A COMPONENT-BASED MODELING APPROACH FOR SYSTEM DESIGN: THEORY AND IMPLEMENTATION

Jeffrey L. Stein and Loucas S. Louca
Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, MI 48109-2125
stein@engin.umich.edu and cyl@engin.umich.edu

ABSTRACT

Previous research has developed search algorithms for deducing Proper Models of dynamic systems. These minimum complexity models (with physically meaningful parameters) can reduce the design cycle where modeling and simulation is (or should be) a part of the design process. To apply these algorithms effectively to realistic systems, a Component Modeling Procedure consisting of a two-level representation is proposed. This procedure along with the algorithms are implemented in a computer program CAMBAS (Computer Aided Model Building Automation System). CAMBAS uses expandable bond graph models (templates) of components stored in libraries, which the design engineer selects to build a "word bond graph like" representation of the system. CAMBAS then automatically assembles the global bond graph of the system. This system bond graph is processed by the deduction search algorithms to generate the Proper Model. An illustrative example is provided to show the potential of CAMBAS for automating the production of Proper System Models for the design of multi-energy domain systems.

Keywords: Model reduction, Model acceptance, Model credibility, Model design, Model templates.

1. INTRODUCTION

Global competition is forcing a reduction in design cycle time. Thus, the costly and time consuming approach to design or redesign a system based on building and testing a prototype is not reasonable. Instead, design decisions must rely on insight about a system gained from computer simulation of mathematical models of the system to be designed. The purpose of this paper is to describe the development of a modeling procedure and its implementation in software whose purpose is to assist design engineers to rapidly obtain Proper Design Models of systems under development.

A model is an abstraction of the system and hence captures only certain aspects of its behavior. Heuristic procedures for generating this abstraction are just beginning to be discovered, and, thus, there is a lack of educational and industrial practice tools to help engineers develop modeling skills. As a result, engineers require a long time to develop good modeling skills, and typically these engineers are generally labeled as engineering analysts and are often separated from the primary design activities. Design engineers, who need to use models to help them with design decisions, are left with few tools to augment their limited modeling experience.

Recognizing this need, Wilson and Stein (1993), developed an automated modeling software package called Model Building Assistant (MBA) based on a model order deduction algorithm (MODA). MBA was developed as a "proof of concept" computer program and, therefore, has several limitations as a practical modeling tool. These include: applicability to only a small class of systems, a primitive and inefficient system description interface, and representation of only a small number of fixed component models. MBA is a good "proof of concept" tool but is not a practical modeling tool for design engineers. In addition, it was developed before the work of Ferris et al. (1994) and Ferris and Stein (1995) who introduced another deduction algorithm (Extended MODA) and the ability to handle modal component models. Therefore, it is the objective of this paper to describe (1) a modeling approach (language) to improve the ability of design engineers to communicate with a modeling tool and (2) to implement this language along with the MODA and Extended MODA in a software program intended to automate the production of proper models for realistic systems. The premise of this work is that design engineers would use proper models in the early stages of the design cycle to improve the productivity of the design process, if the proposed software existed to automatically generate proper models with minimum effort and modeling experience.

The paper first provides some background about modeling, proper models, and MODA in section 2 and, then in section 3, a component modeling approach is introduced. Section 4 describes the implementation of the component modeling approach in a software program (CAMBAS). An illustrative example to introduce the use of CAMBAS and clarify how it can be used by engineers to assist them in the design process is presented in section 5. Discussion and conclusion sections, sections 6 and 7, respectively, follow.

2. BACKGROUND

2.1. Physical System Modeling: Representations

Physical system modeling can be illustrated as shown in Figure 1. Modeling begins by engineers examining some real phenomena (system or proposed system), then based on some engineering decisions that account for system behavior (test data, system specifications), system structure and complexity, the important effects are included in the model. This model is usually referred to as an Ideal Physical Model (IPM) (Cannon, 1967). Many computer programs are available to convert an IPM into a mathematical representation that can be analyzed or numerically integrated (ADAMS 1992, DADS 1990, ENPORT 1992, CAMAS 1993, etc.).

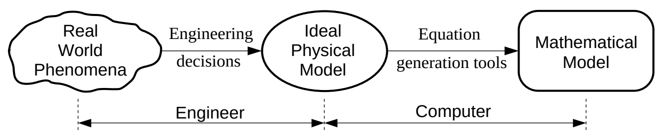


Figure 1: Physical System Modeling (Wilson and Stein 1992)

The IPM is a collection of interconnected ideal elements (e.g., masses, springs, dash-pots, for mechanical models or generalized inductors, capacitors, etc. for bond graph models), that represent the ideal dynamics of the system. Development of a good IPM, requires engineering intuition, judgment, and experience that is gained through extensive exposure to system modeling (Karnopp et al. 1990). This task becomes even more difficult when the system is complex, i.e., large number of ideal elements and more than one energy domain.

A representation for IPMs that provides a systematic and unified technique to represent the dynamic behavior of engineering systems is bond graphs (Rosenberg and Karnopp 1983; Karnopp et al. 1990). Bond graphs are energy based descriptions of systems where the ideal energy elements (generalized inductors, capacitors, and resistors) are connected together by energy conserving junction structure elements (0 and 1 junctions, transformers and gyrators). Bond graphs are easy to manipulate and provide the necessary information to create a set of ordinary differential equations representing the aggregate system behavior, but do not provide an explicit means by which to generate the proper IPM in bond graph form. This is particularly true for complex systems, when manipulation of the large bond graph becomes awkward and more difficult to understand for design engineers.

A less detailed, higher level representation is word bond graphs, where major subsystems are represented by words. In this case, multiport subsystems are established, with bonds interconnecting these subsystem representations (see Figure 2). This representation is more compact and provides key information for design engineers. However, there are no software tools that allow a designer to build a word bond graph and then automatically generate the underlying detailed bond graph needed for analysis and simulation. Ultimately, for the engineer to gain better insight into the possible behaviors a proposed system might exhibit, simulations are clearly required.

