Editors: Denis Donnelly, donnelly@siena.edu
Stephen Nash, snash@gmu.edu

THE ROLE OF COMPUTERS IN MECHATRONICS

By Kevin Craig

MECHATRONICS IS THE SYNERGISTIC COMBINATION OF MECHANICAL ENGINEERING, ELECTRONICS, CONTROL SYSTEMS, AND COMPUTERS. THE KEY ELEMENT IN MECHATRONICS IS THE INTEGRATION THROUGH DESIGN OF THESE areas from the very beginning of the design process—no afterthought add-ons allowed.

Synergism in design sets a mechatronic system apart from a traditional, multidisciplinary system, which requires integration. A mechatronic approach, however, leads to synergism in design. The difficulty in the mechatronic approach is that it requires a system perspective: system interactions are important, system modeling is required, and feedback control systems can go unstable. Mechatronic design concepts include direct-drive mechanisms, simple mechanics, system complexity, accuracy and speed from controls, efficiency and reliability from electronics, and functionality from microcomputers. Starting at design and continuing through manufacture, mechatronic designs optimize the available technologies to produce quality precision products and systems in a timely manner with features that the customer wants. The real benefits to industry of such an approach to design are shorter development cycles, lower costs, and increased quality, reliability, and performance.

In a mechatronic system, computer, electronic, and control technology allow changes in design philosophy that lead to better performance at lower cost. Automotive engine-control systems are a good example; here, a multitude of sensors measures various temperatures, pressures, flow rates, rotary speeds, and chemical composition and sends this information to a microcomputer. The computer integrates this data with preprogrammed engine models and control laws and sends commands to various valves, actuators, fuel injectors, and ignition systems to manage the engine's operation for an optimum combination of acceleration, fuel economy, and pollution emissions.

In this article, I'll examine what role computers hold in the field of mechatronics, and what we as educators need to focus on for all engineering students, not just mechanical engineering students.

Is Mechatronics New?

Mechatronics is simply the application of the latest, cost-effective technology to the design process to create more functional and adaptable products, and even though forward-thinking designers and engineers have applied such techniques for years, there is something new here.

Mechatronics encompasses the knowledge base and the technologies required for the flexible generation of controlled motion. It demands horizontal integration among the various disciplines as well as vertical integration between design and manufacturing.

Furthermore, mechatronics is a significant design trend—a mixture of technologies and techniques that together can help design better products. It is already having a profound influence on the way all mechanical engineers are now expected to design and on the way professors must now teach design. Mechatronics has gained industrial and academic acceptance worldwide as a field of study and practice.

How Are Mechatronics Engineers Using Computers?

In mechatronic systems, computers play a variety of roles. They model, analyze, and simulate mechatronic systems and components and, as such, are useful for control design. As part of measurement systems, they are used to measure the performance of mechatronic systems, to determine the value of component parameters, and to validate models experimentally. Finally, they form the central component in digital control systems for mechatronic designs.

The Role of Computer Modeling and Simulation

To evaluate concepts generated during the design process without building and testing each one, the mechatronics engineer must be skilled in the modeling, analysis, and control of dynamic systems and must understand the key issues in hardware implementation. The essential characteristic of a mechatron-

Copublished by the IEEE CS and the AIP 1521-9615/03/$17.00 © 2003 IEEE

ics engineer, and the key to success in the mechatronics field, is a balance between two skill sets:

- Modeling (physical and mathematical), analysis (closed-form and numerical simulation), and control design (analog and digital) of dynamic physical systems
- Experimental validation of models and analysis and understanding the key issues in hardware implementation of designs

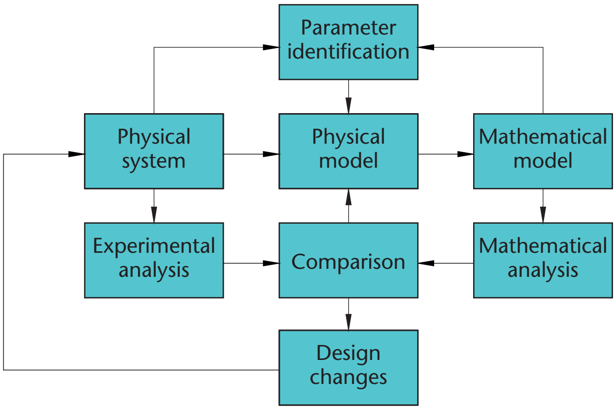

Figure 1. The dynamic system investigation process used to understand and possibly improve a mechatronic system.

