From Design to Implementation with Simulink and LEGO NXT

James C. Peyton Jones, Connor McArthur, Tyler Young
Center for Nonlinear Dynamics & Control
Villanova University, PA 19085

The use of mobile robotics in teaching has the potential to be revolutionized by a) advances in low-cost, computationally powerful target hardware, and b) automatic code generation or ‘rapid prototyping’ tools which allow these devices to be programmed directly from high-level Matlab / Simulink-based designs. This paper describes progress on a National Science Foundation and MathWorks sponsored project aimed at bringing all these elements together for practical use and benefit in the classroom. The paper describes the Villanova University LEGO Real Time target (VU-LRT) which extends the MathWorks Real Time Workshop to enable target implementation on the LEGO MindStorms NXT brick. The VU-LRT blockset enables students to access the hardware capabilities of the 32-bit LEGO NXT brick from within Simulink environment, and to automatically generate and cross-compile the necessary code for real time autonomous implementation. The tool is similar to the existing ECRobot target, but the latter is constrained by its function call architecture and non-standard use of the Simulink Real-Time Workshop. The presented toolbox is much simpler to use. LEGO hardware I/O is represented in the Simulink design mode as blocks for accessing motors, encoders, push-buttons, ultrasound sensors, light sensors and more. Blocks have also been developed to allow the LEGO target to communicate in real time with a host computer over USB or BlueTooth communications. An example system, consisting of a closed loop *dc* motor speed control system is discussed.

1. Introduction

The use of mobile robotics in education has been shown to yield numerous tangible benefits. Many researchers have found that student's motivation to learn increases significantly with hands-on robotics-based projects, [1-3]. Others have successfully used robotics as a unifying theme in introductory courses, [4-7], and still others have used robotics as way to attract women into Computer Science, [8]. Mobile robotics is also an important research area in its own right. High profile applications include the NASA Mars Rover, DARPA Grand Challenge, and DARPA Urban Challenge, though the range of research activities is very broad. Underlying much of the recent industrial and research development has been a) the advent of increasingly powerful computational devices and sensors which have become available as a natural outgrowth of Moore's law, and b) modern Automatic Code Generation or 'Rapid Prototyping' tools which allow these devices to be programmed directly from high-level system designs – without the traditional difficulties associated with low level issues, fixed point computation, or C-language programming. These tools also have the potential to transform the use of robotics or other embedded applications in education, enabling students to undertake more complex and challenging problems while focusing on the high-level pedagogical goals rather than low level issues.

In this paper, a new rapid prototyping toolbox, the Villanova University LEGO Real Time target (VU-LRT) is presented. The toolbox enables high-level designs coded in the Matlab/Simulink environment to be automatically cross-compiled for execution on the low-cost but remarkably capable LEGO MindStorms NXT brick. The paper is organized as follows. Section 2 discusses target hardware selection, as well as software alternatives, rapid prototyping tools and the choice of the Simulink design environment for this project. The specifics of the VU-LRT toolbox are presented in section 3, and a brief example is given in section 4. Finally brief conclusions and plans for future work are discussed in section 5.

2. Hardware and Software Alternatives

A wide variety of robot hardware platforms are in use within the STEM community. Of these, most devices, such as the Parralax BOE Bot [9], HandyBoard [10], ActivMedia [11] and first generation LEGO RCX brick [12] for example, are still based on 8-bit processors. Typically these machines run at clock speeds of 20 MHz or less, have 32 KB or less of RAM and are hard to program because of finite word length and memory issues. At the other end of the spectrum, the targets used in research and industry are certainly more capable, but come at a prohibitive cost. Recently, however, much lower cost 32-bit machines have become available at very similar cost to their 8-bit predecessors. The LEGO Mindstorms NXT, for example, is a 32-bit machine with twice as much memory as the older RCX version (Fig. 1) [13]. It has Bluetooth communications which allow a master unit to communicate with up to three secondary NXT units, and features 4 input ports and 3 output ports. Users can also access the inbuilt push-buttons and on-board loudspeaker. The output ports can be used to drive *dc* motors and incorporate encoder position feedback, and the input ports can be connected to an increasingly broad variety of sensors available from LEGO MindStorms or from third party suppliers such as HiTechnic and Vernier Software & Technology. The LEGO MindStorms NXT was therefore chosen as the hardware platform for this project.

