# On Computational Synthesis of Complex Electromechanical Systems
## *Position paper*

**Therani Madhusudan**
MIS Dept., Univ. of Arizona
Tucson, AZ 85750
madhu@email.arizona.edu

### Abstract

Developing new approaches to aiding computational synthesis of modern electromechanical systems is a major need. Current techniques use product representations that reason with single abstractions such as either geometry or physical dynamics. Further, these techniques are utilized in the context of static design processes. This article proposes the development of computational frameworks wherein both the process of design along with the product being designed are reasoned with in an integrated manner. Developing such a framework would require advances in product models that integrate geometric and behavioral abstractions. Further, development of new process models would require integration of planning and machine learning techniques that reason with these new product representations. An integrated framework would aid in the development of better cost-effective synthesis tools and allow for assimilating and reusing many kinds of design knowledge. Potential approaches towards developing such a synthetic framework are outlined.

## Introduction

Design of modern electromechanical systems involves the configuration of a wide variety of building block components into an interconnected topology. For a given set of design requirements, many topologies are feasible as solutions. Concomitantly, the design process of generating these feasible topologies is also varied. The process involves cognitive reasoning steps such as idea generation, refinement, search, modeling, testing to name a few. The final result of the overall design activity, the final assembly, is shaped by the *complex interplay* between these cognitive reasoning steps and the underlying models of the building block components. Current approaches to facilitating computational synthesis are primarily focused on improving the underlying modeling abstractions of the building blocks. Our current understanding of the interplay between the design process and the evolution of the design artefact per se is rather minimal. Our current knowledge regarding interaction between design reasoning tasks and the underlying model are highly limited and very specific to the example domain/task under consideration.

The building block components that form the basis of modern electromechanical systems are differentiated along multiple abstractions (Whitney & others 1994)such as:

- *Physical instantiation:* The component is either hardware (usually the basic components that provide the functionality) or software (usually the controlling logic that integrates the subcomponents).
- *Model-type:* The hardware can be analog or digital components, lumped parameter or distributed parameter models, deterministic models or models with various levels of uncertainty considerations in input-output behavior. These model types usually have a linear or non-linear model formulation characterizing the generic behavior of the component.
- *Function:* The functionality of a component is primarily due to its power or signal processing capability, or its material construction or due to its spatial geometry or some combination of all of the above. Modeling functions is a major research area and a variety of functional classifications have been proposed in the literature.
- *System-type:* The assemblies constructed from these basic components can provide their functionality based on the energy, material and information flows while interacting with the environment. Basic systemic models such as *open or closed* with conservative or non-conservative flows (of energy, material, information) across the system boundaries. Modeling the interaction of the assembly with the environment is a complex issue.

For each of the above models a variety of computational representations exist in the literature to reason about specific aspects of the abstraction. For example B-rep and CSG representations to reason about spatial aspects of a design. However, the task of reconciling inconsistent inferences from each of these models when reasoned with independently is left to the designer. Computational frameworks that combine these multiple abstractions in an integrated manner are needed for furthering the state-of-art in computational synthesis.

In a similar manner, the process of design has also been classified into multiple stages(Finger & Dixon 1989), namely, a) design generation phase consisting of conceptual design, parametric design and configurational design and b) design evaluation consisting of optimization based

---

evaluation, manufacturability based evaluation and other criteria-based schemes. A generic generate and test enumerative computational approach forms the backbone of most of the synthesis techniques. Further, for each of these subphases of the design process a variety of computational frameworks have been suggested ranging from genetic algorithms search, constraint satisfaction based search, grammar-oriented approaches, system modeling oriented approaches, optimization oriented schemes and rule-based approaches in the literature. These techniques are illustrated in the context of a particular set of abstractions of the building blocks mentioned above. Further, many of the computational schemes embed a variety of heuristics that guide the process. Each such technique considers a restricted set of inputs and provides decision-making guidance in a fixed content. However, during real-world design, the design process evolves whenever knowledge is uncertain or incomplete. Knowledge gathering activities are interleaved with decision-making activities. The point is the design process adapts during design to ensure satisficing of design requirements and optimization of design criteria. Approaches to synthesize the design process during the design task are essential before design tools really aid the designer in exploring the design space in a beneficial manner.

