A Low Cost Mobile Robot for Engineering Education

Carlos Cardeira, José Sá da Costa
Grupo de Controlo, Automação e Robótica – IDMEC
Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa
{cardeira, sadacosta}@dem.ist.utl.pt

Abstract. **Robotic competitions are becoming widely used on education. The use of products from the market makes building robots possible at a very low-cost awakening the engineering nature of students and researchers. In this paper we present a low cost mobile robot which uses a normal laptop to guide itself through a track. The low cost robot involves the use of one or more cameras to control the position of the robot's position relative to the environment. Motors are from battery operated screwdrivers and the interface with the laptop is based on an USB card, recently made available from a data acquisition cards manufacturer. The goal for this work is to build a large set of these robots, lend them to the engineering students and make them participate in a local university contest. Moreover, along the years, the student integrates more and more knowledge in the robot allowing it to perform as better as their knowledge increases along the course.**

I. INTRODUCTION

The use of products from the market for robot components, together with some engineering spirit, enables the emergence of low-cost robots for many application areas.

In this article we suggest the use of these robots for education. University lends the robots to students to let them freely use their own laptops to control the robot, independently of classes, and join in the end of every semester for a local competition, like some competitions already existing in many universities [Schilling 2002], [Bruder 2003]. Moreover, according to the level of the students, higher challenges are made to the students letting them feel how their increasing knowledge actually improves the robot performance.

This experience is being conducted in the Control, Automation and Robotics Group involving students both from Mechanical and Computer Science Engineering and follows some experience already acquired on building robots for competitions [Lima 1998], [Camara 2000], [Almeida 2000], [Freitas 2005].

II. MOBILE ROBOTS – STATE OF THE ART

In the state of art of mobile robots morphology we may mainly find legged and wheeled robots. Legged robots were always very challenging due to their likeness to living beings way of locomotion.

Early in the 80's, 3D Hopper [Murthy 1983] was an amazing one legged robot able to move and stabilize itself over a single leg. [Zeglin 1991] presented Uniroo which was kinematically similar to a real kangaroo. Nowadays, legged robots are subject of intense researches with important budgets mainly from Japanese companies. Honda Humanoid Robots P3and Asimo [Honda 2003], [Sakagami et al. 2002], and Sony's Qrio [Geppert 2004], [Sony 2004] are good examples of such humanoid robots.

With more than two legs, we may find quadruped robots like Tekken II and Patrush II [Fukuoka 2003], [Kimura 2001], [Kimura 2003], Sony's Aibo [Kaplan 2001], [Sony 2004], and several works with robots with much more legs or inspired in biologic insects [Delcomyn 2000].

Much simpler and easier to build and control, robots with wheels are the simplest for a beginner. Some topologies exist varying the number of wheels, the type of steering and driving. We may find four wheeled robots, but the simplest solution is to build two-wheeled robots which usually have two driven wheels near the centre of the robot, along with a caster wheel. The two driven wheels are able to make the robot steer and drive in almost any direction. Robots with omnidirectional wheels are by far the most agile but somewhat more difficult to control [West 1995].

III. DEVELOPMENT

As stated before, low cost was an important goal when building these robots. The processing unit is a normal laptop. Laptops are more and more common among university students. It is very common that among two or three students in a group, at least one as a laptop. So the students use their own laptops to control the robots. Students complain less about the equipment when they use their own laptops, which in many cases are much updated and powerful than those the department could afford to them. They are responsible for damages so they pay a lot more attention to the programs they run. Hence, we did no count the laptop cost on the robots cost.

Moreover, the robot uses a normal webcam which the students usually already possess. Anyway they are very cheap and we counted their cost on the robot. We did not include the costs of the software licences (Visual Studio, Matlab/Simulink, Labview, etc) because we use either campus or students licences.

In the following sections we present the several choices and costs. The cost depends on the choices of some components but the overall cost of the robot is under 200 Euros.

Lego Mindstorms are a common solution for low cost robots. However the number of I/O ports they provide is very limited and the processing capacities and languages available can't be compared, in number and in quality, to the languages and operating systems available for laptops. Tasks like image processing, visual servoing, PID tuning can be done with a normal laptop but are not straightforward or even impossible with actual Lego Mindstorms.

