Robotics research and education with LabVIEW

Phillip John McKerrow
School of Information Technology and Computer Science
University of Wollongong
Wollongong
NSW, 2522
Email: phillip@uow.edu.au

Abstract

Robotics research includes developing complex real-time software for testing theories and algorithms. Robotics education includes setting practical assignments on real robots and vision systems for a growing number of students. Both of these tasks can be achieved simply and relatively inexpensively using Macintosh computers running LabVIEW software, with both mobile robots and robots built from LEGO.

Introduction

Robotics research includes developing complex real-time software for testing theories and algorithms. Functions performed by the software include data acquisition, motor control, processing of sensor signals, perception of objects and navigation through the environment. Writing this software in C++ or Java can take a long time. While developing a user interface is easier in Java than in C++, Java has not proved itself to be suitable for the low-level real-time code. This has to be done in C or C++ and then interfaced to Java using the Java native interface.

The time to develop this software can delay research by months. While developing software is an essential part of research training for Computer Scientists, it is not for Mechanical Engineers. Engineers have much less, if any, formal training in writing software, and hence limited software development skills. The time to learn these skills limits the type of research projects they are willing to undertake.

LabVIEW provides a solution to both the limited skills and the limited time problems. LabVIEW is a graphical programming language developed for data acquisition and control. National Instruments, the developers of LabVIEW, sell a range of data acquisition, video capture, signal generation, and motion control cards that work with LabVIEW software on Macintosh computers. Also, there are libraries of functions that plug into LabVIEW for signal processing, computer vision (IMAQ) and motion control.

The issues of software development skills and time to develop software also impact on robotics education. Most university subjects in robotics are taught in the final year of an undergraduate degree or at postgraduate level. The reason for this is the high level of skills and broad range of assumed knowledge required by such subjects. Robotics is a multidisciplinary subject requiring knowledge from Computer Science, Mechanical Engineering, Electrical Engineering, Mathematics and Physics.

Even with this background, specifying practical assignments with robots that students can complete within the time constraints of a subject is difficult. Several approaches are taken to this problem. One is to develop most of the software for the robot and then ask the students to complete one component of it. This approach enables the student to gain experience on real robots and sensors within a clearly defined framework. However, it requires someone to develop the framework. One limitation of this approach is the cost of the robots, sensors and software. While some Universities have teaching laboratories with up to 10 robots, most are limited to one or two research robots, which severely restricts class size.

A second approach is to develop robot simulators, both for mobile robots and robot arms. Simulators have the advantage that they do not limit class size, but realistic simulations take a long time to develop.

Their biggest disadvantage is that they simulate ideal worlds, so a program that works on a simulator may not work with the uncertainties of a real-world situation. As a result, the student doesn't experience the practical problems associated with using their algorithms.

A third approach is to limit the assignments to the solution of problems in a text. The latter provides a good theoretical foundation but is rather unsatisfactory as far as gaining practical hands-on experience. A fourth approach is to build programmable cars from radio controlled cars. Each year, the Computer Science department at ANU lends several of these systems to schools. Groups of students program them to solve a robotics problem and enter their solution in a competition.

LEGO's ROBOLAB system puts a fifth approach within the reach of many universities and schools. For the last decade, a few Universities (such as MIT, ETH and Edinburgh) have been teaching robotics subjects where the practical component is a robot builder's competition (Jones and Flynn, 1993).

Each group has developed a low-cost robot kit consisting of a microcontroller card, a method of programming it, and a modular system for building a robot (often LEGO based). LEGO's ROBOLAB system uses a language called RCX to program the computer brick. RCX is a set of robot control and sensing functions that can be built into a robot control program using the LabVIEW iconic development environment.

