

AC 2007-953: DEVELOPMENT OF MODEL BASED DESIGN CURRICULUM

sameer prabhu, The MathWorks

Sameer M. Prabhu, Sr. Applications Engineering Team Leader, The MathWorks, Inc., Novi, MI 48375. Ph. (248) 496-7944, FAX: (248) 596-7959, Sameer.Prabhu@mathworks.com. Sameer has over ten years of experience applying The MathWorks' products in various application areas. As a Principal Applications Engineer in the Detroit, Michigan office, Sameer manages a team of applications engineers focused on working with customers in the automotive and commercial vehicles industry to address the systems integration challenges posed by increased adoption of electronics in these industries. Prior to joining The MathWorks, Sameer engaged in the R&D of complex control systems through his work at Visteon, Caterpillar, and Tata Motors. The results of his work have been documented through publications and United States patents. Sameer graduated from University of Bombay with Bachelors in Mechanical Engineering and received his Ph.D. in Mechanical Engineering from Duke University in the area of robotic controls and artificial intelligence. He also holds an MBA from the University of Michigan

Liz Callanan, The MathWorks

Liz Callanan, Corporate Relations Director, The MathWorks, Inc., Natick, MA 01760. Ph. (508)647-7417 FAX: (508) 647-7001. Liz.Callanan@mathworks.com

Liz designs and manages the community relations at The MathWorks with strong ties to academia and world-wide research programs. Since joining The MathWorks in 1991, she has directed entrepreneurial programs in engineering education, business development, corporate communications, and community relations. She has forged partnerships to realize The MathWorks goal of "advancing the pace of discovery in science and engineering" via launching a student-priced MATLAB and Simulink, establishing a software product development program with engineering thought leaders, introducing third-party book and product programs, and managing student competitions such as "Challenge X: Crossover to Sustainability." Prior to joining The MathWorks, Liz worked at a Massachusetts Institute of Technology spin-off software company and managed HR for Saga Holidays, Ltd. of the UK.

She serves on advisory boards of Sustainable Business Network, Dana-Farber Cancer Institute, and National Public Radio. Liz holds a B.A. from Wellesley College.

Zachariah Chambers, Rose-Hulman Institute of Technology

Zachariah Chambers, Associate Professor, Rose-Hulman Institute of Technology, Terre Haute, IN, 47803, Ph. (812) 877-8904, FAX: (812) 877-8895 chambez@rose-hulman.edu. Zachariah Chambers is an Associate Professor of Mechanical Engineering at Rose-Hulman Institute of Technology where he earned his mechanical engineering BS and MS in 1994 and 1996 respectively. He obtained his Ph.D. in engineering science and mechanics in the field of Computational Fluid Dynamics. He teaches primarily on the freshman and sophomore level and serves as faculty co-advisor for the Challenge X : Crossover to Sustainable Mobility hybrid SUV competition.

Marc Herniter, Rose-Hulman Institute of Technology

Marc E. Herniter, Associate Professor, Rose-Hulman Institute of Technology, Terre Haute, IN, 47803. Ph. (812) 877-8512, FAX: (812) 877-8895, Marc.Herniter@ieee.org. Marc Herniter is an Associate Professor at Rose-Hulman Institute of Technology (Ph.D., Electrical Engineering, University of Michigan, Ann Arbor, 1989); Dr. Herniter's primary research interests are in the fields of power electronics, electric vehicles, and alternative energy systems. He has worked on power electronic systems that range in power levels from 1500 W to 200 KW. He is the author of several textbooks on circuit simulation, MATLAB, and computer usage. He joined the faculty of the ECE department at Rose-Hulman Institute of Technology in August 2001. Previously, he was

on the faculty of Northern Arizona University for 11 years.

Development of Model-Based Design Curriculum

Abstract

Model-Based Design is increasingly prevalent in industrial sectors including aerospace and automotive, but lacking from college and university curricula. The need for students to be adept at the modeling of systems, their associated subsystems, and overall system controller was the impetus for The MathWorks to partner with Rose-Hulman Institute of Technology to “bridge the gap” between industry and academia by developing a Model-Based Design course within the context of a series hybrid-electric vehicle. Students apply Model-Based Design, beginning with simple component models for a complex, multi-disciplinary system in need of control. Incremental improvements to the components and controller enable students to assess the impact of their changes and determine if the associated level of complexity is warranted. Moreover, the deployment of “executable specifications” enables students to identify and rectify errors early in the design process, whereby they may save dramatically on time and cost. Assessment data are being collected to qualify the academic impact of Model-Based Design.

Introduction

Given competitive pressures, and time and cost constraints, developing a product on time and within budget requires a systematic approach to design and realization. The systematic design and realization process in the aerospace and automotive industries is typically represented by a V diagram as shown in Figure 1.

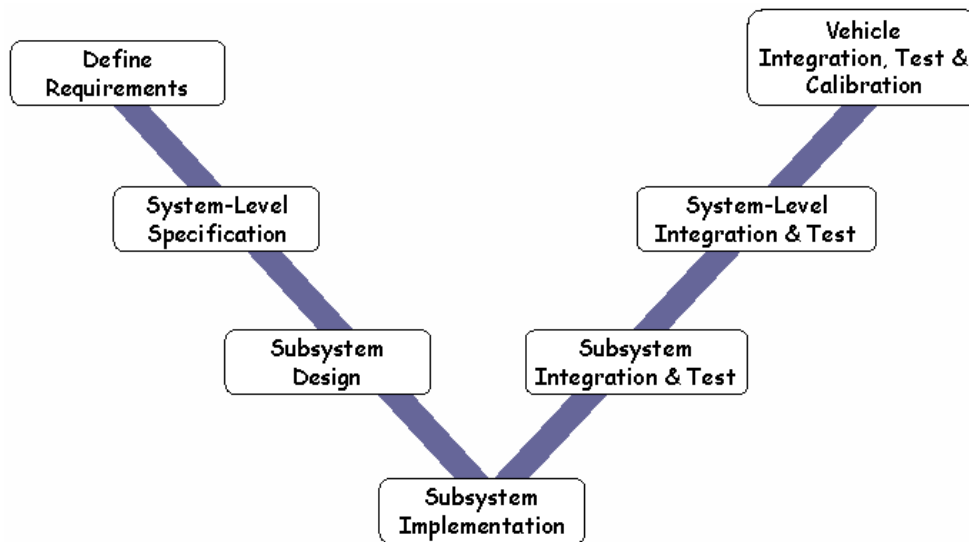


Figure 1: A V Diagram of the System Design and Realization Process

Each of the two branches of the V correspond to distinctly different activities:

1. The left branch captures the decomposition of the initial system requirements into subsystems and components that are specified and implemented at a detailed level.
2. The right branch represents the realization of these subsystems and components and their integration.

In a traditional product development approach, engineering teams observe strict boundaries between the various design activities in the process, and they communicate by passing design documents back and forth. This approach has several drawbacks such as difficulty in keeping the documents synchronized with the current state-of-art of the design, errors associated with recording and understanding the design functionality in the form of a document, and the need to manually code the application. Also, when documents are used as deliverables and shared electronically, engineers often duplicate efforts and it is difficult to trace the source of errors along a paper trail.

