1991

Automated modeling to support conceptual design

L Balasubramaniam
Carnegie Mellon University

Juan Pedro. Paz-Solda╠ün

J Rinderle

Carnegie Mellon University.Engineering Design Research Center.

Automated Modeling to Support Conceptual Design

by

L. Balasubramaniam, Juan Pedro Paz-Soldan, James R. Rinderle

EDRC 24-55-91

Table of Contents

Submitted for publication in "Computer-Aided Design"

Automated Modeling to Support Conceptual Design

L. Balasubramaniam
Graduate Research Assistant

Juan Pedro Paz-Soldan
Graduate Research Assistant

James R. Rinderle
Associate Professor

Department of Mechanical Engineering
Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

The need to consider a number of different alternatives at an early stage in design has been well established. It is hypothesized that providing a tool for automatically modeling and analyzing devices and relating individual component characteristics to device behavior would aid the conceptual designer by facilitating the consideration of more varied alternatives. In this paper we identify the characteristics of conceptual mechanical design that must be accommodated in any tool to aid designers at this stage. The role of natural interface, modularity and numerous types of flexibility requirements have been established and addressed in a prototypical design environment. The nature of automated modeling and issues pertaining to modeling relevance and model simplification and analysis techniques are also briefly discussed in the context of a design-analysis tool for automatic formulation and analysis of dynamic system models.

Introduction

The study of conceptual design as an engineering problem-solving activity has been motivated by the need to develop tools to aid the conceptual designer in developing and comparing alternative solutions to the design problem. Such a tool should complement human capabilities by automating tasks which humans find laborious or difficult. Although the cognitive processes involved in conceptual design are still poorly understood, one view of preliminary design activity indicates closely coupled cycles of synthesis and analysis: The designer alternates between addressing the questions, "What would satisfy the functional requirements?" and "Does this configuration satisfy the specifications?" [Bell 81, Johnson 71, Mann 77, Asimow 62]. Although some synthesis strategies exist which help designers to select components and configurations [Pahl 84, Ulrich 89, Hoover 89], it is still necessary to analyze the candidate designs. Furthermore, the knowledge of how different components contribute to the behavior of the device provides a powerful guide to the synthesis procedure itself. Because conventional analysis methods cannot operate on the sketchy and vague information that is available at this stage, designers more often than not rely on rule of thumb approximations to determine overall trends. Such analysis is invariably a precursor to further refinement of the design and the cycle continues. Designs thus evolve from a preliminary concept by a process of continuous evaluation and refinement. Every step of this process necessarily consists of assessing the extent to which functional specifications have been satisfied. There is clearly a need to evaluate device behavior based on behavior of individual components. A tool that satisfies this requirement while operating with sketchy information typical of the conceptual stage is expected to be an effective aid to a designer.

A Sample Design Problem

In a typical design scenario, the designer must first synthesize a configuration from the initial functional specification. There is not a one-to-one mapping between functions and configurations: A wide variety of configurations can perform roughly the same function. For each proposed configuration the designer must specify the basic layout of the device, identify appropriate components, visualize how the components will fit together and mentally simulate how the device will perform.

Consider for example, two of the many possible configurations that might be considered when designing a print head positioning system for a personal computer printer. The function of the positioning system is to transport the print head platform across the expanse of the printer width and bring it to a stop so that a character can be printed. The two solutions shown in Figure 1 use different ways of converting the torque delivered by the motor to a force that can drive the sliding platform across the printing area. In the first configuration the motor is attached to the printer casing and drives a toothed belt which is connected to the platform. In the second configuration, the motor is mounted on the platform itself and a traction wheel drives the platform.

To arrive at the configurations described above the designer needs to reason about abstract functionality of components. From an informal protocol study conducted with three mechanical engineering graduate students, [Paz-Soldan 87, Paz-Soldan 89] reports that designers, when given this print-head positioner problem, started by decomposing the overall functionality into sub-functions and then identified components which could achieve those particular functions. The overall functionality of the printer-head positioner was split into *constraining* the platform to allow only linear motion and *driving* the configuration. From a knowledge of

