Automated Modeling to Support Design

by

J. R. Rinderle, L. Balasubramaniam

EDRC 24-31-90 C. 3

Proceedings of the Second ASME
Design Theory and Methodology Conference
Chicago, Illinois
September, 1990

Automated Modeling to Support Design

James R. Rinderle
Associate Professor
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, PA. 15213

L. Balasubramaniam
Research Assistant
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, **PA.** 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 behavior of an artifact would aid the conceptual designer by facilitating the consideration of more varied alternatives. We describe a tool for automatic formulation of dynamic system models from user specified interconnection of components. The underlying methods are based on Bond Graph formalism using causal reasoning to simplify the model. The simplified model is not only more tractable for analysis and simulation but also facilitates inferences about the dominant characteristics of the designed artifact.

Introduction

Conceptual design encompasses the development and evaluation of different design configurations. 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. Human designers are good at visualizing the geometric interaction among the components but they are not as good at understanding how the components function together as a group. In this paper we present some ideas underlying a design tool for evaluating the dynamic performance of a mechanical device. We believe that numerical simulation models, useful though they are, do not promote insight about design trade-offs. Models that preserve the relationships between the different components of the physical system seem to be more appropriate for this task. This research focuses on techniques that allow building and analyzing symbolic models of dynamic behavior given a description of the artifact in terms of its components and kinematic connections among them.

An Environment for Conceptual Design of Dynamic Systems

A conceptual design environment [Paz-SoHUn'88a, PazSoldan 88b] should meet certain primary requirements: First, it should allow the designer to interact in a natural and convenient fashion using sketches, references to standard components and brief description of custom-made components. A second set of requirements for such an environment relates to modeling flexibility. The user should be able to refine the modeling deuil is well as aggregate simple component models into more complex components that can be subsequently thought of and manipulated as a unit. Finally, it should give useful feedback about the performance of the designed artifact.

A design environment built to simplify configuration evaluation has to emulate the modeling skills of a good engineer. Modeling of physical systems involves a good amount of engineering judgment and manipulation skill. Engineering judgment is called for in choosing an appropriate model for the situation under consideration. While modeling a bicycle, such as shown in Figure 1, ihe bicycle frame can be considered as a single inertia or as a large set of point masses in constrained motion. Modeling insight allots ihe 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 pan. reasoning about and deleting parts of the model which either correspond to constrained degrees of freedom or degrees of freedom * hich are not excited. The model of the bicycle, for example, may or may not include the rotary inertia of the frame depending on the boundary conditions of the problem. Proficient modelers do this frequently and make analysis simpler. The design environment should emulate this behavior as much as possible.

Approach

The natural interface and modeling flexibility requirements can be satisfied by adopting a component-based representation *here each component is represented as a collection of geometric and (~r,caonal primitive elements. The designer can build up a doue by aggregating components, a process which will involve specifying spatial component locations and the connections *between* components. The process of aggregation thus proceeds on two

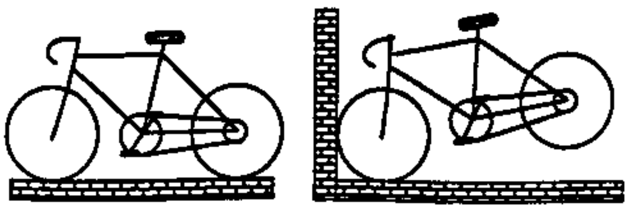


Figure 1: A Bicycle with Two Different Boundary Conditions

fronts: Component geometry is assembled to obtain device form and component level behavior is combined to model device function.

The flexibility requirements can be satisfied since the dynamic and geometric model of each component can be individually modified without having to change the way the components are aggregated. The integrated representation of geometry and behavior models also facilitates using form-function component relations to relate parameter values [Rindcrle 87, Colburn 90).

The modular aggregation and arbitrary resolution of dynamic models is accomplished through a novel use of Bond Graphs[1] [Paynter 61, Rosenberg 75] a formal graph based representation used for physical system modeling. A modular fragment of a Bond Graph is associated with each component and as the components are connected the individual models are collected into a device model. Bond Graph theory provides a consistent basis for this process of aggregation, so that the kinematic constraints between components, specified by the designer, are sufficient information for assembling the component bond graph modules. The aggregate Bond Graph is simplified and reduced to a set of differential equations which are used to reason about the dynamic behavior of the designed system.

Issues

The first of the two major issues is to develop models for the primitive set of components and kinematic connections. The second *is* to simplify the first-cut model produced so that dominant behavior is clearly brought out

Bond Graph Representation

The number of power connections that can be made to a primitive bond graph element is not the same as can be made to the physical component which it represents. Most engineers, for example, would consider a spring to be a two-port device corresponding to the two ends which can be connected. In a bond graph, however, the compliance element, C, which represents a spring is a one port element A natural interface demands that the user continue to think of a spring as a two-port device. To overcome this problem we use bond graph fragments with topology similar to the physical components which they model. Figure 2 shows a "two-port-spring" model. The -0- is a power conserving, common force multi-port element. As such, the bond graph fragment shown in Figure 2 requires that the same force acts on both ends of the spring and that the spring velocity is actually the difference of the velocities at the two ends. Figure 3 shows the bond-graph mass model which is somewhat more complicated. The model is a three-port. Each port corresponds to a distinct location and includes three bond-graph connections to accommodate rotation as well u independent X and Y velocities. The forces at these ports excite the rotational and txmslational energy storage modes of the mass. The degree to which a force at one of the pom causes rotation or translation depends on the location of interaction. The -TF- elements in the model account for this by transforming an arbitrary force and moment at a position to equivalent forces and moments about the center of gravity. The moduli of the transformer (TF) elements depends only on the location of force/moment application relative to the center of gravity. A mass may of course interact with other components at an arbitrary number of points. We define a composite mass to accommodate this arbitrary degree of connectivity by establishing rigid connections among many mass elements and thereby preserve the natural designer interface. The small hatched circles at the end of the bonds represent a consistent power-sense assignment

Kinematic connections are also modeled as bond graph fragments. The model of a connector has a «1«, or common velocity, junction corresponding to every velocity, X, Y or rotational, that it constrains. A pinned connection includes two -1* junctions corresponding to common X and Y velocities. A rigid connection, as in Figure 4, includes an additional -1- junction because rotation of two rigidly connected masses is identical.

