

Concepts of system in engineering

Sunny Y. Auyang

This talk explores three concepts of *system* in engineering: systems theory, systems approach, and systems engineering. They are exemplified in three dimensions of engineering: science, design, and management. Unifying the three system concepts is the idea of *function*: functional abstraction in theory, functional analysis in design, and functional requirements in management. Signifying what a system is *for*, function is a purposive notion absent in physical science, which aims to understand nature. It is prominent in engineering, which aims to transform nature for serving the wants and needs of people.

The engineering system concepts are compared to the notion of system as a seamless web in postmodern technology studies. As Friedrich Nietzsche remarked: “*Being profound and seeming profound.* – Those who know that they are profound strive for clarity. Those who would like to seem profound to the crowd strive for obscurity. For the crowd believes that if it cannot see to the bottom of something it must be profound. It is so timid and dislikes going into the water.” [The Gay Science § 173]

Systems in everyday life, technology study, and engineering

Systems are everywhere in our daily discourse, from the solar system to home entertainment systems, from water treatment to national security systems. Railway networks and electric power grids are systems, so is the internet and the world wide web. Then there are the more intangible political, social, and financial systems. “The system works,” people say when things go smoothly; “the system fails,” when troubles strike. The commonsense notion of system is familiar: the system is a whole comprising interrelated parts with significant complexity; the system is not merely the sum of its parts. There seems to be no restriction on whether the related parts are animate or inanimate, people or things, hardware or software, concrete or abstract. Nevertheless, their interrelations, although complex, should be susceptible to some rational understanding. A muddled mess or an incomprehensible chaos is at best a failed system, if one at all.

“System” is a buzzword in technology studies, although its significance beyond the commonsense notion is obscure. Systems are said to be heterogeneous, but the heterogeneity often turns out to be a jumbling miscellany and the system a sausage with medley indiffereniable stuffing. Perhaps the most prominent notion of “system” is expressed by the metaphor of a “seamless web.” Introduced by Thomas Hughes, the prime promoter of “systems thinking,” it conveys the notions of holism, perfection, and resistance to analysis. It is popular in postmodern scholarship, which is hostile to rationality.

Systems are central to engineering. Engineers are responsible for designing and building transportation, communication, and other systems that operate in the real world. Their systems concepts are clear. Here I explore three: systems theories, systems approach, and systems engineering. None comes close to a seamless web. The systems approach in engineering, with its emphasis on analysis and modularity, is opposite to the anti-analytic postmodern “systems thinking.”

Systems versus seamless webs

In his book *Science of the Artificial*, Herbert Simon gives the parable of two watchmakers, each designs a watch with a hundred parts. In the first design, the hundred parts are so thoroughly interrelated that the watch totally falls apart if any one is removed. The watch is expensive because it must be assembled in one breath and cannot be repaired by replacing parts. The second watch has similar performances, but its parts are grouped into ten modules, which can be replaced if defective. Because it is simple to assemble and easy to modify, the second watchmaker is able to offer his products at a lower price and drive the first watchmaker out of business.

Simon’s first watch design is akin to a seamless web; his second design, an engineering system. A seamless web is good if perfect, but perfection is more often than not illusory. Seamless webs are prone to disasters, because they can be unraveled by the tiniest of loose ends. The effect of a tiny flaw propagating unhindered and creating web-wide havoc is one of the root causes of “normal accidents,” as Charles Perrow called it. To control such disastrous scenarios is a major reason for *modularity*. To limit potential damages, engineers are careful to introduce seams and modular boundaries into their systems. An electric power grid provides an example. Once electricity is fed into the grid, it flows automatically according to physical laws and the conditions of the entire grid. The risk of such seamless configuration is well known. Effects of a small mishap, such a lightning striking out a transformer, can cascade through the grid, leading to power failures in large regions. The 1996 Northwest power blackout and the 2003 Northeast blackout each cost damages in excess of a billion dollars. Engineers tolerated the seamless web not because they deemed it superior but because they could do little about its faults. Until very recently, they had no way to switch high-voltage currents on the grid in real time. Even so, they did try their best to install gates between regions for damage control. New England escaped the 2003 Northeast blackout because the gate designed to disengage it from the grid worked in time – it was saved by a seam.