This paper advocates the development of a computational framework to support engineering design that focuses on advancing component representations at multiple levels of abstraction and concurrently, study the process of decision-making during design that covers for incomplete and inconsistent models, erroneous requirements and allows for inducing new knowledge based on experimental data. Two major aspects need to be addressed, development of computational representations with better fidelity to the real world and development of computational processes that exploit these representations and extend the same in a dynamic manner. Developing better representations requires an improved understanding and modeling of physical phenomena. Development of better process models requires an understanding of the task of composition defined at multiple levels of abstraction and also the ability to acquire new knowledge from past experiences or learn via model development from experiments. A key approach to developing better product representations is based on new developing new combinatorial topological algebras that combine both geometry and physical phenomena in an integrated manner. Reasoning using such representations should incorporate the planning-oriented ideas for synthesis of new processes and machine learning approaches either to model new aspects of the product or the design process. Further, to facilitate intelligent search in these large search spaces, the evaluation criteria need to consider complete life-cycle constraints such as those from manufacturing, usage, serviceability etc.

The ideas outlined in this article have evolved during the course of the authors dissertation studies on facilitating computational synthesis for electromechanical systems(Madhusudan, Sycara, & Navin-chandra 1996; Madhusudan 1998). The rest of the paper is organized as follows: The next section provides an overview of the process of electromechanical systems design and highlights the open issues in achieving this integrated framework followed by an outline of potential approaches to resolving some of the open issues. Concluding remarks are in the final section.

## Overview of Electromechanical Systems Design

Electromechanical systems design is a knowledge intensive activity. I present an example of electromechanical design to highlight the interplay between product and process representation and reasoning. Consider the synthesis of a system that drives a machine tool slide relative to a cutting tool as shown in Figure 1. The cutting tool interferes with the surface of the workpiece and metal is removed from the workpiece surface during the forward stroke of the machine tool slide. No interference takes place between tool and workpiece during the return stroke. The cutting tool is lowered or the machine tool slide is raised through a fixed distance to ensure interference between the cutting tool and workpiece, before the next forward stroke of the machine-tool table. The only concern is with designing an assembly that *provides the reciprocatory motion* for the machine-tool slide. For this discussion, the synthesis of an indexing mechanism that lowers the cutting tool or raises the machine tool slide between every cycle consisting of forward and return strokes is *not* addressed.

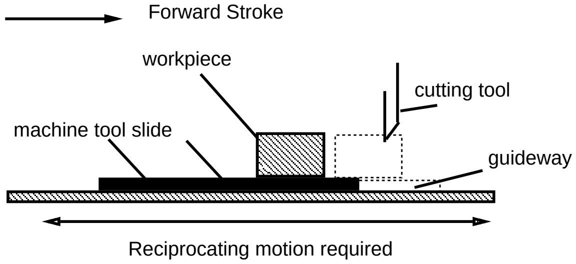

Figure 1: Machine tool slide specification

Design requirements include guidelines such as a) the drive system for the machine-tool slide must provide enough power to overcome the cutting force due to interference and the frictional force between the machine-tool slide and its guide ways. b) The force on the machine tool slide remains constant in the early part of the forward stroke and there is a gradual increase of the force to a peak value and thereafter, a gradual reduction of the force to a constant value during the cutting process. The constant force is the frictional force resisting machine-tool slide motion. The time-history of the velocity of the machine-tool slide may be specified for the forward and return strokes depending on the nature of the surface finish required for the workpiece surface. c) Design specifications also require that it should also be possible to operate the machine-tool slide at different stroke frequencies depending on the nature of the cutting operation. Rough cutting of workpieces can be performed at high frequencies (short forward and return strokes at high velocities) while finishing of metal surfaces are performed at slow speeds and low frequencies. The design requirements provide numerical values and time-histories for the cutting forces, frictional forces, velocities for the machine-tool slide and the frequencies of the strokes. d) It is also assumed that the power source for the machine-tool drive system is a wall-outlet and is to be driven independently of other power equipment in the machine tool though it is possible that the power for the machine-tool drive system comes from a gear-box in the machine tool. The wall-outlet provides a maximum voltage and there is a bound on the current that can be drawn from the outlet without blowing a fuse. The wall-outlet may provide an AC or DC power supply. e) It must also be possible to manually override the system when the cutting tool and workpiece are jammed. Facility must be provided to interrupt the system for brief periods to adjust workpiece location, replace worn out cutting tools, add cutting coolants *etc.* and resume complete operation quickly.