Small microcontrollers like Microchip or Atmega families provide a very interesting alternative, but again the programming capabilities are limited to the existing cross compilers available. Moreover they provide good solutions for analog or digital I/O ports but not for handling image processing cameras. It is actually possible to do that [Kaiser 2001], but they are not as straightforward as connecting a webcam to a USB laptop port.

A. Morphology and Motors

As we showed there are several morphologies of robots. Trying to keep it simple, we decided to build a robot with three wheels. Two wheels provide differential driving and the other wheel is free (caster wheel). This simple structure allows a high mobility of the robot.

As stated before, the use of products from the market enables the construction of low cost robots. A differential driving robot will need two motors. Very likely it will need also a gear boxes to reduce the motors speed and increase the torque. It will need batteries. Hence, a battery charger will also be needed to recharge the batteries. Bought individually, these components would lead to an expensive robot.

Fig. 1. Electrical screwdriver disassembled.

Fortunately, in the market, there are very low cost battery operated electrical screwdrivers, proving it all in a single set, which may be adapted to a low cost mobile robot.

These screwdrivers (or battery operated drills), can be bought for just a few euros (for the one presented in figure 1, the cost was 7.5 Euros). When disassembled, they include a DC motor, a gearbox, batteries and charger. Buying any one of these products in the specific robots market would immediately exceed the cost of the electrical screwdriver.

Figure 2 shows the mobile robot mechanical structure made up of two electrical screwdrivers. In the image, it is easily seen that the screwdrivers were cut; the batteries, which were inside the screw-driver, were put along the robot, to keep the robot with a reason-able size.

Fig. 2. Robot mechanical structure.

B. Data acquisition and control card and power interface

Nowadays many students already have their own laptop computer. They have plenty of accessories, like webcams, which mainly interface the computer through USB ports which are rapidly replacing the old serial and parallel interfaces.

However, regarding data acquisition and control cards, usual PCI cards are not suitable for laptops and PCMCIA data acquisition cards are still very expensive.

Fortunately, new cards have recently been introduced in the market providing at a low cost, typically around 100 USD, digital and analog inputs and outputs [MC 2004]. These cards interface the personal computer through USB ports, allowing their use with common and portable laptops (see Fig 3).

Fig. 3. Example of a new USB card [MC 2004].

To interface the card with the motors a very simple MOSFET based driver circuit was developed to make the interface between the PWM output of the card and the circuit that powers the motors through the batteries. When using battery operated variable speed drills, we could use the power circuit inside the drill to drive the motors.

Hence, we used a USB card to interface the laptop with the motors. There are plenty of free I/O ports in the acquisition card that might be used to cope with analog or digital sensors that may increase the ability of the robot to sense the environment.

Figure 4 presents a general view of the built robot. The first prototype was named “Rasteirinho”

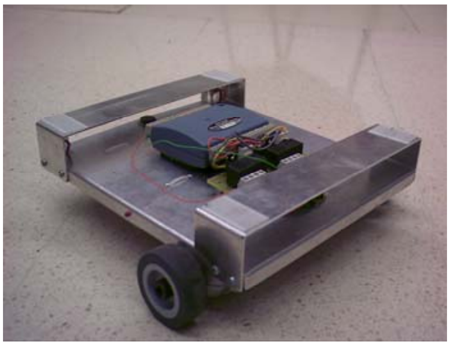

Fig. 4.The robot structure.

C. *Processing unit*

The processing unit is a normal laptop that is placed on the top of the robot. The advantage of using a normal laptop is that it is easy in a group of two or three students to find one with a laptop. Using their own laptops and interfacing it to the robot (which is lent to them), students may work with it at any place and hours independently of the availability of the laboratory.

The laptop is fixed to the robot by easily removable Velcro bands (in fig. 4, four Velcro bands are visible on the top robot corners).

