1990

Form-function characteristics of electron-mechanical designs

J Rinderle
Carnegie Mellon University

Carnegie Mellon University.Engineering Design Research Center.

Form-Function Characteristics of Electron-Mechanical Designs

by

J. R. Rinderle, E. Colburn, S. Hoover, J. Paz-Soldan, J. Watton

EDRC 24-29-90 £, $

Form - Function Characteristics of Electro-Mechanical Designs

James R. Rinderle
Associate Professor

Eric R. Colburn
Stephen P. Hoover
Juan Pedro Paz-Soldan
John D. Watton
Graduate Research Assistants

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

The Nature of the Mechanical Design Problem

Mechanical designs are solutions to multi-faceted problems. In this regard, mechanical design is no different from most other engineering design disciplines. However, in many design disciplines such as software design and circuit design, designs can be characterized as collections of weakly-interacting functional modules, each of which implements one of the functional requirements. In these domains, good designs can often be accomplished by successively decomposing requirements until the lowest level requirements match the behavior of some preexisting design component. Direct transformation and recombination can then be used to complete the design. On the other hand, good mechanical designs are often highly-integrated, tightly-coupled collections of interacting components because the cost, size, and weight of mechanical components makes a direct application of the decompose and transform strategy impractical. In well designed mechanical devices a simple correspondence between specific functional requirements of the product and individual components in the design does not usually exist. The converse is also true, i.e., a specific component does not contribute to a single function of a product.

We might summarize by saying:

- The form-function relations in mechanical design are complex.
- Function is not isomorphic to form.
- Design strategies based on functional decomposition and direct mapping into physical components are not generally applicable in mechanical design.

The absence of a close correspondence between functional and physical descriptions in mechanical design has important ramifications on the design process and on the efficacy of designs themselves. It is the designer's task to specify a *form* to satisfy constraints on *function* and *fabrication*. In light of this task, and the complexity of the form-function structure, we might ask what makes a good designer? One attribute, among many perhaps, is that good designers understand and take advantage of the subtle relationships which exist among function, form and fabrication for a mechanical device. If we are to reduce our dependency on the intuition of experienced designers we must also extract and utilize the relationships between form and function in designing a product.

Toward that end we assert that form-function characteristics of components and devices exist and can be identified from physical principles. Furthermore, we believe that design decisions are dominated by considerations having to do with relationships among form and function.

In our program of design research we have embarked on four areas addressing these basic questions. Firstly, we are seeking to identify the form-function characteristics of components and to abstract high level design relationships. Secondly, we are aggregating component level behavior to device level in a way that will consider the context and resolution of the design stage. Thirdly, we are seeking to identify the dominant design relationships which arise as a result of this aggregation and fourthly, we propose a synthesis strategy based on the opportunistic utilization of component form-function relations. Each of these four topics is discussed briefly in the sections which follow.

Identifying Form-Function Relations for Physical Systems

Much of the design process consists of selecting, sizing and configuring standard components into systems. During preliminary design, a designer reasons abstractly about the components, concentrating on high-level characteristics in order to evaluate alternative configurations. Consider, for example, the design of a print head positioning mechanism used in a dot-matrix printer as shown in Figure 1. To reason about the feasibility of the proposed configuration the designer does not refer immediately to a catalogue to select a particular motor, but instead relies on his experience to estimate the size, weight, torque, and power of the motor, ignoring other more detailed geometric and behavioral characteristics.

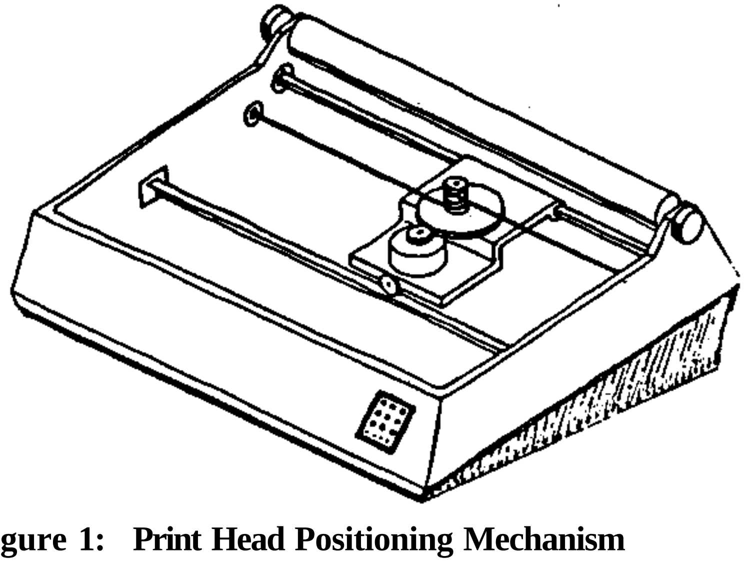

Figure 1: Print Head Positioning Mechanism

As **the** design is refined **the** designer will consider characteristics **at** many levels of **abstraction. He reasons about high-level parameters such as weight and positioning speed because these are the terms which express the design specifications. He also employs intermediate-level parameters such as** stress, **aspect ratio and current density because these help him reason efficiently about the** device, **and** he considers **low-level parameters,** for **example, length, diameter and material** since these **are** the design variables **that he directly** controls.

In order to reason **about** these various parameters designers use their experiential knowledge of the relationships between the form of a device and **the** behavior it exhibits. Knowledge of the relationship between weight and motor torque, for example, enables a designer to quickly evaluate the performance of the print head positioner The configuring and evaluation of a complex design is aided by an understanding of these *form-function relations* which express inherent characteristics of the components that comprise the design [1,2].

