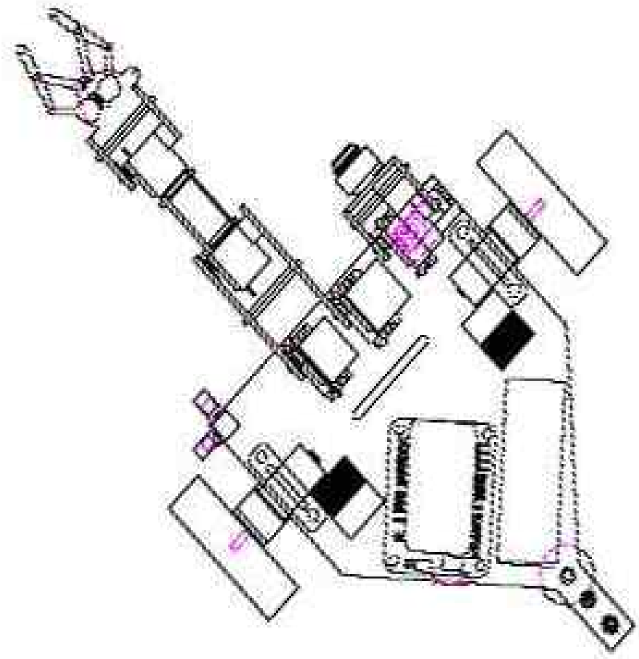
Small Autonomous Mobile Robots for Teaching Introductory Programming to Engineering Students

Z. Carpenter, J. C. Diaz, Ph.D., G. R. Kane, Ph.D., P.E.
EE-U Tulsa, CS-U Tulsa, EE-U Tulsa

Abstract:

Spirit and Opportunity, the mobile robots or rovers on the surface of Mars, motivated the construction of a fleet of rovers with various capabilities and sensors to be used to teach introductory programming. The rovers were designed and built by the students in our EE senior level robotics class using mostly off-the-shelf components. The rovers were designed to be programmed to achieve a series of predefined tasks autonomously such as line following, wall hugging, and pick and place. The rovers were integrated into the introductory computer science class for engineers to involve students with practical applications of physics and math. The curriculum and projects were designed to provide a hands-on technical problem solving experience in a programming class in order to increase the retention of the concepts taught. This paper provides an overview of the rovers and our experiences when they are deployed in the classroom.



Introduction:

At the time that NASA's Spirit and Opportunity rovers walked around the surface of Mars, TU engineering students were programming the *Tulsa* rovers learning the intricacies of autonomous robotics.

