Proceedings,
The 2013 ASEE North Midwest Section Conference,
October 17-18, 2013, Fargo, North Dakota.



# Engineering Education in Context: 2nd- and 3rd-Year Required Systems Engineering Courses

Dr. Kevin Craig
Greenheck Chair in Engineering Design & Professor of Mechanical Engineering
Marquette University kevin.craig@marquette.edu


## Abstract

In modern multidisciplinary engineering practice, the necessary skill set includes modeling and analysis of multidisciplinary dynamic engineering systems, control system design and implementation, and sensors and actuators with the necessary electronics. Theory and best industry practice must be in balance when mastering these skills. Presently, we devote separate courses to each skill and somehow think that learning each skill very well will somehow magically enable the student to critically think and integrate all to solve a real-world problem. This approach is ineffective. As a result, the ABET-required senior capstone multidisciplinary design course too often becomes a design-build-test exercise with the emphasis on just getting something done. Students rarely break out of their disciplinary comfort zone and thus fail to experience true multidisciplinary, model-based system design. Two courses were created to address this. *Electromechanical Engineering Systems* (2nd-year) and *Multidisciplinary Engineering Systems* (3rd-year) are required courses in the mechanical engineering curriculum and were developed and taught over the past two years. They each consist of two hours of class time and two hours of small-group (12 students per session) hands-on, hardware and software, studio sessions each week. They are each taught in the context of modern engineering practice and real-world problem solving.

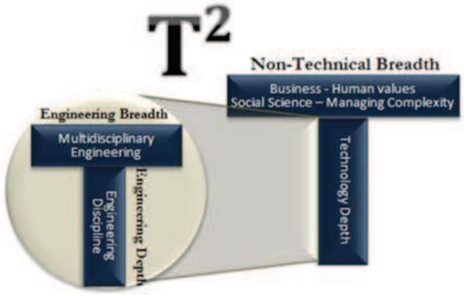

## Introduction and Motivation

If a young person wants to be a complete baseball player, he must be able to field, throw, run the bases, hit, and hit with power, and all these skills must be applied in an actual baseball game. To achieve this goal, he learns all these skills at the same time, improving gradually in each one while playing actual games and, over time, develops into a complete baseball player. The result is more than just the sum of the skills learned, but a sense of confidence and savvy that makes him a winner.

In modern multidisciplinary engineering practice, the necessary skill set includes modeling and analysis of multidisciplinary dynamic engineering systems, control system design and implementation, and sensors and actuators with the necessary electronics. Theory and practice must be in balance when mastering these skills. If "playing a game" means putting these together to create a system to solve a problem, then that rarely happens in engineering education, and if it does, it happens for only a few students who aggressively seek out, in a team-based setting, that integrated, total experience. We devote separate courses to each skill and somehow think that learning each skill very well will somehow magically enable the student to graduate and critically think, integrate it all, and solve a real-world problem. In the baseball analogy, this would be utter madness, yet in engineering education, it is routine.

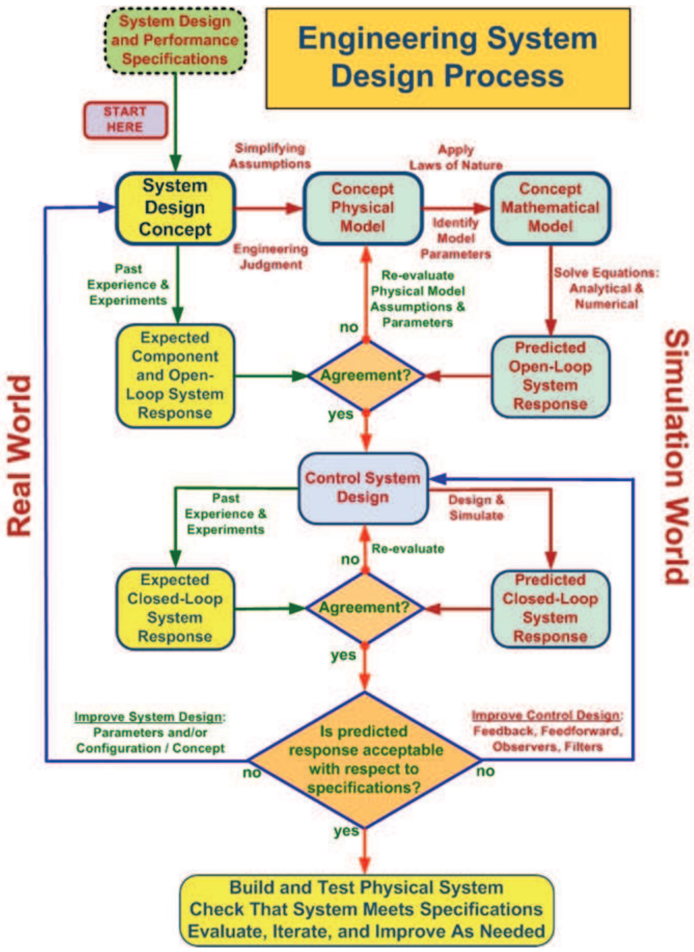
The present situation then is that undergraduate engineering education today is ineffective in preparing students for multidisciplinary system integration and optimization – exactly what is needed by companies to become innovative and gain a competitive advantage in this global economy. While there is some movement in engineering education to change that, this change is not easy, as it involves a cultural change from the silo approach to a holistic approach. The ABET-required senior capstone multidisciplinary design course too often becomes a design-build-test exercise with the emphasis on just getting something done. Students rarely break out of their disciplinary comfort zone and thus fail to experience true multidisciplinary-system, model-based design. What is needed are multidisciplinary systems courses, with a balance between theory and hardware, between academic rigor and the best practices of industry, presented in an integrated way in the 2nd and 3rd years that prepares students for true multidisciplinary-system, model-based engineering at the senior level and beyond.