Figure 1 illustrates the steps in a dynamic system investigation; it shows the process used to understand and possibly improve a mechatronic system. The distinction between physical and mathematical modeling is key, as is the importance of both analytical and numerical solutions to the model equations.

To generate a physical model, approximations must be made to the actual physical system: small effects are neglected; the environment's influence is ignored; elements are assumed to be lumped instead of distributed; the dynamics are assumed to be linear; parameters are assumed to be constant; and noise and uncertainty are ignored. These approximations have a direct effect on the physical and mathematical model. Neglecting small effects, for example, limits the number of equations, and environmental independence reduces the equations' complexity. Other approximations result in linear ordinary differential equations with constant coefficients; neglecting uncertainty avoids the use of statistics in the model. In most cases, a design consideration is to develop the simplest model that adequately depicts the system dynamics' complexity.

The model's predicted dynamic behavior is only half the story because these results, without experimental verification, are questionable at best and useless at worst. Comparing the predicted dynamic behavior with the actual measured dynamic behavior is the key step in the dynamic system investigation process.

We should apply the steps in this process not only when an actual physical system exists (as in reverse engineering), and we want to understand and predict its behavior, but also when the physical system is a concept in the design process that should be analyzed and evaluated. After recognizing a need for a new product or service, we can use past experience (personal and vicarious), awareness of existing hardware, understanding of physical laws, and creativity to generate design concepts. The importance of modeling and analysis in the design process has never been more important than in this situation. Obviously, the build-and-test approach can no longer evaluate such design concepts because it is too costly and time-consuming. Validating the predicted dynamic behavior in this case, when no actual physical system exists, then depends even more on the engineer's past hardware and experimental experience.

The Use of Real-Time Computers

A mechatronic system typically involves continuous variables: elements rotate or translate in space, fluids or gasses flow, and heat or energy is transferred. Computers are, by their nature, digital elements. Variables are represented in a computer by discrete values or simply by collections of zeroes and ones. For a computer to be used as the controller for a mechatronic system, therefore, the continuous variables must be converted to discrete variables for processing and then back again to continuous variables. This might seem obvious, but what is not so apparent is that the computer algorithm forms an inherent separation between the processing of the signals and the signals themselves, which is not true of other mechatronic system components.

Even if we use digital logic elements and the signals are converted to discrete form, the flow of information will still be continuous through the elements. When we use a computer for the control element, this information flow is broken and buried in the computer algorithm. As an example, computer algorithms sometimes mimic continuous proportional-integral-derivative (PID) control laws. When the execution of this algorithm is analyzed—even if the effects of sampling and quantization are included—we assume that the signals are processed just as if they were being determined by continuous processing elements. In reality, if we examine the computer code at the machine level (not in the high-level language in which it might be written), it bears very little resemblance to a differential equation representation of the

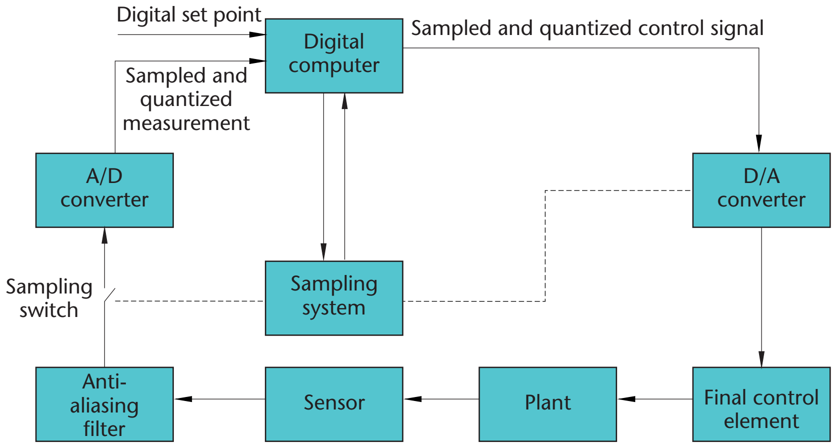

Figure 2. In a digital control system, the sensor is usually an analog device requiring sampling of the sensor's voltage via an appropriate sampling system synchronized with the computer's clock and analog-to-digital (A/D) conversion at the computer input. The final control element is also typically an analog device, and digital-to-analog (D/A) conversion is required at the computer output. An anti-aliasing filter (low-pass filter) prevents high-frequency signals, sampled too infrequently, from appearing as lower-frequency signals, which have no physical meaning.

