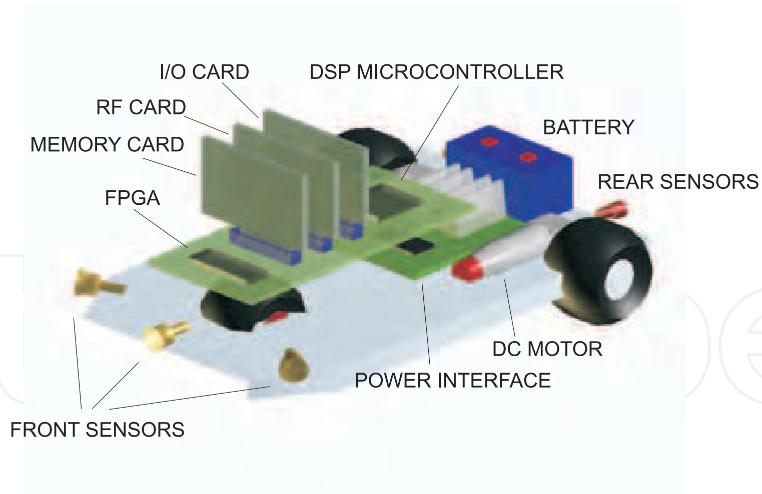


Fig. 2. Illustrative mobile robot platform and elements.

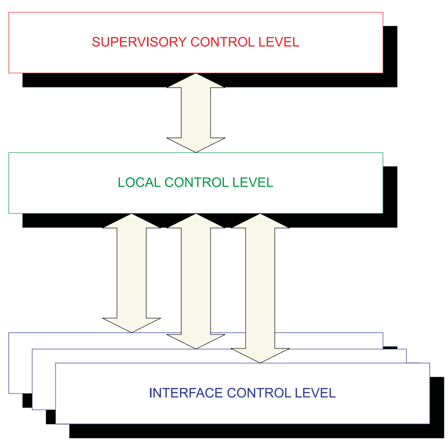


Fig. 3. Different control levels of the proposed system.

- **Local onboard control level**: In this level control was processed by the mobile robot embedded software implemented in a multicore DSP processor. The control strategies allowed decision making to be done at a local level, with occasional corrections from the supervisory control level. Without communication with the supervisory control level, the mobile robot just carried out actions based on obtained sensor data and on information previously stored in its memory.
- **Interface control level**: This was restricted to strategies of control associated with the interfaces of the sensor and actuators. The strategies in this level were implemented in hardware, through FPGA(Field-Programmable Gate Array) device.

Figure 4 depicts the control architecture with more details, with the levels controls implemented on the mobile robot platform.
Architecture, from the point of view of the mobile robot, was organized into several independent blocks, connected through the local bus that is composed by data, address and control bus (Fig. 5). A master block manager operates several slave blocks. Blocks associated with the interfaces of sensors and actuators, communication and auxiliary memories were subjected to direct control from the block manager. The advantage of using a common bus was the facility to expand the system. Inside the limitations of resources, it was possible to add new blocks, allowing an adapted configuration of the robot for each task.

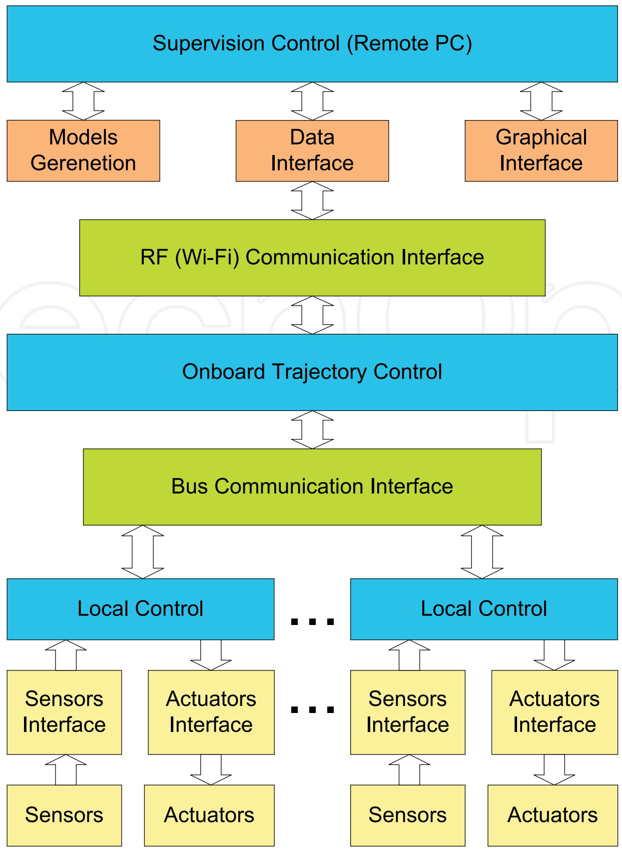


Fig. 4. Mobile Robot control architecture.

3.1 Control architecture blocks description

- **Supervisory control block**: Is the high level of control. In this block is managed the supervision of one or more mobile robots, through the execution of global control strategies. Is implemented in an IBM PC platform and is connected with the local control level, in the mobile robot, through Ethernet wireless WI-FI link. This protocol uses IEEE 802.11a standard for wireless TCP/IP LAN communication. It guarantees up to 11 Mbps in the 2.4 GHz band and requires fewer access points for coverage of large areas. Offers high-speed access to data at up to 100 meters from base station. 14 channels available in the 2.4 GHz band guarantee the expansibility of the system with the implementation of control strategies of multiple robots.
- **Master manager block**: Responsible for the treatment of all the information received from other blocks, for the generation of the trajectory profile for the local control blocks and for the communication with the external world. In communication with the master manager block, through a serial interface, a commercial platform was used, which implemented external communication using an Ethernet WI-FI wireless protocol. The robot was seen as a TCP/IP LAN point in a communication net, allowing remote supervision through supervisory level. It's implemented with Texas Instrument TMS320C6474 multicore digital signal processor, a 1,2 GHz device delivering up to 10000 million instructions per second (MIPs) with highest performing.
- **Sensor interface block**: Is responsible for the sensor acquisition and for the treatment of this information in digital words, to be sent to the master manager block. The implementation of that interface through FPGA allowed the integration of information

