

TEACHING AI CONCEPTS USING A ROBOT AS AN ASSISTANT

Eleftherios Karapetsas

*Department of Informatics
Alexander Technological Educational Institute
of Thessaloniki
elefkar@it.teithe.gr*

Demosthenes Stamatis

*Department of Informatics
Alexander Technological Educational Institute
of Thessaloniki
demos@it.teithe.gr*

ABSTRACT

In this paper we present our approach in incorporating a robot platform as a teaching tool in an undergraduate course of Artificial Intelligence. The platform is used in the context of a series of Lab exercises/assignments, helping students understand and develop AI algorithms. Students are able to construct models of real world situations – load their algorithms to the robot and test their behaviour. A re-configurable maze (with walls, corridors, dead ends and target nodes) is used as a closed world where the robot moves and executes the specific AI algorithm. The robot acts as a kind of teaching assistant for the students, being able to visualise the step by step execution of their algorithms. It also possesses some kind of intelligence in the sense that, Knowing the Lab context (range of algorithms, types of possible maze configurations and other special conditions) is able to face problems arising from real time side effects of the robot function, hide them from the students, who are able to concentrate on the AI algorithms themselves.

Keywords

Artificial Intelligence, Robotics, Search Algorithms, Graph problems, Maze

1. INTRODUCTION

Robotics have been incorporated, as tools for teaching Artificial Intelligence (AI) concepts, in the curricula of Computer Science and Computer Engineering programmes for a number of years [1,2,3]. Recently low cost robots made this incorporation more frequent and more effective [4,5,6]. As reported in the literature there is a number of benefits in amalgamating robotics with learning.

As pointed out by Greenwald et al. [2], “*Robotics combines the fantasy of science fiction with practical real-world applications and engages both the imaginative and sensible sides of students*”.

Other benefits in using robots for teaching AI concepts are:

- Active-problem based learning is enhanced.
- Cooperative and Collaborative types of learning are promoted when students work in groups
- A kind of tangible computing is also promoted, where students are able to construct real world models and understand difficult abstract AI concepts.

However that approach is not without its downsides, as there have been some problems noticed. Most crucial among them is that using robot platforms for teaching purposes makes projects for students harder and more time consuming than traditional ones [4]. This is due to the fact that most of the project time is spent in making robots work reliably than using them as vehicles for understanding AI concepts.

In this paper we present our approach in incorporating a robot platform as a teaching tool in an undergraduate course of Artificial Intelligence. In our case the robot is used, in the context of a series of Lab exercises/assignments as a teaching assistant, helping students understand and develop AI algorithms. Students are able to construct models of real world situations – load their algorithms to their robot assistant and test their behaviour. A re-configurable maze (with walls, corridors, dead ends and target nodes) is used as a closed world where the robot moves and executes the specific AI algorithm.

Our approach differs from the ones reported earlier in that robots are used in the context of the AI course to solve problems in a closed lab situation, which means that students have to complete pre-specified Lab exercises in a more or less predefined amount of time. The robot acts as a kind of teaching assistant for the students, being able to visualise the step by step execution of their algorithms. It also possesses some kind of intelligence in the sense that, Knowing the Lab context (range of algorithms,

types of possible maze configurations and other special conditions) is able to face problems arising from real time side effects of the robot function, hide them from the students, who are able to concentrate on the AI algorithms themselves.

In the following section we present a set of AI concepts whose teaching can be enhanced by the use of a robot. The specifics of these concepts were turned into demands that resulted in choosing and extending the robot platform we use. The description of the platform, together with hardware and software extensions needed, is presented in the next two sections (3 and 4). We discuss the learning context in which our platform is used for teaching AI concepts in section 5. We then present some initial results together with future work in the last section of the paper.

2. AI CONCEPTS UNDER CONSIDERATION

Teaching AI concepts to undergraduate students might often lead to frustration on the student's part. Trying to grasp the workings of AI algorithms only in theory and/or computer simulation can be quite hard for the average student. Incorporating robotics as teaching tools is proving a good step forward in alleviating this problem.