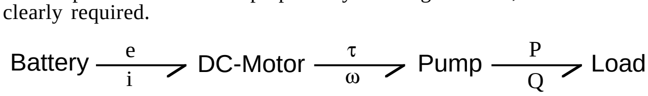


Figure 2: Word Bond Graph of a Servo-Hydraulic System

Because of the value of this compact word bond graph representation, one of the objectives of this work is to include this feature in the automated modeling environment. A "word bond graph like" representation will be used as a high level representation in the model development and then bond graphs will be used as a low level representation of the aggregate system behavior.

2.2. Automated Modeling: Proper Model Deduction

As a feasibility study to develop modeling tools for machine tool drive systems, Wilson and Stein (1992) developed a model order deduction algorithm (MODA). This iterative search algorithm deduces the proper complexity component models required to have the system model (a collection of components) predict all system eigenvalues within some user specified Frequency Range of Interest (FROI). They termed these system models Proper Models, because they have the minimum complexity required to meet the performance specifications and they have physically meaningful parameters. Proper Models are particularly useful for design because they contain the minimum information needed to show the relationship between the design parameters (physical dimensions and material properties) and the dominant system dynamics. Before explaining MODA the concepts of component rank and model boundedness-unboundedness are presented.

Wilson and Stein (1992) used the concept of rank to classify the complexity of the component model. Each component has a rank 0 model, which is the minimum complexity model as well as higher rank models that represent more complex models of the component. From a vibrational dynamics perspective, the rank 0 represents the rigid body model. The rigid body model plus the first and second modes of vibration correspond to the rank 2 model. Another concept used by Wilson and Stein (1992) is the concept of bounded and unbounded components, which corresponds to a finite or infinite maximum rank value, respectively.

MODA is an algorithm that specifies the complexity (rank) of each component model required for the system model to have a spectral radius of a certain value. That is all system natural frequencies will be within some specified FROI. MODA is an iterative search algorithm that generates the proper model, and the search starts with the rigid body model (all components have a rank 0 model). At each iteration the rank of a component is increased, the natural frequencies and spectral radius of the system are calculated, the spectral radius is recorded**,** and then the rank is set back to its default. This process is repeated for all expandable components (current rank less than maximum rank) in the system. An iteration ends by increasing the rank of the component that causes the minimum increase in the spectral radius. The algorithm continues until the spectral radius exceeds the FROI or there are not any more expandable components available.

Furthering this idea, Ferris et al. (1994) and Ferris and Stein (1995) developed Extended MODA, a proper model deduction algorithm containing an accuracy criterion. Extended MODA is a two-step procedure. First the critical system eigenvalues (CSE) are established using MODA as described above. Then the ranks of the component models are continued to be increased until the CSE converge within some specified tolerance. This is done sequentially by increasing the rank of the component whose current_rank +1 model causes the largest change in any of the CSE. This is continued until the largest change in the CSE movement is less than a user supplied tolerance. In addition to Extended MODA they also developed the use of modal models as part of the possible set of component models and formulated their models both with and without bond graphs. They argued that their techniques would be useful to design engineers. This work tries to further this claim by implementing their techniques in the software tool described in this paper.

Thus the objective of this work is to develop an automated modeling environment for design engineers that incorporates MODA and Extended MODA model deduction algorithms along with a "word bond graph like" interface for entering the system description.

3. COMPONENT MODELING APPROACH

In order to develop a more algorithmic approach to modeling, the modeling paradigm shown in Figure 1 is expanded as shown in Figure 3. The process of generating a mathematical model is now expanded to include the generation of a Proper Model and includes stages (groups of steps). The first stage, Decomposition, consists of the engineer isolating the system from the environment (selecting inputs and outputs) and identifying the components that comprise the system. The second stage, Synthesis, includes selecting the expandable models (templates) for each component. These models are then joined to form the system model (the IPM). Then, if the IPM is not in equation form, the IPM is converted into a set of state equations. Finally, the third stage, Model Deduction, consists of analyzing the model (e.g., determine the system eigenvalues) and using a model deduction search algorithm such as MODA to find a proper system model. Note stages two and three are completely algorithmic and, therefore, can be computerized.

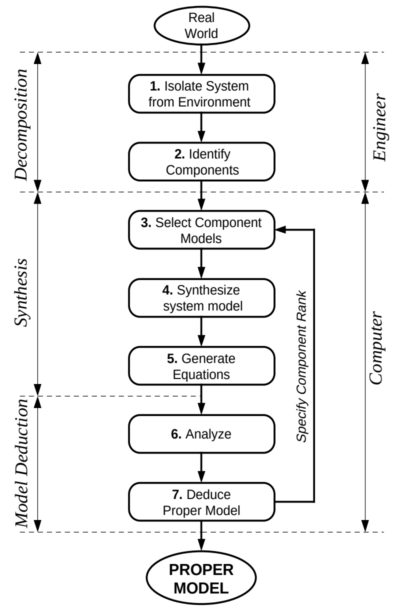


Figure 3: Component Modeling Procedure

3.1. The Synthesis Stage: A Two Level Representation

A two-level representation is proposed for this new component modeling procedure, in particular, steps 2 through 4 in Figure 3. These steps convert the real system isolated in step 1 to a system model as shown in Figure 4. Two levels are proposed for the outputs of steps 2 and 3. These include: The Component Level where each identified component of the system is given a label (name) and associated with a class of possible component model templates. These templates are in an expandable template form. That is, they can have different complexities from a rank of zero (0) up to the maximum rank allowed for that component. The next level is the Element Level, and is used for the output of the step 3, "select a component model." Here, a specific component model (i.e., a model of specific rank) is defined. This is done automatically by the model order deduction algorithm (see section 3.2). Finally, the output of the "system model synthesis" step is the system model of a specified system rank that is equal to the sum of the component model ranks.