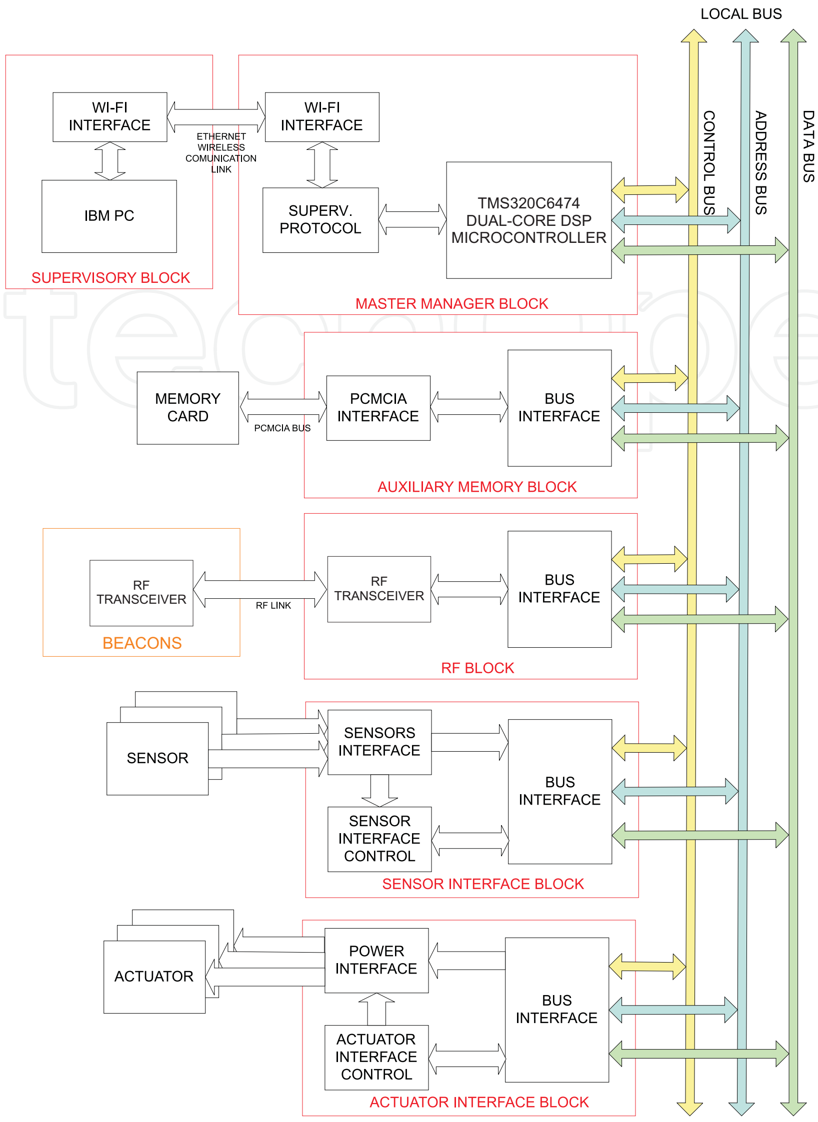


Fig. 5. Hardware architecture block diagram of the proposed system.

from sensors (sensor fusion) locally, reducing manager block demand for processing. In same way, they allowed new programming of sensor hardware during robot operation, increasing sensor treatment flexibility.

- **Actuator interface block**: This block carried out speed control or position control of the motors responsible for the traction of the mobile robot. The reference signals were supplied through bus communication in the form of digital words. Derived information from the sensor was also used in the controller implemented in FPGA. Due to integration capacity of enormous hardware volume, FPGA was appropriate to implement state machines, reducing the need for block manager processing. Besides the advantage of the integration of the hardware resources, FPGA facilitated the implementation and debugging. The possibility of modifying FPGA programming allowed, for example, changes in control strategies of the actuators, adapting them to the required tasks.
- **Auxiliary memory block**: This stored the information of the sensor, and operated as a library for possible control strategies of sensors and actuators. Apart from this, it came with an option for operation registration, allowing a register of errors. The best option was

an interface PCMCIA, because this interface is easily accessible on the market, and being a well adapted for applications in mobile robots, due to low consumption, little weight, small dimensions, high storage capacity and good immunity to mechanical vibrations.

- **RF beacons communication block**: It allowed the establishment of a bi-directional radio link for beacons data communication. The objective, at the first moment, is establish communication with all beacons in the environment, not at same time, but one by one, recognizing the number of active beacons and their respective codes. At second moment, this RF communication block sends a determinate code and receive back the same code, transmitted from respective beacon. The RF ToF is calculated by DSP processor. To implement this block was used a low power UHF data transceiver module BiM-433-40.

4. Mobile robot simulator

The use of the system has begun to gather the main points for generation of the mobile robot trajectory. The idea is to use a system of photographic video camera that catches the image of the environment where the mobile robot navigates. This initial system must be able to identify the obstacles of the environment and to generate a matrix with some strategic points that will be good for input data for the system of trajectory generation. Figure 6 presents a general vision of the considered simulator system.

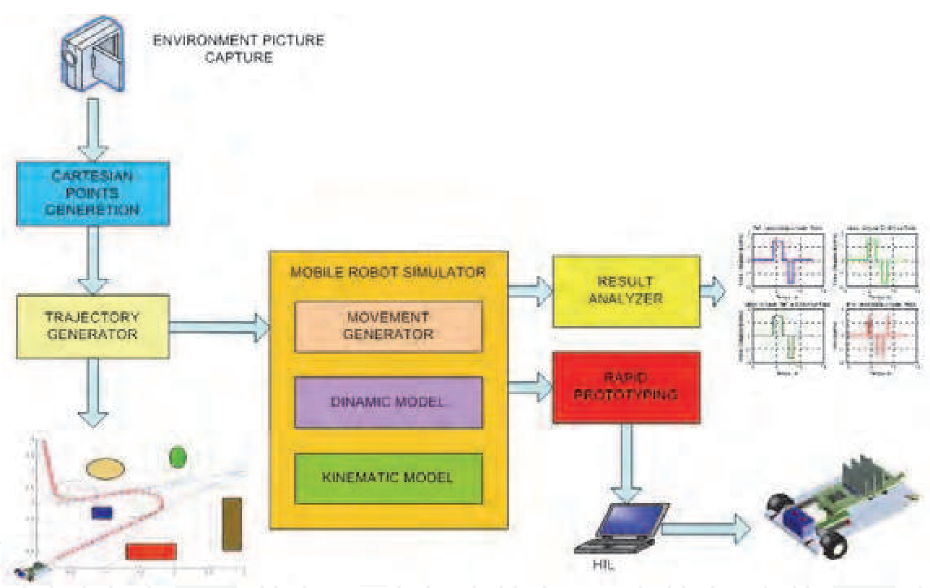


Fig. 6. General vision of the trajectory generator system.

This system is particularly interesting and can be used, for example, in robotic soccer games, where the navigation strategies are made from images of the environment (soccer field) and the obstacles are the other robots players. As it's described follow, with this system, the best trajectory can be defined and traced, respecting always the kinematics holonomics or nonholonomics constraints of the robotic systems in question, and to make all the simulation of the system foreseeing imperfections and analyzing results before the final implementation of the control system in the mobile robot (Melo & Rosario, 2006).

4.1 Mobile robot control structure

The tasks carried through for the mobile robots are based on the independent movement of each degree of freedom, coordinated from a trajectories plan based in its kinematic model.

