# SIMULATION OF A SOLAR-HEATED HOUSE USING THE BOND GRAPH MODELING APPROACH AND THE DYMOLA MODELING SOFTWARE

by

Spyros Andreou

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1990

## STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Andreou

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Dr. François E. Cellier May 30, 1990

F. E. Cellier Date
Associate Professor
Electrical and Computer Engineering

## ACKNOWLEDGEMENTS

I wish to express my grateful appreciation to my parents, Savvas and Andromachi, for their continuous support throughout my academic career.

Furthermore, I would like to extend my deepest appreciation to my academic advisor Dr. Francois Cellier for his valuable assistance to this project. He was always willing and available for help whenever I requested it. I also thank the other two members of the examining committee, Dr. Malur Sundareshan and Dr. Hal Tharp.

Special thanks to Qingsu Wang for her assistance in the DYMOLA software. Also, to Dr. Granino Korn for making available his computer to obtain the DESIRE outputs. And, finally to all my friends who contributed to this project to become reality.

## TABLE OF CONTENTS

**TABLE OF CONTENTS** (continued)

## TABLE OF CONTENTS (continued)

## LIST OF FIGURES

**LIST OF FIGURES** (continued)

## LIST OF FIGURES (continued)

## LIST OF FIGURES (continued)

## ABSTRACT

This thesis discusses the application of the bond graph modeling technique directly coded into the Dynamic Modeling Language (DYMOLA) for simulating a solar-heated house. Scientists throughout the years have investigated the exploitation of solar radiation for space heating. In this thesis, the physical behavior of such a system is modeled and simulated in a convenient, robust and fast manner. The bond graph modeling methodology has found widespread use in a wide range of systems. DYMOLA is a modeling language well suited to represent bond graphs. DYMOLA is a program generator that can map a topological system description, such as a bond graph, into a state-space description expressed in the form of a DESIRE simulation program.

# CHAPTER 1

# INTRODUCTION

Engineers are trained in inventing new means which will eventually lead to easier solutions of their problems. One of them is to model and then to simulate a certain number of physical systems encountered in their everyday life with the main target being to first predict and secondly to study their physical behavior.

The goal of this thesis is to present a modern and advanced modeling-simulation technique applied to a solar-heated house. The bond graph modeling technique as well as the Dynamic Modeling Language (DYMOLA) will be used.

There exist a number of bond graph modeling tools on the market. The best established tool is ENPORT-7 (Rosencode Associates Inc., 1989), a SPICE-like bond graph language with a graphical front end. Other tools are TUTSIM (van Dixhoorn, 1982) and CAMP (Granda, 1982). However, none of these systems is able to handle truly hierarchical bond graphs as they will be essential for our endeavor. DYMOLA (Elmqvist, 1978) is the only modeling language available which can handle truly hierarchical nonlinear bond graphs in a completely general fashion.

It is attractive to many engineers to study the possibility of exploiting the freely available solar radiation for heating a house. For a successful design of such a facility, it is essential that the system behavior can be simulated so that various alternatives can be tested prior to implementation. DYMOLA together with the bond graph approach to physical system modeling is expected to be the quickest and most accurate method compared with others used in the past to describe such a system. Bond graphs were invented in 1960 by Henry Paynter, an MIT professor (Paynter, 1961), and DYMOLA was designed at the Lund Institute of Technology in 1979 by Hilding Elmqvist in his Ph.D. dissertation (Elmqvist, 1978). However, the application of DYMOLA to express bond graphs is new and has never been done before.

Bond graphs find many applications in various engineering disciplines because they make modeling more systematic, because they make it easier to deal with interfaces between subsystems of different types (e.g., electro-mechanical couplers), and because they simplify the verification of a correct energy flow across such interfaces and within the subsystems. They are able to provide a common modeling methodology not only for electrical, mechanical and other frequently simulated systems, but also for less commonly simulated systems such as chemical, ecological or biomedical systems. They offer a more general graphical representation than either block diagrams or signal flow graphs since they preserve both the computational and the topological structure of all the systems mentioned above. As the word indicates, a bond graph is a collection of elements bonded together. More information about this unique modeling technique is provided in the second chapter.

After modeling our solar-heated house into bond graphs, the produced diagrams are directly coded into DYMOLA, a modular hierarchical continuous-system modeling language. Its main advantage is that it can deal with large-scale systems in a modular and hierarchical manner. Moreover, it is very well suited to implement the bond graph modeling methodology, and is able to map bond graphs into state-space descriptions of the type $\underline{x}' = \underline{f}(\underline{x},\underline{u},t)$. Special features of DYMOLA are found in the third chapter.

The main subject of the fourth chapter is a demonstration of the way in which DYMOLA can be used to solve the presented problem. The transition from the bond diagram to DYMOLA code is a straightforward procedure requiring several simple rules being presented in a concrete and succinct manner. DYMOLA is so powerful that it can automatically evaluate the causality of the bond graph, produce a state-space description for the system, as well as generate a simulation program coded in either DESIRE (Korn, 1989b) or SIMNON (Elmqvist, 1975), two direct executing continuous-system simulation languages. Moreover, a simple electrical network is included, transformed first into its bond diagram and then into DYMOLA code, with the hope that the reader will follow and comprehend all the presented steps in a convenient manner.

As mentioned before, the case study presented in this thesis is a solar-heated house, a relatively complicated system involving various subsystems and various types of energy. The configuration under study consists of a flat-plate solar collector, one solid body storage tank, water loops, a heat exchanger, and the habitable space. Each part is governed by a set of first order differential equations illustrating the energy flow through the subsystem. Each subsystem is directly transformed into a bond graph representation. The various parameters used for the simulation were taken from an older study of a similar solar-heated house performed in the late 70's (Kass, 1978), from other sources in the literature (Deffie and Bechman, 1980) and from using our physical intuition and common sense.

It is hoped to have the opportunity to apply both the bond graph modeling technique and the dynamic modeling language in industry observing the physical properties of various systems. Being able to translate them into bond diagrams and then code them directly into DYMOLA is, indeed, an exciting experience.

# CHAPTER 2

# BOND GRAPHS

In this chapter the Bond Graph methodology is discussed extensively. It starts with an overview of this unique modeling technique, then it gives some basic definitions with illustrations and it discusses the concept of causality. Furthermore, a reference to Pseudo Bond Graphs and Thermal Systems is given.

## 2.1 Overview

Engineers needed to find a more general graphical (symbolic) representation which attempts to preserve both the computational and topological structure of any kind of physical system. They found out that block diagrams and signal flow graphs only preserve the computational but not the topological structure. Thus, a relatively new and powerful representation is that of Bond Graphs which has been introduced by Henry Paynter in the early Sixties (Paynter, 1961). Many types of physical systems have been studied using bond graphs including electrical networks, mechanical rigid bodies, hydraulic, thermal and energy transduction phenomena. Some researchers refer to Bond Graphs also as Bond Diagrams. We shall use both terminologies interchangeably.

It is true, however, that for the beginner this modeling language is quite abstract. Block diagrams and signal flow diagrams can be more easily comprehended. Nevertheless, for the case of modeling the solar-house, a relatively complicated system involving many different types of energy flow between its interconnected parts, it appears that the bond graph procedure is more appealing due to its ease of application and greater information content.

Modeling a physical system is a simplified abstract construction used to predict its physical behavior. That is exactly what the bond graph modeling methodology is performing.

The purpose of this chapter is to introduce the reader to this abstract modeling methodology and to provide enough information so that he/she can easily comprehend it.

## 2.2 Basic Definitions

### 2.2.1 Multiport Elements, Ports, and Bonds

The nodes of the graph are called Multiport Elements designated by alpha-numeric characters such as 1 and R, as shown in Figure 2.1(a). The places where a multiport element can interact with its environment are called Ports designated by line segments incident on the element at one end. Figure 2.1(b) shows the 1 element having three ports and the R element having one port. When pairs of ports are combined

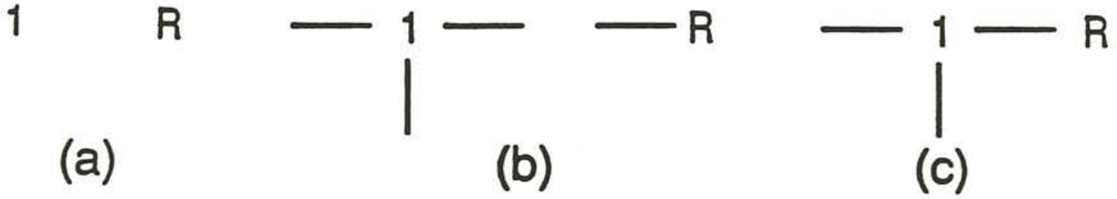


Figure 2.1 (a) Multiport elements
(b) The elements and their ports
(c) Formation of a bond

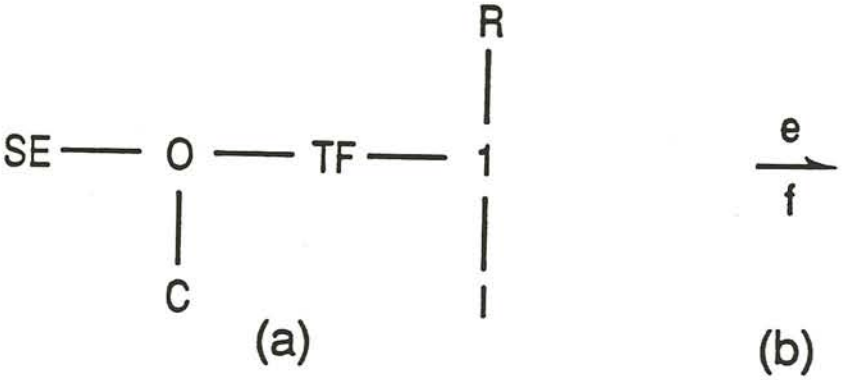


Figure 2.2 (a) Bond graph
(b) A bond

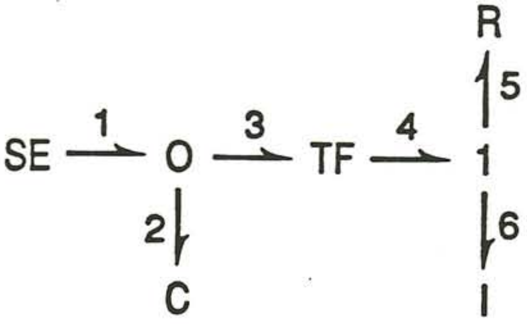


Figure 2.3 The Bond Graph with powers directed and bonds labeled

together, bonds are formed. Thus, bonds are connections between pairs of multiport elements. For example, Figure 2.1(c) shows a formation of the bond between 1 and R.