AI topics taught using low cost Robot platforms vary from introductory ones, such as blind search algorithms, to advanced ones, such as robotics themselves, machine vision and genetic algorithms. A representative list of such topics can be found in [1]. In our case, topics chosen fit well with the syllabus of an undergraduate course offered in the 7th Semester of the Department of Informatics, TEI of Thessaloniki, Greece. The aim of the course is to introduce problem solving techniques, schemes for Knowledge Representation and application areas of Artificial Intelligence such as Planning, Natural Language Understanding, Neural Networks and Intelligent Agents.

The following list of topics is under consideration in the series of lab exercises where our robot platform is used:

- Blind search algorithms (depth-first, breadth-first)
- Heuristic search algorithms (hill-climbing, best-first, A*)
- Generate and test algorithms (using backtracking techniques)
- Planning

Based on the nature of all of the above algorithms we have chosen a reconfigurable maze to represent their solution/state space. In this respect all lab exercises were designed to comply with the maze.

We have constructed such a maze as a reconfigurable panel. Students are able to reorganise the maze according to their needs by moving toy walls which in turn form corridors and dead ends. Target or goal nodes and objects in the maze are indicated by the use of a predefined set of signs (numbers and arrows). The maze then acts as the environment where the robot has to move, construct a graph representing the solution/state space of the problem under consideration and process it by using the specific algorithm uploaded.

3. THE ROBOT USED

The robot we are using is the SRV-1 from Surveyor Corporation [7]. It is equipped with a powerful Analog Devices Blackfin BF537 processor running at 500MHZ and able to reach about 1000 MIPS. That kind of processor combined with a 1.3 megapixel digital camera allows for real-time visual processing to use in the autonomous operation of the robot. Adding to that, 2 laser pointers are mounted on the right and left side of the camera giving the robot quite an effective way to detect obstacles and with some modifying a semi-accurate means to measure distances. Inside its chassis are four precision DC motors which give it the ability of differential drive, something of utmost importance in our application.

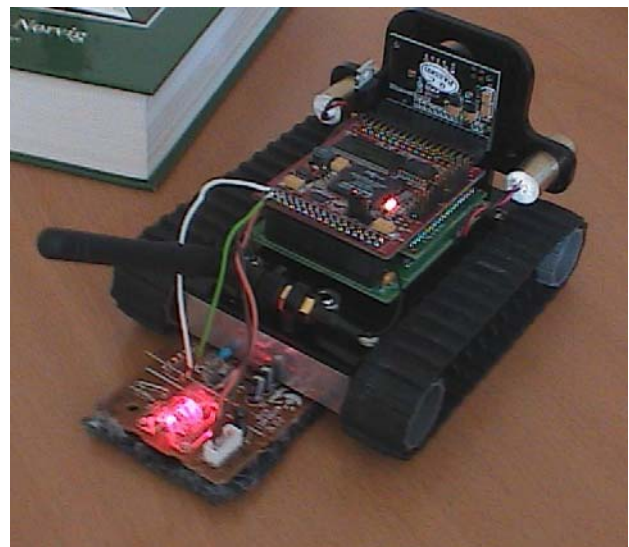


Figure 1: SRV-1 with our added extension

Moreover SRV-1 comes with a number of features that make it pertinent to our application:

- **Powerful processor:** The processing power of the BF537 allows for the implementation of advanced Artificial Intelligence algorithms in real time. Combined with the omnivision camera which sends frames to the processor via DMA (Direct Memory Access) the SRV-1 is a very good choice for an autonomously

operating robot. That is since we needed a processor which could keep some kind of internal state while at the same time analyzing the frames captured by the camera and determine the state of its environment. All of these should be accomplished fast enough that the robot does not miss any sensor readings from the maze's environment.

