

A Multidisciplinary Digital-Control-Systems Laboratory

Gregory L. Plett, David K. Schmidt
University of Colorado at Colorado Springs

Abstract

This paper describes a multipurpose and multidisciplinary control-systems laboratory that is being developed at the University of Colorado at Colorado Springs. It is shared by Electrical and Computer Engineering (ECE) and Mechanical and Aerospace Engineering (MAE) students, allowing more efficient use of space and equipment, better use of funds, and elimination of overlap among individual departmental labs.

The composition of the laboratory and its use with an introductory feedback-control-systems course has been described by Plett and Schmidt.¹ In this present paper, we build on the previous work and outline how the lab is being used to augment digital control systems courses at the senior undergraduate level and graduate levels. Experiments and advanced student research projects (illustrating effects particular to digital control systems) with a magnetic levitation device and a control moment gyroscope are described.

We have found the labs to be very helpful in aiding student understanding of control-systems concepts. Student comments indicate that real learning has taken place by using a hands-on lab experience that would have been missed if a purely theoretical approach had been taken.

I. Background and Goals

The control-systems laboratory at the University of Colorado at Colorado Springs (UCCS) had not been paid much attention for years. One major deficiency was that it had not a single device to control! All lab experiments were accomplished via simulation, either on a *Comdyna GP-6* analog computer,² or on one of the lab's digital computers using Matlab and Simulink by *MathWorks*.³

Simulation using either method has limitations. The need to control real hardware, and not just simulations, is known to all who design and build real control systems. How this applies to control-systems education is emphasized in a paper by Bernstein.⁴ Modeling and simulation rarely capture the complete picture—physical system identification is required; control experiments often focus attention on performance and implementation issues that are overlooked and difficult to capture in simulation; experiments can reveal whether or not assumptions made when making a control design are realistic; and experiments provide a way to identify control methods that seem to work under real-world conditions as well as those that clearly don't. This final point leads to real learning.

One opportunity arising from the lab's neglect was that we were free to start from scratch with its redesign, and when choosing the dynamic systems that would be the primary focus of experiment.

Bernstein's paper discusses a number of devices he built to demonstrate different control concepts. We could not duplicate his approach as we could neither afford the time to build these experiments ourselves, nor did we have the budget to outfit many workstations. We also felt that it would be of greater benefit to the students if we required that they learn to control many aspects of only one or two devices. Then, the dynamics of only a few systems need be thoroughly understood; hence, more time may be devoted to studying how control-systems theory applies.

We came across another article promoting the control-systems laboratory at the University of Illinois at Urbana-Champaign.⁵ An appealing quality of this facility is that it is shared among several departments. The control-systems laboratory at UCCS had previously been housed and operated by the ECE department, but a new MAE program in the college needed similar facilities.

We concluded that a revived laboratory was essential, and should meet the following goals:

1. Hands-on: The new lab should promote control-systems education with experimentation, requiring identification and control of physical device(s). The laboratory course should be designed to complement and synchronize with the lecture course in order to best reinforce concepts learned in class with hands-on experience.
2. Economy: As much as possible, space, money and student time should be economized. A multidisciplinary facility, shared between ECE and MAE classes would allow efficient use of space and equipment, better use of available funds, and elimination of overlap among individual departmental labs. Focusing experiments on a single device rather than a plurality of devices would result in economies of space, money and student time.

To achieve these goals we carefully planned the new laboratory. As part of this process, we consulted with local industry. The advice we received was very helpful to us, and the hardware-in-the-loop laboratory configuration we implemented is useful for both educational and training purposes as it is very similar to that used by these companies when designing their own control systems. Local industry support has continued to be very strong. For example, one company has supplied instructors for our control courses and lab courses on an honoraria basis.

When planning the lab, we were careful to select apparatus that would allow us and our students to exercise the greatest range of control topics. With regard to the dynamic systems, we required devices or configurations that would demonstrate linear (or nearly linear) control, nonlinear control, control of stable and unstable systems, control of multi-input multi-output systems, and some really challenging problems for advanced students. With regard to the controlling mechanisms, we required ways to implement (or emulate) continuous-time and discrete-time (digital) control systems. Specific items required to fully explore digital control are:

- The capability of sampling analog data at a user-specified rate;
- The choice of using either fixed- or floating-point arithmetic;
- The ability to implement discrete-time computational structures.

Grant DUE-981009 from the National Science Foundation Directorate of Undergraduate Education has allowed us to accomplish these goals. A description follows.

II. Choice of Lab Devices

We decided to base our new lab primarily around the Magnetic Levitation (MagLev) Unit and Control-Moment Gyroscope (Gyro) Unit by *Educational Control Products (ECP)*.⁶ These two devices are shown in Figure 1. Together, they exhibit many important properties of dynamic systems from the point of view of control theory. A matrix of important attributes in dynamic devices, as well as the coverage by specific devices is listed in Table I.



Figure 1. The two lab devices. The MagLev device is to the left; the Gyro device is to the right.

TABLE I
Attractive attributes of the selected dynamical devices.