### 2.2.2 Bond Graphs

A bond graph is a collection of multiport elements bonded together. In a more general perspective it is a linear graph with nodes being the multiport elements and with branches being the bonds. An example of a bond graph is shown in Figure 2.2(a) having seven multiport elements and six bonds.

Another definition:

"A bond, represented by a bold half arrow, is nothing but a connector that simultaneously connects two variables, one across variable, in bond graph terminology usually referred to as the 'effort' e, and one through variable, called the 'flow' f" (Cellier, 1990a). Refer to Figure 2.2(b) as well as to the next subsection for more information.

### 2.2.3 Port variables

There are three direct and three integral quantities associated with a given port.

The first two direct quantities are called Effort, e(t), one across variable and Flow, f(t), one through variable, assumed to be scalar functions of an independent variable (t).

The scalar product of effort and flow is called Power defined by

$$P(t) = e(t) \cdot f(t) \tag{2.1}$$

comprising the third direct quantity.

In Figure 2.3, the same bond graph is drawn but it has its powers directed and bonds labeled. The direction of the positive power is indicated by a half-arrow on the bond.

Two of the three integral quantities are Momentum, p(t), and Displacement, q(t), which are related to effort and flow respectively as

$$p(t) = p(t_0) + \int_{t_0}^{t} e(\tau)\,d\tau \tag{2.2}$$

and

$$q(t) = q(t_0) + \int_{t_0}^{t} f(\tau)\,d\tau \tag{2.3}$$

The third is Energy which is related to power as

$$E(t) = E(t_0) + \int_{t_0}^{t} P(\tau)\,d\tau \tag{2.4}$$

The net energy, represented by $E(t)-E(t_0)$, is transferred through the port in the direction of the half-arrow (positive power) over the interval $(t_0,t)$.

### 2.2.4 Basic Multiport Elements

There are nine basic multiport elements divided into four categories according to their energy characteristics. There are two Sources, two Storages, one Dissipation and four

Junctions. The two sources, the two storages (capacitance and inertance) and the dissipation (resistance) are 1-port elements whereas two of the junctions (transformer and gyrator) are 2-port ones and the other two (0 and 1) are at least 3-port elements. The following Figure 2.4 shows the symbol, definition and name of the nine basic multiport elements. In the figure, Φ stands for a general function relating two variables.

### 2.2.5 Extended Definitions

Although the following features are beyond the scope of this thesis they are worth mentioning. The term Field is also used in bond graph terminology. Thus, there are C-fields, I-fields and R-fields which are multiport generalizations of -C, -I and -R respectively. Moreover, there are the Modulated Transformer (MTF) and Modulated Gyrator (MGY).

Later in the fifth chapter, when the bond graph of a three-dimensional cell is constructed the R-field is used (three resistors are connected in x, y, z directions, see Figure 5.6a).

### 2.2.6 Generalization to Basic Physical Types of Systems

We have already seen four generic variables effort, flow, momentum and displacement. The following Figure 2.5 demonstrates a presentation summarizing the above four

| SYMBOL | DEFINITION | NAME |
|---|---|---|
| $SE \xrightarrow{e}$ | $e = e(t)$ | source of effort |
| $SF \xrightarrow{f}$ | $f = f(t)$ | source of flow |
| $C \xleftarrow[f]{e}$ | $e = \Phi(q)$<br>$q(t) = q(to) + \int fdt$ | capacitance |
| $I \xleftarrow[f]{e}$ | $f = \Phi(p)$<br>$p(t) = p(to) + \int edt$ | inertance |
| $R \xleftarrow[f]{e}$ | $\Phi(e,f) = 0$ | resistance |
| $\xrightarrow{1} \underset{m}{TF} \xrightarrow{2}$ | $e_1 = me_2$<br>$f_2 = mf_1$ | transformer |
| $\xrightarrow{1} \underset{r}{GY} \xrightarrow{2}$ | $e_1 = rf_2$<br>$e_2 = rf_1$ | gyrator |
| $\begin{array}{c} \xrightarrow{1} 0 \xrightarrow{3} \\ \uparrow 2 \end{array}$ | $e_1 = e_2 = e_3$<br>$f_1 + f_2 - f_3 = 0$ | common effort junction |
| $\begin{array}{c} \xrightarrow{1} 1 \xrightarrow{3} \\ \uparrow 2 \end{array}$ | $f_1 = f_2 = f_3$<br>$e_1 + e_2 - e_3 = 0$ | common flow junction |

Figure 2.4 Definitions of the basic multiport elements

| | | | **Generalized** | **Generalized** |
|---|---|---|---|---|
| | **Effort** | **Flow** | **Momentum** | **Displacement** |
| | **e** | **f** | **p** | **q** |
| Electrical | voltage<br>$u\,[\mathrm{V}]$ | current<br>$i\,[\mathrm{A}]$ | flux<br>$\Phi[\mathrm{V\,s}]$ | charge<br>$q\,[\mathrm{A\,s}]$ |
| Translational | force<br>$F\,[\mathrm{N}]$ | velocity<br>$\upsilon\,[\mathrm{m\,s^{-1}}]$ | momentum<br>$I\,[\mathrm{N\,s}]$ | displacement<br>$x\,[\mathrm{m}]$ |
| Rotational | torque<br>$T\,[\mathrm{N\,m}]$ | angular<br>velocity<br>$\Omega\,[\mathrm{rad\,s^{-1}}]$ | twist<br>$\tau[\mathrm{N\,ms}]$ | angle<br>$\theta\,[\mathrm{rad}]$ |
| Hydraulic | pressure<br>$P\,[\mathrm{N\,m^{-2}}]$ | volume<br>flow<br>$\Phi_V[\mathrm{m^3 s^{-1}}]$ | pressure<br>momentum<br>$\Pi[\mathrm{N\,m^{-2}s}]$ | volume<br>$v\,[\mathrm{m^3}]$ |
| Chemical | chemical<br>potential<br>$\mu[\mathrm{J{\cdot}mol^{-1}}]$ | molar flow<br>$\frac{dN}{dt}[\mathrm{mol{\cdot}s^{-1}}]$ | | molar mass<br>$N\,[\mathrm{mol}]$ |
| Thermo-<br>dynamical | temperature<br>$T\,[\mathrm{{}^{\circ}K}]$ | entropy flow<br>$\frac{dS}{dt}\,[\mathrm{W\,{}^{\circ}K^{-1}}]$ | | entropy<br>$S\,[\mathrm{J{\cdot}\,{}^{\circ}K^{-1}}]$ |

Figure 2.5 Presentation of a summary of the four generic variables being used in some common physical systems

generic variables being used to the most common physical system types.

### 2.3 The Concept of Causality

Bond graphs have the property of preserving the topological as well as the computational structure of a system. When, for example, a given electrical system is transformed into bond graphs its topological structure is quite evident to the reader. Nevertheless, its computational structure cannot be seen easily. Thus, the introduction of bond graph causality comes into account.

We say that in bond graphs inputs and outputs are specified by means of the causal stroke. It is a short perpendicular line made at one end of a bond or port line. It indicates the direction in which the effort signal is directed, implying that the other end which does not have a causal stroke is the one that the flow signal arrow points. Figures 2.6(a) and 2.6(b) illustrate succinctly the meaning of causality (causal stroke).

The following Figure 2.7 shows the nine multiport elements with their desired causal forms and relations. It is worthwhile saying that for resistance both causal forms (as shown) are physically and computationally possible. However, for capacitances and inertances we would rather pick the causalities that numerically integrate over all state variables.

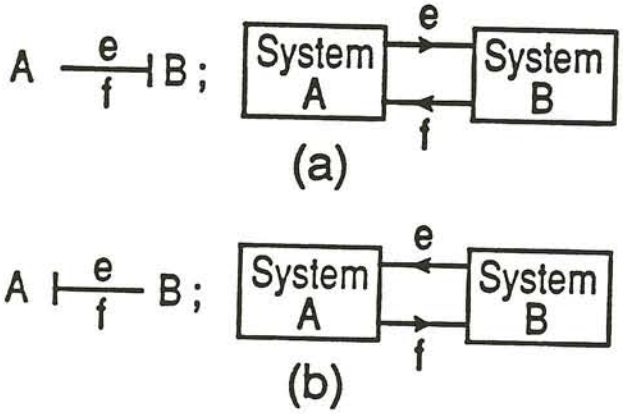


Figure 2.6 Meaning of causal strokes

(a) Effort is output of A, input to B;
flow is output of B, input to A
(b) Effort is output of B, input to A
flow is output of A, input to B

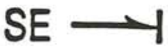
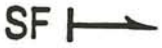
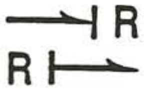
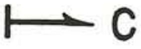
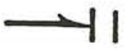
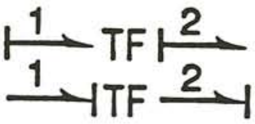
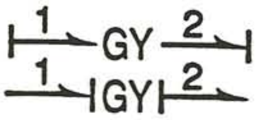
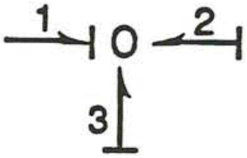
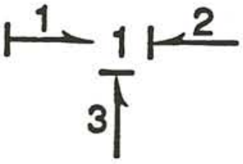
| ELEMENT | CAUSAL FORM | CAUSAL RELATION |
|---|---|---|
| Effort Source |  | $e\,(t) = E(t)$ |
| Flow Source |  | $f\,(t) = F(t)$ |
| Resistance |  | $f = \Phi_R^{-1}(e)$<br>$e = \Phi_R(f)$ |
| Capacitance |  | $e = \Phi_C^{-1}(\int f dt)$ |
| Inertance |  | $f = \Phi_I^{-1}(\int e dt)$ |
| Transformer |  | $e_1 = me_2\,,\ f_2 = m\,f_1$<br>$f_1 = f_2/m,\ e_2 = e_1/m$ |
| Gyrator |  | $e_1 = rf_2\,,\ e_2 = rf_1$<br>$f_1 = e_2/r,\ f_2 = e_1/r$ |
| O - Junction |  | $e_2 = e_1\,,\ e_3 = e_1$<br>$f_1 = -\,(f_2 + f_3)$ |
| 1 - Junction |  | $f_2 = f_1\,,\ f_3 = f_1$<br>$e_1 = -(e_2 + e_3)$ |

Figure 2.7 Desired causal forms and relations of the basic nine multiport elements

## 2.4 Pseudo Bond Graphs and Thermal Systems

Because of the fact that the solar-heated house is a thermal system, it is time to introduce some bond-graph representations for such a thermal system. Thermal systems have been presented as analogous to electrical systems, usually with temperature analogous to voltage and heat flow analogous to current. With this analogy in mind we have sources analogous to voltage and current sources, thermal resistors and capacitors, and 0 and 1 junctions. However, there are no thermal inertias (inertances).