A computational language to represent these specifications is needed. The specifications describe aspects of geometry and physical dynamics of the final design. Operational conditions and life-cycle maintenance criteria are outlined. Implicit in these specifications are interrelations between geometry, material and physical dynamic characteristics of the final design. Current approaches may provide separate structured models for geometry and dynamics and unstructured (possibly textual descriptions) for material and manufacturing requirements.

A variety of electro-mechanical assembly alternatives can be configured to meet the above requirements as shown in Figure 2. Each of these systems is assembled from a set of components, each component with a specific dynamic behavior. In configuration (1), an electrical motor is connected through a shaft to a clutch and the clutch is connected to a slider-crank mechanism. The slider of the slider-crank mechanism is rigidly connected to the machine-tool slide. The clutch provides the ability to manually override the system or provide brief interruptions during operation. The electrical motor can be an AC motor or DC motor depending on the nature of the wall-outlet. The slider-crank mechanism converts the rotary motion of the motor shaft to linear reciprocatory motion of the machine-tool slide. The thick arrow denotes the direction of *power flow* in the system. In configuration (2), the slider-crank mechanism is replaced by a Whitworth quick-return mechanism. The quick return mechanism has different velocities and time durations for the forward and return strokes. In configuration (3), the quick return mechanism is replaced by a rack and pinion mechanism. The rack is rigidly connected to the machine-tool slide. Forward and return strokes are provided by changing the direction of current flow to the motor and thus the direction of rotation of the shaft connected to the pinion. The thin arrow connecting the rack and pinion mechanism, the position sensor and the direction changing switch denotes the direction of *signal flow*. The signal provides information on the machine-tool table position to the switch. When the forward stroke is completed, the position sensor tracks the position and triggers the switch and the return stroke begins. In configuration (4), the rack and pinion is replaced by a a lead-screw mechanism. The change in direction of rotation for forward and return strokes is obtained through a gear-box that provides two directions of rotation. Position information is used to change gears in the gear-box and change direction of rotation. In configuration (5), an electric motor is connected to a pump and the pump is connected through a bi-directional valve to a double-piston cylinder mechanism. Forward and return strokes are obtained by changing the direction of oil flow through the valve. Position information on machine-tool slide is used to change the settings of the bi-directional valve. Configurations (1) and (2) are open-loop systems while (3), (4) and (5) are closed-loop systems. A variety of feasible component topologies are generated during the initial concept design phase of an artifact. This phase of design wherein these component topology alternatives are created and compared with each other is called *Conceptual design*.