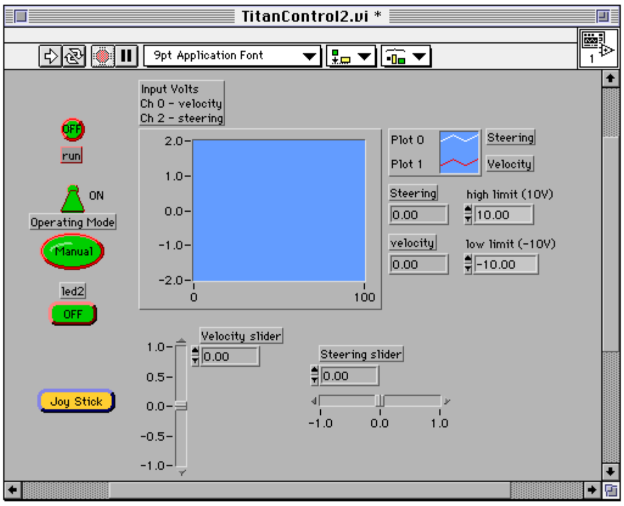

Figure 1. Virtual control panel built with LabVIEW drag and drop elements and glued together with graphical language.

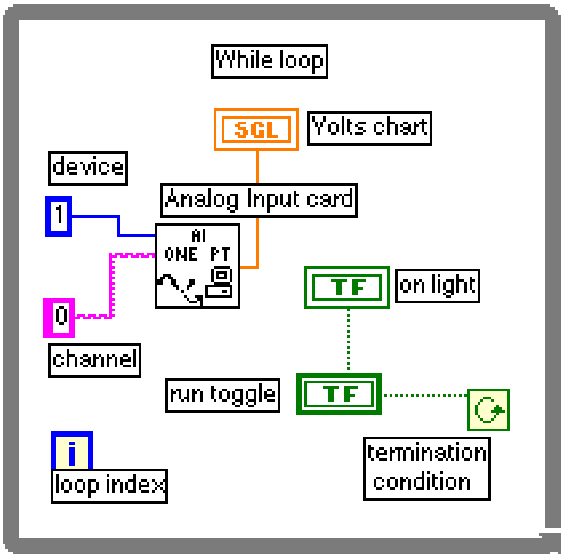

Figure 2. Example LabVIEW program diagram, a while loop with a single analog input

LabVIEW

LabVIEW is an object-oriented language that is programmed graphically. A LabVIEW program is called a virtual instrument and consists of two parts: a front panel and a wiring diagram. The front panel is a software analogue for an instrument front panel. The front panel of a virtual instrument that controls the motion of the Titan robot is shown in Figure 1. This front panel includes switches, indicator lights, a chart for displaying velocity and steering angle and sliders for controlling velocity and steering angle. Each control on the front panel is represented by a virtual instrument icon on the program diagram (Figure 2).

Programming tools are located on three pallets: a tools pallet, a controls pallet and a functions pallet. Each function (virtual instrument) is represented on the functions pallet by an icon. To add a function to a program, select it and drag it onto the diagram window. To add a control to the front panel, select it and drag it onto the front panel. Functions are connected using wires to indicate data flow through the program. The type of the data is indicated by the colour of the wire. Functions include timers, drivers for I/O cards, programming constructs (including selection and loops), and the human interaction instruments mentioned above. Virtual instruments are constructed from other virtual instruments, resulting in a hierarchy of programming layers.

Programming in LabVIEW enables the rapid prototyping and testing of new ideas. Through the use of threads and global variables LabVIEW supports multiple processes. Also, processes can be attached to a timer so that they run every *n* msec. In addition to data acquisition and control, LabVIEW has special libraries for signal processing and computer vision. Also, task specific functions not supported by LabVIEW can be written in C and linked to form a new virtual instrument for use in LabVIEW programs.

Figure 3. Titan 4 wheel-drive robot sensing beans in a vegetable garden.

Titan

Titan is an outdoor mobile robot that we built from a 4-wheel drive wheel chair (Figure 3). An Australian paraplegic was unable to purchase a wheel chair that could be driven across parks and sports fields. So he teamed up with an engineer to design a new wheel chair with large pneumatic tyres. Each wheel is mounted on the shaft of a gearbox powered by pancake motor (Figure 4). The motors are emf controlled in pairs (one pair per side) by PID controllers. The user manually controls forward velocity and steering direction with a 2D joystick.

