2006-675: INTRODUCING DISCRETE EVENT SYSTEMS INTO AN UNDERGRADUATE CONTROLS COURSE

Richard Hill, University of Michigan

Richard C. Hill received the B.S. degree in Mechanical Engineering from the University of Southern California in 1998, and the M.S. degree in Mechanical Engineering from the University of California, Berkeley in 2000. At Berkeley he worked as both a graduate student researcher as well as a graduate student instructor. From 2000 to 2002, he worked at Lockheed Martin Corporation on satellite attitude determination and control. He also spent two years as a high school math and science teacher in the San Jose area. He is currently pursuing the Ph.D. degree in Mechanical Engineering at the University of Michigan, Ann Arbor. His research interests lie in modular and hierarchical control of discrete-event systems, nonlinear control, and network control.

Dawn Tilbury, University of Michigan

Dawn M. Tilbury received the B.S. degree in Electrical Engineering, summa cum laude, from the University of Minnesota in 1989, and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley, in 1992 and 1994, respectively. In 1995, she joined the faculty of the Mechanical Engineering Department at the University of Michigan, Ann Arbor, where she is currently an Associate Professor. She won the EDUCOM Medal (jointly with Professor William Messner of Carnegie Mellon University) in 1997 for her work on the web-based Control Tutorials for Matlab. An expanded version, Control Tutorials for Matlab and Simulink, was published by Addison-Wesley in 1999. She is co-author (with Joseph Hellerstein, Yixin Diao, and Sujay Parekh) of the book Feedback Control of Computing Systems. She received an NSF CAREER award in 1999, and is the 2001 recipient of the Donald P. Eckman Award of the American Automatic Control Council. She belongs to ASME, IEEE, and SWE. She was a member of the 2004-2005 class of the Defense Science Study Group (DSSG) and is a current member of DARPA's Information Science and Technology Study Group (ISAT). Her research interests include distributed control of mechanical systems with network communication, logic control of manufacturing systems, performance management and control of computing systems, and uncertainty modeling in cooperative control.

Introducing Discrete Event Systems into an Undergraduate Controls Course: Development, Pilot Study and Assessment

Abstract

Discrete event systems are characterized by discrete states with event-driven state transitions. The modeling and control of such systems is often handled in a rather ad-hoc manner. Emerging research in developing analytical results for 'discrete event control' is leading to the formation of graduate level courses in the United States, and more commonly in Europe and Canada. This paper outlines a small instructional module introducing this material into a typical undergraduate controls course. Results of the implementation of such a module during the Fall 2005 semester are discussed.

Introduction

Courses on systems and control have become standard elements of most any undergraduate curriculum in mechanical engineering. These courses are very useful to their students and typically cover material ranging from the modeling of physical systems, to system analysis and controller design. In general, techniques for analysis and design are taught in both the time and frequency domains, and in the undergraduate curriculum are applied to continuous linear time invariant single input single output systems. While it is true that the techniques learned are very powerful and cover a large portion of the techniques the students will find being employed in industry, the students generally complete these undergraduate courses with little understanding of the limitations of the techniques they have learned. The students often fail to recognize that most systems they will encounter in the real world are inherently nonlinear, time varying, have multiple inputs and multiple outputs, and that controllers are often implemented digitally. Although the typical undergraduate curriculum is crowded enough as it is, and students have the ability to learn how to deal with these difficulties through elective courses and graduate study, it is important that the students come away knowing that these subtleties exist and that there are techniques out there for dealing with them. Much of the advancement taking place in controls education seeks to address these deficiencies through laboratory work and project-based learning.[1] [2]

A specific area of control system design and analysis that even many advanced graduate students are unaware of is that of discrete event system (DES) control. Discrete event control is often confused with digital control. Whereas digital control systems involve the sampling of an inherently continuous system at discrete time intervals, a DES is a system whose states are inherently discrete. For example, a milling machine is in an initialization state or a fault state, or the controller for an inverted pendulum is in the swing-up mode or the balancing mode. Another distinguishing characteristic of DES is that their evolution is event driven, not time driven. The transition between states occurs because of an event (such as a part arrives, or a continuous variable enters some range of values); the evolution doesn't necessarily correspond to the passage of time.

A benefit of introducing a DES type representation of a system to students is that it reinforces certain concepts they would have already learned in a systems modeling course. Students can see that the purpose of a model is to provide an abstraction that enables an engineer to capture only those aspects of the system they need for their analysis and design goals. For example, the exact same system can be represented by a free body diagram and a set of differential equations, or as a finite state machine. The former representation is useful for traditional classical control, the latter for discrete event control. Throughout this process, concepts like state, inputs and outputs, and the difference between a static and a dynamic system can all be reinforced.