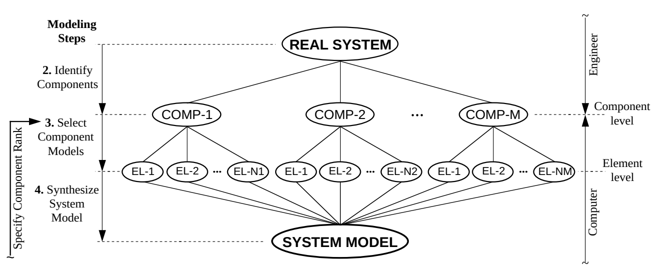


Figure 4: Two Level System Representation

This representation of the Component Level is similar to the word bond graph concept only the subsystems are components instead of words. At the Element Level, each component is represented by an IPM, which in this study is a bond graph (see Section 4.1.2.). Finally to generate the system model in bond graph form, the bond graphs of all components are interconnected based on the connections generated by the user during decomposition.

3.2. Model Deduction Stage: Algorithms

The modeling procedure shown in Figure 3 includes the model deduction stage to determine the proper complexity of the component models such that the system model meets some specification. Two examples of deduction algorithms discussed earlier, MODA and Extended MODA, use a frequency metric, and thus require the use of an eigenvalue solver step in the procedure labeled "Analyze." After synthesizing the system model, the spectral radius (MODA) or the change in the critical system eigenvalues (Extended MODA) determines if another iteration of adjusting the ranks of the component models is necessary. If not, the proper system model has been found.

4. IMPLEMENTATION OF THE COMPONENT MODELING APPROACH (CAMBAS)

To help evaluate the potential of the component modeling approach as an effective automated modeling tool for design, CAMBAS, a computer program has been implemented. This section describes CAMBAS, which was developed using the C programming language and the OSF/Motif® graphics commands. The Motif® commands perform all the graphic interfaces that take advantage of the RISC workstations capabilities (color, memory, and speed). The C programming language is used to implement bond graph processing, system synthesis, equation generation, and eigenvalue solver.

CAMBAS is an automated modeling environment that encapsulates steps 3 - 7 of the component modeling procedure shown in Figure 3. It provides a user-friendly graphical interface for building and selecting existing component building blocks to build a high level "word bond graph like" description of the system. The component description is processed through a model deduction algorithm to generate a proper model of the system in bond graph form. CAMBAS provides the engineer with utilities to define, move, edit, and delete any component. After all components are defined the connect-disconnect tool is used to define the connections between the different components of the system. The parameters (mass, stiffness, damping, diameter, length, modulus of elasticity, etc.) of each component are defined using the edit utility. Finally, to aid the design engineer in visualizing how the component model complexities (rank) are changed as the proper model is deduced, at any time the bond graph of any component can be displayed using the expand utility. The model configuration and parameters can be saved to a file for future use.

4.1. Component Model Selection

4.1.1. Component Level Representation

As described before, the system is decomposed by the design engineer into components. Each component of the system is a model template that consists of ideal elements and an expandable junction structure. Each component model has a finite number of power ports to communicate with the other components of the system and the environment and is represented as a block with its power ports shown as dots (see Figure 5). This is like a word bond graph where the multiport subsystem is represented as a block instead of a word. The block has a bond graph template description inside, and its ports are shown as dots. Depending on the energy domain these dots have different physical meaning and in all graphical displays are presented with different colors. For example, in the translational mechanical domain, a dot represents a rigid connection (joint); in the rotational mechanical domain, a rigid coupler; in the hydraulic, a pipe flange or fitting etc. The connections between the components in the system are noted as line segments, and they represent power flow. Using this block representation a graphical description of the system is generated, where each component of the system is represented as a block and their connections as lines.

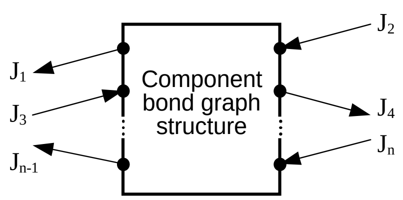


Figure 5: A Component Block Used for the Component Level Representation

Within CAMBAS the different components used to build the model are stored in a library. Each entry (component model) in the library consists of a bond graph description, element parameters, and text of the component description. (This is described in more detail in the next section.) This library can be altered by defining new components. The design engineer simply selects the suitable components from the library. For effortless component selection (search), the different components in the library are grouped according to their general characteristics (motors, pumps, speed reduction, etc.). Basic information about the dynamics and unique features of each component is also provided, to assist in the component selection.

4.1.2. Element Level Representation

The components used in the component level representation are either bounded or unbounded. The model for a bounded component is a bond graph that has a finite number of possible structure complexities (discrete component). The model for an unbounded component is a bond graph that has no limit on the maximum rank (continuous component). These component bond graphs are generated using a bond graph editor and the model is stored in the library for later use in building the system description (at the component level). Because the bond graphs on this level are not used to model an independent system, the nature (effort or flow) of the inputs of the system is not known, therefore, a new element is introduced to represent the joints of the components mentioned earlier. This is a dummy input/output element implying that other bond graph structures may be attached at this point. For example, a DC motor can be represented as shown in Figure 6. The electrical port provides the possibility to connect the motor to the electrical port of another component such as an amplifier (electric drive). The rotational port represents the mechanical power flowing from the motor and could be connected to another mechanical (rotation) component with a rotational port such as a flywheel.

