# Design of a multi-purpose low-cost mobile robot for research and education

Sol Pedre[1] and Matías Nitsche[2] and Facundo Pessacg[2] and Javier Caccavelli[2]
and Pablo De Cristóforis[2]

[1] Centro Atómico Bariloche, Comisión Nacional de Energa Atómica, Argentina.
[2] Departamento de Computación, Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires, Argentina.
`sol.pedre@cab.cnea.gov.ar`, `mnitsche@dc.uba.ar`, `fpessagc@dc.uba.ar`,
`jcaccav@dc.uba.ar`, `pdecris@dc.uba.ar`

**Abstract.** Mobile robots are commonly used for research and education. Although there are several commercial mobile robots available for these tasks, these robots are costly and do not always meet the characteristics needed for certain applications, being very difficult to adapt because they have proprietary software and hardware. In this paper, we present the design ideas, development and applications of a mobile robot called ExaBot. Our main goal was to obtain a single multi-purpose low-cost robot -more than ten times cheaper than commercially available platforms- that can be used not only for research, but also for outreach and education activities. Its body, sensors, actuators, processing units and control board are well detailed. The software and printed circuit board developed for this project are open source to allow the robotics community to use and upgrade the current version. Finally, different configurations of the ExaBot are presented, showing several applications that fulfill the requirements this robotic platform was designed for.

## 1 Introduction

Mobile robots can be found in many fields, ranging from missions in hostile environments for human beings (such as in space exploration), to home service robots (such as autonomous vacuum cleaners). To develop new applications, it is necessary to have test platforms. Thus, mobile robots are commonly used at research laboratories as well as universities. Nowadays, there are many commercial mobile robots available for this purpose.

The most popular commercial research robots are those of Adept MobileRobots and K-Team companies, in particular the Adept's Pioneer [1] robots and K-Team Kheperas [2]. However, many times commercial robots do not quite fit the necessary characteristics and are difficult to adapt since they have proprietary software and hardware. Moreover, a big drawback is their cost: the basic Pioneer 3-DX academic price is around $4,500, the basic Khepera III academic price is around $3,000 and the basic Koala II (also from K-Team) around $9,000. Extra sensors, processing elements or part replacements are also quite expensive.

An attempt to provide a low-cost platform is TurtleBot 2 [3]. TurtleBot 2 is a mobile robot with open-source software based on ROS (Robot Operation System). It is designed for 3D vision applications as its main sensor is a Microsoft Kinect. Although its creators claim that it is inexpensive, the market price for this robot is around $2,000.

Due to high prices and sometimes robots available in the market do not meet your needs These problems led some universities to develop their own robotic platforms, to lower costs and/or tackle particular tasks for which commercial robots are not well suited. Even universities with extensive experience in robotics have started to propose cheaper robots for research and/or education. Some works present designs of mini robots similar to Kheperas. Such is the case of Rice University's r-one platform [4] and Harvard's miniature robot Kilobot [5].

Educational robots are also a growing field. The most commonly used commercial robots for education are the Lego kits [6]. Although these kits are widely used in K-12 education, they are not suitable for research or undergraduate and graduate education. Some universities have also developed robots for K-12 education. Such is the case of Miniskybot, a 3D printable mobile robot proposed by the Autonomous University of Madrid [7].

In this paper we present our approach to develop a mobile robotic platform, more than ten times cheaper than similar commercial research robots like those of Adept MobileRobots and K-Team companies. Instead of lowering costs by tailoring the design to a particular task -like other academical designs-, we built a highly reconfigurable robot. Our goal was to design a single robotic platform that could be tuned for different research experiments, outreach activities and undergraduate education. For these reasons, we decided to build a small size robotic platform, with reconfigurable sensing capabilities and reconfigurable processing power. In this paper, we present the design ideas, development and applications of the ExaBot, a new multi-purpose low-cost mobile robot.

The rest of the paper is organized as follows: section 2 presents the main design goals and ideas, section 3 presents the resulting system design; section 4 presents results, showing the different configurations used for a wide set of applications; and section 5 outlines conclusions and future work.