- **Wi-Fi access:** SRV-1 comes equipped with a Lantronix matchport board enabling Wi-Fi access to the robot from any computer with Wi-Fi capacity. The option to communicate with the robot via a wireless connection gives us the added bonus of being able to teleoperate the robot if the need arises, while at the same time we can extract and process data from the robot's environment at a remote terminal in real time. This gives us the option to manipulate the robot's decisions and see what the robot is thinking while it is executing the AI algorithms, something vital for the student's understanding of the inner workings of AI algorithms.
- **Long battery life:** The robot's power source is a Li-Poly battery which after full charging can power the robot for 4-5 hours, an amount of time which makes autonomous operation possible during laboratory exercises.
- **Combination of good price and adaptability:** Choosing SRV-1 as our platform for AI education was due to needing a relatively inexpensive robot (at the range of € 300), and at the same time completely adaptable to our current or any future application (completely re-programmable firmware and expansion board to add any sensors needed). SRV-1 fits those needs since the user can program its firmware in C using the GNU bfin-elf-gcc compiler and also add any extensions he needs via General Purpose I/O pins conveniently located in a 32 pin header on the rear side of the processor chip. All that coming for a relatively inexpensive price, which makes that robot a very good candidate for equipping an Artificial Intelligence laboratory.

For our application the robot needed to be able to infer its position inside the maze. To accomplish that it needed a sensor for odometry. Unfortunately the robot was not equipped with motor encoders so we had to add some kind of dead reckoning sensor for odometry. After some consideration we settled on an optical sensor much like that of an optical mouse. So we made the required extensions to the robot as can be seen in Figure 1. We mounted the sensor on the rear side of the robot in a position

where it could measure without interferences and connected it to the processor. The optical sensor returns the change of position in deltaX and deltaY values. This provides a lot more accuracy on the readings than any motor encoder ever would. For example our sensor returns 1000 cpi (counts per inch) making it very sensitive to any kind of movement. That makes the optical sensor ideal for the task of mapping our robot's movement in the maze up to a certain degree of accuracy.

4. A SET OF FUNCTIONS ADDED

For this application to be successful we needed the robot to be able to accomplish a set of functions. More specifically we needed:

- A robotic agent able to extract data from the environment accurately enough to infer its position in the maze and decide what to do with said data according to the student's algorithm currently being interpreted by the robot.
- To develop a console which would be ran on a remote computer connected to the robot and allow students to interface with it, see video from the camera, change various settings, setup and upload the neural networks and write and upload the algorithms.

The SRV-1's firmware originally had some visual processing functions like colour segmentation and histogram generation. Moreover it included complete motor controlling functions as well as image capturing and Wi-Fi connection via xmodem protocol. It gave us a robust platform upon which we could build on and extend to match the needs of our application.

In order to complete our application we had to totally modify the firmware and add many additional functions:

- **Camera's frames real-time processing:** We wrote functions to measure the intensity of the left and right laser beams and also implemented a modified version of Canny's edge detection algorithm [9] to detect openings to the right and left of the robot finding the possible routes leading out of each and every node. So we turned the camera's captured frames into two virtual sensors. One sensor having to do with the lasers and the other with edge detection. So the *checkLasers()* function returned the existence or not of an obstacle near enough and also returned the intensity of the two laser beams. As for the *detectWall()* function which incorporates edge detection, it finds openings

in the walls of the maze and registers them as new ways for the robot to explore.