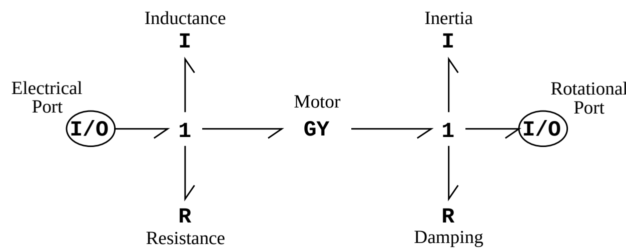


Figure 6: Element Level Representation (Bond Graph)

The complexity (rank) of the expandable template models representing components, is realized by assigning a descriptor to each element of the component. The descriptor has the property of specifying which elements can be removed or added (expandable elements) in order to change the rank of the component. For example in a rank 0 model all expandable elements are removed. To increase the rank by one, any of the expandable elements is added. When the expandable component model is developed using the bond graph editor all possible elements are included, i.e., the most complicated representation is specified. The deduction algorithm will later specify which elements, if any, have to be included (i.e., determine the rank of the component model to meet the system specifications).

For the unbounded (continuous) components, lump parameter (finite segment) or modal expansion approximations (Karnopp et al., 1990) can be used to define the model. In both cases to define the model, the geometry and material properties are required from which the bond graph model can be generated. The bond graph in both of these cases is a repetitive structure and can be automatically generated given the number of lumps or modes to be included in the model. Thus, for the unbounded components the rank issue is easier to incorporate than for the bounded components, due to the fact that the number of lumps (segments) or modes defines the rank of the model (see Ferris et al. 1994; Ferris and Stein 1995). The expandability of the unbounded components is realized by just changing the number of lumps or modes. This is done automatically by the deduction algorithm.

4.2. System Synthesis

After completing the configuration of the system (interconnecting the components) using the graphical interface, the system bond graph is generated based on a given rank for each component. (Typically the rank for each component starts at zero, and the deduction algorithm determines the rank at each additional iteration of the search to find the Proper Model.) When the system bond graph is synthesized, the dummy I/O elements and other redundant junction structure elements are eliminated, e.g., eliminate adjacent 0 or 1 junctions and then, if necessary, the equivalent I-C-R elements connected on the same 0 or 1 junction are determined. These simplifications make the system bond graph easier to analyze (reduces bond graph size and number of energy storage elements with derivative causality) during the subsequent steps of the procedure.

4.3. Equation Generation And Analysis

The state equations are generated from the bond graph generated by the system synthesis routine. The bond graph has a form that is compatible with a procedure outlined in Rosenberg and Karnopp (1983) for generating state equations. This procedure was selected for convenience. It is relatively simple to implement for linear bond graphs containing no implicit fields. Other procedures exist that could have been implemented (Broenink 1986 and Broenink et al. 1991).

The analysis routines needed by CAMBAS are determined by the deduction algorithms used. At this time MODA and Extended MODA are implemented, and they both require an eigenvalue solver. The eigenvalues of the state space matrix A are calculated using an eigenvalue solver obtained from numerical recipes in C (Press et al. 1994). This completes the process of generating and analyzing a model of a given complexity.

4.4. Proper Model Deduction

The model deduction tool provides the design engineer with various manual or automated model deduction options. These options allow the deduction of various types of models including: Rigid, Flexible, and Proper Models. The rigid model contains only the inertial and damping elements while the flexible model includes only compliance elements. These models are useful for determining power requirements and the static compliance of systems, respectively (Wilson and Stein 1992). These two options fall under the manual deduction option. The rigid body model is generated by first specifying the corresponding component ranks (zero) and then proceeding once through the steps of the component modeling procedure (see Figure 3) up to “generating the dynamic equations.”

When Proper Models are desired, the automated model deduction option is selected. The design engineer only needs to supply the FROI and eigenvalue accuracy, and then specify either MODA or Extended MODA to deduce the proper model. The program automatically sets the ranks of all the component models to zero and then systematically produces increments in the rank of each model until the FROI is exceeded (MODA) or the accuracy criterion on the critical system eigenvalues has been met (Extended MODA).

The system model generated by any of the model deduction options includes: the component ranks, the system bond graph, the state space vector and input vector, the A and B matrices, and the eigenvalues of A. These are generated at each iteration of the deduction algorithm and the ranks and eigenvalues are displayed and stored. Finally the proper system bond graph of the deduced model can be plotted.

5. EXAMPLE

An example is provided in this section to illustrate the features and use of CAMBAS. The example focuses more on the development of the model of a system rather than the use of the model to design the system. Therefore, many of the design specifications are not included in this example. Assume that the multi-energy domain system shown in Figure 8 is proposed to move a big load under precise control (controller not shown). The open loop system is powered by a DC motor which drives a hydraulic pump via a shaft. The pump, through an intake pipe, pressurizes the hydraulic fluid from the reservoir to the required high pressure. Then the high pressure fluid is directed to the hydraulic cylinder through a supply pipe, where the pressure is transformed to force by the piston of the cylinder. The generated force drives the load through a connecting rod. The fluid from the secondary chamber of the cylinder is directed to the reservoir through a return pipe. For safety purposes there is a secondary return pipe with a relief valve.