## 2 General Design Considerations

The goal of the ExaBot is to have one single robotic platform that allows to carry out research, education and outreach activities, focusing on the low cost compared to its commercial counterparts and with similar functionalities. Therefore, we address a three way general design that trades off between size, cost and functionality.

**Size:** The body of the ExaBot should be small enough to be transported around easily, but also big enough to support many sensors and different processing units. On the other hand, the dimensions of the chassis should be large enough to allow a single-board computer inside. Also, the robot should be able to carry a laptop or a mini external PC on top of it. The relation of robot cost to robot size indicates that off-the-shelf components are the best option. If the platform is too small it gets expensive due to the advanced technologies and fabrication techniques required (e.g. micro-electromechanical systems, micro assembly, etc.), and to the lack of off-the-shelf components. On the other hand, if the platform is too big, the cost of the chassis increases considerably and it also becomes more difficult to use because it requires a large workspace. Therefore, we decided for a medium size for the ExaBot (for more details see Section 3.1).

**Cost:** The cost is one of the main constraints of the robot design. It should be in the order of ten times less compared to its commercial counterparts. Thus, the chassis, sensors and actuators of the robot should be inexpensive, but allow a wide application spectrum. Using a pre-built body, off-the-shelf components and developing the electronics of ExaBot on our own, it is possible to achieve this goal. Because the robot should be small we decided to use small, cheap and still readily available sensors. On the other hand, as the ExaBot has the ability to carry a laptop mounted on it, we can use it as the main processing unit, decreasing significantly the cost of the robot.

**Functionality:** When considering functionality one must consider the domain that the robot is expected to work in, and what it is expected to do in that domain. As we want a multipurpose robot, that can be used for a variety of activities, the ExaBot should be designed to support many different sensors and processing units, and to be easily reconfigured with a particular subset of them for a given task. Regarding locomotion, the robot should be able to operate in both indoor and outdoor environments (see details in section 3.1). Regarding sensing, the ExaBot should have a variety of removable sensors for different applications (see section 3.2). Finally, the robot should have different processing capabilities depending on the task (see section 3.3).

### 2.1 Design flow

From the point of view of desing, a mobile robot can be thought as an embedded system that deals with real world interactions and control. Thus, it is necesary a design methodology that support the cooperative and concurrent development of hardware and software (co-specification, co-development, and co-verification) in order to achieve shared functionality and performance goals for the system. To develop the ExaBot, we adapted a traditional hardware-software co-design flow to the particular field of mobile robot design. Although it would be interesting to go over each design stage, this exceeds the length and aim of this work. In this paper we will only outline the first stage, i.e. goal definition and body, locomotion and sensors definition, and show an overview of the final system with the particular angle of reconfigurability. In Fig. 1 the main stages of the flow can be seen together with the general stages of traditional co-design flows based in processors and ICs.

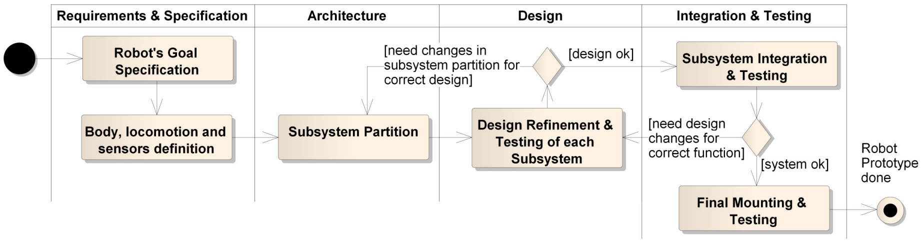


**Fig. 1.** Hardware-software co-design flow for the ExaBot. The horizontal swimlines show the equivalent stages from traditional general co-design flow.