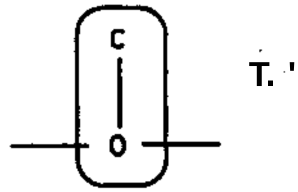


Figure 2: Internal model of a Spnng

Need for Simplification

The component models that we use as building blocks have to be general enough to be able to serve in many different situations. Hence, the first-cut models produced by our implementation lack the conciseness of the models made by proficient modelers. The models will contain constrained elements which will not be excited at ail for the configuration under study. They will also contain a number of energy storage elements, masses and springs, connected so that they cannot all be independently assigned energy variables. The standard techniques of equation formulation from bond graphs, are computationally expensive if dependent elements are present and fail completely for certain classes of Bond graphs.[2] Because the first-cut graphs we produce often contain such intractable elements it is necessary to simplify the graph before we formulate the equations of motion.

[1] A bond graph is a lumped parameter model of a dynamic system in terms of idealized sources, energy storage elements, transformers, gyraion and dampers very much like an electric circuit diagram. [Rosenberg 83] is an excellent introduction to this subject.

[2] This class includes bond graphs, with constraining junction smjcturts. These junction structures, we show later, are not really n-pons as the behavior of internal variables is independent of what is externally imposed and some effort How variables cannot be expressed in terms of state variables.

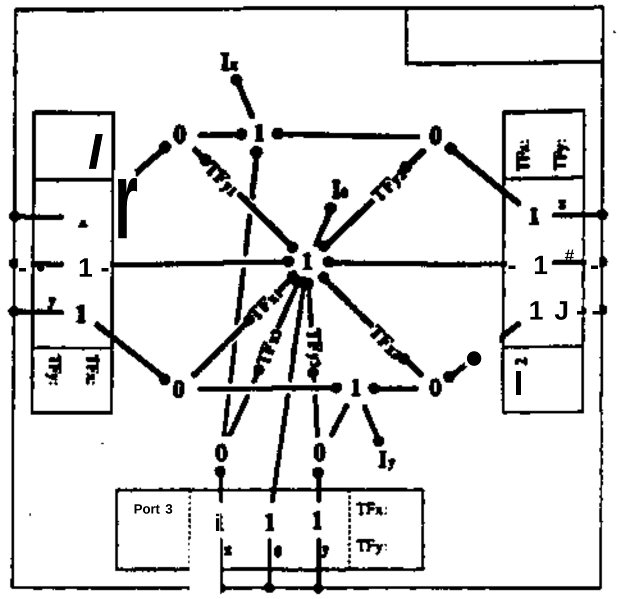


Figure 3: Bond graph model of a massive body

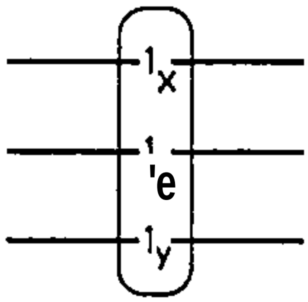


Figure 4: Bond graph model of a rigid connector

Techniques of Bond Graph Simplification

We have identified two classes of simplification procedures, both of which arise from studying the causal assignment in a bond graph.

Simplification: Inert bond graph segments

The first step of simplification is to remove bonds and junction structure elements which have not been connected. Because each of the ports on the model represents a point of mechanical interaction, the absence of a connection indicates that no force is applied at that point in the associated direction. An open port therefore behaves exactly as if a source of effort of zero magnitude was applied. Rather than explicitly including those effort sources and subsequently pruning the bond graph we choose to directly simplify the graph.

An effort source of zero magnitude incident upon a •!- junction has no effect, and can be deleted because efforts at a -1- junction must sum to zero. Furthermore, because the variable associated with a -I- junction is a velocity, imposing an additional effort on that node does not affect any aspects of causality assignment We use that reasoning in combination with the deletion of two-port -1- junctions and two-port 4» junctions (being careful to respect sign convention) to delete the unconnected ports and their extensions until reaching a •0- junction in the graph. The effect of the applied source of effort on a -0- junction is to impose a zero effort on all of the other power bonds emanating from that junction. We can therefore continue our deletion through -Cj- junctions on all other paths until reaching -1- junctions. In a similar manner, a source of zero velocity attached to a -0- or -1- junction can also be propagated to delete several ports.

Inert elements also arise when kinematic constraints on a multi-degree of freedom body interact to impose zero velocity or effort on elements which are not explicitly constrained. This would arise, for instance, if two gears keyed to a shaft are meshed with two other gears which are keyed to another shaft as illustrated in Figure 5. Neither shaft can route unless the gear ratio of one of the meshed pairs is exactly the same as that of the other. The way the gears are coupled imposes zero velocity on them. In bond graph terms the connected set of -0% -1- and TF junctions, henceforth referred to as a junction structure, representing the kinematic connection constrains the attached elements. We seek to identify and delete bond graph elements and junction structures that represent such static parts. The constitutive equations describing a junction structure consist of a set of effort equations afd flow equations. In general the flow equations have the form:

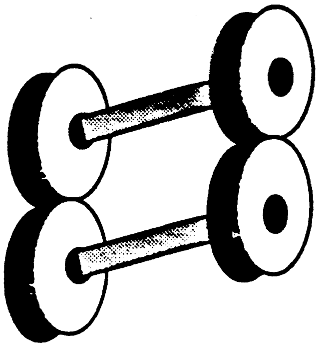
$$\begin{bmatrix} f_+ \\ f_I \end{bmatrix} = \begin{bmatrix} A_1 & A_3 \\ A_2 & A_4 \end{bmatrix} \begin{bmatrix} f_- \\ f_I \end{bmatrix}$$