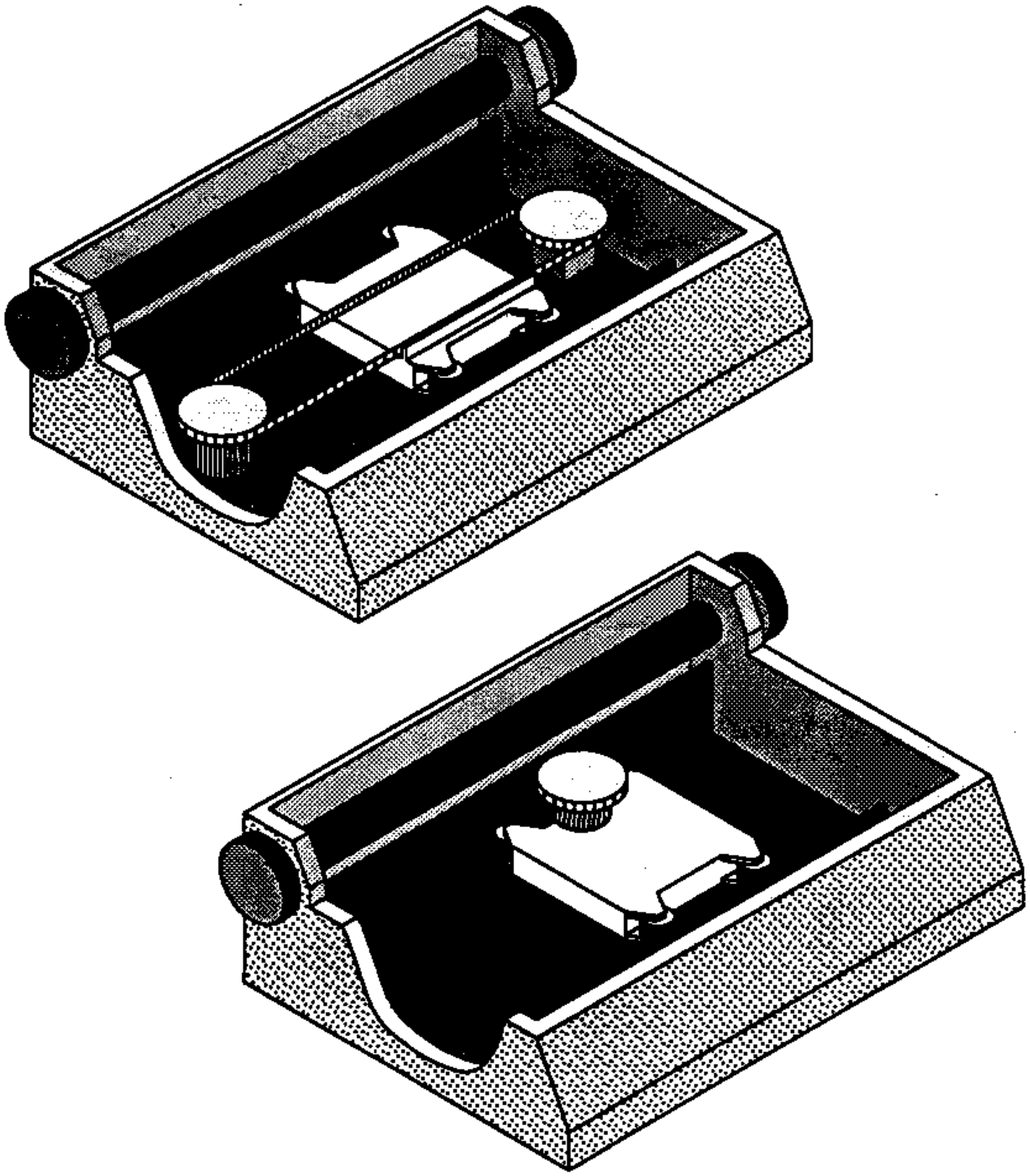

Figure 1: Two Possible Printer Configurations

existing components the subjects rapidly converged on electric motors as the most suitable component for *driving,* however, they spent a significant amount of time deciding what type of motor to use. This suggests that rule-of-thumb analysis aimed at understanding how specific component behavior would influence the behavior of the group as a whole is a demanding task; more so because the behavior of an aggregate of components is more than the aggregation of their individual behaviors. The presence of *emergent* behavior such as oscillations or geometric interference complicates analysis to the extent that even intelligent guesses about component/configuration behavior are not always valid and can be made with confidence only when the components and configurations involved are familiar from some previous design activity.[1]

In our research we therefore focus on techniques that allow building and analyzing symbolic models of behavior that preserve the relationships between different components of the physical system. These relationships form the link between component characteristics and device behavior and thereby guide design modifications to meet specifications. We believe that such techniques may form a framework for design-analysis tools capable of performance evaluation and may facilitate the identification of useful design modifications during the conceptual phase of design.

