MECHATRONICS

Mechatronics is a process for developing and manufacturing electronically controlled mechanical devices. Many of today's automated equipment and appliances are complex and smart mechatronics systems, composed of integrated mechanical and electronic components that are controlled by computers or embedded microcomputer chips. As a matter of fact, mechatronic systems are extensively employed in military applications and remote exploratory expeditions (1,2). Industrial mechatronic systems are used extensively in factory automation and robotic applications, while commercial mechatronics products are widely found in office and home appliances as well as in modern transportation. Successful systems and products are the ones that are well designed, well built, and affordable.

The term *mechatronics* was coined in 1969 to signify the integration of two engineering disciplines—*mecha*nics and elec*tronics.* In the early 1970s, Japan was the largest ship and tanker builder in the world and its economy depended heavily on oil-driven heavy machinery and steel industries. The 1973 oil crisis saw the crude oil prices skyrocket from \$3.50 per barrel to over \$30.00 per barrel. The consequent disastrous impact on its oil-dependent shipping industry prompted Japan to rethink about its national economic survival and strategies. Microelectronics and mechatronics were two emerging technologies embraced by Japan as major industrial priorities after the crisis.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*.

Definition. Several definitions for Mechatronics can be found in the literature (3–17). For example, the *International Journal on Mechatronics: Mechanics–Electronics–Control Mechatronics* (5) defines it as the synergetic combination of precision mechanical engineering, electronic control, and systems thinking in the design of products and manufacturing processes. Others stated it to be a synergetic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial product and processes. Mechatronics requires systems engineering thinking aided by computer simulation technology that enhances complete understanding of how its design decision affects decisions of the other discipline counterparts. It describes ways of designing subsystems of electromechanical products to ensure optimum systems performance.

To be more competitive and innovative, new mechatronics requirements often call for "smart" performance in dealing with operational and environmental variations. So to include for product competitiveness and added value of today's microchip and microprocessor technology, one may generalize the definition of mechatronics as follows:

Mechatronics is a systems engineering process for developing and integrating of computer-based electronically controlled mechanical components in a *timely* and *cost-effective* manner into smart affordable *quality* products that ensure optimum, flexible, reliable, and robust performance under various operating and environmental conditions. We refer to such a well-designed and well-integrated automation system as a *mechatronic system* or *product.*

The three key words in the definition are quality, time, and cost. The product must be safe, reliable, and affordable to consumers. To the manufacturers, the product must be produced quickly and efficiently and must be profitable; profit is indeed the name (purpose) of the game. Readers should not be surprised by the encounter of other contexts of mechatronics. In exploratory research, for instance, mechatronics thinking is used to develop customized systems where cost may not yet be a significant issue.

EXAMPLES OF MECHATRONICS SYSTEMS

There are (almost) endless examples of mechatronic systems and products. It would not be meaningful to attempt a compilation of the available products and automated systems (see AUTOMATION and ROBOTS). However, to emphasize a huge trend in research, development, and engineering effort that adds up to billions of dollars each year, we will take a look at the automotive mechatronics in some detail.

Today's fully loaded modern automobile easily carries over 30 automotive mechatronic systems to provide a high level of ride comfort and road handling, along with devices for safety, fuel economy, and luxury (18,19). Modern cars are controlled by several onboard embedded microcontrollers. A list of automotive mechatronic systems is provided here to emphasize the point: electronic ignition, electronic fuel injection, electronic controlled throttle, emission control, computer-controlled transmission and transaxles, cruise control, anti-lock brakes, traction control, computer-controlled suspension, steering control, body control functions such as power lock, windows, automatic wipers, sunroof and climate control, safety functions such as airbags, security systems, keyless entry system, instrument panel display, stereo system, and so on. Some examples of the latest automotive innovations about to hit the market are an anti-squeeze power window and sunroof, vehicle yaw stability control systems, collision warning/avoidance systems, noise and vibration cancellation, anti-roll suspension, hybrid electric vehicles, navigation aids, a built-in automotive personal computer, and others. The automobile industry invests heavily in research to develop these products. It is not surprising to find that several auto companies and suppliers are investigating similar mechatronic products at the same time. Thousands of engineers are employed to work in the area of automotive mechatronics.

THE MECHATRONICS CHALLENGE

Demand for mechatronics products prevail in today's market as consumers become more affluent and opt for gadgets that enhance performance in the products. In addition, as science and technology advance, the requirements for mechatronics systems often become more and more sophisticated and demanding. Many manufacturers realize the trend and the potential highly profitable market. They also realize that the challenge is in getting quality mechatronics product to market in a timely and cost effective manner.

An original equipment manufacturer (OEM) must have well-balanced and well-planned business and engineering strategies to compete in the market. In this article, we will set aside the essential business strategies and address only the relevant engineering aspects of developing a mechatronic product.

Figure 1 shows a simple model for conceptualizing a mechatronic system, emphasizing the control, sensor, actuator, and process entities that make up the system. It presents a general summary for understanding the overall configuration of the system and its connectivity to user command inputs, environmental influence, system states, and commanded outputs. Expected performance requirements for the mechatronic system can be defined at this level.

Figure 2 is basically the same mechatronic system as that shown in Fig. 1, but is conceived as a product made out of an electronic module and a mechanical module. The components in these modules may include the multitechnologies shown in the figure. The final selected components in this design will be used in actual implementation of the mechatronic product.

The engineering challenge to the manufacturers is to transform the concept of the mechatronic system (Fig. 1) into the multitechnology modules that make up the mechatronic product (Fig. 2). The development must be timely and cost effective and must ensure quality in the product.

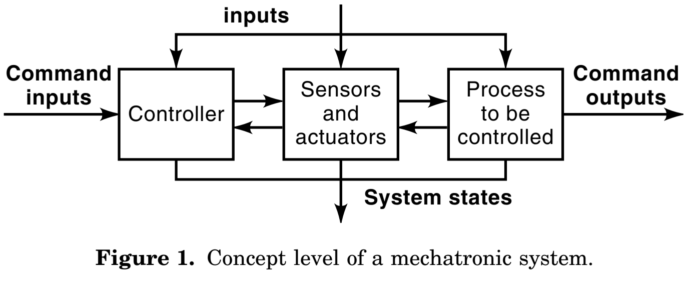


Figure 1. Concept level of a mechatronic system.

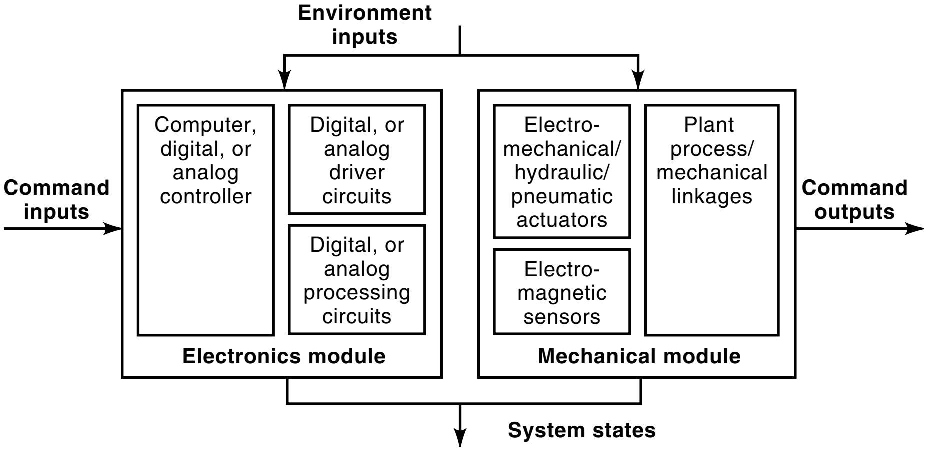


Figure 2. Electronics and mechanical modules of a mechatronic system.