Engineering teams in the automotive, aerospace, and other industries have turned to Model-Based Design to address these problems.¹ Model-Based Design enables them to effectively manage increasing product complexity, more stringent performance requirements, and shorter product development cycles. By using models in the early design stages, engineers can create what are known as "executable specifications" that enable them to immediately validate and verify specifications against the requirements. Thus Model-Based Design allows engineers to detect errors earlier when the cost to fix them is less.

Through the use of automatic code generation technology and hardware-in-the-loop testing, Model-Based Design alleviates errors introduced during manual implementation and realization tasks and shortens the path to product delivery by generating code for testing, calibration, and final production.

Because students do not have the opportunity to work on large systems (e.g., a complete automobile or airplane) in a typical classroom setting, Model-Based Design is not formally taught in the conventional undergraduate curriculum. Students are indeed exposed to building models as part of a typical control design curriculum; however, the emphasis of such courses is not on teaching the rationale behind and fundamentals of adopting Model-Based Design and applying it to engineering systems. Students learn about pieces of subsystem development, focusing on specific bits of the whole without appreciating the impact a piece or subsystem can have on the total system.

To date, academics have preferred to introduce an understanding and mastery of the "building blocks" before putting them into context. As a result, engineering graduates entering the workforce must negotiate a significant learning curve in terms of mastering and applying Model-Based Design, which is increasingly prevalent in industry. In order to help ease the transition for engineering graduates and bridge the gap between companies industry and educational institutions, it is necessary to introduce students and faculty to Model-Based Design.

The MathWorks is leading Model-Based Design adoption at universities through an array of student competitions that mirror some several core engineering industries: automotive, aerospace, communications, and biotechnology. The aim is to introduce a holistic view of full-

scale models in order to arrive at a complete picture of the system and the interrelated component parts. While today Model-Based Design is taught mainly in industry to address real large-scale problems, industry users who have academic ties as advisors and professors see the value of introducing Model-Based Design in the classroom.

By proactively working with student competitions and related curricula at forward-thinking universities, The MathWorks will help foster the incorporation of Model-Based Design into university curricula. This strategy enables academics, using industry partnerships, to leverage existing industry models and the knowledge of practicing industry professionals. The MathWorks sponsors several student competitions including Challenge X, the Smart Radio Challenge, AIAA student competitions, SAE Student Competitions, DARPA Urban Challenge, MIT Vehicle Design Summit, and the international Genetically Engineered Machine competition (iGEM). During the course of these competitions, The MathWorks conducts workshops to teach Model-Based Design to both students and faculty, who gain the opportunity to apply these concepts throughout their work to achieve prescribed goals.

MathWorks workshops have resulted in the development of a Model-Based Design course at Rose-Hulman Institute of Technology. The course covers the implementation of Model-Based Design for the development of a hybrid-electric vehicle; however, the method could be applied to the development of any large system. The course covers developing models for individual components, connecting the component models together to form a system, controlling the system, simulating the system, using automatic code generation to run real-time simulations, and then deploying the controller for the system on a production target. The goal of the course is to take a system from conception to realization using the tools and methods presently used by industry.

A hybrid electric vehicle is an ideal system for this course because it is inherently interdisciplinary, requiring mechanical, electrical, and computer engineers to work in unison. A typical hybrid electric vehicle would have an internal combustion engine, electric motors, a battery, and a supervisory controller. The Model-Based Design course allows students to see the relationship between all of the components as well as demonstrate a method for developing a large system.

In this paper we discuss the application of Model-Based Design for embedded systems design and present examples from the automotive and aerospace industries. We then present a model for a course in Model-Based Design, and discuss how to measure its effectiveness in teaching system design and implementation.

Introduction to Model-Based Design

The need to bring innovative, high-quality products to market faster is driving the increased use of models during the design and realization process. The ability to model and simulate systems plays a key role in the product development process by allowing manufacturers to test whether the system meets requirements using virtual prototypes rather than physical prototypes. In the early design stages, the models become “executable specifications,” i.e., the specifications can be immediately verified against the requirements. Models can also be used to communicate between

engineering teams with different specializations, allowing them to collaborate during all stages of the design process. Moreover, initial design models can be incrementally extended to include increasing implementation detail. By using such virtual prototypes in the form of models, designers can explore multiple design alternatives to optimize the design and discover errors in the system early on. Even more sophisticated is the use of model transformation to generate different representations of the same system, which further minimizes the effort to move from one design stage to another. In particular, the use of automatic code generation technology and hardware-in-the-loop testing alleviates errors introduced during manual implementation and realization tasks and shortens the path to product delivery by generating code for testing, calibration, and final production. An important benefit of Model-Based Design is the traceability of design decisions all the way down to the implementation. So test results can be directly interpreted as high-level design decisions.

Thus, Model-Based Design provides efficiencies in product development that enable companies to deliver products on time, remain within budget, and fulfill initial requirements. As a result, Model-Based Design is an integral part of the systems development process in automotive, aerospace, and other industries. Model-Based Design covers a broad spectrum of applications and industries, a few of which are highlighted in the next section.

Model-Based Design in Industry

A big challenge in the automotive industry is to develop powertrain solutions that meet restrictive emissions and increased fuel economy standards. This requires innovative thinking to develop a solution, and the first to market also gains a significant competitive advantage. When faced with such a challenge, Nissan replaced their paper-based design process with Model-Based Design and developed the first gasoline vehicle to meet the PZEV (Partial Zero Emission Vehicle) standard². By building models of both the engine and the emission control strategy, Nissan was able to validate their designs and refine the control strategies. They also reduced the number of sensors required and achieved a significant cost savings. Further, the ability to simulate and validate their designs early allowed them to reduce development time by 50%.

Another significant challenge in the automotive industry is dealing with the increased number of electronic control units (ECUs) on a typical car and testing these systems effectively to ensure that the entire networked car works. Vehicle Systems Integration worked with the U.S. Department of Transportation (DOT) to create a hardware-in-loop simulator that integrates all the ECUs for a heavy truck and tests them using dynamic models of the vehicle in real-time. With this approach, they avoided costly test track runs and also were able to test safety-critical systems that are difficult to test in real-world conditions³. By using this testing environment, VSI was able to test various system configurations and operating scenarios by simply modifying the models. As a result, they were able to detect system integration problems early when the cost to fix them was less.

In the aerospace industry, designing increasingly sophisticated avionics and flight controls is further complicated by developments such as unmanned aerial vehicles (UAVs) which adds autonomous mode of operation to these already complex systems. Further, since development of these vehicles is often carried out by geographically distributed teams, coordination and teamwork is crucial. Engineers at BAE Systems used a common design tool to build models of

both the controller and the complex dynamics of the vehicle and its environment. This enabled them to test their ideas using simulations, which in turn reduced the testing cycle time and minimized errors and manual documentation work since the embedded software and documentation were automatically generated from the same models.⁴ Model-Based Design is affecting positive change in other industries as well.⁵

Training Industry Professionals in Model-Based Design

As discussed, the automotive industry is at the forefront of Model-Based Design. DENSO, a key supplier of automotive systems to the world's major carmakers, relies heavily on Model-Based Design to reduce development time and improve software quality. They believe it is not enough to develop advanced technology and tools; it is equally important to develop the infrastructure and skills to support the technology.⁶ As part of this effort, they have developed a rigorous training program for Model-Based Design with a track for managers to get familiar with the concepts, and another track for engineers who focus on building models. DENSO is also building an infrastructure for Model-Based Design. Since models cut across teams, they have developed internal tools to manage multiple versions of the models. DENSO cited the ability to build human skills and infrastructure simultaneously with technical issues as critical success factors for their Model-Based Design efforts.