Titan is steered by controlling the difference in velocity between the two pairs of wheels. Thus, the joystick steering input is added to the velocity on one side and subtracted from the velocity on the other. Normally, differential velocity steering of a 4 wheeled vehicle results in skid steering. This is not desirable for a wheel chair, because skid steering can damage carpets and lawn, and scuff cork floors and concrete.

An innovation in steering design, which has been patented (Pagett, 1996), enables differential velocity to achieve Ackerman steering (car like steering). Ideally, an Ackerman steering mechanism turns the front steering wheels by different angles so that the axes of all 4 wheels pass through a common point. As a result, no sideways force is applied to the wheels and hence, no skidding. In practice, car steering only achieves this ideal at one steering angle.

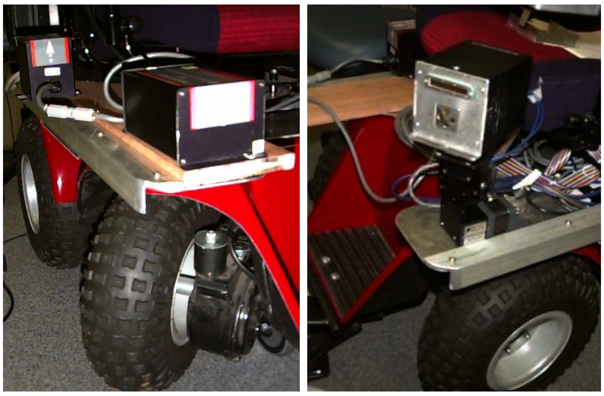

Figure 4. Titan front drive system and gyro-stabilised digital compass (left) and CTFM sonar mounted on a pan & tilt unit (right)

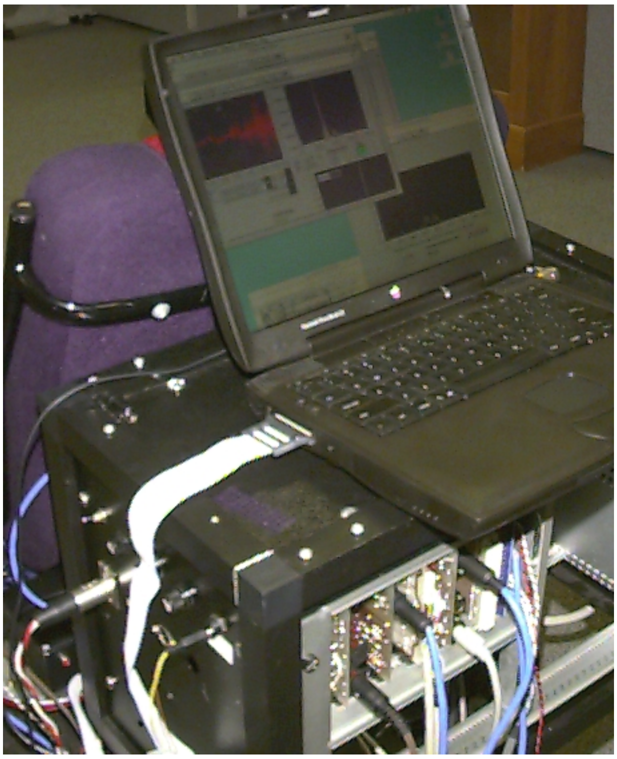

Figure 5. G3 Powerbook and electronic rack mounted on Titan

To convert the wheel chair into a robot, we mounted sensors, an electronic rack and a 233MHz G3 Macintosh Powerbook on it (Figure 5). The sensors include rear wheel velocity encoders, a steering encoder; a gyro stabilised compass, inclinometers, and an ultrasonic array sensor (Figure 4). The electronic rack conditions signals from the sensors to the computer interface and from the computer interface to the actuators (Figure 6). A general-purpose analogue and digital I/O card connects the electronics to the Powerbook via a PCMCIA slot. All equipment is battery powered.

To control the Titan robot we have written several virtual instruments (Figure 7). The sensor virtual instruments read and process the data from the sonar array, the compass and inclinometers, and the velocity and steering encoders. The actuator virtual instruments control the view direction of the sonar with a pan and tilt mechanism and the motion of the robot with the motors in the wheels. A navigation virtual instrument uses the sensor data to navigate the mobile robot.