Desirable dynamic attribute	MagLev	Gyro
1. Linear single-variable, stable	Y	Y
2. Linear single-variable, unstable	Y	Y
3. Nonlinear single-variable, stable	Y	Y
4. Nonlinear single-variable, unstable	Y	Y
5. Linear multi-variable, little I/O interaction	Y	N
6. Nonlinear multi-variable, large I/O interaction	N	Y
7. Dynamically rich system	N	Y
8. Electromechanical system	Y	Y

The MagLev (described in more detail in Section III) may be used to exercise many skills. It can be configured as open-loop stable or unstable, so may be used to teach practical concepts of stability and stabilization. It may be configured as a single-variable system (controlling the position of a single disk) or as a multi-variable system (controlling two disks). Additionally, the plant is nonlinear, so techniques for small-signal and feedback linearization must be employed. In small operating ranges it is approximately linear, so standard linear control techniques work. Not to be underestimated, this device provides dramatic and interesting demonstrations. The actuators and sensors are clean, high-quality devices, and the entire system is ruggedly constructed. This

device is especially well-suited to demonstrate analysis and design techniques taught in classical analog and digital control courses, and to teach introductory modern analog and digital control.

The Gyro (described in more detail in Section IV) may also be used to exercise many advanced skills. Two of the four mobile axes are directly actuated, and all four axes have sensor feedback. In its most general configuration, it is a very nonlinear and dynamically rich system with large input-output interaction, and is used in more advanced controls courses and student projects.

III. Description of the Magnetic Levitation (MagLev) Device

Two views of the magnetic levitation system are depicted in Figure 2. Upper and lower electromagnetic drive coils produce a magnetic field in response to a dc current. One or two magnets travel along a glass guide rod. By energizing the lower coil, a single magnet is levitated by a repulsive magnetic force. As current in the coil increases, the field strength increases and the levitated magnet height is increased. For the upper coil, the levitating force is attractive.

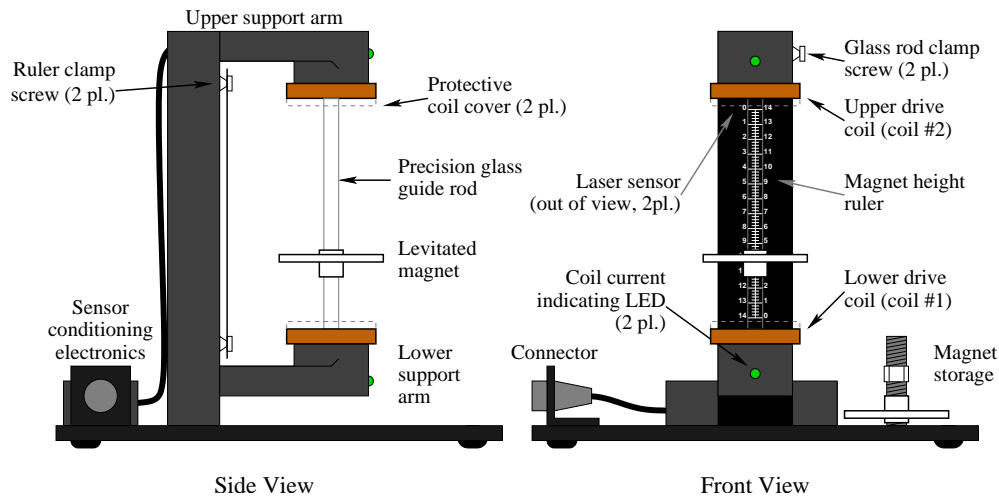


Figure 2. Two views of the MagLev device.

The magnets are of an ultra-high field strength rare earth (NeBFe) type. A dry-lubricated guide bushing at the center of the disk slides up and down the rod. A white reflective surface covers most of the disk. Two laser-based sensors make use of the reflective properties of the disk surface to measure the magnet positions. The laser beams are spread by an optical element into a fan shape and are projected onto the diffuse white surfaces of the magnets. Photodetectors view the beams and generate voltages proportional to the amount incident beam power. The lower sensor is typically used to measure a given magnet's position in proximity to the lower coil, and the upper one for proximity to the upper coil (both $\approx 8\text{ cm}$ range). Sensor-conditioning circuitry makes the design immune to stray light noise, such as turning room lights on and off, and rejects most induced electronic disturbances. Thus a relatively low noise signal is output from the amplifier box.

For many control scenarios, a general-purpose PC is used as the controller. An interface card in the PC contains D2A and A2D circuits connected to a "breakout box" which the student can

access. A power amplifier/sensor conditioner box drives the MagLev device. The student may connect the breakout-box signals directly to the amplifier box using “banana cables.” A pictorial description of the system setup is shown in Figure 5. The software running on the PC is discussed in Section VI.

IV. Description of the Control-Moment Gyroscope (Gyro) Device

The control-moment gyroscope may be configured in a variety of different ways. Torque is applied via direct-acting, reaction, or gyroscopic drive mechanizations. One or two such input torques may be applied simultaneously and the degrees of freedom may be constrained in such a way as to create plants ranging from simple one degree of freedom rigid bodies to complex systems with two torque inputs and four angular outputs. The system may be operated in regions where its salient behavior is linear, or in a more global workspace where the behavior is highly nonlinear. Thus this dynamically rich system provides a testbed for experiments ranging from demonstration of fundamental principles to advanced research.

