Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Application and evaluation of Lego NXT tool for Mobile Robot Control

A. Valera[†], M. Vallés[†], L. Marín[†], A. Soriano[†], A. Cervera[†], A. Giret[‡]

[†]*Instituto de Automática e Informática Industrial*
[‡]*Grupo de Tecnología Informática e Inteligencia Artificial*
Universitat Politècnica de València, Valencia, Spain
P.O.Box 22012 E-46071. Tel. 34-96-3879570 Fax 34-96-3879579
e-mail: {giuprog, mvalles}@isa.upv.es, leomarpa@posgrado.upv.es, ansovi@maii.upv.es, adceran@ei.upv.es, adgibo@upvnet.upv.es

Abstract: Mobile robot control education is a motivating subject due to the practical work that it involves. For this reason it is very important the platform chosen for the laboratory activities with the students. This work will present different activities based on Lego NXT developed for teaching Control Engineering. The Lego tools have been used for the last 5 years for the practical works of "Mecatronica", a course of the last year of Computer Science Faculty of the Universidad Politecnica de Valencia (Spain). The tasks that the students must do during this course deal with system identification, dynamic control of the robot's wheels, kinematic control of the mobile robot, path generation etc.

At the end of the Mecatronica course, the students must fill an opinion poll about the work of the course, the tools used, etc. The paper will also describe the student's opinions, and they will verify that the Lego NXT is a very interesting and motivating tool. In addition, it is a very good platform for promoting the team work due to its multidisciplinary nature

1. INTRODUCTION

Robotics and, in particular, mobile robotics, is becoming successful (for high school and university courses) due to a combination of factors (Weinberg and Yu, 2003). The students are very motivated with robotics because they can physically experiment its work. So, using a suitable tool for the laboratory work becomes a very important decision.

Despite that usually the educational material for process control is very expensive, it is possible to buy different economical robot platforms (ActivMedia's Pionner robot (ActivMedia Robotics Web site), MIT's HandyBoard and Cricket controller cards (The Handy Board home page), The LEGO Group's LEGO Mindstorms (LEGO Mindstorms home page), etc.). These platforms usually consist of controllers, electronic sensors, low cost mechanical systems and/or small robots. These ones don't allow as much precision as an industrial robot but they are enough for the educational processes. Moreover, these platforms are very interesting for promoting the team work due to its multidisciplinary nature. So, each member of the team can be in charge of the development of a specific part of the work, as the physical development of the robot, programming its different subsystems, developing the strategies and response planning, etc.

This paper shows the benefits of using LEGO Mindstorms NXT as the platform for the development of practical works in mobile robot control education. This system, available since August 2006, has several advantages. The first one is the flexibility that this system has, since due to its sensors and actuators, and so to the programmable control unit, it is possible to develop a great variety of projects and activities.

Another advantage is the price: with the construction pieces and the electronic devices this platform has an approximated cost of 250 euros, which is much more economical than the price of other robotic platforms.

Unlike the previous version of LEGO based on the RCX that only admits the communications with an infrared tower, the new NTX version allows the communications by USB and Bluetooth. This way, the limitations of the infrareds are eliminated since much more longer distances are allowed, they don't need a correct orientation between the sender and the receiver and moreover, the communications are not lost if there is an obstacle between them.

This paper shows different activities where the control of mobile robots based on LEGO is established and the results of some surveys conducted amongst students that show the suitability of the selected platform for this kind of courses.

2. LEGO MINDSTORMS NXT

2.1 Introduction

In 1998 LEGO provide the first Mindstorms set; Robotics Invention System (RIS 1.0). Besides the typical LEGO pieces, the kit provides dc motors, sensors and, the most important thing, the RCX. RCX is the programmable "brick" that allowed not only the movement, but also sensing and acting over the environment. It is based on the microprocessor H8 from Hitachi, and provides analog to digital converters, serial communication and timers.

RCX was developed in collaboration with LEGO and the MIT (Resnick *et al.*, 1996), (Papert, 2000). On the first version, RCX provides 6 input/output ports; although later they were limited to the currently 3 ones for power consumption reasons.