Requirements of a Conceptual Design Environment

The requirements of a conceptual design environment and the representation developed to meet those requirements should be derived from an understanding of the conceptual design approach used by human designers. It is crucial that the environment allow maximum freedom for the creativity of the designer.

To facilitate the initial description as the device is conceptualized, the representation should require a minimum of input and should accommodate a language that is natural to the domain of interest. For the design of electro-mechanical devices such a language includes the specification of a set of components and the way in which components interact.

A component usually represents the smallest unit a designer will consider (although one designer's component may be another designer's system). Subsystems are developed by aggregating components. A conceptual design environment should allow the modular aggregation of components, giving the designer the flexibility to try different instances of the same type of component, *e.g.* different types of motors, different types of springs.

Such an environment should also allow the designer to *abstract* or remove detail from specific components in order to focus attention on particular characteristics of components. Abstraction is critical to avoid obfuscation of the relations between behavioral and geometric component parameters. We discuss two types of abstractions used to deal with complexity [Paz-Soldan 89]:

- **Functional perspectives:** The set of behavioral properties of a component that are relevant to the designer depend on the intended use

---

[1] This could possibly explain why designers tend to patch-up existing solutions rather than experiment with entirely new and innovative configurations [Paz-Soldan 87, Ullman 87].

or functionality of the component For example, at different stages of design refinement and after having selected a particular motor for the application, the designer will consider different behavioral aspects of the motor depending on the type of issues he is considering: thermal, electrical, kinematic, and so forth. Different behavioral component models are required to support each of these functional perspectives of a single component This multiplicity of modeling views (compounded usually with the need for complex mathematical analysis) complicates the task of verifying the functionality of the preliminary configuration. The conceptual design environment should allow the use of multiple functional perspectives in order to allow the selective emphasis on important aspects of a component's behavior.

- Variable resolution: Within a given functional perspective, designers vary the amount of detail of a mental model of a component or a set of components. For example, consider a model for a motor. Within a given functional perspective, say the kinetic perspective, the designer can mentally model a motor with varying degrees of resolution. He may initially assume it to be an ideal source of torque. Later, he may want to include the motor's internal losses or its clcctrodynamic effects. A conceptual design environment should allow the use of models of varying resolution if it is to provide a natural modeling environment for the designer to establish limits on device performance or to model components that have not been completely specified.

Requirements of an Automated Modeling Tool

A design environment built to simplify configuration evaluation has to support the use of good modeling skills. The essence of modeling of any sort is to include only the effects relevant to the question posed. This involves a good amount of engineering judgment and manipulation skill. Typically the designer or modeling expert determines on an ad hoc basis which effects must be included to correctly predict device behavior. For example, while studying the dynamics of the print-head positioner, the motor mass might be neglected. On the other hand, the rotor inertia contributes significantly to the behavior and so it must be considered. Modeling insight allows the engineer to concentrate on the relevant aspects of the problem, while ignoring dimensions where essentially nothing interesting is happening. Making such simplifying assumptions involves, in part, reasoning about and deleting parts of the model which either correspond to constrained degrees of freedom or degrees of freedom which are not excited. The inertia of the printer casing, for instance, is of no relevance in the study of print-head dynamics. If modeling is to be automated and is to provide useful feedback, it is necessary that we are able to include all effects which are relevant and automatically exclude those which are not relevant Irrelevancies in the model not only increase the effort required for analysis and simulation but also confound efforts to identify component characteristics that critically affect behavior.

Approach

Representation

The requirements discussed above are met in large measure by a careful choice of the underlying representation of the designed object The natural interface and modeling flexibility requirements are satisfied by having a component based representation where each component is represented as a collection of geometric and behavioral primitive elements. Electro-mechanical devices tend to be aggregates of standard components, *e.g.* motors, pulleys, gears etc. It seems natural, therefore, to also design this class of devices by modularly aggregating such components. Modularity in representation not only gives a structure to the cognitive process but at the same time enhances the flexibility available to the designer.