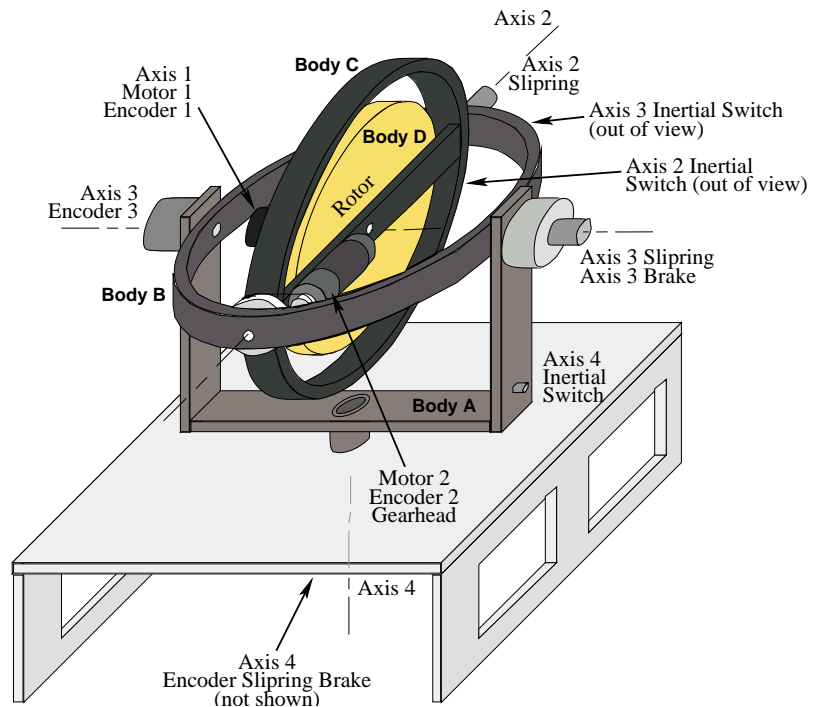


Figure 3. The control-moment gyroscope device.

The plant, shown in Figure 3, consists of a high inertia brass rotor suspended in an assembly with four angular degrees of freedom. The rotor spin torque is provided by a rare earth magnet type dc motor (motor 1) whose angular position is measured by an optical encoder (encoder 1). The first transverse gimbal assembly (body C) is driven by another rare earth motor (motor 2) to effect motion about axis 2. Another optical encoder provides feedback of the relative position of bodies C and B.

The subsequent gimbal assembly, body B, rotates with respect to body A about axis 3. There is no active torque applied about this axis. A brake, which is actuated via a toggle switch on the controller box, may be used to lock the relative position of A and B and hence reduce the system degrees of freedom. The relative angle between A and B is measured by encoder 3. Finally, body A rotates without actively applied torque relative to the base frame (inertial ground) along axis 4. The axis 4 brake is controlled similarly to the axis 3 brake and an optical encoder provides position feedback.

Inertial switches, or g-switches are installed on bodies A, B, and C to sense any overspeed

condition in the gimbal assemblies. For axis 2, limit switches and mechanical stops are provided at the safe limit of travel. When any of these normally closed switches sense a high angular rate condition, they open and thereby cause a relay to turn off power to the Controller box. When this power is lost, the fail-safe brakes (power-on-to-release type) at axes 3 and 4 engage. Also upon loss of power, the windings of motors 1 and 2 are shorted, thereby effecting electromechanical damping. Thus all axes are actively slowed and stopped whenever an over-speed or over-travel condition is detected.

Precious metal sliprings are included at each gimbal axis to provide for continuous angular motion. These low noise, low friction, sliprings pass all electrical signals including those of the motors, encoders, g-switches, limit switches, and brakes to the control box.

V. Hardware Interface

In order to interface the MagLev and Gyro units to the host computer, a data acquisition card for the PC is needed. The MagLev requires two standard D2A and four standard A2D channels, and the Gyro requires four optical encoder inputs in addition to two standard D2A outputs. We also wanted to find a board that would work with both Real-Time Windows Target (RTWT) by *MathWorks* and Real-Time Linux Target (RTLTL)⁷ by *QRTS*⁸ as we were initially unsure which operating-system we would choose to use on the host computer. Boards with both optical encoder inputs as well as A2D and D2A channels are rare. Finding a compatible set of hardware and software drivers was also a challenge.

One solution we considered was to use a general-purpose DSP board to perform data acquisition. The board would be outfitted with sufficient analog I/O, and could be programmed to decode encoder inputs using digital I/O. An advantage of the DSP approach is that high control-loop bandwidths are often possible due to the efficiency of the DSP. A disadvantage is expense.

The other solution we considered was to use a generic I/O card with encoder inputs. This is a less-expensive approach, but places the control-loop processing burden on the main system CPU and so tends to decrease the control bandwidth that can be achieved.

Our search led us to four I/O boards that could meet our purposes from a hardware point of view: The interface board *ECP* sells with their control devices, the *Humusoft*⁹ MF604, the *Quanser*¹⁰ MultiQ and the *ServoToGo*¹¹ Model 2. At the time we made our decision, the *ECP* board was only supported by *ECP* proprietary software (it is now supported under RTWT and RTLTL via software from *QRTS*), the *Humusoft* board was supported under RTWT only (and did not have enough I/O to support some lab devices not described in this paper), and both the *Quanser* and *ServoToGo* boards were supported under RTLTL only. Helping our decision, we received indication from *QRTS* that they had plans to support the *ServoToGo* board under RTWT (it now is). We chose to use the *ServoToGo* board because it allowed us flexibility with regard to operating system, and was the most cost-effective solution.

Among other features, the *ServoToGo* board supplies eight A2D channels, eight D2A channels and eight encoder input channels. This is more than sufficient for the needs of our laboratory.