There is one major obstacle. The product of temperature and heat flow isn't power. Heat flow is by itself a power. Engineers, then, decided to name such a bond graph in which the product of effort and flow isn't power a pseudo bond graph. As long as the basic elements in the pseudo bond graph are correctly related to the e, f, p, and q variables, the rules for the regular bond graph technique can be usefully applied. The true bond graph results (see Figure 2.8), if temperature and entropy flow are used as effort and flow variables respectively. Indeed, the product of temperature and entropy flow is power.

The following Figure 2.8 shows a thermal resistor and 1-junction as well as a thermal capacitor and 0-junction which are going to be used in the fifth chapter during the modeling procedure of the solar house.

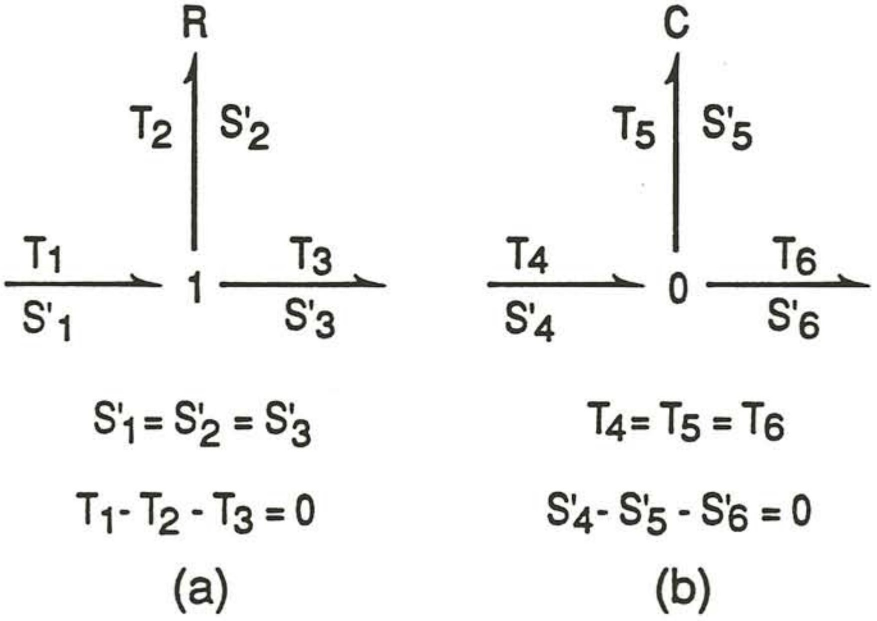


Figure 2.8 (a) Thermal resistor and 1-junction
(b) Thermal capacitor and 0-junction

Despite the fact that in the literature pseudo-bond graphs are more popular than the true-bond graphs in modeling thermal systems, it may be argued that using the latter ones will be more appropriate for modeling the solar house. True-bond graphs are better suited to represent the energy flow across a junction to and from other types of energy, such as mechanical, electrical, hydraulic, or pneumatic. Thus, as shown in the previous figure, temperature (T) would be the effort variable and entropy flow (S') would be the flow variable.

# CHAPTER 3

# DYMOLA

DYMOLA (Dynamic Modeling Language) is presented in this chapter. It is focused more on DYMOLA's main features, capabilities, unsolved problems and other important properties rather than on its software aspects. This thesis focuses more on modeling aspects than on software engineering aspects. More information about all the syntactic aspects surrounding the software can be found in Wang's thesis (Wang, 1989), particularly in the third chapter. Some of the figures presented in this chapter are very similar to those in Wang's thesis (3rd chapter).

## 3.1 Overview

As mentioned before, DYMOLA is a modeling language rather than a simulation language since it does not have its own simulation engine. It equips the user with a more comprehensible and better modularized hierarchically structured model description. A DYMOLA translator has as its input the hierarchically structured model, whereas as its output the governed model equations which are gathered into system equations.

DYMOLA is instructed by a compiler switch as to the desired simulation language the output is to be generated in.

Presently, DYMOLA supports DESIRE, SIMNON and FORTRAN and it would not be difficult to enhance it to support other languages, such as ACSL, as well.

DYMOLA uses two concepts: the submodel concept as well as the cut concept. These will be clarified later in this chapter.

There exist currently two different implementations of DYMOLA, one coded in PASCAL and the other coded in SIMULA. The first one runs on VAX/VMS and on PC compatibles, while the latter runs on UNIVAC computers.

## 3.2 Special Properties of DYMOLA Model Descriptions

### 3.2.1 Some Properties

The following are properties of a DYMOLA model. Some are quoted directly from Cellier's book (Cellier, 1990a), others are paraphrased:

(1) DYMOLA variables can be of two types: the terminal type and the local type. If they are connected to something outside the model, they will be of the terminal type; otherwise, they will be of the local type (connected inside the model).

(2) Terminals might be either inputs or outputs, frequently depending on the surroundings to which they are connected. The user has the right

to declare them the way he wants them to be by explicitly specifying input or output.

(3) DYMOLA constants can be of the parameter type if the user wishes to do so. Parameter values can be assigned from outside the model, but they can alternatively also assume default values.

(4) "Terminals can have default values. In this way, they don't need to be externally connected" (Cellier, 1990a).

(5) The first time derivative of state variable x can be expressed in two ways, either through der(x) or through x'. Second derivatives can be written as either der2(x) or x".

(6) The user cannot set initial conditions for the integrators inside a model, showing clearly a flaw of DYMOLA.

(7) The syntax expression = expression is used in DYMOLA equations, being solved for the proper variable during the process of a model expansion. DYMOLA accepts the fact that the left hand side of an equation can have der (temperature), while temperature appears on the left hand side of another.

(8) When multiplying terms by a zero parameter, they are automatically eliminated during a model expansion. For example, if we have

$$L\alpha = 0.0 \tag{3.1}$$

and the model equation

$$L\alpha * der\ (i\alpha) = u\alpha - ui - R\alpha * i\alpha \tag{3.2}$$

then the above is replaced by

$$0.0 = u\alpha - ui - R\alpha * i\alpha \tag{3.3}$$

resulting in the following three possible simulation equations:

$$(a)\quad u\alpha = ui + R\alpha * i\alpha \tag{3.4}$$

$$(b)\quad ui = u\alpha - R\alpha * i\alpha \tag{3.5}$$

$$(c)\quad i\alpha = (u\alpha - ui)/R\alpha \tag{3.6}$$

depending on the environment in which the model is used.

If $L\alpha \neq 0.0$, then the model equation is always transformed into

$$der\ (i\alpha) = (u\alpha - ui - R\alpha * i\alpha)/L\alpha \tag{3.7}$$

(9) "The above rule indicates that parameters with value 0.0 are treated in a completely different manner than all other parameters" (Cellier, 1990a). Parameters which are not equal to zero are maintained in the generated simulation code, whereas the ones with 0.0 value are not represented in the simulation code.

(10) DYMOLA models are modular because the equations can automatically be solved during model expansion.

### 3.2.2 The "Cut" Concept

When advancing to higher levels of the hierarchy, the number of the parameters will be growing. Similar to real systems where wires are grouped into cables and cables are grouped into trunks, the concept of "cut" has been introduced in DYMOLA to group variables together. Cuts correspond to complex connection mechanisms of physical systems like electrical wires, pipes and shafts. A more precise definition is the following: "Cuts are hierarchical data structures that enable the user to group individual wires into buses or cables and cables into trunks. A cut is like a plug or a socket. It defines an interface to the outside world" (Cellier, 1990b).

The following two figures, 3.1 and 3.2, show a model of a conductance (inverse of resistor) illustrating the different model descriptions before and after using the concept of cut. This example demonstrates how a continuous model achieves modularity.

With cut declarations the input and output variables do not change the model description by switching them. Nevertheless, the main advantage of cut is the two types of variables, the across and through variables with which there

```
    model name : conductance
    input : I
    output : V
    parameter : G
    equations : V = I/G

or  model name : conductance
    input : V
    output : I
    parameter : G
    equations : I = V • G
```

Figure 3.1 Model of a conductance using input output declaration

```
model name : conductance
cut : A(Va / I ) B(Vb / -I )
local : V
parameter : G
equations : V = Va -Vb
            V = I / G
```

Figure 3.2 Model of a conductance using cut declaration

is associated as in the real physical world a connection mechanism. The equations which describe the physical laws at the connection mechanism are automatically generated by the declaration of cut and the connection statements.

