



## **Work-in-Progress: Using Hardware-based Programming Experiences to Enhance Student Learning in a Senior Feedback Controls Lecture Course**

**Prof. Johné' M Parker, University of Kentucky**

Johné M. Parker is an associate professor of Mechanical Engineering at the University of Kentucky.

**Dr. Stephen L. Canfield, Tennessee Technological University**

**Dr. Sheikh Khaled Ghafoor, Tennessee Technological University**

Sheikh Ghafoor is an associate professor in the Department of Computer Science at Tennessee Technological University. He received his M.S. and Ph.D. in Computer Science from Mississippi State University. His primary research includes parallel, distributed computing, and high performance computing. His current research is in autonomic resource management for high performance computing environment, programming model for parallel adaptive applications, and fault tolerant computing. Dr. Ghafoor is also very interested, and actively engaged in research in the area of computer science and engineering education. Dr. Ghafoor has been principal investigators and investigator on grants from NSF and DOE.

**Kassy Moy Lum**

# Work-in-Progress: Using Hardware-based Programming Experiences to Enhance Student Learning in a Senior Feedback Controls Lecture Course

## Abstract

This paper describes proposed enhancements in the re-design of a senior-level Mechanical Engineering Feedback Controls course. Though several changes in course design have been made over the sequence of three successive offerings of the course, the primary focus of this paper is improving students' ability to apply programming and computational problem-solving skills to understand and solve Controls problems. This investigation builds upon a model to use hardware to integrate programming experiences throughout the curriculum; in the model, the three learning principles deemed critical for success are student engagement, knowledge transfer and self-directed learning. In traditional Controls lecture courses, core course concepts are generally considered to be a bit abstract to a considerable percentage of students and there are often disconnects between theoretical course concepts, computational solution techniques and the behavior of real-world systems. Each of these challenges inhibit the three principles deemed critical for success in learning and it is posited in this paper that the introduction of programming involving hardware in the Controls course will enhance the three principles, resolve disconnects and improve overall student learning.

The proposed re-design introduces programming experiences performed on micro-controller hardware to illustrate key topics and solution methods (e.g., system modeling and controller design) into a traditional Feedback Controls lecture course. The authors have previously taught the course numerous times, so a well-paced course schedule and solid foundation of course notes are already in place. Additionally, hybrid and problem-based learning (PBL) techniques have been incorporated into prior offerings, which enhances student engagement and allows both sufficient time to introduce programming modules and the ability of the instructor/research assistant (RA)/teaching assistant (TA) team to give necessary assistance and feedback during the programming experiences.

A hands-on programming toolkit developed by Canfield and Abdelrahman<sup>1</sup> for direct programming of micro-controller units (MCUs) uses MatLab as the programming environment. Using this toolkit, MCUs have been used to teach initial programming skills to engineering students in a context that matches their notions of engineering. In the Feedback Controls course, the model will be used to enhance programming skills in a context that enhances senior students' understanding of a somewhat abstract area of Mechanical Engineering. The overall goal is for students to be able to apply and observe the implementation of control algorithms on real-world hardware, without being hampered by significant obstacles or requirements for implementation. The MatLab-to-MCU toolbox effectively addresses this challenge, allowing students to implement their control algorithms using MatLab, a language they have previously used to *simulate* system response, "directly" on micro-controllers with little additional overhead requirements.

In particular, a series of labs are proposed for implementation using the MatLab-to-MCU toolbox via a Motorola processor implemented on a Dragon12 evaluation board. Proposed activities and assessment instruments are described and compared to the traditional (lecture-based) and hybrid-learning/problem-based formats. The paper concludes with a description of the current status, proposed assessment instruments to analyze the efficacy of the model in upper-level coursework and proposed next steps.

## Motivation and Related Work

Systems and Control Courses in Mechanical Engineering expose students to core course concepts in which the relationships between the mathematical underpinnings, practical design procedures and subsequent implementation are considered abstract for a considerable percentage of students; thus, there are often considerable disconnects between theoretical course concepts, computational solution techniques and usage in real-world systems.

Related discipline-based education research (DBER)<sup>2</sup> has identified three key insights regarding STEM education challenges that might address these disconnects:

- student-centered learning strategies (including team-based learning) can enhance learning more than traditional lectures,
- students often have incorrect understandings about fundamental concepts, and
- students are challenged by important aspects of the domain that can seem obvious to experts

Similarly, programming is a valuable engineering tool where students experience significant disconnects between constructs and effective application in an engineering context in latter courses. In [3, 4], the three principles of learning deemed critical for successful programming in engineering contexts are

1. Student Engagement: Engaging students' current knowledge to construct new knowledge
2. Knowledge Transfer: Students ability to transfer early programming skills to new contexts, applications and environments
3. Self-directed learning: Students assuming control of their learning in programming to adapt to the rapidly evolving demands of computational techniques in engineering

Therefore, by implementing controls algorithms analytically designed in companion assignments, Systems and Controls courses are ideal to investigate effective ways to address all three areas of the DBER study and simultaneously augment students' ability to use programming as a tool. The primary objectives for this paper are two-fold: i) increasing students' competency and understanding of fundamental Controls course concepts and ii) simultaneously demonstrating to students that implementing control algorithms is a specific application of programming concepts learned earlier in the curriculum.