VI. Software Interface

In order for students to access the I/O board to control the MagLev and Gyro, a software interface to the board is required. Rather than requiring that the students write C-language code and interrupt-service routines (much less, debug same) we chose to use a Matlab/ Simulink/ RTW/ RTLTL interface.

Matlab is a software environment that provides great computational power and professional graphical output. The Control-Systems Toolbox, in particular, greatly aids control-system analysis and design. Simulink is a block-diagram graphical-user-interface based simulation package that works within the Matlab environment and allows linear and nonlinear, continuous-time and discrete-time simulation. An example Simulink diagram is displayed in Figure 4.

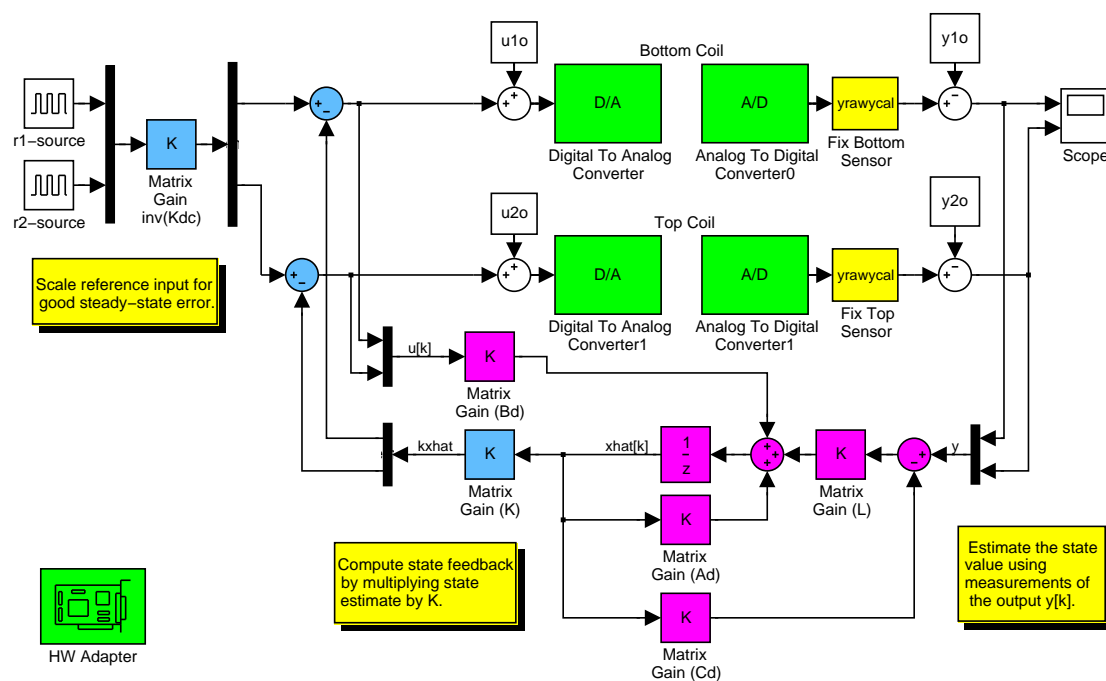


Figure 4. Example Simulink diagram to perform full MIMO control of MagLev.

Simulink does not directly access the *ServoToGo* interface board. Rather, the *Real-Time Workshop* (RTW)—accessible from Simulink through a pull-down menu item—writes generic C code to implement the Simulink block diagram. Then, either the *Real-Time Windows Target* (RTWT) or *Real-Time Linux Target* (RTLTL) is used to generate host-machine specific code and execute it. All of this happens with the click of a mouse; the student writes no code at all!

Using the host CPU to execute the control algorithm—as done with RTWT and RTLTL—results in economies since a separate DSP or CPU is not required. However, other performance issues arise. Specifically, traditional computer operating systems such as MS-DOS, Windows 95/98/NT/2000, Unix and Linux are not designed for real-time operation. Although RTWT is available, there is debate regarding whether or not Windows NT is yet a reliable platform for real-time systems.^{12, 13}

There is no guaranteed maximum latency between an interrupt and its service, which makes real-time applications unpredictable.

Other alternatives include QNX or VxWorks, commercial real-time operating systems. These have been used for educational purposes before; an example is the QMotor program developed for QNX.¹⁴ We decided not to use these systems due to the costs involved.

The alternative we chose is Real-Time Linux, or RTLinux. Linux¹⁵ is a free Unix-like operating system for i386 (and family), Alpha and Sparc processors. By itself, Linux is not suitable for real-time systems, but a free patch called RTLinux adds functionality to Linux to allow real-time code to execute.^{16,17} One paper has already described an RTLinux approach to control education, using Matlab and Simulink, but not using RTW and RTLTL.¹⁸ A disadvantage of this implementation is that we would need to write code to interface with the I/O boards; The Matlab/Simulink/ RTW/ RTLTL system does not require that we write any code at all.

One disadvantage of using RTLinux is that basic Unix system-administration skills are required by the lab administrator in order to set up and maintain the lab computers. This was not a problem for us as we have some experience in this area. Another potential disadvantage that concerned us is that our students tend to be more familiar with Windows than Linux. We wanted the lab to provide control-systems education and not operating-system education! So, we were careful to set up the lab computers to minimize the specific knowledge of Linux required (almost none), and although the students grumble from time to time, they do not seem to have been deterred.