Mechanical devices interact with each other at their interfaces. The interactions can be specified in terms of forces and velocities which exist over time. If a component is removed from a complicated device and is replaced with some other component which effects precisely the same force and velocity characteristics as the original component, then the device behavior as a whole will not be changed. Thus a modular component-based representation allows a designer to consider alternate designs *e.g.* one employing a stepper motor rather than a d.c. motor, without having to recreate the entire model. As a modeling artifice, it also allows a component or set of components to be considered at varying levels of resolution.

Flexibility requirements that arise from the need to consider different functional perspectives of the same component can be implemented in two ways. The first one is similar to the implementation of variable resolution: Several different behavioral models are stored in the component model library which emphasize particular behavioral aspects of the component, depending on the intended use. For example, we can store two different behavioral models for a piece of tubular piping: one which implies its use as a conduit for various fluids, and one which models it as a massive body with inertial characteristics.

The second way to implement functional perspectives is to use a very general component model that incorporates several potential uses. When the component is connected to other components, the particular connections connect only certain sections of the internal behavioral model. When the subsystem is defined and the individual behavioral component models are collected, the unnecessary sections of the behavioral models can be removed automatically.

The first method, with the simple ad hoc model for each perspective, gives more control over the internal modeling process to the designer. It is quite general and a wide variety of perspectives can be accommodated in this fashion, but it also necessitates that the designer decide which physical effect are relevant and significant enough to be modeled. The designer has to be aware of the implications of choosing a particular behavioral model for a component. Making a design modification in some component of the device or simply altering the intended use of the device may require that the models of other components in the device be redefined. For instance, modeling the stiffness of the toothed belt may not be important when a stepper motor is used, but becomes critical with a d.c. motor because the control mechanism changes from open-loop to closed loop. To be able to model the device behavior correctly the model of the toothed belt must be changed appropriately.

A drawback of the second approach is that, since it is almost impossible to envision all the possible uses of a component, the underlying behavioral representation may not reflect the designer's intended function for the component. On the other hand, the complex general model can free the designer from modeling details but only if there is an algorithmic way to remove superfluous modeling representation or analysis results.

Within a given perspective, a component may have behavior that is:

- Relevant and significant.
- Irrelevant.
- Relevant but insignificant.
- Non-existent (because of context).
- Irrelevant and insignificant.

The relevant and significant behavior is the only one in which the designer is interested. Behavior irrelevant from one perspective may, however, be relevant from another *e.g.* heat conduction in an electrical perspective. Insignificant behavior will manifest itself only under sufficient resolution and does not affect the gross characteristics of the device *e.g.* torsional oscillations of the motor shaft in the kinetic perspective. A particular behavior, that is usually associated with a component, may become non-existent because of the context, *i.e.* because of other components and connections among components. Gears which are arranged such that none can rotate is an example of a context causing a behavior to be non-existent. This absence of behavior emerges from the nature of the components and the kinematic connections between them but is not inherent in and cannot be deduced from the behavior of any individual component. Deletion of all these uninteresting behaviors may be done pre- or post- simulation, however, model simplification prior to simulation helps to isolate the characteristics that critically affect behavior and thereby aids designers in identifying superior configurations.

Form - Function Relations

A component based representation also accommodates the use of pre-compiled form-function relations. Engineering components can be characterized by these relations whose origins lie in the physical laws that govern their behavior [Rinderle 87, Colburn 90]. These relations are of particular value in the conceptual design stage where the designer-analyst is interested only in general trends and not in specific numerical values. As an example, the motor mass and torque capacity relations discussed in [Colburn 90] would be used in the print-head positioner example not only for analysis of different configurations but also perhaps to deduce that acceleration gains over a threshold value are likely to be modest because the increased torque capacity is offset by increased inertia of the motor itself.

Description of Domain and Solution Techniques

In our demonstration software, we address questions pertaining to aggregation of behavior but not form-related issues such as geometric interference. Furthermore, the behavioral primitives are restricted to model planar dynamics. As is clear from the print-head example, evaluating the dynamic behavior of a set of components functioning as a group is useful to the designer of such a device. The various representation requirements detailed above can be satisfied for this domain through a novel use of bond graphs [Paynter 61, Rosenberg 75], a formal graph based representation used for physical system modeling.[2] A modular fragment of a bond graph is associated with each component and each type of kinematic connection. As components are connected the individual models are connected into a device model. Bond graph theory provides a consistent basis for this process of aggregation, so that the component model fragments and kinematic constraints between components specified by the designer are sufficient to assemble the overall bond graph model of the device.