A second reason for introducing discrete event control is to inform students that there are formal methods for designing logic control for systems with discrete states. Most students, and practicing engineers for that matter, would approach this design task in a rather ad-hoc manner. While it seems simple enough, without formal methods it gets very hard for a designer to keep track of everything going on in a system if there is even a moderate level of complexity. Anyone who has ever had the "check engine" light come on in their car, or had their computer lock up for absolutely no apparent reason might appreciate this.

Students can also be introduced to the concept of a hybrid system, a system which has discrete and continuous aspects to it. For example, a hybrid system might be a continuous system with two modes of operation where a switch between these modes may mean that the controller uses a different set of gains, or that the system behaves fundamentally differently in the two modes, like a plane in subsonic and supersonic flight. A result that students often find surprising is that a system that is stable within each of its modes can be driven unstable if the modes are switched between in a certain manner. While it is too much to ask for students to go too far into the analysis of a hybrid system, with the assistance of powerful simulation packages like SIMULINK, they can rather easily understand the basic phenomena that may occur in practice.

Most attempts to introduce the topics of DES and hybrid control have been reserved for specialized graduate level courses.[3] At the undergraduate level, some universities have created specific courses that address the more practical aspects of logic control implementation, specifically with regard to programmable logic controllers (PLCs).[4] [5] Other courses have been generated that cover manufacturing automation in general, including some aspects of DES control. An undergraduate level text that could be used in this type of manufacturing automation course is given in the bibliography.[6] The undergraduate controls curriculum of the Michigan Technological University includes a 3 hour module on "Discrete-Action Control" that addresses some aspects of modeling systems with state-transition type logic, while also introducing PLCs.[7]

As part of the upper division elective controls course (ME 461) here at The University of Michigan, a module has been developed to introduce students to the topic of DES and hybrid control. This module is unique in that it addresses both discrete event and hybrid systems, and also focuses on theoretical material rather than on PLC implementation. The module was implemented during the Fall 2005 semester and was covered over the course of two 50 minute class periods. The results of the module were assessed through homework and exam questions, as well as a student survey. The purpose of this paper is to present the content of the module and to discuss the results achieved.

Control Curriculum

The only required controls course in the undergraduate mechanical engineering curriculum of the University of Michigan is ME 360, *Modeling, Analysis and Control of Dynamic Systems*. The focus of this course is the modeling and analysis of various mechanical, electrical, fluid, thermal and multimode systems, along with the introduction of feedback control. Models are represented using differential equations and transfer functions, while analysis is performed in both the time and frequency domains. The course in general follows the structure of the assigned text.[8]

ME 461, *Automatic Control*, is a popular upper division elective that builds off the foundation built in ME 360. Frequency domain and complex plane analysis is covered in more depth, with the graphical methods of root locus, Bode and Nyquist plots emphasized. These techniques are applied in addressing the design of PID and lead/lag controllers. State space methods are also introduced as part of the course. The material covered in this course is similar to that found in several standard undergraduate controls textbooks.[9] [10] [11]

ME 360 and ME 461 have a structure consistent with the undergraduate controls sequence found at many American universities and neither addresses the topic of DES. The University of Michigan is one of a handful of domestic universities to offer a graduate course in DES (EECS 661), though it is becoming more common for such courses to exist, especially as occasional special topic type courses. EECS 661, *Discrete Event Systems*, is typically offered once every two years. While somewhat rare in the United States, this type of material is more commonly covered in the controls curriculum of many European and Canadian universities.

DES Background Information

As stated earlier, DES models are characterized as being event driven systems with discrete states. DES may be represented in a variety of ways, two of the most popular being **Petri nets** and **finite state machines (automata)**. An example finite state machine (FSM) model is given in Figure 1. This model represents a machine with the discrete states, *idle*, *working*, and *broken*. Transition between states is driven by the machine events *start*, *finish*, *break*, and *repair*.

The combined behavior of multiple finite state machines is generally characterized by an operation called **parallel composition**. The idea being that if two machines share an event, then the occurrence of that event in each of the machines must be synchronized. If a machine has an event that is not shared, then that event may occur without regard to the other machines.