Consider the following example: Three submodels defined as "$G_1$," "$G_2$" and "$G_3$" have A and B as their cut variables. $V_\alpha$ and I are the across variable and the through variable associating with cut A, respectively, being declared as (see Figure 3.3)

cut A ($V_\alpha$/I)

Using the connect statement

connect $G_1$:A at $G_2$:A at $G_3$:A,

the following equations are automatically generated:

$$G_1.V_\alpha = G_2.V_\alpha \tag{3.8}$$

$$G_2.V_\alpha = G_3.V_\alpha \tag{3.9}$$

$$G_1.I + G_2.I + G_3.I = 0 \tag{3.10}$$

The above equations describe what exactly happens at the boundary of the subsystem where two or more elements are connected. "Thereby, all across variables (to the left of the slash separator) are set equal, and all the through variables (to the right of the slash operator) are summed up to zero" (Cellier, 1990b).

When several cuts are grouped together, a hierarchical cut is formed in the same way as individual wires are grouped together into a cable.

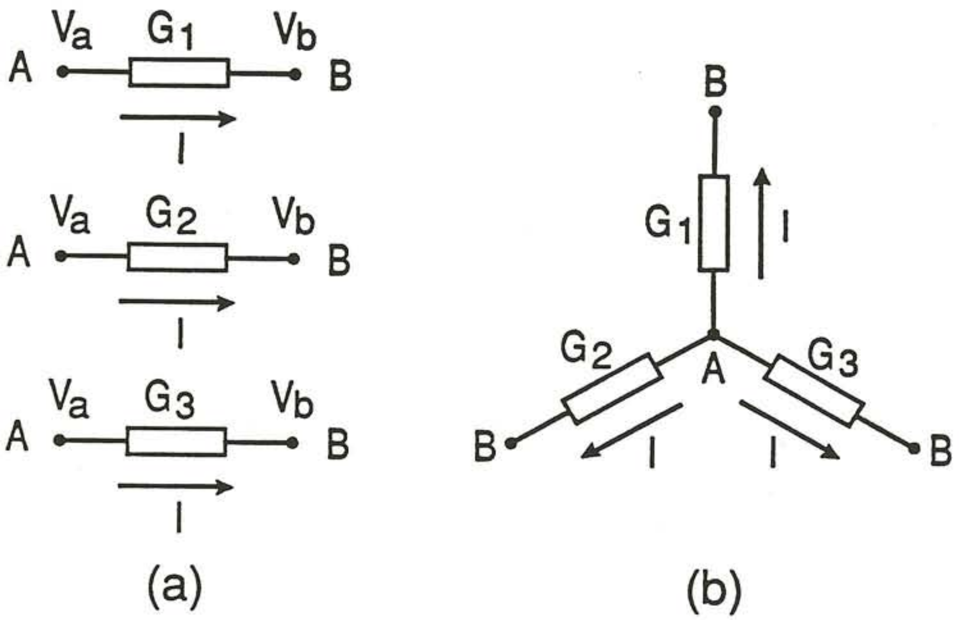


Figure 3.3 Three submodels connected at port A

(a) Three submodels
(b) Connected at A

Using the concept of cut, the following conclusions can be derived:

(1) When varying the I/O variables, a model in a continuous system can avoid a change in its model description.

(2) It separates the physical laws which describe the static and dynamic properties of the model from the physical laws which dominate at several subsystems at their connecting points.

(3) Models in DYMOLA are said to be in proper modular form so that the user can build them in a hierarchical modular manner.

The above concept can be extended to other systems such as mechanical, hydraulic and thermal systems. Being interested in the last ones, it is worthwhile mentioning that temperature and pressure are across variables, whereas heat flow is a through variable.

### 3.2.3 The "Submodel" Concept and "nodes" in DYMOLA

A submodel might be an atomic model, i.e., a model without coupling, or a coupled model.

Figure 3.4 illustrates atomic models, whereas the following one (Figure 3.5) illustrates a coupled model. In the latter one, the submodels of a resistor and a capacitor are depicted which are in modular form. The resistor's only

**Example 1 : A RESISTOR**

```
model resistor
  cut A (VA / I ) B (VB / -I )
  local V
  parameter R = 1
  V = VA-VB
  R • I = V
end
```

**Example 2 : A CAPACITOR**

```
model capacitor
  cut A (VA / I ) B (VB / -I )
  local V
  parameter C = 1
  V = VA-VB
  C • der (V) = I
end
```

Figure 3.4 Examples of atomic models in DYMOLA

```
model prc
  submodel resistor (30)
  submodel capacitor (20)
  cut A (VA / I )
  cut B (VA / -I )
  connect resistor : B at capacitor : B at B
  connect resistor : A at capacitor : A at A
end
```

Figure 3.5  Example of a coupled model prc

parameter R is 30 (indicated by the (30) in the submodel statement), whereas the capacitor's only parameter C is 20.

By coupling the two atomic models together, a coupled model "prc" is produced. "prc" stands for parallel connected resistor and capacitor. The coupled model is in proper modular form and can be used to construct larger systems. This concept of coupled models in DYMOLA is shown in Figure 3.6.

The "node" statement will be seen very often in a DYMOLA program. Nodes are convenient ways to make several connections acting like the power distributor. We plug several appliances into one distributor. For example, we can have

```
node n
connect x:A at n
connect y:B at n
```

which is equivalent to the single statement

```
connect x:A at y:B
```

### 3.2.4 Hierarchical Model Structure in DYMOLA

Figure 3.7 depicts a system named "S" decomposed into several subsystems: "$S_1$", "$S_2$", and "$S_3$". "$S_2$" is decomposed into "$S_{21}$" and the last subsystem is further decomposed into "$S_{31}$" and $S_{32}$" showing an overall hierarchical structure.

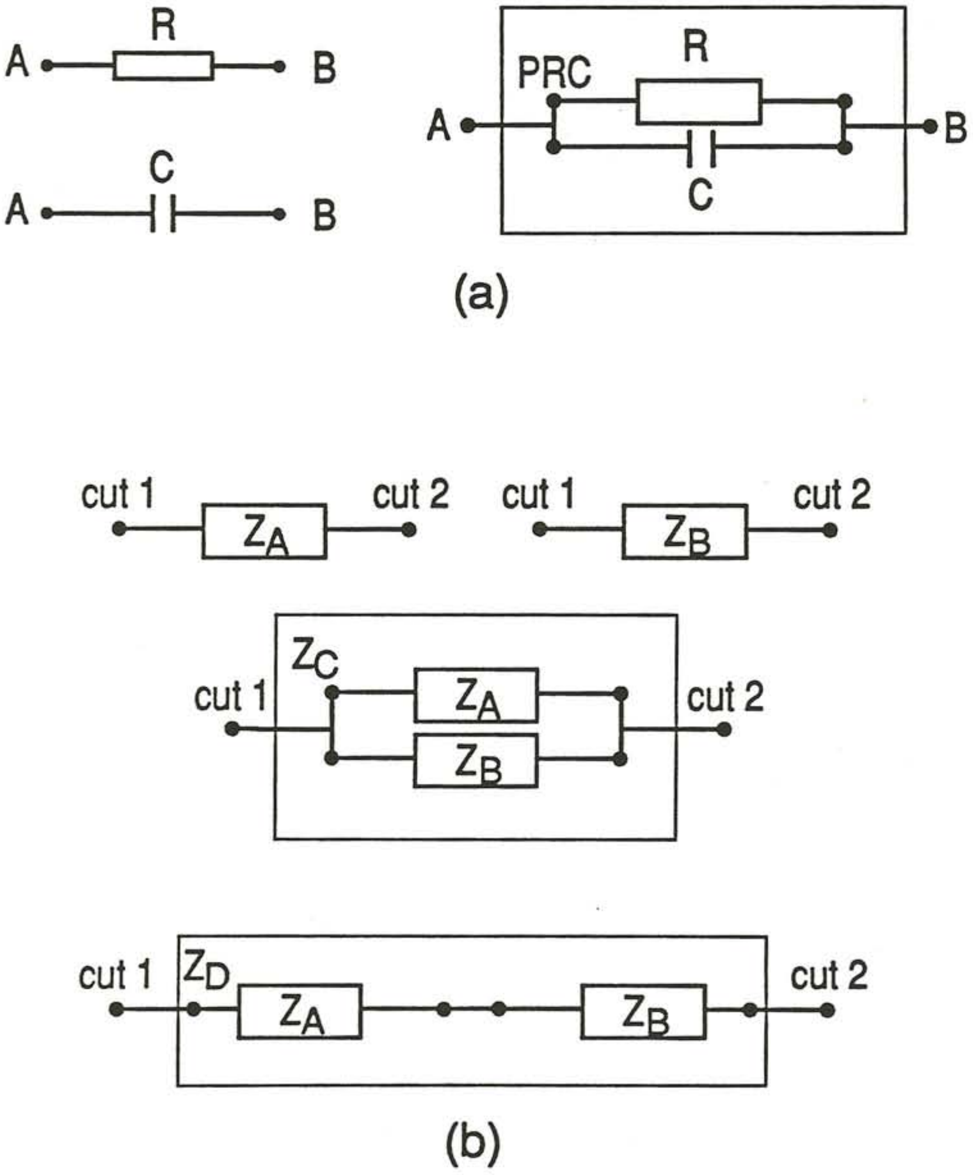


Figure 3.6 Coupled models in DYMOLA

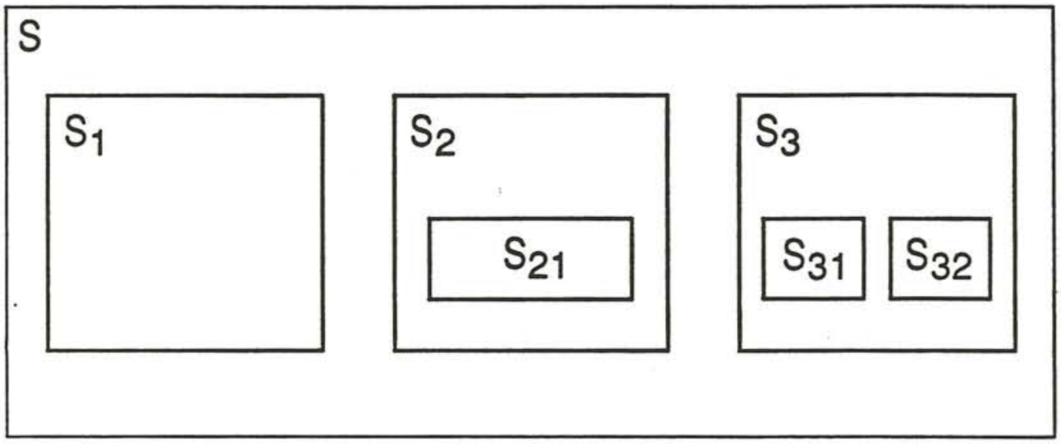


Figure 3.7  A hierarchically structured system

Figure 3.8 depicts one way to describe the hierarchical structure of the system (S) in DYMOLA. However, this technique has a serious flaw. For example, if system "$S_{21}$" and "$S_{32}$" are the same, the model specification must be repeated. In order to avoid duplicating subsystems with the same models, DYMOLA introduces a term called "model type."