Although vast experience is beneficial it may not be sufficient to capture some relationships such as those between parameters of different levels of abstraction, for instance between the power consumption of an electric motor and the winding wire diameter. The two are related, but to a designer the form of the relation would not likely be obvious. Determining this sort of relation is useful to the designer because the design requirements are frequently given in terms of the high-level behavior desired, but the designer's task is to specify a detailed description of this component.

Form-function relations also exist between parameters that are at or near the same level of abstraction but that are normally associated with physically separate parts of the design, such as the rotor and stator of a motor, or parameters that exist in different energy domains, such as electrical and mechanical. Understanding these relationships aids the system designer by explicitly showing how high-level characteristics of the components are related to one another, allowing him to reason about tradeoffs between competing requirements or objectives, for instance the tradeoff between power and size of an electric motor. It is the consideration of these relations that dominate preliminary design decisions, for they are used to evaluate tentative configurations.

High-level form-function characteristics are a reflection of the underlying physics of the device and are abstractions of the complex interactions of low-level parameters. A change in the torque requirement for the motor **may** be met by adjusting the wire size, the number of poles, or a host of other parameters. However, other characteristics such as weight, will also change. Determining the resulting relationship between torque and weight is difficult to do directly. Instead designers usually gain this sort of knowledge empirically over years of design experience, acquiring **a** "feel" for the characteristic relations of a device. Empirical methods however, are subject to error due to hasty generalizations and narrowness of experience. In addition, these relations are dependent on the state-of-the-art, therefore major changes in technology can render the knowledge of particular form-function relations obsolete.

Motivated by the usefulness of form-function relations and the difficulty in obtaining them, we have developed a representation of electromechanical devices and a method for automatically identifying relevant form-function relations from it. The details of the representation and the method used are discussed in other papers [3, 2].

Device Representation

Our approach is based on **the** declarative **representation of a device** as **a set** of **parameters related by** constraints. **The** constraints **arise** from physical laws, spatial relationships **and** material limitations. Collectively these constraints define **the** space of acceptable designs[1].

In this model each parameter describes some characteristic of the form (such as a physical dimension or material density) or behavior (such as velocity, stress, or torque). The constraints relate the parameters typically through equalities or inequalities[2]. Equality constraints are relationships between parameters that always hold and may be the result of physical law (e.g., */= ma),* may be imposed as a requirement of the design (e.g., *voltage = llvolts)* or may define a geometric relationship (e.g., $A = KD^2/4$). Inequalities are often used to express physical limitations (e.g., *temperature < melting temperature),* imposed requirements (e.g., *torque > 2Aft-lbs)* or spatial relations (e.g., *OD>ID).* The compositional nature of the constraints allows the model to be easily expanded to an arbitrary level of description by the addition of parameters and constraints.

A collection of these constraints forms a network [4] or a *bipartite graph [5]* with each node representing either a constraint or a parameter. In this graphical representation of the constraint model, each parameter node is linked to all of the constraints that it participates in, and each constraint is linked to all of its participating parameters. A constraint network for a brushless, unhoused d.c. electric motor under stall conditions is shown in Figure 2. Note that most of the inequality constraints and numerical limitations on the parameters are not shown in the figure for the sake of readability.

The constraint network itself represents a prototypical device or class of devices and *satisfying* the network of constraints by assigning a value to each parameter such that none of the constraints are violated, results in an instance of the class[3]. Thus we can view the satisfaction of a constraint network as analogous to parametric design. But our goal in developing this representation *is* not to support parametric design, it is rather to aid the designer by providing a way to automatically identify relevant form-function relations.

Design Context

A network of constraints represents the nature of a class of devices, but not a particular design problem. The specific requirements and objectives of a problem constitute a *design context,* which can be cast over the network to represent the given task. The context further constrains the space of the solutions to those that are acceptable for the particular situation and specifies the criteria necessary to identify the "best" design among these. Thus both a design and its corresponding set of form-function relations are directly related to the context under which it was created. For example one designer may be designing a motor for a given torque while trying to minimize weight, while another may be trying to maximize the efficiency while maintaining a fixed diameter. While both designers may be designing motors of the same class.

Although both the parameterization and the choice of relevant constraints both influence the design space.

[2] We are currently using only these but our approach does not preclude other constraints such as boolean, discrete, differential equations, etc.

Satisfaction of a constraint network is in general very difficult. As stated by Gosling in [6] "...boolean satisfiability can be cast in this framework and is NP-complete, so general satisfaction is at least as hard." But there have been many techniques developed and applied to this problem including constraint propagation. monotonicity analysis, relaxation and optimization.