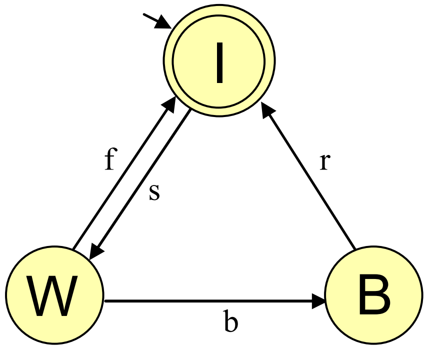

Figure 1 example finite state machine

One can imagine that if we had two instances of the finite state machine given in Figure 1, the first machine could be in the idle state while the second machine was in the idle, working, or broken state. Following this logic, there are combinatorially a total of nine possible states when considering the two machines together. Therefore, a factory of ten machines with five states each could have as many as 5^{10} states. The exponential growth of the number of states serves to demonstrate the limitation of applying ad-hoc methodologies to controlling DES.

Another way in which the behavior of DES is characterized is in terms of sets of strings of events. For the system given in Figure 1, an example string might be that the machine starts then finishes, *sf*, while another string might be that the machine starts, breaks, and then is repaired, *sbr*. A set of strings (or 'words') is referred to as a **language**. The problem of control of DES is generally formulated as trying to limit the system behavior to a certain subset of strings. For instance, if two machines like the one given in Figure 1 were operating together, two possible strings would be $s_1s_2f_1f_2$ and $s_2s_1f_2f_1$, where the subscripts refer to events of the first and second machine. If for some reason it was deemed that the operation of the first machine has higher priority than the second machine, a controller might be developed that allows the first string but prevents the second. Another controller might be needed to prevent a machine from starting operation if the piece it produces is going to overflow some downstream machine in the manufacturing process. Formal methods exist for generating and analyzing these types of controllers. In general, controllers are represented by finite state machines that generate the set of desired strings and the closed loop behavior is captured by the parallel composition of the open loop system FSM and the controller FSM. Methodologies also exist for dealing with situations where certain events either cannot be controlled or observed.

The bibliography of this paper cites a couple of references that serve as good introductions to the field of DES control.[12] [13]

Module Content

Since there are many important topics to cover in the undergraduate controls curriculum, there is generally not space available to teach students how to design and implement DES control. Rather, the purpose of this module is to introduce the concept of DES, raise awareness and reinforce some more fundamental concepts from the students' previous learning. A more in-depth level of instruction will be left to a specialized graduate level course.

The first component of this module is to teach the students what a DES is through modeling a physical system as a finite state machine. In the course of this process, the finite state machine is referenced to the modeling the students had done in ME 360. Noting how the same physical system can be modeled in two different ways reinforces the students' understanding of why and how engineers go about generating an abstraction of the physical world. Furthermore, such concepts as state, and inputs and outputs can also be reinforced.

The students are then taught how to model the concurrent operation of multiple DES through introduction of the parallel composition. As an example, the experimental setup shown in Figure 2 was brought to class and modeled using finite state machines and the parallel composition operation. Not only does this example illustrate the material, but the fact that the resulting model ends up having 1,179,648 states motivates for the students the need for formal methods. It is apparent to the students that with so many states, it would be impossible for a human designer to keep track of all the possible behaviors of the system. While this instructional module will not teach the students how to specifically design a DES controller, it will make them aware that analytical tools do exist.