"A model specified as 'model type' represents a generic model of a general class of objects. This 'model type' can be used to generate several models with a submodel statement so that duplication will be avoided" (Wang, 1989).

For instance, the "model resistor" and the "model capacitor" in the model specification can now be defined as "model type resistor" and "model type capacitor." The following Figure 3.9 demonstrates the same model specification as before but now using model types.

After creating model types of any system, it naturally comes to the user to declare libraries of models. This library is set up first when a system is modeled and then the hierarchy can be specified.

### 3.3 Generation of DESIRE Models

As mentioned before, DYMOLA is used to generate not only SIMNON and FORTRAN model but also DESIRE models. The following command is used for this purpose:

<u>output desire model</u>

```
Model S
  model $S_1$
  ......
  end
  model $S_2$
    model $S_{21}$
    ......
    end
  ......
  end
  model $S_3$
    model $S_{31}$
    ......
    end
    model $S_{32}$
    ......
    end
  ......
  end
......
end
```

Figure 3.8 Description of the hierarchical structure of a system in DYMOLA

```
model type PRC
  submodel (resistor) rtwo (30)
  submodel (capacitor) cone (20)
  cut A (VA / I )
  cut B (VB / -I )

  connect rtwo : A at cone : A at A
  connect rtwo : B at cone : B at B
end
```

Figure 3.9 Model specification for "prc" using model type

However, before proceeding, the user has to issue the command:

partition

which manipulates all the equations emanated by the model description and connection mechanism. This works in the following way: First, the computer will determine if a variable is present in an equation or not. Secondly, it finds out for which variable each equation must be solved. Thirdly, it partitions the equations into smaller systems of equations which must be solved at the same time. At the very end, it sorts the equations into the correct computational order.

### 3.3.1 Creation of a DESIRE Simulation Program

To create a DESIRE Simulation Program, a control portion of the DYMOLA program is added. In order to run the simulation of a continuous system, the basic information for simulation control such as simulation step, communication points and simulation time are required.

#### 3.3.1.1 Description of the Simulation Control Model

Its syntax is "cmodel" and it must be stored into a file with the same filename as that of the controlled system. It is indicated by the file extension "ctl" and is comprised of three parts:

(1) basic part

(2) run control block

(3) output block

In the basic part the following information is stored:

(1) simulation time

(2) simulation step size

(3) number of communication points

(4) inputs (optional)

The reader should consult Wang's thesis (Wang, 1989) concerning the format of the basic part.

The run control block involves the run control statements which can appear in the run-time control part of a DESIRE program.

The output block must contain the simulation output requirements. There are four output statements which are "dispt", "dispxy", "type", and "stash." Wang's thesis gives extensive details concerning their syntactic structures which are beyond the scope of this thesis. For this thesis, we require simulation graphs, so the "dispt" statement is going to be used.

#### 3.3.1.2 Obtaining Executable DESIRE Programs

The command

output desire program

will create executable DESIRE programs. First, the program verifies if the simulation control model associated with the system exists. Secondly, if the above is true, then an executable DESIRE program is generated; otherwise, an error message is displayed.

The procedure of generating DESIRE models will be shown with examples in the next chapter where a direct procedure of transforming bond graphs into DYMOLA code is developed.

### 3.4 Some Unsolved Problems

Currently, DYMOLA is still in a developing stage. A fair amount of research is needed to make DYMOLA a more productional code. There are, indeed, some unsolved problems which are listed below. They are good research topics for DYMOLA's future enhancement and advancement.

(1) DYMOLA is currently able to eliminate variables from equations of type $\alpha = \beta$. However, it is unable to eliminate variables from equations of type $\alpha \pm \beta = 0$.

(2) DYMOLA must be able to find out duplicate equations and to get rid of one of these automatically. This is very important for hierarchically connected submodels.

(3) "DYMOLA should be able to handle superfluous connections, i.e., if we specify that $w_2 = -w_1$, it is obviously true that also $b_2 = -b_1$" (Cellier, 1990a). (w is the angular velocity and

b is its corresponding angle.) Currently, DYMOLA cannot let the user specify this additional connection and eliminate superfluous connections during the model expansion.

(4) DYMOLA must be capable of recognizing that connections of outputs of integrators can always be transformed into connections of inputs of such integrators. For example, having $i\alpha_3 = i\alpha_2$, it is obviously true that $i\alpha dot_3 = i\alpha dot_2$. This reformulation can help eliminate structural singularities.

(5) "Groups of linear algebraic equations are currently grouped together and printed out by DYMOLA without being solved. DYMOLA should be able to rewrite the system of equations into a matrix form, since DESIRE can handle matrix expressions efficiently and future versions of DESIRE will include efficient algorithms for inverting matrices" (Wang, 1989).

(6) If, for example, the following expression is written

$$x^2 + z^2 + 2 * Y - 10 = 0$$

and it is desired to be solved for x or z, then problems will arise. DYMOLA cannot solve for second or higher order equations. It can solve for Y, however.

(7) DYMOLA can handle only continuous-time systems. It still cannot handle discrete time systems although DESIRE can handle them.

The aforementioned unsolved problems are the most noticeable ones. For more information, the reader can refer to Cellier's book (Cellier, 1990a) and Wang's thesis (Wang, 1989).

# CHAPTER 4

# CONSTRUCTION OF BOND GRAPHS AND THEIR TRANSFORMATION INTO DYMOLA

After discussing both the bond graph methodology and the Dynamic Modeling Language in the previous two chapters, this chapter focuses on the way to combine these two tools for modeling and simulating. A demonstration for constructing a bond graph for a simple electrical network is given and then its graph is transformed into DYMOLA code. It is a simple, direct procedure as will be seen.

## 4.1 Overview

A detailed procedure for constructing the bond graph is provided. The sample system is going to be a simple electrical network. Several diagrams are drawn demonstrating the step by step procedure so that the reader can follow it without any difficulty.

Once the bond graph for the given system has been constructed, it can be directly coded into DYMOLA. There are, however, several rules for this procedure that should be observed. They are stressed in the subsequent sections of this chapter. The basic bond graph modeling elements of R, C, L, TF, GY and bond can be described once and for all and stored away in a DYMOLA model library called "bond.lib."

At the very end, the DYMOLA coded program is run on the PC. It is going to be seen that DYMOLA is so powerful that it can automatically evaluate the causality of a bond graph, generate a state-space description for the system and finally generate a simulation program in currently either DESIRE or SIMNON, two "flat" direct executing continuous-system simulation languages. DESIRE is going to be used for this purpose.

## 4.2 Some Basic Rules for Constructing Bond Diagrams for Electrical Networks

Before proceeding to our construction of a bond diagram for a simple electrical network, we need to meet some regulations given in this section.

(1) In the 0-junction, all effort variables are equal, whereas all flow variables add up to zero.

(2) In the 1-junction, all flow variables are equal, whereas all effort variables add up to zero.

Therefore, for an electric circuit diagram the 0-junction is equivalent to a node, or a node in a DYMOLA program (Elmqvist, 1978). Moreover, the 0-junction represents Kirchhoff's current law, whereas the 1-junction represents Kirchhoff's voltage law. If two junctions are connected with a bond, one is always of the 0-junction type while the other is always of the 1-junction type. It can be said that

0-junctions and 1-junctions always toggle. Neighboring junctions of the same type can be amalgamated into one.

## 4.3 Construction of a Bond Diagram of a Simple Electrical Network

Because of my familiarity to electrical networks, I have chosen a simple electrical network to demonstrate the step by step procedure for constructing its bond graph.

The network is shown on Figure 4.1, with its node voltages labelled a, b, c and r.

### 4.3.1 The Step by Step Procedure

The following steps must be followed for constructing its bond graph:

(1) It is better to use voltages than currents, so Figure 4.2 shows three 0-junctions (voltage junctions) being laid out with subscripts corresponding to the nodes. The reference node is not represented by a 0-junction.

(2) Then, we represent each branch of the circuit diagram by a pair of bonds representing two 0-junctions with a 1-junction in between them (1-junction = current-junction). This is displayed in Figure 4.3

(3) Setting $V_r$ to zero, we can remove the bonds connecting the rest of the circuit (see Figure 4.4).