Figure 1. The LEGO MindStorms NXT brick and associated peripherals

Advances in embedded hardware have also been complemented by advances in the software tools used to program these devices. Recently, for example, tools have appeared to enable rapid prototyping / automatic code generation from high-level student-friendly design environments. To varying degrees, these tools allow users first to simulate their designs, implement them on target hardware, and finally to tune system parameters while the code is actually running on the target. This development cycle is both practical and educational and is widely used in industry. Specifically, these tools include MicroSoft Robotics Studio (MSRS), LabView from National Instruments, and Matlab / Simulink from the Mathworks. The Matlab / Simulink environment which is arguably the most pervasive in the STEM community, is already tightly integrated into the research activities and educational curriculum at Villanova and other institutions. Simulink was therefore chosen as the design environment for the project.

The advantages of Matlab for developing educational robotics applications has not gone unrecognized. Dr. Behrens, from the Institute of Imaging and Computer Vision, in Aachen, Germany developed the RWTH toolbox for wirelessly sending commands and receiving data to/from a LEGO NXT platform [14]. This 'remote control' approach has the control algorithm running on the host PC with the robot acting primarily as a dumb sensor / actuator. Though simple, the approach is limited to low bandwidth control applications due to the time varying delays which inevitably occur in the host-target communication channel. A truly embedded / real-time solution, very similar to that advocated in this paper, has also been developed by T. Chikamasa in the form of the Embedded Robot (ECRobot) coder [15]. The ECRobot toolbox represents a significant advance from hand-coded algorithm implementation and has been used by the authors and others in earlier projects [16,17]. However its function-call based architecture does not conform to the normal Simulink Real-Time Workshop design process, and it imposes significant constraints on user designs. The VU-LRT toolbox aims to provide a more user-friendly blockset and to integrate more seamlessly with the standard RTW design process. However, it still builds on the same real-time operating system used by ECRobot and has benefitted considerably from this earlier work.

3. The Villanova University LEGO Real Time target (VU-LRT)

The VU-LRT provides a blockset and toolchain which extends the MathWorks Real Time Workshop (RTW) tool to enable target implementation on the LEGO NXT. The Real Time Workshop generates stand-alone C code for implementing algorithms modeled in Simulink and Embedded Matlab code. In order for the code to run on embedded target hardware, however, some target-specific modules which specify how the code is to be cross-compiled and how physical input/output interfaces are to be coded, must first be developed. The process is illustrated in Fig. 2. A user design, represented as a Simulink model, is the start point of the process. When the user initiates the 'build' process, the Real Time Workshop automatically generates the corresponding C code, as well as a 'makefile' defining the automated process for cross-compiling this code into a real-time executable. The VU-LRT toolbox provides the NXT-specific template that is used by the Real Time workshop to generate the makefile, as well as target specific code which is linked together with the model code in order to access the various NXT input / output devices. Also shown in the figure is the inclusion of 'External Mode' code which enables the user to communicate with the NXT target at runtime, as indicated by the dotted line in the figure. This feature is very useful for tuning model parameters during execution and for monitoring data, but is not yet implemented in the VU-LRT toolbox.

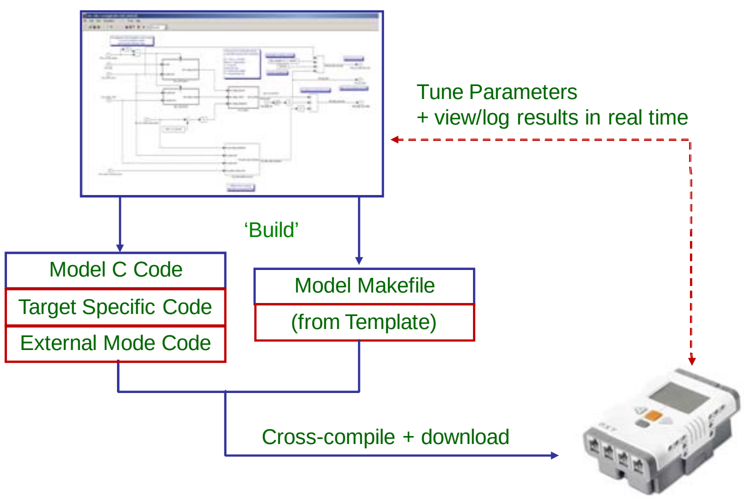

Figure 2. The Rapid Prototyping Process

In practice, these details are largely hidden from the user, for whom any target specific features are encapsulated within the user-friendly VU-LRT blockset shown in Fig. 3. The user can drag and drop any of these blocks into their design in order to access any of the NXT's inbuilt features or attached sensors and actuators. A summary of these blocks and their function is as follows:

