The Automated Symbolic Derivation of State Equations For Dynamic Systems*

Jane Macfarlane
Lawrence Livermore Laboratory
P. O. Box 808 L-307
Livermore, California 94550

Max Donath
University of Minnesota
Mechanical Engineering Department and the Productivity Center
111 Church St. SE
Minneapolis, Minnesota 55455

Abstract

A system which is capable of generating symbolic state equations for dynamic systems is described. The underlying object-oriented knowledge representation structure is based on the bond graph modelling methodology which abstracts a complex system into a network of simple relations that describe the behavior of the system. Constraint propagation techniques provide a method for determining the causal relationships that must exist between system variables in a state determined system, thereby identifying a set of state variables for the system. The relation network and the state variables are manipulated by MACSYMA to define the state equations for the physical system dynamics. The availability of state equations in symbolic form allows the engineer to assess the influence of component parameters on the overall function of the physical system without having to resort to simulation iterations. This representation establishes the foundation for a model based reasoning system currently under development.

1. Introduction

Expert problem-solving systems have been successful at solving very specific problems, but have failed at expanding into problems that are similar in focus but slightly different in function. The difficulty often arises because the knowledge base is semantically derived from the interviews with an expert. If, however, the basic principles of the domain are the foundation for the knowledge and inferencing, then the system ought to be broad enough to reason about a variety of domain problems. This requires that the organization of the basic principles of a domain fit into a well defined and cohesive knowledge representation structure.

The system described in this paper utilizes a modelling methodology which provides a structured approach to system dynamics modelling. This methodology, called bond graphs, is used to abstract physical systems into basic elements that represent localized dynamical properties of a small part of the system. Models of complex systems are built as a collection of the basic elements and a network of relations that are derived from the representation. These, in turn, define a mathematical description of the dynamics for the physical system.

State equations are derived from two pieces of information about the model:

- (1) the network of basic relations that model the physical system, and
- (2) the definition of the independent state variables in the system.

The network of relations is generated from an underlying object representation which allows the user to choose basic elements

and graphically build the bond graph representation for the system. Each element object inherits a relation that defines its' dynamic property and an associated parameter symbol. Consequently, the parameter is left arbitrary and the user is not required to specify any particular parameter values.

The state variable selection is determined by examining the causal relationships that must exist between the variables in order for the system to be a state determined system. Constraint propagation techniques are used for assigning the causal relationships for the relation network. Strict causal relationships are defined first and propagated through the network in an effort to determine dependent variables. Elements that have not yet been assigned a causality are then assigned a preferred causality and the effects are further propagated through the network of relations. Elements that exhibit a particular type of causal relation, namely an integral relation between its variables, are independent energy storage elements. The state variables are selected by identifying the independent energy storage elements for the system and picking the independent variable from each such element.

The network of relations and the state variables are down loaded into MACSYMA for symbolic processing. MACSYMA combines the relations and eliminates secondary variables in order to produce the system state equations. Since the state equations are generated in symbolic form, the user gains important insight into how component parameters effect the overall function of the device. If detailed performance results are still desired, then simulation capabilities can be directly implemented.

2. Bond Graphs

The system described in this paper is based on a modelling methodology used to characterize physical system dynamics, called bond graphs. A bond graph is an abstract representation of a system whereby a collection of components interact with each other through energy ports. Energy ports are places in the system where energy is exchanged. As such, this representation essentially describes how power flows through the system.

Bond graphs resemble schematic diagrams; however, they go beyond the schematic in that they represent the explicit behavior of a system which can, if desired, be transformed into a mathematical model that describes the system. A bond graph element may either represent a simple component in the system or it may represent a physical effect which must be included in the representational model. A major benefit of using bond graph techniques is that, besides being a compact representation, they may be used to model energy transformation across many boundaries including mechanical, electrical, hydraulic, pneumatic, chemical, magnetic and thermal dynamics. In addition, they provide uniform representations which encompass

*This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

actuators that are often not well handled by other lumped parameter modelling systems. For example, a motor can be readily modelled by a bond graph gyrator element and a number of other elements depending on the level of detail desired. Although a well-defined method for generating the mathematical model from the bond-graph representation exists, reasoning about the system dynamics can be directly inferred from the bond graph.