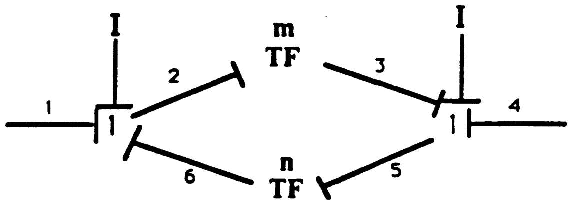
where [fj is the column of flows (velocities) imposed by the junction structure on the elements connected to it, [f J is the column of flows imposed on the junction structure, [fj] is the column of internal flows and Aj, A_2, A3, A_4 are matrices of constants. These equations express flow variables in terms of *other* variables more fundamental in a causal sense and hence the terms on the diagonal of the partition A_2 are all zero. Now, let there exist a junction structure in a bond graph with admissible causality on which all external bonds impose effort[3] In this specific case, [fj is a null column because none of the external bonds impose flow Then the flow equations, of this junction structure must take the form:

$$\begin{bmatrix} f_+ \\ f_I \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} [f_I]$$

Given this form of the flow equations it is possible to identify conditions for a junction structure to impose zero vclociry on all elements connected to it. Such a case is illustrated in Figure 5 which shows the bond graph and corresponding flow equauons of rwo shafts locked by two pairs of meshed gears.

[3] [Kamopp 75a] states thai such a junction structure cannot e»««. w Me thai argument is valid if bond graphs are created using the standard nJet prrwnbed in [Rosenberg 83], such junction structures can arise »hen borvJ &*?*& are assembled from sub-system models. Our contention is thai a jorwu^n structure such as this represents kinematic constraints which although self<om<urnt »nhm the bond graph frame-work represent either redundant consaiinu *lnuamg motion or conflicting constraints precluding motion.





$f_+ = [f_1, f_4]$

$f_I = [f_2, f_3, f_5, f_6]$

$f_1 = f_2$
$f_4 = f_5$
$f_2 = f_3\, m$
$f_3 = f_5$
$f_5 = f_6\, n$
$f_6 = f_2$

$$A_2 = \begin{bmatrix} 0 & m & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & n \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \mathrm{Det}[I - A_2] = 1 - mn$$

Figure 5: Two shafts locked by meshing gears

A junction structure cannot cause any element to have zero velocity if that element is imposing flow causality on the junction structure. This violates the definition of causality. Hence if the junction structure is to impose zero velocity on all attached elements then it is necessary that the elements impose only efforts on the junction structure. From the flow equation, which were derived for the above situation we have

$$[f_I] = [A_2]\ [f_I] \text{ and hence } \quad [I - A_2]\,[f_I] = 0;$$

If $\mathrm{Det}[I - A_2]$ is equal to zero then f_I is arbitrary and f_+ is not necessarily equal to zero. If $\mathrm{Det}[I - A_2]$ is not equal to zero then $[f_I] = 0$ and hence from the flow equation $[f_+] = 0$. From these arguments we conclude that the necessary and sufficient condition for a junction structure to impose zero velocity on all elements connected to it is that all external bonds must impose effort on the junction structure and $\mathrm{Det}[I - A_2] \neq 0$. For the example shown in Figure 5, we can see that the determinant is zero only if the gear-tooth ratios are exactly the same for the two pairs. In all other cases the two shafts are completely locked.

To determine if a junction structure is constraining, we first check that valid causality can be assigned to the junction structure. We then strip away all the elements and check that efforts are imposed on all the junction structure ports. This determines if the junction structure could be statically determinant. If flows are imposed on one or more of the ports, then these ports are deleted and the reduced junction structure is examined to determine if it is statically determinant or kinematically redundant.

Simplification: Dependent energy storage elements

The second issue, removing dependent energy storage elements, can be resolved because bond graph representation allows us to identify two primary classes of behavior preserving graph transformations to replace connected inertias and springs by their equivalences, and thus obtain a minimal representation of the system.

The first set of these transformations have analogues in electrical circuit theory: Energy storage elements connected in parallel or series are replaced by their equivalents, after moving them across transformers if necessary. Elements connected in star or delta formation can be considered as complex impedances. An I element has derivative causality when current or velocity is imposed on it. In a general star network of I's, current or velocity will be imposed on one I. Figure 6 shows a mechanical, electrical and bond-graph version of the star connection. The current in branch *c* is a linear function of the currents in branches *a* and *b*. In the dual delta form, as in Figure 7, all of the I elements have force or voltage as input and also have parameter values that are independent of the complex term $j\omega$ which cancels out in the transformation process. A similar situation exists when the energy-storage elements are connected, in parallel or series, so that they share the same effort or flow.

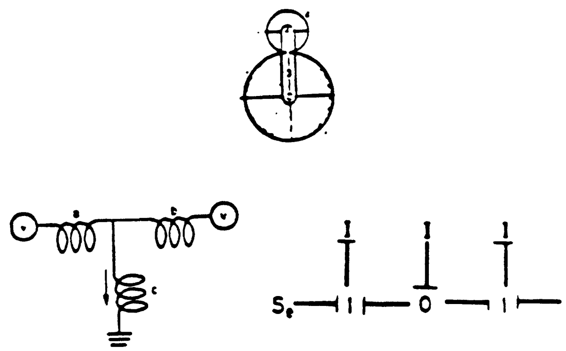


Figure 6: Mechanical, Electrical and Bond graph Star

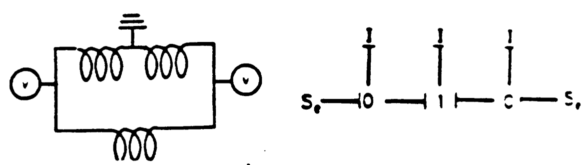


Figure 7: Equivalent Delta

The second set of transformations come from the energy expressions. In a number of cases of derivative causality, the energy contribution from the dependent element can be expressed as a linear combination of the energy contribution from more than one energy storage element. In this case the dependent inertia cannot simply be added to a single independent inertia but must be distributed among several of them. If the parametric values of the independent elements are suitably changed and the dependent element removed, the energy expressions will not change. Since graph simplifications that do not change the expressions for kinetic and potential energy leave the dynamic equations unaltered, this method can also be used to eliminate derivative causality. Simplification of a model of the three planet epicyclic gear shown in Figure 8 is i typical application of this transformation. In modeling this device the planets would be added incrementally. In the bond-graph, the three planets will show up as three different inertias which have some linear combination of the velocities of the sun and the carrier arm. The three planet-gears can be reduced to one by adding equivalent inertias to the sun, carrier arm and the remaining planet itself. This class of transformations is especially useful because the energy expressions give a global view of the possible simplifications and thus avoid graph walking.