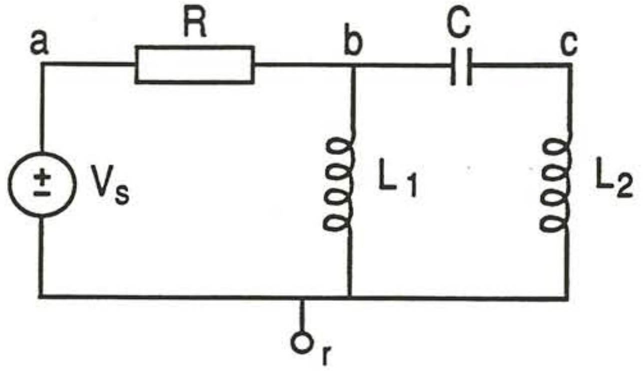


Figure 4.1 An electrical network with nodes labelled (r = reference)

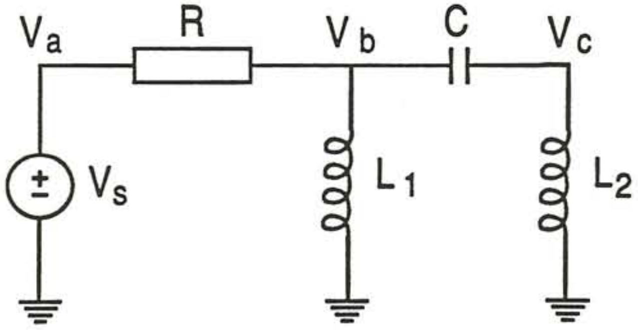


Figure 4.2 Layout of voltage junctions (0-junctions)

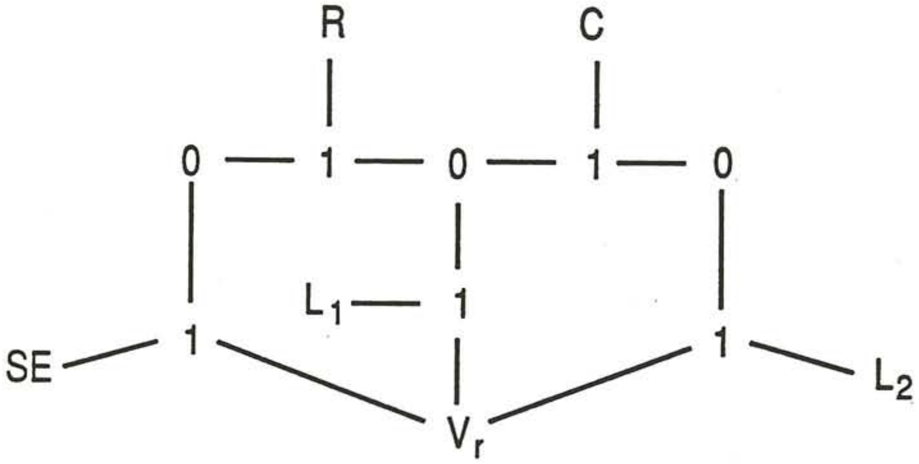


Figure 4.3 The assembly of components and source

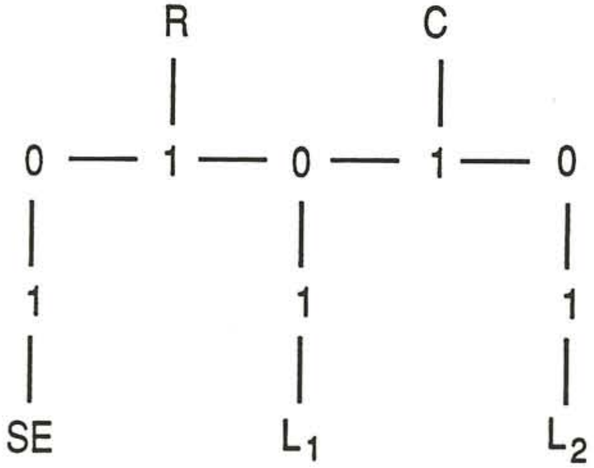


Figure 4.4 The cancellation of reference node and associated bonds

(4) If only two bonds go to a junction then this junction and its bonds are replaced by a single bond (see Figure 4.5). This has been done for the whole circuit demonstrated on Figure 4.6.

(5) Arrows point in the same direction as the branch currents are picked. The effort variables are assigned on the side the arrow points on the bond and for active elements such as sources (voltage, current) the arrow points towards the junction, whereas for passive elements (such as resistor, capacitor and inductor) the arrow points away from the junction.

So the bond graph is complete as indicated on Figure 4.7. The following Figure 4.8 demonstrates for each bond its voltage (effort variable) and current (flow variable) as well as the arithmetic values for each element. This is the detailed completed bond graph. R stands for resistance, C stands for capacitance, I stands for inductance (or inertia) and SE stands for effort source.

Now, let us assign causalities for the network. It can be seen how the energy flow is distributed throughout the electrical network. This is also shown on Figure 4.8 and it has been done based on Figure 2.7. Fortunately, every condition has been met, so our network is said to be a causal one.

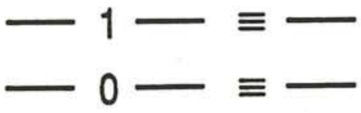


Figure 4.5 The condensation of bonds

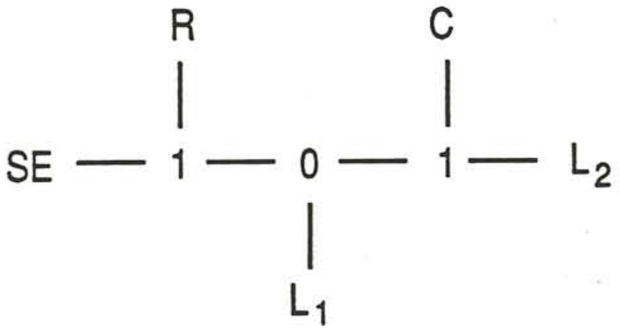


Figure 4.6 The reduced graph

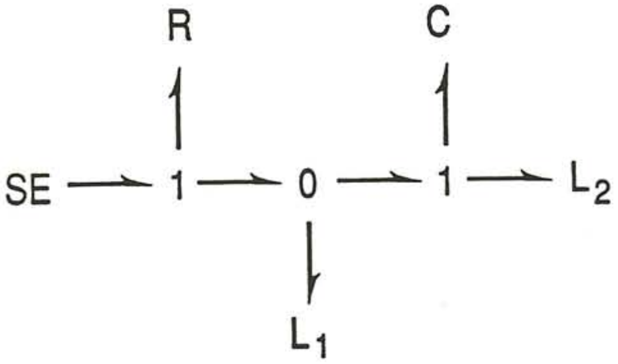


Figure 4.7 The bond graph

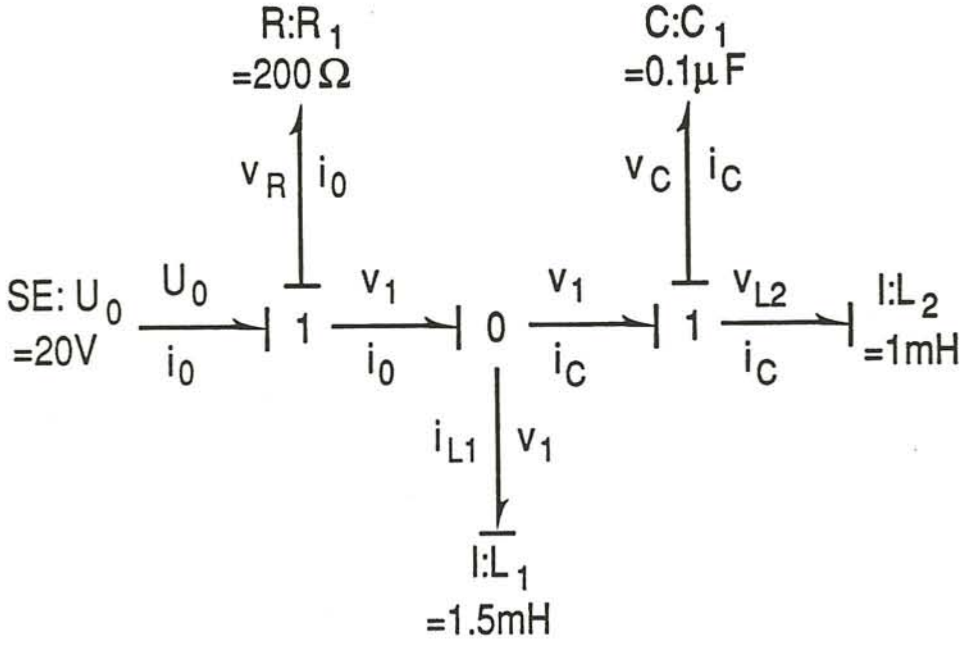


Figure 4.8 The completed bond graph with its causalities

### 4.4 Transformation of Bond Graphs into DYMOLA Code

After constructing the bond graph for the selected simple electrical network, we are ready to transform it into DYMOLA code which is a straightforward procedure. The following rules must be observed, however:

(1) The 0-junctions are equivalent to DYMOLA's "nodes."

(2) There is no DYMOLA equivalent for 1-junctions; however, if the effort and flow variables are interchanged, then they are the same as 0-junctions.

(3) Having the above in mind, a model type "bond" which simply exchanges the effort and flow variables can be created and installed in DYMOLA's library. Besides, the elements R, C, L, TF and GY which describe the basic bond graph multiport elements are installed once and for all in DYMOLA's library. They are illustrated in Figure 4.11.

(4) "In DYMOLA, all elements should be attached to 0-junctions only. If we want to attach an element to a 1-junction, then we need to place a bond in between" (Cellier, 1990a). The expanded bond graph is shown on Figure 4.9.

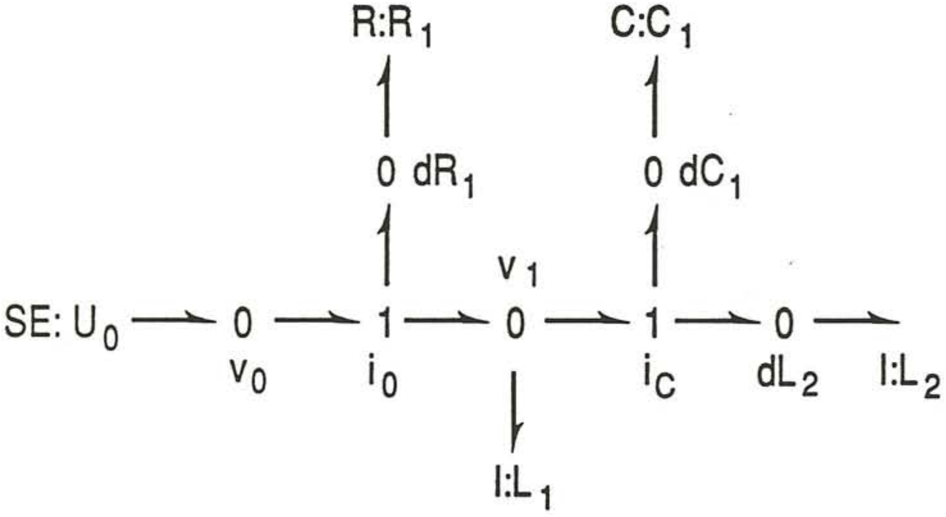


Figure 4.9 DYMOLA expanded bond graph with each node indicated

(5) Neighboring junctions are always of the opposite sex, i.e., 0-junctions and 1-junctions always toggle.