The manufacturers invest in research, development, and manufacturing processes to produce products. A key to successful management of quality, time, and cost lies in a systems engineering perspective and approach (20) to the development process of mechatronic system.

The importance of mechatronics philosophy is quite evident when we reflect on the enormous success of Japan's electronics and automobile industries. The idea has since spread around the world, especially in Europe and the United States. Many industrialists, research councils, and educators have identified *mechatronics* as an emergent core discipline necessary for the successful industry of the next millenium.

SCOPE OF THIS ARTICLE

This article is written with an application engineer in mind. He or she has come up with a viable mechatronics concept or is assigned a mechatronics project. The objective is to build it as well as possible, as inexpensively as possible, and in the shortest amount of time. The idea is to avoid getting bogged down with heavy engineering mathematics and to look for state-of-the-art techniques and tools to expedite the development of the product.

This article emphasizes a systems engineering approach to the development process of mechatronics systems. It stresses the use of computer-aided engineering (CAE) tools for expediting the design and analysis of mechatronic products. It also addresses the foundations needed for dealing effectively with the multitechnology mechatronics. It is written with the assumptions that the reader has some or sufficient background in certain engineering discipline(s) related to mechatronics. It discusses the current trend and practice in *process, techniques, tools,* and *environment* for dealing with mechatronics. Finally, it provides an evaluation of the direction that mechatronics is heading toward in the future. It does not include details of physical system integration and manufacturing processes.

FOUNDATION FOR MECHATRONICS

Science, Technology, Engineering, Business, and Art

In a broad sense, successful mechatronics endeavors often involve one or more of these disciplines:

1. Science—discovery of new materials, methods, and so on
2. Technology—adaptation of technologies for innovation
3. Engineering—development and manufacturing of new products
4. Business—ability to gauge market, opportunity, and profit
5. Art—experience and skills that beat the competition

Coupled with the fact that mechatronics is a multitechnology discipline, the range of the knowledge is usually beyond that of a normal person. An exception may be the case of an exceedingly simple mechatronics endeavor. Mechatronics in general, therefore, is inherently a team effort, rather than a single individual effort.

In the high-technology world of today, however, a paradigm of systems engineering has been applied to improve the efficiency of teamwork. An important point in that paradigm is the use of CAE tools to communicate and explore possibilities of sophisticated ideas among the team members. The CAE tools may include expert knowledge systems, computer simulation, computer graphics, virtual reality immersion, and so on. If we are bold enough to accept it, the CAE tools can effectively be treated as "personal assistants" in the team. When wisely employed, they can assist engineers to ensure quality in the design of the mechatronics system, shorten time for analyses, and reduce cost of development, through countless computer simulations and evaluations of the mechatronics systems.

Multitechnology, Multiengineering, and Systems Engineering

Figure 3 is a pictorial summary of the technologies and engineering that mechatronics can entail. The left half of the figure depicts the mechatronics system as a real-time product that responds to programmed event and user command and reacts to environmental circumstance. It shows the multifunctional interfaces of the mechatronics system including mechanical mechanisms, sensors and actuators, input and output signal conditioning circuits, and computers or embedded microcontrollers (18,19,21).

Shown in the right half of the figure, an integration of such a system would require certain appropriate skills and experience from mechanical, electrical, electronics, and computer engineering. At the implementation level, skills for dealing

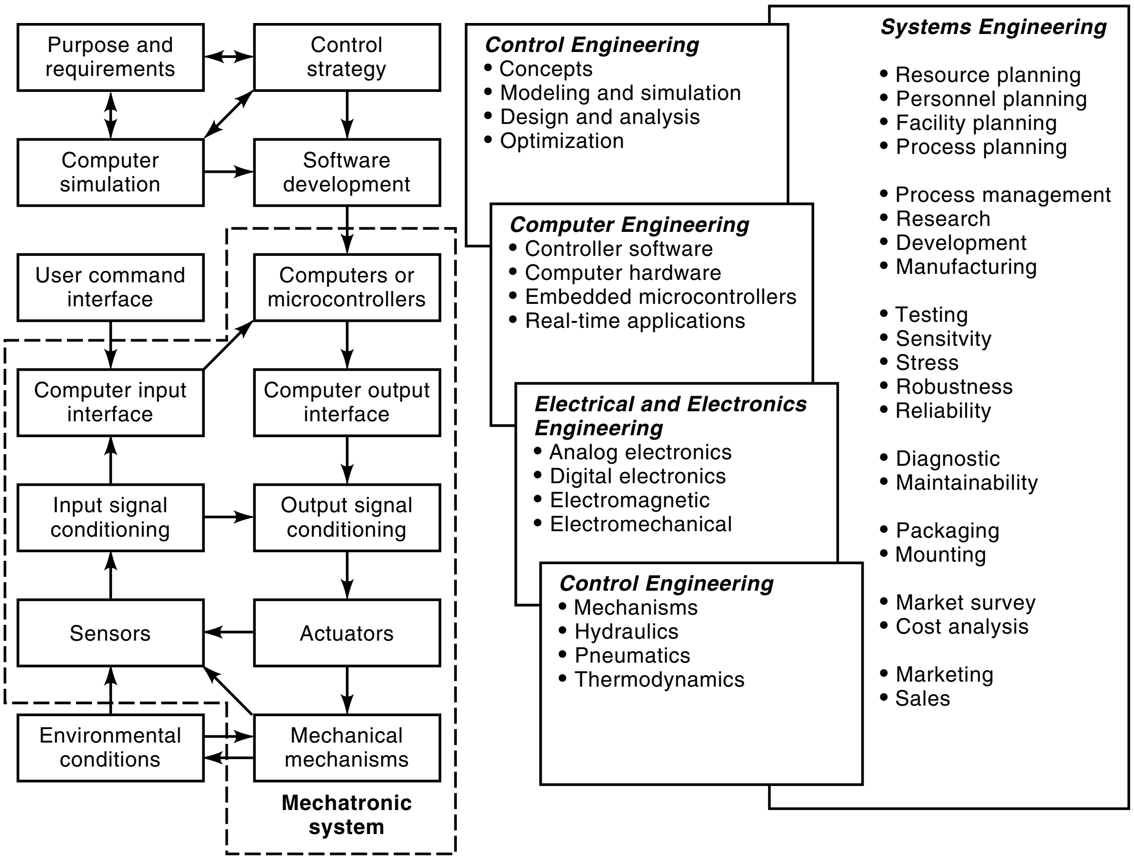


Figure 3. Multitechnology, multiengineering, and systems engineering nature of mechatronics.

with computer, electrical, electronics, electromagnetic, electromechanical, mechanical, hydraulic, pneumatic, and thermal components will be desired (22). At the concept design level, however, background in control theory will be needed to translate the purpose of the product into its technical requirements and define a control strategy with the aid of computer simulation study (23). The software development will then implement the control scheme in the system.

The right half of Fig. 3 also concerns with systems engineering to complete the job—that is, to bring the mechatronic product into being (20). Such an endeavor would entail the following: planning of resource, personnel, facility, and process; management of process; research, development, and manufacturing; product testing; evaluation of sensitivity, stress, robustness, and reliability; packaging and mounting; marketing; maintenance; and cost analysis and management. This reiterates the fact that teamwork is a necessary requirement when dealing with a mechatronic product life cycle (see SYSTEMS ENGINEERING TRENDS).