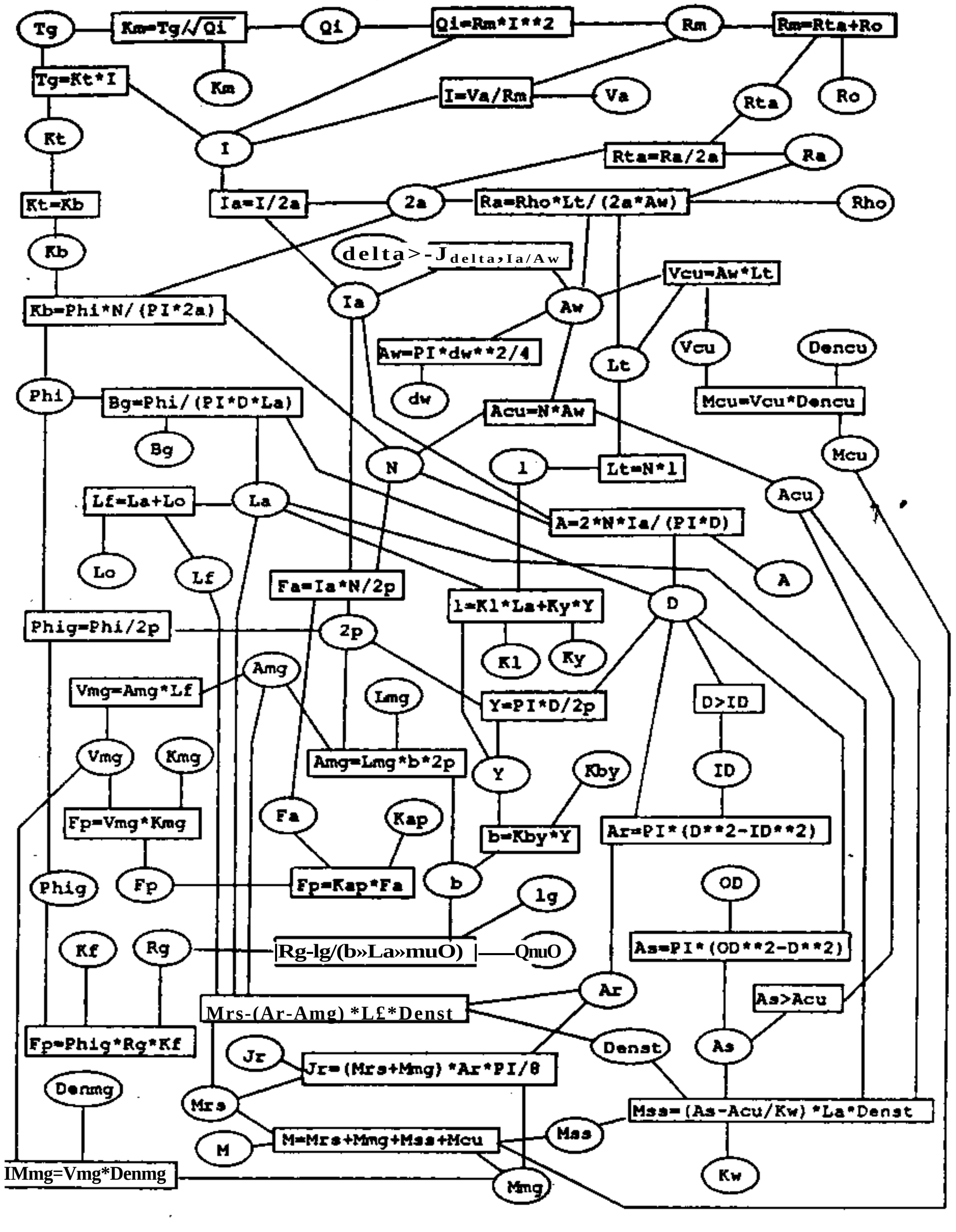

Figure 2: Example Constraint Network: D.C. Electric Motor

the first will produce a different design than the second, and the relations between characteristics such as torque and weight or efficiency and diameter will be different for the two contexts.

Approach

To aid the designer, we determine form-function relations from a constraint-based model of the underlying physics of a device. Our approach is based on solving a sequence of optimization problems, corresponding to a continuum of design contexts. This technique results in a series of optimas which can be plotted to show the relationships between various parameters[4]. The following example illustrates this approach.

Example: Brushless D.C. Motor

Consider the use of a frameless brushless d.c. motor in a printer as seen in Figure 1. The designer wishes to reduce the length of a standard motor to fit it in a particularly tight space. It is necessary to reduce the length as much as possible while maintaining the same torque and inside diameter and minimizing weight. The designer may be interested in increasing the outer diameter to effect this change, however, he will want to know how changing the diameter may affect some of the other aspects of the design.

Figure 3 shows the form-function relations obtained for this context from the constraint network shown in Figure 2. Each of the points marked on the curves represents a computer generated optimum of a prototype motor. Each curve then relates diameter of the motor to some other parameter. The relations are shown as relative changes in the outer diameter (measured along the abscissa) versus relative changes in the other parameters (measured along the ordinate). Thus the origin represents the standard motor (optimized for minimum mass), and the curves represent relative changes from that point.