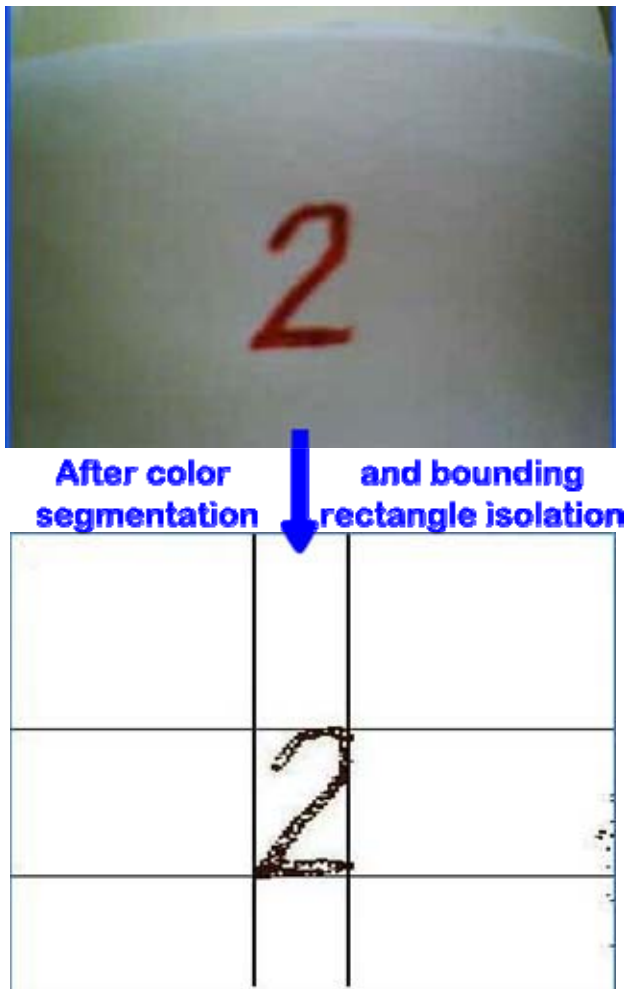


Figure 2: Visual processing

- Sensor data interpretation:** Following the above we had to define functions to interpret the data from the sensor's readings and determine when and how nodes, walls, the goal and all the other elements of the maze's environment were detected. Subsequently these data formed our internal data structures which represented the world inside the robot's "mind".
- Additional visual processing:** We developed functions to isolate the pre-determined signs and number patterns from the frames captured by the camera so we could feed them to the neural networks used, for training and recalling. Colour segmentation was used for that purpose, isolating the number or sign and finding its bounding rectangle as can be seen from Figure 2. The contents of the bounding rectangle are then shrunk to the neural network's input neurons size and are fed to the network. So we added a

segmentColor(color) function to find the sign or number in the image and then *isolateObject(object)* to isolate the object's pattern in the camera's captured frame.

- Keeping an internal state:** Moreover we created a series of functions keeping track of the heading (NORTH, EAST, SOUTH, WEST) of the robot as opposed to using a compass sensor which could be severely affected by the noise generated by the motors. In addition we wrote drivers to enable communication between the optical sensor and the blackfin. Hence with *opticalRead()* we can read data from the optical sensor and maintain the total position change of the robot in the X and Y axis.
- Movement control:** Additional routines to control the robot's movements and make sure it is going straight and is not bumping into walls had to be developed. They were all incorporated into a function, *align()*, which uses data from the lasers, the optical sensor and edge detection to assure that the robot goes as steadily straight as possible, is aligned with the walls and finally makes sure that the rotations used to turn the robot are as close to 90 degrees as is possible.