As well as the extension of the hardware and software of this system in different ways (Baum, 2000), (Baum *et al.*, 2000), promoted by the enthusiasts, recently works based on this platform have been published in special editions of journals such as IEEE Robotics and Automation Magazine (Weinberg and Yu, 2003), (Klassner and Anderson, 2003), (Greenwald and Kopena, 2003) or IEEE Control Systems Magazine (Gawthrop and McGookin, 2004).

On January of 2006 the next generation, LEGO Mindstorms NXT, was introduced on the International Consumer Electronics Show. Besides other minor changes on the electronic sensors and the construction pieces, the new version incorporates a new control unit: the NXT.

As well as the previous version, it is possible to obtain a fast prototyping system for the development of a great variety of activities combining the construction blocks, easy NTX programming and its input and output interface. This allows that the system has been accepted as a tool for the university research and education.

2.2 LEGO Mindstorms NXT Components

From our point of view, the most important components of LEGO Mindstorms NXT are the control unit based on microcontroller, the electronic sensors and the actuators.

The control unit, named as the NXT intelligent brick, is based on a powerful 32-bits microcontroller: ARM7, with 256 Kbytes FLASH and 64 Kbytes of RAM.

For programming and communications, NXT has an USB 2.0 port and a wireless Bluetooth class II, V2.0 device. Moreover, it also incorporates a matrix graphical LCD screen of 100x64 pixels, a real sound loudspeaker and 4 buttons that allow an easy way of programming thanks to a very intuitive environment based on icons.

Also, NXT has 4 inputs (one of them provides an IEC 61158 Type 4/EN 50 170 expansion for future use) and 3 analog outputs, so besides having an additional input than in the previous version and taking into account that the encoders sensors are completely integrated on the electric actuators in this version, it is possible to connect more sensor devices.

The new LEGO version offers 4 electronic sensor types: touch, light, sound and distance. The touch and light sensors are an improved version of the RCX sensors. The new sound sensors can measure sound levels in decibels (dB) as well as in adjusted decibels (dBA). Besides the light sensors, NXT has other sensors that allow robots "to see": the distance sensors. They are ultrasounds based sensors, which allow measuring distances between 0 and 255 cm, with a precision of ±3 cm.

The last most significant LEGO components are the actuators. In this case, they consist on dc motors that incorporate, as have been mentioned before, encoders sensors integrated. These sensors have a 1-degree precision, which improves the previous version encoders' precision. Moreover, thanks to an inner gears assembly of the motor, this can supply high torque (16.7N.cm working at 117rpm with a power of 9V).

For avoiding the motor of very high intensities, these are equipped with a PTC resistance in serial mode with the motor, in such a way that its value rapidly increases when the temperature becomes higher. This way, the intensity supplied to the motor is limited.

A more detailed description about LEGO Mindstorms NXT motors can be found in http://www.philohome.com/motors/motorcomp.htm).

Fig. 1. LEGO Mindstorms NXT sensors (light, sound, distance and touch) and dc motor.

3. PROGRAMMING ENVIRONMENTS

Different environments can be chosen for programming the NXT microcontroller. The development environment selection depends on the complexity of the computations that the mobile robots are going to develop and the programming knowledge that one has.

The first option that can be used is to directly program the NXT with the buttons that are available for doing this. A system based on icons that allows programming very simple algorithms is available for this. Obviously, the applications that can be developed are very limited because the system is very simple and so it is not valid for the development of complex applications.

The second option for programming is the LEGO Mindstorms NXT Software application. This software is based on the LabVIEW motor of National Instruments and Mac OS X and Windows XP compatible. Using this environment it is possible to develop much more complex applications, being able to specify in a very intuitive graphical environment loops, jumps, motors and sensors control, sending messages by Bluetooth, etc. However, as the previous option, it is not suitable for subjects of technical or higher schools of industrial engineering or computer science.

A third environment for applications development available for LEGO mobile robots is ROBOTC. It has been designed by the prestigious Robotics Academy at the Carnegie Mellon University. The software can be used both with the NXT system and with the RCX one.

ROBOTC is a powerful programming language based on C that works in a Windows environment. With him, it is possible to write the programs that, after being compiled, will be downloaded and executed on the LEGO robot. ROBOTC is a cross platform that also allows debugging the robot applications in real-time.