(6) Fortunately, we do not need to worry about causalities. DYMOLA is perfectly capable of handling the causalities as is seen during the execution of the algorithm assigning them. However, as we saw earlier, we were perfectly capable of assigning causalities. This is not true every time. For example, a non-causal system results when we try to connect two sources of different values.

Now we are ready to translate the expanded bond graph into DYMOLA code as indicated by Figure 4.10. The code is self-explanatory as we use the statements "submodel", "connect", and "node" which had been analyzed in the previous chapter. Furthermore, the various DYMOLA model types as well as the Experiment used for simulating the network are shown in the next two figures. Experiment is the simulation control model as described in chapter three.

DYMOLA can furthermore be used for obtaining various results such as causality, elimination of redundant equations, derivation of a state-space representation, and generation of a simulation program for DESIRE. The ultimate

```
{bond graph model for a simple RLC network}

@rlc.r
@rlc.c
@rlc.i
@rlc.se
@rlc.bnd

model RLC

   submodel (SE) U0
   submodel (R) R1(R=200.0)
   submodel (I) L1(I=1.5E-3),L2(I=1.0E-3)
   submodel (C) C1(C=0.1E-6)
   submodel (bond) B1,B2,B3,B4,B5,B6
   node v0,i0,v1,dR1,ic,dL2,dC1
   output y1

   connect U0 at v0
   connect B1 from v0 to i0
   connect B2 from i0 to dR1
   connect R1 at dR1
   connect B3 from i0 to v1
   connect L1 at v1
   connect B4 from v1 to ic
   connect B5 from ic to dC1
   connect C1 at dC1
   connect B6 from ic to dL2
   connect L2 at dL2

   U0.E0=20.0
   y1=L2.e
end
```

Figure 4.10  DYMOLA code of the bond graph shown on Figure 4.9

```
model type R
  cut A (e / f)
  parameter R=1.0
  R*f = e
end
```

```
model type I
  cut A (e / f)
  parameter I=1.0
  I*der(f) = e
end
```

```
model type C
  cut A (e / f)
  parameter C=1.0
  C*der(e) = f
end
```

```
model type SE
  cut A (e / .)
  terminal E0
  E0 = e
end
```

```
model type GY
     cut A(e1/f1) B(e2/ -f2)
     main cut C[A B]
     main path P<A - B>
     parameter r=1.0
     e1=r*f2
     e2=r*f1
end
```

```
model type TF
   cut A(e1/f1) B(e2/ -f2)
   main cut C[A B]
   main path P<A - B>
   parameter m=1.0
   e1=m*e2
   f2=m*f1
end
```

```
model type SF
    cut A(./ -f)
    terminal F0
    F0=f
end
```

```
model type bond
  cut A (x / y) B (y / -x)
  main cut C [A B]
  main path P <A - B>
end
```

Figure 4.11  The various basic DYMOLA model types

```
cmodel

  simutime 50.0E-6
  step 50.0E-9
  commupoints 101

  ctblock
    scale = 1
    XCCC = 1
    label TRY
      drunr | if XCCC<0 then XCCC = -XCCC | scale = 2*scale | go to TRY
                        else proceed
  ctend

  outblock
    OUT
    yl=L2$e
    dispt yl
  outend

end
```

Figure 4.12  Experiment used for the network

goal is the generation of an executable DESIRE program using the following commands:

```
$ dymola
> enter model
- @ rlc.dym
> enter experiment
- @ rlc.ctl
> outfile rlc.des
> partition eliminate
> output desire program
> stop
```

clarifying the last portions of the last chapter.

Then, we can run DESIRE using the following commands

```
$ desire
> load 'rlc.des'
> run
> bye
```

The generated DESIRE program as well as the state-space representation are shown in the following two figures respectively (see Figure 4.13 and Figure 4.14). The statements above the DYNAMIC declaration of the generated DESIRE program describe the experiment to be performed on the model, and the other statements describe the dynamic model. The time of the whole compilation is less than a tenth of a second. Finally, the DESIRE output of our network is shown in

```
----------------------------------------
-- CONTINUOUS SYSTEM RLC
----------------------------------------
-- STATE C1$e L1$f L2$f
-- DER dC1$e  dL1$f  dL2$f
-- OUTPUT y1
-- PARAMETERS and CONSTANTS:
R=200.0
C=0.1E-6
L1$I=1.5E-3
L2$I=1.0E-3
-- INITIAL VALUES OF STATES:
C1$e=0
L1$f=0
L2$f=0

----------------------------------------
TMAX=50.0E-6 | DT=50.0E-9 | NN=101
    scale = 1
    XCCC = 1
    label TRY
      drunr | if XCCC<0 then XCCC = -XCCC | scale = 2*scale | go to TRY
                        else proceed
----------------------------------------
DYNAMIC
----------------------------------------
-- Submodel: RLC
B3$x = L2$f + L1$f
-- Submodel: R1
R1$e = R*B3$x
-- Submodel: C1
d/dt C1$e = L2$f/C
-- Submodel: RLC
B1$x = 20.0
B4$x = B1$x - R1$e
```

Figure 4.13  Generated DESIRE Program
(continued on next page)

```
-- Submodel: L1
d/dt L1$f = B4$x/L1$I
-- Submodel: RLC
L2$e = B4$x - C1$e
-- Submodel: L2
d/dt L2$f = L2$e/L2$I
----------------------------------------
    OUT
    y1=L2$e
    dispt y1
----------------------------------------
/--
/PIC 'rlc.PRC           '
/--
```

Figure 4.13 Generated DESIRE program (continued)

```
RLC                 B3.x = L2.f + L1.f
R1                  e = R*B3.x
C1                  dere = L2.f/C
RLC                 B1.x = 20.0
                    B4.x = B1.x - R1.e
L1                  derf = B4.x/I
RLC                 L2.e = B4.x - C1.e
L2                  derf = e/I
```

Figure 4.14 State-space representation of the network

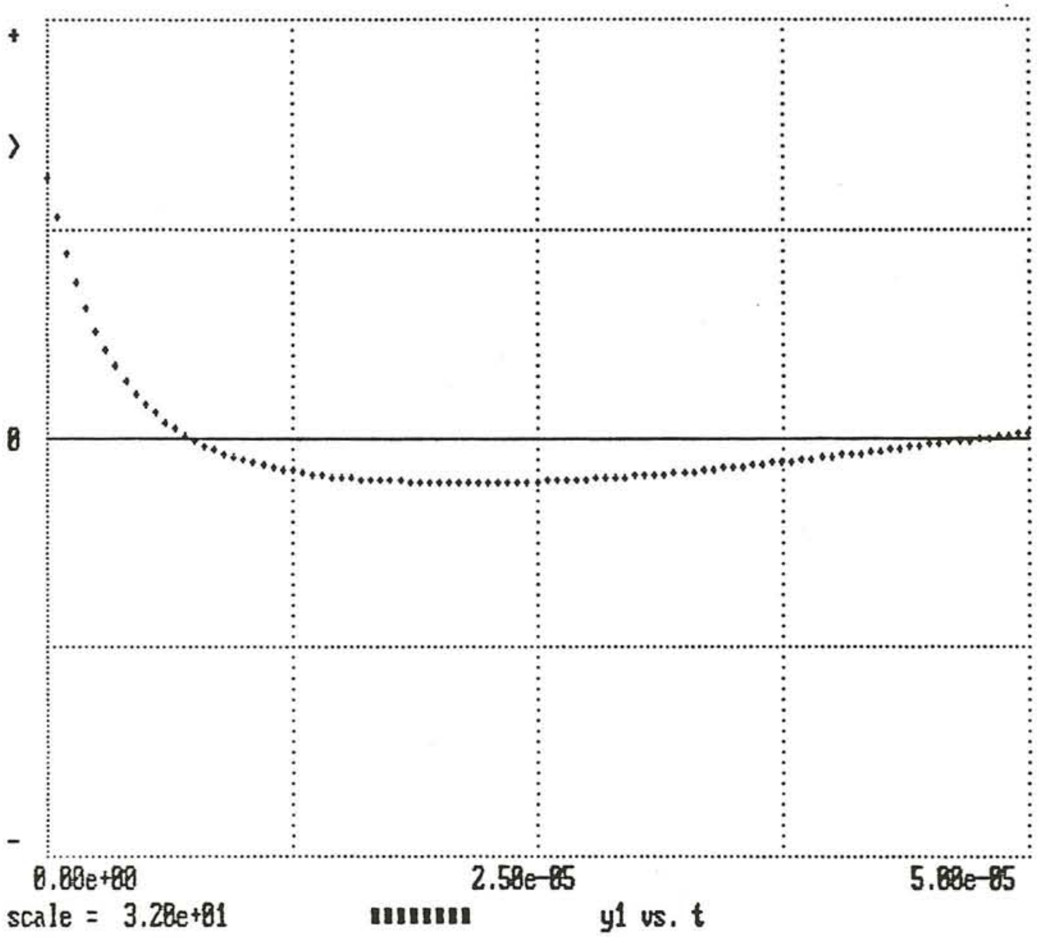


Figure 4.15 DESIRE output

the last figure of this chapter (Figure 4.15). Several other runs can be performed but the ones presented are the most important. For further details, the reader should consult the seventh chapter in Cellier's book (Cellier, 1990a).

The same procedure is used in the next chapter for simulating the solar heated house.

