Int. J. Engng Ed. Vol. 21, No. 6, pp. 1039–1042, 2005
Printed in Great Britain.
0949-149X/91 $3.00+0.00

Using Mobile Robots for Controls and Mechatronics Education*

DANIEL P. STORMONT and YANG QUAN CHEN
Department of Electrical and Computer Engineering, Utah State University, Logan UT, USA. E-mail: yqchen@ece.usu.edu

Mechatronics is playing a greater role in industry and providing a realistic educational experience is becoming equally important. This paper discusses what mechatronics is and the traditional approach to mechatronics education. It then provides details about the approach we have been working on at Utah State University using inexpensive mobile robots in mechatronics education, including the hardware used, the use of MATLAB®, Simulink®, and Stateflow® for software development, and the difficulties encountered so far. The paper concludes with future plans for the mobile robot laboratory experiments and the development of a Simulink® toolbox for mobile robots.

INTRODUCTION

MECHATRONICS is a field that is becoming increasingly important with the rise in the number and complexity of embedded systems. From disk drives to automobiles, the merging of electronic, mechanical, and control systems is becoming commonplace. To keep up, we believe it is essential that the educational experience students receive at Utah State University reflect the multidisciplinary nature of modern systems. To this end, we have been working on laboratory equipment and assignments to provide an end-to-end design experience that parallels the process a student might apply in industry. Mobile robots provide an inexpensive platform for combining mechanical, electronic, and control systems to create an integrated system that provides visceral feedback to the students. Testing the system is more interesting and more realistic because the results of the design decisions are readily apparent as soon as power is applied to the robot. Because the robots are fully autonomous and self-contained, the students are able to gain experience with a truly embedded system. This is much more satisfying an experience than experimenting with a simulation or with lab equipment that is tied to a PC.

WHAT IS MECHATRONICS?

The first definition of mechatronics was provided by Yasakawa Electric Company in 1969, as follows: 'The word, mechatronics, is composed of 'mecha' from mechanism and the 'tronics' from electronics. In other words, technologies and developed products will be incorporating electronics more and more into mechanisms, intimately and organically, and making it impossible to tell where one ends and the other begins.' [1]

After thirty years, this definition had become more inclusive, reflecting the current state-of-the-art in mechatronics: 'A mechatronics system is not just a marriage of electrical and mechanical systems and is more than just a control system; it is a complete integration of all of them.' [2]

Thus, it is apparent that today's students need to be exposed to the concept of designing a complete mechatronics system, taking into consideration the electronic design, the mechanical design, and the control system (including computer programming).

TRADITIONAL APPROACH TO MECHATRONICS EDUCATION

Controls laboratories have typically approached mechatronics education through the use of desktop plants interfaced to a personal computer. An example of this type of equipment is the Quanser inverted pendulum shown in Fig. 1. The controller runs on the PC in the form of a Simulink block diagram and the power for the plant is provided by an external power supply. There are valuable insights that can be gained by using this type of equipment. It is easy to change parameters for the controller, often while it is running; it is easy to compare different controllers by changing the Simulink diagram; and a variety of displays can be used to get feedback about the operation of the system. Obviously, these types of experiments are valuable in mechatronics education. The drawback to using these plants is that they are not a standalone, embedded system. That is where the mobile robots come in. Using mobile robots for mechatronics education can take the traditional approach one step further by allowing the students

* Accepted 21 July 2005.

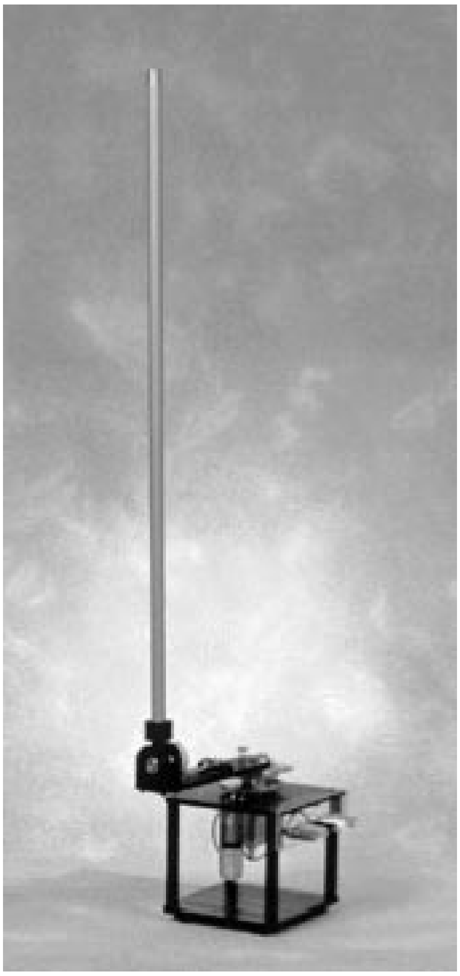

Fig. 1. The Quanser inverted pendulum plant.