Throughout industry, models are increasingly used as part of the design and realization process, however, the commitment to training engineers on Model-Based Design varies significantly. Training ranges from "on-the-job" learning to formal programs such as the one at DENSO. In some organizations, training related to Model-Based Design involves offering classes on control systems, since Model-Based Design has its origins in complex control problems. Another common training approach is to fund classes and seminars offered through professional institutes, such as the Hydraulics Systems Modeling and Simulation class offered by Milwaukee School of Engineering.⁷

Still other organizations in aerospace, automotive, and other industries offer formal training in systems engineering and thus provide a good initiation for engineers entering into a systems engineer role. The International Council on Systems Engineering (INCOSE) maintains a comprehensive listing of available training and certification classes.⁸ Similarly, other organizations are building skills in the mechatronics area due to the growing trend multidomain system design. That is, systems that were once purely mechanical may now have electronic, hydraulic, and control elements. Some companies have focused on teaching modeling focused on mechatronics, typically by allying themselves with a university that offers formal curriculum in this area.

As can be seen, there is a variety of approaches practiced in the industry to train engineers in Model-Based Design. While some large companies can allocate the time and resources to offer formal training, a vast majority of engineers do not have access to such resources. Therefore, it would be beneficial to industry if academia focused on training engineering students on Model-Based Design so that graduating engineers can minimize the ramp-up time when they begin their professional careers.

Bridging the Gap Between Industry and Academia

The Approach

Since 2000, The MathWorks, a leading developer of software for technical computing and Model-Based Design for engineers and scientists in industry, government, and education has sponsored local, national, and international student competitions with funding, software donations, and coaching. This is one proposed means to effecting curricula change. The most intriguing partnerships have arisen in student competitions with industry co-sponsors who have already adopted Model-Based Design in their own development and business practices. Simply, they can no longer envision using “build and test” methods that pre-date Model-Based Design. By partnering in student competitions with customers, collaborators, and fellow sponsors, The MathWorks can integrate tools at a deeper level while also establishing a more pervasive presence. Simultaneously, the rigorous test of Model-Based Design in a competition setting is ideal for investigating new ways of presenting the material. Students and professors are motivated to optimize the use of the tools provided.

The corollary to industry use of Model-Based Design is the demand for technically prepared students who will join the workforce to become Model-Based Design customers, designers, and possibly teachers. The aim is to carefully build on past success of student competitions in order to realize new opportunities in burgeoning engineering fields worldwide – systems biology, communications, industrial automation, aerospace, automotive, among them. Customized curriculum for Model-Based Design that is grounded in the partnerships of industry and academia will enhance the understanding of engineering concepts and appreciation of the big picture impact that is already accepted “in practice.”

The MathWorks recognizes the future power of Model-Based Design in academia. To both prepare students for future careers and to enhance their understanding of the entire design process from the whole to the tiniest detail, the company is committed to introducing Model-Based Design in a variety of engineering departments. Many universities already use The MathWorks software tools; they simply lack training materials / curriculum or their instructors are unfamiliar with Model-Based Design. To move the work forward, The MathWorks is focused on thought leaders in a variety of subjects in academia and industry to gain input on the best means of efficiently reaching out to students via their professors.

To help promote the use of Model-Based Design in an academic setting, The MathWorks has developed the following resources:

- A robust book program with more than 1,000 titles based on the core tools of Model-Based Design: MATLAB[®] and Simulink[®]
- A series of Model-Based Design training modules focused on the automotive industry [see below] from Rose-Hulman Institute of Technology professors
- In-person and online training courses in MATLAB and Simulink
- Online curriculum sharing by a variety of universities
- A host of compelling industry examples
- Experienced mentors
- Industry experts on staff

Why is This Approach Valid?

Academia is often credited with great flexibility to explore new concepts. This may be true in research matters, but regarding curriculum changes in many university settings it is not the case. An exception is found with the professors who lead student teams to achieve goals in industry funded competitions; who seek out the most innovative methods available to support their teams' objectives. They want to provide the most useful tools for their teams and the best training materials for the related courses they teach. In the past six years, The MathWorks has been impressed by the creativity and hard work these student advisors pour into the experience. The MathWorks also recognized that a groundswell of enthusiasm for a new curriculum must come from industry demands and motivated professors who can both introduce new concepts and make adjustments, if not wholesale changes, to curriculum.

How Will This Approach Be Implemented?

Student competitions will be incubators for this experiment in Model-Based Design curriculum development. The MathWorks has found that having motivated students working on complex problems with the guidance of faculty and industry advisors provides an ideal opportunity to test the veracity of a new training concept.

Today, Model-Based Design is taught on a limited basis at the university level by teachers at progressive universities or those undergraduate and graduate programs that are aligned with industry offerings (internship programs) to prepare graduates for the workforce. More often, it is in a work environment that students become acquainted with the value of the approach. Student competitions can provide that first glimpse of the work environment in a controlled, time-bound setting to experiment with the success of new curriculum materials across a number of subjects (university teams.)

The MathWorks is committed to Challenge X,⁹ a four-year student competition for 17 North American universities, tasked with optimizing the technical design of an already fuel-efficient vehicle. This team works in partnership with the Department of Energy, General Motors, and more than 15 other industry partners. The MathWorks is also dedicated to Smart Radio Challenge, which poses four challenging technical questions to eight carefully selected teams from around the world. The selected co-headline sponsor is Texas Instruments. Other competitions include those in automotive, systems biology, and aerospace with invested industry collaborators.

With strong supporters from industry already in place as joint mentors in advising teams to select resources that will yield the best results, Model-Based Design stands to gain greater acceptance by university professors and students.

Model-Based Design in the Curriculum

Model-Based Design is typically used to describe large systems with complex interactions between components, and the standard curriculum approach does not lend itself to tackling such

big projects in one class. In Model-Based Design, highly simplified models of the subsystems are developed and the subsystems are connected together to form a more complicated overall system. Once the input/output characteristics of the subsystems are finalized, different engineers can work independently at different levels on the model. Some engineers can work at the subsystem level and develop accurate models of the subsystem while other engineers can work at the top level and refine the model at the system level. Using this method, accurate models of the component subsystems and overall system can be developed concurrently.