Recent education research<sup>2, 5, 6</sup> indicates that undergraduate education is undergoing a revolution fueled by both student preferences and DBER. Much engineering research addresses student preferences by focusing on teaching and learning styles in STEM education and/or on the best practices in and benefits of team-based learning<sup>7-11</sup>. In the National Academies report<sup>2</sup>, scaffolding was suggested as a strategy to help students' difficulties with important aspects of the

domain that seem easy or obvious to “experts” like the instructor. Key scaffolding strategies include breaking a large task into smaller parts, working in peer groups and prompting<sup>12-15</sup>; as students gain confidence and competency in a topic, the instructor removes the scaffolding. The efficacy of scaffolding combined with team-based learning principles has also been investigated<sup>9-11, 16, 17</sup>.

Employing pedagogically-based improvements to the engineering programming experience throughout the undergraduate program has also received significant attention in the literature. Significant focus has been on modifying the approach to teaching engineering computing to first-year students<sup>18-21</sup>. Also suggested is initially employing the use of computing tools (such as spreadsheets, Matlab or MathCAD<sup>18, 22, 23</sup>) as an alternative to high-level programming languages (C or Fortran).

The model proposed by Canfield and Abdelrahman<sup>1</sup> for improving the programming experience for engineering students is designed around a principle of learning<sup>3</sup> and emphasized within a context related to STEM education<sup>4</sup>. In cited works, judicious repetition of programming applications from the initial course throughout the curriculum is suggested to move students’ engagement and programming competency toward advanced, open-ended problems in higher level courses.

In the model<sup>1</sup>, exposure to programming is also suggested throughout the curriculum; in this paper, we propose an approach that focuses on programming experiences that are in a context appropriate for upper-level engineering students by incorporating programming curricular linkages in a senior-level Controls course.

Since engagement is a critical principle for successful programming in engineering, several measures for engagement and attitude will be employed in evaluating the effectiveness of the model in this study. Measuring of attitude toward different engineering disciplines is described<sup>24-27</sup>. The literature eludes that, in general, attitude is measured by assessing students’ perception; thus, student responses will be incorporated into assessment and be measured using some form of Likert<sup>27</sup> scale. Additionally, student understanding and competency will be measured using pre- and post-tests on key topics.

## **Hardware Setup and Proposed Activities**

The activities found in the literature were primarily for first-year engineering students and an initial exposure to programming. The focus of this study is to encourage students’ use of programming constructs as a tool throughout the curriculum; specifically in this case, a senior-level Feedback Controls course is chosen as the test-bed. The authors build upon their prior work<sup>28, 29</sup> to design activities which provide a context-appropriate demonstration of engineering practice for Controls-related topics. Given that the development toolbox allows students to write programs as MatLab m-files, which are cross-compiled and loaded using a single MatLab command at the prompt, this is an environment with which students are already familiar; e.g., MCU-specific functions are contained in a MCU toolbox that is similar to the Controls and other MatLab toolboxes students have used in previous courses and will use in the initial analytical and numerical activities proposed in this study.

## Course Hardware:

The MCU board proposed for this study is a Dragon 12 Plus 2<sup>30</sup> (Figure 1) running CodeWarrior<sup>31</sup> with a 16 bit, 24 MHz CPU, 256K Flash EEPROM, 12K RAM, serial communication, 10 bit A/D, timer channels, pulse width modulation (PWM), and discrete and interrupt I/O. Input devices include eight dip switches, 4 momentary switches, [16 key] keypad, IR proximity sensor and photoresistor. Output devices include 2 16-digit LCDs, single-row LEDs, 4-7 segment LEDs and a Piezo speaker. The abundance of I/O devices provided by the Dragon board allows this hardware tool to be effective in supporting programming-related activities across a broad range of the curriculum, including senior-level coursework, such as Controls.