D. *Sensing*

To sense the environment, students might use sensors that are widely available in the market, and could interface them to the USB card free I/O ports. However, keeping it simple and challenging, we decide to guide the robot through normal webcams, widely available at a low cost (less than 20 USD). The USB camera acts as a general purposed sensor to guide the robot through the environment. A low-cost, flexible link, web camera for guiding the robot is presented in fig 5.

Fig. 5. Web camera for guiding the robot.

Fig. 6 presents the robot, together with the laptop, ready to be programmed.

Fig. 6. The complete robot.

E. *Programming*

Using a normal laptop we are not limited to specific languages. Any language available for PC with drivers for the USB card and able to read webcams might be used.

1) Programming the USB Card

The USB card comes with a so called universal library that allows programs written in many languages to be used. In particular, among others, it provides libraries for Visual Basic and C++, Borland C++, Delphi, Watcom C++ and .NET languages like VB.NET and C#.NET. Drivers are also available for Matlab and Labview.

Our first choice was C#.NET, because it becomes widely used and there is plenty of information about how to start. Beginners that read [Foxall 2002] were able to do simple programs to test the motors. A simple program to send analogue output orders to each motor allowed us to do a program to go straight, turn left or right and stop.

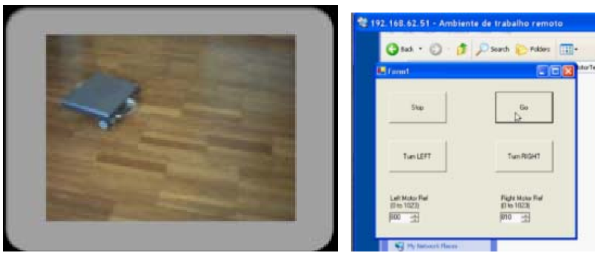

Fig. 7. Robot remotely controlled.

Using a wireless communication with the laptop the robot was easily remote controlled. The robot’s laptop was connected to the internet through a wireless adapter (actually many laptops have, nowadays, built-in wireless adaptors). With a normal desktop computer we were able to initiate a remote desktop session in the robot’s laptop using windows XP standard features. Fig. 7 shows the robot “Rasteirinho”, remotely controlled, doing basic operations like go straight, turn left or right and stop.

2) Image acquisition

To let the robot sense the environment we looked for already developed code in C# for acquiring the image to an array in C#. In [Pierce 2003] one may find the tools for grabbing an image from a webcam for later processing.

Fig. 8 shows the user interface of an open source program to acquire the image from a webcam [Pierce 2003].

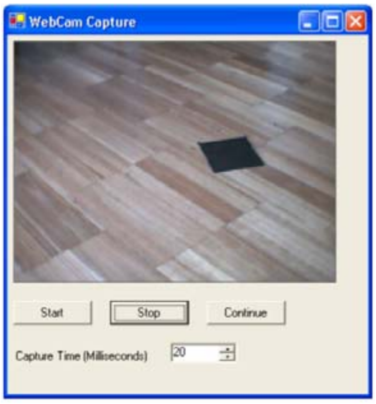

Fig. 8. Webcam open source program [Pierce 2003]

3) Image Processing

The simplest processing algorithm was the computation the centre of mass of the binary image, for which there are well known algorithms available on image processing literature [Ritter 1996].

Considering that $a(x, y)$ are pixels of a $(n \times m)$ image, the coordinates of the centre of mass $(\bar{x}, \bar{y})$ are just given by:

$$\bar{x} = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m} x.a(x.y)}{\sum_{x=1}^{n}\sum_{y=1}^{m} a(x.y)}$$

$$\bar{y} = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m} y.a(x.y)}{\sum_{x=1}^{n}\sum_{y=1}^{m} a(x.y)}$$

There was no need of processing the 640x480 pixels of the image grabbed by the webcam, because it would be too much time consuming without significant increase of accuracy. Processing one on each ten samples both horizontally and vertically (64x48 pixels) we were able to perform a precise enough centre of mass computation.

Figure 9 shows the processed image. First, on every ten pixels, one is set to black or white if the computed grey level of the picture is higher or lower a given threshold.

Then the centre of mass of the binary pixels is computed according to the equations. For a correct threshold level, a cross is drawn in the picture to indicate the centre of mass, which will approximately be in the centre of the target.