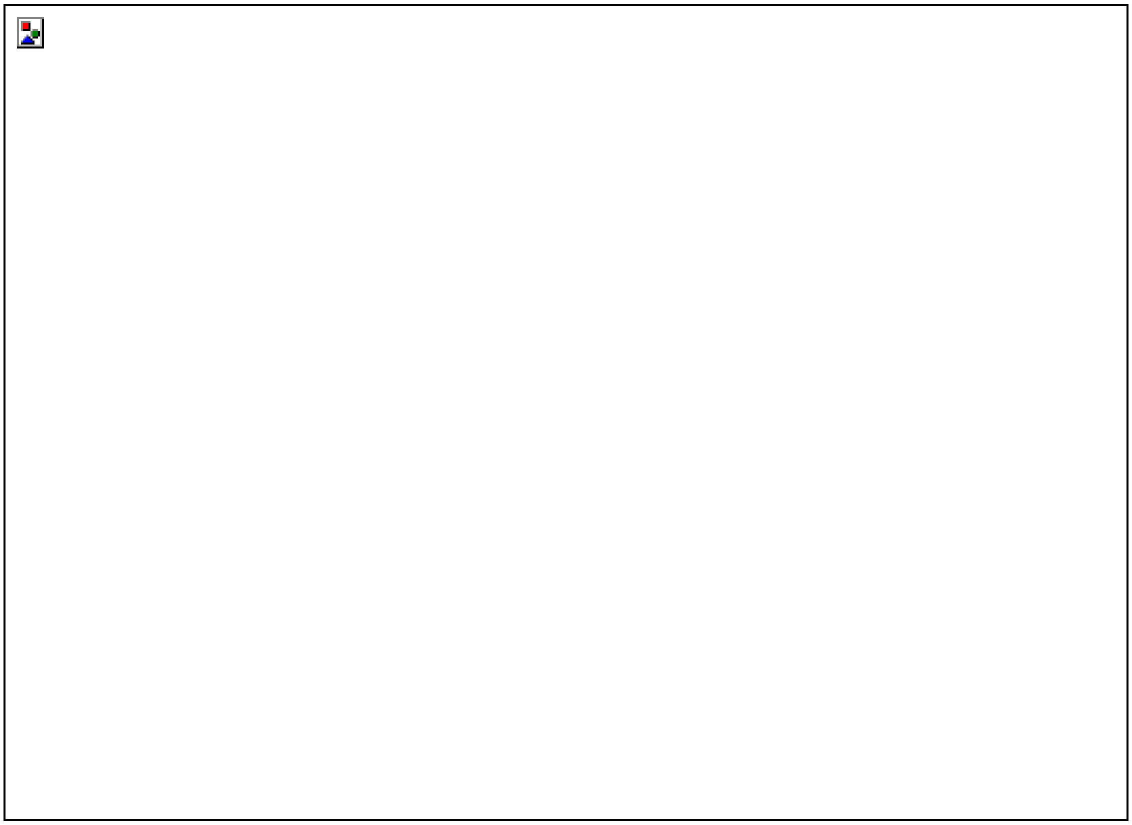
A fleet of rovers with identical capabilities and sensors were built using mostly off the shelf components. The programming experience was designed as a group experience where several students collaborated to formulate and implement their ideas to solve the tasks at hand. Ideas for solutions can be easily tested enabling a wholesome exchange to take place.

The design of the *Tulsa* rover is presented below. Each rover is equipped with a robotic arm (4 degrees of freedom), an active IR sensor installed on a 'pan and tilt' base, distance measuring sensors (one looking to the front and the second at a right angle), a pair of short range (approximately an inch) photo sensors to detect a contrast change on the floor, and a custom circuit board that provides an acoustic sensor and a voice. The rover also has two independent PM motors and PWM drivers controlling wheels on opposite sides of the robot , a rechargeable battery, and a programmable controller. The controller is programmed in C with Microchip MPLAB IDE.

Students in **EE4353 Robotics** designed and then assembled the rovers as a class project. The design of a single rover was straightforward; the difficulties encountered in producing 10 units were almost overwhelming. The instructor had to continually fight the idea that we were building 10 individual prototypes as the students wanted to incorporate new techniques in construction with each robot completed. The students in this class were for the most part juniors and seniors in Electrical Engineering although there was as Mechanical Engineering graduate student and an Engineering Physics major.

CS2503 Scientific Programming is a Freshmen / Sophomore class for mostly Mechanical and Chemical Engineering majors with a few Engineering Physics and Mathematics majors. The course uses "C" as the language of choice. These students were users of the rovers. They had no input into the rovers' design and construction.

Programming of the rover is achieved by compiling C programs on a PC and downloading the object code to the rover controller via a serial port connection or tether. The rovers were deployed on a practice field consisting of a 15ft by 30 ft black commercial grade carpet with white gaffer's tape markings. The carpet can be rolled up and stored when not in use. The field was not near the regular computer labs but was within TU's wireless LAN coverage. Laptops were available for use by the students for making small corrections to their programs and downloading to the rovers at the field site.



Rover Components (detail):

Each robot will fit in a cube 15 inches on a side when the arm is in its upright position. The controller used for the rovers was the 2004 EDU Robot Controller by Innovation First [1]. The EDU controller collects signals and processes them using a Microchip 18F8520 PIC microprocessor. The program in this micro controller takes the data, determines what state changes need to be made, and sets the outputs to the appropriate states. The robot controller comes with a default program that handles most control needs. This program is a variation of the program supplied to teams involved with the FIRST Robotics competition [2]. The program contains a fast program loop to support sensor activity with a portion of the loop protected by a timer so that control actions can made synchronously. The students modified portions of this program to provide the required functions for the rovers.