Do technological universities and industry have a common goal? I believe fundamentally they do: solve the most urgent problems that face society to give people throughout the world the quality of life they all yearn for and nurture a planet that can sustain and enhance this quality of life indefinitely. Universities now need to do this through an integrated, multidisciplinary curriculum that recognizes this need and a delivery system that nurtures students to master the fundamental knowledge and the problem-solving process, along with the technological tools, to become catalysts for change, i.e., critical thinking problem solvers. Industry urgently needs to attain and retain a competitive advantage by organizing multidisciplinary teams to apply human-centered, model-based design techniques to these problems. Universities need to develop, and industry needs to hire, $T^2$ engineers (see diagram) as envisioned by, for example, Stanford University, and IDEO and MAYA, two of the world's leading design firms. The diagram is a composite of their views. Engineers need to have depth in an engineering discipline with multidisciplinary engineering breadth to communicate with engineers from other disciplines and lead them. The problems are multidisciplinary and a siloed approach to solving them will fail. But technological depth is not enough! Once engineers apply human-centered design to identify the real problem and model-based design to identify a technologically-feasible solution, they then must determine if the proposed solution is viable and sustainable from a business point of view and usable from a managing complexity point of view. Technological depth and non-technical breadth are essential for innovation to happen, as engineers need to grow professionally daily.




The two courses described in this paper – Electromechanical Engineering Systems (2nd-year) and Multidisciplinary Engineering Systems (3rd-year) – are required courses in the mechanical engineering curriculum and have been developed and taught over the past two years. They each consist of two hours of class time and two hours of small-group (12 students per session) hands-on, hardware and software, studio sessions each week. They are each taught in the context of modern engineering practice and real-world problem solving following the *Engineering System Design Process* shown in the diagram below.

The top two drivers in industry today for improving development processes are shorter product-development schedules and increased customer demand for better- performing products.  As engineering systems are becoming ever more multidisciplinary and complex, can these two goals be achieved at the same time?  Challenges inhibiting multidisciplinary product development fall into two categories: the multi-domain nature of the complete system and integration of the domains, and finding errors early in the development cycle and testing before hardware is available.  Once a system is in development, correcting a problem costs 10 times as much as fixing the same problem in concept. If the system has been released, it costs 100 times as much.   The *Engineering System Design Process* shown addresses these challenges. Through system modeling and simulation, it facilitates: understanding the behavior of the proposed system concept; optimizing the system design parameters; developing optimal control algorithms, both local and supervisory; testing control algorithms under various scenarios; and qualifying the production controller with a simulated version of the plant running in real time (hardware-in-the-loop testing), before connecting it to the real plant.




The *Engineering System Design Process* provides an environment that is rich with numerical and graphical analysis and design tools that stimulate innovation and cooperation within design teams.  It aims to reduce the risk of not meeting the functional requirements by enabling early and continuous verification throughout the entire design workflow.

The key concept in the courses is human-centered, model-based, multidisciplinary engineering problem solving.  The key emphasis, in both class and studio, is to strive to uncover the questions a student is asking himself/herself as he/she attempts to solve a problem and then give him/her the insight and understanding, based on physical principles and best industry practices, to ask the right questions.  This requires quality time in studio with small groups of students working interactively, as well as a focus on applications of content in class rather than presentation of content.  This can only happen if students prepare for both class and studio sessions.  To foster student preparation, voice-over power point slides were created and posted on the D2L (Discover to Learn) web site to accompany all detailed course and studio notes.

## Course Descriptions and Topics Covered

Electromechanical Engineering Systems

Model-based design: physical and mathematical modeling of electrical, mechanical, magnetic, and electromechanical systems. Dynamic analysis: time response and frequency response; analytical and numerical simulation. Electromechanical actuators: solenoid, vibration exciter, and brushed dc motor. Introduction to measurement systems: analog and digital; mechanical position / motion, electrical, and magnetic sensors. Electronics for actuators, sensors, and controls. Introduction to control systems: analog vs. digital, open-loop vs. closed-loop, stability, and performance. Introduction to On-Off and PID control. Industrial examples emphasizing integration. Studio exercises throughout the course using oscilloscope, function generator, power supply, multimeter, breadboard, and the Arduino microcontroller with electrical, mechanical, and electromechanical systems. Extensive use of MatLab / Simulink / Simscape / Real-Time Auto-Code Generation.

| Session | Topic |
|---|---|
| 1 | Digitization: Sampling, Aliasing, & Quantization |
| 2 | Electrical Systems 1: Resistors, Capacitors; Kirchhoff's Laws; Sources & Meters; Loading; Norton & Thevenin Circuits; Impedance Matching |
| 3 | Dynamic Response 1: 1st-Order Systems, Continuous and Discrete; Time and Frequency Response |
| 4 | Control Systems 1: Open-Loop, Feedforward, & Feedback Control; Stability & Disturbance Rejection; P & PI Control |
| 5 | Electrical Systems 2: Op-Amp Circuits, Comparator, Schmitt Trigger; Inductors, Diodes, Transistors; PWM |
| 6 | Dynamic Response 2: 2nd-Order Systems, Continuous and Discrete; Time and Frequency Response |
| 7 | Control Systems 2: Command Following; PID Control and Tuning; Digital Implementation |
| 8 | Measurement Systems: Performance Specifications; Signal Conditioning; Component Interconnection & Loading |
| 9 | Mechanical Systems 1: Mass, Compliance, Damping; One-DOF Rotational and Translational Systems; Gear Trains & Linkages; Newton & Energy Methods |
| 10 | Mechanical Systems 2: Two-DOF Rotational and Translational Systems; Mechanical / Electrical Analogy; D'Alembert's Method; Lagrange's Equation |
| 11 | Electro-Mechanical Systems 1: Laws of Faraday, Ampere, & Lenz; Magnetic Field, Circuit, & Magnetic Materials |
| 12 | Electro-Mechanical Systems 2: Magnetic Systems with Mechanical Motion; Solenoid, Dynamic Vibration Exciter |
| 13 | Magnetic Levitation System: Modeling, Analysis, Control; Microcontroller Implementation with PWM & MOSFET |
| 14 | Brushed DC Motors: Modeling, Analysis, Control; Analog & Digital Motion Sensors; Optical Encoders; Pulse-Width Modulation; Microcontroller Implementation with H-Bridge |