PID algorithm. This has practical implications both for modeling the exact operation of the computer as a control element and for validating that the computer code actually produces the desired response to signals.

Other issues are involved when the mechatronic system controller is implemented in software. Software execution is often asynchronous to the other time constants in the system (that is, the software execution and system response are often not synchronized). We can make software synchronous by syncing it to the sampler period, but this typically limits performance and is difficult if the computer is to be used for tasks other than control.

Once a computer is contained as an element in a mechatronic system, developers are tempted to use some of the processing power to provide additional functionality or ease of use for the product. This additional code can affect, sometimes adversely, the operation of the real-time controller execution. Code testing and safety are also issues. The engineer must determine that the system operates deterministically and safely for all possible combinations of input signals and for all possible states in the algorithm's execution. For real-time systems, execution order for the code is often not predictable because it can depend on a particular combination of input signals. Generating simple code, providing for code testability, using established software quality assurance practices, and developing extensive documentation are ways to achieve system determinism and safety. Often, a hardware interlock—a safety system using electronic or mechanical hardware—is often included in software-controlled systems.

Code operation must be further verified as the code is modified and reused for systems other than those for which it was developed. Unlike other controllers, computer code is portable, but this means its possible reuse requires more thought. Using standard software packages, processors, modular code, and commercial real-time environments increases the possibility for reuse.

Incorporating a digital processing component into a mechatronic system raises several issues. Furthermore, we must take into account certain considerations whenever digital signals are processed. The computer is important, but the computer "component" of many mechatronic machines and processes is often not the critical system element in terms of technical or economic factors. Rather, components external to the computer—actuators, sensors, the sampling system, and the anti-aliasing filter—are more often limiting factors in system design. Figure 2 shows a general computer-control configuration.

Because most mechatronic systems are analog, we must have both analog-to-digital (A/D) and digital-to-analog (D/A) converters, which serve as translators that let the computer communicate with the outside analog world. In most cases, the sensor and the final control element are analog devices, requiring A/D and D/A conversion at the computer input and output, respectively. There are exceptions, of course. In most cases, though, we can think of the sensors as providing analog voltage output, and the final control element will accept an analog voltage input.

In a real sense, some of the problems of analysis and design of digital control systems (beyond the issues associated with software) are concerned with taking into account the effects of the sampling period T and the quantization size q. If both T and q are extremely small (for example, with a sampling frequency 50 or more times the system bandwidth with a 32-bit word size), digital signals are nearly continuous, so we can use continuous methods of analysis and design. It is most important to understand the effects of all sample rates, fast and slow, and the effects of quantization for large and small word sizes. Lower-cost computers are typically slower and have a smaller word size. Speed and word size is not an issue in prototyping because one would use a fast computer with a large word size (16 bits), but with prod-

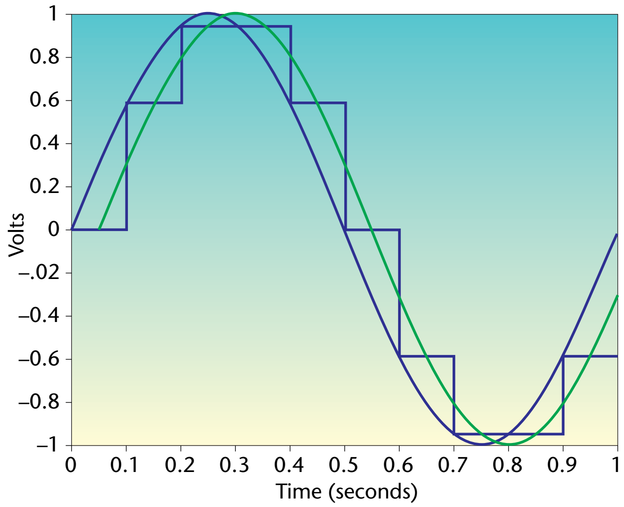

Figure 3. Continuous output from an analog controller and the computer controller output processed through a D/A converter with a zero-order hold, a signal discrete in amplitude and continuous in time (such as a staircase function). Drawing a curve through the midpoints of the steps shows that one effect of the D/A conversion is to add a time delay of one half the sample period compared to the analog controller output.

uct realization, typically slower microcontrollers with limited word size (8 bits) are used to keep cost down. With too small a word size, the signal that the controller processes will have large errors over the original analog signal.