The EDU controller can be programmed in C or assembly language. The Windows program used to edit, compile, and debug the code is MPLAB IDE [3]. The compiler C-BOT is used in conjunction with MPLAB IDE to produce object code for the EDU controller. This object code is loaded into the EDU controller with stand-alone loader also provided by Innovation First that will display debug information if the controller remains tethered to the PC. Debug information is not available if the rover is running by itself.

The arm and 'pan and tilt' on each rover are provided in kit from LynxMotion [4]. The arm has servos to control each of three joints (shoulder, elbow, and wrist) , and one servo to control the gripper. The 'pan and tilt' has two servos. These six servos together with the two motors drivers that control the wheels use all the 8 pulse width modulated (PWM) outputs of the EDU controller. An active IR sensor is located on the 'pan and tilt' and can be used to determine approximate direction and distance for the arm to reach and grip objects that have reflective IR tape installed in them. These sensors provided by Banner Engineering [5] use a modulated IR transmitter and a synchronous detector to obtain a range of 50 feet in most lighting situations.

Two infra-red distance sensors (GP2Y0A02YK donated by Sharp Electronics [6]) are located so the rover can sense objects to the left and in front of the rover. The sensors are capable of detecting distances between 0.15 m and 1.5 m. The output voltage is not a linear function of distance and students will forced to provide an interpolation routine to obtain an actual distance measurement. In practice, however , only a nearness (binary) signal is required and the true distance is never calculated. These sensors can be used to follow a wall such as in a maze. The high contrast photo sensors are located at the bottom of the rover. These can be used in a line follower algorithm. An acoustic sensor (microphone, amplifier, band pass filter and detector) was designed by students in the robotics class. The design was adapted from a circuit used to start robots in a fire fighting competition [7]. The students decided to add a "voice" to the rover using a speech recorder chip similar to those used in "tapeless" answering machine. This enables the rover to provide sensor reports without a tether in place. Because the playback messages are of random length and subject to change, the timed control loop that is peculiar to the IFI software must be adapted to handle an external untimed event, i.e. an interrupt. This is a sophisticated topic for an introductory programming class. The fidelity of the recorder chips is such that the gender of the person recording the feedback message is easily recognized. We have boy and girl rovers!

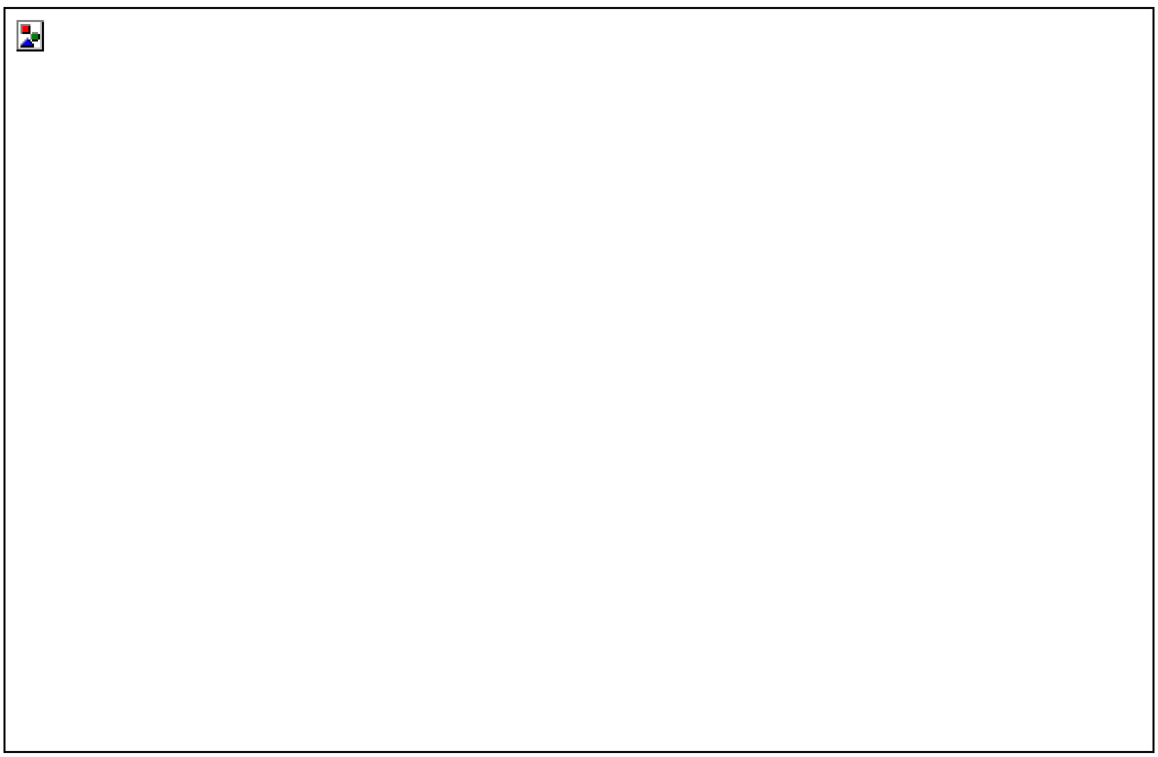
Calibration:

Although the fleet of rovers have identical capabilities and sensors, each of these motors and sensors require independent calibration. At one time, it was planned to build a sensor board that would have calibration hardware to reduce the sensor output of the black/white sensors to a single bit instead of an analog voltage. The response of the IR phototransistor in our sensor varies from unit to unit. We require a exercise to calibrate the black/white threshold voltage. Calibration is assisted by using the debug mode of the MPLAB IDE environment and the printf C-command to a debug window on the PC that is tethered to the EDU controller. This provides a flavor of engineering design not available in any regular programming lab experience.

Programming Projects:

The projects designed for the CS class in order of increasing difficulty include line-following, wall-following, object location use the active IR sensor, and object pickup using the robotic arm. These basic algorithms can be mixed to provide graded competitions for group of students. The reader may wish to verify that the well known keep-a-hand-on-the-wall algorithm for maze solving does not require a left, front, right sensor triad but only forward and side. This is why we only have two SHARP distance sensors on each rover.

We have students working on a project this summer to use the Banner sensor on the pan-and-tilt platform along with some strategically place reflection poles to obtain absolute position on the field. We hope this will lead to projects like "go to the northwest corner of the field to retrieve an object and return to the starting point."   This task will require more intense calculations than the other projects. The C-bot software has no trig functions. In fact, there is no floating-point support.



Conclusions:

A fleet of mobile robots or rovers with various capabilities and sensors were built using mostly off the shelf components by the students in our senior level EE robotics class. They were built to achieve a series of predefined tasks autonomously. These rovers were designed to be integrated into introductory computer science classes to inspire students in the practical applications of physics and math.

The curriculum and associated labs were designed to enhance the experience of taking a programming class where the labs have a physical response, thus increasing the retention of the concepts taught. The programming experience was designed as a group activity where several students collaborated to formulate and implement their ideas to solve the tasks at hand. Ideas for solutions can be easily tested enabling wholesome exchanges to take place. Calibration of the motors and sensors provides an early hands-on engineering experience. Some of the experiments involve real-time feedback from the rovers, which is not available in any regular programming exercise.

Bibliography:

[1] "Innovation First" is a mark of Innovation First, Inc. Copyright
http://www.innovationfirst.com
[2] F.I.R.S.T. robotics competition
http://www.usfirst.org
[3] Microchip Technology Inc.
http://www.microchip.com
[4] http://www.lynxmotion.com
[5] http://www.bannerengineering.com
[6] http://sharpsma.com
[7] "A Microphone Frequency Detector" Eli Kohlberg
http://www.seattlerobotics.org

Biographies:

Zachary Carpenter is a junior in Electrical Engineering at the University of Tulsa. He has been involved with robotics related research designing and building robots for over 2 years.

J.C. Diaz is a professor of Computer and Mathematical Sciences at the University of Tulsa. Professor Diaz has taught programming for science and engineering majors for the past several years.

G.R. Kane is a professor and the Chair of the Electrical Engineering Department at the University of Tulsa. Professor Kane has taught the Senior Robotics course for the several years now and enters multiple teams in the annual FIRST robotics competition.