Bond graph modelling methods are based on three fundamental concepts.

1. Energy in the physical system is modelled by considering two intrinsic variables called effort and flow.
2. A physical system is modelled as a collection of simple elements that exchange energy through ports.
3. A causal relation exists between each pair of effort and flow variables.

Detailed descriptions of bond graph methods and model development may be found in [1,2,3]. This paper will only highlight the basic ideas.

2.1 Power Variables

Energy transfer at each energy exchange port in the system is characterized by effort and flow variables. Power transfer is associated with the port since power is the product of the effort and flow values. Effort and flow variables for the various types of physical systems are shown in Table 1.

2.2 Elements

Bond graphs are made up of nine different types of abstract elements which are classified by the number of energy exchange ports they have. One port elements consist of the capacitive element, the inductive element, the resistive element, the effort source, and the flow source. Two port elements are made up of transformers and gyrators. Multiport elements are parallel (p)** or series (s) junctions that represent energy distribution points. Table 2 lists the elements by category and the associated notation.

**The s- and p- junction terminology is used in Europe. In North America, the 0- and 1- junction terminology is traditionally used. For semantic reasons, we prefer the s- and p- junctions.

Each element represents a basic relation that exists between its associated effort and flow variables. Table 3 describes the relations.

Elements are connected together via bonds which are lines that indicate an ideal power flow from element to element. All losses are represented by resistive elements. A half arrow bond indicates the direction of the power flow. Figure 1 shows a representative bond graph for a shaft exhibiting compliance along its length, losses at the support bearings and carrying an inertial load with input and output torques and angular velocities shown.

Using the symbols of the bond graph language, an abstract model of the physical system may be created. Since elements are modular in nature, primary, secondary, and tertiary effects (both linear and nonlinear) can be included depending on the level of detail desired. The symbol graph with its inherent relations comprises all the required information for describing the system.

2.3 Causality

In order to characterize the system dynamics, it is necessary to determine the state variables. Bond graph modelling methods provide a well defined process for selecting the independent energy storage elements, the basis of causality assignment. This causality that we speak of is essentially a computational causality, e.g. is flow computed based on effort or vice versa. However, this computational "flow" provides us not only with the describing equations but also with a great deal of insight into the system's dynamics.

The assignment procedure is based on an analysis of the power variables associated with each bond in the graph. Since a bond represents an ideal power flow, the power must be equivalent at each end of the bond. Therefore, the effort and flow variables are inter-dependent. Once one of the power variables is defined, the other power variable is set by the relationships associated with the adjacent bond graph elements.

The causality assignment procedure defines which power variable is imposed onto each element in the system. In Figure 2a, element A independently imposes an effort onto element B; consequently B imposes a flow onto A. The notation

Table 1. Power Variables for Various System Types

System Type	effort	flow
Mechanical Translation	force	velocity
Mechanical Rotation	torque	angular velocity
Electrical	voltage	current
Hydraulic/Pneumatic	pressure	flow
Thermal	absolute temperature	heat flux

Table 2. Element Categories and Notation

Category	Element Name	Type	Notation
one ports	effort source	energy source or sink	S_e
	flow source	energy source or sink	S_f
	capacitor	energy storage	C
	inductor	energy storage	I
	resistor	energy dissipator	R
two ports	transformer	energy transformation	TF
	gyrator	energy transformation	GY
multiports	parallel junction	energy distribution	p
	series junction	energy distribution	s