Figure 5 shows a pictorial representation of the entire control system, including software and hardware components. The student works within a Matlab environment where he or she enters a block diagram into Simulink. When the diagram is complete, the student selects “RTW Build” from the Tools menu, and RTW generates C code and builds it with help from RTLTL. The student can then start the code running using “Start” from the simulation menu. The code runs as a kernel module, accessing the MagLev through the *ServoToGo* interface card. Signals from the card are routed through a cable to a breakout box. The student wires the breakout box to the appropriate power amplifier using “banana cables.” This setup has proven to be very usable.

The final lab setup comprises the following hardware for each workstation: One *ECP* MagLev “plant-only” unit, one *ECP* Gyro “plant-only” unit, one Pentium-III class computer with a *ServoToGo* Model 2 I/O board, one *Comdyna* GP-6 analog computer, one *Protek* 3003B dc power supply, one *Protek* B-803 sweep function generator and one each *Agilent* HP-3468B multimeter HP-54602B digital oscilloscope (with HP-54657B unit for phase measurement). The test-and-measurement equipment are used for system identification and control-system debugging.

The following software is used on each lab computer: Red-Hat Linux 6.1 (free download available from <http://www.redhat.com>), RT-Linux 2.0 (free download available from <http://www.rtlinux.org>); the following *MathWorks* products: Matlab 5.3, Release 11, for Linux, Simulink 3.1, Release 11, for Linux, Real-Time Workshop 3.0, Release 11, for Linux; and the *QRTS* product RTLTL Version 1.1. The C development packages should be selected when

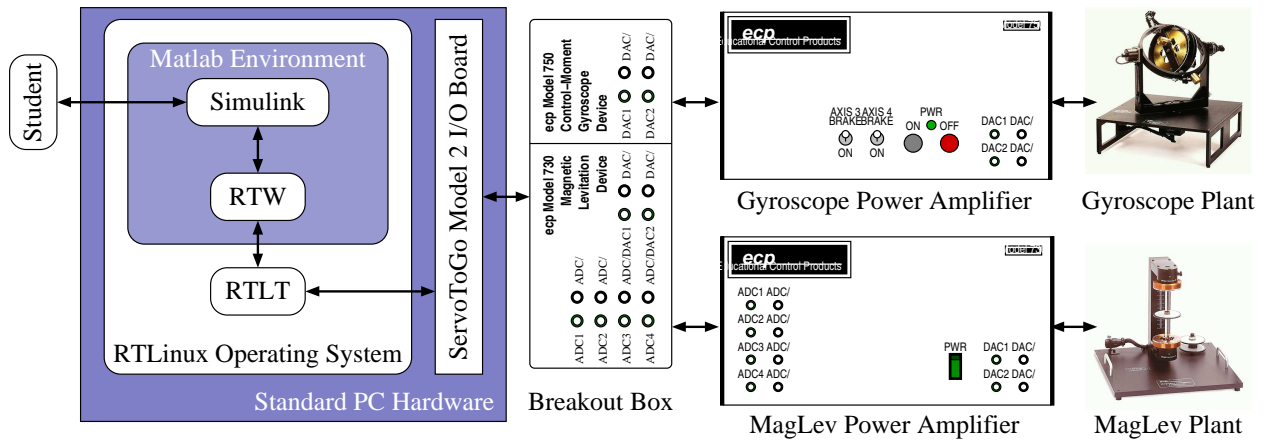


Figure 5. Pictorial representation of system setup including hardware and software integration.

installing Red Hat Linux Version 6.1.¹

VII. Two directions

We started the laboratory revival in the summer of 2000. Since then, the lab has been used for the laboratory courses: *ECE4530: Control Systems Laboratory*,¹⁹ and *ECE4560: Digital Control Laboratory*,²⁰ the latter being a focus of this paper. It has also been used for course projects in *ECE4540: Digital Control Systems*, *ECE5520: Multivariable Control Systems I*, and *MAE5421: Digital Flight Control*, and for Senior Design Projects and Independent Study Topics. A previous paper has discussed the implementation of *ECE4530*,¹ and this paper discusses *ECE4560* and *MAE5421*.

VII-A. The Undergraduate Digital Control Course and Laboratory

The undergraduate senior elective *ECE4540: Digital Control Systems* is a standard lecture-based course covering digital control systems from both frequency-domain and state-space points-of-view. A textbook by Franklin and colleagues is used.²¹ A standard feedback control course is prerequisite. The *ECE4560: Digital Control Laboratory* course meets in the laboratory described in this paper, and has the lecture course as a co-requisite. The two courses are designed to coordinate with each other as much as possible so that the experimentation compliments and illuminates the theory. A Gantt chart showing the relative phasing of the two different courses is shown in Figure 6.

The laboratory course is organized into an introductory lab and four main units of labs. The introductory lab introduces the students to the laboratory facility, instructs them on how to log on to and use the lab computers, and prescribes proper care of the equipment. The students are given a simple assignment in Matlab/ Simulink/ RTW/ RTL. In their junior year they are exposed to Matlab in their curriculum, but Simulink may be unfamiliar to them, and unless they have also

¹ It is our understanding that QRTS no longer supports RTL. If this proves to be a problem, we will migrate our lab to RTWT.