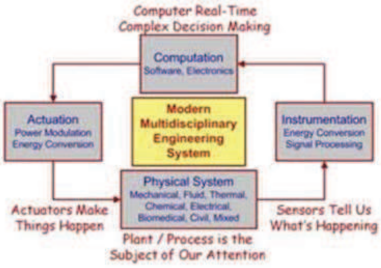
Multidisciplinary Engineering Systems

Physical and mathematical modeling of electromechanical, thermal, fluid, and multidisciplinary systems. Dynamic analysis: time response and frequency response; analytical and numerical simulation. Parasitic effects: compliance, backlash, friction, saturation. Control system design: root-locus and frequency-response methods. Electromechanical actuators: brushless dc motors and step motors. Modeling, analysis, and control of thermal systems and fluid power systems. Case studies: self-balancing transporter and H-Bot robot. Industrial case studies emphasizing integration. Studio exercises throughout the course using oscilloscope, function generator, power supply, multimeter, breadboard, and the Arduino microcontroller with electromechanical, thermal, and fluid systems. Extensive use of MatLab / Simulink / Simscape / Real-Time Auto-Code Generation.

| Session | Topic |
|---|---|
| 1 | Control Systems 3: Root Locus Analysis & Design; Case Studies |
| 2 | Control Systems 4: Frequency Response Analysis & Design; Case Studies |
| 3 | Control Systems 5: Advanced PID & Digital Control Design; Case Studies |
| 4 | Self-Balancing Transporter System: Modeling, Analysis, System ID, Sensor Fusion, Control, & Microcontroller Implementation |
| 5 | System Parasitic Effects: Compliance, Backlash, Coulomb Friction, Saturation, & Time Delay |
| 6 | H-Bot Robot: Trajectory Planning, Electronic Cams, Motor Selection, Modeling, Analysis, Control, & Microcontroller Implementation |
| 7 | Brushless DC Motors: Modeling, Analysis, Control, & Microcontroller Implementation |
| 8 | Stepper Motors: Modeling, Analysis, Control, & Microcontroller Implementation |
| 9 | Thermal Systems: Basic Equations; Temperature, Heat Flow, Capacitance, Resistance (Conduction, Convection, Radiation); Temperature Sensors |
| 10 | Fluid Systems 1: Fundamental Concepts & Fluid Properties; Basic Equations; Pressure, Flow Rate, Capacitance, Resistance, Inertance, Impedance; Flow & Pressure Sensors |
| 11 | Fluid Systems 2: Coupled Fluid Tanks: Modeling, Linearization, System ID, Experimental Verification, Control Design & Implementation |
| 12 | Fluid Power Control 1: Steady-State Characteristics of Circuit Components & Steady-State System Performance ; Valves, Servovalves, Pumps, & Motors |
| 13 | Fluid Power Control 2: System Dynamics; Dynamic Response of Common Components and Circuits; Hydromechanical Actuators |
| 14 | Fluid Power Control 3: Control Systems; Servovalve-Motor Speed Control; Servovalve-Linear Actuator Position Control |

**Examples of Studio Activities** (Read left to right, then down)

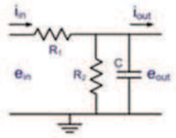
## Electromechanical Engineering Systems Mid-Semester Case Study




## Course Mid-Semester Case Study

- This integrating experience is to take an existing physical system (electrical first-order system here, but it could be any physical system) and have it meet desired dynamic performance specifications.

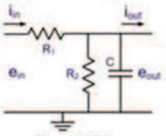
$t_{r_{10\%-90\%}} \leq 0.015$ sec
$M_p \leq 25\%$
$t_{s_{1\%}} \leq 0.09$ sec
Control Effort $\leq 13$ V
SS Error = 0




Electrical Physical System

**Process**

– Can the system alone (i.e., open loop) meet the performance specifications, i.e., unit step response with desired rise time, overshoot, and settling time?




R1 = 100 KΩ
R2 = 100 KΩ
C = 1 μF

Physical System

- Make simplifying assumptions (e.g., pure and ideal resistors and capacitor, no loading) and create the physical model.
- Apply KVL and KCL, along with component constitutive equations, to obtain the mathematical model.

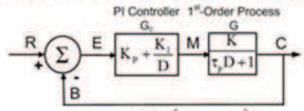
$$\frac{e_o}{e_i} = \frac{\frac{R_2}{R_1+R_2}}{\frac{R_1R_2}{R_1+R_2}CD+1} = \frac{K}{\tau_p D+1} = \frac{0.5}{0.05D+1}$$

1st-Order

– Analysis (use hand calculation or MatLab analysis) shows that the system time constant is too large and operating open-loop cannot meet the performance specifications. What to do?

– Change the physical system! Here we assume that the physical system cannot be changed.

– Apply closed-loop (feedback) control to obtain the desired response and use the PI (proportional-integral) controller.




90% of the controllers used in the world are PI Controllers!

$$\frac{C}{R} = \frac{G_cG}{1+G_cG} = \frac{\frac{KK_I}{\tau_p}\left(\frac{K_P}{K_I}D+1\right)}{D^2+\frac{KK_P+1}{\tau_p}D+\frac{KK_I}{\tau_p}} = \frac{\omega_n^2(\tau D+1)}{D^2+2\zeta\omega_n D+\omega_n^2}$$

– Comparison of the actual transfer function with the standard-form transfer function gives the following relationships:

$$\tau = \frac{K_P}{K_I} \qquad \omega_n^2 = \frac{KK_I}{\tau_p} \qquad \zeta = \frac{1+K_PK}{2\sqrt{\tau_p K_I K}}$$

– We now have relationships between the control gains, $K_p$ and $K_i$, and the dynamic performance indictors for a pure second-order dynamic system, $\omega_n$ and $\zeta$.

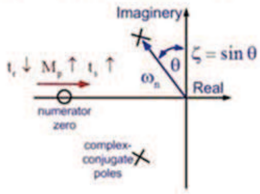
$$K_I = \frac{\tau_p\omega_n^2}{K} \qquad K_P = \frac{1}{K}\left[2\zeta\sqrt{\tau_p K_I K}-1\right]$$

$$t_{r_{10\%-90\%}} \approx \frac{1.8}{\omega_n} \qquad M_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \qquad t_{s_{1\%}} \approx \frac{4.6}{\zeta\omega_n}$$

– But Wait! This is not a pure 2nd-order dynamic system. There are numerator dynamics – the numerator has a 1st-order term! Not to worry – why?

– We know that for a pure 2nd-order dynamic system, with the damping ratio ζ between 0 and 1 (typical of most operating engineering systems), the roots of the differential-equation characteristic equation are complex conjugates (indicated by an **x**).

The numerator dynamics, i.e., the 1st-order term, has a root $-1/\tau$ called a zero, indicated with a **o**. As the zero moves along the real axis closer to the pole locations, system dynamic behavior is affected as shown. Take this effect into account in the design!