Table 3. Element Relations

Element	Relation
capacitor	$\text{effort} = \int \frac{1}{c} * (\text{flow}) dt$ where c is the capacitive characteristic
inductor	$\text{flow} = \int \frac{1}{i} * (\text{effort}) dt$ where i is the inductive characteristic
resistor	$\text{effort} = R * \text{flow}$ where R is the resistive characteristic
transformer	$\text{effort}_{in} = m * \text{effort}_{out}$ $m * \text{flow}_{in} = \text{flow}_{out}$ where m is the transformer modulus
gyrator	$\text{effort}_{in} = r * \text{flow}_{out}$ $r * \text{flow}_{in} = \text{effort}_{out}$ where r is the gyrator modulus
parallel junction	$\sum \text{flows} = 0$ $\text{effort}_i = \text{effort}_j$
series junction	$\sum \text{efforts} = 0$ $\text{flow}_i = \text{flow}_j$

Figure 1a. Mechanical Schematic

Figure 1b. Equivalent Bond Graph (Explicit Form)

(a)

If A imposes an effort into B the causal stroke is positioned next to the B element. This further implies that a flow is imposed on A by B.

(b)

If A imposes a flow into B the causal stroke is positioned next to the A element. This further implies that an effort is imposed on A by B.

Figure 2. Causal Stroke Assignment with Respect to Element A

for assigning causality uses a perpendicular line positioned at either end of the bond called the *causal stroke* (see Figure 2). It is important to note that the power flow direction associated with each bond is independent of the causality and is consequently not included in the figure.

Some elements by definition have strict assignment rules. Table 4 lists possible causality assignments for each element

type in order of restrictions from most restricted to least restrictive. Effort and flow sources must, by definition, exhibit the identified causality. The elements labeled with restricted causality are those which may only be assigned in the manner listed in the table.

Integral causality implies an element exhibits an integral relation between its effort and flow variables. In the case of the C

Table 4. Causality Assignments

required causality	effort source	flow source
restricted causality	transformer	transformer

gyrator	gyrator

parallel	parallel

series	series
preferred integral causality	integral causality	derivative causality
	inductor	inductor
arbitrary causality	capacitor	capacitor
	resistor	resistor

element, this means that the effort value may be derived as the integral of the flow value. If the causality were assigned as derivative causality (the causal stroke at the other end of the bond), it would mean that the flow value is derived as the derivative of the effort value. This assignment would violate the definition of a state variable in a state determined system. If derivative causality must be assigned, it means that the particular variables are dependent on another energy storage element in the model and as such, there would be no state variable associated with this element. Finally, the resistive elements only dissipate energy and do not require any particular causality assignment.

2.4 Existing Systems for Bond Graph Modelling

There are several software systems available for bond graph analysis based on numeric procedures. ENPORT [4] and TUTSIM [5] provide simulation capabilities given a bond graph model and its associated parameter values. The simulation packages are integrated with the bond graph interface package.

CAMP is a system that converts a bond graph description into a FORTRAN module that may be input to a simulation language such as ACSL or CSMP. CAMP also provides a menu interface that translates domain specific words into the associated bond graph element and uses a simple set of modelling rules for leading a user through the model building process [6]. This is a needed first step towards providing a transparent front end which will relieve the user from the necessary details of bond graph modelling techniques.

As a consequence of being simulation based, ENPORT, TUTSIM and CAMP must require the user to specify parameter values for the bond graph elements. This forces the user to either guess the parameter values or have an in-depth knowledge of the system being modelled. However, in many cases the engineer is presented with vaguely defined problems and is

asked to determine the parameters in order to guarantee a specified function of the physical system. Consequently, by using a simulation based approach the engineer is viewing only a small portion of the behavior space for the system.

A new system [7], currently under development, represents the bond graph model using the PROLOG language. This system allows the user to input PROLOG commands customized to specific domains and provides assistance in determining model causality failures. It takes a more general approach by maintaining the bond graph representation and the associated parameter set in symbolic form. However, the system requires the user to manually manipulate the underlying data structures in order to build the state equations.

Our paper describes a system which allows a user to graphically build a bond graph model from domain specific elements. The entire representation is in symbolic terms residing in an object based environment. State equations are symbolically derived and therefore have no particular parameter assumptions. The causality assignment procedure is monitored by the user and at any failure point the state of the assignment process is available for analysis. Enhanced analysis of the causality assignment procedure will give the engineer added insight into modelling errors and help identify system design strategies.