Computer-Aided Engineering Tools

As mentioned earlier, CAE tools are employed to assist designers and engineers in carrying out the development of mechatronics. Computer-aided design (CAD) packages have been used to render a graphical mock-up of solid models in the design of package, looks, fits, and mounting for mechatronic products. CAD is a widely used technique in mechanical design and analysis in the automobile and aerospace industries.

For the purpose of this article, we are concerned with CAE tools that assist engineers in designing control schemes, conducting performance analysis, and selecting the right components for the mechatronic system. The CAE software therefore must simulate the responses of dynamics system and allow control applications to be evaluated. Examples of such Computer-Aided Control Systems Design (CACSD) packages include Matlab/Simulink®, Matrix$_x$/SystemBuild®, P-Spice®, Electronics Workbench®, Easy-5®, Saber®, and so on. These software packages have a *schematic capture feature* that interprets block diagrams and component schematics for the simulation. This convenient feature lets the engineer concentrate on the engineering problem rather than the mathematical aspects of the simulation.

Saber (24–26). Of the above packages, the one that stands out as the industry standard is the Saber simulator. Saber has been accepted in the automotive industry as the CAE tool for dealing with mechatronics design and analysis. In fact, auto suppliers are now required to use Saber to communicate mechatronics design and analysis problems to General Motors, Ford, and Daimler-Chrysler. Figure 4 explains why Saber is well accepted by the industry. It illustrates a multitechnology nature of mechatronics where interdisciplinary knowledge of engineering and teamwork are key to the endeavor. The Saber simulator can be used to model the cross-disciplinary mechatronic system and provide an interactive platform for experimentation, discussions, and communication among the team of designers, engineers, and managers for the project. It provides a common medium to predict "what if" scenarios for all concerns and leads to "optimal" trade-off decisions.

An easy way to appreciate the Saber simulator is to imagine a virtual "mechatronics superstore" inside the cybernetic space that offers the following products and services:

- The "store" has a large inventory of commercially available electronics and mechanical components for you to choose. It also contains templates with which you can define new specifications for the components. [Saber has libraries of over 10,000 mechatronic parts (represented by component icons).]
- You have unrestricted "shopping" privilege that lets you "buy" and "exchange" any number of parts. (Drag and drop components and templates in a workspace window.)
- The "store" has a "assembly" facility where you can "integrate" the parts together into a working model, according to the schematic of your mechatronic design. (Connect parts to design a schematic.)
- It also has a "testing" facility with signal generators and display scopes for observing, validating, and verifying responses of the newly assembled mechatronics model. (Conduct simulation of system response.)
- It provides a means of conducting performance analysis and component analysis to check how good the selected parts are, and it delivers reports on the results. (Check performance requirements, and investigate components for stress and robustness.)
- You may conduct as many designs, analyses, and experiments in this store as you wish until you satisfy the requirements of the mechatronic product that you plan to build. (Discuss, redesign, and optimize.)
- You may bring your teammates to participate in the above activities. (All players from the start until the end of the mechatronic product life cycle can be included in the discussion using the simulation.)

Saber therefore facilitates *virtual prototyping of mechatronics functions* with realistic models of commercially available parts. Analogy, Inc., the company that produces Saber, collaborates with many OEMs such as Motorola, Texas Instruments, Harris Semiconductors, and Mabuchi Motors to model components and also validate and verify their characteristics as accurately as possible. An application engineer can use the verified models in the schematic without the burden of deriving mathematical formulation, programming, and debugging of codes. He or she can request performance reports from the virtual prototype simulation. As you can imagine, Saber provides support in the form of virtual parts, a facility, and a "personnel assistant."

Driving the Point Home. To illustrate the point further, Fig. 5 illustrates the actual schematic used in Saber simulation to represent the conceptual level design of a servo-positioning system. Figure 6, on the other hand, is the Saber implementation-level schematic for the system with selected components. A major process in mechatronics is to translate Fig. 5 into Fig. 6. The simulator helps the engineers to design the virtual prototype shown in Fig. 6 and analyze the integrity of the selected components. More details of this example will be presented in a later section.

Breadth and Depth of Disciplines in Mechatronics

It is clear that development of a complex mechatronic system will require an experienced engineering specialist with depth of expertise and breadth of experience to lead the team for the project. Of course, this is not necessarily the case if we are dealing with simple mechatronic systems. In either case, learning on the job is often one of the means of getting the job done. Indeed as an added benefit, a multitechnology CAE tool can be a big help in learning and confirming ideas in disciplines other than your own. It complements your knowledge and that of your team.

This article assumes that the reader and his team have certain backgrounds in control, computer, electronics, and

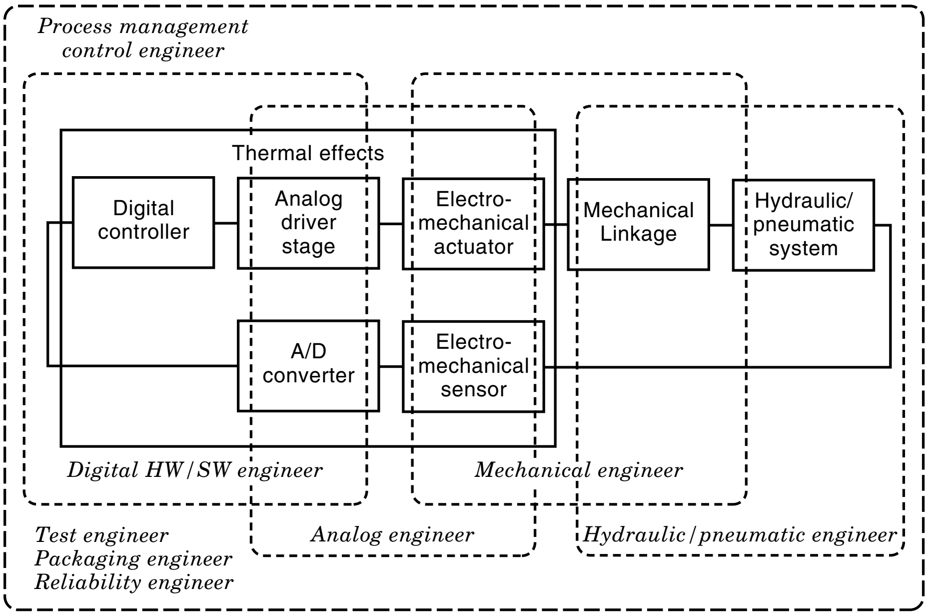
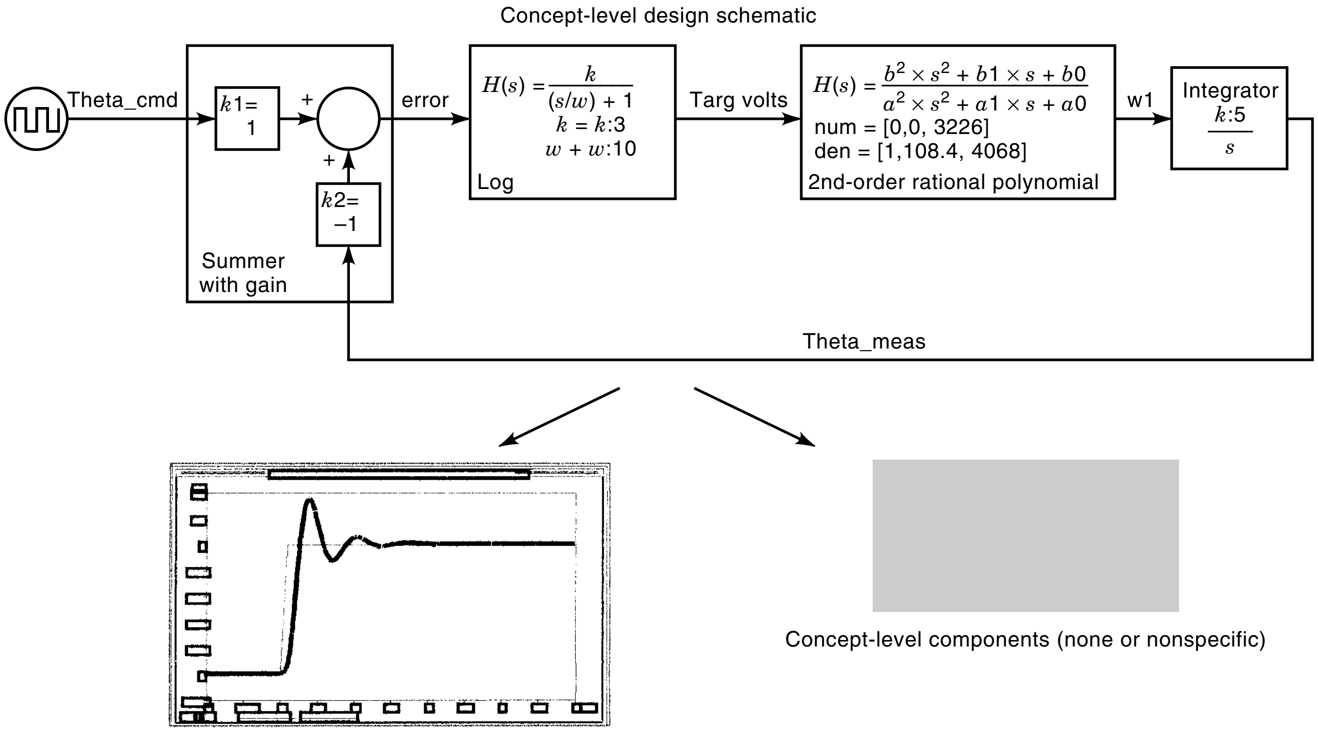