Lecture Topic

1. Introduction to digital control.
2. Review of continuous-time control.
3. Emulation of analog controllers (I).
4. The z-transform.
5. Emulation of analog controllers (II).
6. Sampling and reconstruction.
7. Discrete-time systems.
8. Stability analysis techniques.
9. Digital controller design (transfer fns).
10. State space--continuous time.
11. State space--discrete time.
12. Discrete estimator design.
13. Review.

Laboratory Topic

Laboratory Orientation

1. Discrete-time simulation with Simulink.

Unit 1: Design by Emulation

2. Time-domain controller emulation.
3. Frequency-domain controller emulation.

Unit 2: Digital Effects

4. Sampling, aliasing, zero-order hold.
5. Discrete-time plant modeling.
6. Filter structure & finite-precision effects.

Unit 3: Transfer-Function Controller Design

7. Frequency-response control design.
8. Numeric optimal PID control design.
9. Ragazzini's direct control design method.

Unit 4: State-Space Controller Design

10. State-space model verification (disc.-time).
11. State-feedback control design.

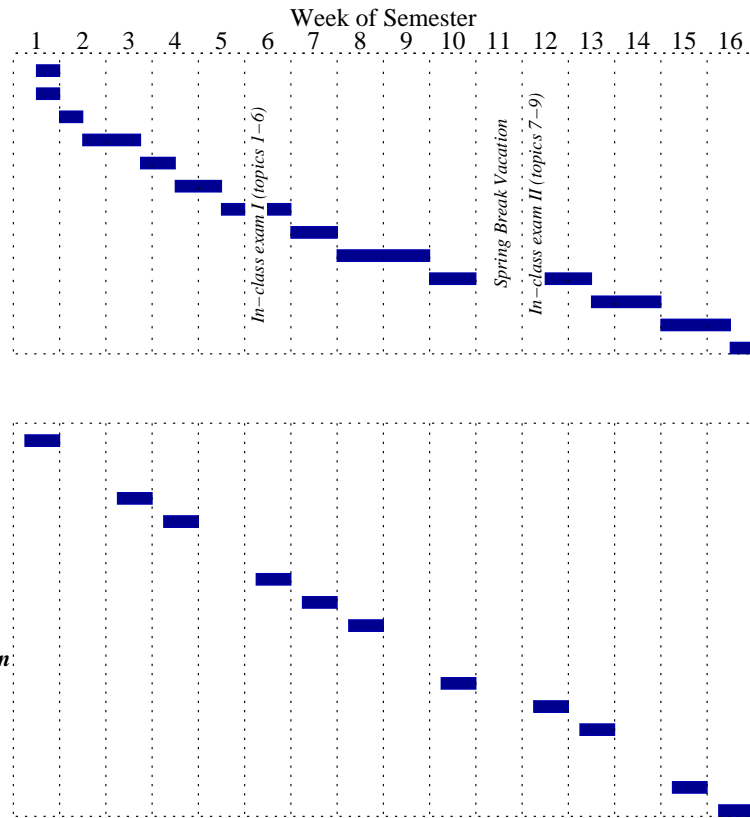


Figure 6. Gantt chart illustrating synchronization between lecture topics and lab topics. Lectures are on Tuesday/Thursday, and labs are on Thursday.

taken the *ECE4530: Control Systems Laboratory* course, they will not have seen RTW or RTLTL before.

The first unit following the introductory lab has the students emulate continuous-time controllers in a discrete-time implementation. (The lecture course has already covered, by the appropriate time, a review of continuous-time control, the z -transform, and emulation of continuous-time controllers.) In the first lab, they emulate a continuous-time controller by approximating an ordinary differential equation (controller) with a difference equation. In the second lab, they perform lexical substitution of 's' with a function of 'z' to discretize the controller.

The second unit has the student explore various digital effects. (The lecture course has already covered, by the appropriate time, sampling and discrete-time system analysis.) In the first lab they investigate the effect of sampling rate. In the second lab, they make a discrete-time model of the plant. In the third lab, they explore finite-precision arithmetic and filter coefficient effects.

The third unit covers controller design via transfer-function methods. (The lecture course has already covered, by the appropriate time, various different transfer-function control design methods.) In the first lab, the students use standard Bode frequency-response methods. In the second lab, the students perform a numerical search for controller parameters via gradient descent

in order to optimize a cost function. In the third lab, students use Ragazzini's direct control design method.²¹ All three labs are fairly complicated SIMO control configurations.

The fourth unit introduces the student to state-space control design methods. (The lecture course has already introduced the students to the appropriate concepts). In the first lab, the students are asked to verify a state-space model of the plant, configured as a two-input two-output system. They also perform state-feedback control via direct state measurement. In the second and final lab, the students design and implement a multivariable controller using a full-order estimator. This lab provides a climax to the course.

A lab reader has been prepared for this class, and is available on the Internet.²⁰ If you would like to use lab(s) from this reader, please contact the first author at: glp@eas.uccs.edu.

VII-B. The Graduate Digital Control Course

The graduate level course, *MAE5421: Digital Flight Control*, is a single course rather than a theory course coupled with a separate lab course, as described above. This graduate course is designed primarily for students majoring in mechanical or aerospace engineering, although by not requiring vehicle dynamics as a prerequisite (though desirable) we have not intentionally discouraged electrical engineering majors.