## 3 Robot subsystems

### 3.1 Base Chasis and locomotion

In order to fulfil the goals presented in previous sections a pre-built mechanical kit, the Traxster Kit [8] was selected as a base for the development of the ExaBot. This kit consists on a light, strong and small sized chassis, so it is transportable and can accommodate multiple sensors and processing units. As locomotion system it has two caterpillars, each connected to direct current motors with built-in quadrature encoders (see section 3.2). This allows in-place turning with simple motor commands, in contrast to other models (e.g. Ackerman steering), providing good stability and traction during motion, and helping to reduce odometry errors. Caterpillars were preferred instead of wheel because they allow to overcome small obstacles. This is especially important for outdoor rough terrains.

### 3.2 Sensors

Since the ExaBot should adapt to different applications, we can distinguish a base set of sensors intended for outreach activities and also a set of high-level sensors intended for research activities. There are also some sensors included for proper robot control. The base low-level sensors included are:

**Proprioceptive Sensors:** Wheel quadrature encoders (built-in in the Traxster kit motor) are used to sense the movement of each motor. With this information, a PID controller is implemented which allows velocity-based command to be processed by the robot. Furthermore, odometric readings can be used as a base for localization information (such as position and angle), to be fused to other sensors. To further monitor the motors, a current consumption sensor for the motor driving circuits are included. Finally, to control the battery charge level another sensor is used that measures the voltage values.

**Range-finders:** While lasers are the most precise and reliable, their cost exceed by several times the intended final cost of the robot and therefore were not considered for the educational configuration. Thus, the rangefinders currently used on the ExaBot are infrared (IR) rangefinders an sonar sensors. These are cheap and can return a single distance reading, which simplifies programming reactive behaviors. By mixing IR and sonar range-finders, both short-range punctual and mid-range wide sensors can be included in the same robot, allowing for different use cases. In particular, the ExaBot was given a ring of 8 Sharp GP2D120 IR (4-30 cm) range finders and a Devanatech SRF05 sonar (1-400 cm).

**Line-following:** To enable line-following behaviors, two infra-red light detectors are included in the bottom front of the ExaBot.

**Bumpers:** Contact switches are placed at the front in order to easily detection collisions and program simple evasive reactive maneuvers.

**Other sensors:** There are several ways to add new sensors into the ExaBot. For one, the pins of the microcontrollers were mapped efficiently to maximize useful free ports. Also, sensors that implement SPI communication can be added to the communication bus (see subsection 3.4). An example of this capability was the recent inclusion of an IMU package (3-axis gyroscope L3G4200D, 3-axis accelerometer ADXL345 and 3-axis magnetometer MC5883L). Other types of low-level sensors can be used to explore further education-related tasks.

Of course, all the sensors can be removed or moved around the robot. Moreover, for research activities, the robot can carry sensors commonly used for autonomous navigation methods, such as laser range-finders and cameras. The final configuration depends on the task at hand, as can be seen in section 4. In Fig. 5 several ExaBot configurations are presented.

### 3.3 Computational power

The processing power can be divided in two levels: a low level control board (for sensor and motor control), and a high level processing unit for more complex algorithms. For the low level processing units, we choose to use Microchip PICs, in particular the 18F family. These are cheap, widely used microcontrollers that provided all the necessary modules to control the selected sensors and actuators. The high level processing unit can be any small or medium-size embedded computer. So far we have used the ExaBot with an embedded TS7250 PC104, an embedded Mini-ITX board (AT5ION-T Deluxe), an Arduino board, a RaspberryPI computer, an Android smartphone and common laptops (see Section 4).

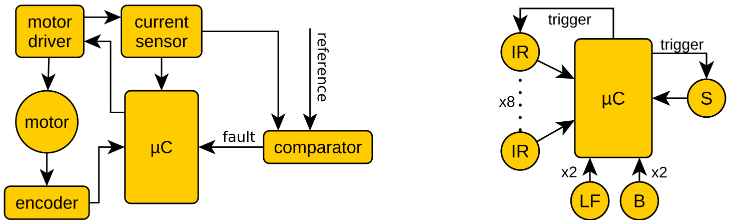
### 3.4 Control board

In this section we shortly describe each subsystem of the control board, with particular interest in the reconfigurable schemes, and its interface to a high-level computer mounted on the robot.