This programming environment has different very interesting tools that, for example, allow configuring the actuators and sensors that are going to be used in a simple way, it provides a file administrator that enables to upload and download files on the NXT, to monitor in real-time the motors and sensors states and variables of the system, etc. ROBOTC also provides different low-level functions that enable, among other things (www.robotc.net), to verify and control the robot batteries level, to control the execution of the system tasks concurrently, to program timers that enable to specify the required sample times for the applications, system dc motors control and Bluetooth communications, to obtain the analog and/or digital sensors values, to provide a library with mathematic functions, to enable the interaction with the user from the NXT buttons and screen, to read and write text files, etc.

Once the source program has been developed and validated, the ROBOTC compiler generates the machine code that can be downloaded with an USB cable into the NXT brick. When it has been introduced into the mobile robot, the program execution can be launched.

Because the programming language used at the ROBOTC environment is similar to C, the complexity of the tasks that the robot has to do and the control algorithms can be higher. From the point of view of the university teaching, NXT programming allows to propose a very broad variety of interesting and motivating activities from the point of view of robotics.

Nevertheless, note that, although the NXT brick is based on a powerful 32-bits microcontroller, it is still a microcontroller. Due to this, when it is wanted to develop with the robots activities or control algorithms that need a great computational capability, that use complex mathematical operations or, for example, that need different sensors from the ones that LEGO offers, we can have problems.

With the objective of avoiding these problems, a PC can be used as a control unit instead of NXT. The PC calculates the control actions, which will be the reference positions for the right and left wheels (in the case of a mobile robot). Once the reference positions have been calculated, these are sent to the NXT by the Bluetooth device. In this way, a cascade control is established: at the PC the cinematic robot control is established, and at the NXT the dynamic control is kept, that corresponds to a PID control for the velocity of the wheels.

Since a general purpose PC is used for establishing a more complex control there are not problems related with the programs size, the number of variables, the mathematical complexity, the sensors used, etc. If it is needed, the PC has access to the different sensors of the mobile robots since it can obtain its value using the Bluetooth wireless communication system. For this kind of environment any programming language can be used, since the only requirement that is needed is that the PC could write and read dates from the Bluetooth device. For example, in http://www.answers.com/topic/lego-mindstorms can be found a broad list of the different programming languages that can be used.

4. MOBILE ROBOT CONTROL ACTIVITIES PROPOSED

Once the possible programming environments that can be used with LEGO Mindstorms NXT have been described, next, some applications that have been developed and used are going to be showed.

4.1 DC Motor Identification

Before we deal with the design or tuning of simple velocity or position controllers that allow obtaining more advanced controllers (based on the robot cinematic), the transfer function of the used motors needs to be obtained. Due to this, one of the first activities to be developed is to obtain the LEGO dc motor identification. For this, a program that introduces a step input to the motor can be written, storing in a vector the system outputs. In this case, it will be read the motor axis position using the incremental encoder built into the motor. For getting these values, the same ROBOTC program is in charge of writing in a text file the position values obtained in each loop step. Once the execution of this program has been finished, the values can be downloaded from the NXT to the PC using the USB cable, and afterwards they can be read directly from Matlab (for example) in order to analyze them.

In case that a dc motor identification in velocity terms is wanted to be done note that no sensor is available for measuring this magnitude directly. So, the easiest way to do this is to use a mathematical approximation of the velocity from the position values. Anyway, it is needed to say that if a simple first order approximation is done the obtained results will not be too much correct.

From the obtained values of position (and velocity) of the motors, the transfer function of the system can be obtained. For this, the Matlab/Simulink functions can be used that also enables to simulate and validate the obtained results.

Figure 2 shows the real system output (motor position) and the simulated output from the identified transfer function. Figure 3 shows the real velocity and the simulated one of the dc motor.