A system can operate seamlessly without being a seamless web. An example is the internet, which is a patchwork of many networks: landline telephone networks, wireless computer connections, satellite links, and more. All participating networks use packet switching. Otherwise each has its peculiar internal operating principles. A basic design principle of the internet is to preserve as much as possible the autonomy of these internal principles, so that each network can be individually modified and improved. To tie disparate network together in the internet, engineers design routers and protocols at the interface between two networks, so that signals can pass smoothly between them. The routers are seams, good seams. The superiority of

an engineering system lies not in seamlessness but in its good seams or good interfaces between its parts.

Besides the illusion of perfection, the gist of the seamless-web metaphor and associated “systems thinking” is the holistic aura that shields it from critical analysis. The resistance to analysis makes them look profound, but they are basically obscure and muddled, which are major critiques of postmodern studies. In stark contrast, the systematic approaches in science and engineering strive for clarity.

Being profound is different from seeming profound. Tackling with real-world complexity is different from using the word “complexity” to decorate simplistic ideas dreamed up in arm chairs. Simon’s parable is intended to illustrate an approach for designing complex artificial systems that must operate in the real world, but its wisdom applies equally to research in natural science. In principle, the universe is a whole in which everything is connected to all others. Gravity and electromagnetism, the two forces that act between all things above the nuclear level, have infinite ranges. However, if we must treat the universe as a seamless web and grasp everything in it at once, our tiny brains would be so overwhelmed that we would fail to understand anything at all. None of our concepts would be valid, because concepts invariably make distinctions and “carve nature at its joints,” as Plato said.

Science is successful partly because scientists are content to proceed one step at a time, to bite off what they can chew, and refrain from confusing grandiloquent such as seamlessness. Instead of trying to tackle the whole universe at once, scientists take things apart and examine bits and pieces, acknowledging their own limitations. The methods of Socrates were said to be division and collection; Galileo, resolution and composition. Descartes and Newton discusses analysis and synthesis. Engineers practice functional decomposition and physical integration in systems design.

Analysis clarifies. Analysis is also called reduction, and "reductionism" to scientists means the importance of analysis. However, "reductionism" has also become a philosophical dogma asserting that a system is *nothing but* its constituents, e.g., you are nothing but your genes or neurons. Ideological reductionism, which slights synthesis, has engendered much debate in the philosophy of science, debates that generate more heat than light.

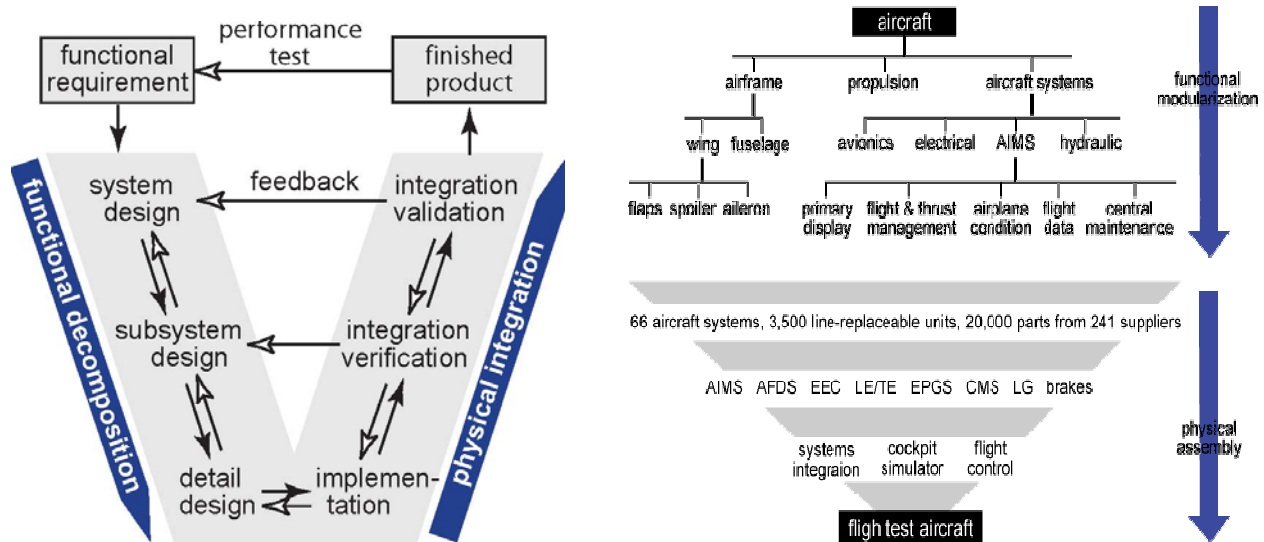
Holism rejects analysis and sees only the whole. Reductionism rejects synthesis and sees only the parts. The systems approach in engineering integrates analysis and synthesis. It brings to relief the dictionary definition of a system as a whole with interrelated parts, emphasizing internal structure.