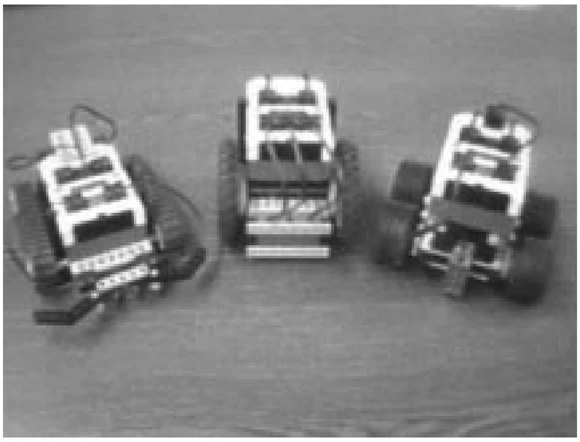

Fig. 2. Three Lego MindStorms robots.

to design a controller and then download it into an autonomous platform for testing.

MOBILE ROBOTS

We are currently developing Simulink blocksets for robots designed around two microcontroller boards. One is the RCX used by Lego MindStorms and the other is the Microchip PICmicro-based QIC board from Quanser.

Lego MindStorms robots

While it is tempting to think of Legos as toys for children, the Lego MindStorms Robotics Invention System (RIS) is a serious robotics tool. The RCX brick that contains the microcontroller is very flexible and powerful. Using an infrared tower to communicate to the RCX, it is possible to load up to five user-defined programs. It is also possible to load new firmware into the RCX, which means it can be programmed in a wide variety of programming languages, including BASIC, Forth, Java, and C. The C language is the most useful for our purpose since the MathWorks Real-Time Workshop can generate C code from a Simulink diagram which can then be compiled for use on the robot.

Figure 2 shows the variety of robots that can be built using the standard RIS kit. Because it is so easy to build different configurations of robots using the Lego parts, students can be very creative in designing and testing their own robot designs.

The biggest disadvantage to the Lego MindStorms kits are the limited input/output ports (three of each) and the small number of sensors available (although there are websites and books that describe how to design other sensors for the MindStorms).

Robots based on the Quanser QIC board

Quanser Consulting produces the QIC control board. This controller is composed of two components: a core board and a carrier board. The core board contains either a Microchip PIC16F877 or PIC18F452 microcontroller and the circuitry to program it, as well as access to all of the ports on the microcontroller. The core board is sized so that it can be plugged into a breadboard for rapid prototyping. The core board plugs into a carrier board that provides +5 VDC power for the electronics, as well as two connectors for encoders, a low-current motor supply, and a high-current motor supply. Figure 3 shows three robots we have built using the QIC board.

One of the robots uses a chassis built for a mobile robot competition sponsored by Ball Aerospace at Utah State University. The robot is a differential drive design using two high-quality DC gearhead motors with shaft encoders, an ultrasonic ranger, an infrared ranger, two contact switches, and a 9-V rechargeable battery pack. Because this robot would be somewhat difficult to replicate, we also built two robots using bases that are available commercially.

The robot with the treads was built by removing the radio control circuitry from a Radio Shack Sentinel tank and replacing it with the QIC controller board. (The Sentinel is no longer available from Radio Shack, but many discount stores and toy stores carry virtually identical RC tanks.)

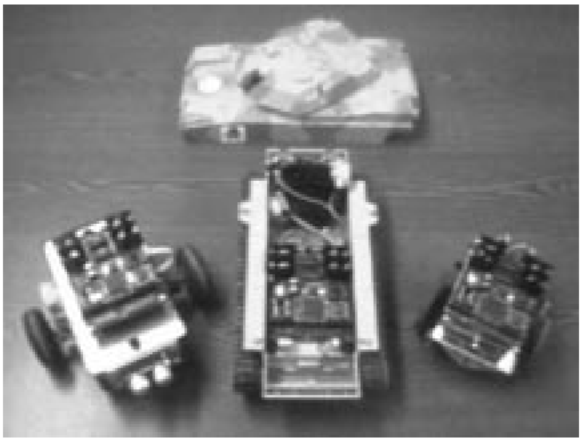

Fig. 3. The three robots built using the Quanser QIC board. The robot on the left was custom built, the other two used commercial bases.