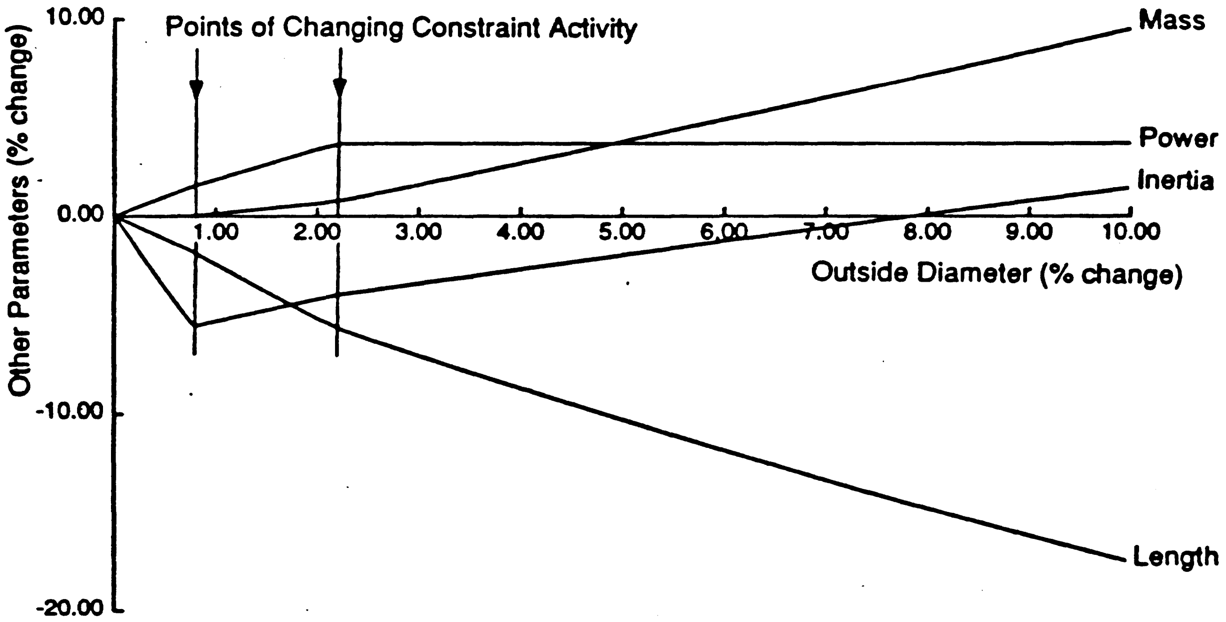

Figure 3: Form-Function Relations for a D.C. Motor

---

[4] This method is similar to what are called *interaction curves* in the optimization literature [7].

Constraint Activity

The relations shown are the result of the interaction of the defining constraints. In particular, the activity of the inequality constraints[3] shape the relations. We note that relationships among design parameters change dramatically with changes in the activity of the constraints resulting in the various cusps in the plots. Of particular interest in this example is the relationship between outer diameter and the motor inertia. Initially increasing the diameter actually decreases the inertia, but the rotor will reach a limit on strength and further increases in the diameter will call for a more robust and therefore higher inertia rotor. Understanding the nature of a design in this way can be very insightful. Thus determining the active constraints that reflect what is limiting the design can aid in reasoning about the design.

In general predicting the constraint activity is very difficult due to the complex, non-linear nature of the underlying equations and inequalities. The constraint activity depends on both the constraint network and the specific context of the design. For this reason it is not possible to determine *a priori* the active constraints, instead they must be determined for each context. Because they are difficult to predict and because they influence the form-function relations so strongly, the knowledge of active constraints is valuable to the designer.

An Environment for the Conceptual Design of Mechanical Systems

Component form-function relations can help the designer identify tradeoffs for classes of components such as motors, pulleys, masses and so forth, but ultimately the behavior of interest is that of the overall configuration, which is made up from an aggregation of components. Moreover, for mechanical devices the particular geometric interactions between components have to be taken into account to determine the overall system behavior. Reasoning about these behavioral and geometric relations can become a difficult task for the designer as he considers more numerous aggregations of components and as he refines the selected configuration. In this section we discuss the representation requirements for conceptual design environments that can support an interactive aggregation of standard components and the subsequent refinement of a selected configuration.

Representation of Component Aggregations

To allow the aggregation of components into a system, the design environment has to allow the *modular* aggregation of *behavioral* component models. Since mechanical devices include complex geometric relations among components, the internal component representation should include *geometric* component models as well.

Depending on the particular characteristics of the configuration, the designer may want to focus on specific aspects of component behavior. Consider the printer head drive configuration shown in Figure 1. For this configuration, the designer may want to investigate the effect of the linear inertia of the motor on the aggregate dynamic behavior of the drive system since the motor rides on the platform. To model adequately the behavior of the same component connected in different ways, the environment representation has to support several *functional perspectives* for a given component.

If the designer finds that the motor's rotational inertia has a much greater relative effect on the system dynamics than the its translational inertia, he may want to modify the resolution of the motor's behavioral model to investigate its electrical system dynamics. Alternatively, he

---

[5] An active inequality is one that *is* binding as a strict equality.

may want to assume an extremely simple motor model (e.g. an ideal source of torque) to **establish a limit on maximum expected performance. To provide the designer the flexibility of varying the modeling detail of a component, the internal component model representation has to allow** *variable resolution* **of component models. The need for variable resolution also applies to the geometric model of components to support the gradual refinement of the** geometric **detail of** the design.

Approach

The representation requirements of modular aggregation, functional perspectives and variable resolution have been achieved in an experimental implementation of an environment for conceptual electro-mechanical design. MEDA[6] **is** an interactive graphical environment in which component models can be aggregated modularly **into** a preliminary system description. Mechanical components can be connected by specifying kinematic relations between them. The designer can group connected components into subassemblies **to** consider the behavior of a meaningful part of the overall device. Once a subassembly **is** identified, the internal modular models of component behavior are automatically collected into a system description, with consideration of the functional perspective implied by the components' connectivity. Through the use of appropriate behavioral primitives, component models with variable resolution can be defined and stored in a component library.

The implementation is targeted to the design of electro-mechanical devices which may require complex dynamic analysis and therefore MEDA automates the process of dynamic model development. For this domain, the component connections are kinematic relations. The kinematic relations specified by the designer introduce constraints between behavioral component models when a subassembly system model is collected. After the behavioral component models are collected into a system model, **the** resulting system representation can be converted into a set of differential equations for subsequent numerical or symbolic processing.

MEDA is implemented in C as an extension **to** a commercial modifiable CAD platform [8] in a Sun™ 3/160 workstation. The details of the implementation and a more elaborate discussion of the requirements for conceptual design representations can be found in [9] and [10] respectively.