In the most of the cases, the tasks programming is planned with anticipation and a map of the environment is loaded in the robot memory board. The mobile robot accomplishes the trajectory with sequence of independent movements of each axle, until reaching the desired final position. From the knowledge of these articulated positions, A generator of references (profile of speeds) based on the kinematic characteristics of each joint is easily implemented (Siciliano *et al.*, 2009).
For accomplishment of tasks in level of cartesian coordinates system and for generation of the reference signals for the position controller of each robotics joint of the mechatronics system in study, the establishment of mathematical model based in the kinematics of the system becomes necessary. Therefore, the control of a robot needs procedures to transform the data of positioning reference, such as the linear speed and the bending radius, in cartesian coordinates, when it is desired to realize the control through a cartesian referential (Shim et al., 1995). The Fig. 7 illustrates the mobile robot structure of control with the representative blocks of the trajectory generation, dynamic and kinematic model of the system.

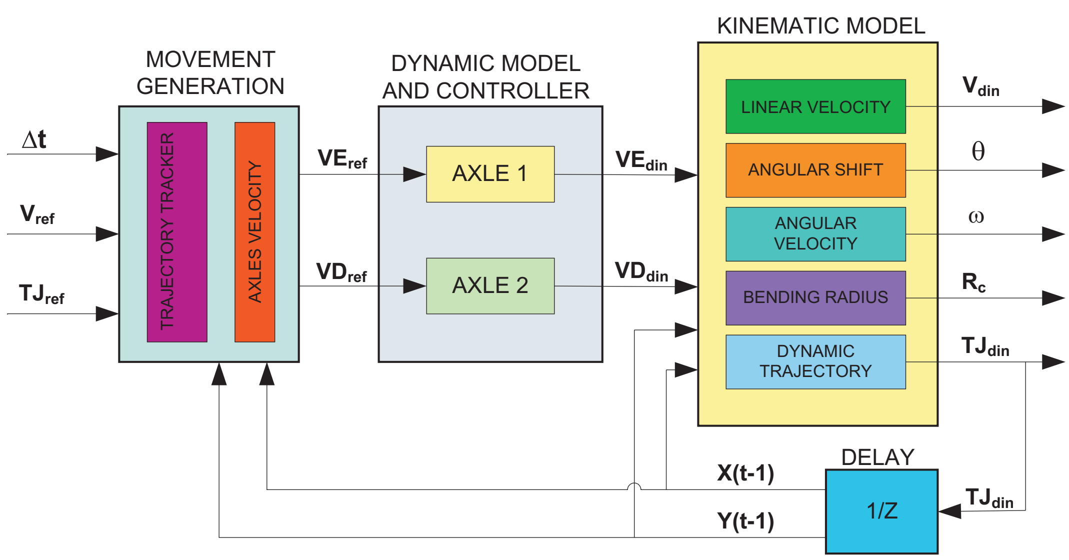


Fig. 7. The mobile robotic control structure.

The trajectory generator receives the references data, such as the positioning vector $\mathbf{X}_{ref} = [\mathbf{x}_{ref}, \mathbf{y}_{ref}, \theta_{ref}]$, the robot reference linear speed V_{ref} and the robot instantaneous trajectory radius R_{curv}, that are converted into VE_{ref} (linear speed of the left wheel) and VD_{ref} (linear speed of the right wheel). These differentiated speeds are received by the controller, and in the dynamic model of the system, they are sent to the respective wheels of the robot, through its actuators. Then are generated by the controller the vectors VE_{din} (dynamic linear speed of the left wheel) and VD_{din} (dynamic linear speed of the right wheel). Into the block of the kinematic model, these data are converted into the vector final positioning of the robot $\mathbf{X} = [\mathbf{x}, \mathbf{y}, \theta]$.

4.2 Trajectory embedded control

The figure 8 illustrates an example of an environment with some obstacles where the robot must navigate. In this environment, the robot is located initially in the P1 point and the objective is to reach the P4 point. The supervisory generating system of initial cartesian points, must then supply to the module of embedded trajectory generation, the cartesian points P1, P2, P3 and P4, that are the main points of the traced route.

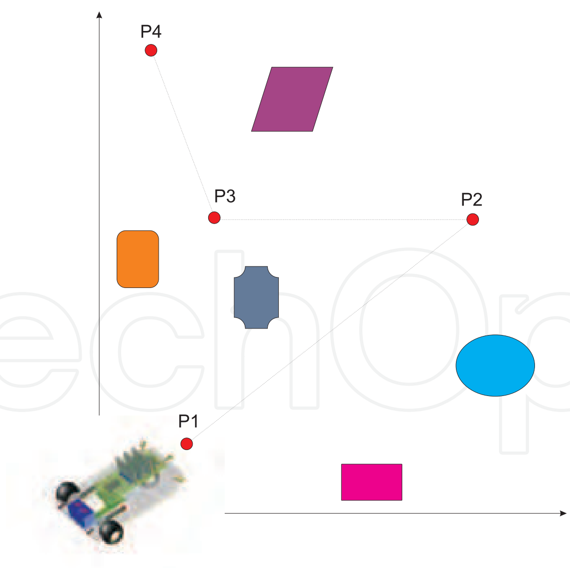


Fig. 8. Example of an environment with some obstacles where the robot must navigate.

The use of the system begin with the captation of main points for generation of the mobile robot trajectory. The idea is to use a system of photographic video camera that catches the image of the environment where the mobile robot will navigate. This initial system must be capable to identify the obstacles of the environment and to generate a matrix with some strategical points that will serve of input for the system of embedded trajectory generation.
The mobile robot embedded control system receives initially, through the supervisory system, a trajectory to be executed. These data are loaded in the robot memory that are sent to the module of trajectory generation. At a time the robot starts to execute the trajectory, the dynamic data are returned to the embedded controller, who, with the measurements and sensing, makes the comparisons and due corrections in the trajectory.
The trajectory embedded control system of the mobile robot is formed by three main blocks. The first one is called movements generation block. The second is the block of the controller and dynamic model of the mobile robot. Third is the block of the kinematic model. Figure 12 illustrates the mobile robot control strategy implemented into Matlab Simulink blocks and than loaded in the embedded memory of the DSP processor by HIL(hardware-in-the-loop) technique.
The mobile robot embedded control is implemented with kinematic and dynamic model for axles control and the movement generator modules. Figure 7 illustrates the blocks diagram representing those modules.
The input system variables are:

- Δt is the period between one pose point and another.
- $\mathbf{TJ}_{ref}$, is the reference trajectory matrix given by supervisory control block with all the trajectory dots pose coordinates (x, y, θ).
- V_{ref}, is the robot linear velocity dynamics informed by supervisory control block so that robot can accomplish one particular trajectory.