Systems approach as analysis-synthesis

Let us consider a familiar engineered system, an automobile. When you are driving it, it works as a unitary whole. Any trip to the garage will convince you that it is a system composed of many interrelated subsystems: power train, transmission, ignition, steering, braking, lubricating, suspension, and more. Each of these subsystems is in turn made up of many parts, for instance

the clutch, stick shift, and gear box for transmission. The gear box in turn consists many components, and so on down to nuts and bolts.

Cars are designed by engineers. Their systems approach is illustrated in many textbooks by the *Vee* model, originally due to Forsberg and Mooz. The down stroke of the *Vee* represents functional analysis, the upward stroke physical synthesis.



When engineers want to design a car, they do not start with a bunch of nuts and bolts. They start with the car as a whole. Of course, the car does not yet exist. Thus they start with a conception of the intended car, more specifically, a set of *functional requirements* for it: what it is supposed to do, what performances are expected of it. To ascertain satisfactory functional requirements is usually an important and difficult task, especially when the system is very complex. We will return to it later in the context of systems engineering.

After design engineers have arrived at a satisfactory conception of the intended car, they proceed to *functional analysis*. They decompose the conceived car into functional subsystems with proper interfaces, e.g., a subsystem *for* power and a subsystem *for* transmission, and how the two are to work together. Then they further analyze a subsystem into its interrelated components, until they get manageable parts that can be specified to the last details. They have reached the bottom of the *Vee*. Turning the corner of the *Vee*, the thousands of parts are manufactured to specifications. They are then tested, brought together, and assembled into larger and larger subsystems, finally into a car ready for test drive. We are back at the top of the *Vee*, but this time we have a concrete system – a real car – instead of a mere conception of it.

Through functional decomposition, detailed design, and physical assembly, engineers know and specify the whole, its parts, and their interrelations clearly and at all compositional scales. The subsystems at intermediate levels are crucial for managing complex systems, for they enable engineers to introduce complex details one step at a time.

In short, the *systems approach* integrates analysis and synthesis. It is most effective in treating complex phenomena, for it:

- encompasses both the holistic and modular views
 - grasps the arching features of the whole system
 - analyzes it into parts with proper interfaces
 - synthesizes knowledge about the parts to understand the whole
- grasps the system and its details in many levels
 - decomposes a subsystem into sub-subsystems, and so on, to the last details
 - changes focus to view different levels so the mind is not overwhelmed by complexity
- abstracts and hides information to focus on a task :
 - simplifies the system by treating its parts as black boxes except their interfaces
 - hiding information is *not* discarding it; a black box can be opened at will
- makes a complex system more tractable
 - a part can be studied or designed with minimal interference from other parts
- controls damage and improves safety
 - confines most effects of a defect within a subsystem, preventing a system-wide collapse

The systems approach, in which one analyzes the details of parts in order to know the whole, works not only in engineering but also in natural science. Consider for example systems biology, which aspires to study organisms as wholes. The idea appeared in the 1960s, but lied dormant because the properties of the biological constituents remained in the dark. Meanwhile, biologists analyzed organisms into organs and tissues and cells and molecules. Now molecular biologists have deciphered the genome of many organisms, including humans. They have reached the bottom of the *Vee* and gained tremendous amount of knowledge. Yet they found it falls far short for understanding organisms. Then they turn to synthesis and investigate how genes function in cells and organisms as wholes. As they do so, systems biology springs alive. Several universities, including Harvard here, are establishing departments of systems biology. Four decades of analyzing organic constituents turn systems biology from philosophy into science.