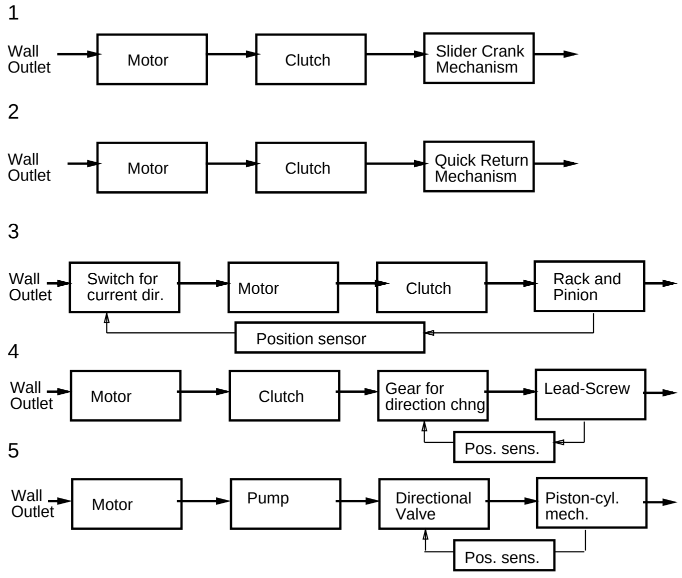

Figure 2: Topologies for Machine-tool slide drive system

Conceptual design is an open-ended design activity wherein the basic framework of a design is laid out. This design phase allows for the generation of innovative solutions and establishment of potential interconnections between multiple levels of abstractions. Further, during this phase, the design guidelines may be refined and altered as potentially unsolvable problems are identified and ambiguities in the requirements resolved. Further, during this phase, a variety of subtasks may be executed that are aimed at gathering knowledge (exploratory and diverging) and integrating knowledge (convergent and refining). Supporting conceptual design computationally requires the ability to reason at high levels of granularity and extrapolate from incomplete information. Further, key features of the product need to be isolated and refined early. For example, it must be possible to identify if the design is even physically plausible. Capturing and utilizing product and process knowledge, developing computational processes for making *connections* between disparate nuggets of information and possibly discovering *new* solutions are essential.

Generating a topology of possible components is not enough to satisfy a given set of design specifications. In topology (1) of the foregoing example, a motor that generates the requisite amount of torque to drive the slider-crank mechanism may not be available from a manufacturers' catalog. Even if such a motor is available, its current requirements might overload the wall-outlet. Assuming a motor that meets all torque and current conditions is available, it may be that a slider-crank mechanism with the requisite range of reciprocation may have to be built instead of using one off the shelf. *The point is, a topology of components may be rendered infeasible because of the parametric requirements of the specification.* In configuration (5), a piston-cylinder mechanism with requisite piston cross-section area may not be available off the shelf. To validate a topology of components, one has to check the parametric aspects of each component. Motors are available from catalogs for particular horsepower requirements and rpm specifications. In configuration (1) of Figure 2, if a motor with a higher rpm than the minimum rpm required is chosen, a gear mechanism is required to reduce speed from the motor shaft to the drive shaft of the slider-crank mechanism. This adds another component to the initial topology and we obtain the topology in Figure 3. Searching for available component's based on parametric requirements and choosing compatible components is the *Parametric design*.

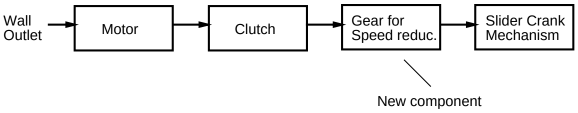

Figure 3: Refined topology for Motor slider-crank assembly

Parametric design is a well-understood phase of design and is routinely practiced on a large scale. However, the choice of the parameters is done based on models that idealize the real world phenomena via models. Consideration of multiple parameters in a concurrent manner and accounting for uncertainty in parameter values is a major problem. Further, reasoning about the spatial effects of parameter choices and interactions is usually performed via costly simulations. Fast, reliable approaches to understanding and evaluating parameter choices are needed. Further, another open issue in parametric design is utilizing experimental data regarding phenomena embedded by the components. An open issue is the reconciliation of discrete and analog model parameter choices. Current approaches rely on a trial-and-error approach guided by experiments. With the advent of embedded systems and the wide-spread use of software-driven controllers, developing robust parameter choice algorithms is essential.

Choosing parameters for components in a given device topology does not conclude the design process. In the machine-tool drive system, there may be spatial specifications that dictate the orientation of shafts between the different components. In topology (1) of Figure 2, the motor shaft is perpendicular to the direction of reciprocation of the slider-crank mechanism. A collinear shaft may be obtained by use of a worm gear assembly to convert the motor shaft rotation into a perpendicular direction for crank rotation in the slider-crank mechanism. A design requirement that specifies the volume of space that may be occupied by the machine tool drive system may be used to trade-off the different possible topologies. Since hydraulic systems require large space, we may not consider topology (5) if a small volume of space is a design requirement. This phase of design wherein spatially dependent decisions are made is called *Configuration design*.