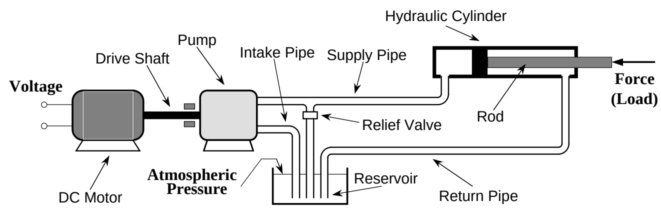


Figure 7: Servo Hydraulic System Schematic

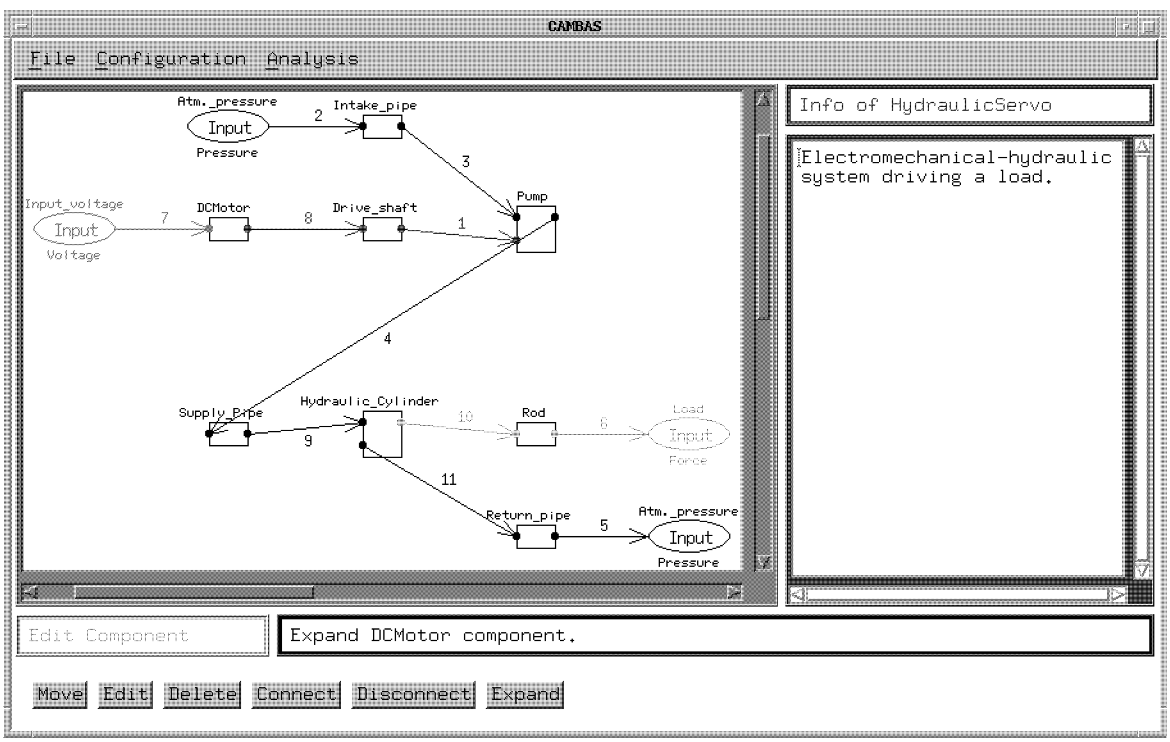


Figure 8: The Component Level Description of the “Hydraulic Servo” As Seen Through The CAMBAS Graphical Interface.

5.1. System Decomposition

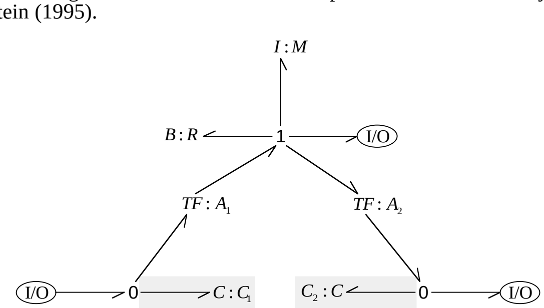
The component modeling procedure (Figure 3) requires two constitute decomposition steps that are not incorporated into CAMBAS. Consequently, the proposed system as shown in Figure 8 already delineates the system from the inputs as well as identifying the individual components that comprise the system. The inputs are the load (force) applied to the piston connecting rod, the applied motor voltage, and atmospheric pressure, which are the only ways that the system can communicate with the environment. The components of the system are identified as the intake pipe, pump, supply pipe, hydraulic cylinder, return pipe, DC motor, drive shaft, and connecting rod. The proposed design of these components is given by the parameters given in the Appendix. These values were taken from Kostopoulos (1992).

5.2. Component Model Selection

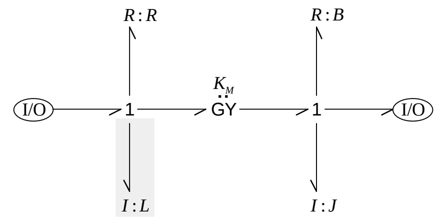
At this point the designers are ready to use CAMBAS to build their model. The component models are selected from the component library, imported, edited, and given a name. These are graphically connected to produce a high-level description of the system (see Figure 8). Note that the components with different labels do not necessarily have different component templates. For example, all of the pipes use the same template. Of course the parameters for each pipe are different and, thus the final rank of each pipe template used in the proper system model will be, in general, different. Also note, for this example, the maximum operating pressure is assumed to be lower than the set point of the relief valve; thus it will always be closed. Therefore, the relief valve component is not included in the system description. Finally, Figure 8 shows the component level description as it appears on the computer screen. Note the descriptive text in the window to the right of the model. This can be edited by the user. The text in the bar above this window is written by the program and simply tells the user the file name under which the model and text are stored. At the bottom of the screen are several buttons which provide editing capability (described in section 4) to the user. Of particular interest is the “Expand” button which allows the user to examine the bond graph model inside a selected component block.

The models for the motor, pump and cylinder are bounded templates. The underlying assumption is that there is a maximum complexity allowed for these models. The model templates taken from the library of the drive shaft, rod and pipes are unbounded templates. With unbounded templates the option of using a finite segment or modal representation is provided. The three pipes of the system are modeled using the finite segment approach and the shaft and rod using finite modes (Ferris and Stein 1995).