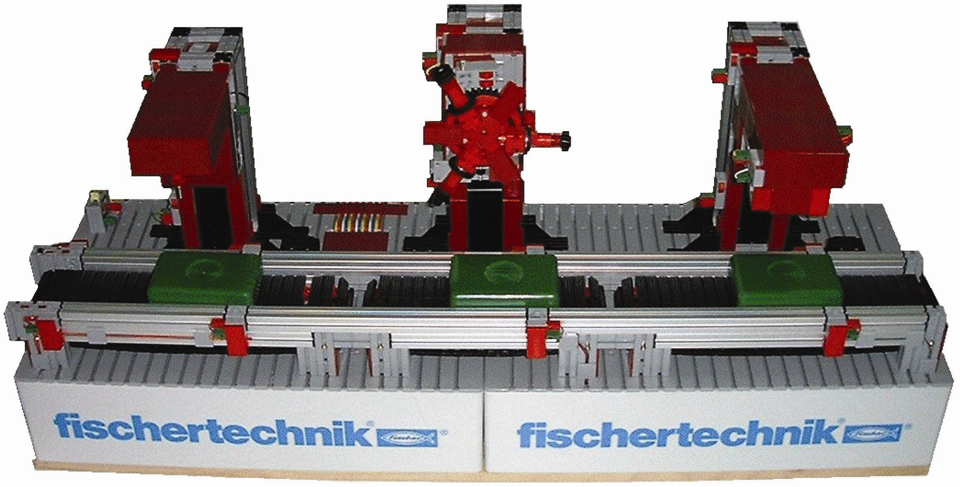

Figure 2 in-class example

The final part of this instructional module is to introduce systems that combine aspects of discrete and continuous models. These types of models are referred to as **hybrid systems** and their study is at the forefront of some of the most interesting research being done in the field of controls today. Examples of hybrid systems are all around us and can basically be thought of as physical systems whose fundamental behavior switches between different modes of operation. Examples include gain scheduling where different controller gains are switched between, or a situation where the physics of the plant changes, like a satellite changing the orientation of an antenna or solar panel. Therefore a hybrid model is just like a traditional system described by differential equations, except that there are discrete events that cause the system to transition to a behavior described by an entirely different set of differential equations.

Again it is beyond the scope of this DES module to provide students with the analytical tools to deal with hybrid systems. However, we can raise their awareness of the existence of such systems, and further, we can make them aware of the fact that the results they have learned in ME 360 and ME 461 don't necessarily apply to hybrid systems. This awareness of the limitations of their knowledge is essential and is a concept that runs through the controls curriculum in the discussion of other topics, like nonlinearities. Beyond exposing the limits of the analytical tools at their disposal, the students are also shown the usefulness of simulation in giving insight into the behavior a feedback control system. While the simulation isn't able to guarantee stability or a certain level of performance, it is able to give the students some level of confidence in a design. The specific example hybrid system that was given in the course of this DES module was an automobile transmission.

Since this material on DES and hybrid systems isn't covered in any standard undergraduate controls text, a short 3-page handout was provided to the students as a supplement.

Assessment

The assessment of this instructional module came in primarily two pieces. The first piece took the form of two homework problems and a problem on the final exam. The students' performance on these problems gave some sense of whether or not the students understood what they were being taught. The second piece of assessment was the end of course survey. The students' responses on this questionnaire indicated how much the students valued and enjoyed the DES module, especially in comparison to other portions of the course.

Of the two homework problems, the first question was primarily a traditional type problem where the students had to design a controller to place closed loop poles at a prescribed location and then simulate the closed loop response of the system. The twist came in that the students were told to design two such controllers, both of which were stable, then were instructed to simulate the response when these two controllers were switched between with a prescribed frequency. The surprising result seen by the students was that the switching resulted in unstable operation, even though each of the controllers produced a stable response when acting individually. While the given pole locations and switching frequency were contrived to cause instability, the fundamental lesson they provide is a powerful one. Figure 3 shows an example SIMULINK model used in the solution of the homework problem.

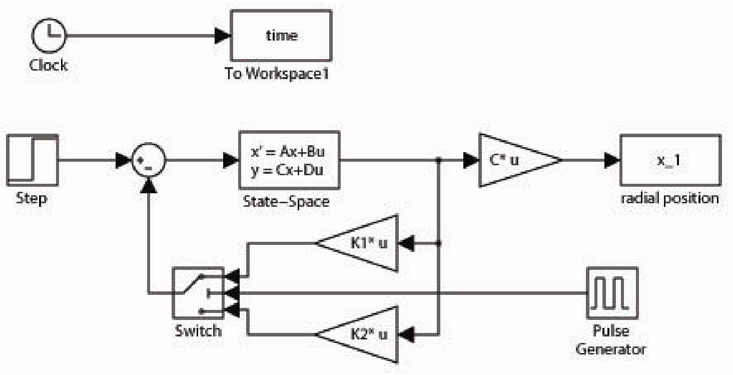

Figure 3 hybrid homework problem

The overall results from this problem were in general favorable with the average score being ~2.61 out of a possible 3. Almost all the students grasped that switching between stable controllers does not guarantee stable operation, which was the primary goal of the problem. Furthermore, the students seem to understand that even without having the tools to guarantee analytical results, they could have a reasonable amount of faith in their designs by simulating the system over a range of conditions. Some students tried to explain the mechanism by which the instability was caused without much success, though that material wasn't emphasized in the instruction of the unit. The majority of the points lost on the problem were related to the controller design or trouble implementing the simulation.

Another aspect of the course which reinforced the results of this problem was the class final project. As part of their final projects many students had systems involving switching, either between control gains, or between different plant models. Another interesting note is that following the class in which hybrid systems were introduced, many students came up to ask questions about the material and about what further classes they could take to learn more.