Automatic Identification of Critical Design Relationships

Once a design configuration has been determined it can be parameterized so as to obtain a set of algebraic constraint equations. These equations describe the functional relationship between the important behaviors of the design and the parameters describing the configuration. The ease with which quantitative and qualitative evaluations can be made depends, in large part, on the complexity of the design equations and constraints involved. Even very simplified design equations may be puzzling because changing the value of one of the design variables may influence many of the functional requirements. As a result, detailed analytical methods are often applied to any competing design configuration alternatives. The results of the analysis are used to judge the merits of the design configurations. Experienced designers, on the other hand, often shortcut the detailed analytical work by recognizing important relationships which govern the performance of the design configuration. This is accomplished by identifying important relationships among requirements and design variables, such as a critical ratio, a nondimensional

---

[6] **MEDA stands for Mechanical Engineering Design Assistant.**

parameter, or a simple difference; e.g. the column aspect ratio in structures, the Reynold's **number in fluid mechanics, or the velocity difference across a fluid coupling. This achieves convenience and expediency in quantitative evaluations and enhances the qualitative physical reasoning associated with the design activity to better enable the designer to focus his creativity on the essential aspects of the proposed configuration.**

The discovery **of** such **critical relationships among** parameters has **been made** on an **ad hoc** basis by experienced designers **and engineers.** Although certain nondimensional parameters are well known and methods exist **for** identifying such parameters, **there are not,** in general, strategies which assist the designer **in** identifying physically significant relationships which dominate the behavior of a particular design configuration. We seek to do this by establishing methods to identify physically significant new variables and to use them in performing a *transformation* of variable on the constraining design equations and inequalities. The terms *alternative formulation* or *reformulation* refer to a description of the same design configuration but with alternative design variables.

Two Bar Truss - Case Study

As an example of a variable transformation consider the design of a two bar truss [11]. A truss of the type shown in Figure 4 can be used to support both a vertical and horizontal load. The truss deflection must be limited and the truss must withstand the loads without yielding or buckling. The parameters describing the truss are height //, halfwidth *B*, tube diameter d_f thickness f, and the modulus of elasticity, *E*. The relationships among actual stress, critical buckling stress, deflection and the design parameters are shown in Figure 5. Also shown in the figure is a transformation of variable applied to the truss equations. Changing design parameters does not change the truss design itself, only the form of the problem.

The Nature of Design Complexity and Coupling

Note, that each of the truss design relations depends on all four of the geometric design parameters making it difficult to reason about the design. When this occurs, we say that the desired functions are *coupled* or that the design is coupled. Simon [12], Preiss [13], Rinderle and Suh [14] are among those who commented on the nature of coupling in designs. The complexity due to coupling depends, at least in part, on the designers selection of the product functional requirements and the design parameters. One structure of particular interest, which we refer to as serially decomposable, results from transformations that cause the design equations to become solvable without iteration. The transformation performed in Figure 5 makes the truss equations serially decomposed.

Generating and Choosing Useful Transformations

It is known that almost all of the nondimensional variables commonly used are one of four types: ratios of lengths, forces, energies, or properties [15]. This follows from the fact that all of the fundamental equations of continuum analysis can be nondimensionalized with these four types of nondimensional variables. In many cases the forces and energies are constructed from more primitive variables. For instance, Reynold's number is the ratio of fluid inertial force to fluid viscous force and in turn these forces are constructed from variables for fluid velocity, viscosity, length, and mass density. In fact, all the common nondimensional variables in fluid mechanics can be constructed from fluid force ratios. Rules for the construction of forces can be compiled for many mechanical engineering disciplines. In order to construct and use such rules, information other than fundamental units is required for each component variable.

The identification of dimensional variables is also enhanced using physical meaning rules. In the field of geometry, for instance, areas can be constructed from a radius squared but not

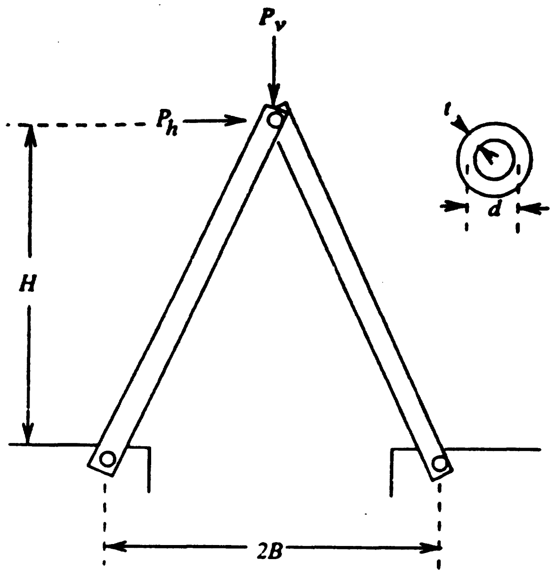

Figure 4: Two Bar Truss

The original design requirement equations are:

$$\sigma_{compression} = \frac{\sqrt{B^2+H^2}}{2\pi td}\left(\frac{P_v}{H}+\frac{P_h}{B}\right) = f_1(B,H,d,t)$$

$$\Delta\sigma_{buckling} = \frac{\pi^2E(d^2+t^2)}{8(B^2+H^2)} - \frac{\sqrt{B^2+H^2}}{2\pi td}\left(\frac{P_v}{H}+\frac{P_h}{B}\right) = f_2(B,H,d,t)$$

$$\delta = \frac{P_v(B^2+H^2)^{1.5}}{2\pi tdH^2E} = f_3(B,H,d,t)$$

The following transformation:

$$d = d,\ H = H,\ R = B/H,\ A = \pi dt$$

produces an alternative form of the equations:

$$\sigma_{compression} = \frac{\sqrt{R^2+1}}{2A}\left(P_v+\frac{P_h}{R}\right) = g_1(R,A)$$

$$\delta = \frac{P_vH(R^2+1)^{1.5}}{2AE} = g_2(R,A,H)$$

$$\Delta\sigma_{buckling} = \frac{E(\pi^2d^4+A^2)}{8H^2d^2(R^2+1)} - \frac{\sqrt{R^2+1}}{2A}\left(P_v+\frac{P_h}{R}\right) = g_3(R,A,H,d)$$

where for both sets of equations the requirement constraints are:

$$\sigma_{compression} \leq \sigma_{yield},\ \Delta\sigma_{buckling} \geq 0,\ \delta \leq \delta_{max}$$