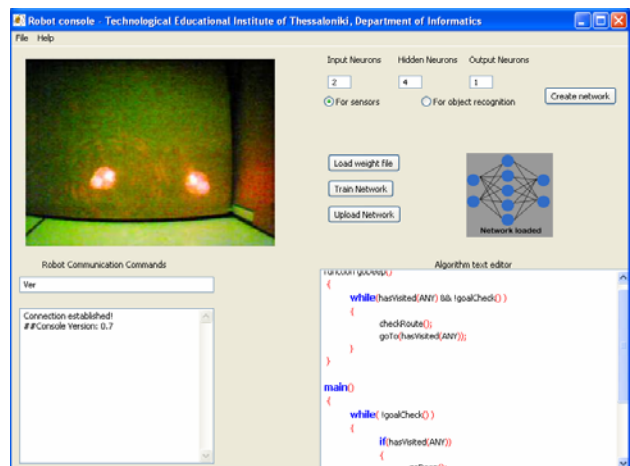


Figure 3: Robot console snapshot

- Algorithm interpreter:** In addition to all of the above a system to interpret and run the student's algorithms had to be developed into the robot's firmware. By doing so, the students are able to write their algorithms in some intermediate language and upload them to the robot which in turn would act as a kind of intelligent debugger. So we developed a C-like pseudo code language, based on the existing C-interpreter of the robot's firmware for the students to write their algorithms in and for the

robot to understand and interpret into normal firmware code.

- **Remote console:** Moreover a console program able to show the current frames captured from the robot's camera had to be developed. In addition it needed to do so only when requested since if it did otherwise it could encumber the processor of the robot and make autonomous operation difficult. That same program had to also be able to interpret pseudo code written by the students and upload it into the robot's firmware. Lastly the console program needed to be able to easily alter various settings of the robot and upload them to the robot. A snapshot of this console can be seen in Figure 3.

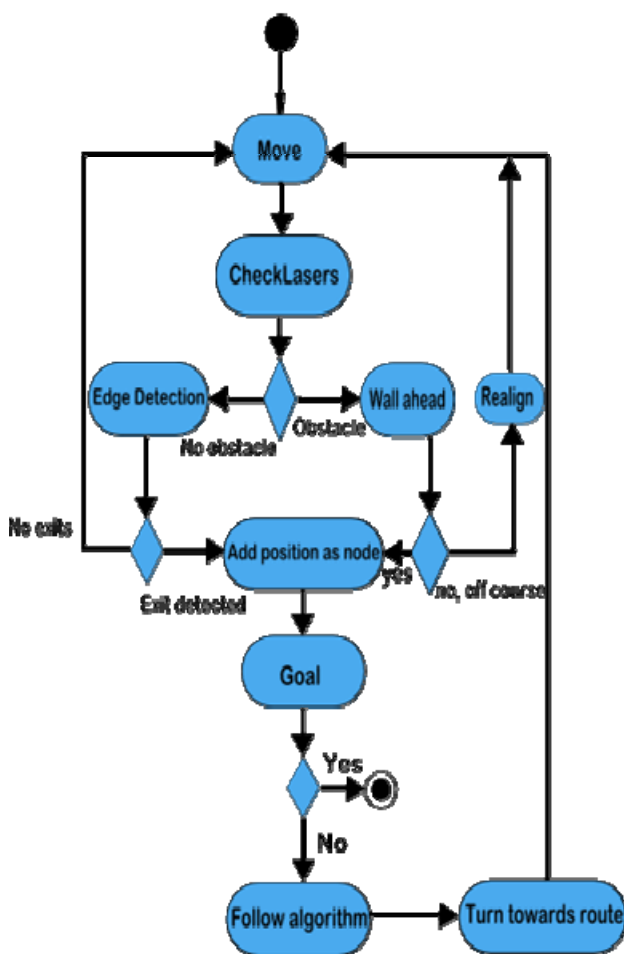


Figure 4: Activity diagram of the robot's main loop

Summing up a system had to be developed to integrate all of the above into a compact firmware solution which would be able to control the robot accurately enough to solve the maze according to what each algorithm required of it. The console along with the robot's firmware forms the complete

platform upon which our learning system for Artificial Intelligence is based.