– Choose $\omega_n = 118$ and $\zeta = 0.64$ → $K_i = 1392$ and $K_p = 13.1$

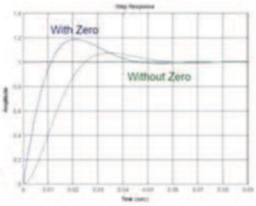
– The predicted performance values for a pure 2nd-order system are:

$$t_{r_{10\%-90\%}} \approx \frac{1.8}{\omega_n} = 0.015 \qquad M_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} = .073 \qquad t_{s_{1\%}} \approx \frac{4.6}{\zeta\omega_n} = .061$$

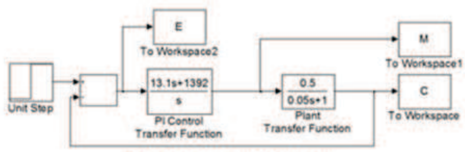
Note the effect of the zero: $t_r \downarrow$, $M_p \uparrow$, and $t_s \uparrow$

pole locations: -75.5 ± 90.7i
zero location: -106.3

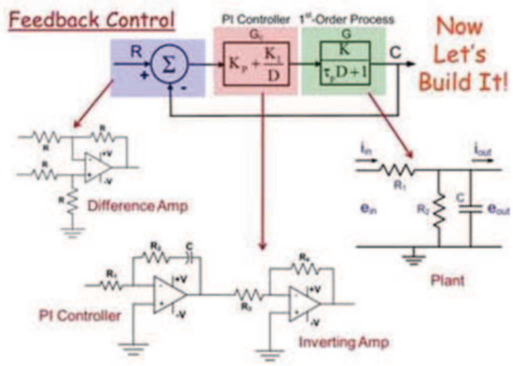
System Performance
$t_r$ = .0086 $M_p$ = .185 $t_s$ = .063

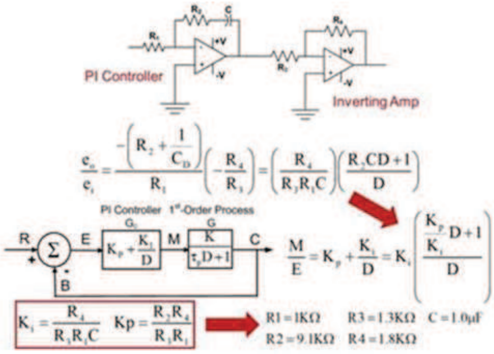


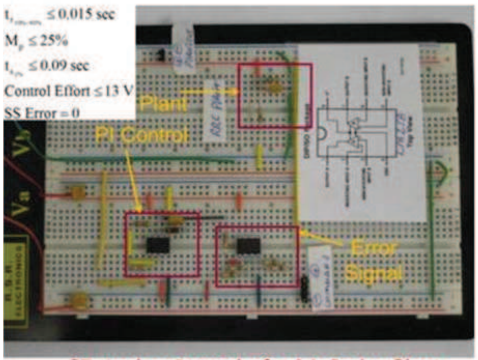

– Our design meets the performance specifications. But what about the control effort? We are going to first implement our design with analog op-amps and we know that the maximum output of an op-amp is about 13 volts when powered by ± 15 V. We use Simulink to check the control effort.




PI Control of a First-Order Plant
MatLab Simulink Block Diagram

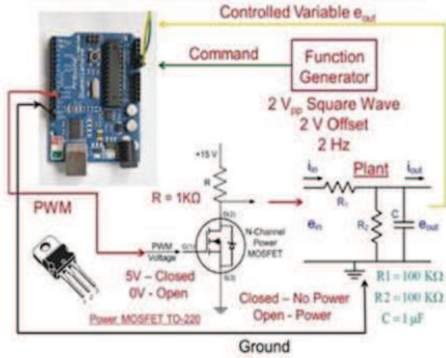


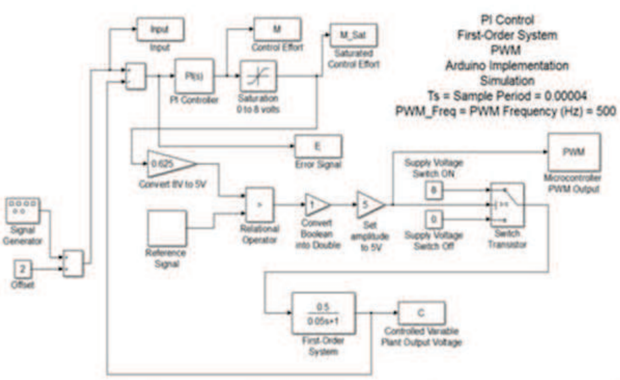




PI Analog Control of a 1st-Order Plant

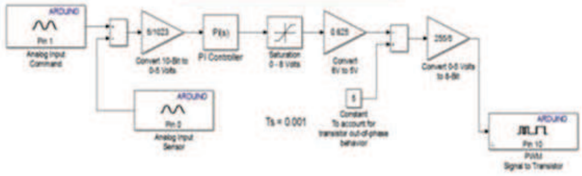
PI Control of a 1st-Order Electrical System