Figure 4. Overlapping disciplines and teamwork in mechatronics.


Concept-level design schematic

Concept-level simulation (ideal)

Concept-level components (none or nonspecific)

Figure 5. Concept-level design, analysis, and components.

mechanical engineering. Many of these backgrounds are covered elsewhere in the Encyclopedia. This article chooses to emphasize the systems engineering process (20) for designing and analyzing a mechatronic system. It deals with the problem at the system level, the subsystem level, and the component level with the help of a CAE tool. Although the Saber simulator is the main CAE tool used in developing the illustration, we describe its features and capabilities in a generic way so as to emphasize the concept of the process.

PROCESS AND TECHNIQUES FOR DESIGNING AND ANALYZING MECHATRONIC SYSTEMS

Process in Mechatronics Design and Analysis

The process can be grouped as follows: (1) requirements and specifications process, (2) top-down design process, and (3) bottom-up analysis process.

Requirements and Specifications Process. This is a stage where the engineers use their experience to envision the performance of the mechatronic systems to be built. Technical specifications are derived from nontechnical user requirements.

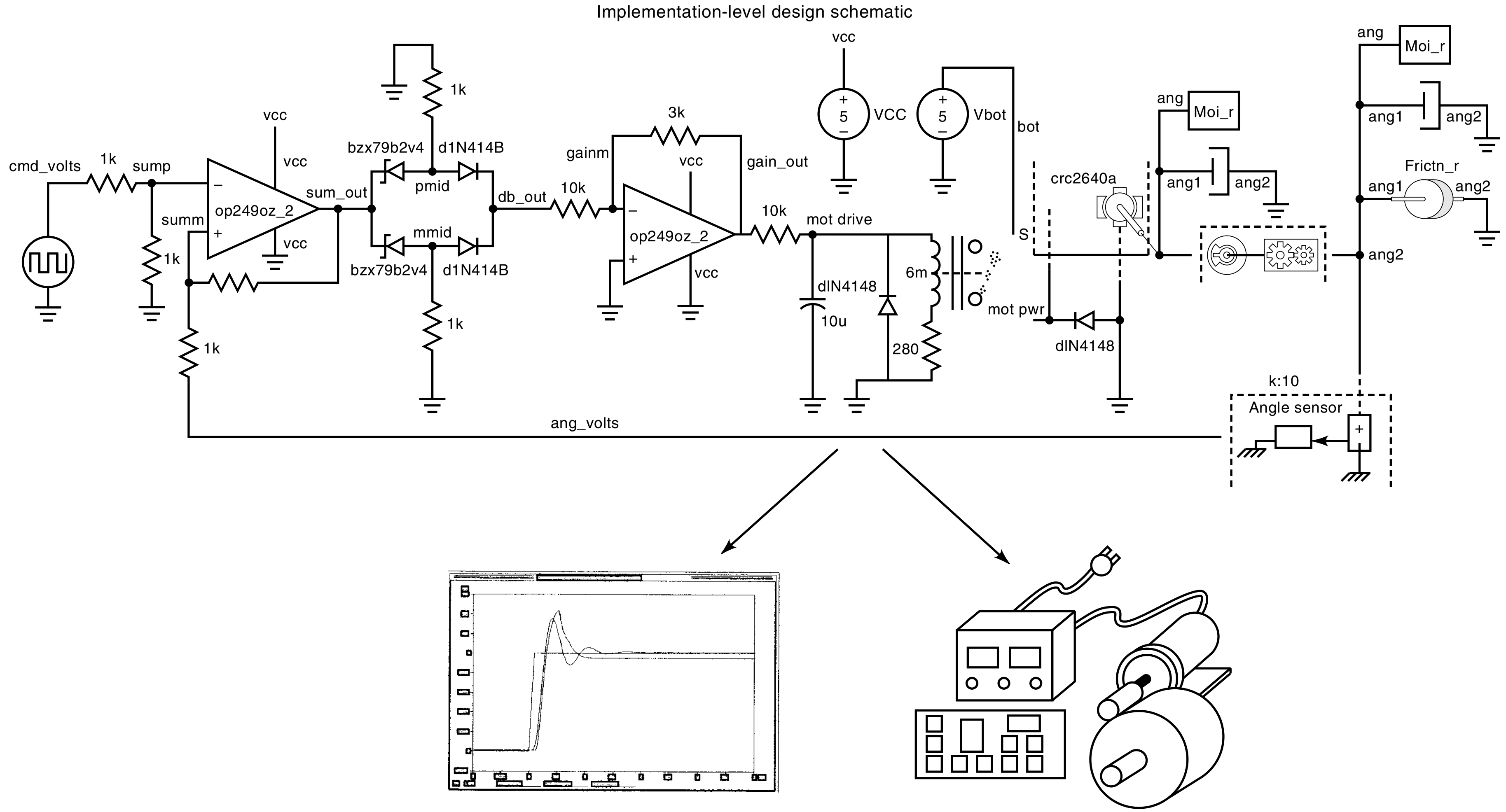
User requirements are qualitative descriptions of what the users need, want, desire, and expect. They are often stated in nontechnical terms and are not usually adequate for design purposes. However, they provide a subjective qualitative means of characterizing and judging the effectiveness of a system or product.

Technical specifications are derived from the user requirements. They spelled out the required characteristics in clear, unambiguous, measurable, quantitative technical terms to which the engineering team can refer. The technical specifications define the engineering design problems to be solved and are directly traceable to the user requirements.