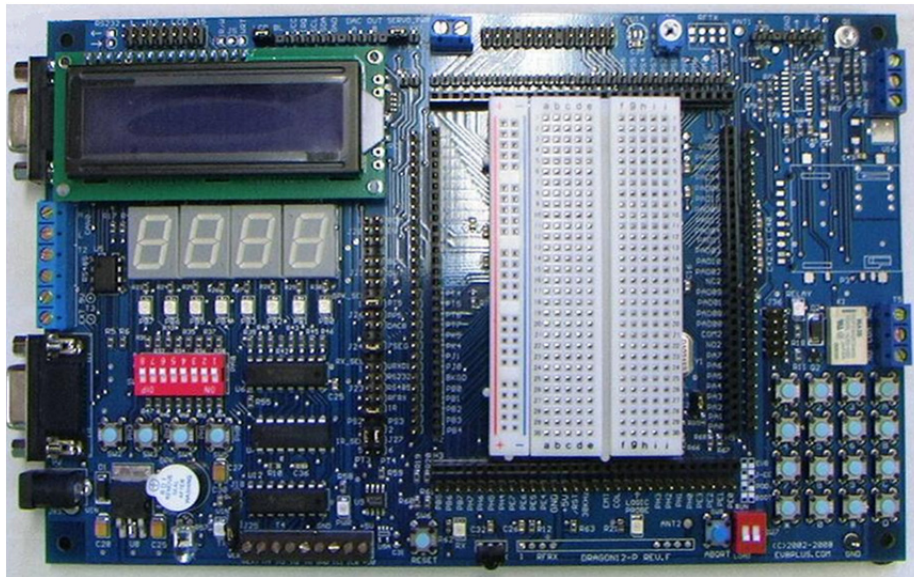


Figure 1. Dragon 12 Plus MCU board<sup>30</sup>

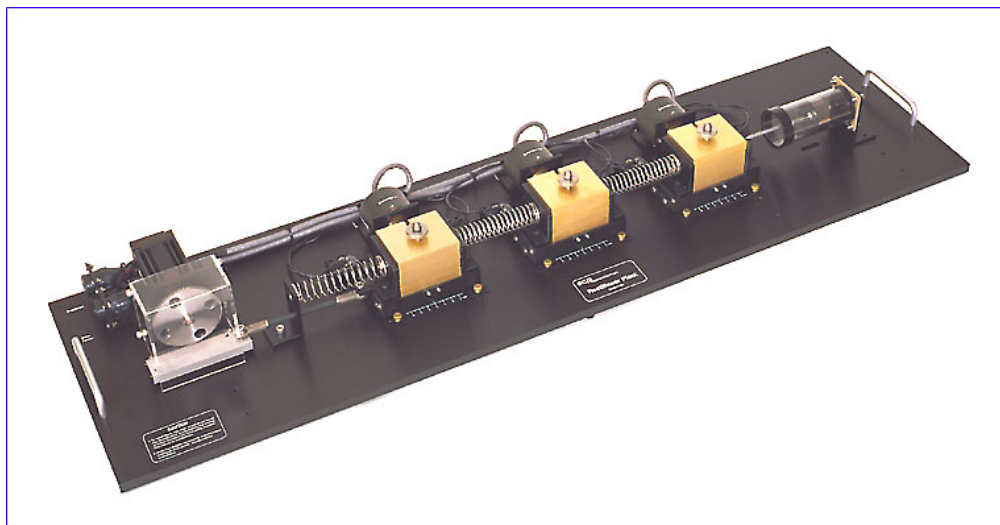


Figure 2. ECP<sup>®</sup> Rectilinear System<sup>32</sup>

The Educational Controls Products (ECP<sup>®</sup>) Rectilinear System (Figure 2) is the proposed hardware analogy for initial mass-spring-damper system activities proposed in Table 1; the rectilinear system is highly configurable and able to realize numerous schematics with great fidelity: sixteen unique configurations are possible for a three degree-of-freedom single-input-single-output system. Free or constrained configurations at either end and multiple possible adjustments in the mass, damping and spring constant values yield adjustable system poles and zeros in the 1.5 to 7 Hz range. High resolution encoders (1600 counts/cm) provide feedback and actuation is provided by 8N rack-and-pinion Fe-Co brushless motors. ECP<sup>®</sup> also provides a torsional (mass MOI-torsional spring-damper) system with similar flexibility that will be evaluated for use in future activities.

**Table I. Proposed Activities for Initial Assessment**

Chapter	Topic	Proposed Activity
2	Laplace and Modeling Review	Activity 1: Analytical Determination of Model and Investigation of Solution
4	Time Response	Activity 2: Analytical Investigation of System Transient and Steady-state Responses; Experimental Determination of Transfer Function and Validation of Model using Rectilinear Plant, Matlab and MCU
8	Root Locus	Activity 3: Root Locus; Analytical Investigation validated by Experimental Implementation for Proportional Control
9	Controller Design Using Root Locus	Activity 3: Analytical PID (or lead/lag) Controller Design validated by Experimental Implementation using Rectilinear Plant, Matlab and MCU
10	Frequency Response	Future Activity: Experimental Determination of Transfer Function using Frequency Response Methods; Validation of Model using Rectilinear Plant
11	Controller Design Using Frequency Response Methods	Future Activity: Analytical Lead/Lag Controller Design; Validation using Rectilinear Plant, Matlab and MCU

### Description of Proposed Activities and Supporting Resources

Online resources, in addition to the hybrid learning structure of this proposed (and previous) offering of the Design of Feedback Controls course, were designed to simultaneously address the concerns detailed in the National Academies report<sup>2</sup> and allow sufficient time to effectively implement activities proposed in Table 1. Activities 1 and 2 are also appropriate for inclusion in the pre-requisite junior-level Modeling of Dynamic Systems course and may be introduced into the earlier course upon initial feedback and assessment from the Controls course investigation.