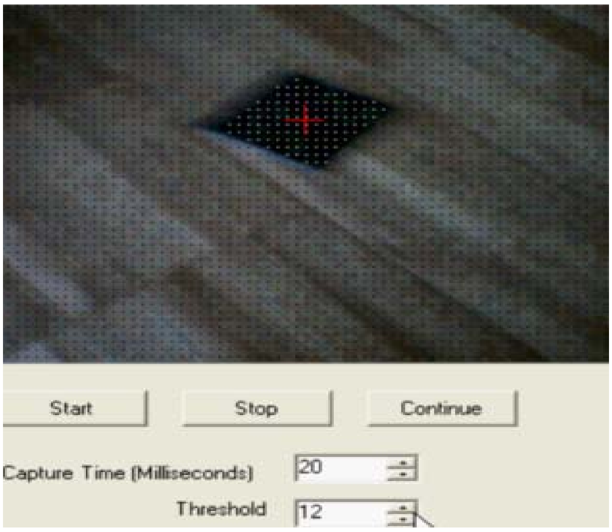

Fig. 9. Processing the image.

Fig. 9 shows the result of the image processing. The red cross in the target centre indicates the computed mass centre of the binarized pixels.

IV. INTEGRATION

Software modules containing the image processing system and the modules for driving the motors were given to the students and made available in a web page. It would be up to the students to integrate the modules building a control program to make the robot follow a given target in the floor. The robot autonomously acquires the image, computes the centre of mass of some lines, computes the error to the centre of the image and computes the output for the motors using a straightforward proportional controller. With the overall integration we were able to measure the performance of the system. The control loop includes image acquisition, centre of mass computation, running the proportional controller and actuating the motors through the USB card. Running on top of windows XP, the control loop was performed at 7Hz rate using a Pentium III 800 Mhz laptop. With a Pentium IV 2.4 GHz the control loop was performed at a 23 Hz rate. In the former case, the main limitation was on the image acquisition rate but with a normal laptop we almost achieved the maximum frame rate of the camera (25 frames/s). This rate was enough to make the robot follow the track smoothly.

Fig. 10. Following a continuous line.

Fig. 10 shows the robot autonomously following a continuous line.

V. RESULTS

The minimal setup shown allows the students to have a standard departure point that allows the robot to fulfil some of the tasks with little changes. However innovation is encouraged (otherwise it would be just reinventing the wheel all the time). Regarding innovations, students already proposed solutions programmed in matlab, others proposed solutions with simulink and realtime windows target, one proposed a microchip microcontroller version, with optical digital sensors, that was able to follow the line in a bang-bang manner, others used Labview and their image acquisition toolbox. So there is still space for innovation, mainly because the platform is generic and can support a large number of languages and operating systems.

A local robotics competition is organized in the end of each semester. We are now on the third edition. In the previous editions, more or less ten teams participated. The limitation is connected to the reduced number of available robots, actually five. Every end of semester the number of robots grows. With a number of 40 robots we expect to attend a class of 120 students. Till now, only the small classes participated.

Till now we had teams with students from Mechanical and Computer Science Engineering departments. We encourage mixed teams from freshman to seniors of the same or different courses. Control and Systems Identification, Signals and Systems and Industrial Automation courses already participated in this contest.

In the competition, the students are not evaluated individually (only the robot is), so the laptop owner has the possibility of dedicating more hours to and taking more advantage of the experiment. This can cause the other members of the team to rely on him/her for doing most of the work. This can be detected by the courses teachers which may grade the students individually. The competition just provides an information about the robot performance and the grade the students get will depend on much more criteria.

The contest consisted of two parts: a free demonstration and the actual competition.

The robot had to follow a target that consisted of a black CDROM box placed in the floor and attached to G-Force.

G-Force [Freitas 2005] is a robot, made up of industrial components, which is able to perform preprogrammed trajectories repeatedly. By letting the movement of the target be controlled by the same preprogrammed robot (G-Force), we guarantee the same conditions to all the competitors.