The second homework problem focused more directly on how to generate a DES model. In the problem students were given a finite state machine model of a poorly designed 35 cent gumball machine, Figure 4. The design was flawed in that the machine didn't accept nickels, didn't give change, couldn't deal with unexpected inputs, and would get stuck in a state it couldn't get out of if exact change wasn't input. The students were asked to assess the given model and to design an improved gumball machine. Furthermore, the students were asked to determine the total possible number of states in the case that multiple machines were acting concurrently. The average score on the problem was ~2.83 out of 3 indicating good understanding. Every student in the class recognized that the given gumball machine model would reach a deadlock for a certain sequence of events, this was the primary flaw of the given machine. Furthermore, almost every new design addressed the deadlock problem, though most did so in a fairly simplistic manner. For instance, quite a large number of students addressed the problem by returning a gumball without change if extra money was input. A large segment of students also chose to return all of the money input without a gumball if exact change wasn't given. The last part of the problem is where the majority of the points were lost, though most provided the correct answer and seemed to recognize just how quickly the number of states of their systems grew.

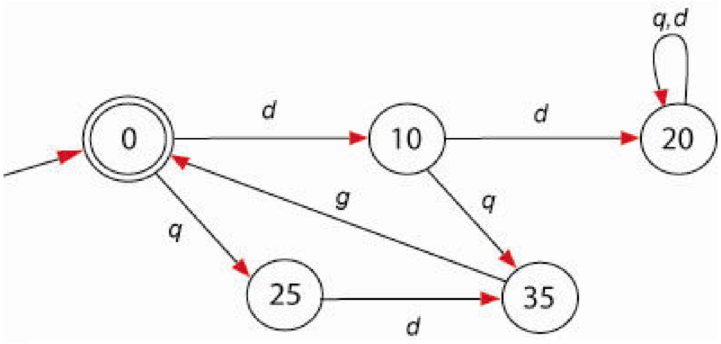

Figure 4 DES homework problem

The relevant problem included on the final was very similar to the DES problem from the homework. Again the students were given a flawed finite state machine, Figure 5, this time a 15 cent soda machine. First the students were asked to determine the action taken by the machine when two nickels were inserted followed by a dime. Every student in the class recognized that the machine reaches a deadlock and doesn't return a soda even though more than 15 cents had been input. The second part of the problem was to modify the machine to reflect a new soda price of 25 cents and to make the machine work "correctly." Most students were able to get this part of the problem correct also. Some students were able to modify the machine to reflect the new price, but didn't resolve the problem of a deadlock state. This deficiency might be attributed to some students not understanding what was meant by the requirement that the machine work "correctly." A handful of students weren't able to produce a soda machine that worked in any capacity. In total, the class averaged 83.3% on this problem, where the mean for the rest of the exam was 73.4%.

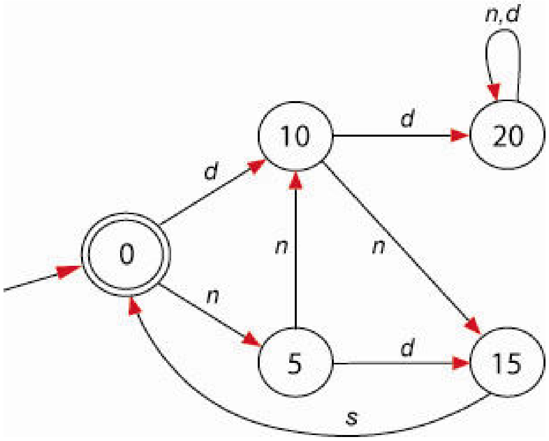

Figure 5 DES final exam problem

The final piece of assessment was the end of class survey. One part of the survey was a table that listed each section of the course where the students were supposed to indicate whether they thought the material was covered 'not enough,' 'about right' or 'too much.' The results of this question are displayed in Figure 6. Another question specifically asked whether or not the students thought the DES module was a valuable addition to the course. The problem with these types of questions is that the students don't necessarily have the perspective or experience to really evaluate how useful the material will be for them professionally; however, we still believed their feedback was worthwhile. The majority of students, 56%, thought the DES module took up about the right amount of class time. Most students also felt the material was valuable as reflected by comments like, "Lots of practical systems are discrete event systems. An introduction to the topic is really helpful." Of those students that would have liked to have seen things done differently, many felt that they did not learn enough from the module. In particular, they either didn't necessarily understand the application of the material, or would have liked to have seen some instruction on how control is implemented for DES. Some typical comments include, "not enough time to understand how to use and why we should care," and, "I think it was good to be exposed, but I don't feel we learned any control methods for it." These students therefore either wanted to see the module expanded, or done away with altogether, so more time could be spent on the other parts of the course, "… we should either cut it or do more."