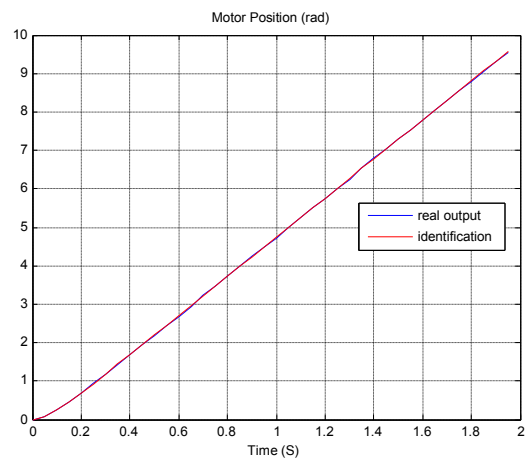

Fig. 2. Real and simulated position (from identification) of the LEGO dc motor.

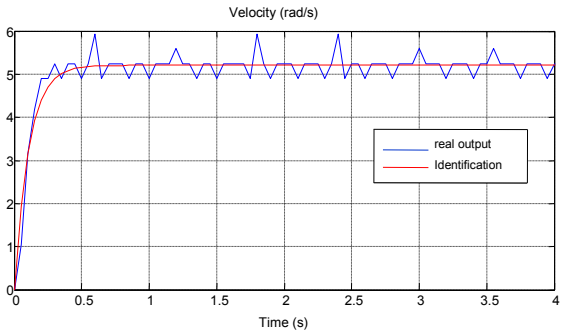

Fig. 3. Real and simulated velocity (from identification) of the LEGO dc motor.

4.2 DC Motor Control

From the values of velocity and position of LEGO motor and/or the transfer functions obtained it is possible to proceed to the implementation of the position and velocity controllers. In this case, the methods based on Ziegler-Nichols have been used. For this, the significant parameters of the process from which adjusting the controllers will be done have to be obtained.

The first tuning method is performed by a proportional controller. The proportional gain is increased (from zero) until the output of the control loop begins to oscillate. The critical gain KC (gain value that makes the system critically stable) and TC (repetition period of the system response) are obtained.

With the second tuning method, a step reference is provided to the system input, and three parameters are obtained: K (system gain), TP and T0 (times obtained from the system time constant).

Using these significant parameters, different position and velocity controllers have been adjusted. Figure 4 shows the real output and the reference for the system position with a proportional regulator. Figure 5 shows the motor velocity with a proportional-integer regulator.

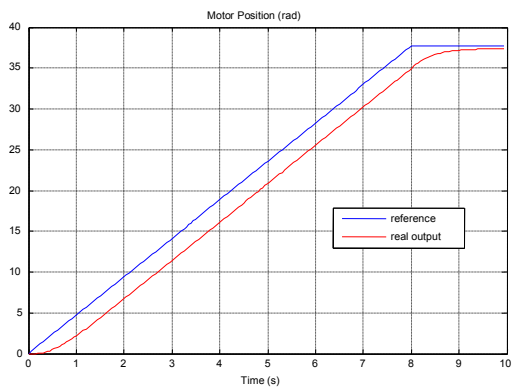

Fig. 3. Position control of the LEGO dc motor with a ramp reference.

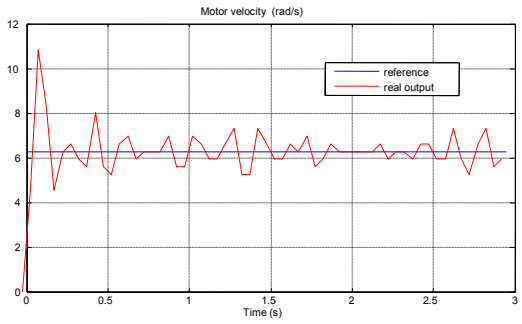

Fig. 5. Velocity control of the LEGO dc motor with a step reference.

4.3 Robot Movement Control

Once a simple position and velocity control of the motors have been established, it is possible to afford the kinematic control of the robot. To do this, the chosen robot configuration (differential or tricycle configuration, for example) has to be taken into account. It is possible to establish a great variety of different controllers, as for example the trajectory control based on decentralized point:

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = \begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \end{bmatrix} + \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \begin{bmatrix} x_{ref} - (x + e\cos\theta) \\ y_{ref} - (y + e\sin\theta) \end{bmatrix} \tag{1}$$

where $x_{ref}, y_{ref}, \dot{x}_{ref}$ and $\dot{y}_{ref}$ is the position and velocity references for the robot, x, y and θ is the robot position and orientation and e is the decentralized point distance.