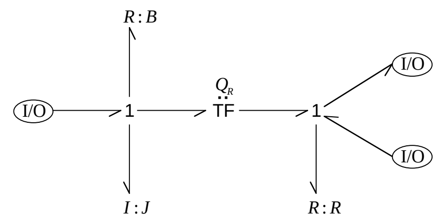
The component model templates and their possible rank values are shown in Figure 9. Note for the bounded components, the removable elements are shaded. The maximum complexity model (rank 2) for the hydraulic cylinder includes the compliance of each chamber. For the rank 0 model the two compliances are removed to create a rigid model. The maximum rank for the DC motor model is 1. Removing the winding inductance yields the rank zero model. The model of the pump is assumed to be fixed; that is, it only has a rank 0 model. This assumption is made here for convenience and should not be interpreted by the reader as appropriate for all situations. The rank of the pipes, rod, and shaft range from zero to infinity, and their general rank N models are also shown below. To change the rank of the pipe, a spring and mass pair are added or removed from the model (Ferris and Stein 1995). For the shaft model the rank is increased or decreased by adding or removing the elements that represent the next mode (modal mass I, modal stiffness C, and corresponding junction structure elements), respectively. The reasons for using finite segment or modal models is explored in more detail by Ferris and Stein (1995).



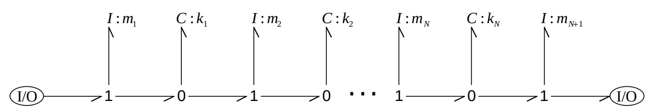
Rank 0-2 Hydraulic cylinder (shaded elements are removable)



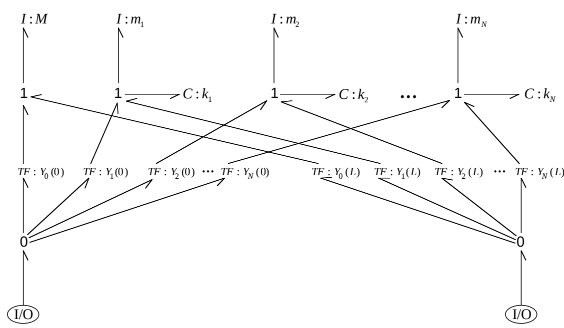
Rank 0-1 DC motor



Rank 0 Pump



Rank N Pipe (Finite Segment)



Rank N Shaft or Rod (Modal Expansion)

Figure 9: Component Template Models

5.3. Proper Model Deduction

At this stage, CAMBAS internally generates the system model, state equations, and eigenvalues. Because these outputs occupy a lot of space, only a few key results will be shown. Initially, the engineers might want to examine power requirements, reflected inertia, and required torque of their design. This can be studied using the rigid body model (all rank 0 components). Once the rigid body option is selected, the rank zero system bond graph is synthesized and the state space equations are generated. This bond graph is shown in Figure 10 as would be seen by the user. In this model all mechanical compliances are removed and the resulting model consists only of inertia and viscous energy losses. Note the bond graph is automatically simplified to create equivalent elements. These equivalent elements allow the designer , for example, to determine the reflected inertia at the motor or to determine the total losses in the system. Simplification of the graph also generates a graph with integral causality, which simplifies the task of generating the state equations and thus determining the dominant time of the system (the eigenvalue of A).

To meet other design objectives, such as the open loop bandwidth specification, a model of greater bandwidth is required. The question is, which components must be increased in rank to improve the model’s high frequency fidelity. This question can be answered using CAMBAS by deducing the proper model with MODA (and/or Extended MODA). First a FROI of 350 rad/s and an eigenvalue tolerance of 0.5 % are specified by the designer. Then the proper model is deduced by running Extended MODA. Extended MODA first uses MODA to determine the CSE and then increases the rank of the component models so as to cause the critical system eigenvalues to converge.

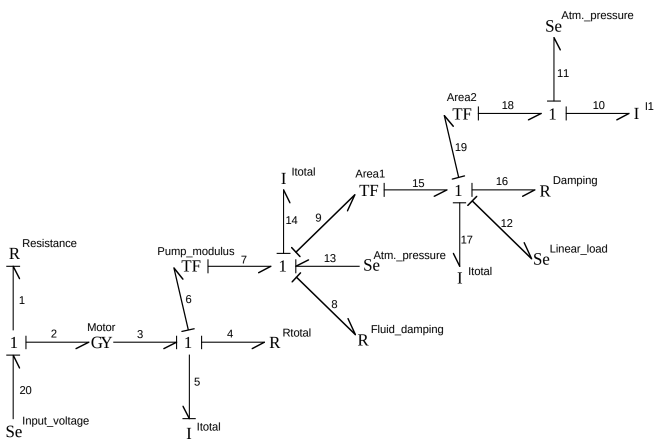


Figure 10: Bond Graph of the Rigid Body Model

The results of this deduction process are shown in Figure 11. Five (5) iterations of MODA are required to define the CSE [i.e., there are five complex conjugate pairs of eigenvalues with five associated natural frequencies plus the time constant (inverse of the real eigenvalue) associated with the rigid body mode]. These CSE are associated with the components as follows: The hydraulic cylinder model has a rank of 1. The supply pipe has rank 1 and return pipe rank 3. All other components still have a rank 0. Eighteen (18) additional iterations are required by the Extended MODA algorithm to cause the CSE to converge within the stated tolerance. During this process an additional eigenvalue of the CSE (the one associated with the rank 3 model of the supply line) moves outside the FROI. The resulting CSE are shown in Table 1.

The results in Figure 11 also show that the return pipe is the weakest dynamic link. From a modeling perspective this may mean that a modal representation of the pipe would be useful to reduce the amount of complexity of this component model required to meet the specifications (see Ferris and Stein 1995). From a design perspective, it appears that the high frequency behavior of this system is limited by the return pipe dynamics. That is, the dominant eigenvalue pair is generated due to the behavior of the return pipe. It could be argued, therefore, that in order to improve the frequency response of the system the dominant eigenvalue pair should be pushed away from the origin. The return pipe should be redesigned to improve its frequency response. For example, make it shorter and smaller in diameter.