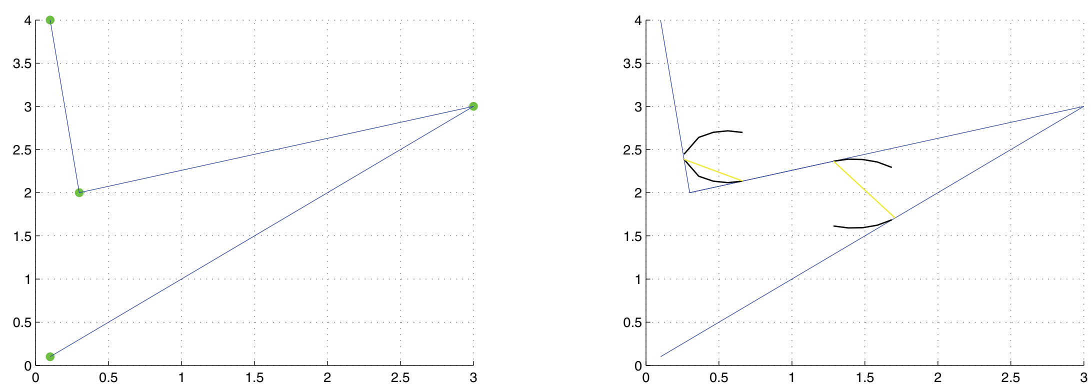
The embedded system output variables are:

- $\mathbf{TJ}_{din}$, that is the robot dynamic trajectory matrix, given in cartesian coordinate format.
- $\mathbf{V}_{din}$, is the dynamic linear velocity of the robot.

- $\mathbf{R}_c$, is the mobile robot ICC.
- θ, is the orientation angle.
- ω, is the angular velocity vector.

4.3 Trajectory Generation Block

The Trajectory Generation Block of trajectory receives some important points from the camera system so that the trajectory can be traced to be realized by the mobile robot. These points form a cartesian matrix containing more or less points, depending on the complexity of the environment. For testing reasons and for validation of the system, the number of points to be fed the system was fixed in four. Nevertheless, the number of points can be increased depending on the complexity of the environment where the mobile robot will navigate. Another data important to be used by the system have relation with the holonomics constraints of the modeled mobile robot. The bending radius must be informed to the system to be performed in the trajectory. A time that, for a reason or purpose tests, was fixed in four the number of cartesian input points, must supply the radius of the two curves to be executed for the robot. The information of distinct radius makes the system more flexible, making the trajectory able to be traced with different bending radius, depending on the angle of direction displacement and on the robot restrictions.



(a) Initial points given to trajectory generation. (b) Assay to the delineated trajectory.

Fig. 9. Initial points given by camera system and Assay to the delineated to trajectory.

The graphic of the Fig. 9(a) illustrates the initial points for the trajectory generator.
In this example, it was captured from the camera system and transmitted for the simulator the vector $\mathbf{x} = [.1 \quad 3 \quad .3 \quad .1]$ and the vector $\mathbf{y} = [.1 \quad 3 \quad 2 \quad 4]$, with the bending radius for first and the second semicircle given by the vector $\mathbf{r} = [.4 \quad .3]$. All the measures are in meters.
In the Fig. 9(b) it's possible to see the tracing of the straight lines, the semicircles and an intermediate segment of straight line indicating the start and the end of the tracing of each circular movement to be executed by the mobile robot.
The final tracing of the mobile robot trajectory can be observed in the Fig. 10, represented by red spots.

4.4 The virtual simulator implementation

Now the main characteristics of a simulator of mobile robotic systems are presented. It was implemented from the kinematic and dynamic models of the mechanical drive systems of the robotic axles, for the simulation of different control techniques in the field of the mobile

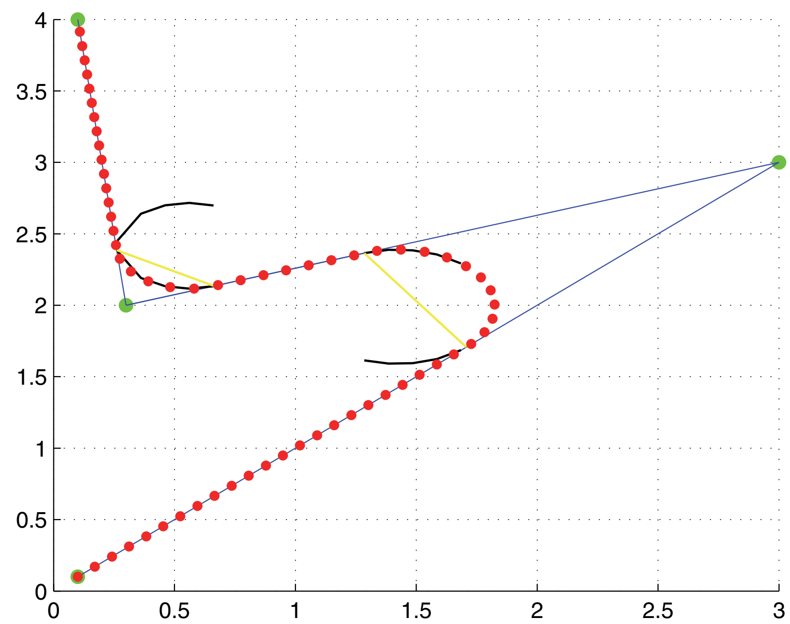


Fig. 10. The final tracing of the mobile robot trajectory.

robotics, allowing to deepen the concepts of navigation systems, trajectories planning and embedded control systems. This simulator, designed in modular and opened architecture, as presented in section 3, allows the direct application of some concepts into of the mobile robotics area, being used for its validation, and as main objective of this study, the model of an prototype of mobile robot with nonholonomic kinematic constraint and differential drive with two degrees of freedom (movement of linear displacement and rotation).

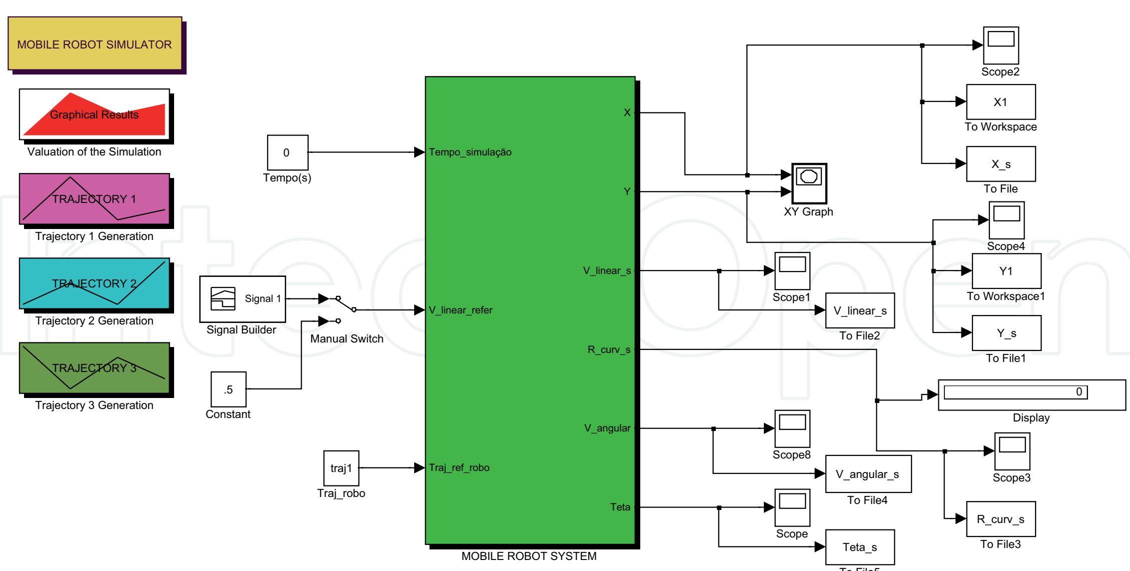


Fig. 11. The Virtual Simulator first page.