The performance design and analysis for a mechatronic system are accountable to two technical specifications: *functional specifications* and *integrity specifications.* A *functional specification* specifies how well the system must perform in normal conditions expected of the system. It seeks a workable scheme for the problem. Functional specifications are a collection of performance measures, which is defined below. An *integrity specification* defines how well the system and its specific components must perform under expected strenuous conditions. It ensures that there are no weak links in the design. Examples of integrity specifications are sensitivity and stress analyses (26) as well as statistical and varying component analyses.

Design and Analysis Process. Mechatronics design and analysis deal with what is achievable through application of engineering technology. They comprise two complementary processes described below.

Top-Down Design Process. This stage is where engineers can become creative in their design to achieve the requirement for the mechatronic system. A *top-down design* is a validation process that ensures that the selected design and components are consistent and complete with respect to the *functional specifications* of the mechatronic system. The validation process is used to ensure that we are working on the *right problem* by guiding the detail design towards the functional requirements (27). The process does the following:



Implementation-level simulation analysis (higher fidelity)

Implementation-level components (specific)

Figure 6. Implementation-level design, analysis, and component selection.

- It begins with a schematic of an initial conceptual-level design to establish the operation and technical performance specifications for a mechatronics concept.
- It translates the concept design into a preliminary implementation-level design with specific components, satisfying technical specifications in the presence of the interfacing environments and operating conditions.
- It deals with problems in the intermediate stages of design during the transition through necessary new redesign iterations and requirement variations.

Bottom-Up Analysis Process. This stage is where engineers become critical of the preliminary design and set out to check the soundness or integrity of the design. A *bottom-up analysis* is a verification process that expands on the selected design solution to ensure that it meets the *integrity requirements.* It assures that we have solved the *problem right* by catching potential trouble spots before they become expensive and time-consuming crises (27). The process does the following:

- It carries out sensitivity analysis, stress analysis, and statistical analysis of the selected design under various expected strenuous conditions.
- It checks out feasibility and soundness of the selected design with other engineering groups such as manufacturing, testing, and reliability before commencing to build hardware prototype or "breadboard."
- It deals with problems of component selections and availability through iterations of redesign with the top-down design group.

Techniques for Mechatronics Design and Analysis

There are so many techniques and aspects regarding designing a mechatronics system (6–19,21,22) that it is not possible to mention all of them here. In this section, we have selected to highlights only three basic aspects as examples of design and analysis techniques that engineers should consider in the pursuit of designing a high-performance, robust, and reliable product. The three aspects are attention to (1) technical specifications to ensure that user requirements are met, (2) sensitivity analysis to ensure robustness to parameter variations, and (3) stress analysis to ensure reliability.

Technical Specifications. Derived from user requirements, technical specifications are used to guide the design of the mechatronic system. As explained earlier, we may categorize the technical specifications as functional and integrity specifications. Another useful specification is the term called performance measure.

What a Performance Measure Is. A performance measure, normally denoted by the symbol J, is a scalar numerical index that indicates how well a system accomplishes an objective (23). The index can be measured from the waveform characteristics of signal responses generated by the system in experiments, simulations, or theoretical analyses. A performance measure or index therefore is essentially a score that is used to rank the performance of systems. Simple performance measures that can be directly extracted from an output response of a system are maximums/minimums, rise time/fall time, steady-state value, settling times, initial value, peak-to-peak value, period, duty cycle, and so on. Other indexes require some computational effort—for example, frequency response bandwidth, resonance magnitude and frequency, average, root mean square, sum of weighted squared errors, power and energy, and so on. Figure 7 illustrates the details of some simple performance indexes in a step response. Performance measure J is used to evaluate sensitivity analysis, which is part of the integrity specifications.

Functional Specifications and Performance Measures. Functional specifications are made up of one or more performance measures that can be used to define the desired system performance more rigidly. The selected performance measures should be complementary and not conflict with each other. For example, settling time and percent of maximum overshoot complement one another in defining the specifications,

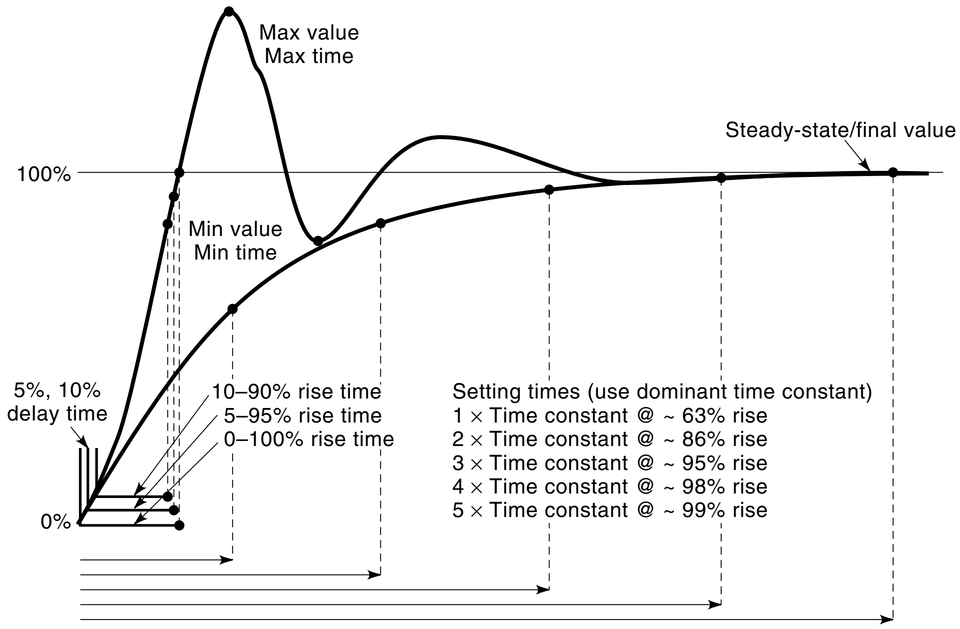


Figure 7. Candidates for performance measures in step responses.

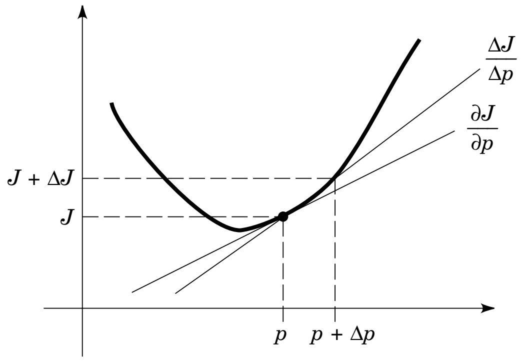


Figure 8. Definitions of sensitivity gradients.

whereas settling time and rise time may conflict in requirements. The functional performance specifications should be validated against "fuzzy" user requirements as well as used to check the performance of the component-level or implementation-level design.

Sensitivity Analysis

What a Sensitivity Analysis Is. A sensitivity analysis is a study that examines how sensitive a specified performance measure is to variation in the values of components or parameters in a system. For example, it can be used to determine how much the speed of a motor is affected by a change in the gain of an amplifier or a drop in the voltage supply.

How a Sensitivity Analysis Can Improve a Design. One can use the information obtained from a sensitivity analysis to identify which part of the system has significant impact on the system performance. Based on the finding, one may redesign the system to reduce the sensitivity and hence improve the robustness with respect to the particular parameter. The analysis can also be used to select appropriate tolerance values for the design to ensure that performance specifications are met.