Figure 3 shows the effects of sampling. The single most important impact of implementing a control system digitally is often the delay associated with the D/A converter. This pure delay results in a substantial phase shift in the closed-loop feedback system and often limits the control operation. Because most control systems will be implemented digitally on a computer, it is important for the mechatronics engineer to understand that implementation's key issues. Time delays degrade the performance of all control systems, and in a digital control system there are several sources of time delay—the major one being from the D/A converter.

In a feedback system, the analog signal coming from the sensor contains useful information related to controllable disturbances (relatively low frequency), but it can also include higher frequency noise due to uncontrollable disturbances (too fast for control system correction), measurement noise, and stray electrical pickup. Such noise signals cause difficulties in analog systems, and low-pass filtering is often needed to allow good control performance. The phase shift from this filter also adversely effects control system stability.

Finally, in digital systems, a phenomenon called aliasing introduces new aspects to the area of noise problems. If a signal containing high frequencies is sampled too infrequently, the output signal of the sampler contains low-frequency (aliased) components not present in the signal before sampling. If the higher frequency signal is sampled too infrequently, the result will be exactly the same values as for the low-frequency signal. From the controller's standpoint, there is no way for the system to distinguish which signal is present. If we base our control actions on these false low-frequency components, they will, of course, result in poor control.

The theoretical absolute minimum sampling rate to prevent aliasing is two samples per cycle. However, in practice, rates of about 10 are more commonly used. A high-frequency signal, inadequately sampled, can produce a reconstructed function of a much lower frequency, which cannot be distinguished from that produced by adequate sampling of a low-frequency function.

What Skills Do Engineers Currently Lack?

As a result of teaching mechatronics at Rensselaer Polytechnic Institute and of conducting many hands-on mechatronics workshops for engineers in industry, I've observed many things.

First, control design and implementation is still a specialists' domain. Most mechanical engineers have no practice designing and implementing a control system as part of a design. This grave deficiency can be traced back to their undergraduate engineering training: control is usually taught very late in the curriculum, if at all, as a mathematical exercise with hardly any design and very little implementation.

Secondly, few practicing engineers perform any kind of physical or mathematical modeling. The standard procedure in industrial design is to build and test each design concept. Engineers are under such pressure to deliver hardware, though, that they are not given the time to model and develop physical insight. Even if engineers are given the time to model, physical modeling is a subject neglected in undergraduate engineering education because, in most instances, the problem presented starts not with the actual system but with a physical model. The problem ends with computer simulation results and no comparison to the actual system behavior. Also, teaching modeling requires experience with actual physical systems—something that many faculty members lack.

Third, mathematics is a subject that is not viewed as enhancing engineering skills but as an obstacle to avoid. At the