To do this, a ROBOTC algorithm has to be programmed that at each step of the control loop performs the operations:

- Calculate the reference values for the robot position and velocity
- Estimate the value for the robot position from the previous position and the encoder sensors information, and the robot angular and linear velocities
- Calculate and apply the control actions from the trajectory control with decentralized point of equation (1), and store in a vector the reference and position values of the robot

Another kinematic control algorithm that has been implemented is the pure pursuit path controller. In this case, time is not considered at the path generation, so it is based only in the geometric description. Assuming a robot constant velocity (V), the wheels velocities can be calculated as following:

$$v_r = V * \left(1 + \frac{b}{2}\gamma\right) \qquad v_l = V * \left(1 - \frac{b}{2}\gamma\right) \tag{2}$$

where b is the distance between the robot wheels and γ:

$$\gamma = \frac{-2((x_{ob} - x_r)\cos\theta + (y_{ob} - y_r)\sin\theta)}{((x_{ob} - x_r)^2 + (y_{ob} - y_r)^2)} \tag{3}$$

For the γ term calculation, the robot localization (x_r, y_r) and the nearer path point (x_{ob}, y_{ob}) are needed.

Figure 6 shows two examples of the robot movement control based on decentralized point algorithm. It can be appreciated that the robot (red) can follow with quite precision the X-Y reference (blue) that has been specified.

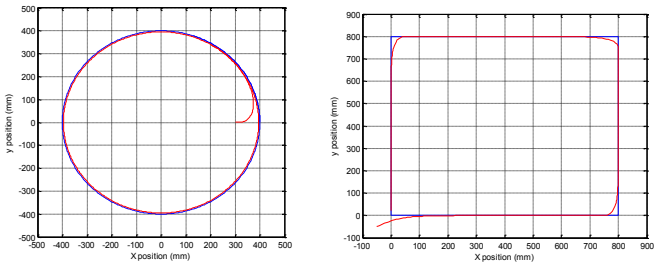

Fig. 6. Reference and real position of a LEGO robot for circular and rectangular references.

4.4 Control of Robots equipped with Advanced Sensors

With the original LEGO sensors interesting applications can be developed, as for example:

Control of pre-programmed movement with obstacles avoiding: the robot must be programmed for achieving different pre-established movements (using the position encoders), but if the robot detects (with the distance sensor) any obstacle, a maneuver for avoiding it must be done. Once it has been accomplished, the robot must follow with the movement that it had specified.

Cleaning robots: it is supposed that the robot is in a fenced and delimited, for example, by a black line, area. In this application the robot must clean as soon as possible all the objects that are found inside it. For detecting them a distance sensor is used, and for avoiding the robot goes out of the area the light sensor can be used.

Sumo fighting robots: it is an application very similar to the previous one, with the difference that, in this case, inside the area there are 2 fighting robots. The robot that goes out of the combat area will lose the combat.

In addition of the original LEGO sensors, nowadays a great variety of different sensor can be bought, as for example vision cameras, magnetic compass, accelerometers, gyroscopes, infrareds searchers, etc. (http://www.hitechnic.com/, http://www.mindsensors.com/). With these sensors (complemented with the sensors described before), more interesting applications set can be developed, as for example:

Football player robots: each robot is equipped with an infrared seeker sensor that specifies to the robot in which area and at how much distance the infrared light source is found. For achieving this application an infrared ball must/can be used and other sensors as the electronic compass or the light sensor.

Objects searching by artificial vision: as has been mentioned, nowadays are available some artificial vision cameras that can be connected directly to the input ports of NXT. Thanks to these, it can be implemented, for example, applications where the robot has to search or follow to objects of a concrete colour.

The following figures show several pictures of the different applications that have been developed. Figure 7 (a) shows a photograph of the robot following a ball through the information obtained with the camera. Figure 7 (b) shows a photograph where the robot uses the camera for following the trajectory delimited by a black line.

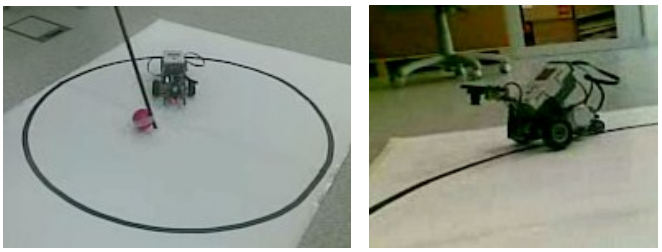

Fig. 7. Some robot applications using artificial vision.