As has been pointed out previously, the appropriate model for any physical component depends on the context. An inertial body may be so constrained that only rectilinear translation is possible. On the other hand the same body may undergo complicated motion in a plane. We have addressed this issue by adopting a model which is general enough to model complex motion but can also correctly model the simpler case of rectilinear translation. Bond graph specific issues in

---

[2] **A bond graph is a lumped parameter model of a dynamic system in terms of idealized sources, energy storage elements, transformers, gyrators and dampers very much like an electric circuit diagram. Besides being formally defined, bond graphs are also broadly applicable across a range of energy domains. Several researchers have used bond graphs in design related research [Rosenberg 75, Finger 89, Ulrich 89, Hoover 89, Macfarlane 89, Prabhu 89, Hood 87]. While [Finger 89, Ulrich 89, Hoover 89, Prabhu 89] address issues of design synthesis and use bond graphs as the representational framework for their synthesis strategies, [Macfarlane 89, Hood 87] use it as a tool for analysis.**

creating such models and the simplification algorithms that allow inferences as mentioned above have been detailed in [Rinderle 91, Rinderle 90]. Simplification is necessary because the use of general models creates much that is superfluous. There are methods to obtain state-space equations from well constructed bond graphs but these methods are computationally expensive if dependent energy-storage elements are present and fail completely for certain classes of graphs.[3] Simplification methods detailed in [Rinderle 91, Rinderle 90] delete parts of the model which are explicitly or implicitly constrained to be static and replace dependent energy storage elements (inertias and compliances) by their equivalences. While deleting explicitly constrained sections is merely a matter of propagating the effects of a *ground* connection, implicit constraints or what we call constraining junction structures arise when kinematic constraints on a multi-degree of freedom body interact to impose zero velocity on elements which are not explicitly constrained. A constraining junction structure would be formed, for instance, if two gears keyed to a shaft are meshed with two other gears which are keyed to another shaft. Fortunately even these can be identified and deleted algorithmically and without much computational expense. Replacing collections of energy storage elements by equivalences involves, in part, identifying parallel-series and star-delta formations. In other cases the dependent inertia cannot be simply *added* to a single inertia but must be distributed among several of them such that the Lagrangian of the system remains invariant. These transformations make the bond-graph model more comprehensible and symbolic equation formulation computationally less expensive.

The representation and the simplification techniques are general enough to handle non-linear behavior as well. Non-linear behavior in such systems may arise from non-linear constitutive laws describing specific components. It may also arise from the nature of component connectivity, as for instance in a four-bar mechanism. Finally changes in component connectivity can also produce non-linear behavior *e.g.* in a Geneva mechanism. While the first two cases essentially fall within the modeling framework described above, the third one will simplify to two or more different models - the active one being decided by the state of the system.

---

[3] This class includes bond graphs with what we call constraining junction structures [Rinderle 91, Rinderle 90]. These junction structures represent kinematic constraints which although self-consistent within the bond graph framework represent either redundant constraints admitting motion or conflicting constraints precluding motion.

A Design Environment

We have augmented a commercially available CAD environment to support automated modeling and analysis of preliminary design configurations. The designer composes a configuration from library components by specifying the type, location and geometric characteristics of the components and the kinematic connections among components. In each case, the dynamic model is a bond graph such that the number of external connections that are allowed correspond to the connectivity of the physical component that it represents- This topological correspondence makes it easy to describe the designed artifact in terms of components and connections among components. When desired, a dynamic model of the device is automatically constructed from the model fragments corresponding to the individual components and the kinematic connections. The aggregate model is simplified to eliminate irrelevant characteristics and to identify common effects so as to simplify the relations between behavior and configuration parameters. The simplified model is then reduced to a set of differential equations whose characteristics may be evaluated or which can be solved numerically or symbolically.