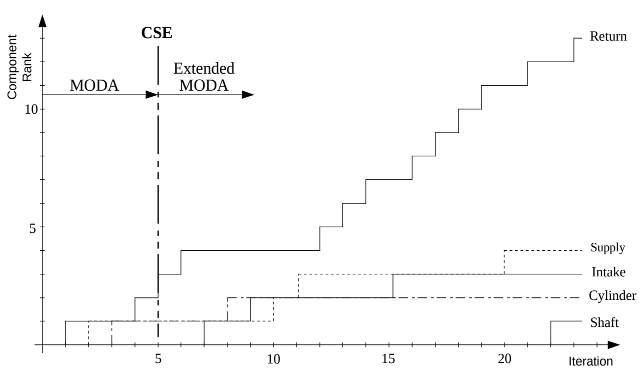


Figure 11: Changes In Component Ranks During The Model Deduction Process.

Table 1: Proper Model Critical System Eigenvalues Expressed as the Rigid Body Time Constant, $\tau = 1/\sigma$ and Natural Frequencies of the System Modes.

	Rigid Body, σ	ω_{n1}	ω_{n2}	ω_{n3}	ω_{n4}
Extended MODA	0.8913	52.387	164.70	210.01	362.55

6. DISCUSSION

6.1. Automated Modeling Tools For Designers

The aim of this research is to develop automated modeling tools to reduce design cycle and to provide better information to the design engineer about how a proposed system may behave and thus how it can be redesigned to improve performance. This section will highlight the implications of the contributions of this paper to the development of automated modeling tools for design.

6.2. Component Modeling Procedure And Model Deduction

The Component Modeling Procedure combined with the model deduction algorithms represents one of the first truly automated model generating ideas. While the idea of component libraries, where the user simply selects those components that are in a system under study, is not new, the fact that in this paper these component models are an expandable template whose complexity is determined automatically in the context of a given model objective, is new.

The potential for this Component Modeling Procedure to lead to the development of better computer aided design tools is great. As discussed in the introduction, design engineers working at the systems level cannot be expected to be experts at choosing the proper model for their design work. The component modeling procedure, allows them easy access to formulating the model by simply requiring them to conceptually divide the (proposed) system into components. In addition, they must be able to find parameters for those components and to specify the frequency range over which they expect the system to operate.

Two of these three requirements, decomposing the system into components and specifying the frequency range, should be relatively easy for the designer to accomplish. The challenge is to obtain the model parameters. However, information on techniques for finding component model parameters could be stored along with the component models. In some cases, such as the modal mass and stiffness matrices for standard objects like a rod, the information required is quite simple (length, diameter, material properties) and the modal matrices can be generated automatically (CAMBAS, in fact, does this). For other components FEM codes or modal testing data might be required (Ferris et al. 1994; Ferris and Stein 1995). For other components, such as for DC motors, most of the parameters are available from the manufacturer. The really difficult parameters to find are friction coefficients. For a proposed system, these can usually only be guessed. Heuristics that experts use to select (guess) these parameters could be stored along with the component models.

6.3. Two Level Representation

The two-level representation (components and elements) gives the designer the flexibility to operate at the higher representation level for ease of model development but still allows access to the “nuts and bolts” of each component model. New component models can be developed or existing ones modified. In addition, output features to help the designer interpret the results can be implemented at both the component and element levels. For example, power flow analysis (Rosenberg and Zhou 1988) could be applied at the element level in the system bond graph. This might help the designer to further modify their designs based on the element's contribution to power flow in the system. In this paper, an output display at the component level (Figure 12) that shows the rank of each component required Proper System Model, helps the designer immediately establish the weak component (dynamic link) in the design.

6.4. CAMBAS

CAMBAS is intended to be a prototype automated modeling tool for design. While it is difficult, if not impossible, to provide the reader with any real sense of how easy to use and powerful any computer tool is, ultimately the purpose of the program is to demonstrate the feasibility and extensive need for such a tool. In this context, the authors wish to highlight the strengths and weakness of the program from a functional (design engineers) perspective.

Many advantages of CAMBAS to the designer come from the implementation of the Component Modeling Approach, the model deduction algorithms, and the two-level representation discussed above. However, CAMBAS greatly facilitates use of these concepts by providing the graphical, menu driven interface to the user. Components can literally be moved in and out of the description by a few mouse driven commands. Details of any component model can be readily examined by highlighting the component of interest and executing the expand function. Finally the designer can literally generate hundreds of models by choosing different modeling metrics (e.g., different frequency ranges including rigid body models).

Some of the limitations of CAMBAS are as follows. One is that it currently does not produce some of the output a design engineer would want in a user friendly format. For example, the results shown in Figure 11, should automatically be produced by the program. Second, the current structural components (rod, shaft) included in the library can have only one dimension. It cannot handle a two dimensional beam (e.g., lateral translation and bending) at this time. In addition, because the model deduction algorithms that are available (MODA and Extended MODA) operate on linear or linearized systems CAMBAS cannot handle nonlinear systems. Linearized nonlinear systems could be handled, but this feature remains as future work. Finally, the current modeling procedure uses a two-level representation, but for more complicated systems (e.g., multi-dimensional systems), a multi-level representation may be required to generate compact higher level representations. The modular nature of CAMBAS should permit future extensions to be easily implemented and thus produce a more general automated modeling environment that is a more powerful tool for designers.

7. SUMMARY AND CONCLUSIONS

A two-level-based Component Modeling Procedure is developed to exploit the power of several existing model order deduction algorithms. This procedure allows the user to describe a system as interconnected components similar in concept to a word bond graph. Each component is represented at two levels, component and element. At the component level the component is represented as a simple block with ports to allow interaction with other component blocks. At the element level the component is represented by a bond graph model template. The complexity of the template is variable and is denoted by its rank. When combined with a model deduction algorithm the rank is set by the algorithm and then is incremented during a search for the proper model of the system where the proper model is defined as the simplest model of the system that meets the system specifications. For a proper model, the minimum rank required of each component, in the context of the system frequency and critical system eigenvalue accuracy requirements, is determined.