Strategic purviews of systems engineering

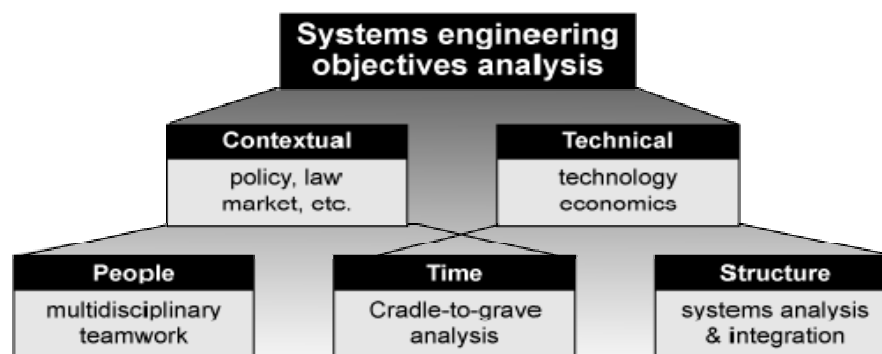
Analysis and synthesis focus on internal structures. A system in this focus is often, but not always, regarded as closed and isolated from the rest of the world. Idealization of closed systems is common in natural science; controlled experiments are designed to realize the ideals as much as possible. On the other hand, the ultimate purposes of engineering systems, be they cars or bridges, are to provide services to some external communities. Thus even when engineers focus on the internal structures of a system, they leave at least a crack at the door. The openings are represented in the system's *functions, input, and output*.

Given a set of functional requirements as ends, the systems approach as analysis-synthesis searches for means to best serve the ends. Systems engineering goes one step further to engage

in *analysis of ends* to determine what functions are required of the system they are charged to design and build. The ends are ultimately decided by various social groups that hold interests in the intended system – clients, as engineers briefly call them. To help their clients in defining feasible and desirable ends, systems engineers negotiate and form working partnerships with them.

Systems engineering arose after World War II to design, develop, and manage complex technological systems with wide societal impacts. Its purview is strategic, taking into account not only a system's life cycle from womb to tomb, but also the people who are involved in various stages of the cycle. Among its missions are:

- ends analysis: requirements engineering,
- life-cycle analysis: concurrent engineering,
- multidisciplinary teamwork: organization for development and management.



Functional requirements: What exactly do you want?

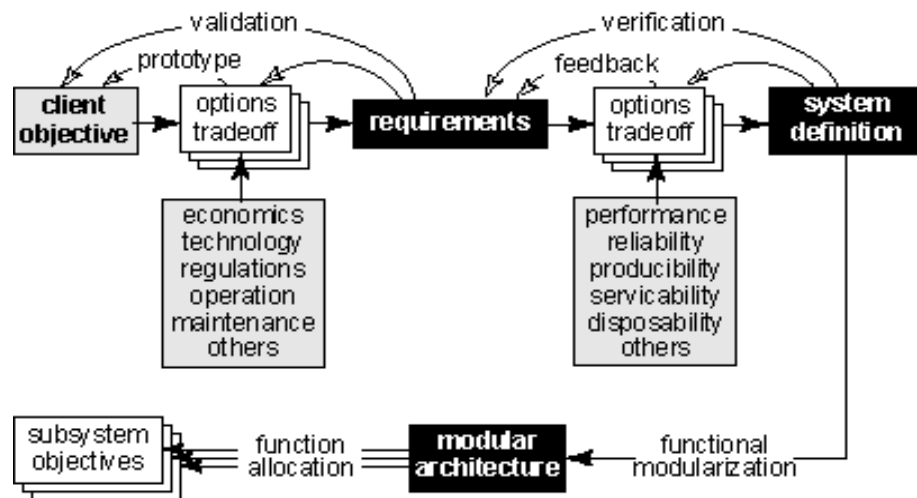
Many engineers avow that the most difficult task of a complex engineering project is to get the requirements right. The task depends crucially on the cooperation between systems engineers and their clients. For complex systems involving novel technologies, people are often uncertain about what they want, or they are unrealistic in their expectations. When the clients divide into several groups, their desires often conflict. It is incumbent on systems engineers to help their clients to clarify their objectives: What exactly do you want? Can you afford it? If not, what options do you have?