### *Motor Control Subsystem*

The motor control subsystem (Fig. 2(a)) is in charge of the DC motor speed control. The microcontroller ($\mu$C) outputs a PWM pulse (Pulse With Modulation) to the motor driver (H-Bridge) in order to maintain a desired speed, while reading the motor encoder values. The control loop consists of a Proportional Integrative Derivative (PID) controller. Appropriate PID constants were set using the Ziegler-Nichols method [9]. The duty and direction are set by the PID controller and are updated every period according to external commands received in higher levels. Experimental results show that the desired speed with this control is always achieved in less than 100 control loops, that is, less than 81 ms. As an extra safety measure, if the motor's current consumption exceeds a reference value, a fault circuit signal is enabled, which overrides all PWM output and hence stops the motors.

The chosen $\mu$C is a 24-pin PIC18F2431, since it is tailored for DC motor control being the only PIC from the family that has several PWM modules and a Quadrature Encoder Interface module for encoder sensing. For sensing current the ACS712 IC was used, together with an LM319 voltage comparator for fault signal generation from reference voltage. The selected H-bridge driver is the L298.



(a) Motor subsystem diagram

(b) Sensor subsystem diagram

**Fig. 2.** Motors and sensors subsystem diagrams

### *Sensor Control Subsystem*

The sensor control subsystem (Fig. 2(b)) controls the installed low-level sensors: eight infrared range sensors, one sonar, two line-following and two bumpers in the default configuration. For this subsystem, taking into account the need for many analog pins, and others, a PIC18F4680 was selected.

Since each sensor controlled by this $\mu$C requires a different scheme to control it, the chosen software architecture solves this issue. In the case of IR sensors, since these produce an analog voltage (as a function of the sensed distance) which needs to be digitized and the $\mu$C has only one ADC module, a round-robin algorithm was implemented to read each sensor in turn. In the case of the sonar, distances are measured as a PWM signal (i.e., the length of the output signal driven high is proportional to the measured distance). To measure these pulses one of the CCP modules of the $\mu$Cwas used. Finally, since bumpers and line-following sensors produce binary digital outputs, their output values are simply polled using a timer. In order to meet the reconfigurability requirement, each sensor can be turned on or off. Since only sensors that are on are checked, this strategy also saves battery.

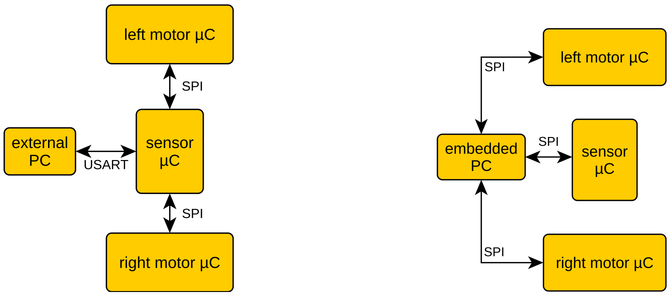
### *Communication Subsystem*

To be able to reconfigure the main processing element, the ExaBot is capable of supporting different types of embedded or external computers. For a completely embedded solution, the ExaBot can be controlled by a specially designed connector, based on the SPI bus (Slave Peripheral Interface). This connector was designed to be used with an embedded PC104 computer, but any other embedded PC that supports SPI can be adapted. This configuration was used for many outreach applications, using the low-level sensors previously outlined. As another option, the ExaBot can be controlled by serial interface (USART) by any type of external computer.

The communication subsystem is specially designed for these two communication options to be switched easily (Fig. 3). Hence, in both cases, the high-level application protocol is the same and only the low-level physical layers change. Moreover, the control board includes all the extra electronics for this change to be easily done (SPI signals multiplexing, and USART ICs). When the ExaBot is used with an external computer connected through the serial port (configuration A), the sensor $\mu$C is the master of the SPI communication. When an embedded computer such as the PC104 is used (configuration B), this computer is the master of the SPI bus and the three microcontrollers are the slaves.