It is possible to show that these transformations not only simplify the graph but also obviate solving sets of simultaneous equations symbolically in the equation formulation process. Thus, using the structure inherent in the Bond graph representation of the dynamic system, we have been able to make the task of equation formulation computationally efficient

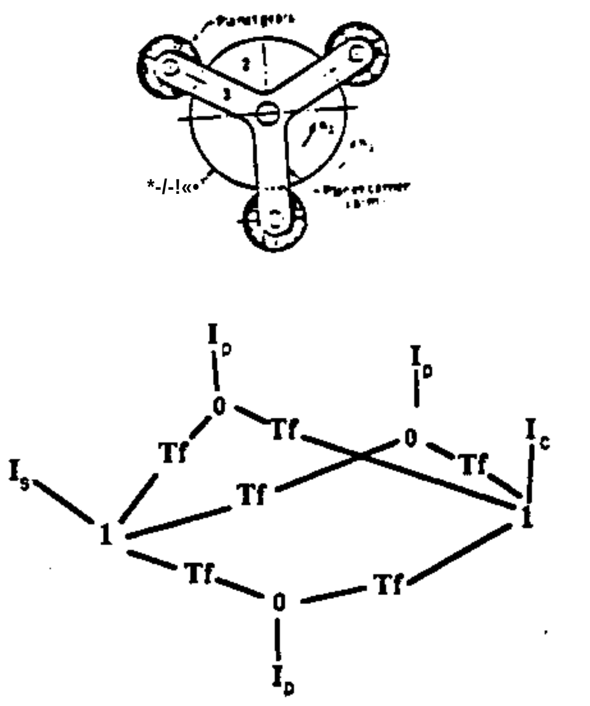


Figure 8: Epicyclic Gear-Train and its Bond Graph Model

Example: Bicycle

A device such as a bicycle shown in Figure 1 consists of a number of rigid masses which interact with each other and impose kinematic constraints on each other. Let us consider the motion of this bicycle in a plane arising as a result of a force applied to the pedal when the pedal arm is horizontal. We determine first which of the rigid massive components must be considered. The crank assembly, the frame, and each of the two wheels are massive bodies and each of them interact with at least one of the other bodies, the ground or the chain. Both wheels for example, interact with the ground at the point of contact between the tire and the ground as shown in the left-hand side of Figure 1. At that point the vertical and horizontal velocities of the tire are zero, assuming that there is no slip between the tire and the road and that the tire remains on the road. The wheel assemblies also interact with the frame at the axle where the wheel and the frame have identical horizontal and vertical velocities, however, no constraint is imposed on relative rotation. The rear wheel also interacts with the crank assembly through the chain. We have chosen to neglect the massive characteristics of the chain and to include the kinematic characteristics as a simple transformer element as indicated in Figure 9. Although we know from experience that there is no motion of the crank assembly relative to the rear wheel, these constraints are not imposed by the chain, and therefore are not included as kinematic constraints between the crank and the rear whecL[4] The crank assembly does, however, share common translational velocities with the frame. Lastly, the crank interacts with the applied force at the pedal. The basic multi-port massive model is used to represent the front wheel, the frame, the rear wheel and the crank assembly as shown in Figure 9. Also shown in the figure are bonds representing the kinematic constraints among these elements. The figure therefore represents a bond graph suitable for determining the motion of the bicycle resulting from a force applied to the pedal.

It is clear however, that this model is much more complex than necessary. Complexity is significant from the point of view of a designer or modeler trying to understand the behavior of the device, and also has significant ramifications on «quttion formulation techniques. A perusal of the figure shows that the overall bond graph is circuitous, redundant, exhibits a high degree of derivative causality, and includes elements and bonds which we know from our own experience have no relevance to the dynamic behavior of the bicycle. It is important therefore that we are able to simplify the bond graph representing the bicycle prior to the application of equation formulation techniques.

The first step of simplification is to remove bonds *nd junction structure elements which have not been connected. Next we consider the simplifications arising from the specific physical location of the interaction ports. The attachment locations are manifest as transformers in the multi-port massive body bond graph element All of these transformers are needed in the general case, however, the specific geometry of the bicycle causes majiy of them to be degenerate. For example, the vertical force applied to the tire at the point of contact with the ground does not contribute a torque on the wheel because the point of contact is directly beio* the axle. In this specific case the parameter TF_{x1} is zero. Because ihe torque is zero, the bond emanating from $TF_{x]}$ incident on the -1- junction has no effect since a zero effort applied to a •!- junction has no effect as discussed previously. Dual reasoning allots us to conclude that the deletion of the bond connecting TF_X, anJ ihe -0- junction will have no effect All other zero modulus transformer elements and their connected bonds may be deleted for etaciJy the same reason. These simplifications are shown in Figurt *10* The same figure also shows simplifications arising from the dc.caon of zero value flow source connections propagated through one junctions up to zero junctions as justified by reasoning d-iJ to ihe justification for deleting zero value effort source elements i^J iheu* connected bonds.

The resulting bond graph is a much simplified version of ihe original but is still too complex. The greatest degree of rr-jjnmg complexity is embodied in the frame element *here ihrce vernal elements and four bond graph circuits are present Ai:rowgh *e

'Redundant constraints can be included, however, (here are LiNuues »cU as benefits IO doing so.