5. THE ROBOT AS A TEACHING ASSISTANT

In developing our robot platform as a teaching tool the following key issues were used as guidelines:

- **The robot platform should be used in a closed Lab situation:** This means that students have to work with exercises/ assignments in a predefined amount of time (about two hours). Our approach is not about teaching AI Robotics. (Students just have a small introduction to Robotics in the theoretical part of the course). In this respect we do not ask students to construct their own robot not even to program it from scratch, a procedure which is harder, more time consuming and applicable only to larger scale open project assignments. We want students to concentrate on the intelligence aspect of algorithms and use the robot as a testing tool. Students are asked to test AI algorithms. Later on they have to develop their own versions and have the robot execute them. In the first case the robot plays the role of a demonstrator and in the second case it plays the role of a debugger.
- **The robot should act as an Intelligent Assistant to the students:** Furthermore our approach is not about using a robot as an agent sensing and acting in its environment. We rather consider the robot as an intelligent agent who knows the Lab context. For this we created a working and reliable system to control the robot in our pre-determined environment and concealed all of its inner workings from the student. Problems specific to robotics, such as localisation, mapping, odometry and object recognition arising as side effects when the robot executes algorithms in the context of lab exercises are solved by embedding the special functions described in the previous section in the robot's firmware. These functions actually make the robot more intelligent, in the sense that they extend its domain knowledge with Lab specific knowledge. Additionally to be able to have a high level interface between the student and the low-level firmware of the robot we provide a mini pseudo-language interpreter. Students are using this language to write and upload their algorithms to the robot through the systems remote console (fig. 3). Based on the above the robot can act as an intelligent assistant to students since it is able to test their algorithms as a moving agent. That agent

is able to act both as a tutor who shows them how AI algorithms are played out in the real world and as a debugger for their own algorithms.

- **The robot platform should provide to students a form of tangible computing experience:** This is achieved by the use of the reconfigurable maze. Representing the maze as a graph leads to a direct visualization of the very meaning of a graph since the student can see each node as a crossroad of the maze and each edge connecting any two nodes as the maze corridor that connects these nodes. In that way students can see the various graph search algorithms play out in a real world environment and better comprehend the inner workings of each algorithm. Moreover the student is lead to understand the notion of planning, which is required in order for the robot to actually follow the algorithm and traverse the maze. Planning for stochastic environments where things sometimes do not go as planned can be difficult to comprehend without understanding the reason why planning is done. As a result being able to see the robot devise a plan in order to go from one node of the maze to the next taking in account any and all discrepancies which can occur can really help towards that goal. In addition the student is taught backtracking. Backtracking is something many students seem to have difficulties comprehending just in theory. It can be elusive for the student to understand since going back to the node of the graph required by the algorithm is a novel technique for the average student who is used to more traditional type of algorithms. That is why by actually seeing a physical agent in the real world going back and retracing its steps according to what is required from the algorithm, the nature of backtracking can be better understood.

6. FIRST CONCLUSIONS AND FUTURE WORK

We have presented our approach in incorporating a robot platform as a teaching tool in the context of an undergraduate course on Artificial Intelligence.

The robot chosen, SRV-1, proved to be a very good cost/effective solution. It was controllable, adaptable and powerful enough to act as the basis for the system we developed.

Having the aim to construct a robot platform that could act as a teaching assistant to students, facing a set of AI topics/algorithms during Lab sections of the course, we proceeded to adding a number of

extensions to the robot (both hardware and software). An optical sensor was adapted for use with the robot in order to measure distances and provide data for solving odometry problems. The firmware of the robot was also extended with a number of functions to solve problems having to do with sensor data interpretation, visual processing, movement control, and correct positioning. All these functions are related to the type of actions the robot performs during the execution of the algorithms in question.

We have also developed a remote console interface to the robot as well a pseudo code language interpreter which are used by the students to control the robot (through a PC or Laptop), upload their algorithms and be able to analyse real time data during robot operation.

The robot platform was tested, on a pilot phase, with a limited number of students. During this first period extensive tests were conducted with emphasis on testing its reliability when executing AI algorithms in the environment of the reconfigurable maze.