3.0 Constraint Propagation for Causality Assignment

Causality assignment is based on establishing the relationships which must exist between system variables in order for the system to operate as a state determined system. As described in Section 2.3, there are elements whose causal relationship is implicit in the definition of the element. However, other elements may have a variety of valid assignments. In this case, the assignment becomes a function of the causal environment in which the element exists.

Since a bond graph defines a network of elements, a causal assignment within the network necessarily affects neighboring elements. Depending on the situation, this may or may not restrict the causal environment enough to allow or prevent further assignments. This process may be viewed as the propagation of causal assignments through the network where each element defines constraints on the causal environment in which it exists.

The order in which the assignments are made proceeds from the most restricted to least restrictive elements as defined in Table 4. Initially, all effort and flow sources are assigned. The effects of the assignments are propagated to the neighboring elements and further assignments are made if possible. If no further assignments can be made, then an energy storage element is assigned the preferred integral causality. The effects of the assignment are propagated as far as possible. If no further assignments can be made, a second energy storage element is assigned integral causality and its effects are propagated. This process continues until all elements have been assigned, or a constraint conflict occurs.

As an example, consider the simple spring mass system in Figure 3. By definition, an effort source must have the causal assignment shown in Table 4. The series junction is therefore directly affected by this assignment (Figure 4a). However, at this point, the series junction has not been restricted to any particular assignment. The next step is to assign integral causality to energy storage elements. Integral causality is assigned to the inductive element, (Figure 4b), which directly affects the series junction environment. Since by definition only one of the attached elements may impose a flow into a series junction, the junction is assigned as in Figure 4c. As a consequence of the junction assignment, the capacitive element has received its assignment. Since all elements have assignments which are consistent with their constraints, the causality assignment process is complete.

This approach to causality assignment is a natural fit to constraint propagation techniques, since the basis of the procedure is to assign a causality and let the consequences of the assignment filter through the network. When causality conflicts occur, the propagation path may be traced back to the elements that created the conflict environment. This information provides important insight into modelling errors. In addition, this approach allows the user to select the desired causality for particular elements in the model, and have the system complete the assignment or identify the conflict if the desired causality cannot be achieved.

4.0 Derivation of State Equations

The state equations are derived from the network of relations defined by the bond graph representation and the definition of a set of state variables for the model. This information is used to build MACSYMA commands for symbolic processing.

The network of relations is defined directly from the bond graph representation, with each element inheriting its associated relation as described in Table 3. Each element also inherits a unique symbol to represent the parameter in the relation. The relation may be redefined by the user at any time if the default relation does not adequately model the element (e.g. to include nonlinear effects).

The state variables are selected by identifying the independent energy storage elements, which are elements in the bond graph that exhibit integral causality. The independent variable for each element is defined as a state variable and the remaining variables in the model are defined as secondary variables.

The final step is to build the MACSYMA commands for combining the relation network and extracting the state equations. This is accomplished by requesting MACSYMA to eliminate the secondary variables from the system of relations and then solve for the state variables.

As an example, consider the hydraulic actuator shown in Figure 5(a), with the bond graph representation in Figure 5(b).

The system creates an analysis window which identifies each element relation that will be used for deriving the state equations. These relations are annotated with an expression number as shown in Figure 6. The state variables and external inputs are automatically determined by the system and displayed to the user. The relations, state variables, secondary variables and external inputs are downloaded to MACSYMA for symbolic processing. The resulting state equations, shown as the last three lines of Figure 6, are presented to the user for analysis. These may be reformatted as the state equations in Figure 7. An interface for displaying the mathematical equations is presently under investigation.

5.0 Implementation

The system described here was developed on a Xerox 1109 work station using a frame based system developed at Lawrence Livermore Laboratory, called FAR. Both FAR and the application are written in CommonLisp. The system communicates with MACSYMA running on a VAX 750.