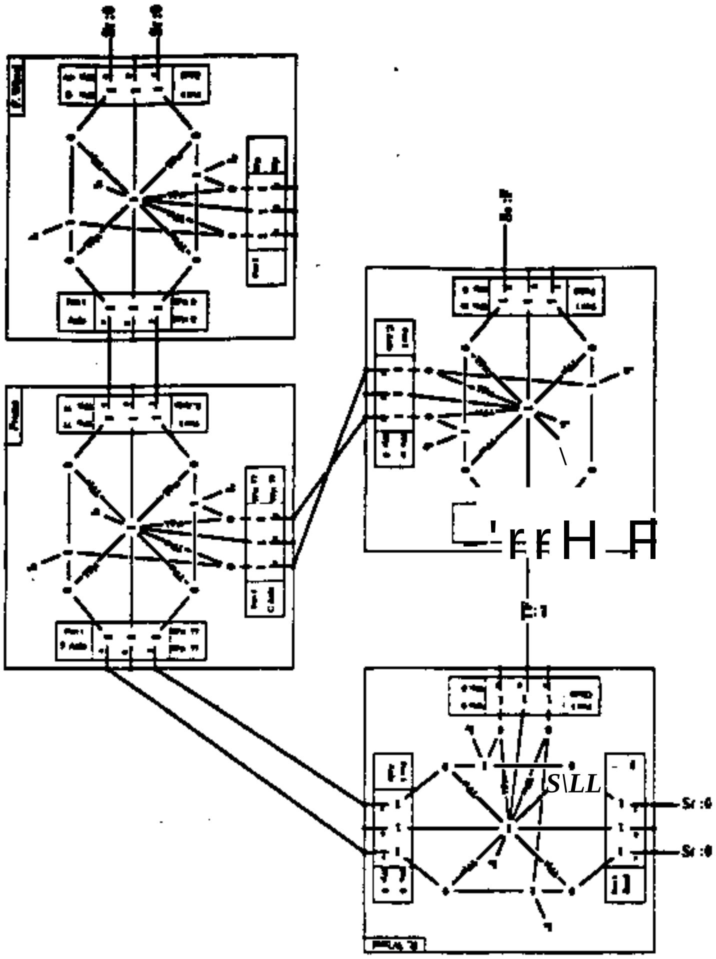


Figure 9: First Cut Model of the Bicycle

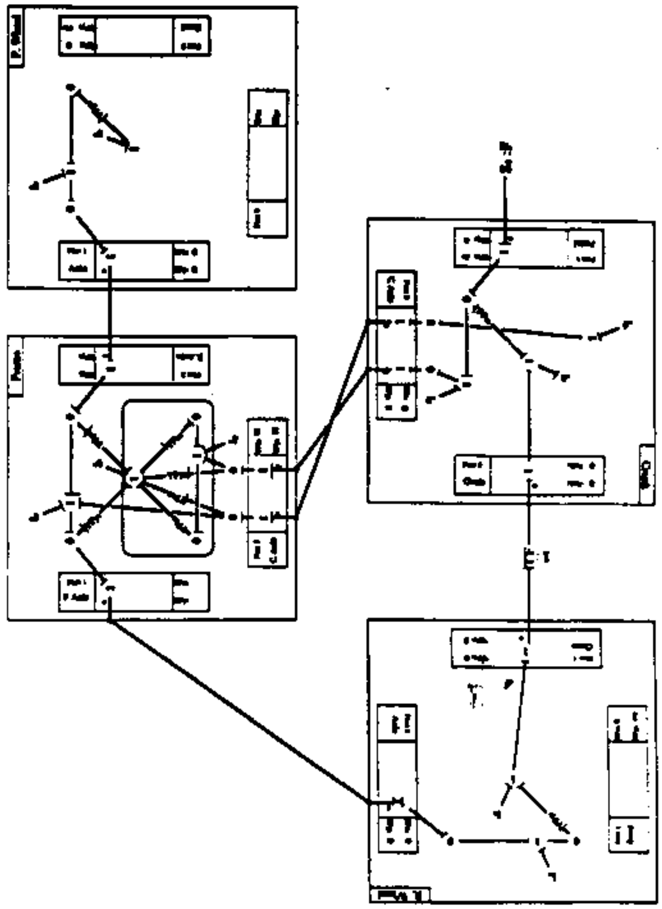


Figure 10: Partly Simplified Model

know intuitively that this is too complex (inertia in the vertical direction cannot for example be important for a bicycle moving horizontally in a plane) we need to identify unambiguously the source of these redundancies. The bond graph fragment highlighted in Figure 10 is a critical sub-structure.

Consider the slightly simplified version of the highlighted area shown in Figure 11 consisting of two -1- junctions, two transformer elements and two inertial elements. Assigning either of the inerrial elements integral causality results in an inconsistency and therefore cannot be valid. Assigning both of the inertial elements derivative causality admits a consistent assignment of causality but one in which causality is circuitous and in which equation formulation techniques would require i simultaneous solution of algebraic equations. In this particular case those algebraic equations can be satisfied only if the two transformer moduli arc reciprocals of each other. In that case this graph represents a kinematic redundancy. If not, the velocity corresponding to each of the •!• junctions must be identically zero. Going back to the bicycle problem, our algorithm will isolate the sub-graph shown in Figure 12. The determinant of [I - $A_2$1 in this case is given by (1- the product of the moduli of TF_{xl} and TF^). The determinant is zero only if the horizontal position of the front axle and the rear axle are identical. This clearly cannot be the case for a stable bicycle. Hence the entire structure shown in Figure 12 can be deleted from Figure 10 to arrive at the simplified bond graph shown in Figure 13.

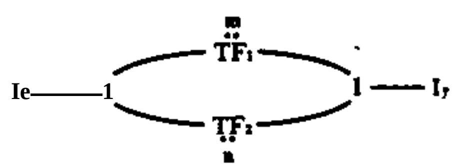


Figure 11: Simplified Example of Constraining Junction Szructure

Although spread over most of the page, Figure 13 is reaily quite simple. It consists of an acyclic bond graph with inertia] elements, transformer elements, a force source and the junction structure. If we view this structure as a tree emanating from the force source as a root, we may simplify the tree by moving from the leaves -p toward the root by determining equivalent inertias across transformers and by combining inertias. We arrive at the bond graph sho*n »n Figure 14. This simplest of bond graphs shows us that the bicyclc behaves simply as an inertia. The effective inertia depends on the translational mass of all elements as well as the rotational inertia of the front wheel, the rear wheel and the crank. 7>e relative importance of these inertias is determined by the transformer moduli determined by the chain sprocket ratio, crank arm length, and the wheel diameters.