Figure 5: Two Bar Truss Equations

from a thickness squared In the domain of rigid body dynamics forces or torques can be **indicated as inertial or viscous and can be hierarchical in nature. For example, an inertial force may consist of the product of a mass and an acceleration, but acceleration in turn can be constructed according to other rules. In this way, for example, it is possible to generate both torque and energy variables (both of which may have units of foot-pounds) but to maintain the distinction between them and to use them appropriately. Construction techniques** of this sort are chosen **for general design domains and applied recursively. Spatial proximity** methods are used **to reduce the list of candidates prior to more complete evaluation based on** design equation structure. In the truss example the transformation is to an area, A, and a ratio of lengths, *?, while two parameters, *d* and //, remain the same. While **the** utility of this transformation is high, as will be seen in the next few subsections, the transformation itself is quite simple. Finding this transformation required a generate and test technique combined with goal oriented methods employing a measure of coupling. Details are described **elsewhere [16].**

Summary of Reformulation Advantages

There are several advantages to using less coupled, especially serially decomposed, design equations. The following ones will be discussed for **the truss problem:**

1. Enhanced ability to find numerical solutions.
    - Ease in determining satisfactory solutions.
    - Ease in making design changes to accommodate requirement changes with minimal iteration.

2. Identification of active constraints.

3. Symbolic computational benefits such as identifying form-function relationships.

4. Cognitive Benefits.

Numerical Solutions

Alternative variables with clear physical meaning allow, and perhaps even enhance, numerical value estimates. This is seen in the truss equations where a numerical solution of the original formulation could become an extended trial and error session of computations. This is because each expression relies on all of the design parameters making the problem one of three simultaneous nonlinear expressions in four unknowns (with additional inequality constraints). The reformulation on the other hand has been arranged in a serially decomposed form and a numerical solution can be easily found in a single iteration. This is facilitated by the fact that as one goes through the list of relations there is one additional parameter for each additional expression. The physical significance of each variable facilitates good estimates, for example the truss aspect ratio /? = 1 is entirely reasonable. If for some reason in the future the requirements on any one functional requirement or design parameter change so that the design is no longer satisfactory then the serially decomposed reformulation has definite advantages. The designer will only have to loop back to the last equation which contains the design parameter or functional requirement which makes the design unsatisfactory. For example, if after a satisfactory solution is found it becomes necessary to change pipe diameter then only the equation for $Ao_{bnckJing}$ need be consulted in the reformulation when making that change in pipe diameter.

Identifying Active Constraints

For a particular design configuration **defined by parametric** conrtraints there may not be **a** solution space for **the** given sptttifications. **This would eliminate that** configuration from consideration as a viable alternative, **or promote a** respecification, or modification of **the** configuration [17]. At this stage of preliminary design it may be difficult to determine if there is a simple design limitation and what constraint or constraints constitute it. Alternative reformulations facilitate the identification of limiting constraints considerably, especially if the problem is made nearly monotonic. Furthermore, reformulations often iroke it possible to employ more powerful symbolic techniques, such as those discussed by Agogino[18] and Choy [19] which are most useful when the design equations are monotonic and serially decomposed.

Considering again the truss problem, now with the objective of minimizing weight. Weight is a function of the original and new parameters respectively and is proportional to volume, $V = indrl\overline{E^1 + H^2}$ or $V = 24//V/\overline{?^2 + 1}$. The reformulation makes it clear that the value of d plays no part in minimizing volume. It influences the satisfaction of the buckling constraint only. The reformulation also makes it clear that the buckling constraint can be satisfied after all the others and that there is an active constraint on compressive stress $(o_{eompr9Isiom} = a_{yiM})$ or an active constraint on maximum deflection (5 = 5^ depending on the numerical values involved$_v$

Symbolic Computational Benefits - Form-Function Relationships

Very often it is desirable to know how some aspect of a design's form, other than the design parameters themselves are influenced by changes in the functional specifications of a design. The problem with developing such relationships analytically is that most design problems are underconstrained, but in some cases a degeneracy exits that will overcome this problem. Identifying such degeneracies so as to develop analytical form-function relationships can be greatly enhanced by a transformation of variable. Consider the truss problem where the expression for volume in terms of new design parameters does not depend on d. This eliminates the expression for $A<J_{bt4Ckling}$ from consideration but still leaves an underconstrained problem. If we set P_h to zero then the two remaining expressions are simplified. From these simplified expressions it is possible to determine that volume is proportional to 5 if the active constraint is on deflection and inversely proportional to $o_{compression}$ squared if the active constraint is on compressive stress.

Cognitive **Benefits**

Cognitive benefits are those which help the designer reason about the design problem. His previous experience, current frame of mind, degree of concentration, etc., will interact with the problem representation and formulation to influence his understanding. Any insights he might have will depend on that understanding. With respect to the trass problem, the ability to easily find numerical solutions enhances the designer's thinking by supplying quick design parameter value estimates. In addition for the trass, insights on active constraints, and relationships between form and function were facilitated by the reformulation. By the time the designer has gone through the exercises above he will look at the trass problem from a different perspective[7].

---

[7] In the words of Simon [12] "All mathematical derivation can be viewed simply as change in representation. making evident what was previously true but obscure. This view can be extended to all of problem solving - solving a problem simply means representing it so as to make the solution transparent".

Results

A test bed computer program, called EUDOXUS has been implemented to test and demonstrate the techniques discussed for identifying critical design relationships. The program has been used in the design of a Geneva mechanism, a simple suspension system, an impact energy absorber, a helical coil spring, a heat exchanger, and the simple two bar truss.