Educational institutions, on the other hand, typically use a bottom-up approach. Students take individual courses on circuits, dynamics, heat transfer, microprocessors, and so on. This is analogous to focusing on the physics and mathematics of the subsystems and never considering how the subsystems interact to form a system. Inevitably, the question “what would we use this for?” arises. The question is usually answered by saying, “well, if we were building this system, and we had this particular problem, we could stick this thing in right here.” Typically, most projects engineers work on part of a larger system. A good example would be an operational amplifier with a gain of one (typically referred to as a buffer where the output equals the input). When building an analog system, buffers are sprinkled liberally throughout a system since they are small and allow an analog system to be broken into several small systems with simplified functions. In the context of a system, the explanation of the usefulness and purpose of a buffer is rather simple. However, when talking about a buffer in the context of a class that covers operational-amplifier circuits, a circuit where the output is the same as the input does not appear to be very useful. The use Model-Based Design in the curriculum would allow educators to focus on the physics of the subsystems, study how the subsystems interact to form a useful system that solves a problem, and answer the questions of why the subsystems are used the first place.

Schools come close to teaching Model-Based Design in control courses and in some design courses where the process of design is taught. In a typical control course, students design feedback controllers for a black box plant model. Students working on the controller do not need to know how the plant is implemented; rather they need an understanding of how the plant will react to specified inputs, and they must understand the input and output signals available or required. Typically, the characteristics of the plant are given and students design the controller while knowledge of the implementation or utility of the plant is not necessary. A control course of this type can be thought of as Model-Based Design on a small scale because the plant and controller could be designed separately as long as the interface between the plant and controller is specified. In a typical undergraduate-level course, the plant is a linear system with one input and one output, so the interface is not very complicated. Because the interface is usually simple, interface specifications such as rate limits or upper and lower bounds on the input and output signals may not be emphasized. Furthermore, the plant is usually given, and students usually do not design the plant model. Thus, students focus on the controller and fail to see the controller as one part of a larger system. Model-Based Design allows students to work on all parts of the system concurrently, including the controller, plant, and interface between the two. By not emphasizing the interface, the concept of concurrent subsystem, overall system, and controller followed by ready system integration is lost.

Another area where the principals of Model-Based Design can be applied is design courses where large teams of students work on a multidisciplinary project for an entire term. An example

would be the EGR286 course at Northern Arizona University.^{10,11} In this course, a large robotic project is split into three distinct subsystems. In the initial design phase, students must specify the function of each subsystem as well as the interface between subsystems. In courses of this type, the design and interface specifications are done on paper, and simulations may not be performed. Model-Based Design would fit well into a course of this type because the models could be used to specify the subsystem and interface requirements, as well as verify system operation before the build phase begins.

Model-Based Design Course at Rose-Hulman Institute of Technology

Rose-Hulman Institute of Technology has introduced a Model-Based Design course as an elective and as part of the Challenge X: Crossover to Sustainable Mobility university competition.⁹ In this course several aspects of Model-Based Design are covered and applied to the design of a hybrid-electric vehicle. The course is a synthesis of several different areas in the design process including modeling, component physics, system design, controller design, and industry-standard software packages.

Modeling

Modeling is covered in several different courses throughout the curriculum, but students do not really understand what is meant by a “model.” For example, a MOSFET operating in the saturation region obeys the current-voltage relationship $I_D = K(V_{GS} - V_T)^2$ where the current I_D is completely determined by the controlling voltage V_{GS} , and K and V_T are constants. Students do not interpret this relationship as a model that has limitations and breaks down under certain conditions. Rather, students interpret this relationship as “this is how the device works,” or as a “law.” They do not interpret the relationship as a model that matches up with empirical observations under certain conditions. Students may also miss the idea that the models can be incrementally improved to make them more accurate. With the MOSFET model, channel-length modulation $I_D = K(V_{GS} - V_T)^2(1 + \lambda V_{DS})$ could be added. This is a second order effect where the current is slightly influenced by the drain-to-source voltage V_{DS} . Typically, λ is small so it is not always included. Students may miss the point that $I_D = K(V_{GS} - V_T)^2(1 + \lambda V_{DS})$ is a better mathematical model of the MOSFET than $I_D = K(V_{GS} - V_T)^2$. Because these devices are fairly abstract and because students have so much material coming at them from so many different courses very quickly, students may tend to regard these equations as “laws,” or “the way things are,” rather than just models that apply under certain constraints.

Rose-Hulman Institute of Technology’s Model-Based Design course discusses subsystems with which students are familiar such as electric motors, batteries, and internal combustion engines. Mathematical models for these components are developed so that students can realize that they are just models that describe how the components work. The initial models are so simple, either rudimentary first-principle models or look-up tables, that students understand that the models only describe the input-output behavior of the component, and are not detailed descriptions of the inner workings of the device. Initially, the simplest possible model for a component is developed and then more detail is slowly added as the limitations of the model become apparent.

A good example of this process is the model for the electric motor. The initial model implements a power balance where the electrical power is equal to the mechanical power. Essentially, the motor is described as a 100% efficient electromechanical energy conversion device with no rpm limitations and infinite power capabilities. Students realize that this is a “model” because they know that electric motors do not have infinite rpm and power capabilities; rather they use electric and magnetic fields to produce movement and these fields are not described in the model. However, under certain limitations, the model does describe the behavior of the component. (In the world of power electronics, this model would be referred to as an average model.) Although this model seems overly simple, it must be understood that the motor model is used in the context of a complicated system that models a hybrid-electric vehicle. Even though the model is simple when viewed as an individual component, its role in the overall system is quite complicated. Thus, from a system engineering point of view, starting with a simple model is preferable and necessary.

As students become more familiar with the operation of the motor model in the context of the hybrid-electric vehicle system, students realize that the model is too ideal, and the component behavior is not accurate in some instances. At this point, component models are incrementally improved to make them more realistic. Improvements to the motor model include adding a torque curve, efficiency less than 100%, and current limits. Improvements are added to components incrementally to allow students to understand the component’s role in the system and to understand how the change just made affects the component and the system.

The concept of modeling is emphasized by slowly improving the component models. When using the ideal motor, which does not have a torque curve, the vehicle can follow a drive cycle and implement regenerative braking. These are big steps when building a hybrid-electric vehicle model and it provides the students a great deal of insight into how the system components interact. However, further analysis shows that the ideal motor model allows the vehicle to do things that students understand are not realistic. Examples are the vehicle accelerating to extremely high speeds and the motors drawing extremely large currents. Students realize these inconsistencies and trace them back to the ideal motor model. When a torque curve and current limits are added to the motor model, the vehicle performs in a more realistic fashion.

Improving the motor model is just one step in the model improvement process. After the motor is improved, students realize other inconsistencies caused by other model components, and those models are also improved. This process of continually observing the model performance, identifying model inconsistencies, identifying the component model that caused the limitation, and then improving the model to fix the inconsistency emphasizes the concept that the models being used are “models” that have certain limitations and can be improved. A detailed description of the evolution of the motor model can be found in a previous paper.¹²

Physics of Components

Modeling of the components requires a certain level of understanding of the physics of the device being modeled. At the level the course is taught, students understand what basic components do but do not have a full understanding of the internal physics. In this course, the

input/output characteristics of simplified components are modeled. The major components developed in the course are listed below.