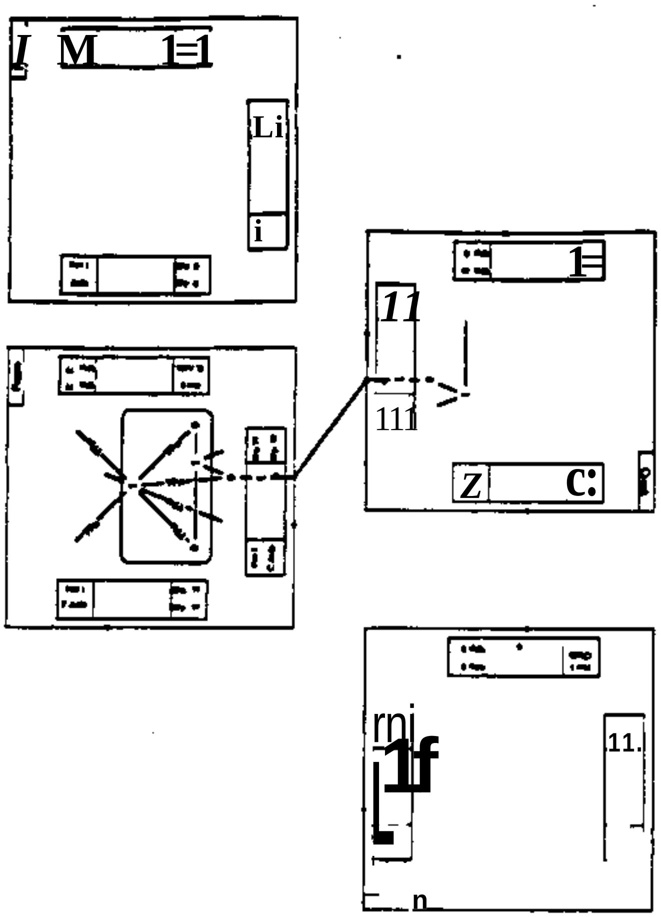


Figure 12: Constraining Junction Stmcture

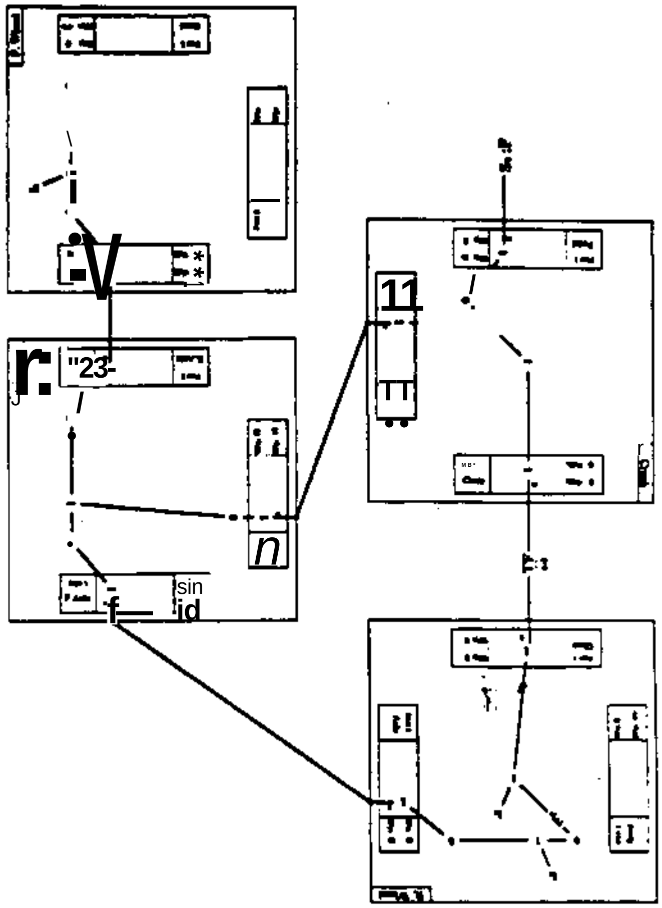


Figure 13: Model without Statically Constrained elements

The simplification made it possible to consider not only motions in the horizontal direction, but also the resulting rotational velocities and a force in the vertical direction. Although direct formulation methods were employed, constraint forces did not need to be considered explicitly.

Now, we can use the same model to analyze a situation such as shown in the right-hand side of Figure 1. Here the bicycle is shown tipping over after crashing into a wall. At the instant of crashing the boundary conditions on the bicycle change. The front wheel now makes contact at two points and the rear-wheel is not constrained to remain on the ground shown in Figure 15. If the rider no longer applies a force on the pedal, the model can be reduced to two separated bond graphs such as shown in Figure 16. These can be further simplified to two equivalent inertias. The equivalent inertia in this case depends on the rotational inertia of the bicycle frame which can no longer be deleted.

Implementation Details

We have implemented such a conceptual design environment as an extension to a commercial CAD system [UN1CAD 87]. MEDA (Mechanical Engineering Design Assistant) provides a graphical interface where the designer can describe an electro-mechanical device by locating components and specifying kinematic connections between them. The user can choose from a predetermined set of components whose bond graph models have been defined by the system developer. The user specifies a complete set of component form parameters and then uses the built-in form-function relationships to deduce the rest of the pnmjdve specifications. If desired, the designer, can override the default values specified by the form-function relationships.