- Battery Volts – outputs the current battery voltage
- Time – outputs the time in ms since the model execution began
- Run Button – outputs a 1 if the 'Run' button is pressed, else 0
- Enter Button – outputs a 1 if the 'Enter' button is pressed, else 0
- Sound Tone – sets the frequency and duration of tones driving the internal loudspeaker
- DC Motor – sets the applied motor voltage as a percentage of battery volts
- Encoder – outputs the number of encoder pulses received as the motor rotates

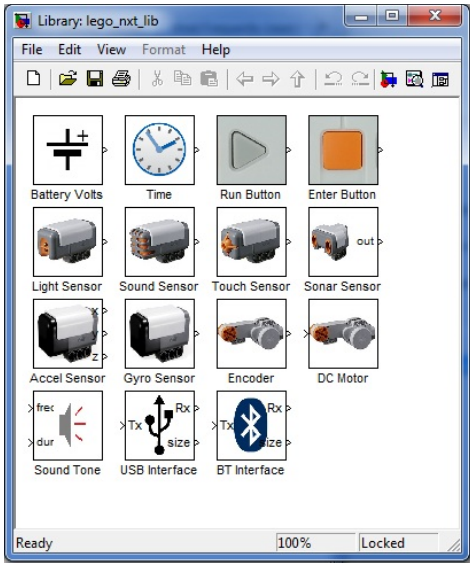

Figure 3. The VU-LRT Blockset

- Light Sensor – outputs a measure of the light received by the light sensor
- Sound Sensor – outputs a measure of the sound intensity received by the sound sensor
- Touch Sensor – outputs a 1 if pressed, else 0
- Sonar Sensor – outputs the distance to the closest object in view by the sonar sensor
- Acceleration Sensor – Outputs acceleration data in three axes (x, y and z)
- Gyro Sensor – outputs the rotation rate of the sensor
- USB Interface – enables the model to communicate with a host PC over USB
- BT Interface – enables the model to communicate with a host PC using Bluetooth

Note that the USB and BT interface blocks provide a means for transmitting and receiving data between the host PC and the NXT during runtime. This is a very useful feature for communicating specific data values or vectors between host and target, although it is not as flexible as the external mode feature described above.

5. Example

Consider(redundant) a simple speed control requirement which might form part of a larger robotics system design. Motor speed is a function not only of the applied voltage, but also by the load. Open loop speed control by setting the motor voltage will therefore lead to speed variations when load disturbances, such as the robot needing to climb a hill, are encountered. However, a constant speed can be maintained using encoder feedback to detect differences between the actual speed and the setpoint, and passing this error through a standard Proportional, Integral, Derivative (PID) controller. Such an algorithm cannot be implemented effectively using the ‘remote control’ approach because of the (time-varying) delays in the communication channel, and the tight timing requirements needed for accurate speed estimation. However, the closed loop control algorithm is readily implemented using the VU-LRT blockset as illustrated in Fig. 4.

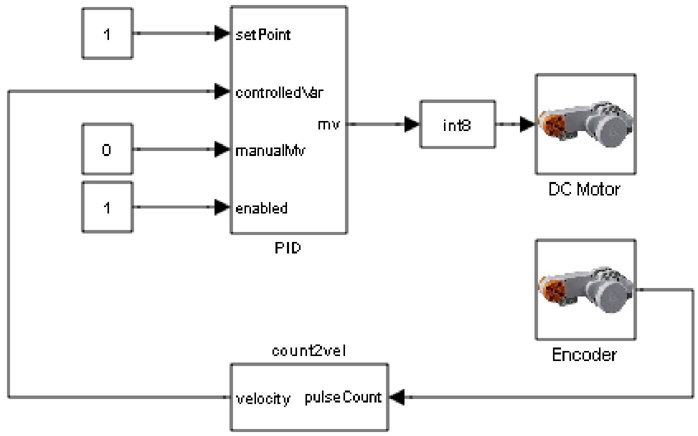

Figure 4. A closed-loop dc motor speed control application

The controller proportional, integral and derivative gains are set by means of the dialog shown in Fig. 5. When the user initiates the build process, the model is automatically transformed into a C program which is seamlessly cross-compiled and downloaded to the target over a USB cable. The downloaded executable is then selected using the brick's menu/button interface, and executed. Students can then see for themselves how the motor torque increases so as to maintain speed when loads are applied. They can also repeat the experiment with different controller gains in order to observe how this affects the resultant closed-loop behavior.