The two-level modeling procedure and model deduction algorithms are implemented in a software program, CAMBAS. This graphical interface, menu driven software program successfully demonstrates the two-level component modeling approach. It also highlights how it can be used by design engineers to quickly make assessments of a proposed design by examining the weak dynamics components in the design. It is argued that this type of automated modeling tool, which focuses the designer on configurational issues rather than details of model implementation, will shorten the design cycle for products where the dynamic performance is important.

8. ACKNOWLEDGMENT

The authors are grateful for the input on generating equations from bond graphs provided by Professor Ronald C. Rosenberg, Michigan State University. The authors also gratefully acknowledge Fulbright (Cyprus-America Scholarship Program) scholarship awarded to Mr. Loucas S. Louca for his graduate studies at The University of Michigan.

9. REFERENCES

ADAMS, 1992. *ADAMS Product Line - Release 6.0.* Mechanical Dynamics, Inc., Ann Arbor, MI.

Broenink, J.F., 1986. *"SIDOPS, a bond graph modeling language, Complex and distributed systems: analysis and control"*. IMACS trans. scientific computation 4, Tzafestas S., and Borne P., editors. North Holland, Amsterdam, Netherlands, pp. 81-6.

Broenink, J.F., J.W. Bekkink, and P.C. Breedveld, 1991. *"Multibond-graph version of the CAMAS modeling and simulation environment"*. Bond Graphs for Engineers, IMACS 13th World Congress.

CAMAS, 1993. *Reference Manual, Preliminary version.* University of Twente, The Netherlands.

Cannon, R.H., 1967. *Dynamics of Physical Systems.* McGraw-Hill, New York, NY.

DADS, 1990. *Reference Manual*.. University of Iowa, Iowa City, IA.

ENPORT, 1992. *ENPORT/PC Reference Manual*. Rosencode Associates, Lansing, MI.

Ferris, J.B., J.L. Stein, and M.M. Bernitsas, 1994. *"Development of Proper Models of Hybrid Systems"*. Symposium on Automated Modeling, 1994 ASME Winter Annual Meeting.

Ferris, J.B., and J.L. Stein, 1995. *"Development of Proper Models of Hybrid Systems: A Bond Graph Formulation"*. Proceedings of the 1995 International Conference on Bond Graph Modeling, Las Vegas, NV.

Karnopp, D.C., D.L. Margolis, and R.C. Rosenberg, 1990. *System Dynamics: A Unified Approach.* Wiley-Interscience, New York, NY.

Kostopoulos, T.N., 1992. *Hydraulic and Pneumatic Systems.* Symeon Publishing Co, Athens, GREECE.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Slannery, 1992. *Numerical Recipes In C* . Cambridge University Press, New York, NY.

Rosenberg, R.C., and D.C. Karnopp, 1983. *Introduction to Physical System Dynamics.* McGraw-Hill, New York, NY.

Rosenberg, R.C. and T. Zhou, 1988 *"Power-Based Model Insight"*. Proceedings of the 1988 ASME Winter Annual Meeting, Symposium on Automated Modeling for Design, ASME Book No. G00460, ASME New York, NY.

Wilson, B.H. and J.L. Stein, 1992. *"An Algorithm for Obtaining Minimum-order Lumped-Parameter Models of Distributed and Discrete Systems"* 1992 ASME WAM Symposium on Automated Modeling, November, Anaheim, CA, Published by ASME Book No. G00747, New York, NY. (also in review for *Journal of Dynamic Systems Measurement and Control*)

Wilson, B.H., and J.L. Stein, 1993. *"Model-Building Assistant: An Automated Modeling Tool for Machine Tool Drive Systems"*. International Conference on Bond Graph Modeling.

APPENDIX

DC motor

Inductance = $2\text{x}10^{-3}$ Henry
Resistance = 10 Ohm
Motor Constant = 0.5 N·m/Amp
Rotor Inertia = $2\text{x}10^{-2}$ Kg·m^2
Damping = $1\text{x}10^{-4}$ N·m·s/rad

Pump

Flow Rate = $1\text{x}10^{-5}$ m^3/rad
Hydraulic Loss = $1\text{x}10^{-4}$ N·s/m^5
Inertia = $1\text{x}10^{-4}$ kg·m^2
Damping = $1\text{x}10^{-5}$ N·m·s/rad

Hydraulic Cylinder

Area$_1$ = $8.85\text{x}10^{-3}$ m^2
Area$_2$ = $5.89\text{x}10^{-3}$ m^2
Compliance$_1$ = $2.066\text{x}10^{-12}$ m^5/N
Compliance$_2$ = $2.325\text{x}10^{-12}$ m^5/N
Piston Mass = 3 Kg
Piston Damping = 0.01 N·s/m

Shaft

Diameter = 20 mm
Length = 0.5 m
Density (Steel) = 7755 Kg/m^3
Shear Modulus(Steel) =$9.31\text{x}10^{10}$ N/m^2

Intake Pipe

Diameter = 32 mm
Length = 4.8 m

Return Pipe

Diameter = 20 mm
Length = 13.5 m

Density (Hydraulic fluid) = 900 Kg/m^3
Bulk Modulus (Hydraulic fluid) = $1.52\text{x}10^9$ N/m^2

Rod

Diameter = 50 mm
Length = 1 m
Density (Steel) = 7755 Kg/m^3
Modulus of Elasticity (Steel) = $2.1\text{x}10^{10}$ N/m^2

Supply Pipe

Diameter = 20 mm
Length = 22.2 m