- Electric motors – Modeled as an electromechanical energy conversion device where electrical power equals mechanical power. The model allows the component to operate as a motor or a generator.
- Battery – Modeled as a voltage source in series with a resistance. The resistance and open circuit voltage of the battery are functions of temperature and battery state of charge. The battery state of charge is calculated by integrating the battery current.
- Engine – Modeled as a device that converts fuel energy into mechanical energy. The output torque is equal to the throttle position times the available torque. The available torque is determined by a look-up table that describes the maximum available torque as a function of engine rpm. Fuel consumption is calculated by integrating the output of a two-dimensional look-up table that specifies the consumption rate as a function of rpm and throttle position.
- Transmission and Chassis– The powertrain is constructed using parts available in The MathWorks SimDriveline.¹³ This product has components for gears, tires, differentials, clutches, and more. In the series model, the electric motor is tied to the rear wheels through a differential. The vehicle acceleration/deceleration is modeled using the SimDriveline Vehicle Solver block. This block takes a force input and calculates vehicle speed based on the road incline, vehicle mass, and the drag coefficient. Although students could generate some of these component models in the class, the SimDriveline components are used to allow more time for system and controller development.

System-Level Model

A major element of the class is the development and integration of the individual subsystem components to form a system which solves a problem. In the example covered in the course, a hybrid-electric vehicle model is developed to meet or exceed certain performance specifications. As the system is constructed, questions such as “what would we use this component for” are automatically answered because students see the component used in a system that performs a function with which they are familiar. The system-level model also shows how the subsystem components interact and specifies the required communication signals.

In traditional courses, subsystem components are typically developed in isolation. Educators know what the subsystems are used for, when simple models are appropriate, and when more detailed models are required. Unfortunately, educators may take for granted that students have an interest in the subject and an understanding of what the component is used for. Educators may incorrectly assume that students are interested in learning about a component and developing more and more accurate models to describe the component. This may be true for a component such as an engine, which students know what it is used for. However, if the component being described is something unfamiliar like a voltage clamp, students may have no idea what it is used for and why it should be modeled, studied, and improved. Using the subsystem components in a system addresses all of these issues. Students automatically see the application of specific components and their use. By developing models, students can see the interaction of the various components, and how component specifications affect the overall system performance. Through

simulations, students can see how their simplified models affect performance and make a judgment as to whether component models need to be improved.

Controller Design

In order for the system to function properly, a system-level supervisory controller must be constructed. This controller is an extension of the classical feedback controllers developed in a standard undergraduate control course. Some of the functions it performs are classical linear feedback such as controlling the speed of an engine. If a student has taken a control theory course, construction of the supervisor allows students to apply the control theory to a real-world system. (In this case, the vehicle model is the plant, and the students understand the plant and have developed the plant model.) For students who have not taken a control course, the basics of classical feedback are covered and a feedback loop with proportional control is created.

In addition to classical feedback systems, the supervisor also tells components when to become active. The supervisor tells the engine when to start, tells the motor how to start the engine, controls battery charging, and controls vehicle shifting. Operations of this type must be performed in a specific sequence and only under certain circumstances. For example, it would not be safe to allow the driver to shift into reverse while the vehicle is moving forward. These functions require the addition of logic to the controller, a topic usually not covered in traditional control courses.

The supervisor must be constructed to oversee and control all functions of the system. By designing a controller for this large system, students gain an understanding of what a controller is, what it is used for, and an appreciation for classical control theory. This is different than giving students a set of differential equations with no physical context and having them control output y through input x_i . Students understand, however, that when the vehicle speeds out of control, bad things happen to the driver, and they had better modify their controller.

Advanced Software

Building a large system model would not be possible without advanced software tools. MathWorks Simulink¹⁴ was chosen to implement the entire vehicle model. This was done for several reasons. First, Simulink is a graphical environment where models are constructed as block diagrams. The graphics and shape of the symbols correspond to symbols with which engineers are familiar. This allows engineers to understand the function of a model by looking at a block diagram rather than looking at thousands of lines of code. Second, The MathWorks has several add-on products that can be used to make specific physical systems easier to model. In the development of a vehicle model, SimDriveline has models for transmissions, gears, tires, and more. Using these built-in models streamlines the development of a large model. Had an electrical system been developed, SimPowerSystems is available. Several other products are available for a wide variety of systems. Third, Stateflow^{®15} allows the easy implementation of a state machine in the controller. Simulink is a natural environment for developing linear classical controllers. However, implementing the logic needed to control a vehicle requires both classical control as well as logic. Stateflow provides a convenient method for adding the required logic to a Simulink model. The last reason for using Simulink is the automatic code generation

capabilities of Simulink through the Real-Time Workshop[®]. A future improvement of the course is to deploy the plant and controller on real-time targets and run hardware-in-the-loop simulations.

Present Course Outline

A course has been developed as a joint venture between Rose-Hulman Institute of Technology and The MathWorks to teach Model-Based Design within the context of developing a hybrid-electric vehicle. The course has been taught as a two-day MathWorks short course and as a two-credit quarter-length course at Rose-Hulman Institute of Technology. The goal of the sophomore-level course is to introduce the concepts Model-Based Design by developing a model for a series hybrid-electric vehicle. Prerequisites are first-year calculus and physics courses. The course involves simulation only – no real-time simulations are attempted. A brief outline of the course is given below.

Part 1:

- Introduction to Simulink
- Using SimDriveline components:
 - Differential
 - Torque and velocity actuators
 - Velocity sensor
 - Tires
 - Inertia, Environment block
 - Vehicle Solver block
- Create vehicle model with rear diff, constant torque motor, and tires. Plot performance.
- Create electric motor using look-up table (torque versus rpm)
- Create a vehicle parameter initialization file

Part 2:

- Creating model hierarchy
- Create an electric motor model:
 - Torque input, voltage input, current output
 - Flat torque curve
 - Regenerative braking and motoring
 - Electrical Power = Mechanical Power
- Create a battery model
- Create a driver block
- Run simulations showing vehicle speed and battery state of charge

Part 3:

- Identify and correct model physical problems shown by the simulation
- Run a simple drive cycle

Part 4:

- Read a drive cycle using the xlsread function
- Set up the initialization file to automatically run
- Modify the simulation to automatically run for the length of the simulation
- Simulate standard drive cycles

- Enhanced motor model:
 - Efficiency
 - Torque versus rpm

Part 5:

- Introduce SimDriveline engine:
 - Investigate torque curve
 - Dissect model
- Create detailed engine model and test outside vehicle model:
 - Fuel consumption – read fuel map from Excel
 - Torque versus rpm – read torque map from Excel
 - Braking torque
 - Engine off torque

Part 6:

- Create the Stateflow controller
- Maintain battery state of charge between min and max limits
- Develop an engine starting method
- Calculate fuel efficiency:
 - Miles per gallon gas equivalency (MPGGE)
 - State of charge correction

Part 7:

- Identify and eliminate model problems
- Identify and eliminate hazards
- Observe model performance to see the effects of changing component properties

Model Discussion

The breadth of the course can be appreciated by looking at some of the components of the models. Although an in-depth discussion of the model is not appropriate here, viewing the top-level block diagram and looking at a few of the models will provide insight as to the complexity of the model and the depth of its development in the class. Figure 2 shows the top-level block diagram of the series hybrid-electric vehicle.