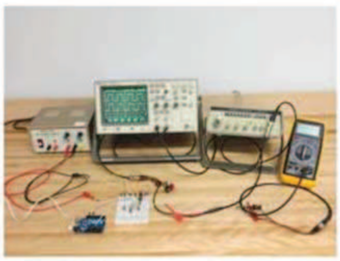

Simulink Simulation Block Diagram

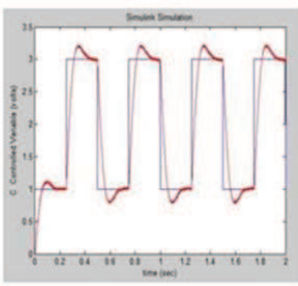



Real-Time Control Code

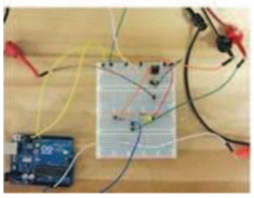


Arduino Microcontroller & Auto-Code Generation in Simulink






Predicted Response



Actual Response

The process works like this, with mathematics and physics learned and applied as needed. A physical engineering system (electrical, mechanical, or electromechanical) is chosen that must behave dynamically in a specified way. The system is first physically modeled with simplifying assumptions and then mathematically modeled by applying the laws of nature and appropriate component constitutive equations to the physical model. We start with a system whose model is first-order and study it from both time-domain and frequency-domain perspectives. Putting the mathematical model in a standard form (i.e., time constant, steady-state gain) allows an engineer to relate performance (e.g., speed of response, steady-state error, relative stability) to the hardware parameters in the physical model. As is often the case, the system cannot meet performance specifications operating open loop. A feedback control system is then designed and implemented. Closed-loop PI control of a $1^{st}$-order model results in a closed-loop differential equation that is second-order with a numerator zero. So $2^{nd}$-order dynamic systems are introduced naturally, as part of the process, along with the effect that a real zero has on ideal $2^{nd}$-order behavior. Again, time-domain and frequency-domain perspectives are emphasized. Once PI control gains are selected by a combination of pole-placement and simulation iteration, it is time to build the system. First an analog op-amp system is built with a difference amplifier and PI controller. Loading effects must be addressed, as must the limit on the control effort due to op-amp implementation. Measurements are compared to model predictions and model adjustments are made. Digital control with the Arduino microcontroller, inexpensive and open-source, is then used with the MatLab / Simulink Automatic Code Generation. Issues such as pulse-width modulation and low-pass filtering (which introduces a real pole), saturation, and A/D and D/A resolution all can be addressed in simulation and then easily in hardware implementation. Loading issues are again addressed with buffer op-amps. In this scenario, the students are “playing the game” from the start!

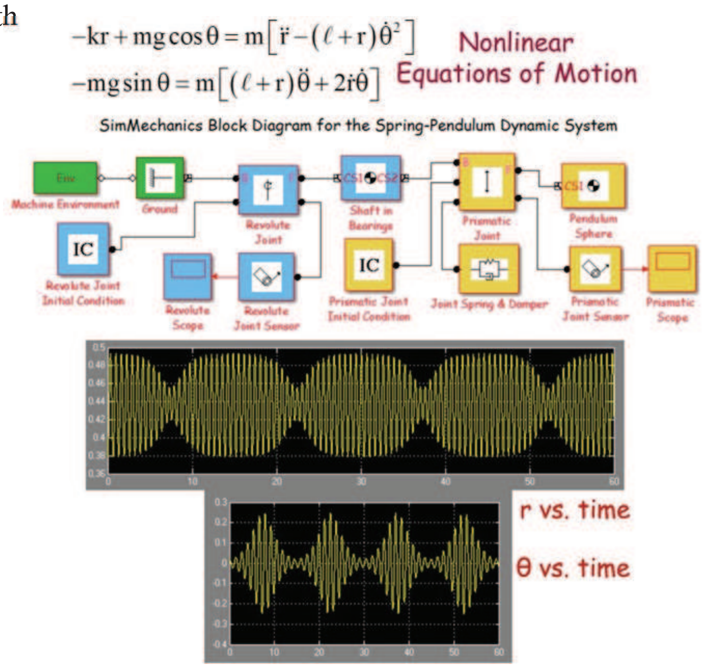
It’s rush hour in the city and you are searching for a parking spot. Once you find one, you will have a brief moment to pull up and back your car in. With automated parallel parking – no problem! – but with horns blaring, what is your back-up plan if the automated system fails? Could you take over and park the car flawlessly? Have you ever done that? Could you even do that now?

In engineering, we are depending more and more on computer tools to model and analyze the systems we conceive. If we do not use these tools correctly, or if the assumptions on which these tools are based do not match our needs, or if the tools have a bug, or …, could we take over and apply basic mathematics and physics to the problem at hand, at least to gain insight, if not the exact answer? Increasingly, the answer is no. We are becoming a profession of computer tool users and some boast that they are free from the unnecessary details of mathematics and physics. In this scenario, there is no back-up plan. You wouldn’t even attempt to park the car, and if you did, you would fail miserably. In engineering, there is much more at stake than simply losing a parking place.

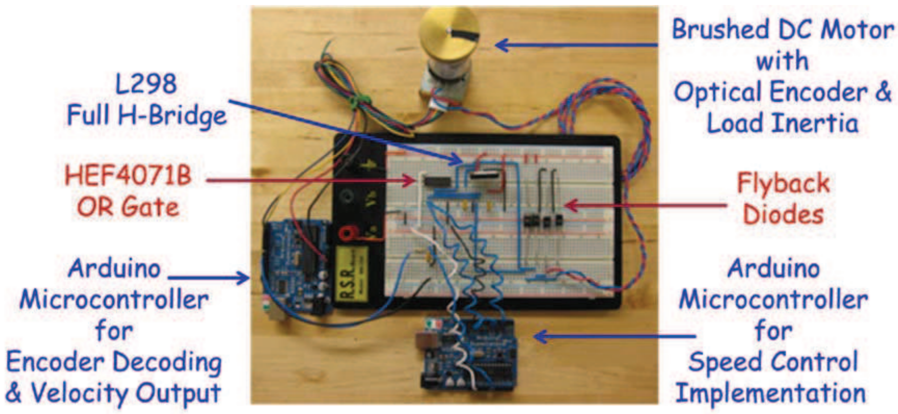
Here is an example of what I am talking about. The spring-pendulum dynamic system, shown and used in these courses, combines the two simplest mechanical dynamic systems there are:

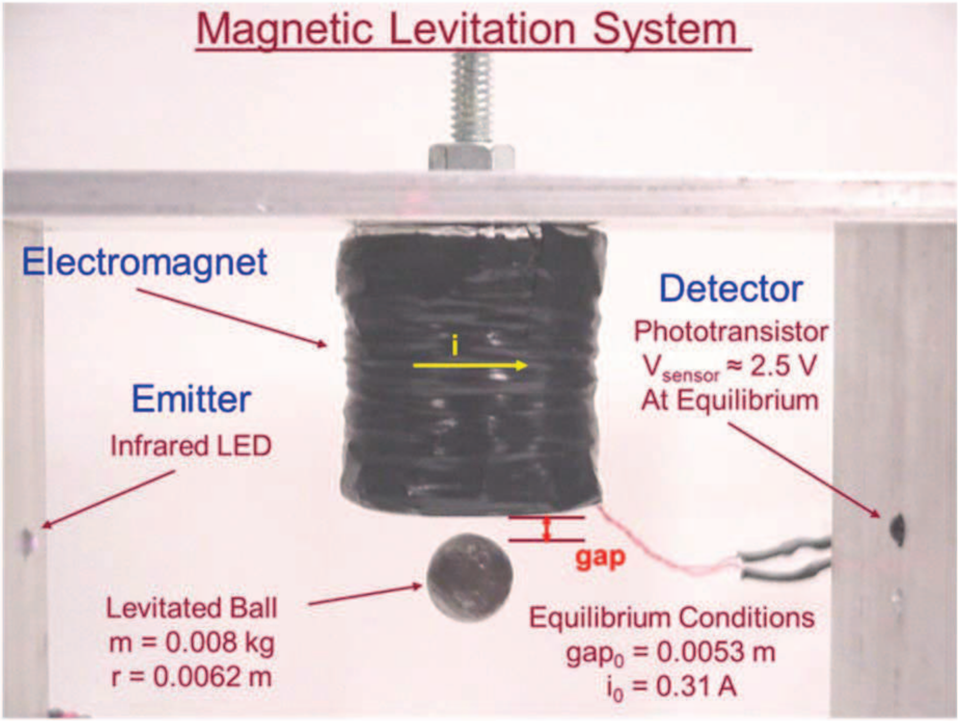
the simple pendulum and the simple spring-mass. The combination is anything but simple and the behavior is mystifying and unexpected.

There are two approaches to model and analyze this dynamic system. They are complimentary and both are essential. The first is to apply Newton's 2nd Law to a free-body diagram of the pendulum mass (neglecting friction and spring pretension), recognizing that the absolute acceleration of the pendulum mass has both radial components (radial and centripetal accelerations) and theta components (tangential and Coriolis accelerations). The resulting nonlinear equations of motion (shown and derived using simple particle dynamics) are coupled and predict a nonlinear resonance for specific values of the pendulum mass, spring length, and spring constant (as shown in the plots) that cannot be predicted by linearizing the equations of motion. This phenomenon occurs in many dynamical systems, e.g., satellites, ships, airplanes, buildings, and machines. The second approach is to use an icon-based modeling and simulation tool like SimMechanics from The MathWorks. The SimMechanics diagram consists of blocks representing rigid bodies and joints, along with blocks for sensors, initial conditions, and scopes. Both approaches predict the same model responses to various initial conditions. Which approach gives the most insight into this unexpected behavior? Which approach is easier to develop and implement? Which approach is easier to communicate? If the block diagram approach gives results that are unexpected, how does an engineer decide if the results are accurate and not the result of incorrect block parameters, or even an incorrect diagram structure? Only a balanced approach leads to complete understanding.



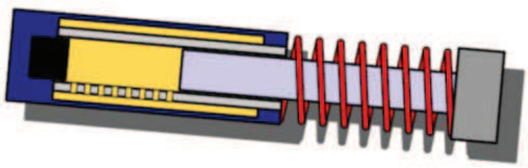

Along with the spring-pendulum dynamic system, the brushed dc motor and magnetic levitation system are systems used in the sophomore course, both with microcontroller control.





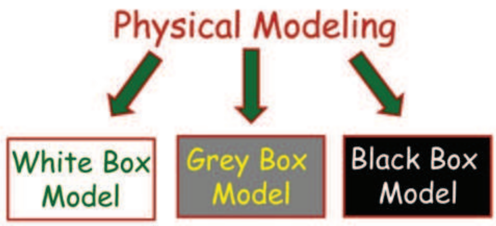

The fundamental concept in the two courses is model-based design, which must replace the design-build-test approach followed by so many companies still today.

In a world where problems are often ignored and allowed to fester for months or years, engineers do not have that option, as engineering problems ignored may lead to financial collapse or, worse, loss of life. Engineers solve problems to help people, and they do that with a sense of urgency. In many situations, a combination of human-centered design with state-of-the-art technology will yield feasible and sustainable solutions. In other more complex situations, physical insight may be incomplete and engineers perform experiments to validate what they do understand and inform what they don't. This approach to problem solving is called grey-box modeling and it existed long before the name was invented.



Let's use as an example the automobile / motorcycle shock absorber. Looking from the inside, it consists of a cylinder surrounding a movable piston. Moved by a shock from the outside, the piston compresses oil inside the cylinder through holes in the wall, thus dampening oscillations. A spring pushes the piston back to its original position, and a rubber stopper prevents the piston from impacting the walls of the cylinder when shocks are too strong. The shock absorber comprises the interaction of the mechanical movements of rigid bodies, the viscoeleastic dynamics of fluids, the elastic behavior of springs, and the deformations of elastic-plastic materials. Looking from the outside, we only are aware of the phenomenological properties. We observe aspects like nonlinear stiffness, nonlinear viscous damping at high frequencies, and hysteretic effects at low frequencies, but we are not able to assign these phenomena to the individual parts of the shock absorber.

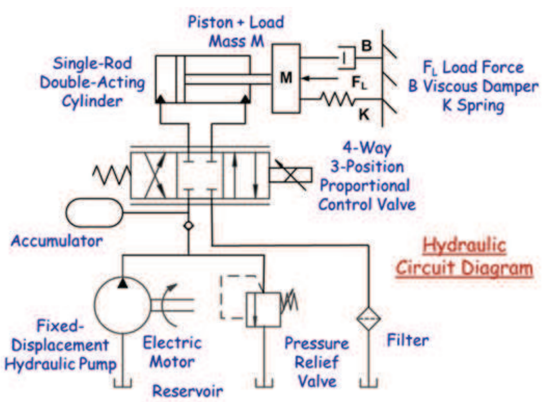
The shock absorber is integrated into a suspension system that must be designed and controlled. Mathematical equations are needed to predict the behavior of the shock absorber and the integrated suspension system. Therefore, a physical model of this physical system must be created and this model is based on simplifying assumptions. Depending on the nature of the simplifying assumptions, models of varying complexity and fidelity result. Information about the real system comes from two different sources: looking from the inside and looking from the outside. Looking from the inside, we apply the laws of nature, together with the constitutive equations of the components, to the physical model to generate the mathematical equations of motion. These are solved by numerical simulation to predict the behavior of the physical model, which must be experimentally verified. Because we use our insight and understanding of the way the system works to create the model, we call this model a *white-box model* and it is an approximate image of the physical system. Looking from the outside, measurements alone of the real system give no insight into the real system, and thus no understanding of how the real system works is brought into the construction of the model. A mathematical model is chosen which fits optimally the measured data. This type of model is called a *black-box model*.