For the simulator development, the constructive aspects of the mobile robot prototype had been considered, including the kinematics and dynamics modeling of drive and control systems. The simulator presents the trajectories generating module that is the first block of

the system and was implemented with the functionality to generate a trajectory for the mobile robot from a matrix of points supplied initially. Another presented block is the controller, implemented in the PID traditional form.
The Fig. 11 shows the initial page of the Virtual Mobile Robot Simulator. The user can choose one of the captured trajectory for analyzing, take a look in the graphical results of the simulation or implement changes in the robot model, by clicking on the mobile robot system (green main block).
The virtual simulator system of the mobile robot is composed of three main blocks. The first one is called movements generation block. The second one is the block of the controller and dynamic model of the mobile robot and the third one is the block of the kinematic model. The Fig. 12 illustrates the mobile robot simulator implemented into Matlab Simulink® blocks.

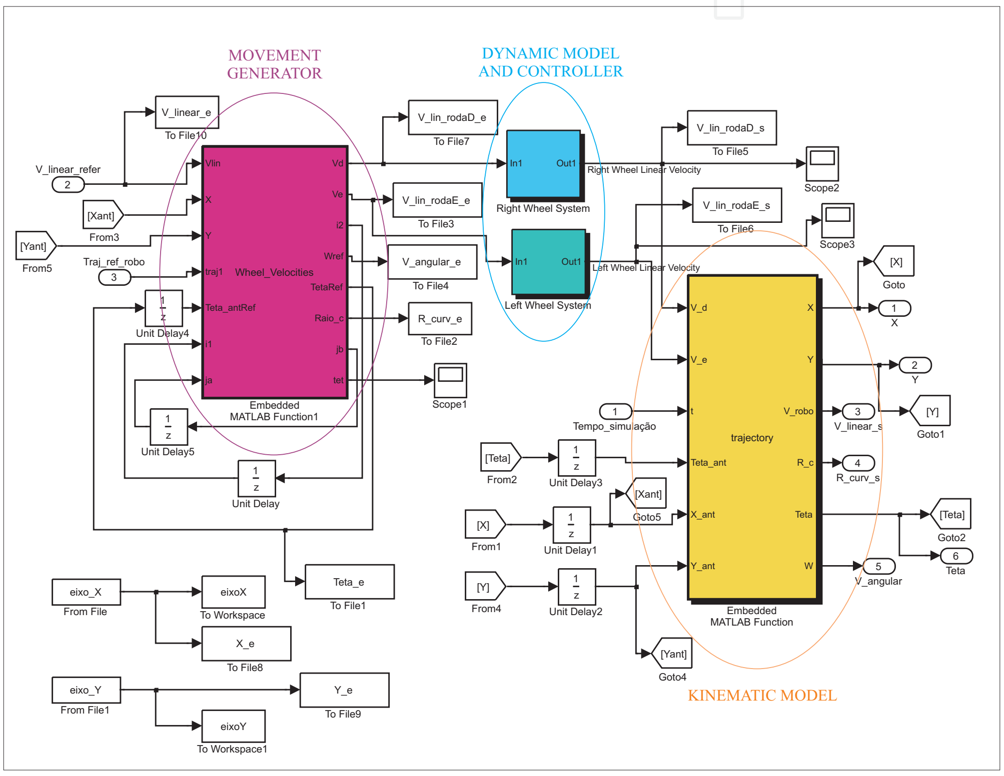
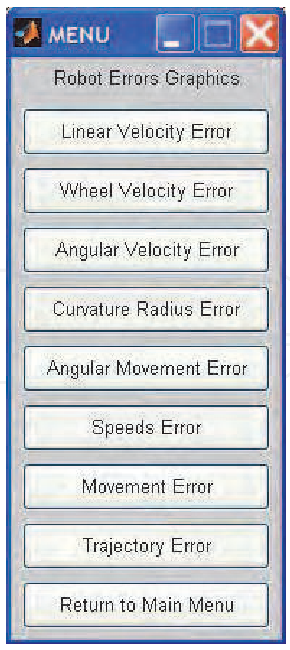


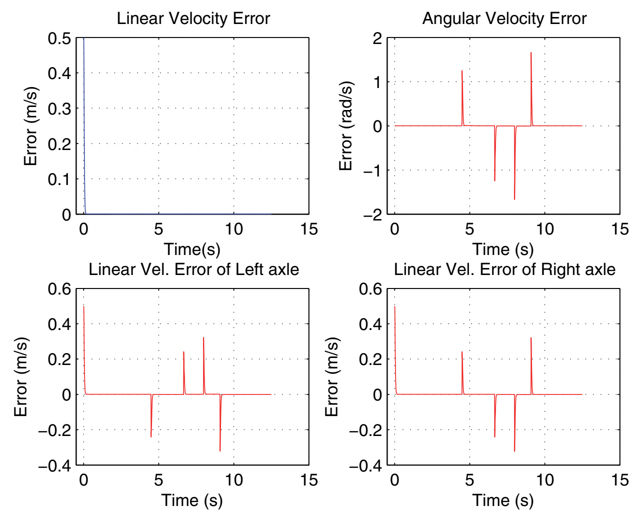
Fig. 12. Mobile robot simulator implemented into Simulink®.

4.5 Results graphical analyzer

The simulator implemented in Simulink® environment allows the visualization of the inputs and outputs of the system in study.
For a better understanding and analysis of the behavior of the system the implementation of a results graphical analyzer becomes essential. In this way, after realizing the simulations in the domain of the time, timings data archives are obtained corresponding to the study variables (angular and cartesian position, linear and angular speed and control signals), that after convenient treatment, make it possible to verify important results for better analysis of the system behavior. The Fig. 13 illustrates a menu of the graphical analyzer of the mobile robotic system in study with an example of generated graphic.



(a) Robot errors analyzer submenu.



(b) A robot speeds errors graphics.

Fig. 13. Submenu of the mobile robot graphical analyzer with an example of generated graphic.