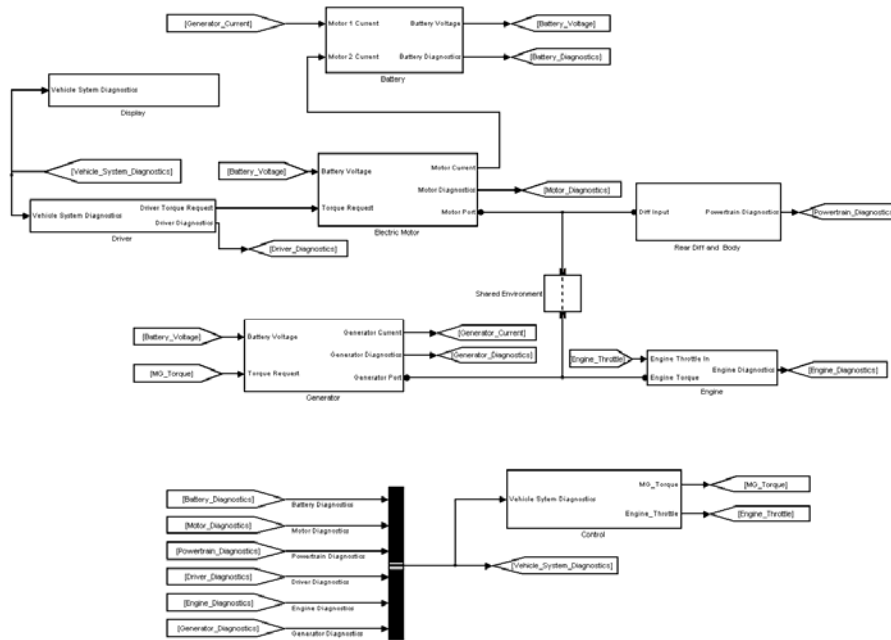


Figure 2: Top-level system block diagram.

The Driver block implements a classical feedback loop to maintain the vehicle speed at the desired speed. The vehicle speed is compared to a desired speed and an error signal is generated and amplified. The amplified signal is the output of the block and is interpreted as the driver torque request. The contents of the Driver block are shown in Figure 3:

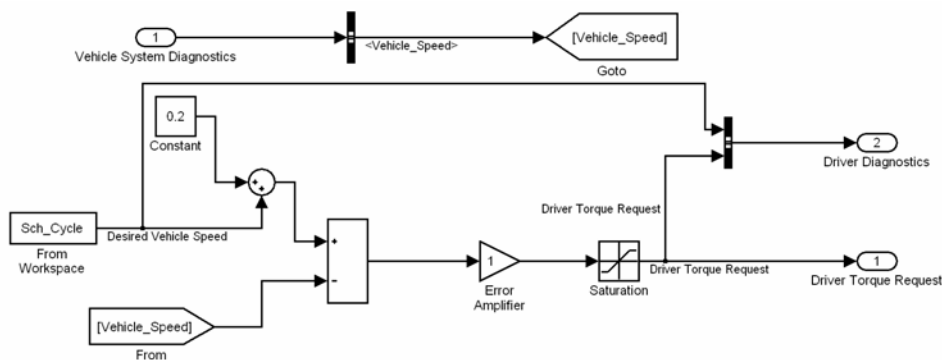


Figure 3: Driver block.

The torque request issued by the Driver block is passed directly to the electric motor. The motor outputs a torque to the rear differential (contained in the rear Diff and Body block) which causes the vehicle to accelerate. The contents of the electric motor model are shown in Figure 4:

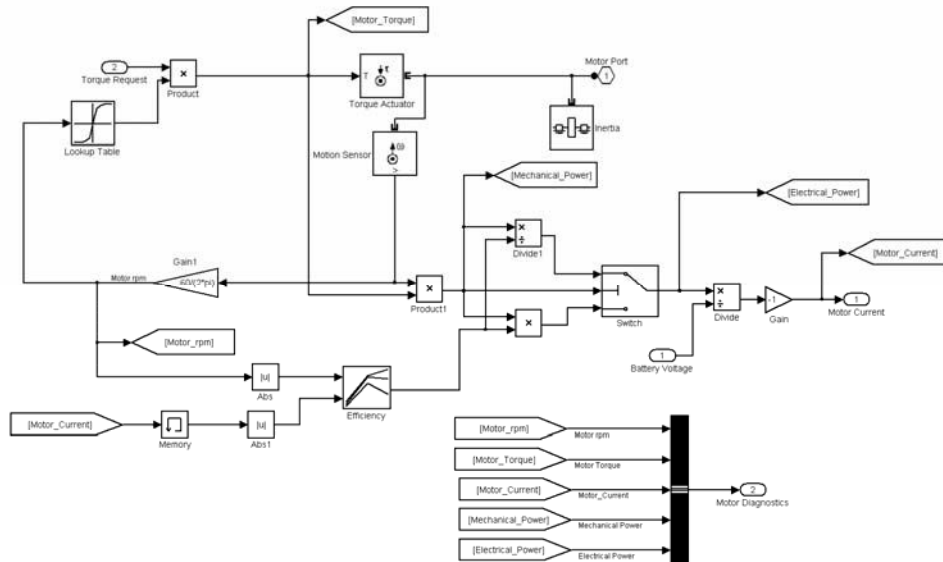


Figure 4: Electric Motor Model

The electric motor model calculates the mechanical output power as the applied torque times the radial velocity of the motor output shaft. This power is then divided by the battery voltage to calculate the motor current. This current is passed to the battery subsystem, which uses the information to calculate the battery state of charge. Note that look-up tables are used to specify the motor torque curve and the motor efficiency.

The Rear Diff and Body block is implemented with SimDriveline blocks and is shown in Figure 5:

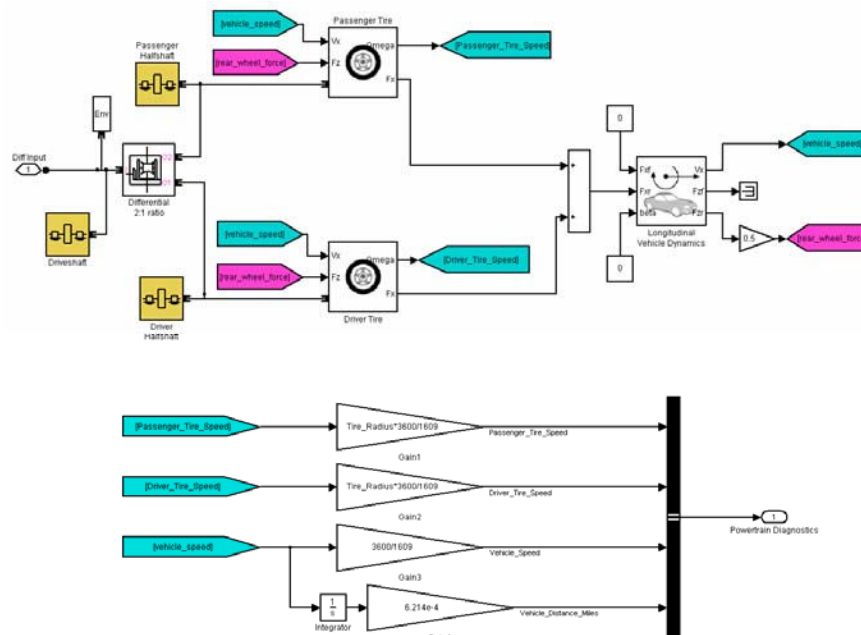


Figure 5: SimDriveline implementation of the powertrain.