This virtual instrument forms the intelligent connection between the perception of the environment and the action required to achieve the navigation task. A master control virtual instrument schedules the others.

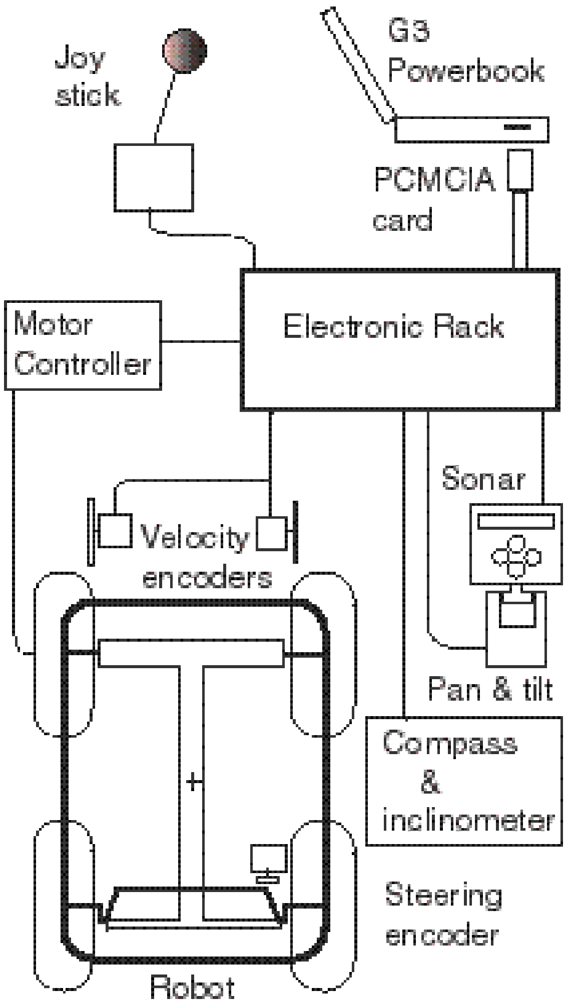

Figure 6. Electrical block diagram of Titan sensing and control equipment

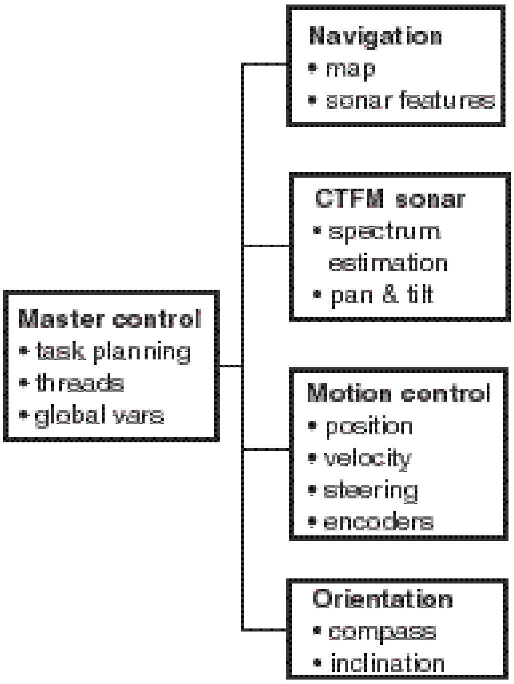

Figure 7. LabVIEW software architecture.

Portable vision

The above sensor suite does not include robot vision. Our research is into the use of ultrasonics to recognise plants (Harper and McKerrow, 1999) and to use them as landmarks (Ratner and McKerrow, 1997). Hence, our first goal was to mount the ultrasonic sensing array on the robot. However, we wish to complement the ultrasonic sensor with a vision system both to obtain a visual record of what the robot is sensing and to experiment with visual recognition of landmarks.

National instruments make a vision card, but it plugs into a PCI slot not a PCMCIA slot. Thus, before we could mount a vision system on the robot, we had to find a way of interfacing a PCI back plane to a Powerbook. MAGMA make PCI expansion chases for various Powerbook models.