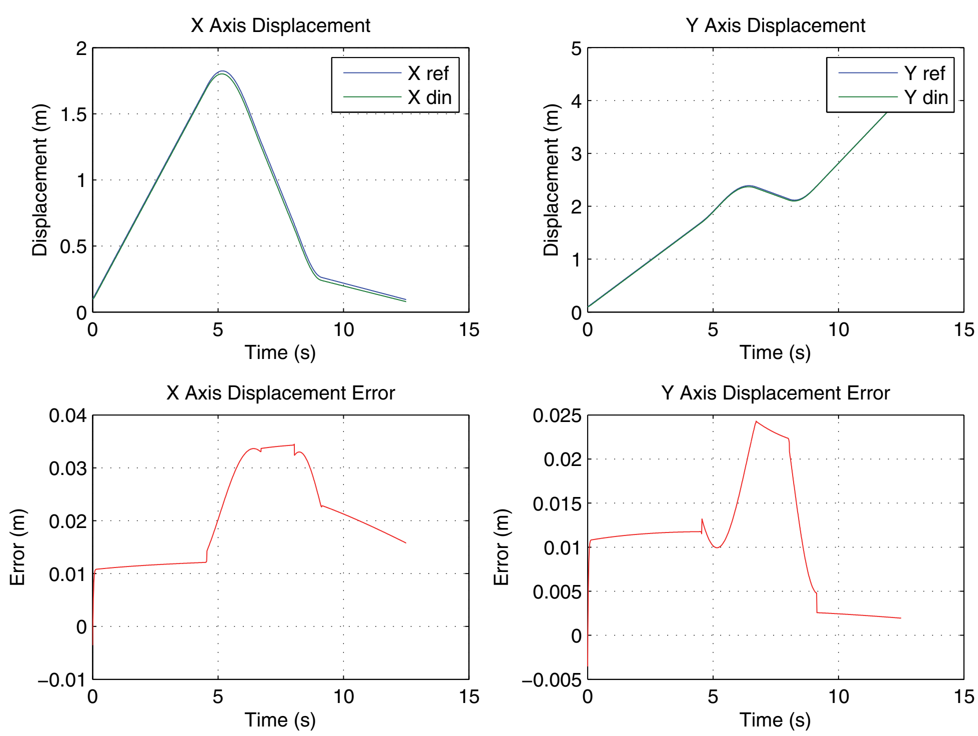


Fig. 14. Dynamic behavior graphics of the robot in the X and Y axis with their errors.

One kind of analysis that is made is in relation to the linear displacement of the robot in axis X and Y. In Fig. 14, the dynamic behavior of the robot with regard to these parameters, as well as the presented errors is illustrated.

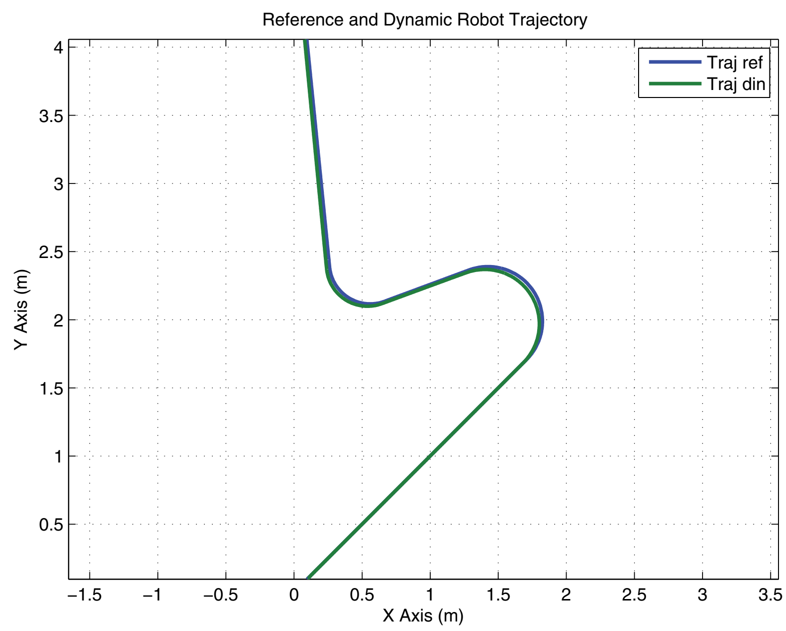


Fig. 15. Cartesian trajectory kinematics and dynamics of the mobile robot.

Another important graphic generated by the system, in the *cartesian trajectory* sub-menu, that is the graphic of the cartesian trajectory kinematics and dynamics of the mobile robotic system in plan XY. The Fig. 15, shows the dynamic tracing of reference and of the trajectory of the mobile robot.
The Fig. 16 illustrates the graphic of the trajectory error.

5. Mobile robot rapid prototyping

In the context of this work, the rapid prototyping is then the methodology that allows the creation of a virtual environment of simulation for the project of a controller for mobile robots. After being tested and validated in the simulator, the control system is programmed in the control board memory of the mobile robot. In this way, an economy of time and material are obtained, firstly validating all the model virtually and later operating the physical implementation of the system.
In this work was used a specific MatLab® tool box that can be used for its real time programming with *hardware-in-the-Loop* (HIL) techniques, that it is one of the simulation techniques utilized in the rapid prototyping systems for embedded mobile robotic controllers. This new technique of simulation, HIL (previously only available in the aerospace and aeronautical industry), can be used for the development and establishment of parameters of embedded mobile robotic controllers (Ledin, 2001).

5.1 HIL (Hardware-in-the-loop) simulation

The HIL technique of simulation is used in development and tests for real time embedded systems. HIL simulations provide a platform accomplish of development for adding the complexity of the plant under control to the tests platform. The control system is enclosed

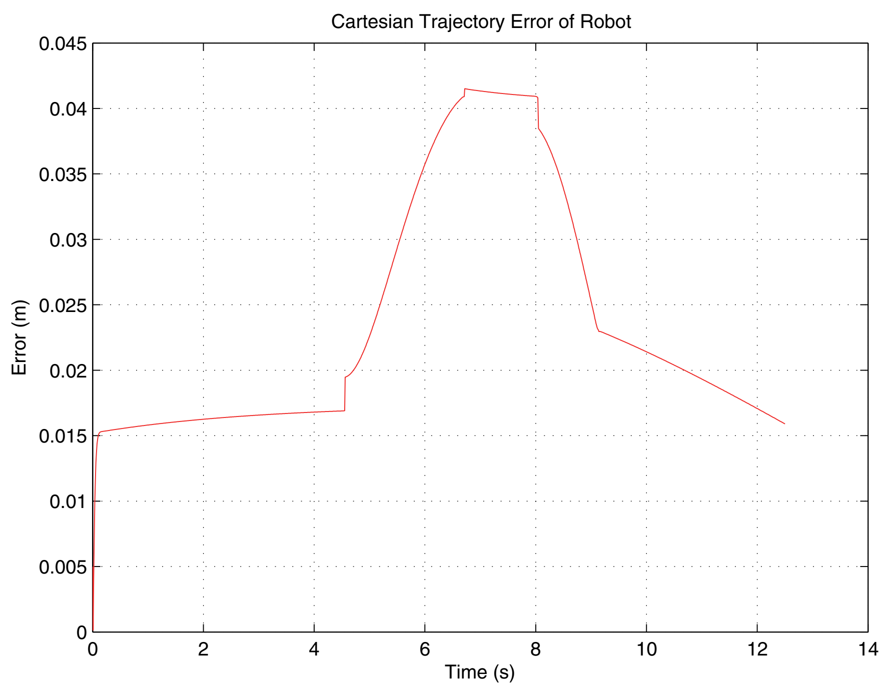


Fig. 16. Error of the kinematics and dynamics trajectory of the mobile robot.

in the tests and developments through its mathematical models representations and all the respective dynamic model (Melo & Rosario, 2006).
The Fig. 17 illustrates the use of the HIL simulation technique for real time simulation of the considered mobile robotic system.