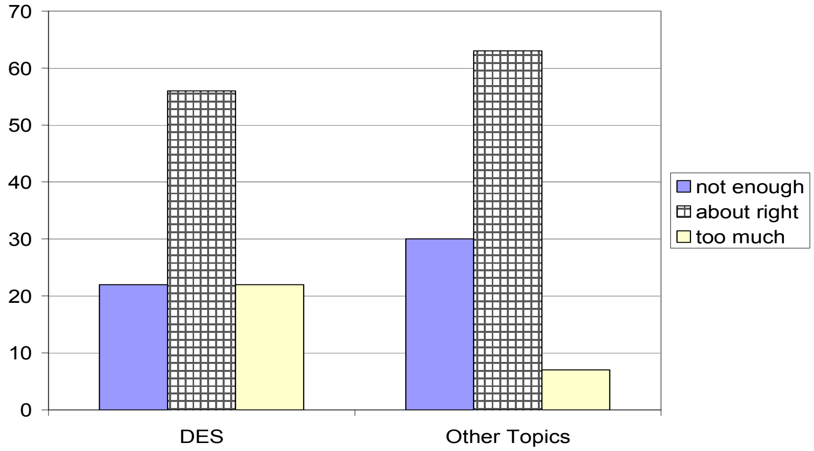

Figure 6 end of course survey results (% of students)

Conclusions

This paper described an instructional module that introduces DES to an undergraduate controls curriculum. The goal of this module is to introduce students to DES and to make them aware that formal methods exist for dealing with such systems. Another goal is to reinforce some concepts covered earlier in the students' controls education. Results of assessment were also presented in this paper indicating that the students understood the necessity for formal methods and understood the basics of modeling DES.

In terms of future modifications, the best way to improve the DES module would be to spend more time on it. One possibility would be to spend one more class period on the module. This extra period would be used to go through a more involved example where a DES controller is actually implemented. While the students wouldn't be tested on this extra material, it would give them a better understanding of the idea behind DES control. Another possibility would be to add an expanded module into an introductory graduate level controls course.

References

[1] Burchett, B.T., "A Control Systems Lab Sequence Designed to Foster Understanding," *Proceedings of the2005 American Society for Engineering Education Annual Conference & Exposition*. Portland, Oregon, June 12-15.

[2] Yoder, J.D., Hurtig, J. and Rider, M., "Providing Hands-on Experiences in a Mechanical Engineering Controls Systems Course," *Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*. Salt Lake City, Utah, June 20-23.

[3] Cassandras, C.G., Gong, W., Ho, Y.C., Hu, J.Q. and Vakili, P., "Boston University-Harvard University-University of Massachusetts/Amherst NSF Research and Curriculum Development on 'Analysis, Control, and Optimization of Discrete Event Dynamic Systems'," *Proceedings of the 1995 American Control Conference*. Seattle, WA, June.

[4] Rider, M., "A Unique, Undergraduate PLC Course," *Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*. Salt Lake City, Utah, June 20-23.

[5] Clough, D.E., "The Missing Link in Process Control Education – Incorporating PLC's Into the ChE's Control Course," *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*. Montreal, Quebec, June 16-19.

[6] Kaman, E.W., *Industrial Controls & Manufacturing*. San Diego, CA: Academic Press, 1999.

[7] Lewis, P.H., "Introducing Discrete-Event Control Concepts and State-Transition Methodology into Control Systems Curricula," *IEEE Transactions on Education*. vol. 37, no. 1, February 1994.

[8] Ogata, K., *System Dynamics*. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.

[9] Franklin, G.F., Powell, J.D., and Emami-Naeini, A., *Feedback Control of Dynamic Systems*. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[10] Nise, N.S., *Control Systems Engineering*. Hoboken, NJ: Wiley, 2004.

[11] Ogata, K. *Modern Control Engineering*. Upper Saddle River, NJ: prentice Hall, 2002.

[12] Cassandras, C., and Lafortune, S., *Introduction to Discrete Event Systems.* Boston, MA: Kluwer Academic Publishers, 1999.

[13] Wonham, W.M., *Notes on Control of Discrete-Event Systems.* ECE Department, University of Toronto, revised 2004.07.01. www.control.utoronto.ca/people/profs/wonham.