Configuration design is one of better understood phases of electromechanical systems design. With the advent of robust solid model representations and visualization capability, manual design generation is well-advanced. However, generation of varied forms from generic regular polyhedra that meet design requirements is an open issue. Form design is not well-understood. For example, the utilization of a metal beam to convey electricity (of a certain voltage) and also support a certain load requires a particular cross-sectional configuration to avoid power leakage and fatigue failure. Design process knowledge of how to meet such multi-criteria requirements is minimal. Further, computational techniques to aid and guide these processes are required. A related problem is the generation of spatial orientations to meet a wide variety of requirements ranging from manufacturing assembly through product maintenance and repair.

The design phases outlined above assume ideal dynamic behavior of the components in real-life operation. Some of the issues that need to be considered to further refine the designs are:a) Inertial and vibration effects. In the machine-tool slide system, the acceleration and deceleration forces of the slide during high-frequency operation have a critical effect on the cutting process.b) Non-linear effects such as elasticity, hysteresis, friction *etc.* affect the dynamic behavior of mechanisms. The shafts transmitting power between components in the device topology of the machine-tool slide system have an inherent flexibility and absorb some energy by flexing. c) Extraneous energy losses as heat and noise affect the behavior and accuracy of the system. In the different device topologies of the machine-tool slide system, heat loss due to friction and hysteresis along the power flow path leads to cooling and heat dispersion problems. d) Manufacturing and assembly concerns with regard to the final behavior and functionality of the electro-mechanical system. Consideration of these effects with detailed analytical models using FEM and other related CAD tools can further refine the device topology. As a final validation, a prototype has to be built and experimentally studied to understand the behavior of the system. A given design can only be verified by prototyping an artifact. Many of these issues are currently done manually and much of the knowledge is experiential in nature. Mathematical models for many of these interacting phenomena are current unavailable or extremely complex to develop.

Evaluation and choice of the design alternatives is also made based on a variety of economic perspectives. From a production management viewpoint, those designs are chosen which are compatible with the rest of the products *i.e.*

for the foregoing example, other kinds of drive systems and components already in production. This enables one to share components (through interchangeability[1]), reduce inventories, share manufacturing tools and also manage costs better. Further, decisions regarding outsourcing or making components in-house are made based on the core technologies of a company and also the capital investments required. Other considerations involve support and service requirements for the different types of designs. Products that share similar design structures require less training for personnel already trained in maintenance of early product designs. Further, in the category of industrial durables and high-tech products, quality and product reliability is of prime concern since many subsystems are combined into complex systems to provide end-user functionality. An open issue is the integration of economic abstractions with the engineering abstractions. Though advanced information systems are being deployed in organizations, it is unclear how to use costs and operational statistics in the development of the design.

The process of design during all these phases varies based on the nature of the electromechanical assembly and its primary functional requirements. The process is complex when the technology or application is new or the requirements extremely rigid. Further, if the solution requires knowledge from multiple disciplines, the design process may involve many knowledge gathering and integration phases before beneficial progress is made towards a feasible solution. The design process is also modulated in the real world by the experiential knowledge of designers. From a computational perspective, tools that capture design processes and allow for search, retrieval, adaptation and reuse when needed during the design phase will be highly beneficial. Additional computational tools that function as information gatherers and facilitate fusion of information (both structured and unstructured) are needed. Such tools may enable the design process by locating new pieces of relevant information or draw new inferences from the currently available knowledge. The interaction between Conceptual, parametric, configuration and the later analysis phases is least understood. After each stage of design, different design options need to be verified with respect to the requirements. Parametric information is needed to compare and verify different feasible device topologies by lumped parameter dynamic simulation. Validation of a design for spatial specifications requires configuration and embodiment information. A change in configuration may change parameters of a component and that change may invalidate a device topology and a different device topology may have to be chosen. How should the overall design process be controlled? Should one always begin with conceptual design? How much of the requirements should one before parametric design? Can these design phase be run concurrently and if so how? How should the design process be executed, when parts of the design (for example certain components in a topology) are fixed and cannot be changed? How can reverse engineered knowledge from competing products be used? The dynamics of the design process are least understood.

Conventional wisdom indicates that software design tools can improve design processes by providing relevant design information at different decision points to designers during synthesis. Present day support tools exist at the latter stages of design *i.e* shape–form design and analysis. Primarily these tools streamline the process of gathering, collating, monitoring and distributing information but are *not active participants* in the process of generating new information and new insights. Design reasoning in the early, unstructured stages is least understood.