Systems engineers consider both technical and contextual conditions. Consider an engineering project, the development of a passenger jetliner. (There are good accounts, by engineers and journalists, on the role of systems engineering in the development of the Boeing 777 jetliner in the 1990s). The major clients are the large airlines, each has its own requirements on range, speed, payload, price, fuel economy, and so on. Besides them, systems engineers also have to consult government aviation policies, environmental regulations, airport facilities, economic and demographic forecasts, and so on. To ensure that the designed airplane can be manufactured cost effectively, traveled on comfortably, operated safely, and maintained easily, systems

engineers elicit opinions from manufacturers, passengers, pilots, flight attendants, and maintenance crews. They then have to make tradeoff among various requirements, for instance between ease of maintenance and lower manufacturing costs. Finally, having coordinated relevant requirements, they design into the airplane affordability, manufacturability, reliability, maintainability, user friendless, disposability, and a host of other -abilities – dispositional properties that they anticipate will function some time in its lifespan. They design *time* into *being*, which is the gist of concurrent engineering.

Stages of system development

- elicitation of cradle-to-grave functional requirements from stake holders
- definition of a system concept that satisfies the life-long requirements
- successive functional modularization of the system concept
- physical integration of components to build the system



An airplane, or your car, is an inanimate thing. But it is more than that, because it also carries interfaces with its intended users. The gentle shape of your car’s seat, the clean layout of its dashboard, are all designed with you in mind. The philosopher Martin Heidegger once contemplated Van Gogh’s painting of a pair of peasant’s shoe and saw in it the meaning of human existence. Use and readiness-to-hand also shine in engineered systems.

An airplane consists advanced materials, high performance engines, onboard computers, navigation and communications systems, and more. The development project of Boeing 777 alone involved some four thousand engineers with disparate expertise from many companies. Just to coordinate their efforts is a major management task for systems engineers. They are responsible for assembling experts in many areas, organizing them into cohesive multidisciplinary teams, integrating their knowledge, and bringing them to bear on the project.

Project management is facilitated by the systems approach of analysis-synthesis. In functional decomposition, systems engineers specify only the functional requirements of what a subsystem, for instance the onboard computer, is supposed to do. Then they let software engineers to figure out how to satisfy the requirements. In this way both parties can utilize their expertise maximally and yet still cooperate tightly. They interact and negotiate on the requirements, but only where it counts, so that they do not suffer from micromanagement and counterproductive interferences.

Systems engineering combines technology development with strategic management, which plans ahead with long-term vision. Since the rise of large-scale industry and giant corporations, engineers had been among leaders of strategic management. They pioneered modern business administration in the nineteenth century and occupy many top executive offices today. Systems engineering attempts to articulate, rationalize, and develop best managerial practices, especially in high-technology areas. It originated in the Cold War and was revitalized in the 1980s, when American industry faced stiff commercial competition from Japan. The Japanese, especially Toyota Motors, are experts in systems engineering. So Americans analyzed Japanese practices and incorporated many of their innovations into systems engineering. The abstraction of general principles from one culture and apply it appropriately in another culture is a mark of scientific thinking, which transcends narrow cultural bounds.

Systems engineering is thriving. MIT and Stanford both created departments for it most recently.

Functional abstraction and systems theories

Systems engineering, which stands at the interface between engineering and management, is conspicuously practical and down to earth. In contrast, systems theories, which lie at the core of engineering science, are mathematical and rather abstract. This in no way implies that systems theories are impractical; they are practical in a general way. Connecting them to systems engineering is the notion of *function*, through which systems theories are applied to particular designs.

