Sean Brennan, Pennsylvania State University

Dr. Sean Brennan has been an Assistant Professor of Mechanical Engineering at Penn State University since 2003 and shares a joint faculty appointment with the Pennsylvania Transportation Institute. Since 1998, he has published over a dozen papers on topics ranging from systems dynamics, modeling, identification, and control; robust control; and vehicle chassis dynamics. His current areas of study include modeling and experimental validation of vehicle dynamics, human-in-the-loop experimental testing, novel sensing and hardware systems, and fault-tolerant sensing and control.
Mechatronics and Systems Instruction Across Graduate, Undergraduate, and Research Applications Using Rapidly Reconfigurable Hardware

Abstract
A challenge with the development of any new laboratory is the cost-effective use of hardware resources. This work discusses the development of a reconfigurable data-acquisition architecture across three different application areas in university mechatronics and control systems instruction setting: undergraduate education, graduate education, and graduate-level research. An analysis is offered of the different operational and educational requirements across these different levels of instruction. In many cases, these educational tiers present non-complementary requirements including different expectations on ease of use, durability, compatibility, software complexity, and performance. This work presents a laboratory development strategy that assists in balancing research and teaching while simultaneously fostering new activity in both areas.

Introduction
When faced with developing laboratory or research hardware across undergraduate, graduate, and research level applications, the historical solution at many educational institutes is simply to purchase and support distinctly different hardware data-acquisition solutions between the three separate areas. Not only is this expensive and unduly time consuming, but also it artificially breaks a natural continuum of instrumentation education across levels of instruction.

There are several means to address the issue of limited time and resources. The simplest is to limit student exposure to hardware, and sadly this is an approach widely used. To counter this budget-centric focus, educators have long anecdotally claimed that interactive laboratory experiments and problem-based learning (PBL) foster a stronger education. Fortunately, there are very comprehensive studies in the recent literature that unquestionably validate problem-based learning\(^1\) and leave no doubt that hands-on interaction is essential to efficient learning.

Another way to provide student exposure to hardware is to seek out low-cost “trainer” type experimental systems. Examples of these abound in the literature\(^2^{10}\), and while these are useful for primarily undergraduate education, their applicability to graduate education and research is clearly limited.

Another option is to purchase or develop a few research-grade systems and allow controlled student access to these devices. While this method is also commonly used\(^11^{18}\), it is generally unclear how well such systems scale in number or durability to situations where hundreds of students or more utilize the equipment.

Yet another option is to pool resources across departments to form a lab facility with sufficient support to hire a controls lab developer that assists with research deployment\(^19\). This very unique and successful approach requires interdepartmental coordination at a level beyond the feasible effort level of an individual faculty member, especially one just starting at a new institution.
This article presents a methodology employed at Penn State University starting in 2004 and continuing to the present employed to integrate research, graduate education, and undergraduate education. The situation at the start of the project was that the author, as a new faculty hire, was faced with the simultaneous challenge of developing a research instrumentation platform, a graduate Mechatronics course, and a controls laboratory suitable for undergraduate system theory and control. Nearly all new faculty are faced with a similar challenge and situation, albeit with different subject material. It is therefore it is hoped that the insights offered in this article provide some guidance for long-term integrative planning of research and teaching across disciplines.

To summarize and illustrate the synergistic integration of this article, the uses of the hardware and software involved are generalized at each of the levels education in Fig. 1. This figure also provides a quick reference for the remainder of the article on how the same core equipment, in this case a Digital Signal Processor (DSP) is reconfigured for each educational purpose.

For reasons justified shortly, the hardware platform chosen for cross-platform integration was the TMS320C6713 DSP by Texas Instruments® mounted on a developer’s kit (DSK) component board manufactured by Spectrum Digital®. The processor operates at 225 MHz with many key mathematical operations occurring in a single processor cycle. This specific processor was particularly attractive because of its extremely low cost including donations and greatly reduced pricing. Yet the system offers extremely high performance in floating point operations and memory access that support research-grade data-acquisition.

This synergistic usage of equipment across research and teaching has led to a number of insights and “lessons learned” that are conveyed anecdotally in this paper whose remainder is organized as follows: First, a discussion is presented of the capability requirements of research-grade, graduate, and undergraduate data-acquisition equipment focusing on the conflicting and
synergistic aspects of each problem area. This leads in to a discussion of equipment selection. Next, issues of deployment are presented in the three education areas: research, a graduate course, and an undergraduate lab. The focus of these case studies is to outline the constraining technical specifications and development hurdles. Next, a section is presented discussing the tradeoffs, e.g. the advantages and disadvantages of the proposed methodology. Conclusions then summarize the main results.