This base has two DC motors. An encoder scheme and appropriate sensors are still being determined, but the sensors can be mounted on the turret and pointed using a hobby servo controlled by one of the low-current drivers on the QIC.

The other robot uses a Parallax BOE-Bot chassis. This is a high quality metal chassis with hobby servos that have been modified for continuous rotation. Infrared emitter-detector pairs are used as encoders on the wheels.

Other robots

Since Real-Time Workshop generates ANSI standard C code, any robot controller board that has a C compiler available for it can be used in the lab. Some of the other controller boards we have considered are the AVR Robot Controller (ARC) board, based on the Atmel ATMega16 microcontroller; the Mark III controller board, which also uses the PIC16F877 microcontroller; and the Sumo11 board, based on the Motorola 68HC11 microcontroller.

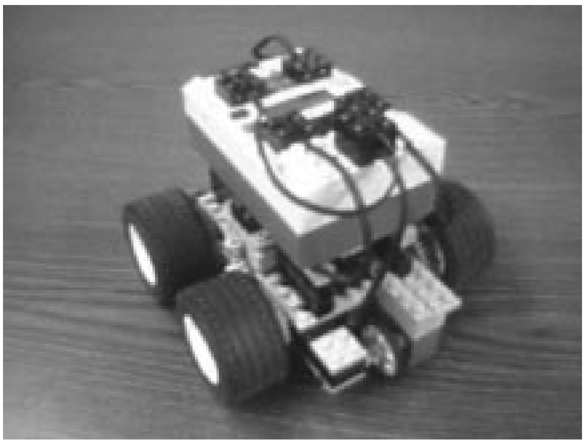

Fig. 4. The rotation sensor (encoder) is turned by the trailing wheel on the Lego MindStorms robot being used for the straight line example.

AN EXAMPLE OF A MECHATRONICS DESIGN PROBLEM

In order to illustrate the design process used with the mobile robots, consider a very simple example. Fig. 4 shows a Lego MindStorms robot that is a four-wheel skid-steered robot trailing a Lego rotation sensor. This sensor counts in integer increments of 16 counts per rotation when rotated forward and decrements an equal amount in reverse. The controller for this example will perform a very simple function: the robot will drive forward until the encoder count is ≥ 100. It will then reverse direction until the encoder count is ≤ 0 and repeat this process continually.

The first step in the design of the controller is to build a Simulink diagram. Fig. 5 shows the result. The motor control is, as might be expected, open loop since there is only one encoder and it is monitoring the robot's velocity, not the angular velocity of the motors or the wheels. To perform the desired operation, a Stateflow controller accepts input from the encoder and outputs a motor velocity and direction value to the left and right motors. (Note that a constant source could also have been used for the motor velocity value since the commanded velocity will always remain the same, only the direction flag will change.) The Stateflow controller only contains two states: *Fwd* and *Rev*. The initial state is *Fwd*, which is the default state for Lego MindStorms robots. The transition to *Rev* will occur when *Count* ≥ 100. Likewise, the transition back to *Fwd* will occur when *Count* ≤ 0.

An important step that should be included in a laboratory experiment would be to have the student simulate the operation of the controller by replacing the encoder with a ramp source starting at 0 with a maximum amplitude of 100 and replace the motors with a scope display.

Next, Real-Time Workshop is used to generate C code from the Simulink block diagram. This step is the most problematic at present since careful block design is required to ensure target requirements are taken into account during code generation. Even then, it is necessary to review the generated code to ensure it is correct for the

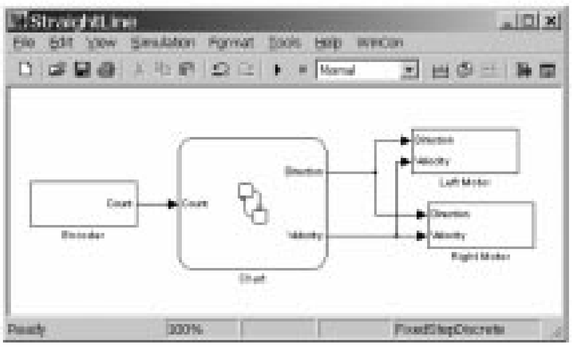

Fig. 5. A Simulink block diagram for the straight line controller.

target microcontroller. This step can be even more complicated if the target microcontroller uses a subset of ANSI C, as is the case with Not-Quite C (NQC) for the Lego RCX controller. (Although the alternative BrickOS is an ANSI C compiler.)