As noted, a well-paced Controls course schedule and solid foundation of course notes were already in place<sup>29</sup>. Additionally, hybrid and problem-based learning (PBL) techniques incorporated into prior offerings enhanced student engagement and allowed both the sufficient time to introduce programming modules and the ability of the instructor/research assistant (RA)/teaching assistant (TA) team to give necessary assistance and feedback during the programming experiences.

Evaluation instruments in previous course offerings included weekly homework, bi-weekly short quizzes, two exams and the final exam. For this investigation, the course schedule, notes, basic learning management system (LMS) and evaluation instruments (homework, quizzes and exams) are proposed to remain the same; available online resources are described in the following section.

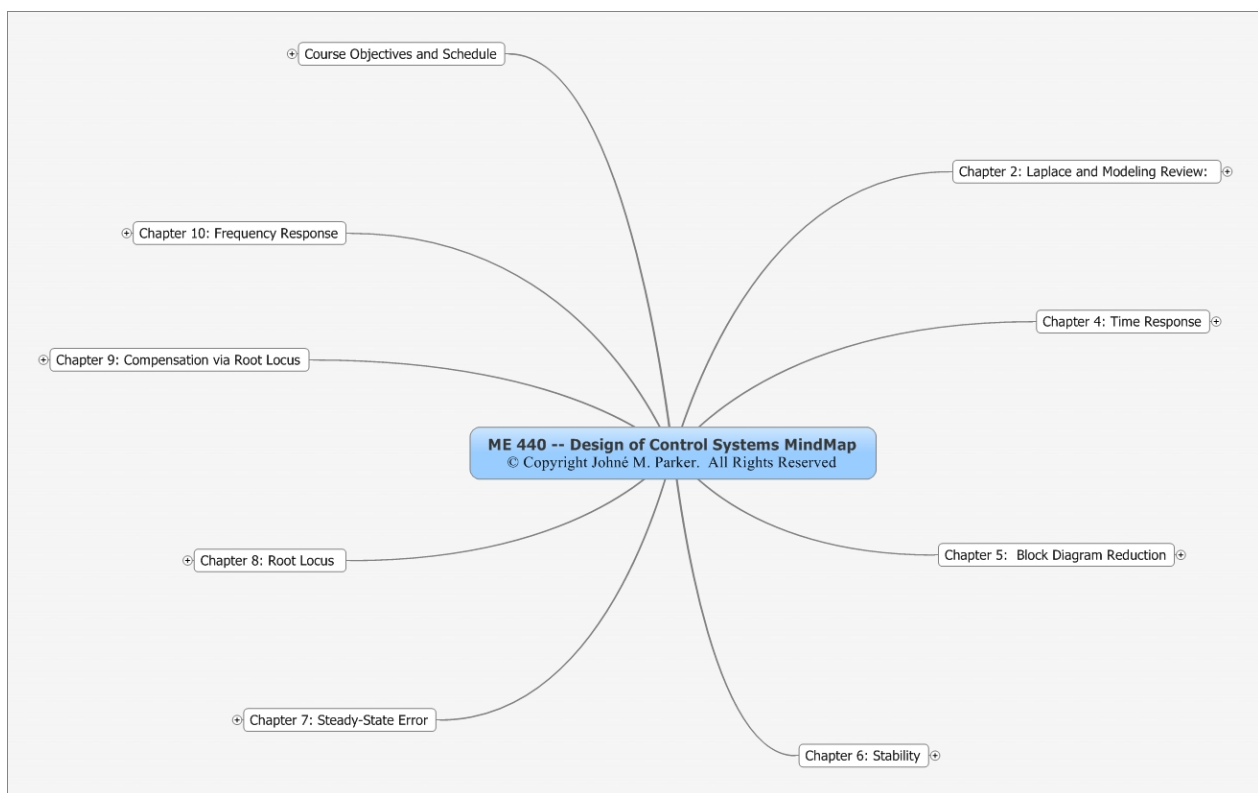


Figure 3. High-level Course Mind-Map<sup>29</sup>

## Online Resources

To support the in-class lectures, group discussions and the proposed hands-on activities, the online LMS, Blackboard, contains an interactive course mind-map (Figure 3), course notes (available for review prior to discussion in class), example solved problems, “vodcasts,” homework and quiz solutions. Vodcasts (short Echo360 modules) discuss a topic or detailed example problem using both audio (the instructor or TA's voice) and video (screen recordings of PowerPoint slides or PDF files of annotated solved problems or course notes); vodcasts are provided for preview prior to a class lecture or posted following a lecture to correct common misconceptions or illustrate important aspects of a topic that were not obvious to or understood

by students (i.e., addressing the third concern in the NAS report<sup>2</sup>). Additionally, Piazza, a novel on-line discussion forum<sup>33, 34</sup>, is used to augment office hours, recitation sessions, in-class Q&A, and to support team-based learning, including the activities proposed in this section; Piazza is freely-available, FERPA compliant, and allows students and instructors to easily and securely post or comment on notes, questions, or polls. PollEverywhere<sup>35</sup> is used to gather data on student understanding and student perceptions in real time. PollEverywhere polls can be embedded into PowerPoint slides or accessed online and answered using cell phones, tablets and laptops (i.e., are more flexible than systems depending upon more proprietary automated response system (or “clicker”) technologies).