Based on these tests it is worth mentioning the following observations:

One of the first things noticeable after even the first few runs of the maze by the robot was that the hardest part was to make sure that all the rotations, and all the sensor readings had to be calibrated to be precise. As precise as humanly possible since even if one sensor reading goes off by a small margin each time the sensor is read, slowly but steadily it will accumulate, leading the robot off course. In order to avoid this problem we added one more routine to the robot which acted as a self-correction mechanism. It aligned itself with the walls in every turn of the maze making sure it was always parallel to the corridors of the maze and that each rotation was 90 degrees and no more.

Another self-correction mechanism we decided to implement after some trials was that of neural networks. Being able to calibrate it each time we altered the maze significantly or even changed the environment completely was a very big advantage. Not to mention the fact that the student could observe which network structures/weights worked and how a neural network can be trained for applications in a real world environment.

Regarding pedagogy issues, the initial feedback from the students - although limited - was positive. Having our system functioning reliably, we intend to use it, in a more systematic way during the next academic year, in order to be able to test its didactic value.

Students engaged in the pilot phase found the use of the robot platform a very good vehicle for understanding AI concepts. At the same time students seem to get a bit disoriented willing to

discuss improvement of Robot behaviour rather than concentrate in using Robots as assistants for understanding AI algorithms. Having students working in groups to split roles might solve this problem adding the advantage of injecting more robotics in their education, provided we will be able to adapt the course syllabus accordingly.

As pointed out by T. Balch et al. [8] robots are tools with “a low floor and a high ceiling—that is, they’re easy for a novice to learn but have enough power so that an expert will continue to use them”. During the pilot phase, as mentioned earlier we initially had to use neural networks as a facility to extend the robot’s firmware. Later, based on that facility and on student “demands” we came up with adding an extra lab exercise to incorporate neural networks as a topic. Students can be taught about neural networks as pattern classification and data filtering machines in the context of maze solving. That is since neural networks can be used to filter the data coming in from the robot sensors to differentiate between correct readings and false readings. Students do not write code for the neural networks, but they are able to define the structure of the network through the provided console interface, train it and upload it to the robot. By seeing a neural network that they trained and whose structure and weights they defined do the task of sensor data filtering for them they can truly comprehend the power of artificial neural networks.

Another area of future work relates to the improvement of the pseudo code language and its interpreter used for uploading AI algorithms to the robot.

7. REFERENCES

- [1] Zachary Dodds, Lloyd Greenwald, Ayanna Howard, Sheila Tejada, and Jerry Weinberg
Components, Curriculum, and Community:
Robots and Robotics in Undergraduate AI
Education, AI magazine Vol. 27 No. 1, 2006
- [2] Lloyd Greenwalk, Donovan Artz, Yogi Mehta
and Babak Shirmohammadi, *Using
Educational Robotics to implement complete
AI solutions*. AI magazine Vol. 27 No. 1, 2006
- [3] Frederick L. Crabbe, Unifying Undergraduate
Artificial Intelligence Robotics: Layers Of
Abstraction Over Two Channels, AI magazine
Vol. 27 No. 1, 2006
- [4] Amruth N. Kumar, *Three years of using robots
in an artificial intelligence course: lessons
learned*. ACM Journal on Educational
Resources in Computing, Vol.4, No. 3, 2004
- [5] Eduardo Ferme and Elsa Fernandes,
*Crossing the river with robots: Changing the
way of working in an AI subject*, 3rd UK
Workshop on AI and Education, 2007
- [6] Eduardo Ferme and Luis Gaspar
*RCX+PROLOG: A Platform to use Lego
Mindstorms™ Robots in Artificial Intelligence
courses*, 3rd UK Workshop on AI and
Education, 2007
- [7] Surveyor SRV-1 Blackfin Robot, Surveyor
Corporation, <http://www.surveyor.com/>
- [8] Tucker Balch et al., *Designing Personal
Robots for Education: Hardware, Software,
and Curriculum*, IEEE Pervasive Computing,
April-June 2008
- [9] J Canny, *A computational approach to Edge
Detection*, IEEE Transactions on Pattern
Analysis and Machine Intelligence Volume 8 ,
Issue 6 (November 1986)