Figure 3 (a). Simple Spring-Mass System

Figure 3 (b). Equivalent Bond Graph

Figure 4. Causality Assignment for Simple Spring-Mass System

Figure 5. Hydraulic Actuator Example

```

Numerical Analysis Window

Model equations for submission to Macsyma:
EXP1:DERIV.FORCE.SPRING1=VELOCITY.SPRING1*K.SPRING1
EXP2:DERIV.VELOCITY.MASS2=FORCE.MASS2*1.0/M.MASS2
EXP3:PRESSURE.PIPE7=FLOW.PIPE7*B.PIPE7
EXP4:FORCE.DASHPOT3=VELOCITY.DASHPOT3*B.DASHPOT3
EXP5:OUT.FLOW.COMMON.FLOW6.TO.PISTON5=OUT.FLOW.PISTON5.TO.COMMON.VELOCITY4*MODULUS.PISTON5
EXP6:OUT.EFFORT.PISTON5.TO.COMMON.VELOCITY4=OUT.EFFORT.COMMON.FLOW6.TO.PISTON5*MODULUS.PISTON5
EXP7:FLOW.PIPE7=OUT.FLOW.COMMON.FLOW6.TO.PISTON5
EXP8:EXT.PUMP8=OUT.EFFORT.COMMON.FLOW6.TO.PISTON5+PRESSURE.PIPE7
EXP9:VELOCITY.SPRING1=OUT.FLOW.PISTON5.TO.COMMON.VELOCITY4
EXP10:VELOCITY.SPRING1=VELOCITY.DASHPOT3
EXP11:VELOCITY.SPRING1=VELOCITY.MASS2
EXP12:OUT.EFFORT.PISTON5.TO.COMMON.VELOCITY4=FORCE.DASHPOT3+FORCE.MASS2+FORCE.SPRING1

Sending symbolic expressions to Macsyma
ELIMINATING VARIABLES ...
SOLVING FOR STATE VARIABLES...
EXPANDING SOLUTION ...

STATE EQUATIONS :
DERIV.VELOCITY.MASS2=      -B.PIPE7*MODULUS.PISTON5+2*VELOCITY.MASS2/M.MASS2
                           -B.DASHPOT3*VELOCITY.MASS2/M.MASS2
                           +EXT.PUMP8*MODULUS.PISTON5/M.MASS2
                           -FORCE.SPRING1/M.MASS2
DERIV.FORCE.SPRING1=      K.SPRING1*VELOCITY.MASS2

```

Figure 6. Analysis Window for Hydraulic Actuator Example

$$\begin{pmatrix} \frac{d\text{Force}_{\text{spring1}}}{dt} \\ \frac{d\text{Velocity}_{\text{mass2}}}{dt} \end{pmatrix} = \begin{pmatrix} 0 & k_{\text{spring1}} \\ -\frac{1}{m_{\text{mass2}}} & -\frac{(\text{modulus}_{\text{piston}}^2 * r_{\text{pipe6}} + b_{\text{dashpot}})}{m_{\text{mass2}}} \end{pmatrix} \begin{pmatrix} \text{Force}_{\text{spring1}} \\ \text{Velocity}_{\text{mass2}} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\text{modulus}_{\text{piston}}}{m_{\text{mass2}}} \end{pmatrix} \text{Pressure}_{\text{pump8}}$$

Figure 7. Formatted State Equations for Hydraulic Actuator

There are three basic object structures: the model data objects, the control objects (called managers), and the constraint objects (Figure 8).

The model objects define the underlying data structure which store information about the model. They are created from a class structure based on the basic bond graph elements. Figure 9 shows the object hierarchy.

Each model element is an instance of one of the bond graph objects. An example of the spring model object from the spring-mass system of Figure 3 is shown in Figure 10.

The control structure for the system is built as a collection of objects, called managers, that have been assigned specific development tasks. Each manager has a particular task agenda and upon receipt of a control message will try to complete all of its tasks. This design is based on work related to constraint propagation by Steels [8]. There are presently two managers in the system, the causality assignment manager and the symbolic analysis manager.