4.5 Mobile Robots Coordination and Communications

The synchronization and coordination between robots is an area of widespread research. Using Bluetooth communication of NXT robots, different architectures to coordinate the movement of several robots, can be developed.

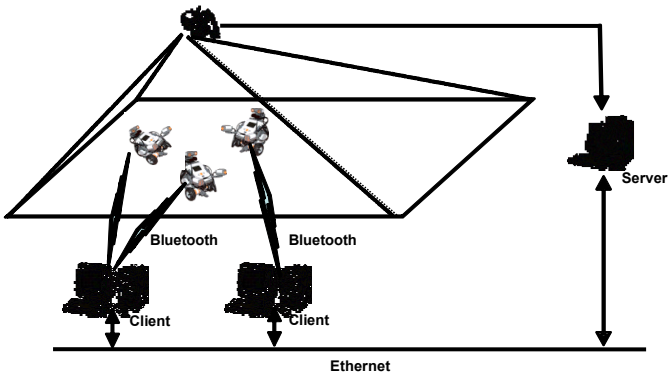

Fig. 8. Control architecture developed.

The control architecture of Figure 8 was carried out. A computer server captures images from a webcam that records the situation where robots cover. The server processes the image and it sends the relevant coordinates to others computers via an Ethernet network. The client computers send data to the NXT robots via Bluetooth. Three subnets compose this architecture: one Ethernet subnet between the computer server and the client computers, and two subnets Bluetooth between these computers and NXT robots. Different applications with a similar architecture have been developed: *The Ring* and *The Maze*.

In *The Ring* application, three robots constantly revolve around a circle with a given radius and centre. At any time, the operator can change the radius and centre parameters.

NXT robots are connected via Bluetooth to a computer. One difficulty that arises is to avoid the collision between the three robots in motion. Using the on-board distance sensors, they coordinate their speed with the obstacles detected on their route to avoid collision between them. It's a greater difficulty when a change of the centre of circular path is achieved, because every robot must move to the new area and resume their path avoiding potential collisions.

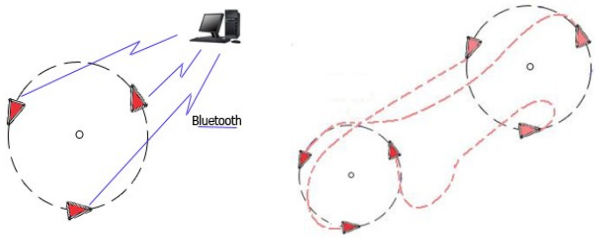

Fig. 9. Example of the Ring application.

In The Maze application artificial vision is used for the automatic path generation. The application incorporates a webcam that is not only used to obtain the robot position and orientation, but also to detect the obstacles. The server obtains the images from the camera and it sends them to the client computers. These computers calculate an optimal path (minimum length) with obstacles avoidance by cell decomposition methods (Nourbakhsh and Siegwart, 2004). For the path generation, the system calculates Splines curves. The trajectories are sent to the robots, so they use them as references for the kinematic control algorithms.

Figure 10 shows three snapshots of the Maze application. In the first one it can be seen the robot, the goal position and the obstacles. In order to make the image processing easier, the robot incorporates a black triangle, and the goal position is an orange triangle. The triangles are used to obtain also the orientation. The obstacles have different shapes, but it is assumed that there are blue objects. The second snapshot is the output of the image processing. The last one shows the path generation obtained.

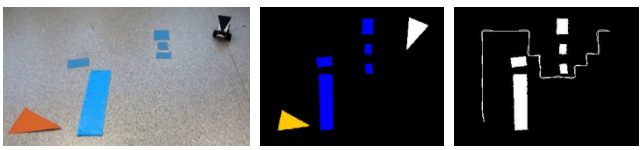

Fig. 10. Different snapshots of The Maze application.