The prerequisites are essentially the same as for the undergraduate ECE course(s) just described—an undergraduate first course in control theory. Much of the theory covered in this graduate course is very similar to that in the undergraduate ECE theory course. However, by offering this course at the graduate level, we expect to move through the material (theory and lab) much more quickly, and ultimately consider a flight-control case study at the end of the course. For example, rather than lecture on the development of a discretized model of a continuous system, the graduate students are asked to develop such a model as a home problem.

The two fundamental differences between the undergraduate ECE courses and the graduate MAE course, in terms of theory stressed, involve treatment of loop-shaping concepts and w -domain analysis and design. This eliminates serious treatment of estimator design, and couches the design problem exclusively in the frequency domain (though both state-space and transfer-function models of systems are used). Additionally, quantization effects and describing functions are also specifically treated in the graduate course.

The labs involve use of both hardware devices, the magnetic levitation device and the gyroscopic device. The MagLev device is used in a manner very similar to that in the undergraduate ECE lab, allowing for the investigation of digital effects like time delay, and quantization. Then a simple continuous controller for one disk is designed and implemented on the MagLev unit. This is followed by the design of a digital controller for the same device using discrete equivalents, direct z -domain, and direct w -domain techniques.

A final design project then involves the gyroscopic device (which exhibits input-output cross coupling), for which a multi-loop controller is to be recommended by the student. The students are required to develop several possible digital designs, evaluate them, and recommend a

controller and sampling interval. Maximizing the sampling interval, while maintaining acceptable performance and robustness is a design objective. There is no requirement that their design be obtained via either discrete equivalence or by direct design. Baseline performance for the students is based on a “good” continuous design developed by the students that meets the design specs provided. During the final weeks of the semester, the flight-control case studies are the subjects of the lectures, and simultaneously the students are working on their final design project.

VIII. Initial Evaluation and Student Feedback

Although the lab has been operational for only two semesters, we have begun evaluating its effectiveness using quantitative and qualitative means. We wanted to test two hypotheses:

Hypothesis: A lab experience provides hands-on learning to improve basic understanding.

Therefore, students taking the lecture course only (the lab is not required) will do more poorly in the lecture course than those students taking the lab as well.

Results: We have data for four semesters for students taking the lecture course only versus students taking both the lecture and lab. The lab syllabus for the first two semesters was based on simulation only; the lab for the final two semesters is the one based on experiment and presented in this paper.

Over the three semesters, grades in the lecture course for students taking both the lab and the lecture averaged to 80%; grades in the lecture course for students not taking the lab averaged to 69%. There is an 11% gap between these two groups of students. There are enough students in the sample to make this result statistically meaningful.

Hypothesis: An experimental lab provides better learning than a lab based on simulation.

Results: For the first two semesters, the gap between lecture-only students and students who took both the lecture and lab was a 5% difference in course grade in the lecture course (both times). For the first semester where students had an experimental lab experience, the gap was 16% (all students elected to take the lab in the fourth semester, so results are not available for comparison). This result is interesting but probably not statistically meaningful since only two students elected *not* to take the lab course in the third semester.

Overall, the quantitative results are in favor of a lab experience whether or not it is experimental. The results seem to be in greater favor of an experimental approach versus a simulation approach, but the sample size of statistics is too small to tell for sure as of yet.

Qualitative evaluation has been done by recording some responses from student lab writeups from the *Control Systems Laboratory* and *Digital Control Laboratory* courses. As you will be able to tell, they are quite candid:

[On analog computer lab] “This lab was boring because we did not get to play with the MagLev.”

[On analog computer lab] “This was a painful experience, I hate the analog computer with a passion, and it hates me.”²

² The analog computer lab that the students did not like has been modified in the lab reader now available. It now includes experimentation with the MagLev, and now has a clearer presentation on how to use the analog computer to perform feedback control.

[On system identification] “These methods are pretty cool in that I never realized that they took into account friction, which makes them more useful.”

[On linearization, comparing linearized model with real system] “Just for fun, we inputted some other waves: square wave, triangle wave, ramp function and so on. The coolest one was when we put in a random signal, and raised the value for u_{mg} [the dc-offset]. The magnet would jump around like a crazy grasshopper, and the model output matched the actual output almost exactly.”

[On frequency-response/ Nyquist] “This lab clears up a lot of concepts learned in class. Like the Nyquist plot, I didn’t realize that the Nyquist plot was just a polar plot until this lab (although I am sure that Dr. Plett probably mentioned it a million times).”

[On the course in general] “To be quite honest, I’m learning more from this lab than previous labs taken and labs that I’m taking right now. I am also impressed how much it actually follows ECE4510 Feedback Control Systems and uses the things that we learned in class.”

Some of these comments indicate that real learning has occurred. One shows that the students were so involved in the lab that they tried some experiments which were not required. The final comment validates that the lecture and lab courses are well coordinated.

IX. Conclusion and Future Plans

A control-systems laboratory has been developed around the *ECP* MagLev and Gyro plants, using Matlab/ Simulink/ RTW and RTLTL software, on the RTLinux operating system. As of the writing of this paper, the lab has been in use for several semesters. A lab manual has been developed for the undergraduate *Digital Control Laboratory* that uses the MagLev device. The lab schedule coordinates with the *Digital Control Systems* senior-elective class so that experiments complement and illuminate the theory. The MagLev has also been used for the final project in other undergraduate and graduate classes. Initial evaluation indicates that the laboratory experience has significantly aided learning of control-systems concepts.