Example: Printhead Positioner

Figure 2 shows schematic representations of the dynamic models corresponding to the two printer configurations illustrated in Figure 1. The components which give rise to the *guiding* function of both printers are identical. They consists of a printhead platform and four guide rollers which run along a tray like structure within the printer. Each of the rollers and the platform itself are massive bodies and in general can move in a plane.[4] The rollers and the printhead platform are each defined in terms of the primitive massive body element which has degrees of freedom corresponding to rotation in the plane, 0, and two independent translation^ motions, X and Y. After having selected the appropriate components the designer establishes kinematic connections among the components. Each roller, for example, is *pinned* to the printhead platform. The nature of a *pinned* connection is that it constrains translational velocities to be identical and imposes no constraint on relative rotational velocities. Therefore, the *pinned* connections shown in Figure 2 show connections between the X and the Y ports of the pinned objects. Similarly *rigid* and *rolling* connections can be established between components. A *rigid* connection constrains all three degrees of freedom. A *rolling*

---

[4] We consider only planer motion in this simple example and in our preliminary implementation. In the case of the printer we presume that other components, for example wheels beneath the printhead platform, maintain the planer orientation of the platform. The actual component models for planar motion and detailed explanations about them can be found in [Rinderle 91, Rinderle 90]. Component models for spatial motion are complicated and large but present no theoretical difficulties.

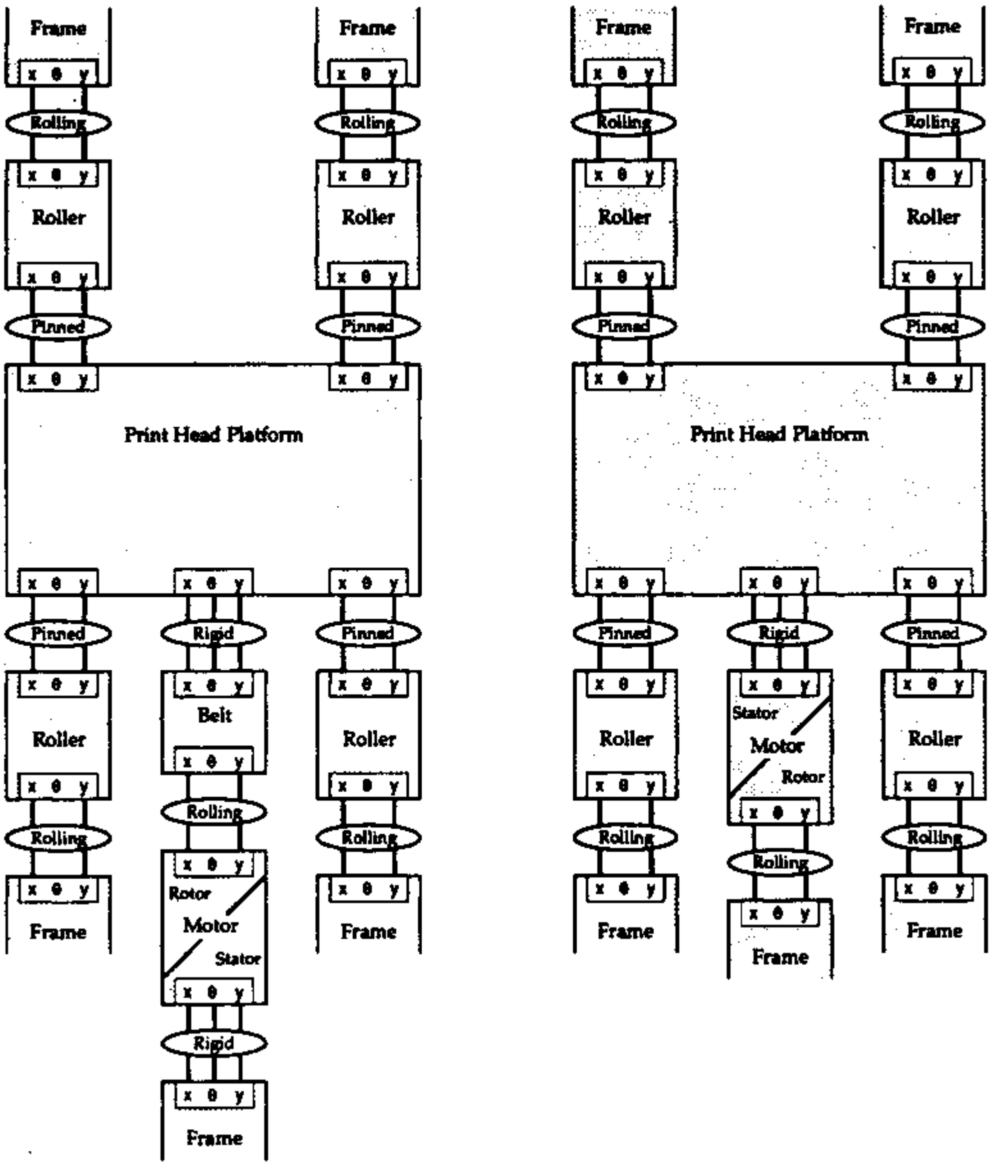

Figure 2: Schematic representation of printer models corresponding to the belt drive (left) and the traction drive configurations shown in Figure 1