The internal PCI bus of the Powerbook is extended by plugging a module into the CD bay. A screened cable connects this module to an external PCI back plane. The back plane can be powered from a general-purpose power supply or from batteries. Expansion chases are not available for the 400MHz Powerbook because of problems with the open firmware.

IMAQ is a set of vision virtual instruments that plug into LabVIEW. Some virtual instruments control and initialise the video card for single frame or continuous acquisition. Another group implements a large set of common vision processing algorithms. The system can acquire images and perform edge detection several times a second. Fast enough for a slow pan of the camera to produce a smooth video of the edge detector output.

At the time of writing we have used this vision system in a stand-alone mode, and are about to commence mounting the PCI back plane on the robot. With the installation of the PCI back plane on the robot, we will replace the PCMCIA interface card with a PCI interface card (with an identical pin out). Also, the Powerbook only has one serial port. At present we are multiplexing three serial signals into this port. To eliminate this multiplexer we will install a PCI 4 port serial card from MegaWolf. National Instruments do not have any Macintosh drivers for their serial cards, but LabVIEW supports the MegaWolf cards. These changes should make the system more robust both physically and electronically.

Education

While the above system is used to train postgraduate students in research methods, it is limited in its ability for use in course work subjects due to the cost of the robots and sensors. However, the images captured by the vision system can be saved on files either in still or movie format.

A student can book the vision system to capture a set of images for an assignment. Then they can analyse them by writing programs with the IMAQ virtual instruments on another computer. The advantage of this approach is that the students get to work with real data. For this purpose, we have purchased a LabVIEW department license for our laboratory of iMacs.

ROBOLAB

In robotics education there is no substitute for hands on experience. Many algorithms that work in simulation fail miserably in a real environment with real sensors. The capability of the sensors currently available for use on mobile robots (vision, laser and ultrasonic) is minute compared to the capability of human vision and bat echolocation. These sensors suffer from noisy signals, drift due to environment changes, ambiguity and our lack of understanding of perception. In addition, the world is uncertain: buildings are not square, floors are not flat, and people move objects. The consequence of the imprecision of sensors and the uncertainty of the world is that robotics is a difficult problem.

LEGO developed ROBOLAB to enable students to build computer-controlled systems and to experiment with them. It is reported that demand for traditional LEGO kits has declined, as children are more interested in computer games. Part of LEGO's strategy appears to be to recapture part of the market by selling computer controlled kits.

ROBOLAB comes in two forms: ROBOLAB and Team Challenge Set. The Team Challenge Set is almost identical to the commercial product called MindStorms (Taylor, 1999). Both contain the same LEGO components for building mobile robots. The only difference is that the 'Team Challenge Set' works on Macintosh Computers as well as on Wintel systems. LEGO is not saying why they have not released a Macintosh version of Mindstorms (a possible reason is that they don't have a USB interface).

ROBOLAB is based on work done by Fred Martin (1992) at the MIT Media laboratory. It consists of a computer brick: a rectangular plastic box that houses a microcontroller, batteries, and an infrared transceiver (Figure 8). On the top is an LCD display, 4 push buttons, 3 output connectors and 3 input connectors. The output connectors can drive motors or lamps. Sensors available to connect to the input connectors include a bump sensor, an infrared proximity sensor and a rotation sensor. One push button turns the power on and off. The second selects one of 5 programs to run. The third starts and stops the program, and the fourth is used for debugging.

Figure 8. ROBOLAB computer brick

Figure 9. “Nortel Networks IT Spring School for Girls” - programming a LEGO automatic house with RCX

ROBOLAB comes with student workbooks for building and programming a car, a legged insect, an automatic house (Figure 9), and a basketball hoop. The ‘Team Challenge Set’ comes with designs for a number of mobile robots and instructions for team projects. Each model is built around a computer brick.

The program for the LEGO model is developed on a host Macintosh and down loaded over an infrared link into the computer brick. An infrared link and serial cable is included in the kit. We had to specifically ask for the Macintosh serial cable. There is no transmitter available for USB, so we purchased a USB to serial converter. The Keyspan USA-28 serial adapter works.