G-Force was preprogrammed to start with very smooth accelerations and to increase the speed while turning left and right in a more or less unforeseeable way. The robots score was proportional to the distance they were able to pursue the target carried by G-Force.

Fig. 11 shows the robot “Rasteirinho” autonomously pursuing the target pulled by the G-Force robot.

Fig. 11. The robot autonomously pursuing a target.

In the second edition the challenge was higher. There were no more targets to follow. The robot had just to make a trajectory based only on arrows placed in the floor indicating the direction to follow. The challenge was such that the robot would need to detect the pattern arrows instead following just the centre of mass of the image. Students were able to use more straightforward tools than the hard C#. Students were divided by the use of Matlab or Labview. The image toolbox of Labview (IMAQ Vision) performed very nicely detecting the arrow patterns with normal webcams, as shown in fig 12.

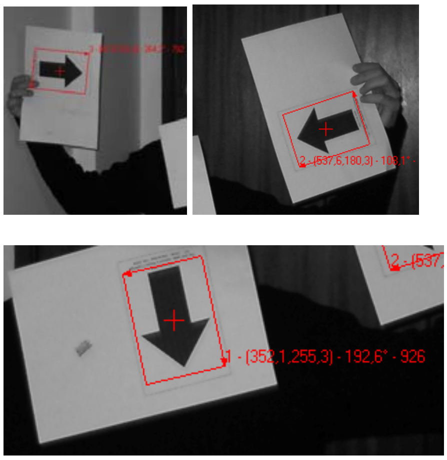

Fig. 12. Testing pattern recognition using Labview Imaq Vision.

In this contest, the winning robot would not be the fastest but the one able to accomplish the mission of going from one lab to another one using the minimum number of arrows on the floor. In this contest among Matlab, Labview and C# many solutions were used. Figure 13 shows one robot finding its way through the arrows.

Fig. 13. The robot guided by the arrows

The students' reaction is mainly positive. What they actually learn depends on the course they are following and the different experiences of the team. Freshmen learn that their lines of code can make much more than reading keyboards and sending graphics to the screen; they actually can make a robot move. Students from control courses learn how to tune their controllers to keep the robot following the target steadily and smoothly.

In any case, the participation on these contests provides a very practical experience on the programming languages they use like C#, Matlab/Simulink or Labview.

VI. CONCLUSIONS AND FURTHER WORK

The low cost robots proved to be effective. Despite their low cost and some limitations they were able to perform tasks usually made by much higher cost robots.

Many newer features become possible every day. Starting with a simple C# solution, it is possible now to program them using more expeditious tools like Labview or Matlab.

These platforms seem to be a good complement for university laboratories. Letting the student bring home the robots will increase the time they stay in contact with the experiments, while giving a totally unexpected use to their personal laptops.

Several possibilities are open regarding the controller. Replacing the laptop by a microcontroller or a PDA are possibilities under study. Modern cellular phones already include cameras, serial ports and may be programmed in JAVA. Sooner or latter we'll see a robot controlled by one of these cellular phones.

Engineering actually extends the frontiers of imagination.

VII. AKNOWLEDGMENTS

The support of the IST Education Improvement program was very precious for this project, as well as the collaboration of many students, colleagues, technicians and friends. To all them, we express our most gratefulness.

VIII. REFERENCES

[Almeida 2000] Almeida, L., Azevedo, J., Cardeira, C., Costa, P., Fonseca, P., Lima P., Ribeiro, F., Santos, V., "Mobile Robot Competitions: Fostering Advances in Research, Development and Education in Robotics", CONTROLO'2000, the 4th Portuguese Conference on Automatic Control, Guimarães – Portugal, October 4-6, (2000), pp. 592-597

[Bruder 2003] Stephen Bruder and Kevin Wedeward, Robotics in the Classroom, IEEE Robotics & Automation Magazine, September (2003), pp. 25-29.

[Camara 2000] Câmara, B., Lopes, J., Marques, C., Lima, P., Cardeira, C., "IQ99 – A Mobile Autonomous Vehicle, in Proceedings of CONTROLO'2000, the 4th Portuguese Conference on Automatic Control, Guimarães – Portugal, October 4-6, 2000, SSMRCP5.