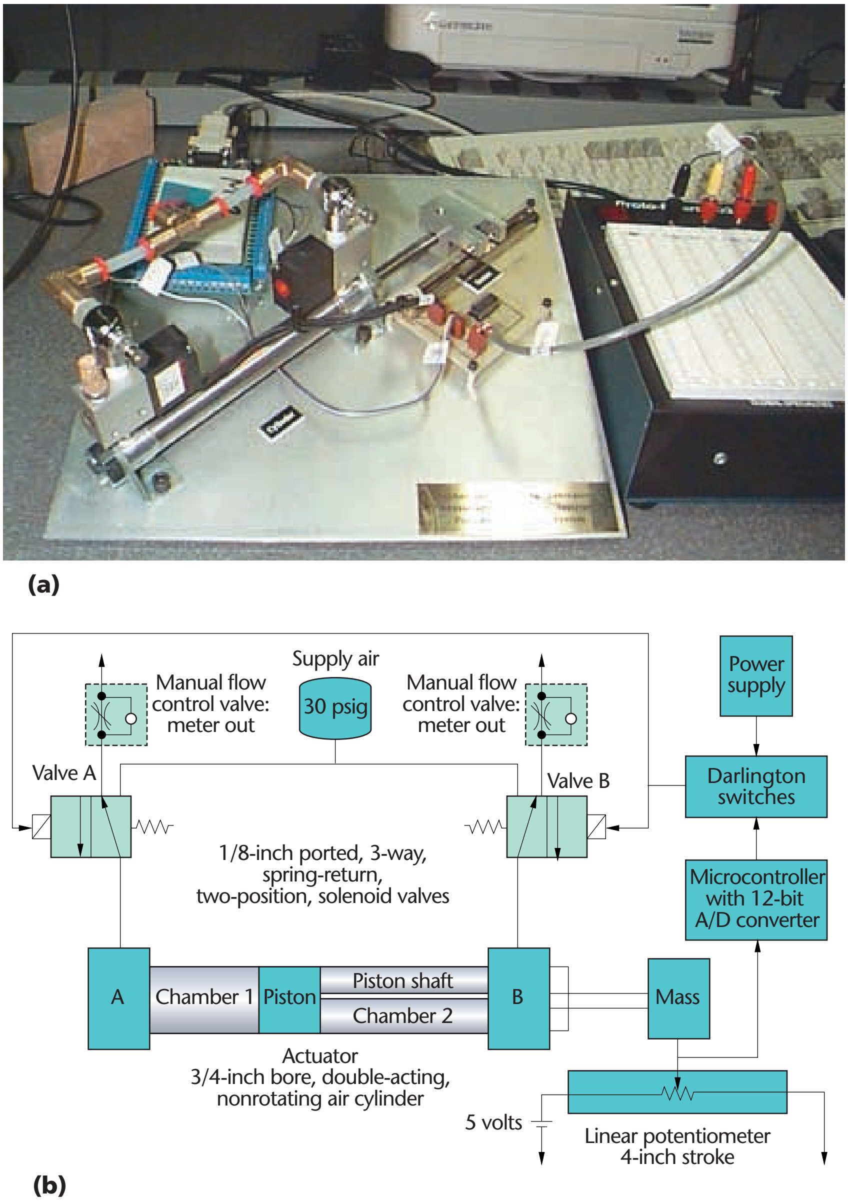

Figure 4. A picture and schematic of a pneumatic closed-loop computer-controlled positioning system. The objective of the positioning system is to use a pneumatic cylinder to accurately position a mass, in the presence of significant friction by using inexpensive, on-off, solenoid-actuated valves to control the motion of the pneumatic piston attached to the moving mass.

upper undergraduate levels, professors do not emphasize enough the importance of mathematics in engineering. Also, the computer programs so widely used in the present curriculum are often used without any understanding of the underlying physical and mathematical principles involved. As a result, students avoid physical and mathematical modeling because any modeling that is done is viewed often as not being very useful and is done by engineers with no hardware experience.

Fourth, few engineers have the balance between analysis and hardware essential for success. Our engineering education system does not emphasize this enough, and the trend will worsen as engineering schools continue to cut costs and rely more on computers to perform virtual experiments. How often have we heard someone say that the experiment must be wrong because the experimental results don't match the simulation results? This is very scary indeed!

Finally, too much emphasis is placed on computer use. The first law of computers for engineers is garbage in, garbage out. Students and practicing engineers look to computers for answers rather than understanding and applying fundamental principles and a structured approach to problem solving to provide answers. The computer is a tool, just as the calculator and slide rule before it were, and nothing more.

Mechatronic Systems' Use of Computers in a Balanced Way

Consider the picture and schematic of a pneumatic servomechanism—a computer-controlled, closed-loop positioning system—in Figure 4. Pneumatic servomechanisms have the advantages of low cost, high power-to-weight ratio, ease of maintenance, cleanliness, and a readily available and cheap power source. However, the disadvantages are high nonlinear friction forces, deadband due to stick-slip Coulomb friction or *stiction*, and dead time due to the compressibility of air.

The design goal is to implement a fast, accurate, and inexpensive pneumatic-actuator system using inexpensive on/off solenoid valves, rather than expensive continuously variable servo valves. To accomplish this task, we must completely understand the physical system, develop a physical model on which to base analysis and design, and experimentally determine and validate model parameters. We must also develop a mathematical model of the system, analyze the system, and compare the results of the analysis to experimental