Typical screenshots of the interactive mind-map (Figures 4 and 5) illustrate that the map parallels each chapter in the chosen text<sup>36</sup>. In addition to providing an overview of course objectives, key goals, governing equations and visual relationships are provided for each chapter; additionally, maps show linkages between latter complex course context and earlier concepts. A detailed mind-map for the entire course is available at the beginning of the semester and modified, as necessary, in near real-time, based upon students' questions and conceptual understanding of core concepts.

The Blackboard course shell also contains a section on MatLab resources. That section contains a link to common MatLab commands, MatLab syntax, locations of student computer labs containing MatLab on campus, basic overview MatLab notes (similar to information covered in the pre-requisite programming course) and a reference MatLab mind-map (similar to the Controls course mind-map images shown in Figures 3-5) which describes the MatLab environment, MatLab basics (assigning values to variables, element operations, basic graphics, help commands, etc.) and programming in MatLab (m-files, input/output, function m-files, anonymous functions, function functions, structured programming concepts and syntax, such as decisions (if, switch), repetitive calculations (for, while), nesting, etc.).

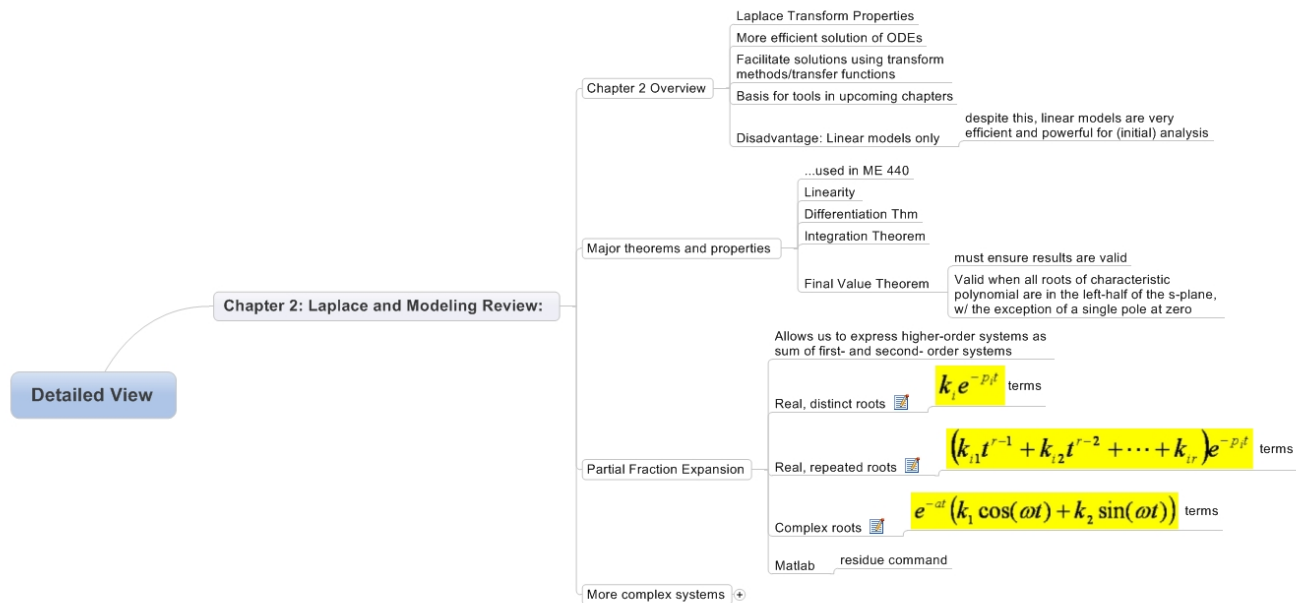


Figure 4. Partially Expanded node of Chapter 2 (Laplace and Modeling Review) with Equations