To enable transparent control regardless of the current configuration, the `libexabot` library was developed, which is capable of transparently handling the appropriate transport mechanism. The library presents the same interface either when running on the embedded PC through SPI or when running on an external computer and communicating via USART protocol. Furthermore, a similar version of the library is designed for handling remote control of the ExaBot via UDP (User Datagram Protocol) when the embedded computer is installed. In case an embedded computer is not desired, remote control can still be achieved by using a Bluetooth-to-serial dongle, for example to control the robot by a smartphone or a similar device. Since these libraries present a very simple `C` based interface, it can be used from any `C/C++` application and even enables the development of high-level bindings for languages such as Python, Ruby, etc.

Finally, using `libexabot` library a ROS (Robot Operating System) node was also developed which allows to control the ExaBot and to recieve odometry and the others sensor data.



(a) Configuration with external computer

(b) Configuration with embedded computer

**Fig. 3.** Communication subsystem diagram

### 3.5 Final Product

Following the previously outlined design guides and requirements, a final Printed Circuit Board (PCB). Each subsystem was initially tested individually using prototyping boards along with specialized test firmware. A final dual-layer PCB was obtained, based on Surface Mount Design (SMD) components. In order to facilitate re-programming of the microcontrollers on the final PCB, the ICP (In-Circuit Programming) functionality was exported using RJ11 connectors in the board itself. Figure 4 shows the final control board of the ExaBot.

The control board can be powered either using external power (during testing) using a standard PC power supply, or using batteries. Since all subsystems were also analyzed in terms of power requirements and in order to isolate the motor and control logic power lines, two battery lines are used per ExaBot. The ExaBot was so far tested with LiIon and LiPo batteries. Without an embedded PC, the robot can be powered for several hours using batteries of moderate capacity (around 2500mAh for control logic).

In order to support the mounting of all sensors, the control board and batteries, the chassis was drilled and adapted as necessary. This also includes the developing of many types of metal mounts which are capable of carrying a laser, low-level sensors and even a laptop on top of the robot.

The cost of the final ExaBot depends on which configuration is used, since it depends on which processing unit and sensors are installed. The cost of the main low-level control board including all the electronics and PCB printing, a base set of sensors, the chassis and batteries is approximately $250. However, the chassis (that includes the motors and encoders) amounts for more than half that cost.

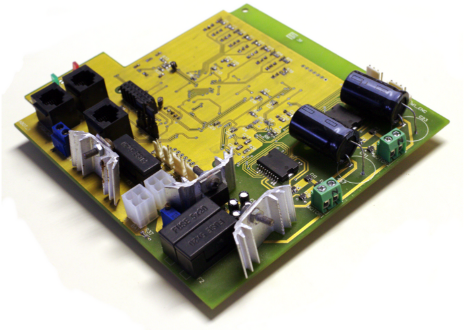


**Fig. 4.** Final ExaBot control board.

## 4 Results and Applications

A short comparison of the ExaBot with similar robots available in the market can be found in Table 1. As already stated, the ExaBot is suited for research and education activities, is reconfigurable in both it's sensing and processing capabilities, and is much cheaper than similar commercial robots.

| Robot | Functionality | Size | Cost ($) | Locomotion based on | Reconfig. sensors | Embedded PC | Max speed ($\frac{cm}{s}$) |
|---|---|---|---|---|---|---|---|
| Pioneer P3AT | Research | Large | 6,000 | wheels | yes | optional | 70 |
| Khepera III | R & E | Small | 3,000 | wheels | no | no | 50 |
| TurtleBot 2 | R & E | Medium | 2,000 | wheels | no | yes | 65 |
| Lego EV3 | Education | Medium | 349 | varies | yes | no | varies |
| Kilobot | Research | Small | 100 | vibration | no | no | NI |
| R-one | R & E | Small | NI | wheels | no | no | 25 |
| **ExaBot** | **R & E** | **Medium** | **250** | **caterpillars** | **yes** | **yes** | **50** |

**Table 1.** Comparison between ExaBot other similar mobile robots. R & E: Research and Education, NI: Not informed.

Different configurations of the ExaBot were used in several applications fulfilling all the goals the ExaBot was designed for. In the next subsections we briefly comment some of these.