A Structured Synthesis Strategy for Mechanical Devices

During the design process the designer transforms an abstract functional description of a device into a physical description which satisfies the requirements. In this sense, design *is* a transformation [20] from the functional domain to the physical domain.[8] However, the basis for selecting favorable transformations and methods for accomplishing transformations are not well understood. Our work is an effort to discover the desirable characteristics of these transformations and develop a structured approach for transforming a device specification into a physical description, thereby creating design alternatives. We contend that form-function relations provide a strong basis for selecting favorable design transformations and that by combining these with the simple guideline of integrating functionality, a robust and useful structured approach to design can be defined. This approach can be used to design a device configured from classes of known components.

A Strategy for Mechanical Design

The direct functional decomposition of a set of device specifications and subsequent one-to-one matching of individual functional requirements to physical components results in weak designs for two reasons. The first of these is that by matching individual functions to some collection of components we forsake the opportunity to integrate functions into more compact or economical collections of components. Secondly, components provide not only the desired function but also many additional, unintended behaviors. The following simple examples illustrate the problems with direct decomposition and demonstrate one method that helps resolve them.

> Consider the design of a speed reducing device. Specifications for the device are given as:
>
> - The ratio of input to output speed must be 8:1.
> - The input and output shafts must lie at right angles to each other.
>
> This set of specifications is already functionally decomposed into two independent functional requirements, the 8:1 reduction and the right angle requirement. The most direct solution is to match each of the individual requirements with a separate piece of physical hardware. A spur gear set may be selected because its behavior matches the functional requirement of an 8:1 reduction. Then a bevel gear set may be selected because its behavior matches the functional requirement of a right angle between the input and output shafts. The resulting physical description then consists of a spur gear pair connected to a bevel gear pair. So, by matching between the decomposed functional description and the behavioral descriptions of known components a physical configuration can be generated. However, this simple case demonstrates that one-to-one matching usually results in a poor design because the resulting device is more complex than is necessary. If instead the matching is done so that functional integration is emphasized, then more compact and economical designs result. Here, by using the form-function relations for bevel gears, it can be determined that a bevel gear provides both of the functions required, a reduction *and* a right angle between the input and output shafts. So, matching *both* functions to a bevel gear results in higher degree of function integration and a more economic execution of the design.

---

[8] It is important to note that the word transformation is used here in a different sense than that expressed in the previous section on Automatic Identification of Critical Design Relationships. Here, the transformation is not applied to a set of equations which describe an already configured design, but instead to a set of specifications which are transformed until they correspond to a complete physical description of a device.

This example illustrates that combining functions in a single device will often result in a superior design. The difficulty lies in properly matching the functions in the specification to actual components. Form-function relations of the components provide a means to this end because they express the relation between behaviors and physical form. However, matching the functional specifications to the physical form of the components is not always as straightforward as in the previous example. The following example demonstrates some of these complexities and some methods for accomplishing an economic execution of the design.

> Consider the design of another speed reducing device. Specifications for the device are given as:
>
> - The ratio of input to output speed must be *40:1.*
> - The input and output shafts must be at right angles.
>
> If the catalogue of available items consists only of straight bevel and spur gear sets, we find that there is no single component which is capable of providing the 40:1 speed reduction. So an immediate matching between these specifications and a physical description is not possible. Instead the specifications must be *transformed* into a form that allows for matching between the specifications and device behavioral characteristics. In this case we apply a transformation which decomposes the specified functionality into a collection of equivalent functions; for example, a 5:1 speed reduction followed by an 8:1 speed reduction and a right angle between the input and output shafts. Note that this transformation preserves the overall functionality expressed in the original specifications. This is an important characteristic of all transformations; that they be function preserving, otherwise the completed device will not meet the specifications. The decision to decompose the specifications in this manner is guided by the form-function relations for the components. They represent the fact that no single spur or bevel gear pair can have a reduction ratio greater than 8:1. Therefore, this particular form-function relation is used to guide the decomposition of the 40:1 reduction ratio into two elements neither of which exceeds an 8:1 reduction ratio. After this decomposition, it is possible to match each of the individual functions, the 8:1 reduction, 5:1 reduction and right angle, into a physical component; however, direct matching would result in a design which is large, costly and complex as was shown before. Alternatively, we can seek *groups* of functional elements which closely correspond to available physical components. In this case, by considering the form-function relations for the known components, bevel and spur gears, we find that we can group or associate the 5:1 speed reduction with the right angle function. So we can view the specifications as consisting of two groups; a 5:1 reduction coupled with a right angle, and an 8:1 reduction. We now find that we can match each of these *groups* with a physical component; specifically, a bevel gear set can satisfy both the right angle requirement and one of the reductions, and a spur gear set can satisfy the second reduction requirement.[9] In this case we have not only achieved multiple functions with a single component but we have identified a particularly favorable selection of components by grouping individual functions into collections that closely corresponded to real physical devices.

This example illustrates the integration of multiple functions into single components and the value of function preserving transformations intelligently applied to the device specifications. It also demonstrates the role of form-function relations in guiding these processes. However, **the approach used** above will **not** always be clear or unambiguous. The matching between specified functionality and component behavioral characteristics is not always as perfect. Physical components have many behaviors and an exact match between specifications and component behavior will not always be possible. If this is the case, unintended behaviors will be introduced into the device. Consider if the lexicon of known components used in the example above included worm gears. A worm gear introduces an offset and a right angle between the input and output shafts. It also can provide a 40:1 reduction ratio. Therefore, a worm gear could be used to integrate all of the desired functionality, 40:1 reduction and right angle, into one component resulting in a lighter, simpler device. But, it also introduces a behavior not required in the specifications, an offset between the input and output shafts. This

---

[9] The relative configuration of the two components can be selected to maintain coplanarity of the shafts or to achieve an offset. The way in which components are configured relative to each other is critical to both function and geometry.