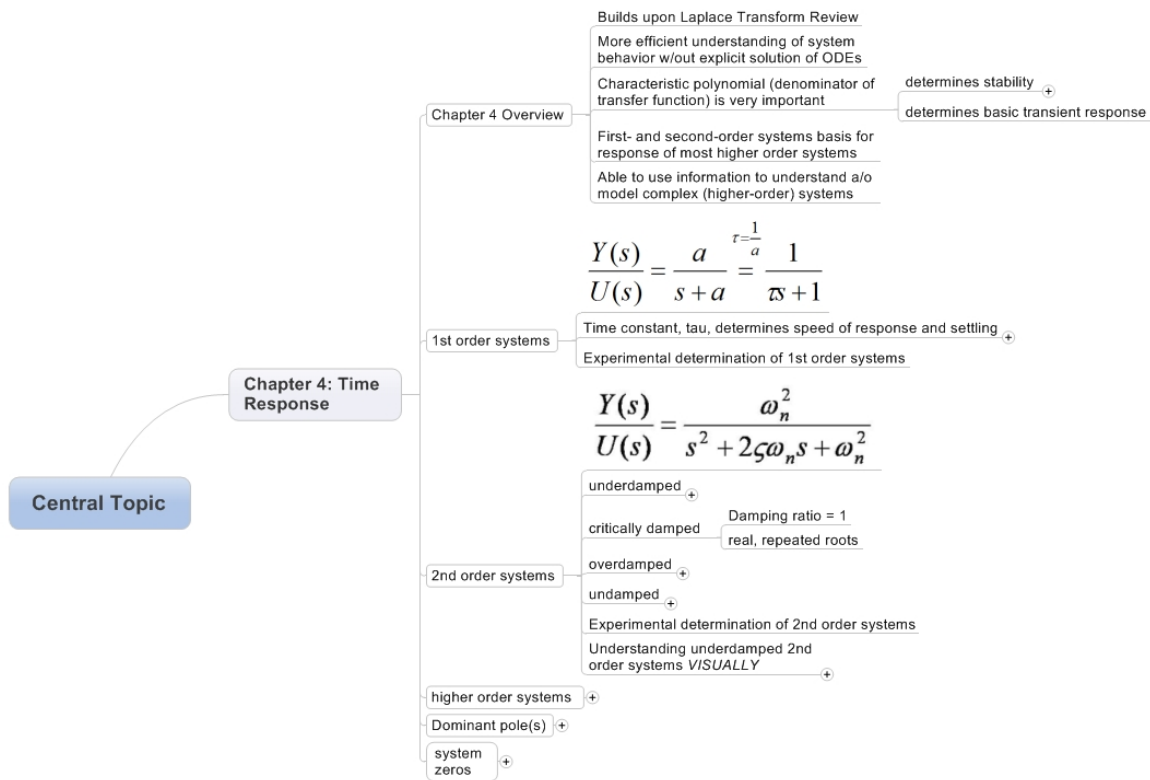


Figure 5. Partially Expanded node of Chapter 4 (Time Response) with Equations

Additionally, a Wiki to support programming activities using the MCU is hosted (e.g., at <http://matlab-nsfwiki.csc.tntech.edu/>) and contains the following content:

- 1) Labs (including an introductory laboratory on the toolkit and Matlab toolbox)
- 2) Help files
- 3) Pre-labs (including pages on pre-requisite programming constructs, such as I/O, decision-making (if/else or switch), loops (for, while) and arrays)
- 4) Resources (example programs, function reference pages and MCU hardware information)

For this study, the existing MatLab mind-map will be revised to both i) reference relevant Wiki pages and ii) link programming constructs to Controls concepts in the course concept mind-map. The linkages between maps will be discussed on Piazza and in class prior to demonstration activities. Clarifying information will also be added to this map in real-time, based upon students' questions or misconceptions.

### Recitation sessions and Hybrid Learning Structure

Optional weekly recitation sessions are available in either a smart classroom or computer lab (containing 50 PCs with MatLab software and an instructor station). The hybrid learning

structure of the course enables review of supporting information online outside of the classroom, facilitating more effective use of recitation sessions and portions of class lecture time to discuss the relationship and implementation of programming in proposed hands-on activities in more detail. Additionally, Piazza and PollEverywhere are used to assess student understanding of core course content and the programming constructs used to implement proposed activities.

### **Proposed activities**

The demonstration activities described in Table 1 will be initially implemented using the ECP<sup>®</sup> Rectilinear system described in the Hardware Setup section above. The rectilinear system (or the equivalent) has been chosen for initial activities, in part, due to its familiarity; i.e., students have seen and solved systems of simultaneous linear equations using lumped mass-spring-damper schematics in several previous prerequisite courses, including physics, differential equations, dynamics and systems modeling. Demonstration activities will be preceded by analytical/numerical assignments investigating similar system schematics to more effectively link demonstration activities to course concepts.

#### Activity 1: Analytical determination of system model and solution (Chapter 2):

Students will first obtain system models of mechanical translational systems using free-body diagrams, and assuming lumped parameters and linear element laws to obtain the system model (i.e., ordinary differential equation) for one and two degree of freedom systems (this activity is identical to initial HW assignments in previous offerings) and manually solve the system of equations to determine the time response using Laplace transforms and partial fraction expansion (or the equivalent), for overdamped and underdamped cases/scenarios.

#### Activity 2: Time Response/Experimental determination of transfer function/Model Verification (Chapter 4)

Students will investigate the transient and equilibrium responses of the systems investigated in Activity 1 using the final value theorem and system time constant(s) or damping ratio and natural frequency values, similar to activities conducted in prior course offerings. Secondly, students will use the Matlab Controls Toolbox or Simulink to investigate the response prior to using the rectilinear system to experimentally determine system time constant (first-order or heavily damped second-order) and damping ratio, damped/natural frequency values (for second-order underdamped cases) and compare the ideal responses to the actual response.

System Modeling and Time Response are pre-requisite course concepts and review topics in Controls covered in the Course; therefore, it is also possible to condense activities one and two into a single demonstration by initially characterizing system performance via time constant(s) and/o damping ratio/natural frequency values (or the equivalent), depending upon course size and student readiness.