Design of real world systems usually involves consideration of all the above-mentioned product and process dimensions. The conventional process of design especially in domains dealing with the physical world and dominated by geometric/spatial considerations (primarily non-VLSI) have evolved engineering design practices and guide books which are a combination of experiential knowledge, experimental test data, analytical models and domain specific rules-of-thumb. Further, this design search space is influenced by additional non-physical constraints based on economic and marketing models. Composition of building blocks (as performed by designers) is driven by both causal (directed) reasoning and a trial-error approach consisting of simulation and physical testing. Supporting these tasks computationally requires advances in both product and process representation and reasoning.

## A possible approach towards an integrated framework

As a first step in our effort towards an integrated framework, the author has explored an integrated approach wherein detailed design process models along with detailed product models (based on an energy-based modeling approach called bond graphs(Paynter 1961; McAdams & Wood 2002; Stone & Wood 2000)) were combined in a computational process called "Elaboration" to generate topologies of how to connect component building blocks. Elaboration is a graph generation search procedure guided both by topology selection rules and an experiential knowledge-base of "working topologies" created by reverse engineering of successful systems. Specifications are provided in terms of behavioral language that describes the input-output temporal behavior of physical parameters. Topology generation is followed by parametric sizing analysis and topologies are validated based on analytical simulation and by visual inspection by designers. Though the technique generates a wide-variety of feasible topologies for a given specification, and identifies analogical designs etc., the following observations were made during studies in real-world contexts:

- Much of design reasoning seems to be dominated by consideration of "functions" that are based on geometric form. Topology generation needs to be supported with geometric reasoning to provide a visual basis. The search space characterizing such geometry-based solutions is extremely large and much of real-world design reduces this search space based on previous successful product geometries or extrapolation based on results from

---

[1] Functional interchangeability is being referred to here in contrast to the manufacturing notion of parts interchangeability.

linear physical models. Acceptance of topologies is feasible only after geometric considerations i.e a top-down design process has to lead all the way to feasible physical realizations.

- Designers are primarily occupied with considerations of multiple energetic (material flows may be treated distinctly) and information flow interactions in various geometric configurations. Much of the design effort is spent in refining a preliminary topology. Further, the process is also influenced by the phenomena of function sharing wherein spurious or interrelated effects are put to functional use to minimize the overall physical package. Major design resources are spent on minimizing and understanding these secondary effects.
- The spatial optimization process is further complicated by considerations of interactions between both analog and discrete component - leading to design of complex control laws to regulate the system behavior. These laws are again primarily identified via experimental testing.

Approaches need to be developed to integrate geometry oriented reasoning with an understanding of the underlying physical processes and provide a physical basis for the computations. Further, these computations need to be conducted at various levels of granularity with the currently available information during the design process.

Our current work is exploring improving product representations based on chain models of physical systems(Shapiro & Voelcker 1989; Palmer & Shapiro 1993) (based on algebraic topological considerations) wherein both spatial and energetic considerations can be profitably combined and both process/product perspectives treated in a unified manner. Further from a dynamics perspective, integrating hybrid system models in facilitating design validation is a major issue. State space models of discrete phenomena need to be integrated with models of continuous phenomena. Much of the mapping knowledge is currently ad hoc. New mathematical frameworks are required to explore these possibilities and also consider issues such as uncertainties in product properties and effects of variability induced during manufacturing.