[illegible] only if this behavior is acceptable. So, the specifications must be transformed and matched to the components incrementally and the the desired functionality of the device must be maintained after each transformation, grouping or matching of the specifications.

Results

We are developing a design methodology that will both enable the integration of functions into single components and will take advantage of, or compensate for, unintended functionality of components by appropriately utilizing the form-function relations of the components used in the design process. The methodology is suitable for designing mechanical devices composed of standard components. In this case, by design we mean generating the configuration, i.e. the type and arrangement of standard components utilized in the device. The methodology requires a representation of the form-function relations of the standard components, and a representation of the specifications for the device. Also required is the ability to transform the representation of the specifications in a function-preserving manner. Using these representations of the design requirements, the available components, their form-function relations and the function preserving transforms, the most desirable transformations can be identified and applied until a complete physical specification of the device is determined. This methodology has been developed to be applicable to a broad range of design domains. A trial study of its effectiveness has been completed for the example domain of single speed, single input-output geared transmissions. A discussion of its application to this domain and a description of a computer program which utilizes the methodology can be found in [21].

Summary

Form-function characteristics of mechanical components and devices exist and can be identified from physical principles. Furthermore, design decisions are dominated by considerations having to do with form and function. The four areas in our program of research demonstrate that progress is being made to address the absence of a close correspondence between functional and physical descriptions in mechanical design. Tools and methods were described that will reduce our dependency on the intuition experienced designers use to extract and utilize relationships between form and function.

References

1. Rinderle, J.R., "Function and Form Relationships: A Basis for Preliminary Design", *Proceedings from the NSF Workshop on the Design Process*, Waldron, M.B. ,ed., Oakland, CA, February 1987, pp. 295-312.
2. Colburn, E.R. and Rinderle, J.R., "Identifying Design Relations", Internal Lab Report CMU-MEDL-88-25, Department of Mechanical Engineering, Carnegie-Mellon University, 1988.
3. Colburn, E.R. and Rinderle, J.R., "Supporting Design Reasoning with a Constraint-Based Representation", Internal Lab Report CMU-MEDL-88-22, Department of Mechanical Engineering, Carnegie-Mellon University, 1988.
4. Gross, M.D., *Design as Exploring Constraints*, PhD dissertation, Department of Architecture, Massachusetts Institute of Technology, 1986.
5. Serrano, D., *Constraint Management in Conceptual Design*, PhD dissertation, Department of Mechanical Engineering , Massachusetts Institute of Technology, 1987.

6. Gosling, J., *Algebraic Constraints, PhD* dissertation, Department of Computer Science, Carnegie Mellon University, 1983.

7. Siddall, J.N., *Optimal Engineering Design,* **Marcel Dekker,** New **York, NY, 1982.**

8. ***MIPIE: Modifiable/Portable/Extendable, Reference Manual,*** **Unicad, Inc., Boush & Plume** Streets, **P.O.** Box 1839, Norfolk, Virginia 23501,1987.

9. Paz-Soldan, J.P-, *Conceptual Design Environments for Opportunistic Refinement", Internal Lab Report CMU-MEDL-88-24, Department of Mechanical Engineering, Carnegie-Mellon University, 1988.

10. Paz-Soldan, J.P., "Abstraction and Structure in Conceptual Design Environments", Internal Lab Report CMU-MEDL-88-21, Department of Mechanical Engineering, Carnegie-Mellon University, 1988.

11. Parkinson, A.R., Balling R.J., and Free, J.C., "Exploring Design Space", *Computers in Mechanical Engineering,* **March 1985, pp. 28-36.**

12. Simon, H.A., *The Sciences of the Artificial,* second edition, M.I.T. Press, Cambridge, MA, 1981.

13. Preiss, K., "Data Frame Model for the Engineering Design Process", *Design Studies,* Vol. 1, April 1980, pp. 231-243.

14. Rinderle, J.R. and Suh, N.P., "Measures of Functional Coupling in Design", *Journal of Engineering for Industry,* Vol. 104, November 1982, pp. 383-388.

15. Kline, S.J., *Similitude and Approximation Theory,* Springer-Verlag New York Inc., 1986.

16. Rinderle, J.R., and Watton J.D., "Automatic Identification of Critical Design **Relationships",** ***Proceedings International Conference on Engineering Design,*** **Boston,** MA, August 1987, pp. 512-525.

17. Dixon, J.R., "Iterative Redesign and Respecification: Research on Computational **Models of Design Processes",** ***Design Theory '88: The 1988 NSF Grantee Workshop on Design Theory and Methodology,*** **June 1988.**

18. Agogino, A.M., "AI in Computer-Aided Design: Qualitative Reasoning and Symbolic **Computation",** ***Proceedings from the NSF Workshop on the Design Process,*** **Waldron,** M.B. ed., Oakland, CA, February 1987.

19. Choy, J.K. and Agogino, A.M., "SYMON: Automated Symbolic Monotinicity Analysis **For Qualitative Design Optimization",** ***Proceedings ASME Computers in Engineering*** *Conference,* Chicago, July 1986.

20. Mostow, J., "Towards Better Models of the Design Process", *The AI Magazine,* Spring 1985.

21. Hoover, S. P. and Rinderle, J. R., "A Structured Synthesis Strategy for Mechanical Devices", Internal Lab Report CMU-MEDL-88-23, Department of Mechanical Engineering, Carnegie-Mellon University, 1988.