The components can be connected at selected spatial locations by using one of the several kinematic connections provided in the system and the internal dynamic model is updated as the design is modified. This aggregation and manipulation of ihe internal bond-graph model is transparent to the user. Once all the desired connections are specified MEDA is invoked to formulate and simplify a bond graph and to formulate and solve the equaoons of motion if desired.

$$I_{eq} \;\longleftarrow\!\!|\; S_e : F$$

$$I_{eq} = \frac{1}{r_c^2}\left\{ I_{\theta C} + n^2\left[I_{\theta RW} + r_{RW}^2\left(I_{xRW} + I_{xF} + I_{xC} + I_{xFW} + \frac{1}{r_{FW}^2} I_{\theta FW}\right)\right]\right\}$$

with	I_{eq}	≡	Equivalent inertia
	r_C	≡	Pedal/crank radius
	$I_{\theta C}$	≡	Crank inertia
	n	≡	Chain sprocket ratio
	$I_{\theta RW}$	≡	Rear wheel inertia
	r_{RW}	≡	Rear wheel radius
	I_{xRW}	≡	Rear wheel mass
	I_{xF}	≡	Frame mass
	I_{xC}	≡	Crank mass
	I_{xFW}	≡	Front wheel mass
	r_{FW}	≡	Front wheel radius
	$I_{\theta FW}$	≡	Front wheel inertia

Figure 14: Final simplified Bond Graph Model of Bicycle

Related Work

Methods to automatically formulate equations of motion of a dynamic system have been discussed in several papers [Haug 89, McInnis 89, Paul 70, Sheth 72, Orlandea 77]. The Dynamic Analysis and Design System, DADS, [Haug 89] represents mechanical devices internally by a set of constrained second order differential equations. In DADS the two wheels, the frame and the sprocket of Figure 1 each will have have three co-ordinates. There will be eleven constraint equations: two each for the three hinge joints, between the frame and the wheels and the sprocket, two for each wheel restraining the velocity of the point of contact with the ground and finally one between the sprocket and the rear wheel. Hence the bicycle problem will be described by a twenty-three by twenty-three set of differential-algebraic equations using Lagrange multipliers. This is a convenient enough model for simulation but the size and complexity of the model make it impossible for the designer to visualize the design trade-offs. McInnis and Elmaraghy in [McInnis 89] propose another systematic method to automatically model dynamic systems. Their implementation SYSBOND also creates internal bond-graph models of dynamic systems and uses it for analysis. Like MEDA, SYSBOND also does not require the user to be familiar with bond graphs. However SYSBOND does not assemble the system bond-graph from sub-system models but follows the direct formulation procedure in [Rosenberg 83]. The user identifies the independent degrees of freedom and enumerates the kinematic co-ordinates (distinct velocities) of the system. While this could be a hard task for the user, it leads directly to simple models.

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. By this classification MEDA is primarily an analysis tool but the representational rigor of bond graphs allows presenting the results in a way that aids the designer in making decisions.

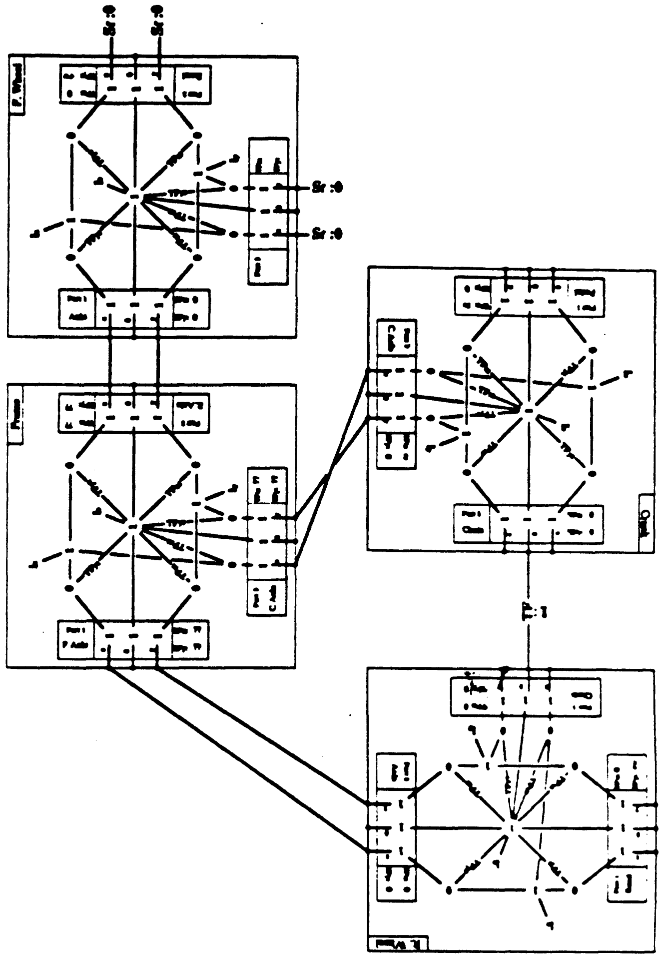


Figure 15: First-Cut model of Bicycle Tipping Over

[Bos 85] gives a bibliography of Bond Graph literature until 1985. Bond graph junction structures with causal or power loops and conditions under which they can be solved have been discussed in [Perelson 75, Rosenberg 79, Karnopp 75b, Ort 73]. These conditions would be used to identify pathological cases in computer programs [Enport 86] that solve bond graphs. Constraining junction-structures, such as discussed in the bicycle example, are causal loops with the added restriction that *all* external bonds impose effort (or flow as the case may be). The conditions for a junction structure to be constraining can be alternatively derived from the theorems in [Rosenberg 79]. However, in the context of the modeling methodology presented here, these structures are not pathological cases but represent modes which are irrelevant under the current set of boundary conditions. [Karnopp 75b] gives several examples of pathological junction structure power-loops that can arise when bond graphs are assembled from sub-system models.

Summary and Conclusion

We have presented here some issues and ideas underlying automated modeling in a design tool. In our design environment we define as primitives idealized physical components and kinematic connections in terms of bond graph fragments that describe their behavior. These bond graph fragments are not only accurate models of the physical objects but also satisfy the requirements of natural interface. The