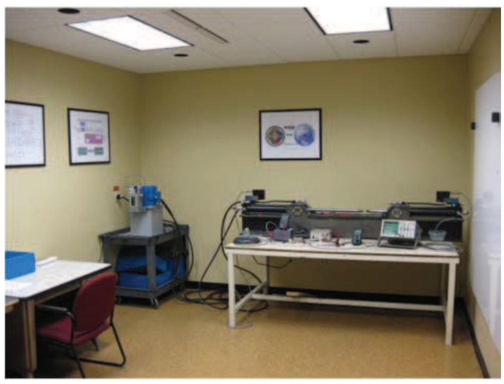



In reality, modeling is always something in between these two views, resulting in a *grey-box model*. White-box models are approximations of reality and always need experiments to identify parameters in the model, validate model predictions, and show where the model is deficient. The set of possible black-box models should always be guided by some knowledge of the phenomenological behavior of the real system. Since the focus of the engineer is solving the problem, the grey-box modeling approach is intuitive and obvious. Engineers contribute to society not only technological solutions to problems, but a solution process that transcends boundaries.

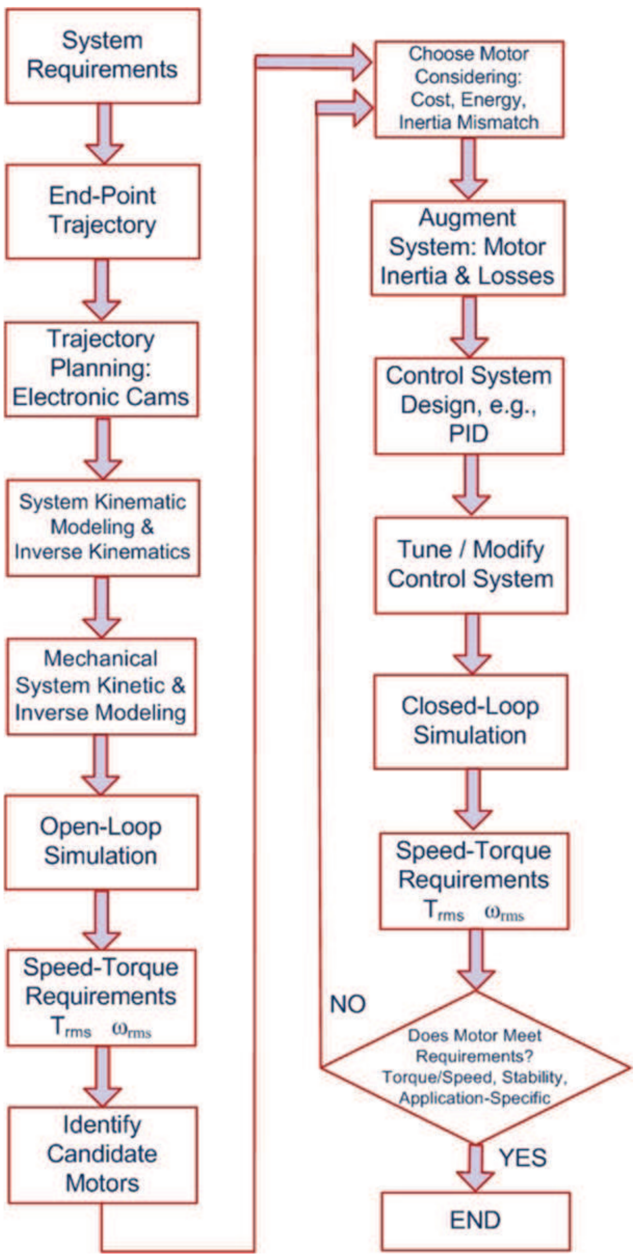
Multidisciplinary Engineering System course studio exercises include:

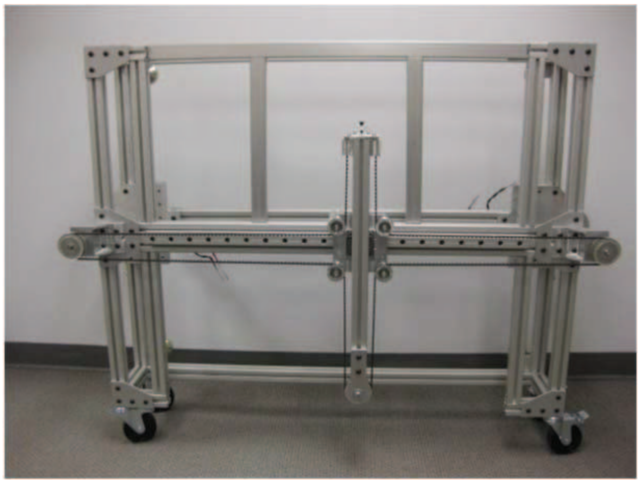
<u>Valve-Controlled Fluid Power Actuator System</u>

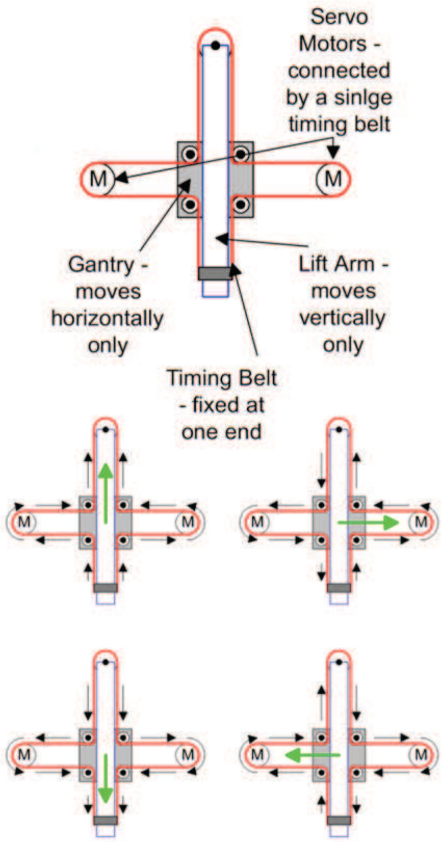


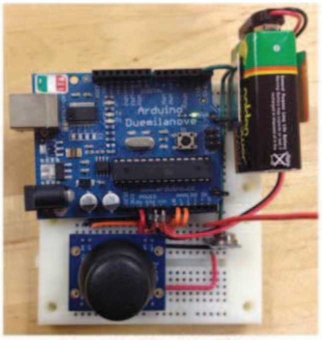
Model-Based Motor Selection for the H-Bot X-Y Positioning Robot