How Sensitivity Is Defined. Sensitivity analysis of a system can be conducted by examining the gradient of performance measure J with respect to parameter p. This *sensitivity gradient* can be approximated by the ratio of variation ΔJ over perturbation Δp, as shown below:

$$S = \frac{\partial J}{\partial p} \approx \frac{\Delta J}{\Delta p}$$

Figure 8 illustrates the sensitivity gradient for a simple parameter variation. The interpretation of the gradient can be more rigorously observed using the Taylor series expansion of $J(p)$ around $p + \Delta p$:

$$\begin{aligned} J(p+\Delta p) &= J(p) + \frac{\partial J}{\partial p}\Delta p + \frac{1}{2!}\frac{\partial^2 J}{\partial p^2}\Delta p^2 + \frac{1}{3!}\frac{\partial^3 J}{\partial p^3}\Delta p^3 + \cdots \\ &= J(p) + \Delta J \end{aligned}$$

In most cases, it is more meaningful to compute the *normalized sensitivity gradient* as follows:

$$S_{\text{N}} = \frac{\partial J/J}{\partial p/p} = \frac{p\,\partial J}{J\,\partial p} \approx \frac{\Delta J/J}{\Delta p/p} = \frac{p\,\Delta J}{J\,\Delta p}$$

where J and p are the baseline performance measure and parameter, as shown in Fig. 8. However, the normalized sensitivity cannot be evaluated if J or p is 0 or very close to 0; hence the direct sensitivity gradient will be used.

How a Sensitivity Gradient Is Calculated. In certain cases where the performance measure J can be explicitly or implicitly expressed as analytical functions of a parameter p, it is possible to evaluate the sensitivity gradient in closed form. For instance, if

$$\begin{aligned} J &= f(y) \\ y &= a(u,\ p) \end{aligned}$$

where the functions f and a are analytical or differentiable at the points of concern, then the sensitivity gradient can be evaluated as

$$S = \frac{\partial J}{\partial p} = \frac{\partial J}{\partial y}\frac{\partial y}{\partial p}$$

The analytical solution can often shed insights into an analysis. An excellent example of this is the derivation of the well-known backpropagation training algorithm for neural networks, as well as its use in optimization and adaptive control methods. The possible drawbacks of the approach, however, include the need to know the explicit (direct) or implicit (indirect) formula that describes the relationships between J and p, and the necessary condition that the functions be analytical (differentiable) at points of interest.

A less mathematically laborious and yet effective approach for calculating sensitivity gradients is to employ computer simulation. The idea is to simulate and compute the performance measures J and $J + \Delta J$ when the system operates under the parameter p and $p + \Delta p$, respectively, where Δp is a small perturbation. The straightforward calculation $S \sim \Delta J/\Delta p$ approximates the sensitivity gradient. This computational technique can be used in the sensitivity analysis of simple and complex systems.

A Sensitivity Analysis Report. The sample report in Table 1 illustrates how a sensitivity analysis points out the parameters that have high sensitivity impact on the system performance measure. Attention should be given to large sensitivity gradients because they indicate that performance measure is highly sensitive to the parameter variations. Redesign of control scheme or circuit configuration may be required to reduce this effect and improve the robustness of the system. As can be seen, the computations in sensitivity analysis can be very tedious, laborious, and time-consuming. The key to the analysis is to employ a computer program to automatically generate the sensitivity analysis report for selected parameters in a design.

Stress Analysis

What a Stress Analysis Is. A stress analysis checks the conditions of components at operating conditions and compares them against the operating limits of the components. The analysis can pinpoint underrated components that are most likely to fail under expected strenuous operating conditions as well as components that are unnecessarily overrated and costly. It is an important design and analysis step for determining the ratings and rightsizing the components.

Table 1. Sample of a Sensitivity Analysis Report

Parameters	Sensitivity Gradient[a] $S = \Delta J/\Delta p$	Normalized Sensitivity Gradient[a] $S_N = (\Delta J/J)/(\Delta p/p)$	Comments
P1	1.811	1.050	OK. S and S_N are low.
P2	0.010	8.800	S_N is high. Check design
P3	190.0	0.290	S is high. Check design
P4	20.01	5.501	S and S_N are high. Check design

[a] Large values in sensitivity gradients S and S_N signify possible weakness in terms of robustness of the design.

What a Stress Measure Is. A stress measure is the operating level of a component or part that occurs during operation. Examples of stress measures are: power dissipation of a resistor, transistor, or motor; reverse voltage across a capacitor; junction temperature of a bipolar transistor; and maximum temperature and current in the coil of a motor, solenoid, and so on.

What Operating Limits Are. Manufacturers of components test their products and supply ratings of maximum operating limits (MOLs) for the components. The MOL may be a single value, or curve or surface function of the operating variables. Figure 9 shows the maximum power dissipation curve of a resistor alongside with the maximum collector current curve for a transistor. The area below the MOL is the safe operating area (SOA). A component operating within this region will experience no stress, whereas it will be overstressed outside of the SOA. Exceeding the maximum operating limits will lead to malfunction.

What Derating Is. Because the MOL ratings supplied by manufacturers are calibrated at specific test conditions, engineers often adjust the ratings by some derating factors to suit their application. The derating factor depends on the quality standards of the parts such as military, industrial, and commercial standards, and it also depends on the operating condition in which the design will be used. A designer usually reduces the MOL rating of components by a derating factor to decrease the SOA, so that the component will be designed to withstand higher stress. Figure 9 illustrates examples of derated maximum operating limits for the resistor and transistor.

How Stress Is Calculated. Stress ratio is the fundamental quantity for indicating a stress level of a component. It is defined as

$$R = \frac{\text{Measured value} - \text{Reference rating}}{\text{Derated rating} - \text{Reference rating}}$$

where measured value is the worst case (maximum or minimum) or cumulative (average or rms) or other operating values observed during an analysis, and derated rating is the adjusted maximum operating limits as explained. The reference rating is an offset value to which both the measured value and derated rating are referenced, as in the case of temperature calculations; in most cases it is equal to zero. It is obvious that the value of $R \gg 1$ indicates overstress while $R \ll$

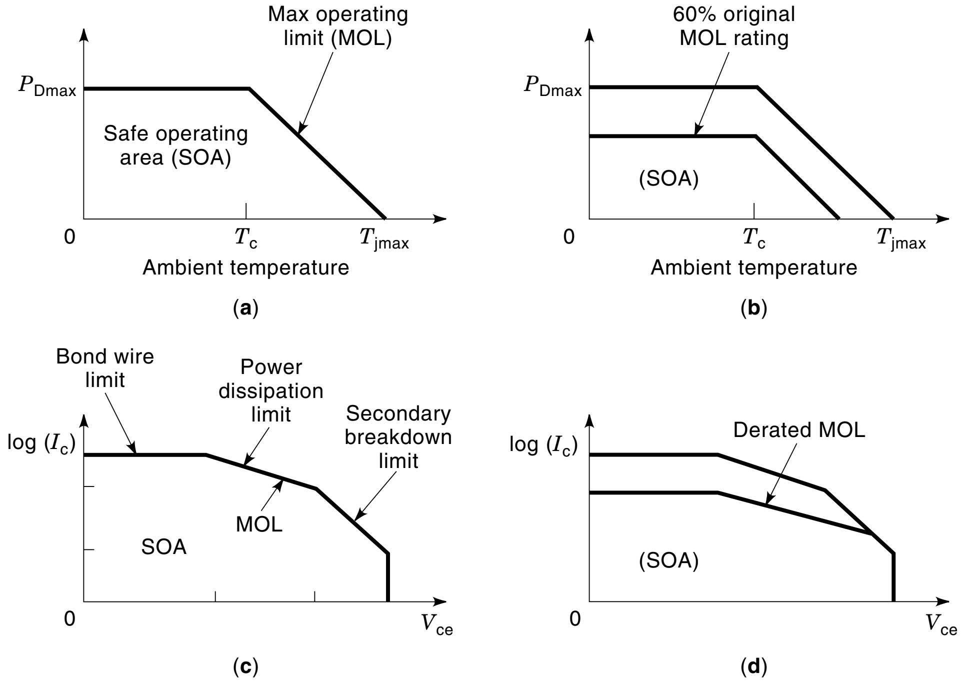


Figure 9. Maximum operating limits and derating of ratings to account for the environment in which the design will be used. (a) Power dissipation rating for a resistor. (b) Sixty percent derating in power dissipation rating implies smaller safe operating area. (c) Maximum current rating for I_c as a function of V_{ce} for a transistor. (d) Maximum current rating derated or reduced so that the stress analysis will select a component that can withstand higher stress.