**Capability Requirements: A Three-Tier Problem**

The instrumentation needs of undergraduate education, graduate education, and research in control theory are fundamentally driven by the project or educational goals. For mechanical control systems which are the focus area of this discussion, there are commonalities in all stages. However, the criteria for a successfully designed data-acquisition system is remarkably different depending the application, leading to conflicts. Both the synergies and conflicts are summarized below:

**Synergies:** All data-acquisition systems share the following needs and desirable features:

- Low cost of ownership and purchase,
- Simplicity in the interface,
- High performance, at least as measured by the mechanical systems controlled by the system.
- Flexible usage

**Conflicts:**

- The relative importance of cost and performance is relative. If a research project is only feasible with the use of a very expensive piece of equipment, then that equipment will almost always be procured. The opposite is generally true of an undergraduate lab equipment.
- Reliability is judged by the expertise of the user. Graduate students are generally able to cope with (and sometimes expected to fix) equipment failures. Undergraduates do not have the time available.
- Portability of the equipment is often required for undergraduate laboratory exercises and in-class demonstrations. This constraint generally doesn’t exist to such an extent for graduate classes or graduate research, where the small number of students allows more direct access to the equipment, e.g. everyone can “crowd around.”
- There are a very large number of undergraduates to use the equipment, moderate number of graduate students, and a few students using equipment for research projects. Hence, the students least likely to be trained in use or repair of the system are most often using (and hence breaking) the system.

These capability requirements lead to an equipment selection that seeks balance between different educational needs.

**Equipment Selection**

Among the capability requirements listed above, the performance needs of graduate research are paramount. If these needs are not met, then there is little chance that teaching equipment will be used within graduate research. This will greatly diminish a faculty’s interest and “buy-in” in supporting the equipment to a level described in this work.
Therefore, the first task in equipment selection at Penn State was to identify the most constraining research usage of the equipment. For the author’s research activities, the study of high-speed vehicle dynamics and automation, the most constraining data-acquisition event was the coordinated control of a real-time vision system with a mechanical control loop. The use of an embedded vision system requires extremely high data transfer rates and processing speeds.

Another key research constraint was for the system to be mobile in a stand-alone embedded platform that consumes little power (e.g. for use in vehicles, aircraft, robots, etc.). This research constraint led to selection of a mobile robot as an ideal teaching and hardware platform because it encapsulated research needs while at the same time providing an accessible and familiar system concept to the students.

**Research Deployment**

The DSP-based system was first deployed at Penn State in a research setting to instrument a test vehicle for a research project. Because the DSP system was relatively low cost ($1000) and highly integrated with MATLAB® and Simulink® through a parallel-port interface, the extension to a vehicle, Differential GPS system, and steering sensors was relatively straightforward. The architecture is shown in Fig. 2.

![Vehicle instrumentation architecture using DSP architecture similar to the course usage.](image)

The dual Texas Instrument DSP’s are the heart of the current data acquisition system. Interfacing with the DSP chips is accomplished through standard address and data-bus pin headers. The pin headers allow for expansion via custom designed daughter cards such as the Digital I/O and CAN boards used in this project. The Texas Instruments® DSP’s allow for the boards to be programmed via Code Composer Studio® (CCS) – a standard C-code programming suite included with the developer’s kit.

Because details of this system can be found in numerous other publications by the author and his students\(^{20-23}\), the focus of this work is to provide details on the educational deployment primarily.
Graduate Deployment
Starting in 2004 and continuing to 2005, the Department of Mechanical Engineering at Penn State University and Texas Instruments® jointly sponsored the development of a new graduate-level course, “Advanced Mechatronics,” that focused on immersive learning via in-class development of high-performance embedded robot systems. Central to the class teaching structure was the use of intensive laboratory experiences in which students developed code and hardware necessary to operate mobile robots for advanced tasks. An example of an advanced task, for example, would be vision-based tracking and recovery of a soda can.

Construction of the robot began during the Summer of 2004, a time period corresponding to the hire of the first graduate student in the author’s research group (not a coincidence). Working closely with the graduate student, the author taught the principles of circuit board layout, robot construction, and shared previous mechatronic designs - and mistakes! - from the author’s previous institution. The graduate student then constructed the first prototype of the mobile robot (Fig. 3), and based on this experience developed subsequent designs of the system.

Course structure: From the beginning, the focus of the course design was to break away from the typical classroom style of shallow coverage of many topics (particularly in Mechatronics), and instead foster exploration of only one or two topics in a single project. This was intentionally to transition the classroom experience more toward the nature of the typical industrial project, where the focus is solely on one project but where work invariably isolates (or “snags on”) one or two topics of key importance. The goal of the course development was therefore move away from the notion of generic laboratories supporting challenging lecture content, and instead use flexible and more generic lecture content to support very challenging labs.