Once the code generation step is completed, the verified code can be compiled using the appropriate compiler for the microcontroller. Fortunately, many of these compilers are free or very inexpensive for common microcontrollers.

Finally, the code is downloaded to the target microcontroller using the appropriate method (e.g., the IR tower for the Lego RCX or a bootloader via serial port for the PIC) and the robot is tested. If any problems are encountered, the process can be repeated until the robot demonstrates the desired behavior during the testing.

It should be apparent that this process closely duplicates a typical design cycle for a mechatronics system, giving the student a 'real world' experience.

CONCLUSION AND FUTURE WORK

We believe that the use of mobile robots in mechatronics laboratories can provide students with more realistic experiences than they would get with traditional lab equipment. There are still some challenges to overcome in order to make these robots and their Simulink blocksets as simple as possible to use, so that the students can concentrate on design fundamentals, not trivial programming issues. However, the development of these robots and the MATLAB and Simulink tools for them is an ongoing effort. We will make these materials available for use by other educators on our robots for mechatronics web site (http://mechatronics.ece.usu.edu/robot/).

REFERENCES

1. T. Mori, Mechatronics, Yasakawa Internal Trademark Application Memo 21.131.01, July 12, 1969.
2. W. Bolton, *Mechatronics : Electrical Control Systems in Mechanical and Electrical Engineering,* 2nd Ed., Addison Wesley Longman, Harlow, England (1999).

Daniel P. Stormont received a BS in computer engineering from the University of Arizona in 1991, an MS in electrical engineering from the University of New Mexico in 1998, and is currently a Ph.D. candidate at Utah State University in Logan, Utah. He recently retired after a twenty-two year career in the United States Air Force and is the recipient of a graduate research fellowship from the Space Dynamics Laboratory at Utah State University. His research interests are in artificial intelligence, swarm intelligence, and autonomous mobile robots. Mr Stormont is a member of the IEEE Computer Society, the IEEE Robotics and Automation Society, the American Association for Artificial Intelligence (AAAI), and the Association for Computing Machinery (ACM).

Yang Quan Chen is presently an assistant professor of Electrical and Computer Engineering Department and the Acting Director for CSOIS (Center for Self-Organizing and Intelligent Systems www.csois.usu.edu) at Utah State University. He obtained his Ph.D. from Nanyang Technological University, Singapore in 1998, an MS from Beijing Institute of Technology (BIT) in 1989 and a BS from University of Science and Technology of Beijing (USTB) in 1985. Dr Chen has 11 US patents granted and 3 US patent applications published. He published more than 160 academic papers and (co)authored more than 50 industrial reports. His recent books include 'Solving Advanced Applied Mathematical Problems Using Matlab' (with Dingyu Xue, Tsinghua University Press. July 2004. 432 pages in Chinese. In press.), 'System Simulation Techniques with Matlab/Simulink' (with Dingyu Xue, Tsinghua University Press, April 2002, ISBN7-302-05341-3/TP3137, in Chinese) and 'Iterative Learning Control: Convergence, Robustness and Applications' (with Changyun Wen. Lecture Notes Series in Control and Information Science, Springer-Verlag, Nov. 1999, ISBN: 1-85233-190-9). His current research interests include autonomous navigation and intelligent control of a team of unmanned ground vehicles, machine vision for control and automation, distributed control systems (MAS-net: mobile actuator-sensor networks), fractional order control, interval computation, biofilm and chemotaxis modeling, nanomechatronics and biomechatronics, and iterative/repetitive/adaptive learning control. Dr Chen has been an Associate Editor in the Conference Editorial Board of IEEE Control Systems Society since 2002. He is a founding member of the ASME subcommittee of 'Fractional Dynamics' in 2003. He is a senior member of IEEE, a member of ASME and a member of ISIF (International Society for Information Fusion).