---

connection is similar to a *pinned* connection in that it constrains relative translational velocities, however, in the case of rolling the position of common velocity is at the surface of the roller rather than at the axis of the roller. The figure also shows more elaborate, composite components, for example the motor. The motor includes two massive elements, specifically the rotor and the stator and a source of torque which acts between them. This composite element is built into the system model just as any other component is. The modularity of this approach makes it possible to consider models of varying resolution for each of the components without much difficulty.

Because **each** of the components in the device has three independent energy storage modes **the** "first cut" model of the device comprises twenty-one energy storage modes. Many of these are of course either irrelevant or redundant because the components cannot all move independently of each other. It is not, however, necessary that the designer indicate which motions are possible: The kinematics of the configuration itself determines which are admissible. These degrees of freedom are identified algorithmically by propagating kinematic constraints, by combining inertias, and by identifying kinematically redundant structures and kinematically constraining structures. In the case of the printhead positioning devices, massive elements joined by rigid connections are combined and massive elements constrained to remain at zero velocity by virtue of a ground or frame connection are eliminated. We then identify constraining junction structures. In the case of the printhead platform it is obvious to even the most novice designers that the platform will not rotate, however, that constraint is not explicit. Rather the absence of rotation emerges from the combined effects of the rollers. The combination of rollers and the platform results in a junction structure precluding rotational motion, therefore, the rotational inertia of the platform may be deleted. After these simplifications the only inertial elements remaining are excited by a single degree of freedom, translation of the device. Further simplification can then be obtained by eliminating so called dependent energy components via a transformation which preserves the Lagrangian of the device. In simpler cases, such as the printhead positioners, it is possible to combine inertias by simple transformations based on the proportionate velocity of adjacent devices. By applying these simplification procedures prior to analysis it is possible to reduce the original 222 port, 294 bond, twenty-one order model of the belt driven positioner to a two port, one bond, first order model comprised only of a force source and an equivalent inertia given by:

$$ {}^{l}\textit{eq}'' = {}^{*}\textit{x.printhead} + \overset{*j}{\wedge^{l}\textit{x.roller}}^{+} \frac{\wedge'\textit{Q.roller}}{\mathrm{T}\wedge \atop \wedge\textit{roller}} + \frac{'\textit{Q.pinion}{*}\mathrm{A}).\textit{motor}}{73 \atop \textit{Opinion}} $$

The structure of the simplified model of the traction drive is identical to the structure of the belt drive model, however, in the case of the traction drive the equivalent inertia also includes the mass of the motor. The contribution of motor inertia and motor mass to the equivalent inertia is explicit in these relationships and provides a basis upon which the designer can determine the relative merits of these alternative configurations. In most cases the effect of this additional inertia is negligible in relation to other inertias arguing for the adoption of the kinematically and geometrically simpler traction drive configuration.

Summary and Conclusion

We have presented here some issues and ideas underlying a tool to aid conceptual designers. We observe that understanding and isolating the contribution of individual components to device behavior provides a powerful guide to the synthesis effort. The benefits of a modular component-based representation in achieving a natural interface as well as in accommodating the contextual and perspective flexibility requirements have been established. The ideas have been demonstrated with a prototypical design tool for automated modeling and analysis of dynamic systems. In our design environment we define as primitives idealized physical components and kinematic connections in terms of bond graph fragments that describe their behavior. The user specifies the kinematic connections between different components of the design and the system translates this into a procedure for aggregating component-level models to form a device-level model. Simplifying the model facilitates drawing inferences about the dominant behavior and makes analysis and simulation more tractable. Characterizing the resulting equations of motion symbolically will enable us to evaluate design trade-offs and determine high level form-function relationships for the designed device.

Acknowledgments

The authors are pleased to acknowledge the support of the Design Theory and Methodology Program of the National Science Foundation (NSF Grants DMC-84-51619 and DMC-88-14760) and the Engineering Design Research Center at Carnegie Mellon University (NSF Grant CDR-85-22616). The authors are also very grateful to Steve Ray and Steve Hoover. The former for his help in the software implementation and the latter for his many thoughtful comments.