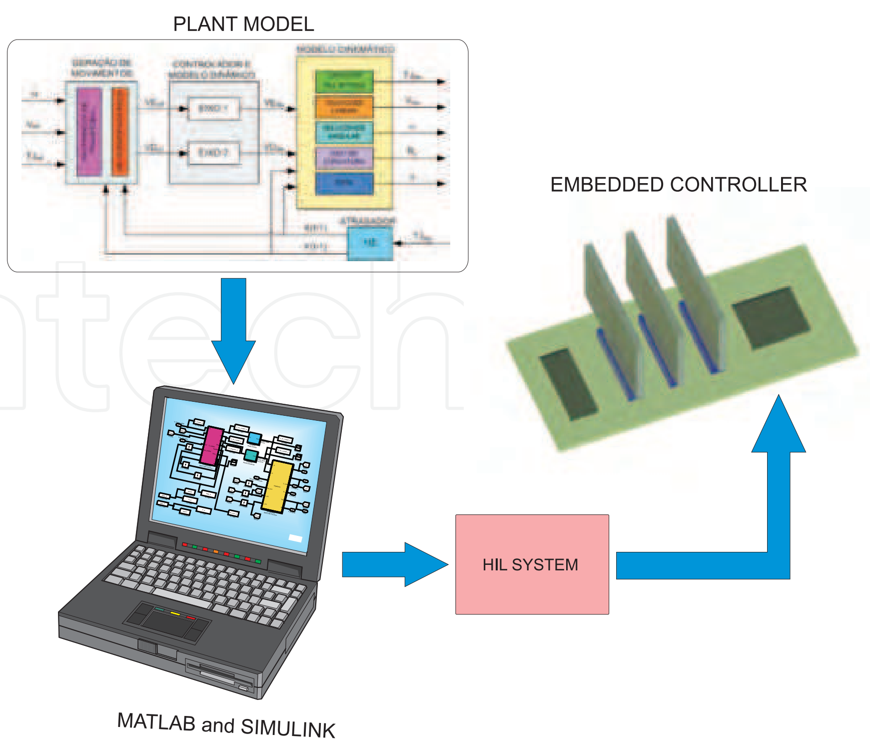


Fig. 17. HIL simulation for mobile robot system.

6. Experimental validation

The Aedromo, a didactic experimental environment for mobile robots, is an ambient used to test and validate the trajectory virtual simulator, the onboard robot control and the rapid prototyping system, all of them describe above. This environment is utilized for many others applications, like robotic soccer game, two or more robots interaction, etc. The supervision and the robots movements coordination are made through a close loop architecture based in a satellite camera over the arena. The information supplied by a video camera is sent for one or two computers for processing. The data obtained from this process are used to generate a sequence of instructions that are sent for the robots. The robots receive the instructions and carry out the actions obeying the predetermined tasks. The instructions are results of developed programmers strategies to execute the tasks and to realize the robot navigation into the environment. The Fig. 18 depicts this environment (Melo & Mangili, 2009).

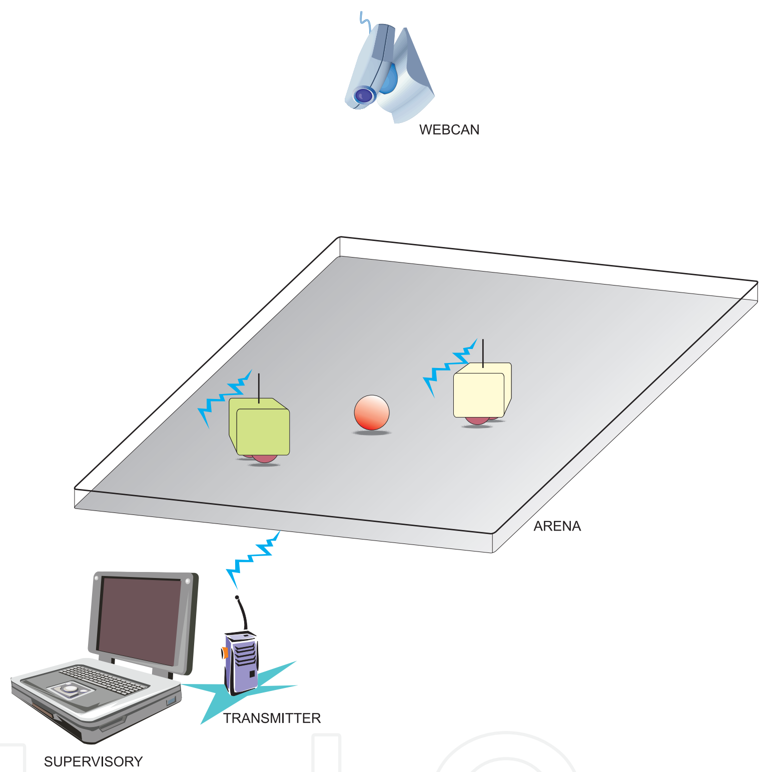


Fig. 18. The Aedromo environment.

The trajectory to be executed by the robot, from trajectory generation software, is illustrated on Fig. 10. This trajectory allows to get parameters of comparison between the real system, represented by de robot prototype, and the virtual system. On the onboard control of the mobile robot a dynamic and kinematic strategy was implemented, in order to find best trajectory, avoiding obstacles on the way, as can be seen in Fig. 8.
The graphic presented on Fig. 19 illustrates the difference of the reference trajectory, showing by the slim line with square blocs (violet) and the dynamic trajectory executed by the robot, showing by the thick line (blue).
Even though the robot executes the proposed trajectory with success, we can see errors between the two trajectory, mainly on curves. The cartesian trajectory kinematics and dynamics of the mobile on simulator was foreseen depicted on Fig. 15 with the trajectory error on Fig. 16. These errors are predictable for the simulator because of the intrinsic dynamic

characteristic of nonholonomic mobile robotics systems with differential drive wheels, which the prototype are made, and for the PID close-loop controller of wheels velocity axles. The necessity and efficacy of the virtual system, proposed by this work are demonstrated in this example.