### 4.1 Research

The main research activities with the ExaBot are related to autonomous visual navigation. In this context, the ExaBot was used as a platform for experiments in different works.

One work presents a real-time image-based monocular road following method [10]. To achieve real-time computation necessary for on-board execution in mobile robots, the image processing is implemented on a low-power embedded GPU. Hence, the ExaBot was configured to use a Mini-ITX board (AT5ION-T Deluxe)- that includes a 16 core NVIDIA GPU- as the main processing unit, and a FireWire camera (model 21F04, from *Imaging Source*) as the only exteroperceptive sensor (see Fig. 5, image **d**). The Mini-ITX board connects through RS232 to the control board as explained in section 3.4. The same configuration was used for experiments in a completed PhD Thesis [11], that proposes a hybrid method for navigation combining the aforementioned method to follow paths, with a landmark-based navigation method to traverse open areas.

Another work presents the use of disparity and elevation maps for obstacle avoidance using stereo vision[12]. In this work, a common notebook or netbook is used as the main processing unit and a low-cost Minoru 3D USB webcam as the only exteroperceptive sensor. The notebook connects through RS232 using a USB-to-serial dongle (see Fig. 5, image **c**).

An embedded method for monocular visual odometry was developed in a master thesis, using an Android-based smart-phone as the main processing unit and its embedded camera as the main sensor. For this work, yet another configuration of the ExaBot was used (see Fig.5, image **b**). Here, the Android cellphone connects either through Wi-Fi to the on-board PC104 or via Bluetooth directly to the control board.

Currently, further autonomous navigation experiments are starting with the aforementioned integration of a SICK TIM310 laser range finder, and a gyro-compensated magnetometer sensor.

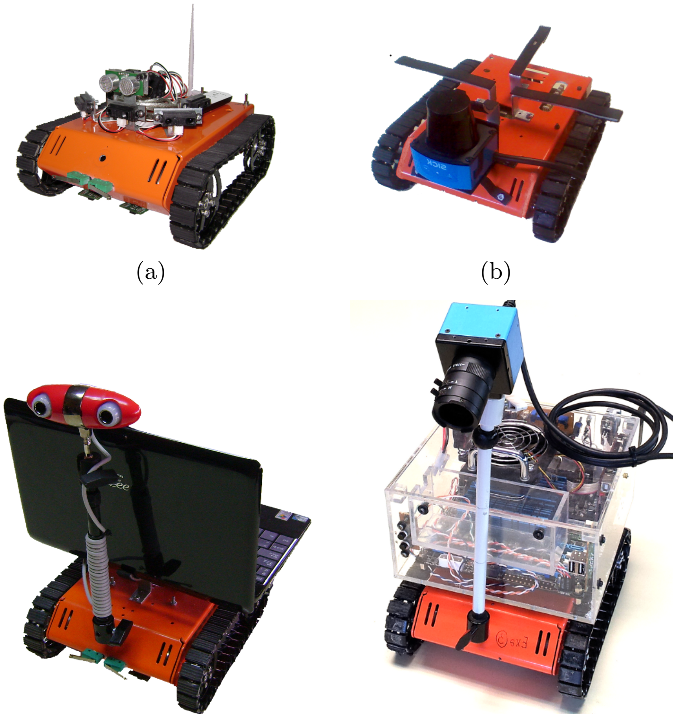
### 4.2 Outreach

The ExaBot is also used in Educational Robotics courses, talks and exhibits. Educational Robotics proposes the use of robots as a teaching resource in K-12 education that allows inexperienced students to approach fields other than specifically robotics. A key problem in this context is to have an adequate easy-to-use interface between inexpert public and robots. For this, we developed a new behavior-based application for programming robots, specially the ExaBot [13]. Robotic-centered courses and other outreach activities were designed and carried on [14]. In the last years, three eight-week courses, five two-days courses, more than ten one-day workshops and talks were taught to different high school students using the developed programming interface and several ExaBots. For this work, the ExaBot was configured with the PC104 as the main processing unit and all the exteroperceptive sensors described in section 3.2: IR telemeters, sonar, bumpers and line-following (see Fig. 5, image **a**).