The causality assignment manager directs the causality assignment procedure. It controls the propagation through a record structure which maintains a current causality value and a constraint causality value for each object. The constraint objects are polled for advice about how the causality should be assigned. If the advice conflicts with existing knowledge about the current values, an assignment failure is flagged. The entire procedure is displayed in a window, so that the user may evaluate the state of the assignment process when the causality failure occurred and determine the cause of failure. The user may then use a menu selection to edit the model and create a valid model.

The symbolic analysis managers task is to build the state equations for the model. It has a procedural task agenda for creating the symbolic equations and preparing them as input to MACSYMA.

6.0 Conclusions

The system described here allows an engineer to interactively develop a model of a dynamic physical system using a

Figure 8. Basic Object Structures

Figure 9. Model Object Hierarchy

Spring1	
name	Spring1
effort	ForceSpring1
flow	VelocitySpring1
parameter	KofSpring1
causality	ImposesAnEffort
connected-to	SeriesJunction2
domain	Mechanical
symbolic-expression	Deriv.ForceSpring1=KofSpring1*VelocitySpring1

Figure 10. Example of Spring Model Object

representation structure which allows reasoning about the dynamics from an abstract model. The system assigns the necessary causal relationships which must exist in a state determined system and selects the state variables for the physical system. The underlying representation defines the elemental relations for the model which describe the behavior. The elemental relations, state variables, secondary variables, and external inputs are automatically assembled and down loaded to MACSYMA. Then the appropriate MACSYMA commands are created for deriving the state equations for the system. The resulting state equations are completely in symbolic form and have no particular parameter assumptions which provides the user with direct information about how the parameter values will affect the system function. The causal analysis methods provide added insight into variable relationships which must exist if the physical system is to operate as a state determined system. Finally, the system provides a well structured representation for reasoning about dynamical systems.

The system is currently being extended in three basic areas. The physical system to bond graph abstraction process is being analyzed in order to assist the engineer, who is not familiar with bond graph techniques, in creating the representation. Secondly, the symbolic state equations will be analyzed and mapped back to the original bond graph representation. With this information, we wish to highlight how a parameter value of a particular element will effect the characteristic equation that describes the system behavior. Finally, a qualitative reasoning system is being developed which will generate potential behavior patterns for the physical system. By integrating the symbolic with the qualitative approach, we hope to reduce the generation of non-physically realizable causal alternatives which are often the by-product of a qualitative modelling system.

Acknowledgement

Special thanks go to the Computation Department of Lawrence Livermore Laboratory for supporting this effort.

References

- [1] J. U. Thoma, *Introduction to Bond Graphs and their Applications*, Pergamon Press, New York, 1975.
- [2] D. Karnopp, R. Rosenberg, *System Dynamics a Unified Approach*, Wiley and Sons, New York, 1975.
- [3] R. Rosenberg, D. Karnopp, *Introduction to Physical System Dynamics*, McGraw-Hill, New York, 1983.
- [4] *The ENPORT-5 Users's Manual*, A. H. Case Center for Computer-Aided Design, Michigan State University, East Lansing, Mich., 1981.
- [5] J. J. van Dixhoorn, "Simulation of Bond Graphs on Minicomputers," *Journal Dynamic Systems, Measurement, and Control*, March 1977.
- [6] J. J. Granda, F. Pourrahi, "Computer Graphics Techniques For The Generation and Analysis of Physical System Models," *Proceedings of the Society of Computer Simulation Multiconference*, San Diego, Calif., January 23-25 1985.
- [7] G. Dauphin-Tanguy, C. Sueur, F. Coutereel, "Presentation of a Processor for the Computer Aided Modelling of Dynamical Systems Through Bond-graph Approach," *International Symposium on AI, Expert Systems and Languages in Modelling and Simulation*, Barcelona, Spain, June 2-4 1987.
- [8] L. Steels, "Constraints as Consultants," Schlumberger-Doll Research, *AI Publications*, Dec. 1981.