References

[Asimow 62]
Asimow, Morris, *Introduction to Design,* Prentice Hall, Inc.,, 1962.

[Bell 81]
Suh, NP., Bell, A.C., Wilson, D.R. and Rinderle, J. R., "Exploratory Study of Constraints on Design by Functional Requirements and Manufacturing," NSF Progress Report DAR-77-13296, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1981.

[Colburn90]
Colbum, E. R. and Rinderle, J. R., "Design Relations," *Proceedings of the Second ASME Design Theory and Methodology Conference,* September 1990.

[Finger 89]
Finger, S. and Rinderle, J. R., "A Transformational Approach to Mechanical Design Using a Bond Graph Grammar," *Proceedings of the First ASME Design Theory and Methodology Conference,* September 1989.

[Hood 87]
Hood, S. J. and Palmer E. R., "The Design Analysis for Reliability Tool," *Proceedings of the American Control Conference,* June 1987.

[Hoover 89]
Hoover, S. P. and Rinderle, J. R., "A Synthesis Strategy for Mechanical Devices," *Research in Engineering Design,* Vol. 1, No. 2,1989, pp. 87-103.

[Johnson 71]
Johnson, Ray C, *Mechanical Design Synthesis with Optimization Applications,* Van Nostrand Reinhold Company,, 1971.

[Macfarlane 89]
Macfarlane, J. F., *Qualitative and Symbolic Analysis of Dynamic Physical Systems,* PhD dissertation, University of Minnesota, 1989.

[Mann 77]
Mann, R. W, "Engineering Design," *McGraw-Hill Encyclopedia of Science and Technology,* 1977, pp. 700-707.

[Pahl84]
Pahl, G. and Beitz, W., *Engineering Design,* The Design Council, Springer-Verlag, London, 1984.

[Paynter61]
Paynter, H. M., *Analysis and Design of Engineering Systems,* MIT Press, Cambridge, 1961.

[Paz-Soldan 87]
Paz-Soldan, J. P., "The Use of Functional and Geometrical Dependencies in Conceptual Design: A Verbal Protocol Study," Internal Lab Report CMU-MEDL-88-19, Department of Mechanical Engineering, Carnegie-Mellon University, 1987.

[Paz-Soldan 89]
Paz-Soldan, J. P. and Rinderle, J. R., "The Alternate Use of Abstraction and Refinement in Conceptual Mechanical Design," *American Society of Mechanical Engineers,* 1989, Paper Number 89-WA/DE-8

[Prabhu 89]
Prabhu, D. R. and Taylor, D. L., "Synthesis of Systems from Specifications Containing Orientation and Positions Associated with Power Flow," *Design Automation Conference,* 1989.

[Rinderle 87]
Rinderle, J. R., "Function and Form Relationships: A Basis for Preliminary Design," *Proceedings of the NSF Workshop on the Design Process,* Waldron, M. B., ed., Ohio State University, Oakland, CA, February 8-10 1987, pp. 295-312.

[Rinderle 90]
Rinderle, J. R. and Balasubramaniam, L., "Automated Modeling to Support Design," *Proceedings of the Second ASME Design Theory and Methodology Conference,* September 1990.

[Rinderle 91]
Rinderle, J.R. and Balasubramaniam, L., "Automated Bond Graph Modeling and Simplification to Support Design," *Journal of Dynamic Systems, Measurement, and Control,* 1991, (under review)

[Rosenberg 75]
Rosenberg, R. C, "The Bond Graph as a Unified Data Base for Engineering System Design," *Journal of Engineering for Industry,* Vol. 97, No. 4, November 1975, pp. 1333-1337.

[Ullman 87]
Ullman, D. G., Stauffer, L.A. and Dietterich, T. G., "Preliminary Results of an Experimental Study

of the Mechanical Process," *Proceedings from the NSF Workshop on the Design Process,* Waldron, M. B., ed., Ohio State University, Oakland, CA, February 8-10 1987, pp. 157-200.

[Ulrich 89]
Ulrich, K. T. and Seering, W. P., "Synthesis of Schematic Descriptions in Mechanical Design," *Research in Engineering Design,* Vol. 1, No. 1, 1989, pp. 01-16.