An MAE graduate level course has also been developed in digital control systems. This course combines theory and experiment with both the MagLev and Gyro plants. Feedback from students in this course is also very positive, with respect to using hardware instead of lecture only.

Future plans include true integration of the MAE and ECE undergraduate feedback control laboratory classes, resulting in a full multidisciplinary experience. This will commence in Spring 2002. We believe that the intermingled perspectives of two disciplines will lead to better-rounded learning.

Acknowledgment

This laboratory was made possible and this work was supported in part by the National Science Foundation under grant DUE-981009. Also special thanks to Ali Pak from *ECP*, and both Nick Costescu and Darren Dawson from *QRTS* for helping to design an affordable system; for email

and telephone support from both. Thanks to both *ECP* and *QRTS* for allowing use of text and diagrams from their manuals in the *Control Systems Laboratory* lab reader.

Bibliography

1. G. Plett and D. Schmidt. Multidisciplinary lab-based controls curriculum. In *CD-ROM Proc. 2001 ASEE Annual Conference & Exposition*, (Albuquerque, NM: June 2001).
2. Comdyna, Inc., 305 Devonshire Road, Barrington, IL 60010, USA. Tel/Fax: (847) 381-7560, <http://www.comdyna.com>.
3. The MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760 USA. Tel: (508) 647-7000, Fax: (508) 647-7001, <http://www.mathworks.com>.
4. D.S. Bernstein. Control experiments and what I learned from them: A personal journey. *IEEE Control Systems Magazine*, 18(2):81-88, April 1998.
5. L. Schmitt. U of I's control systems lab one of premier labs in country. *University of Illinois at Urbana-Champaign "Ingenuity"*, 4(1):4, March 1999.
6. Educational Control Products, 1 Bucksin Court, Bell Canyon, CA 91307, USA. Tel: (800) 486-0840, Fax: (818) 703-6040, <http://www.ecpsystems.com>.
7. Z. Yao, N.P. Costescu, S.P. Nagarkatti, and D.M. Dawson. Real-Time Linux Target: A Matlab-based graphical control environment. In *Proceedings of the IEEE International Conference of Control Applications*, (Anchorage, AK: September 2000).
8. Quality Real-Time Systems, 6312 Seven Corners Center #134, Falls Church, VA 22044 USA. Tel: (703) 533-9142, Fax: (703) 533-9143, <http://www.qrts.com>.
9. HUMUSOFT s.r.o., Novákových 6, 180 00 Praha 8, Czech Republic. Tel: ++420-2-66315767, Fax: ++420-2-6844174, <http://www.humusoft.com>.
10. Quanser Consulting, 102 George Street, Hamilton, ON L8P 1E2, Canada. Tel: (905) 527-5208, Fax: (905) 570-1906, <http://www.quanser.com>.
11. Servo To Go, Inc, 1716 221st Place Northeast, Redmond, WA 98053, USA. Tel: (425) 868-4636, Fax: (425) 868-5323, <http://www.servotogo.com>.
12. M. Timmerman and J.-C. Monfret. Windows NT as a real-time OS? *Real-Time Magazine*, 2Q:6-13, 1997.
13. W.E. Dixon, D.M. Dawson, B.T. Costic, and M.S. de Queiroz. Towards the standardization of a MATLAB-based control systems laboratory experience for undergraduate students. In *Proceedings of the American Control Conference*, (Arlington, VA: June 2001). (Submitted. Available at <http://64.0.106.39/downloads/publications.shtml>).
14. N. Costescu, D. Dawson, and M. Löffler. QMotor 2.0—A real-time PC based control environment. *IEEE Control Systems Magazine*, 19(3):68-76, June 1999.
15. The official Linux Web site. <http://www.linux.org>.
16. The official RTLinux Web site. <http://www.rtlinux.org>.
17. V. Yodaiken and M. Barabanov. Real-Time Linux. *Linux Journal*, February 1997.
18. Y.-C. Chen and Jason M. Naughton. An undergraduate laboratory platform for control system design, simulation, and implementation. *IEEE Control Systems Magazine*, 20(3):12-20, June 2000.
19. G.L. Plett. *ECE4530: Control-Systems Laboratory* lab reader, August 2001. Available at <http://mocha-java.uccs.edu/CSL/>.
20. G.L. Plett. *ECE4560: Digital Control Laboratory* lab reader, August 2001. Available at <http://mocha-java.uccs.edu/CSL/>.
21. G.F. Franklin, J.D. Powell, and M.L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, Reading, MA, third edition, 1998.

GREGORY L. PLETT

Gregory Plett was born in Ottawa Canada in 1968. He received the B.Eng. degree (with high distinction) in Computer Systems Engineering from Carleton University in 1990. He received the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 1992 and 1998. Since 1998, he has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Colorado at Colorado Springs. He can be reached by email at glp@eas.uccs.edu.

DAVID K. SCHMIDT

David Schmidt was born in Lafayette Indiana in 1943. He received his B.S. (with honors) and Ph.D. in Aerospace Engineering from Purdue University, and the M.S. in Aerospace Engineering from the University of So. California. He has served on the technical staffs of McDonnell Douglas Missiles & Space Corp, and the Stanford Research Institute. He has also served on the engineering faculties of Purdue University, Arizona State University, the University of Maryland at College Park, and now is Professor of Engineering at the University of Colorado at Colorado Springs. He can be reached at dschmidt@eas.uccs.edu.