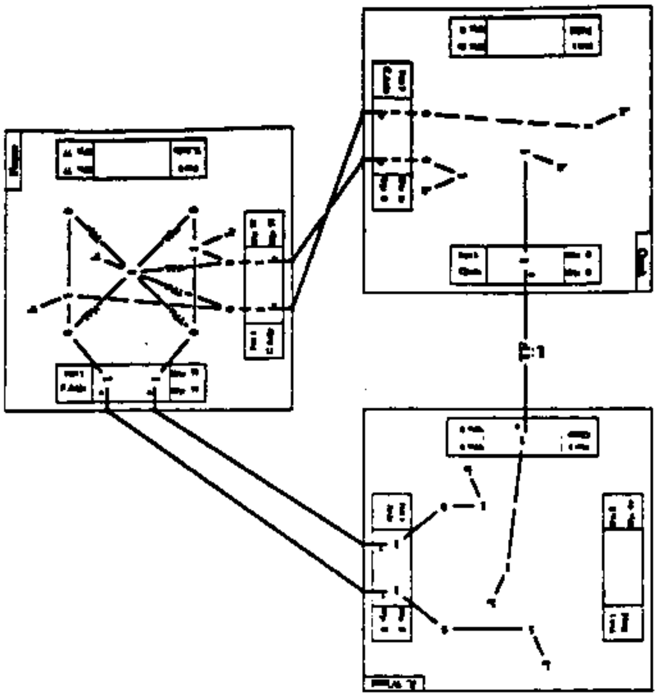


Figure 16: Simplified result of model

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. This model is then simplified to a much reduced problem . Simplifying the model facilitates drawing inferences about 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 implementation of MEDA and the latter for his many thoughtful comments and, in particular, pointing out that a junction structure can be thought of as a n-port with a complicated constitutive law. This lead to the algorithm for isolating inert junction structures.

References

[Bos 85]
Bos, A. M. and Breedveld, P. C. "1985 Update of the Bond Graph Bibliography," *Journal of the Franklin Institute.* Vol. 319, No. 1/2. Feb 1985, pp. 269-286.

[Colbum90]
Colbum, £. R. and Rinderle, J. R., "Design Relations." *Proceedings of the Second ASME Design Theory and Methodology Conference.* Sept 1990.

[Enport86]
*Tk0 Emport-A Urn***Unnunl Pn MVWW A. *»f*fflff far f -» f ^f M*" ^fan* 1986.

[Fuller 89]
Finger. S. and Rinderie. J. R.. "A Timsfonnational Appmch in M ^ n y y Design Using a Bond Graph Grammar," *Proceedings of the First ASME Design Theory and Methodology Conference.* Sept 1989.

[Haug89]
Hang, E J, *Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume 1: Basic Methods.* AUyn and Bacon. Boston, 1989.

[Hood 87]
Hood, S. J. and PaJroer 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.

[Kamopp75a]
Karnopp, D. and Rosenberg, R., *System Dynamics: A Unified Approach,* John Wiley *3L* Sons, New York, 1975.

[Kamopp75b]
Kamopp. D., "Some Bond Graph Identities Involving Junction Structures," *Journal of Dynamic Systems. Measurement and Control.* Dec 1975. pp. 439-441.

[Macfarlane89]
Macfarlane, J. R, *Qualitative and Symbolic Analysis of Dynamic Physical Systems.* PhD dissertation. University of Minnesota, 1989*

[Mclnnis89]
Mclnnis, J. B. and Elmaraghy, W. H.. "Automated Bond Graph Construction and Analysis for MSBD." *Proceedings of ASME Computers in Engineering.* July 1989, San Francisco

[Orlandea77]
Orlandea, N., Chace, M.A and Calahan. D.A, "A Sparsity-Onented Approach to the Dynamic Analysis and Design of Mechanical Systems." *Journal of Engineering for Industry.* Vol. 99,1977. pp. 773-784.

[Ort73]
On, J. R and Manens, H. R.. "The Properties of Bond Graph Juncoon Structure Matrices," *Journal of Dynamic Systems. Measurement and Control.* Dec 1973. pp. 362-367.

[Paul 70]
Paul. B. and Krajcinovic, D., "Computer Analysis of Machines »iih PU/w Motion," *Journal of Applied Mechanics.* Vol. 37.1970. pp. 697-712.

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

[Paz-Soldan88a]
Paz-Soldan, JP. and Rinderle. J. R., "Abstraction and Structure in ConcepuoJ Design Environments," Internal Lab Report CMU-MEDL-8S-21. Department of Mechanical Engineering, Carnegie-Mellon University. 1988.

[Paz-Soldan 88b]
Paz-Soldan, JJ>. and Rinderle, J*JR*.. "Conceptual Design Environment* Requirements and Representation," Internal Lab Report CMU-MEDL-88-24, Department of Mechanical Engineering. C*nvg< Mellon University, 1988.

rPerelson75]
Perelson, A. S.. "Bond Graph Junction Structures," *Journal* c/£)vwvu- S»t;rm. *Measurement, and Control.* June 1975, pp. 189-195.

rPrabhu89)
Prabhu, D. R. and Taylor. D. L.. "Synthesis of Systems from Specifications Containing Orientation and Positions Associated with Po*er Flo*." £)<-<** *Automation Conference.* 1989.

[Rinderle 87]
Rinderle, J. R., "Function and Form Relationships; A Basis for Prtbmiov> Design." *Proceedings from the NSF Workshop on the Design Process.* * 4Jdron. M.B..ed.Ohio State University.Oakland,CA,February 8-10 198? p? >V312.

[R0M0bcrg75]
Rosenberg, R. C. "Tlie Bond Graph at a Unified Data Bate for Engineering Syttem Design,* *Journal ofEnginedngfor Industry,* VoL. No., Nov 1975. pp. 1333-1337.

(Roaeoberg79]
Rosenberg, R. C. and Andry, A. N., "Solvability of Bond Graph Junction Structures with Loop*,"*lEEE Transactions on Circuits ami Systems, VoL* CAS-26, No. 2, Rb 1979, pp. 130*137.

[Rosenberg 83]
Kamopp.D. and Rosenberg. R^/attafeto *to Physical System Dynamics,* McGaw-HiH, New Yoric, 1983.

[Shcth72]
Sheth. P. N. and Uickcr .J. J^ Jr., TMP. A Computer Aided Design Analysis System for Mechanisms and Linkages," *Journal of Engineering for Industry* VoL 94,1972, pp. 454-464.

(Ulrica 89]
Ulrica, K. T. and Seering, W. P., "Synthesis of Schematic Descripdons in *MochMUCMlDesfaS Research in Engineering Design,* Vol.l. No.1,1989, pp. 01-16.

[UNICAD87]
MIPIE: ModipabklPortabUIEiiendabU, Reference Manual. Unicad, Inc., Boush *k.* Plume Streets, P.O. Box 1839, Norfolk, Virginia 23501.1987.