6. Conclusion

Advances in low-cost target hardware, and automatic code generating software tools provide an opportunity for educators to engage students in challenging embedded system applications without the traditional difficulties associated with low-level hardware programming. In particular, the Villanova University LEGO Real Time target (VU-LRT) enables users to access the capabilities of the LEGO NXT brick (a powerful 32-bit machine), directly from a high-level Simulink environment. The motivation and design decisions underlying the development of the toolbox have been discussed, together with some details of its architecture. From a user's perspective, the VU-LRT blockset provides a strong base set of input/output blocks which can be used like any of the other standard Simulink blocks in student designs. Further work is required to add 'external mode' capability to the toolbox, but Bluetooth and USB communications are already in place. Future work will also focus on the design of exemplary laboratory modules which demonstrate how the toolbox can be used to enhance student learning in a classroom environment.

Acknowledgements

The authors gratefully acknowledge support for this project from the National Science Foundation (DUE No. 0837637 [18]), and the MathWorks Inc. This work is neither endorsed nor

maintained by the LEGO group. MindStorms, NXT and LEGO are trademarks of the LEGO group. Please direct all enquiries to the authors.

Bibliography

1. L. Greenwald, and D. Artz, "Teaching artificial intelligence with low cost robots," In Accessible hands-on artificial intelligence and robotics education, ed. L. Greenwald, Z. Dodds, A. Howard, S. Tejada, and J. Weinberg, pp. 35-41. Technical Report SS-04-01. Menlo Park, CA: AAAI Press, (2004).
2. S. Coradeschi and J. Malec "How to make a challenging AI course enjoyable using the RoboCup soccer simulation system, in RoboCup-98: Robot soccer world cup II: Lecture notes in artificial intelligence, vol. 1604, pp.120-124, ed. M. Asada and H. Kitano. Berlin: Springer, (1999).
3. M. Goldweber, et al. "The use of robots in the undergraduate curriculum: Experience reports," Panel at 32nd SIGCSE Technical Symposium on Computer Science Education, Charlotte, North Carolina..
4. F. Klassner, "Robotics as a Unifying Theme for Computing Curriculum 2001", National Science Foundation DUE #0088884, (2001).
5. D. Miller, "Walking Before Running: Filling the Freshman Engineering Gap by Building Mobile Stiquito(TM) Robots", National Science Foundation DUE #0088158 (2001).
6. N. Chao, "A Low Cost Hands-On Laboratory Experience for Introductory Engineering Students", National Science Foundation DUE #0125583 (2001).
7. L. Fairchild, "Robots in an Introductory Survey Course in Computer Science", National Science Foundation DUE #0087963 (2001).
8. N. McNulty, "Understanding Technology through Robots and Multimedia", National Science Foundation DUE #0088370 (2001).
9. Parallax Inc, website: http://www.parallax.com/
10. The Handy Board, website: http://www.handyboard.com/
11. Mobile Robots Inc, website: http://www.mobilerobots.com/
12. LEGO MindStorms RCX, website: http://www.lego.com/
13. LEGO MindStorms NXT, website: http://www.lego.com/
14. A. Behrens, et al. "MATLAB meets LEGO Mindstorms- A freshman introduction course into practical engineering", IEEE Trans. on Education, Vol.53, No.2, (2010), 306-317
15. T. Chikamasa, "Embedded coder robot NXT instruction manual", www.mathworks.com/ matlabcentral/fileexchange/13399/, 2009.
16. McNinch, L. C., Soltan, R. A., Muske, K. R., Ashrafiuon, H., Peyton-Jones, J. C. "An Experimental Mobile Robot Platform for Autonomous Systems Research and Education", Proceedings of the 14th IASTED International Conference on Robotics and Applications, (2009): 412-418
17. McNinch, L. C., Soltan, R. A., Muske, K. R., Ashrafiuon, H., Peyton-Jones, J. C. "Application of a Coordinated Trajectory Planning and Real-time Obstacle Avoidance Algorithm". Proceedings of the 2010 American Control Conference, June 30-Jul 2, Baltimore MD, (2010).
18. J.C. Peyton Jones, F. Klassner, S. Kulkarni, C. Nataraj, "Introducing undergraduates to complex systems through rapid-prototyping of low-cost, networked mobile robots". National Science Foundation DUE No. 0837637.