The input of the differential comes from the output of the electric motor. This torque is split between the two rear tires. Given a vehicle weight, tire speed, and torque input, the tire model calculates a force in the x-direction. The forces from the two tires are summed together to calculate the total force in the x-direction, which will either accelerate or decelerate the vehicle. This force is the input to the Longitudinal Vehicle Dynamics block which calculates the vehicle speed based on the total force in the x-direction, the vehicle weight, and the drag coefficient. The Rear Diff and Body block demonstrates the ease to implement a powertrain using SimDriveline.

The Driver, Electric Motor, Rear Diff and Body, and Battery blocks create a basic electric vehicle. To create a hybrid-electric vehicle, a second energy source in addition to the battery is required. In the top-level block diagram shown in Figure 2, an engine drives a generator, and this generator charges the battery. The engine is not connected to the powertrain. Its only function is to power the generator and charge the battery. Since the engine has no starter, the generator is first used as a motor to start the engine. Once the engine produces power, the generator acts as a generator and charges the battery.

Also shown in the top-level block diagram are the vehicle system diagnostics bus and a display subsystem. The diagnostic bus contains all system-level signals and is used to debug the model and pass information between the various subsystems. These signals are analogous to the signals that would be passed between subsystems in a real vehicle. The display block is a convenient place to display the system-level signals.

The system-level supervisory controller shown in Figure 6:

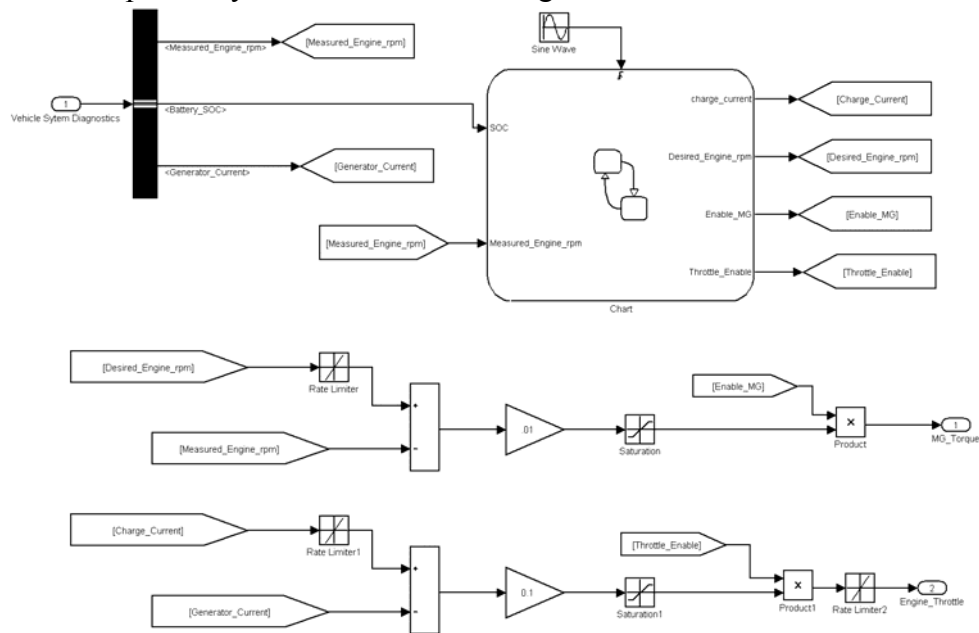


Figure 6: Vehicle supervisory controller.

This block controls the system-level operations of the vehicle. It is assumed that each subsystem has its own controller which makes the component perform to specifications. For example, the engine has an ECU that controls all aspects of engine operation. All the supervisor needs to do is

turn on the ECU and then pass the ECU a throttle signal. Given a throttle signal, the engine then outputs torque.

Shown at the bottom of Figure 6 are two classical feedback loops with proportional feedback. The top loop maintains the engine speed at a specified rpm by controlling the torque load of the generator. This loop is used to start the engine and then maintain the engine speed at a constant rpm. The bottom loop is also a classical feedback system with proportional feedback. Given that the top loop maintains constant engine speed, the bottom loop uses the engine throttle to maintain constant battery charging current. Thus, it is seen that the controller implements a non-trivial multi-input multi-output linear feedback system.

The logic for determining when various components start and stop is implemented using the Stateflow chart. This chart implements the state machine shown in Figure 7:

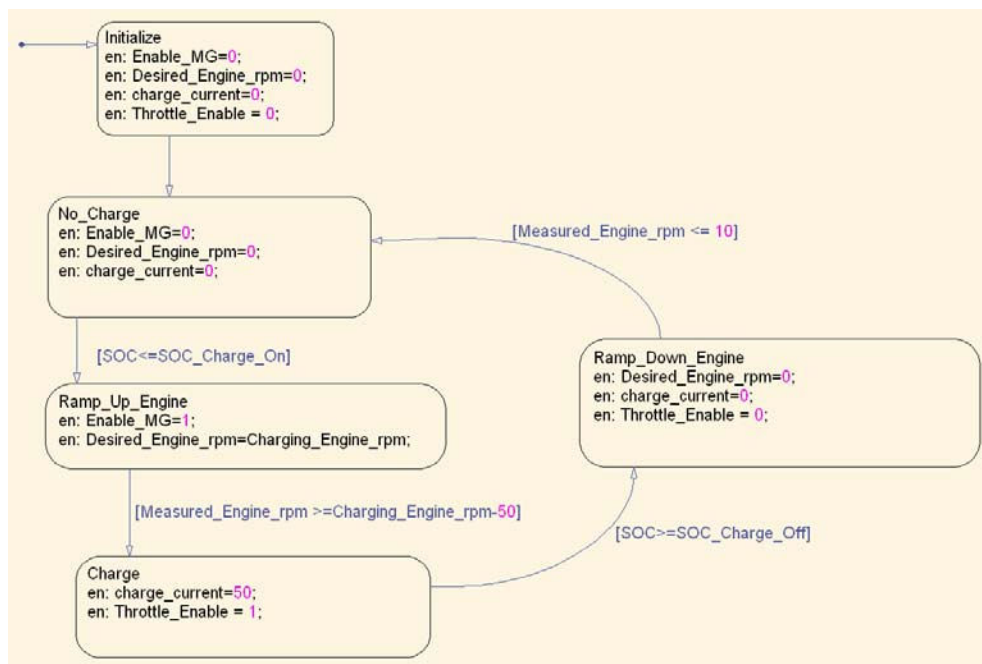


Figure 7: The state machine inside the hybrid-electric vehicle supervisory controller.

This chart controls the battery state of charge and engine starting and stopping. Although it is a fairly simple example of a chart, it can be modified easily to control many other aspects of vehicle operation such as a vehicle shifting and error diagnostics.