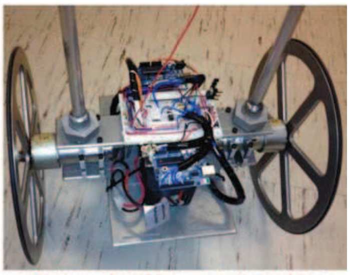





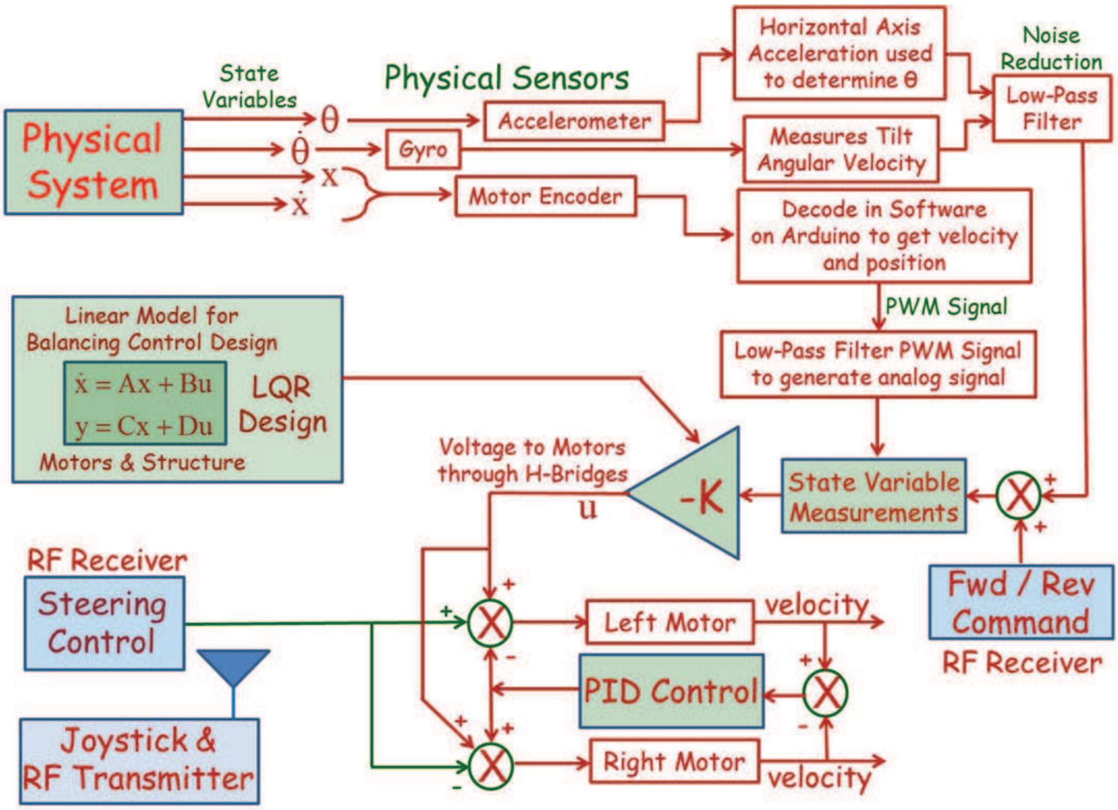
Self-Balancing Transporter



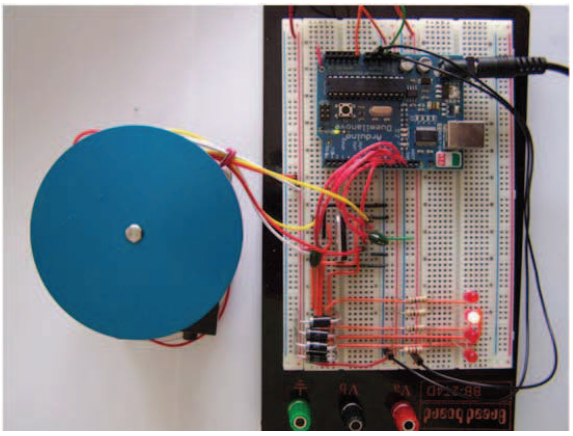
Remote Control for Forward & Backward Motion, and Steering



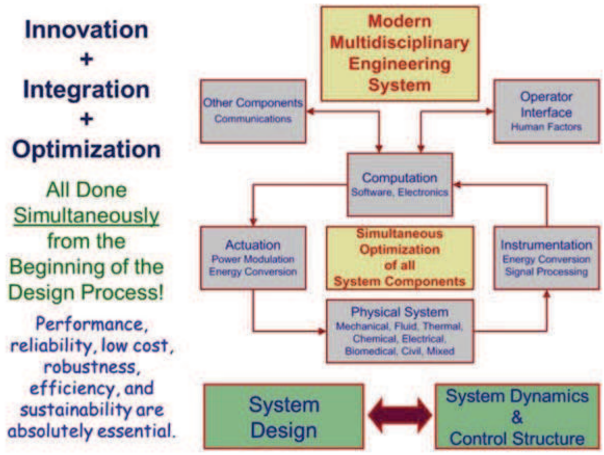
Power, Sensing, and Control Electronics; Motors and Structure



Step Motor Microprocessor Control



The focus in all systems studied is on integration and simultaneous optimization of all system components from the start of the design process.




**Conclusions**

Engineering silos and engineer comfort zones, both in industry and academia, are the two biggest obstacles to innovation. Engineering education taught as separate subjects and not in the context of real-world problem solving that requires integration from the start of the design process is another obstacle to innovation. The way we now teach engineering gives our students no chance to “graduate and hit the ground running.” Hopefully this is a start to addressing this urgent need.

Human-Centered, Model-Based Design
Is The Key To
Modern Engineering Practice!

Critical-Thinking Problem Solvers
Follow A Process
– A Model-Based Design Process –
That Guarantees Success!

Engineering Education Needs to be Rebundled
& Delivered In The Context of
Real-World Problem Solving
With NO Silos and No Comfort Zones