After construction of the robot, eight graduate-level laboratory activities were developed that parallel various stages of the robot construction process. The labs in total occupied approximately 32 one-hour lecture periods, and each lab was designed to be completed in either two or four class periods depending on the complexity of the task. Each lab period began with approximately 10 to 15 minutes of lecture to guide activity, and an additional lecture was provided before each lab segment to allow introduction of lab content.
An additional 3 class periods were allowed for a final contest-style lab assignment that did not include any new hardware or software material, but did require synthesis of all topics in the class. An example final contest is shown in Fig. 4. In this contest, students applied previously learned techniques to program their robot to autonomously find a soda can in a maze, and carry it to a ‘recycle bin’. The bin’s position varied, but was always found under a light in the top section of a maze. The robot was required to pass over certain “detection pads” (A,B,C) to turn on the recycle bin light.

The final 8 lectures of the course were dedicated to class content of the student’s choosing which generally consisted of integration of course material into research. Because nearly all the students taking the course were first-year graduate students, many were initiating research endeavors requiring new equipment or instrumentation setups. Students were encouraged to meld their research tasks with laboratory tasks, a synergy that led to many class-to-research project synergies discussed shortly.

Typical Lab setup: The typical lab setup focused on directed questioning techniques in the lab write up. To use one example, Lab 6 had the following task: “Using only two sensors, make the robot do right-wall following around the classroom’s outer wall edge at the fastest possible speed.” To engage the students, questions were asked such as: Can you do it faster than your neighbor? Students were then asked to define the task before doing anything, again with directed questioning including: What aspects of the system design that they think will most limit their speed? What aspects do they plan to study? What are they assuming? Where do they place their sensors? What are the tradeoffs with each sensor? Students were then encouraged to explore freely: How do you intend to investigate design space? As exploration commenced, students were asked which of their previous questions appear to be more important? This led to planning, where students were asked to write down and commit to a methodology that will quantitatively answer their questions. They then solved their problems by completing their measurements (takes ~ 1 hour). Finally, the lab asked students to evaluate themselves. For example, how sensitive was their algorithm to wall conditions i.e. will it work with non-right angled rooms? Were their assumptions reasonable? What was the single largest limiting factor limiting their speed?

Listing of Lab Activities: The above represents the design structure for one of eight labs. The simplified syllabus for the whole class is listed and summarized below:

1. BIOS: Students learn how to compile programs, comprehend basic compiler mistakes, review binary logic, etc.
2. GUI’s / Networking: Students learn how to design MATLAB® GUI’s, comprehend serial ports protocol, apply protocols to their robot.
3. Daughter cards: Students synthesize and fabricate a new I/O card (!).
4. Glue Logic: Students analyze an existing parallel interface to an analog output device, and synthesize a new one for analog input.
5. DC Motor Control: Synthesize a method to control velocity of robot while compensating for friction, evaluate two different control techniques to steer the robot.
6. Sensors: Students demonstrate knowledge of basic feedback control by synthesizing a method for the robot to navigate an arbitrary maze (often an arbitrary room) using only two sensors.
7. Actuators: Students find and pick up a soda can evaluating different methods to sense and pick up the can the fastest.
8. Vision: Synthesize an algorithm to find and approach a light, and set the can directly under the light.
9. Maze: Apply previous techniques to have your robot find a soda can in a maze, and carry it to a ‘recycle bin’, found under a light in a different location in the maze.

Undergraduate Deployment
In 2005 to 2006, based on the success of the graduate program, the Department of Mechanical Engineering at Penn State University and Texas Instruments® again supported further laboratory development, but in this case the generalization of the hardware to serve as the experimental system for the department’s undergraduate systems and control courses. While many of the graduate labs were easily converted to undergraduate labs and final projects, the programming interface of the graduate-level work (programming in C) was clearly inappropriate for the undergraduate student body. It was clearly necessary to develop a separate set of software tools and equipment specific to undergraduate education.

Pendubot Project and Final Project for Dynamic Systems: In 2006, a laboratory project was developed to provide a hands-on learning component to engineering students in “ME 440 - Modeling of Dynamic Systems” by use of a non-linear, unstable system. Effective modeling and control of this system requires derivation of system equations with Lagrangian methods, linearization of these equations for analysis with ME 440 tools, understanding of state feedback control theory and familiarity with MATLAB®/Simulink® software.