Table 2. Sample Stress Analysis Report

Components	Derated Rating	Measured Value	Stress Ratio[a], R	Comments
Resistor 1				
Power dissipation	1.44 W	2.00 W	72%	OK
Resistor 2				
Power dissipation	0.12 W	2.00 W	6%	Alert, over designed
Transistor				
Power dissipation	40.0 W	25.0 W	180%	Alert, underdesigned
Junction temperature	250°C	125°C	200%	Alert, underdesigned

[a] The stress ratio points out whether a part is underdesigned, overdesigned, or just right for the application.

1 means understress, and $R \sim 1$ implies that stress is neither overstress nor understress.

A Stress Analysis Report. The sample stress analysis report in Table 2 points out the stress level of components. The stress ratio indicates whether a component has been underdesigned ($R \gg 1$), overdesigned ($R \ll 1$), or correctly designed for the application. The overstressed underdesigned parts can lead to malfunction, whereas the understressed overdesigned parts are unnecessary and can be costly. Stress analysis report checks to see if the selected components are right for the job. As in the sensitivity analysis, the computations in stress analysis can be very tedious, laborious, and time-consuming as well. And as in the preceding case, the key to the analysis is to employ a computer program to automatically generate the stress analysis report for selected components in the design.

ILLUSTRATION OF DESIGN AND ANALYSIS PROCESS

Although a sensible systems engineering approach involves all appropriate engineers (see Fig. 4) at the early and subsequence stages in the development process of a mechatronics system, certain subprocesses, such as design and analysis, may inherently be sequential in nature. The process for dealing with performance requirements, design, and analysis of a mechatronic system is well illustrated by considering a servo-positioning system example shown in Figs. 5 and 6 (24). The servo-system could be part of a product with motion control, such as a robot or vehicle. [See also DC MOTOR DRIVES (BRUSH AND BRUSHLESS).] The example illustrates the following ideas.

Requirements and Specifications Process

This process understands the requirements (needs and expectations) of the users and translates them into specifications that engineers can reference to as guidelines for their design.

1.a. *Functional.* Suppose that the user requirement is to position the output angle of the load accurately and quickly at a reference location specified by the user. A control engineer would translate these nontechnical terms into acceptable technical functional specifications such as settling time, overshoot, and steady-state error, for a step response of the output position. The simulation response diagram in Fig. 5 illustrates the idea. Alternative functional specifications may also be employed.

1.b. *Integrity.* Next also suppose that the user will operate the system at strenuous conditions as in a high-temperature environment and at excessively large varying operating levels. An electrical and mechanical engineer must check the integrity of the components used in the system to safeguard against performance deterioration and system failure.

Top-Down Design Process

This process produces a concept-level schematic to validate the requirements and specifications of the mechatronic system. The design process continues to evolve the concept into an implementation-level schematic with selected components for the system.

2.a. *Concept Level.* The top of Fig. 5 shows the concept-level design schematic consisting of a transfer function block diagram representing a simplified ideal model for the system. Here, a control engineer designs and selects a suitable control scheme for servo-positioning system to achieve the functional specifications. Simulation of the ideal model responses validates that the desired servo-positioning requirement for various conditions of operation is achievable. In this example, a step response is used to the specification as shown in the bottom left of the figure. There are no specific hardware components identified at this initial design stage.

2.b. *Implementation Level.* Once the desired functional response from the conceptual block diagram is achieved, the performance specifications and function characteristics are passed on the electrical and mechanical engineers. The top of Fig. 6 shows the implementation-level design schematic for the proposed system where specific electrical and mechanical components for realizing the servo-positioning scheme have been selected. Simulation of the system at this level confirms that the functional specification is met, within acceptable variations, as shown in the bottom left of Fig. 6. The bottom right of the figure shows the *selected components* for the design. This is the *main result* of the top-down design process. At this stage though, we will refer to the result as the *preliminary* implementation-level design since it has yet to pass the component integrity test.

2.c. *Intermediate Level.* The transition between the conceptual and implementation-level designs would require several intermediate stages of design and redesign iterations. For instance, introducing realistic models of mechanical components would introduce undesirable characteristics such as friction, gear backlash, and shaft flexibility, and it can result in the initial control scheme being no longer acceptable. The control, electrical, and mechanical engineers must rework the design to find a solution to the problem. This may involve several iterations of design before arriving at the *implementation-level* schematic.

Bottom-Up Analysis Process

The selected components at the implementation-level schematic are not the final result of the overall design. These selected components must be subjected to rigorous tests to check their integrity or soundness to ensure that they (1) are not the cause of degradation in functional performance under variation, (2) can withstand strenuous operating conditions, and (3) are realistic parts that can be manufactured, tested, and so on.

3.a. *Component Analysis.* The selected components in the implementation-level schematic for the servo-system are subjected to sensitivity and stress analyses. A *sensitivity analysis* report, similar to the one shown in Table 1, will locate the high-sensitivity components in the design, if any. Where necessary, the control scheme, circuits, sensors, and/or actuators will be redesigned or reselected to produce a more robust implementation-level design. Similarly, a *stress analysis* report, similar to the one shown in Table 2, will identify which components in the design are overstressed, understressed, or normal. Resizing of the components will be carried out to improve the reliability of the design in the case of overstress condition, or to possibly reduce cost in the case of understress condition. Alternative solutions to overstress problems may include, as examples, adding a heat sink to cool electronic components, relief valve to limit pressure, damping cushion to reduce stressful impact, and so on.

3.b. *Manufacturability, Test, and Reliability.* The design information and simulation model are shared among manufacturing, test, and reliability engineers for their review. For instance, the manufacturing engineer may question the commercial availability of certain components in the design and may then suggest alternative standard parts and reduced spending. The test engineer may notice that a study may have been overlooked by the design engineers and may then suggest a re-run of simulations to include the new conditions. The reliability engineer may suggest addition of test points in the design for diagnostic purposes.

3.c. *Trade-Off Decisions.* Conducting the top-down design and bottom-up analysis in the virtual prototyping environment let the engineers find potential problems very early in the stage of the development. Modifications are made via rigorous design and analysis development process. At times, trade-off decisions may require modification of the requirements and specifications as well. At the end of arguments, all parties would end up selecting the "optimum" and right component for the job. The decision at this end will produce the recommended implementation-level design, as the main result of the overall design.

Computer-Aided Engineering Tool

As illustrated and described, the development process for designing and analyzing a mechatronic system employs extensive use of CAE software. From the standpoint of the application engineers, this CAE tool is heaven sent, since they must accomplish the design with limited resource and time. Equally important is the fact that it provides a function simulation "blue print" with which the cross-disciplinary team of mechatronics engineers can communicate and verify their multitechnology ideas. The software that has the necessary tools for dealing with the mechatronics development in this case is the Saber simulator. The simulator is a recognized technique that has been adopted as a standard systems engineering practice in the automotive and aerospace industries.