### 4.3 Undergraduate Education

The ExaBot is also used in undergraduate and graduate courses of the Departamento de Computación, FCEN-UBA. In particular, it is used in the Robotics

Vision course. This course covers topics regarding monocular and stereo vision applied to mobile robots. Several algorithms for autonomous robot navigation are implemented and tested by students using the ExaBot.



(a) (b) (c) (d)

**Fig. 5.** Various configurations of the ExaBot: (a) with all low-level sensors and PC104 (b) with a laser range-finder (c) with a netbook and a 3D Minoru Camera, (d) with an embedded Mini-ITX board and a FireWire Camera.

## 5 Conclusions and Future Work

In this paper, we present the design ideas, development and applications of the new mobile robot ExaBot. Our main goal was to obtain a multi-purpose low-cost robot- i.e., ten times cheaper than commercially available research robots- that could be used not only for research, but also for outreach and education.

The main requirement to achieve a low cost robot that can be used for such diverse fields is that the robot is highly reconfigurable. Hence, the ExaBot was designed with many sensors that can be optionally installed, and built-in sensor expansion ports. The high level processing unit and the communication protocol are also reconfigurable. In this manner, many different configurations of the ExaBot have been built and used; keeping the base cost of the robot at around $250. We have successfully used them for research activities, mainly in vision-based autonomous navigation; for undergraduate education; and for robotic-centered courses at K-12 education and other outreach activities.

As future works, we are planning to build the mechanical chassis ourselves to further lower costs. Moreover, further research experiments are planned using the recently incorporated sensors (laser scan and gyro-compensated digital compass), as well as new processing elements such as a BeagleBoard.

## References

1. Adept Mobile Robotics: Pioneer 2-DX and 3-DX (2013)
2. K-team corporation: Khepera, Khepera II and Khepera III (2013)
3. Garage, Willow: Turtlebot (2011)
4. McLurkin, J., Rykowski, J., John, M., Kaseman, Q., Lynch, A.: Using multi-robot systems for engineering education: Teaching and outreach with large numbers of an advanced, low-cost robot. IEEE Transactions on Education **56**(1) (2013) 24–33
5. Rubenstein, M., Ahler, C., Nagpal, R.: Kilobot: A low cost scalable robot system for collective behaviors. In: IEEE International Conference on Robotics and Automation (ICRA). (2012) 3293–3298
6. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: A systematic review. Computers & Education **58**(3) (April 2012) 978–988
7. Gonzalez-Gomez, J., Valero-Gomez, A., Prieto-Moreno, A., Abderrahim, M.: A new open source 3d-printable mobile robotic platform for education. In: Advances in Autonomous Mini Robots. Springer Berlin Heidelberg (2012) 49–62
8. RoboticsConnections: Traxster Kit (August 2012)
9. Bräunl, T.: Embedded Robotics - Mobile Robot Design and Applications with Embedded Systems. Second edn. Springer-Verlag (2006)
10. De Cristóforis, P., Nitsche, M., Krajník, T., Mejail, M.: Real-time monocular image-based path detection. Journal of Real-Time Image Processing (2013) 1–14
11. De Cristóforis, P.: Vision-based mobile robot system for monocular navigation in indoor/outdoor environments. PhD thesis, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires (2013)
12. Piré, T., de Cristóforis, P., Nitsche, M., Berlles, J.J.: Stereo vision obstacle avoidance using disparity and elevation maps. In: IEEE RAS Summer School on Robot Vision and Applications. (2012)
13. Caccavelli, J., Pedre, S., de Cristóforis, P., Katz, A., Bendersky, D.: A New Programming Interface for Educational Robotics. In: Research and Education in Robotics - EUROBOT 2011. Volume 161 of Communications in Computer and Information Science. Springer Berlin Heidelberg (2011) 68–77
14. De Cristóforis, P., Pedre, S., Nitsche, M., Fischer, T., Pessacg, F., Di Pietro, C.: A Behavior-Based Approach for Educational Robotics Activities. IEEE Transactions on Education **56**(1) (2013) 61–66