With the assistance of an undergraduate senior project group, the author developed and tested several concepts for instructional use. These concepts evaluated the reliability, durability, and aesthetics of the system. The final design is a double pendulum system called the Pendubot, which is shown in Figure 5 below. The Pendubot is an under-actuated two link robot. Its main components are two vertically mounted links, a motor,
and two encoders. The motor is mounted at the shoulder joint and provides the driving torque 
for the system, while the two encoders provide system feedback. One is mounted on the motor 
and provides the angular position of link1. The other encoder is mounted at the elbow joint (top 
of link2) and provides the relative angular position of link2 with respect to link1.

The students were then asked to control the Pendubot to stay in the unstable “down-up” position 
shown in Fig. 5 where link1 balances in front of link2. In order to do this, they had to derive and 
linearize the system model, then apply state feedback control via a Simulink® to the DSP.

To implement this project with the aforementioned data-acquisition system, the DSP was 
programmed to interface to a PC system through the parallel port as an external data acquisition 
system. This allowed real-time, high speed (1 kHz) sampling rates, and additionally allowed the 
students to avoid programming the system in C code as used in the graduate course, but instead 
design and test their system directly in Simulink®.

Student testing of the system resulted in positive results. Students in the course “ME 452 - 
Undergraduate Robotics” were able to complete the pre-lab worksheet within 2-3 hours, 
complete the lab activity itself within around 2 hours. So positive were the results, that one of 
the members of the student group developing the equipment recently told the author that he 
sleeps with a picture of this system over his bed!

**Tradeoffs: Advantages**

Foremost among advantages of the described approach is the ability to rapidly switch hardware 
as needed from one application area to another. It only takes a few minutes to change over a 
piece of equipment from a graduate lab to an undergraduate lab, or to borrow research equipment 
to get a teaching lab working quickly. Further, having so many systems around in various 
applications provided an inherent “parts supply” in the case that something went wrong.

Another unanticipated benefit of the above integration is the cross-training between graduate and 
undergraduate students. If a laboratory system stopped working, not only was research hardware 
on hand to substitute for that equipment as discussed earlier, but graduate students highly trained 
in the equipment were often the TAs for the undergraduate course and were also present. In this 
manner, many system faults and bugs were readily worked out with minimal assistance from the 
author after setup.

Another unanticipated benefit of the integration efforts described here were the number of 
erendipitous meetings between faculty of other students and the author, in order to set up new 
research projects and learning activities. So heavy is the demand for such capability that the 
author is now often having to turn away interested parties due to possible over-commitment of 
equipment resources.

**Tradeoffs: Disadvantages**

Without question, one of the largest hurdles in deploying an advanced toolset such as a DSP for 
student use is the lack of resident expertise in such systems at the onset of development. While 
many students and faculty in the department are familiar with microprocessor-based systems 
such as the Stamp® or Atom® series by Parallax®, or the Microchip® PIC systems, Atmel®
processors, or the Motorola® series, there are relatively few students or faculty trained in DSPs at a level allowing creation of an embedded hardware system and supporting software. Fortunately, the issue of lack of expertise on a system architecture is a problem that largely fixes itself as deployment into classrooms proceeds: after teaching a graduate course of 15 students, odds are strong that at least one or two will be available as TAs in the future. However, this key issue about first-time training did affect the nature of system deployment and which students were recruited first into the research group.

Another area of difficulty was that hardware purchased for educational use could not be transferred to research applications due to budget restrictions. Hence, whenever something broke in a teaching setting, the research hardware could be substituted in. But when the research hardware broke, there was no means to substitute the teaching hardware. This limitation is becoming more noticeable as the systems age, since repair and upkeep are replacing first-time assembly as the primary hardware focus.

Another unanticipated challenge to the system usage is the need to support industry-level development tools. These tools, namely the compiling software, require computing permissions not typical of the normal student or even the normal faculty. Because undergraduates are not normally allowed administrator privileges on campus computers, workarounds had to be found to allow some of the software systems to operate that assumed these privileges to be available.

Conclusions
This work discussed the development of a reconfigurable data-acquisition architecture across three different application areas in university mechatronics and control systems instruction setting: undergraduate education, graduate education, and graduate-level research. An analysis was offered of the different operational and educational requirements across these different levels of instruction.

Taking the advantages and disadvantages in balance and having recently completed the hardware development of such a complex system, the author would strongly encourage others to foster such a tight integration of data-acquisition capabilities within their respective research and teaching endeavors. Within other laboratories on campus - fuel cells, fluids, combustion - synergies between teaching and research hardware development are being fostered with similarly successful results.

Acknowledgements
The author was fortunate to have worked on nearly identical platforms as a student at the University of Illinois at Urbana-Champaign. The author wishes to thank Dan Block at UIUC for all his support in developing this system and for graciously donating time, code, and example laboratories that served as the framework for the labs described herein. Without his expertise, guidance, and shared code, a project endeavor of this magnitude would not have been possible for a single faculty or even department worth of faculty!

Bibliography