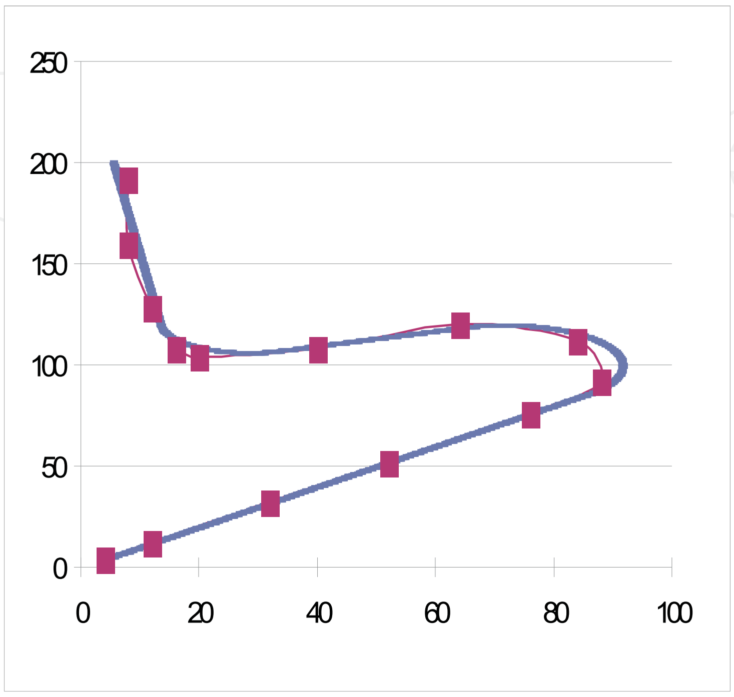


Fig. 19. Reference and dynamic trajectory executed by the robot.

7. Conclusion

How can be seen in this work, is evident the advantages of the HIL(*hardware-in-the-Loop*) and RCP(*Rapid Control Prototyping*) simulation techniques for mobile robotic control systems rapid prototyping. This development approach for embedded controllers is made in this work in the way that all the system is tested and approved in the simulator before being implemented at mobile robot. So that for the system has flexibility in its alterations is necessary that its architecture is opened and reconfigurable.
The techniques of rapid prototyping only can be executed if the plant has its kinematic, dynamic and the controller model implemented in a virtual simulator. In this work, MatLab software provides the necessary computational tools so that all system of simulation and rapid prototyping could be implemented.
The main objective of this work was to propose a generic platform for a robotic mobile system, seeking to obtain a support tool for mobile robotic control systems rapid prototyping. In this way, it presents the virtual environment implementation for simulation and design conception of supervision and control systems for mobile robots, which are capable to operate and adapting in different environments and conditions. This came from encountering the growing need to propose to the research that integrates the knowledge acquired in several domains that stimulates teamwork in order to reach a result. Another objective was to to improve knowledge in the mobile robotic area, aiming at presenting practical solutions for industrial problems, such as maintenance, supervision and transport of materials. Some promising aspects of this platform and simulator system are:

- Flexibility: 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 allowing implementation of sailing strategies for maps.
- The use of the rapid prototyping technique in mobile robotic systems.
- Possibility of modification of control strategies during the operation of the mobile robot in special mechatronics applications.

8. References

Beeson, P., Macmahon, M., Modayil, J., Murarka, A., Kuipers, B. and Stankiewicz, B. (2007). *Integrating multiple representations of spatial knowledge for mapping, navigation, and communication*. In Proceedings of the AAAI 2007 Spring Symposium on Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems, pp 1203–1201.

Braunl, T. (2008) *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*, 3th. Ed., Springer, Perth, Australia.

Dudek, G. & Jenkin, M. (2000). *Computational Principles of Mobile Robotics*, Cambridge University Press, UK.

Ledin, J. (2001). *Simulation Engineering: Build Better Embedded Systems Faster*. CMP Books, USA.

Graf,B; Wandosell, J. M. H. & Schaeffer, C. (2001). Flexible Path Planning for Nonholonomic Mobile Robots. *Proceedings of Angenommen zur Eurobot'01*. pp 456–565, 2001, Stuttgart, Germany.

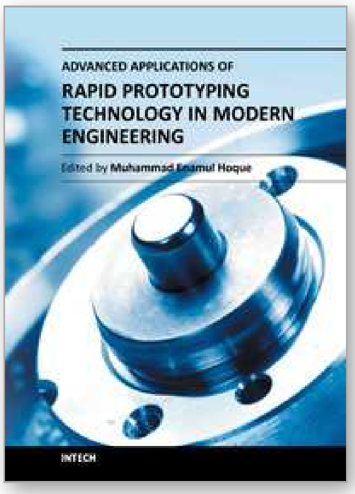
Melo, L. F.; Lima, C. R. E. & Rosario, J. M. (2005). A Reconfigurable Control Architecture for Mobile Robots. *Proceedings of 2nd. Internacional Symposium on Mutibody and Mechatronics - MuSMe 2005*, v. 1. pp. 1–8, september 2005.

Melo, L. F. & Rosario, J. M. (2006). A Proposal for a hybrid opened archtecture with hardware reconfigurable control applied in mobile robots. *Proceedings of IEEE International Conference on Robotic and Bionemetics - ROBIO 2006*,v. 1. pp. 1101-1106, october 2006.

Melo, L. F., and Mangili Jr., J. F. (2009). Virtual Simulator with Mobile Robot Rapid Prototyping for Navigation Systems. ICIA 2009. *IEEE International Conference on Information and Automation*, v. 1. p. 562–568.

Shim, H.-S.; Kim, J.-H. & Koh, K. (1995). Variable structure control of nonholonomic wheeled mobile robot. *Proceedings of IEEE International Conference on Robotics and Automation*. vol. 2, pp. 1694–1699, April 1995.

Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. (2009) *Robotics: Modelling, Planning and Control. Advanced Textbooks in Control and Signal Processing*. Spring, Napoli, Italy.



Advanced Applications of Rapid Prototyping Technology in Modern Engineering

Edited by Dr. M. Hoque

ISBN 978-953-307-698-0
Hard cover, 364 pages
Publisher InTech
Published online 22, September, 2011
Published in print edition September, 2011

Rapid prototyping (RP) technology has been widely known and appreciated due to its flexible and customized manufacturing capabilities. The widely studied RP techniques include stereolithography apparatus (SLA), selective laser sintering (SLS), three-dimensional printing (3DP), fused deposition modeling (FDM), 3D plotting, solid ground curing (SGC), multiphase jet solidification (MJS), laminated object manufacturing (LOM). Different techniques are associated with different materials and/or processing principles and thus are devoted to specific applications. RP technology has no longer been only for prototype building rather has been extended for real industrial manufacturing solutions. Today, the RP technology has contributed to almost all engineering areas that include mechanical, materials, industrial, aerospace, electrical and most recently biomedical engineering. This book aims to present the advanced development of RP technologies in various engineering areas as the solutions to the real world engineering problems.