ROBOLAB is programmed in a version of LabVIEW called RCX. RCX has 2 levels of programming: Pilot and Inventor. All programs at the Pilot level are preprogrammed and students can only change function parameters. The inventor level includes a full pallet of virtual instruments for students to build their own programs. LEGO has attempted to produce a simpler graphical language than LabVIEW that can be programmed in the LabVIEW development environment. We are told that a set of virtual instruments are available from National Instruments which enable the full power of LabVIEW to be used.

Our first experience in using this system was with the “Nortel Networks IT Spring School for Girls” at the University of Wollongong in October 1999 (Figure 9). We ran 4 two-hour sessions for groups or year 11 and 12 girls. In each session teams of girls built a model, programmed it and tested it. The girls responded very positively to ROBOLAB and enjoyed the experience of building and programming a working model. ROBOLAB easily lead them from something they knew (LEGO) to something they didn’t know (simple iconic programming).

In the second half of 2000, we plan to use these LEGO systems in a robot builder’s project in our final year subject in ‘Robot Perception and Planning’. One of the assignments will be to design, build and program a mobile robot to solve a specific problem. A second assignment will be a vision assignment using IMAQ, as discussed in the previous section. A third assignment will be a path planning assignment using a mobile robot simulator that we are developing in Java.

These assignments will give the students a range of hands on experience and enable us to evaluate both LabVIEW and ROBOLAB as educational tools. The main question we have at this stage is, ‘is the ROBOLAB system capable of building robust robots that are sophisticated enough to give students a rewarding learning experience. We suspect that we will need a wider range of sensors than is currently available.

Conclusion

By combining a Macintosh Powerbook, a MAGMA PCI expansion chassis, with National Instruments data acquisition, vision and control cards with LabVIEW software we made a portable instrumentation and control system that we can use in laboratory experiments and to control mobile robots. In a laboratory they can be powered from the mains. On a vehicle, such as a mobile robot they can be battery powered. LabVIEW is a graphical programming system. Using LabVIEW to program robots has allowed students without a Computer Science degree to rapidly develop and test complex research programs. Thus it opens up opportunities for training in robotics research to a wider range of students, and reduces the time to build experimental systems.

LabVIEW and LEGO come together in ROBOLAB and Mindstorms to produce a modular system for the design, construction and testing of mobile robots. As it is low cost compared to industrial robots it makes practical assignments possible in robotics subjects for larger numbers of students.

References

Pagett, J. M., 1996. "Steerable four wheel drive vehicle', U.S. Patent 5482125, Jan. 9.

Ratner D. and McKerrow, P.J. 1997. 'Landmark recognition with ultrasonic sensing for autonomous mobile robot navigation', FSR'97, December, Canberra, pp 552-558.

Harper N. and McKerrow, P.J. 1999. Recognition Plants with Ultrasonic Sensing for Mobile Robot Navigation, Proceedings Eurobot'99, pp 105-112.

Jones, J.L and Flynn, A.M. 1993. Mobile Robots, Inspiration to Implementation, A K Peters, Wellesley, MA.

Martin, F. 1992. The 6.270 Robot Builder's Guide, MIT Media Laboratory, Cambridge, MA.

Taylor, Ken, 1999. LEGO mindstorms - do it yourself robotics, Australian Robotics and Automation Association Newsletter, September, pp 9-10

Web sites

Australian ROBOLAB agent:	www.edex.com.au/ROBOLAB
Fred Martin:	http://fredm.www.media.mit.edu/people/fredm
LabVIEW:	http://www.ni.com
LEGO hackers:	http://www.crynwr.com/LEGO-robotics/
MAGMA PCI expansion chases	http://www.MAGMA.com/
Mindstorms:	http://www.legomindstorms.com/
ROBOLAB:	http://www.LEGO.com/dacta/ROBOLAB/

Acknowledgement

Nortel Networks funded the purchase of the ROBOLAB system through its sponsorship of the "Nortel Networks IT Spring School for Girls" at the University of Wollongong.