Assessment

Teaching Model-Based Design gives students a different perspective on components and their interactions, as well as a different perspective on engineering. The top-down approach is not used in many courses. The questionnaire is designed to assess students' perceptions of the course, and if, indeed, the course helps students understand subsystem components and the role the component plays in the overall system. The assessment questionnaire is given below:

Model-Based Design - Post-Course Survey
Fall 2006

1. Please rate your level of knowledge of and confidence in your ability for each of the course concepts below, using the following scale.

1 2 3 4 5
 Very Low Low Moderate High Very High

| | Knowledge | Confidence |
|---|-----------|------------|
| 1. interaction of components in a system | | |
| 2. process of designing a system by starting small, building components, and integrating them | | |
| 3. creating specifications for system components | | |
| 4. communicating with engineers working on different system components | | |
| 5. how components function together to form a system | | |
| 6. boundaries between various components in a system | | |
| 7. developing a controller for a system | | |
| 8. implementation of a controller in real-time | | |
| 9. testing real-time embedded systems | | |
| 10. verifying and validating system performance | | |

2. Which of the following concepts did you find Model-Based Design to be the most informative for?

- interaction of components in a system
- process of designing a system by starting small, building components, and integrating them
- creating specifications for system components
- communicating with engineers working on different system components
- how components function together to form a system
- boundaries between various components in a system
- developing a controller for a system
- implementation of a controller in real-time
- testing real-time embedded systems
- verifying and validating system performance

3. What changes could be made to the course to enhance your understanding of the concepts listed above?
4. How is this course different from other design courses you've taken at RHIT?
5. This course increased my desire for hands-on experience with vehicle modeling.

Strongly Agree Agree Disagree Strongly Disagree

The course has been offered twice and the assessment questionnaire has been applied once. This is not enough data to support any conclusions, so the data are not presented. This course is a work in progress, and results will be presented in future publications.

Future Course Development

The present course focuses on simulations in the MATLAB and Simulink environment only. Although a large amount of knowledge can be gained about the system and its components, this is not a complete picture of the process used in industry. Future development of the course will involve real-time simulations, processor-in-the-loop simulations, and automatic code generation for deployment on a production target. Using Real-Time Workshop with Simulink facilitates automatic code generation and allows users to easily compile their models and use them on various targets. There are several levels of real-time simulations that could be performed that fit in well with hybrid-electric vehicle modeling and can be accomplished with MathWorks software:

1) Stand-Alone Simulations – In this level, the entire model is compiled as a single module and deployed on a target. The MathWorks automatic code generation tools are used to generate the executable modules. User controls and model status displays are implemented on a PC and communication between the target and PC is accomplished through an Ethernet connection. This level of simulation is the easiest to accomplish, shows the process of converting a model to run on a real-time target, and gives the student a new environment in which to debug and test their models.

2) Processor-in-the-Loop Simulations – In this level, the controller subsystem is separated from the plant model and the two are compiled as separate modules. MathWorks automatic code generation tools are used to generate executable modules for the plant and controller. Typically, the two targets are different. A high-power target is required for the plant since it simulates a large model of the plant. The controller target is usually a much less powerful target and would typically be the same target that is used in production of the vehicle. The two targets communicate with each other using some form of networking. In vehicle systems, this would be a CAN network. User controls and model status displays are implemented on a PC and communication between the target and PC is accomplished through an Ethernet connection. In industry, this level of simulation is used to verify the operation of the controller on the production target using the same communication protocols used in the final product.

After completing the three levels of simulation (simulation in Simulink and the two levels mentioned above) students would have a controller that has been tested rigorously and can be placed in a system with a great deal of confidence in its operation. After completing this course, students will see the entire modeling process from creating ideal models of individual components to implementing a controller on a production target and testing it with a mature model of the plant.

Conclusions

Model-Based Design is increasingly prevalent in industry but lacking from college and university curricula. The need for students to be adept at the modeling of systems, their associated subsystems, and overall system controller is the impetus for The MathWorks to partner with Rose-Hulman Institute of Technology to develop a Model-Based Design course within the context of a series hybrid-electric vehicle. Students follow Model-Based Design, beginning with exceedingly simple component models to develop a complex, multidisciplinary system in need of control. Incremental improvements to the components and controller enable students to assess the impact of their changes and determine if the associated level of complexity is warranted.

This course is a substantial improvement over traditional practice where educators focus on refining components outside of a system model, whereby the students “lose the forest for the trees.” By specifying component interfaces, they learn how to concurrently improve simple models and control strategies to better reflect real-world behavior and meet design criteria.

Pedagogical assessment is underway as the course has been offered once with a survey to capture student perceptions of learning enhancement. As more data are collected, a future publication will focus on learning outcomes. Anecdotal evidence from students suggest that the methodology has merit, particularly in that sophomore-level students are completing the course with hands-on experience in multidisciplinary system modeling and control.

Bibliography

- 1 <http://www.MathWorks.com/applications/controldesign/description/index.html>
- 2 “Nissan Develops Emission Reduction System for Mass-Production Vehicles Using The MathWorks Tools”, <http://www.MathWorks.com/mason/tag/proxy.html?dataid=4388&fileid=20542>
- 3 “Vehicle Systems Integration Designs Hardware-in-the-Loop Simulator for the U.S. Department of Transportation”, <http://www.MathWorks.com/mason/tag/proxy.html?dataid=5297&fileid=21675>
- 4 “BAE SYSTEMS Controls Develops Autopilot for Unmanned Aerial Vehicle Using The MathWorks Tools” <http://www.MathWorks.com/mason/tag/proxy.html?dataid=2886&fileid=21724>
- 5 http://www.MathWorks.com/company/user_stories/
- 6 “DENSO's Model-Based Design Capability to Contribute TO OEM's Success”, <http://www.MathWorks.com/industries/auto/iac06/presentations/13-Suzuki-DENSO.pdf>
- 7 “Introduction to Hydraulic Systems Modeling and Simulation”, <http://www.msos.edu/seminars/seminars/ihsms.shtml>
- 8 <http://www.incose.org/>
- 9 <http://www.challengex.org/>
- 10 Larson, D., Tester, J., Wecker, S. and E. Brauer, (2001), "The Practice and Process of Engineering Design through Robotics, EGR 286 Engineering Design II - The Process", AAAI Spring Symposium on Robotics and Education, Palo Alto, March. .
- 11 Collier, K., Hatfield, J., Howell, S., and D. Larson, (1996), "A Multi-disciplinary Model for Teaching the Engineering Product Realization Process", 1996 Frontiers in Education Proceedings, Salt Lake City, UT.
- 12 “Model-Based Design for Hybrid Electric Vehicle Development,” Marc E. Herniter, Zachariah Chambers, Sameer M. Prabhu, Matthew Stevens, Roydon Fraser, and Michael Fowler, Presented at The 22nd International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exposition, EVS 2006 paper No. P10019-002, October 23-28, Yokohama, Japan.

- 13 The MathWorks Inc., "SimDriveline User's Guide," Version 1.3, The MathWorks Inc., Natick, MA, March, 2006.
- 14 The MathWorks Inc., "Using Simulink," Version 6.4, The MathWorks Inc., Natick, MA, March, 2006.
- 15 The MathWorks Inc., "Stateflow User's Guide," Version 6, The MathWorks Inc., Natick, MA, March, 2006.