Nowadays we are working with the implementation of a new application for the robot coordination. It is based on a multi-agent system in which individual software agents interact with each other in cooperative/competitive manner pursuing a global goal. JADE (Jave Agent Development) framework is used for the development of the multi-agent system. It is a software Framework fully implemented in Java language that it simplifies the implementation of multi-agent systems through a middle-ware. Lego robots are programmed in Java using the Java programming environment leJOS NXJ for the implementation of the agents.

5. RESULTS OF THE SURVEYS

At the Polytechnic University of Valencia, two kinds of surveys are conducted amongst students. The first one deals with the teacher quality but also involves some questions about the learning process, the evaluation model from the students' point of view and if the course has been motivating. The results are interesting due to the fact that it is possible to compare with the department mean.

Table 1 contains the registered students of "Mecatronica" course.

Table 1: Registered students of *Mecatronica*

Year	2005	2006	2007	2008	2009
Registered students	31	33	30	22	32

At the following figure, the comparison between the course and the department mean about if the course has been motivating is showed. As can be seen, in all cases the value obtained that indicates if the course has been motivating is higher than the department mean.

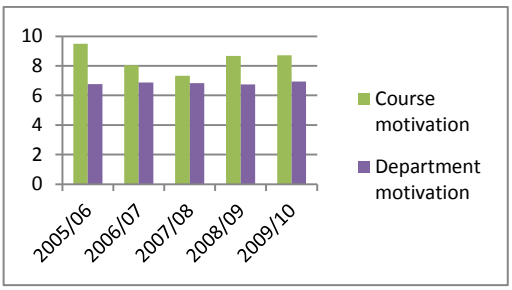

Fig. 11. Comparison between course and mean department motivation

The second kind of surveys deals with the quality of the laboratory tasks. Since the mobile robotics subject is a very practical one, the results of this survey are very important for obtaining feedback from the students' point of view. It contains 13 questions related with different issues as quality, scheduling and resources used for the laboratory classes. Here, the ones that we consider the most relevant are commented.

For example, question number 2 asks the student if the laboratory work emphasizes the application of the most relevant concepts of the subject and the results through the last four years are showed in figure 12.

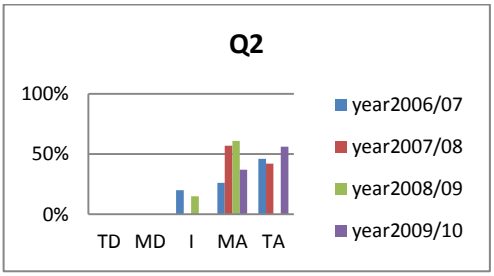

Fig. 12. Last four years evaluation about the application of the most relevant concepts emphasis at the lab work

The acronyms that appear at the x axis of the graph indicate the level of agreement with the assertion. So, *TD* means totally disagree, *MD* mostly disagree, *I* indifferent, *MA* mostly agree and *TA* totally agree.

Next question that we consider relevant is number 3 that asks the student if the laboratory work can be considered motivating and the results are showed at figure 13.

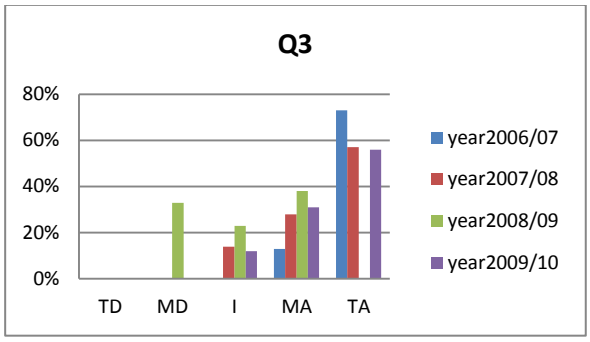

Fig. 13. Last four years evaluation of the lab work motivation

Another important question is number 4 that ask the student if the laboratories of this survey can be considered as good laboratories. The results are depicted at figure 14.