#### Activity 3: Controller Design using Root Locus

Students will analytically investigate the unity-feedback response of systems investigated in Activities 1 and 2 using root locus techniques and proportional control. Students will then

analytically design a proportional-plus-derivative (PD) or lead controller to improve transient response (e.g., increase damping) prior to implementing the control algorithm using the rectilinear system, MCU and Matlab. The expected and actual responses will be compared and discussed in group activities, in class and via the Piazza discussion forum. Student understanding will be evaluated both before and after the proposed programming activities using surveys via PollEverywhere and Piazza.

### **Possible Systems for Future Implementation**

Educational Controls Products (ECP<sup>®</sup>) also provides a torsional (mass MOI-torsional spring-damper) system with similar flexibility as the Rectilinear system; the activities proposed using the Rectilinear system in Table 1 can be easily expanded to investigate the analytical and experimental performance of a 1, 2 or 3 degree-of-freedom torsional system in future activities

Additionally, a pendulum for future activities has been developed by the authors. Similar to translational and rotational mechanical systems such as provided by the ECP<sup>®</sup> systems, pendulum schematics and system models are both familiar to students and fairly easy to mimic with fidelity. Similar analytical and experimental activities as those proposed for the translational and rotational ECP<sup>®</sup> systems can be conducted for a pendulum setup. Moreover, pendulums systems provide an opportunity for students to examine the importance/validity of simplifying assumptions (e.g., linearization/small angle, neglecting the mass (MOI) of rod, etc.) and provide a good foundation for demonstrations utilizing more advanced system models.

Many of the systems described in this section, as well as activities 1, 2, and the first proposed frequency response activity, are appropriate for inclusion in a junior-level Modeling of Dynamic Systems course which is often a pre-requisite of the Feedback Controls Course. Upon implementation, initial assessment and feedback from students, modifying proposed programming activities for inclusion in the Modeling course will be considered and evaluated.

### **Expected Challenges and Assessment Opportunities**

Mechanical Engineering enrollments are growing nationally, so the challenges detailed in the National Academies study<sup>2</sup> are exacerbated in a large lecture course with three populations<sup>29</sup>: students with an excellent foundation in course pre-requisites, students with minor gaps in pre-requisite fundamental principles and students with significant gaps in pre-requisite fundamental concepts.

A second challenge is that the initial model<sup>1</sup> addresses the three principles of learning by exposing students to programming throughout the curriculum. In the initial implementation of this study, students may have only been exposed to significant programming activities in the pre-requisite computer science course (i.e., may not have been exposed to curricular linkages in prior engineering courses). Although a significant challenge, this is also an opportunity to design an implementation that is suitable for effectively using the model in upper level courses without waiting several years for a freshmen cohort that has been exposed to first-year programming experiences and curricular linkages in previous engineering courses to reach the junior or senior year.

## Bibliography

- [1] Canfield, S. L., and Abdelrahman, M. A., 2009, "Enhancing the Programming Experience for Engineering Students through Hands-on Integrated Computer Experiences" Proceedings of the 2009 ASEE Southeastern Section Annual Conference, Marietta, GA, April.
- [2] National Academy of Sciences. 2012, "Undergraduate science and engineering teaching needs improvement." ScienceDaily (21 May 2012. Web. 29 May 2012).
- [3] Bransford, J. D., Brown, A., & Cocking, R., 2000, How People Learn: Mind, Brain, Experience and School, Expanded Edition, Washington, DC: National Academy Press.
- [4] Committee on How People Learn, A Targeted Report for Teachers, How Students Learn: History, Mathematics, and Science in the Classroom, M. Suzanne Donovan and John D. Bransford, *Editors*, THE NATIONAL ACADEMIES PRESS Washington, D.C., 2005.
- [5] Calechman, S, 2012, "Learning Revolution," Delta SKY Magazine.
- [6] Vander Ark, T., 2011, "Getting Smart: How Digital Learning is Changing the World," Jossey-Bass (Wiley and Sons).
- [7] Felder, R. M., and Silverman, L. K., 1988, "Learning and Teaching Styles in Engineering Education," *Engr. Education*, 78 (7), pp 674-681 (updated online 2009\*).
- [8] Fink, D., 2003, "Creating significant learning experiences: An integrated approach to designing college courses." Jossey-Bass (Wiley and Sons).
- [9] Michaelsen LK, Knight AB, and Fink LD, 2004. "Team-based learning: A transformational use of small groups in college teaching." Stylus Publishing, Sterling, VA.
- [10] Michaelsen, L. K. and Black, R. H., 1994, "Building learning teams: The key to harnessing the power of small groups In higher education," in *Collaborative Learning: A Sourcebook for Higher Education*, Vol. 2, edited by S. Kadel, & J. Keehner, National Center for Teaching, Learning and Assessment, \_\_\_\_, State College, PA
- [11] Michaelsen, L. K., Watson, W. E. and Black, R. H., 1989, "A realistic test of individual versus group consensus decision making." *Journal of Applied Psychology*. 74(5), 834-839.
- [12] Lipscomb, L., Swanson, J., and West, A., 2004, "Scaffolding," *Emerging Perspectives on Learning, Teaching and Technology*, edited by M. Orey,
- [13] Vygotsky, L. S., 1978, *Mind in Society*, Harvard University Press, Cambridge, MA.
- [14] Borthick, A., 1999, "Designing learning experiences within learner's zones of proximal development (ZPDs): Enabling collaborative learning on-site and online"
- [15] Kao, M., Lehman, J., and Cennamo, K., 1996, "Scaffolding in hypermedia assisted instruction: An example of integration," *Association for Educational Communications and Technology*.
- [16] Parmelee, D. X, and Michaelsen, L. K, 2012, "Twelve tips for doing effective Team-Based Learning (TBL)," *Medical Teacher*, Volume 32, pp. 118-122.
- [17] Levine, RE, 2008, "Peer evaluation in team-based learning," *Team-based learning in health professions education*, edited by Michaelsen, LK, Parmelee, D, McMahan, K, and Levine RE., Stylus Publishing, Sterling, VA