Adam Smith, whose *Wealth of Nations* appeared in the same year as his friend Watt's steam engine, observed a relation between machines and abstract systems: "Systems in many respects resemble machines. A machine is a little system, created to perform, as well as to connect together, in reality, those different movements and effects which the artist has occasion for. A system is an imaginary machine invented to connect together in fancy those different movements and effects which are already in reality performed." Smith's remark adumbrated the abstraction from concrete machineries prevalent in engineering systems theories.

In a purposive context, a thing has two kinds of characteristic. The first are physical properties: its materials, structures, and motions. The second are functional characteristics: what it is for, what services it performs. *Systems theories abstract from physical properties and bring to relief functional characteristics*. Not surprisingly, systems abstraction is absent from physical sciences and peculiar to engineering, which aims to design systems that serve people.

We can distinguish two broad types of concepts. Substantive concepts address *what are*, things and their properties. Functional concepts address *what for*. Functions tacitly point to a context, the larger environment in which a system operates and provides services. Functional concepts occur less in physics than in biology, especially evolutionary biology, where natural selection selects adaptable functions. Functional concepts are sort of controversial in natural sciences, because they connote some sense of purpose. Evolutionary biologists who evoke them often refer to engineering; there functional concepts are intuitive, prominent, and successful. Among functional concepts are dispositional properties, which indicate what a thing would do under certain circumstances. Properties such as solubility and flammability occur in physics. They are much more numerous in engineering, whose cradle-to-grave considerations cover properties such as affordability, reliability, durability, manufacturability, and a host of other abilities. Many of such dispositional properties are kind of abstract and intangible. To treat them property engineers have to introduce concepts to define them clearly and performance metrics to measure them. Such concept formation is a kind of scientific thinking.

Functions are paramount in engineering, because engineered systems are designed for service. Thus it is no surprise that they get the center stage of systems theory.

Information theory: an example of systems theory

Examples of systems theories are control theory, information theory, theories for estimation and signal processing. Some systems theories, for example classical control, existed before WWII. But like elsewhere in engineering, a turning point occurred during the war. Mathematician Nobert Wiener introduced two concepts into engineering, probability and optimization, which revolutionized system theoretic thinking. His monograph was called “yellow peril” by engineers, because it had a yellow cover and because its difficult mathematics was overwhelming. But one generation later, engineer scientists became totally comfortable with abstract mathematics. Systems theories are highly mathematical, and their theorem-proof format has more the flavor of pure mathematics than the calculus and applied mathematics familiar in physical theories.

Perhaps the most famous of systems theories is the information theory introduced by Claude Shannon. He opened his classic paper *A mathematical theory of communication*:

“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. We wish to consider certain general problems involving communication systems. To do this it is first necessary to represent the various elements involved as mathematical entities, suitably idealized from their physical counterparts.”

The paragraph highlights the essential characteristics of a systems theory:

1. *Functionality*: the theory is *for the sake* of reliable communication. Reliability is a purposive notion. Shannon gave a brief definition of it here, and would introduce precise metric for it in the paper in terms of probability of errors.

2. *Abstraction*: the theory idealizes from physical properties.

3. *Generality*: the theory is concerned not with particular communication systems but with general principles underlying all kinds of reliable communication systems. Focuses on the general and the particular distinguish between engineering science and engineering design.

4. *Mathematical representation*: the functional relations among communication components are represented in mathematical terms.

Take for example signal processing, a widely applicable systems theory. A signal is something that carries information, and signal processing is the transformation of signals for efficient manipulation, transmission, or storage. Signals come in a variety of physical media: acoustic, mechanic, electromagnetic. Signal processors engage disparate physical mechanisms: electric, electronic, piezoelectric. When you speak on a phone, your information is carried in acoustic signals, which are immediately transformed into electromagnetic signals, which go through modulation, digitization, coding, multiplexing, demultiplexing, decoding, and other signal processing procedures before they are transformed back into acoustic signals heard by your friend on the other end.