Environment

The last building block to support the above process is the environment. The necessary technical environment includes computing facilities, work group consisting of experts, and organizational, managerial, and technical supports. Another important infrastructure may involve information technology, whereby the secure use of *intranet* and *internet* makes it possible to share information rapidly among the mechatronics team. The competitiveness in the high-technology business demands such an enabling environment.

Other Examples

The above example was picked for its simplicity and familiarity to a reader, for the purpose of explaining the development process. There are literally hundreds of other examples that follow similar design and analysis process that is aided by the CAE tools. Readers may refer to Refs. 8 and 24–26 for further reading.

EVALUATION

The domain of mechatronics has expanded from simple electronics and mechanics technologies to complex automation, control, and communication technologies with embedded computer intelligence (20). Mechatronic systems are ubiquitous in military, industrial, and commercial applications. They may exist in the form of unexciting but extremely useful products such as factory robots, household appliances, and so on, or the form of exciting systems such as unmanned vehicles for space and remote exploration, as well as military applications. Consumers have benefited tremendously from mechatronic products such as a video camera with full automatic features, automatic teller machines, and the automobiles. It's what we associate with the term "high tech."

It may be reiterated that successful mechatronics endeavors usually stem from a combined application of science, technology, engineering, business, and art. Evidence of these endeavors can be found in innovative use of materials, parts, and better software techniques. Examples are miniaturization of remote control devices, transponders, micromachines, and so on, and use of more sophisticated methods such as fuzzy logic and neural network to enhance original performance of mechatronic systems. The profitable mechatronic product endeavors are the ones that achieve *quality* products, in minimum *time* and *cost.* The systems engineering development process presented here illustrates a means to accomplish this objective.

The path from nurturing a concept to bringing a product into being normally undergoes three stages of development.

1. *Phase 1.* The Basic Research stage, where concept design and analysis are carried out to determine feasibility of the mechatronics concept. This conceptual level stage is the “requirements and specifications” process.
2. *Phase 2.* The Exploratory Research stage, where prototypes are integrated to investigate the feasibility of the mechatronics applications. This stage can be likened to the “top-down design” process to validate that “we are doing the right job.”
3. *Phase 3.* The Product Development stage, which deals with manufacturing process, testing, and reliability issues to bring the product to life. This final stage is the “bottom-up analysis” process to verify that “we are getting the job done right.”

According to the scale of a US Government research funding agency, the ratio of resource funding for Phase 1 to Phase 2 to Phase 3 is approximately 1 : 10 : 30. This illustrates the relative importance of the processes. Many textbooks and articles in the academic literature describe mainly the *functional* performance design process of building mechatronics systems products. They do not emphasize the importance of the component integrity analysis. On the other hand, the practice in the industry heavily emphasizes integrity analysis verification while maintaining functional design validation. This is necessary to ensure the development of high-quality mechatronic products. This is the key point of this article.

Next, one should review the important role of the CAE tool. The philosophy of computer simulation is simple: It’s the ability to predict system performance. With accurate computer models, simulation helps engineers to fully comprehend the problems at hand and enables them to conduct “what if” studies to predict, correct, optimize, and select the right components. The CAE tool used in this mechatronics study was Saber, which is a virtual function prototyping facility. As alluded to in the text, a mechanical CAD tool could be incorporated in the mechatronics study to visualize the motion, the physical layout, the shape, the size, and the color of the mechatronic product. CAD has been adopted in the aerospace and automotive industries. A current trend in the industry is to combine prototypes of virtual functions with virtual mock-ups in a virtual reality environment where a human user can “feel” how the mechatronic product perform, all inside the cyberspace.

Finally, the breadth of disciplines required for a mechatronics project can be quite broad (e.g., electronics, mechanical, hydraulics) and the depth required of a discipline can be quite deep (e.g., details of real-time embedded controller). It is through training and experience that an engineer (from any one of the mechatronics disciplines) will gain sufficient knowledge to lead a mechatronics project and team.

Mechatronic systems and products will keep pace with the progress of technologies and methodologies, and they are here to stay. Mechatronics is the key discipline to the current and future high-tech industries.

BIBLIOGRAPHY

1. *Assoc. of Unmanned Vehicle Syst. Int. Mag., USA,* quarterly issues.
2. *Unmanned Vehicle Syst. Mag., UK,* quarterly issues.
3. *Int. J. Mechatron.*
4. *IEEE/ASME Trans. Mechatron.*
5. *Int. J. Mechatron.: Mech.–Electron.–Control Mechatron.*
6. D. M. Auslander and C. J. Kempf, *Mechatronics: Mechanical Systems Interfacing,* Upper Saddle River, NJ: Prentice-Hall, 1996.
7. W. Bolton, *Mechatronics—Electronic Control Systems in Mechanical Engineering,* Reading, MA: Addison-Wesley, 1995.
8. R. Comerford, Mecha . . . what?, *IEEE Spectrum,* **31** (8): 46–49, 1994.
9. J. R. Hewit (ed.), *Mechatronics,* Berlin: Springer-Verlag, 1993.
10. M. B. Histland and D. G. Alciatore, *Mechatronics and Measurement Systems,* New York: McGraw-Hill, 1997.
11. J. Johnson and P. Picton, *Mechatronics: Designing Intelligent Machines,* Vol. 2: *Concepts in Artificial Intelligence,* London: Butterworth-Heinemann, 1995.
12. L. J. Kamm, *Understanding Electro-Mechanical Engineering: An Introduction to Mechatronics,* New York: IEEE Press, 1996.
13. N. A. Kheir et al., A curriculum in automotive mechatronics system, *Proc. ACE ’97, 4th Int. Fed. Autom. Control (IFAC) Symp. Adv. Control Educ.,* Istanbul, Turkey, 1997.
14. D. K. Miu, *Mechatronics: Electromechanical & Contromechanics,* Berlin: Springer-Verlag, 1993.
15. D. Tomkinson and J. Horne, *Mechatronics Engineering,* New York: McGraw-Hill, 1996.
16. G. Rzevski (ed.), *Mechatronics: Designing Intelligent Machines,* Vol. 1: *Perception, Cognition and Execution,* London: Butterworth-Heinemann, 1995.
17. S. Shetty and R. A. Kolk, *Mechatronics Systems Design,* Boston: PWS, 1997.
18. R. Jurgen (ed.), *Automotive Electronics Handbook,* New York: McGraw-Hill, 1995.
19. D. Knowles, *Automotive Computer Systems,* New York: Delmar, 1996.
20. C. J. Harris, *Advances in Intelligent Control,* London: Taylor & Francis, 1994.
21. P. D. Lawrence and K. Mauch, *Real-Time Microcomputer Systems Design: An Introduction,* New York: McGraw-Hill, 1987.
22. C. T. Kilian, *Modern Control Technology: Components and Systems,* St. Paul, MN: West, 1996.
23. B. J. Kuo, *Automatic Control Systems,* Englewood Cliffs, NJ: Prentice-Hall, 1985.
24. Automotive Applications Using the Saber Simulator, Analogy, Inc., 1992.
25. *Proc. Autom. Analogy Saber Simulator Users Resource,* Livernois, MI, 1997.
26. Stress and Sensitivity Option, Release 3.2, Analogy, Inc., 1993.
27. J. N. Martin, *Systems Engineering Guidebook: A Process for Developing Systems and Products,* Boca Raton, FL: CRC Press, 1997.

KA C. CHEOK
Oakland University