Current design process information is implicitly embedded in the tools as imperative algorithms. Explicit representation of such processes and approaches for their composition from primitive processes and adaption of processes retrieved from process repositories are needed. Towards these end we are developing planning based process repositories and utilizing Case-based reasoning(Kolodner 1993) techniques for their composition and reuse. These repositories are being built by observing designers and design teams and also based on case studies of product design. Much of the design process knowledge is proprietary and is difficult to obtain. However, we are currently studying trade journals and public design handbooks for guidelines regarding the design processes. A key aspect of the design process is the use of downstream information to guide design choices. Consideration of capabilities of modern day ERP systems and utilization of information available in these systems is an important step during design. How to develop and adapt the design process as resource availability changes in an organization is an important issue to consider as design processes are developed. Our current state of knowledge regarding design processes is minimal. Observations in a few industrial contexts have highlighted that engineering design is a highly knowledge/experience driven activity with much of it being routine versus innovative. Innovation can occur at any level of abstraction. Mapping the design search space at multiple levels of abstraction concurrently and the process of navigating between these search spaces is important to develop useful design tools that can acquire knowledge from designers and in turn support their tasks appropriately. Ethnographic and observational studies in design organizations are needed on a large scale before the design process can be well-understood. Storage, reuse and adaptation of such experiential knowledge on a large scale is a viable approach towards robust process design.

## Concluding Remarks

Design of electromechanical systems is an iterative activity with iterations between the different design phases and even within one phase. Design could begin with spatial embodiment information regarding a physical component and proceed in a bottom-up manner instead of the top-down scheme described in the above example. Further, decisions regarding a product involve both economic and engineering constraints considered together and specific steps are usually never delineated in detail. Some socio-economic constraints. A study of product design history (Petroski 1992; ame 1995; Glegg 1969) indicates that the design process is very complex and highly unstructured. Supporting such a task with computational tools requires one to understand *what subtasks of the design task can be computationally supported, what pieces of knowledge are available, what and where are the knowledge gaps and what kinds of tools and methodologies can be used to support the development of computational tools.* This article outlined the need for an integrated framework wherein both the product and process are designed in an integrated manner.

Our current work is focused on developing better product and process models. Much of the research in design has focused on product design versus process design. Our perspective is to view products as carriers of physical processes. The computational process of design can then be viewed as a process of assembly of physical processes wherein their carrier-units (the product) need to satisfy a variety of spatio-temporal physical constraints. Computational synthesis may be benefit if we focus on providing a framework to integrate the already available abstractions into a well-knit reusable framework.

## References

1995. *American Inventions*. Barnes & Noble, NY.

Finger, S., and Dixon, J. R. 1989. A review of research in mechanical engineering design. *Research in Engineering Design* 1(1).

Glegg, G. 1969. *The design of design*. Cambridge.U.P., London.

Kolodner, J. L. 1993. *Case-Based Reasoning*. San Mateo, California: Morgan Kaufmann Publishers.

Madhusudan, T.; Sycara, K.; and Navin-chandra, D. 1996. On synthesis of electromechanical assemblies. In *ASME DTC*. ASME.

Madhusudan, T. 1998. *On Synthesis of Electromechanical Assemblies-Automated Generation and Evaluation of Design Alternatives*. Ph.D. Dissertation, Carnegie-Mellon University.

McAdams, D. A., and Wood, K. L. 2002. A quantitative similarity metric for design by analogy. *Journal of Mechanical Design* 124.

Palmer, R., and Shapiro, V. 1993. Chain models of physical behavior for engineering analysis and design. *Research in Engineering Design* 5:161–184.

Paynter, H. M. 1961. *Analysis and Design of Engineering Systems*. Cambridge, MA: MIT Press.

Petroski, H. 1992. *The evolution of useful things*. Knopf, NY.

Shapiro, V., and Voelcker, H. 1989. On the role of geometry in mechanical design. *Research in Engineering Design* 69–73.

Stone, R., and Wood, K. 2000. Development of a functional basis for design. *Journal of Mechanical Design* (122).

Whitney, D., et al. 1994. Problems and issues in design and manufacture of complex electromechanical systems. Technical Report CSDL-R-2577, Charles Stark Draper Laboratory, Inc.

This article surveys the System Level Synthesis framework, which presents a novel perspec-tive on constrained robust and optimal controller synthesis for linear systems. We show how SLS shifts the controller synthesis task from the design of a controller to the design of the en-tire closed loop system, and highlight the benefits of this approach in terms of scalability and transparency.Â Throughout, we emphasize practical and efficient computational solutions, and demonstrate our methods on easy to understand case studies. 1 Introduction. The systems we seek to design and control are becoming increasingly complex, be it in their dynamics, their scale, or their interaction with the environment.