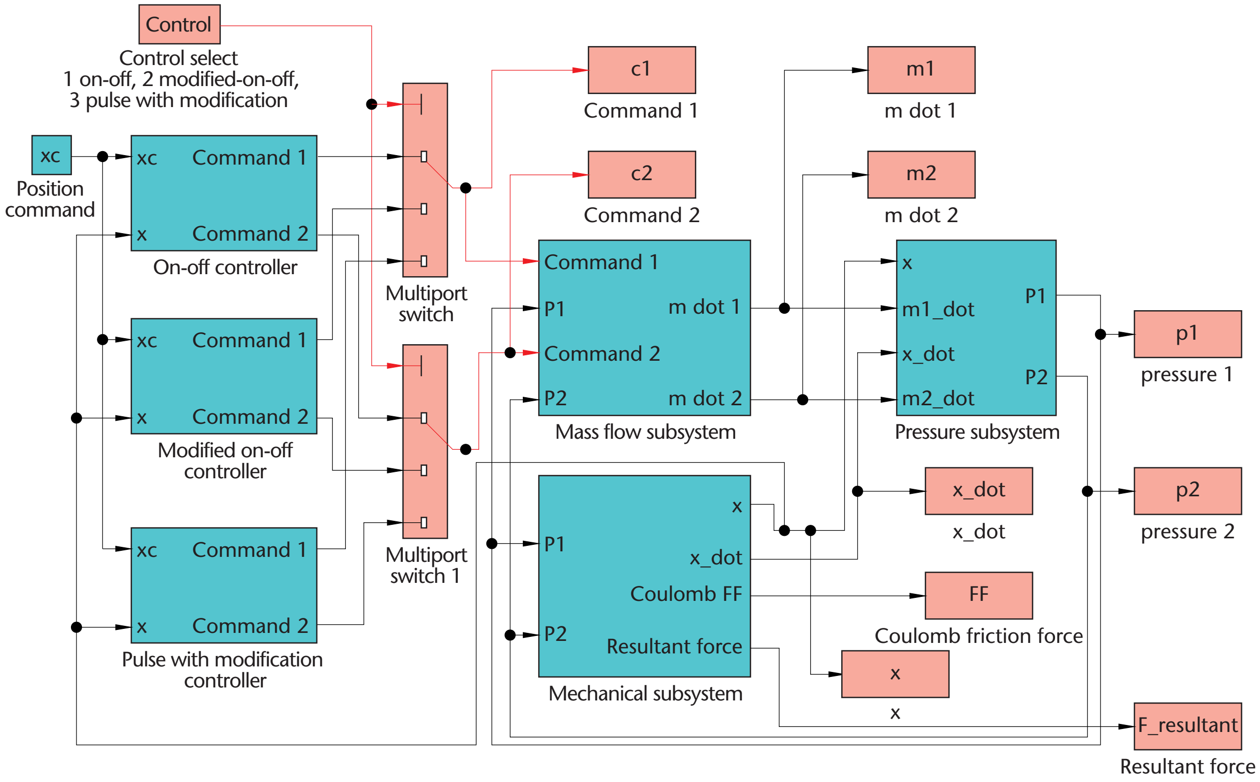

Figure 5. This MatLab/Simulink block diagram shows the mathematical models of the various system components: three different control schemes, the equations describing the pneumatic cylinder operation, the equations describing the motion of the mass with friction, and the equations describing the flow of air through the solenoid-operated valves. It represents a model close to reality, and is used primarily to predict behavior prior to hardware implementation.

measurements to validate the model. We then design a closed-loop position control system using on/off, modified on/off, or pulse-width modulated control. Finally, we implement the control system and experimentally validate its predicted performance.

Figure 5 shows a MatLab/Simulink model of this system. The mathematical model is highly nonlinear, as are the various control schemes. A computer numerical simulation is needed to understand the behavior of the system and the various control schemes. A data acquisition system measures the various system inputs and outputs and validates the numerical simulation. We need a computer (a microcontroller, in this case) for the real-time implementation of the various control schemes. A variety of computer numerical simulation tools are available, some requiring a detailed mathematical model while others enable virtual prototyping (system components are assembled on the computer screen with the component mathematical models hidden in the background).

Integration, fundamentals, balance, and relevance are the key elements in a successful university mechatronics program. Courses must emphasize fundamentals and the application of these fundamentals to industrially relevant problems. The importance of mathematics and the language of science and technology must be emphasized throughout the sequence, and computers should be used extensively but only as a tool to enhance understanding by speeding up detailed work and performing numerical simulations of nonlinear systems. The subject matter's relevance to the student's career must be addressed throughout the program to motivate learning. In other words, the material presented must be tied to the needs of today's practicing engineers and scientists.

All mechanical engineers must become mechatronics engineers, regardless of their concentration. Now more than ever, there is a need for professors to balance modeling and analysis with hardware implementation in their teaching. Industry wants and needs these skills in mechanical engineering graduates, so professors must meet this challenge head on.

Kevin Craig is an associate professor of mechanical engineering at Rensselaer Polytechnic Institute in New York. His research interests include mechatronic system design, control systems, modeling, dynamics, and the study of active materials and their application in design. He received a BS from the US Military Academy, West Point, New York, and an M.Phil. and PhD from Columbia University. Contact him at craigk@rpi.edu; www.rpi.edu/~craigk.