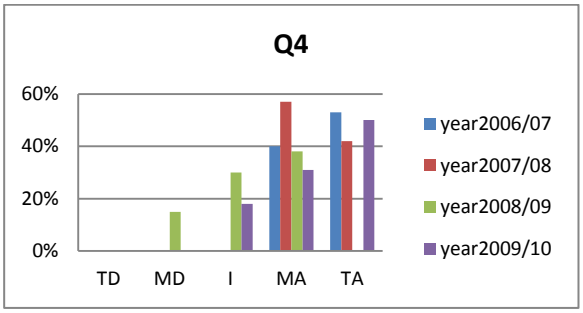

Fig. 14. Last four years evaluation of the lab quality

Finally, question 5 reflects the students' opinion about using of LEGO Mindstorms for the laboratory work since the original question ask if the resources used fits in what the students expect from the subject. The results are showed at figure 15.

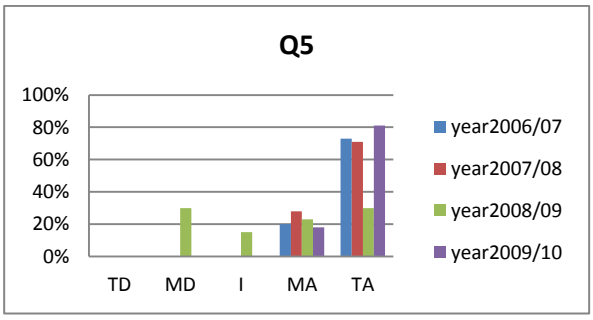

Fig. 15. Last four years evaluation of the LEGO Mindstorms suitability for lab work at mobile robot control education

So, in conclusion from these graphs, the LEGO Mindstorms as platform for the laboratory work is considered as suitable, motivating and that gives quality to this work by the students.

6. CONCLUSIONS

Mobile robot control is a discipline that needs practical work at the learning process. In order to carry out it at the laboratory with the students, the LEGO Mindstorms NXT system is proposed. This platform provides basically 2 advantages: its low price and good features.

Because a wide variety of sensor is available, different control activities can be done: from simple system identification tasks until advanced real-time robot control based on artificial vision, gyroscopic or accelerometers, etc.

This paper has presented various practical activities. With them it is very easy to design robot competitions tests, a popular and effective way to motivate students. This has been reflected at the results of two kinds of surveys achieved by the Universidad Politécnica de Valencia.

ACKNOWLEDGEMENTS

This work has been partially funded by the Ministerio de Ciencia e Innovación (Spain) under research projects DPI2008-06737-C02-01 and DPI2010-20814-C02-02 (FEDER/CICYT) and the Vicerrectorat d'Investigació of the Universitat Politècnica de València under project PAID-05-10-2620. Also, the financial support from the University of Costa Rica is greatly appreciated.

REFERENCES

ActivMedia Robotics Web site [Online]. Available: http://www.activrobots.com

Baum, D., (2000). *Definitive Guide to LEGO Mindstorms*. Berkley, Ca: Apress.

Baum, D., M. Gasperi, R.Hempel, and L. Villa (2000). *Extreme Mindstorms*. Berkley, CA: Apress.

Gawthrop, P., E. McGookin (2004). A LEGO-Based Control Experiment. *IEEE Contr. Syst. Mag.*, **24, 5**, pp. 43-56.

Greenwald, L. and J. Kopena, (2003). Mobile robot labs, *IEEE Robot. Automat. Mag.*, **10**, **2**, pp. 25-32.

Klassner, F. and S. D. Anderson, (2003). LEGO mindstorms; Not just K-12 anymore, *IEEE Robot. Automat. Mag.*, **10**, **2**, pp. 12-18.

LEGO Mindstorms home page [Online]. Available: http://mindstorms.lego.com/

Nourbakhsh, I. R., R. Siegwart (2004). *Introduction to Autonomous Mobile Robots*. Campridge MIT Press, MA, 2004.

Papert, S., (2000). What's the big idea? Towards a pedagogy of idea power, *IBM Syst. J.*, **39**, **3&4**, pp. 720-729.

Resnick, M., F. Martin, R. Sargent, and B. Silverman, (1996). Programmable bricks: Toys to think with, *IBM Syst. J.*, **35**, **3&4**, pp. 443-452.

The Handy Board home page [Online]. Available: http://handyboard.com/

Weinberg, J.B. X. Yu (2003). Robotics in Education: Low-Cost Platforms for Teaching Integrated Systems. *IEEE Robot. Automat. Mag.* **10, 2**, pp. 4-5.