- [18] Clough, D. E., Chapra, S. C. and G. S. Huvad, 2001, "A Change in Approach to Engineering Computing for Freshmen, - Similar Directions at Three Dissimilar Institutions," 2001 ASEE Annual Conference and Exposition, Jun 24-27, Albuquerque, NM.
- [19] M. H. Naraghi and B. Litkouhi, 2001, "An effective approach for teaching computer programming to freshman engineering students," 2001 ASEE Annual Conference and Exposition, Jun 24-27, Albuquerque, NM.
- [20] Adamchik, V. and A. Gunawardena, 2005, "Adaptive book: Teaching and learning environment for programming education", Proceedings ITCC 2005 – International Conference on Information Technology: Coding and Computing, Las Vegas, NV, Apr. 4-6, p 488-492.
- [21] Calloni, B. A. and D. J. Bagert, 1995, "Iconic programming for teaching the first year programming sequence," Proceedings – Frontiers in Education Conference, v 1, Atlanta, GA, Nov. 1-4, p 99-102.
- [22] Herniter, M. E., Scott, D. R., and R Pangasa, 2001, "Teaching programming skills with MATLAB", 2001 ASEE Annual Conference and Exposition, Jun 24-27, Albuquerque, NM.
- [23] Colombo, M. A., M. R. Hernandez and J. E. Gatica, 2000, "Combining high-level programming languages and spreadsheets an alternative route for teaching process synthesis and design," 2000 ASEE Annual Conference and Exposition, St. Louis, MO, June 18-21, p 773-784.
- [24] Jean-Claude Thomassian, Anoop Desai, and Patrick Kinnicut, 2008, "A Study of Student Attitude towards Media Based Instruction in Introductory Engineering Courses", Proceedings of the 38th ASEE/IEEE Frontiers in Education Conference, October 22 – 25, Saratoga Springs, NY
- [25] Nocito-Gobel, J. M. Collura, S. Daniels, and I. Orabi, 2005, "Are Attitudes Toward Engineering Influenced by a Project- Based Introductory Course?" Proceedings, 2005 American Society for Engineering Education Annual Conference and Exposition, Portland, Oregon, June 12 - 15.
- [26] Besterfield-Sacre, M.E., C.J. Atman, and L.J. Schuman, 1998, "Engineering Student Attitudes Assessment", *Journal of Engineering Education*, 87(2), pp. 133-141.
- [27] Likert, R., 1932, A Technique for the Measurement of Attitudes," *Archives of Psychology*, No.140.
- [28] Ghafoor, S., Canfield, S., Kelley, M. and Hill, T., 2011, "Assessment of Student Attitudes and its Impact in a Hands-On Programming Model for the Introductory Programming Course," Proceedings of the 2011 ASEE Annual Conference and Exposition, Vancouver, Ont.
- [29] Parker, J. M. and Polston, J. D., 2012, "Using Hybrid and Problem-Based Learning Techniques to Enhance Teaching Effectiveness in a Large Feedback Controls Lecture Course," Proceedings of the 2013 ASME International Mechanical Engineering Congress and Exposition, Houston, TX, Nov 9-16.
- [30] Dragon12 plus, [www.evbplus.com](http://www.evbplus.com)
- [31] Freescale Codewarrior, [www.freescale.com](http://www.freescale.com)
- [32] ECP® Rectilinear Plant, [http://www.ecpsystems.com/controls\\_recplant.htm](http://www.ecpsystems.com/controls_recplant.htm)
- [33] "Using Piazza to Encourage Interaction," *The Chronicle of Higher Education*, April 2012.
- [34] Young, Jeffrey, 2012, "Students Endlessly E-Mail Professors for Help. A New Service Hopes to Organize the Answers," *The Chronicle of Higher Education*, (April, 2012).
- [35] PollEverywhere, [www.PollEverywhere.com](http://www.PollEverywhere.com)
- [36] Nise, 2011, "Control Systems Engineering, 6th Edition," Wiley and Sons.