[Delcomyn 2000] Delcomyn, F. and Nelson, M.E. Architectures for a biomimetic hexapod robot, Elsevier, Robotics and Autonomous Systems 30 (2000), pp. 5–15.

[Foxall 2002] James Foxall and Wendy Haro-Chun, Sams Teach Yourself C#™ in 24 Hours, Sams Publishing, March (2002)

[Freitas 2005] Hugo Freitas, Pedro Vilela, Mário Ramalho, Carlos Cardeira, Rui Loureiro, Jorge Bengala, An Industrial Autonomous Guided Robot, accepted to 16th IFAC World Congress, Prague, Czech Republic, July, (2005).

[Fukuoka 2003] Yasuhiro Fukuoka, Hiroshi Kimura and Avis H. Cohen, "Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain based on Biological Concepts", Int. Journal of Robotics Research, Vol.22, No.3-4, (2003), pp.187-202.

[Geppert 2004] Linda Geppert, Qrio, The Robot that could - a technical marvel, Sony's biped humanoid promises to revolutionize home entertainment. IEEE Spectrum, Volume: 41, Issue: 5, May (2004), pp 34- 37.

[Honda 2003] Honda, The Honda Humanoid Robots, www.honda-robots.com, 2003.

[Kaiser 2001] J. Kaiser, P. Schaeffer: " ICU - A smart optical sensor for direct robot control",IEEE International Conference on Mechatronics and Machine Vision in Practice 2001, Hong Kong, China, August 27-29, 2001

[Kaplan 2001] F. Kaplan, et al., Taming robots with clicker training: A solution for teaching complex behaviors, in: Proceedings of the European Workshop on Learning Robots, (2001).

[Kimura 2001] Hiroshi Kimura, Yasuhiro Fukuoka and Ken Konaga, "Adaptive Dynamic Walking of a Quadruped Robot Using a Neural System Model", Advanced Robotics, Vol.15, No.8, (2001), pp.859-876.

[Kimura 2003] The Quadruped Robot "Tekken-II" Challenge to the autonomous walking and running in outdoor environment, http://www.kimura.is.uec.ac.jp/research/Quadruped/photo-movie-tekken2-e.html

[Lima 1998] Lima, P.; Cardeira, C.; "The MiniRobot Project: Learning from Building Small Mobile Robots" IST Science & Tecnology, N. 3, December (1998), pp. 16-22.

[MC 2004] Measurement Computing Corporation, "PMD-1208LS Specifications", http://www.measurementcomputing.com/pdfs/pmd-1208ls.pdf, (2004).

[Murthy 1983] Murthy, S. S., Raibert, M. H., 3D balance in legged locomotion: modeling and simulation for the one-legged case. In Inter-Disciplinary Workshop on Motion: Representation and Perception, ACM, (1983)

[Pierce 2003] C# Webcam Capture, http://www.planet-source-code.com, (2003)

[Ritter 1996] G. Ritter and J. Wilson, Handbook of Computer Vision Algorithms in Image Algebra, CRC Press, (1996).

[Sakagami et al. 2002] Sakagami, Y. Watanabe, R. Aoyama, C. Matsunaga, S. Higaki, N. Fujimura, K., The intelligent ASIMO: system overview and integration, IEEE/RSJ International Conference on Intelligent Robots and Systems, EPFL, Lausanne, Switzerland, vol.3, October (2002), pp. 2478- 2483.

[Schilling 2002] Schilling, K; Roth, H; Rösch, O, "Mobile Mini-Robots for Engineering Education", Global J. of Engng. Educ., Vol 6, No. 1, pp 79-84.

[Sony 2004] Sony AIBO, ERS7M-2, http://www.sony.net/Products/aibo, (2004).

[West 1995] Mark West and Haruhiko Asada, Design and Control of Wheel Omnidirectional Vehicles, IEEE International Conference On Robotics and Automation, Nagoya, Japan, (1995), pp. 1931-1938.

[Zeglin 1991] Zeglin, G. J. Uniroo: A One-Legged Dynamic Hopping Robot. BS Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1991.