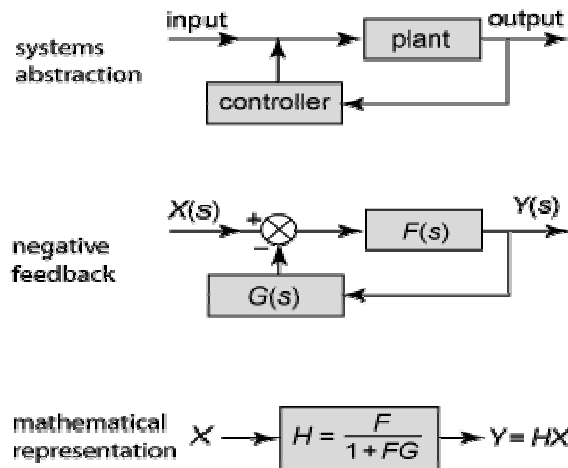
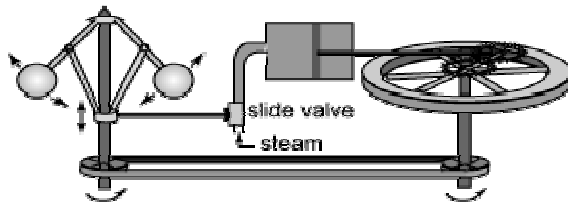
Signal processing theory abstracts from most physical properties, retaining only a few essential features such as the wavelength of signals. It aims to capture, in mathematical terms, the *functions* of signal processors, how they transform the *forms* of signals, for instance the function of digitization transforms a signal from analog form to digital form. Because it captures the fundamental principles covering a wide variety of physical mechanisms, signal processing theory is applicable everywhere from telecommunications to scientific experiments.

Feedback control as an example of functional abstraction

Because systems theories abstract from physical properties, the functional relations they represent can be applied to systems with disparate physical make up. Thus signal processing theories apply equally to acoustic or electrical signals. Mathematical representations enable engineers to manipulate abstract forms, create novel functional relations, and realize them in appropriate physical media.

We can see how system theories work in the case of feedback control. Although the term “feedback control” was introduced in the 20th century, feedback devices had been made for many centuries. One famous example is Watt’s flyball governor, which regulates the steam engine so that the flywheel operates in a steady pace. In systems abstraction, we forget all the physical mechanisms and concentrates on the functional relation between a plant, the flywheel in this particular case, and the controller, which is the flyball governor. The plant and the controlled is each represented by a mathematical expression, F and G.

Now let us add a twist, negative feedback, where the feedback control is *subtracted* instead of added to the input. Negative feedback was invented by Black for achieving low distortion amplification of electrical signals, especially for long distance telephony. Physically, it is totally different from mechanical system. However, we can still represent the amplifier as the plant, F , and a feedback control as G , the only difference is that we have a negative sign here. And we get a simple algebraic expression for the relation between the input and output through negative feedback. The problem for signal amplification is that the amplifier, F , is distortive. However, we can see that if the gain G of the controller is larger, so that FG is much greater than one, then the two F s cancel out and the output is essentially independent of behavior of the amplifier. When you put in different specific forms of F and G , you can calculate the operational characteristics of a specific feedback control system. But even without the specifics, control theory can tell you a lot.



Systems abstraction

- abstract from physical properties
- uncover functional principles under disparate physical systems
- introduce concepts to represent essential characteristics
- represent mathematically and explore logical consequences, bounds, limits, optimality
- invent new concepts and systems
- e.g., feedback control from Watt's flyball governor to negative feedback amplifier

Employing different levels of abstraction to represent complex phenomena, focusing on the topic of interest at hand, is another common method in the natural science. It is used here in engineering science.

Systems in engineering science, design, and management

I have presented three notions of systems. They play major roles in three engineering activities. Systems theories with their systems abstraction are prominent in engineering science, which seek

general functional principles. The systems approach as analysis-synthesis is widely practiced in engineering design of particular systems. Systems engineering strives to rationalize the management of large technical projects and their interfaces with society at large. Their domains, however, are not exclusive. Systems theories are routinely used in design and management. Analysis-synthesis is a time-honored approach in scientific research generally. The broad purviews of systems engineering are sipping into many development projects.

Sunny Y, Auyang

Part of a talk presented at Department of the History of Science, Harvard University

October 28, 2004.