# CHAPTER 5

# CASE STUDY

# MODELING-SIMULATING A SOLAR-HEATED HOUSE

The main goal of this thesis is presented in this chapter. Having studied the bond graph methodology, DYMOLA, and seen how these two tools can be combined together, we are ready to model and then to simulate our solar-heated house. Being a relatively complicated system, it is appropriate for modeling purposes to divide it into several parts, that is, into a hierarchically described structure. Each part is presented by its bond graph converted into its DYMOLA code as well. Finally, all the parts are combined together resulting in the whole model of the solar-heated house.

## 5.1 Overview

Scientists throughout the years have investigated the exploitation of solar radiation for space heating. A solar heating system like the investigated one is any collection of equipment designed primarily to use the sun's energy for heating purposes.

The above system is a relatively complicated one involving many different types of energy. Various methods were used throughout the years for modeling and simulating such a system with mixed results. It is expected that using the method described in this thesis, that is, the bond graph modeling methodology as well as DYMOLA for generating a simulating program for DESIRE, the physical behavior of such systems can be modeled, simulated and evaluated in a convenient, robust, and fast manner.

The investigated configuration consists of a flat-plate solar collector, a solid body storage tank and the habitable space. They are connected with "water loops" circulating water through pipes. Each part is thoroughly studied and analyzed illustrating the energy flow through each subsystem and across the barrier between subsystems. Each one is transformed into a bond graph representation and is then directly coded into DYMOLA which not only generates a DESIRE program but can also provide us with a set of first order differential equations (a state-space representation). The various parameters used for the simulation were taken from various sources as mentioned before (see last portion of this chapter).

## 5.2 Solar Heating

A popular conception of solar heating is to use the solar radiation more or less directly without any natural intermediate steps such as photosynthesis. This can be primarily accomplished by collectors which are devices collecting solar radiation arriving from the sun and converting this radiant energy to a more desirable one such as heat. This converted energy can be transferred by a fluid (usually hot water) and either utilized immediately or stored for later use. This heat can be used for a simple space heating. A general solar heating system is shown in Figure 5.1.

Let us describe in general terms the collectors and the storage tank as well as the habitable space.

Collectors are the heart of any solar heating system, collecting and then converting the solar radiation. The simplest and cheapest one (see Figure 5.2 in section 5.4) is called the flat plate collector. It is a flat sheet of dark surfaced metal possessing one or more layers of glass above and a layer of common insulation below. The metal sheet is heated by sunlight which comes through the glass. The amount of heat that can escape and dissipate can be reduced by the glass and insulation; therefore, the metal sheet becomes very hot. In order to obtain this heat for utilizing it, there are two ways to do it. Either air can be passed above the metal or a fluid can be passed through tubes bonded to the metal. Therefore, the sunlight heats either air or water which are transferred to other convenient locations for use.

When the collector supplies the heated air or water, one of two things must be done--either it can be used at once

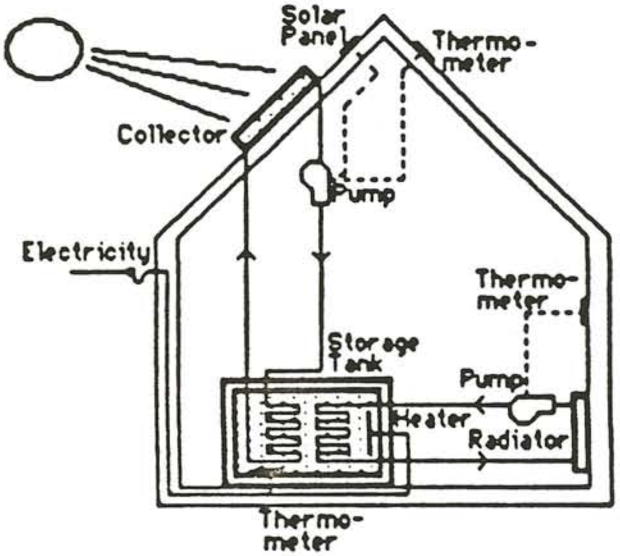


Figure 5.1 A solar heated house

or it can be stored for later use. Particularly, the hot water can be stored in tank systems designed in such a way that cooler water from the bottom can be sent through the collector for heating and then returned to the upper part of the tank. It is not practical to store hot air. The storage tanks are heavy and usually are set below ground.

Having heat either from collectors or the storage tank, we have to use it; for example, for space heating (habitable space). Hot water passes through a heat pump (might be cooling or heating device) and a heat exchanger in which the air blows around the hot water coils from the heat storage tank. Thereby, the habitable space is heated.

Above, the procedure has been described in which solar radiation is converted to a form of energy for heating a house. We are ready now to model the basic parts, that is the collector, the storage tank, the water loops (collector and heater water loop) and the habitable space. The collector water loop (CWL) is connected between the collector and the storage tank; whereas, the other one (heater water loop: HWL) is connected between the storage tank and the heater (heat exchanger). In modeling our solar house, we have been careful to make our system causal, that is, to satisfy all causality conditions. To start with, we have to know some basic thermodynamic concepts which are presented in the next section.

### 5.3 Basic Thermodynamic and General Concepts

The first thing needed is to define entropy and the first law of thermodynamics. So, entropy (S) is defined as

$$S = \frac{Q}{T} \qquad (5.1)$$

where Q is heat (in Joules) and T is temperature (in Kelvin).

The first law of thermodynamics states that the total energy $E_t$, being a constant, equals to the sum of free energy $E_f$ and the thermal energy Q.

$$E_t = E_f + Q \qquad (5.2)$$

Also entropy flow can be defined as

$$\frac{dS}{dt} = \frac{1}{T}\frac{dQ}{dt} \qquad (5.3)$$

and when multiplied by the temperature T gives heat flow which is power needed to construct the bond diagrams.

Moreover, the heat equation

$$\frac{\partial T}{\partial t} = \sigma \cdot \nabla^2 T \qquad (5.4)$$

describes both the thermal conductive and convective flow of heat.

In thermodynamics, we need to familiarize ourselves with three separate physical phenomena providing mechanisms for heat transfer or heat flow. They are <u>conduction</u>, <u>convection</u>, and <u>radiation</u>.

In heat conduction, thermal energy is transported by the interactions of its molecules in spite of the fact that molecules do not move themselves. For instance, when one end of a rod is heated, the lattice atoms in the heated end vibrate with greater energy than those at the cooler end so that this energy is transferred along the rod. In the case of a metal rod, the transport of thermal energy is aided by free electrons which are moving throughout the metal and they collide with the lattice atoms.

In convection, heat is transported by a direct mass transfer. For instance, warm air near the floor expands and rises because it possesses lower density. Thermal energy in this warm air is transferred from the floor to the ceiling along with the mass of warm air.

The last mechanism of heat transfer is through thermal radiation in which energy is emitted and absorbed by all bodies in the form of electromagnetic radiation. If a body is in thermal equilibrium with the environment, it emits and absorbs energy at the same rate. However, when it is warmed to a higher temperature than its environment, it radiates away more energy than it absorbs so that it cools down as the surroundings get warmer. As a result, in Arizona people avoid having dark painted cars because they emit/absorb light much more strongly than light ones.

As we saw in the third chapter, DYMOLA provides a modularized hierarchically structured model description. Thus, the entire solar house has been divided into five major hierarchical structures being the ones as mentioned before. Each of these consists of smaller hierarchical structures (submodels). For instance, the solar collector consists of the loss and the spiral submodels. Furthermore, the spiral comprises of two other smaller submodels being the heat exchanger and the one-dimensional cell. This hierarchy continues even further with the one-dimensional cell consisting of two other submodels, the modulated conductive source (mGS) and the modulated capacitance (mC). All these are described in detail later in this chapter.

All these hierarchical structures provide the researcher a convenient way to study the physical behavior of each particular part of the solar house in greater detail.

In the last portions of the previous section, the term causal was mentioned, i.e., to satisfy all causality conditions. To achieve this, we have to avoid the so-called algebraic loops and the structural singularities. This has been done by choosing the proper elements in assigning causalities and not to have any free choice as depicted in Figure 2.7 and by not overspecifying in the description of each particular model. Thus, our solar-heated house will be a causal system and will possess a uniquely determined causality.

Now we are ready to proceed with the modeling procedure starting in the next chapter.

### 5.4 Flat Plate Solar-Collector Modeling

We shall start modeling the entire solar house by modeling its flat plate collector. A simpler description than the previous given one for the solar collector is to imagine it as a black body accumulating solar heat through radiation so that the temperature raises inside. The collectors (may be one or several ones) are usually filled by air possessing a large heat capacity. Inside them, a winding water pipe goes back and forth between the two ends in order to maximize the pipe surface. Let's call this a water spiral. The heating of the water in the pipe occurs when a mostly conductive heat exchange takes place between the collector chamber and the water pipe. We shall describe the collector water loop as a pump which circulates the water from the collectors to the storage tank. As a result, the heat transfer occurs in a mostly convective manner. The water spirals can be connected either in parallel or in series and the pump is driven by a solar panel. The solar light is converted into electricity inside the panel. As a result, the pump circulates the water only on a sunny day, which is meaningful. Furthermore, the water pipe is protected by a freeze protection device which also switches the pump on when the temperature falls below 5°C outside.

A model depicting such a flat plate solar collector is shown in Figure 5.2. The efficiency of the solar collector

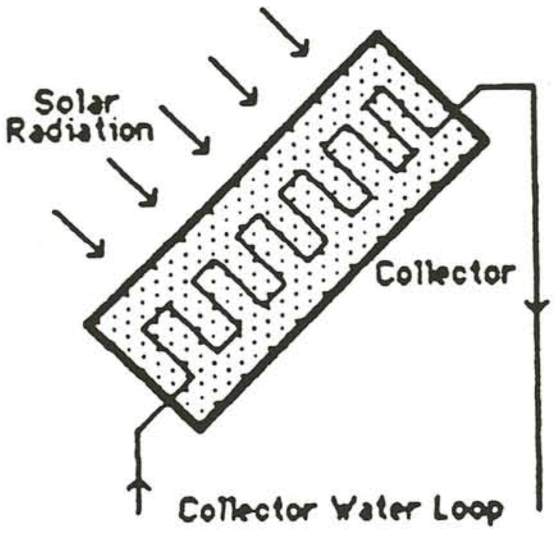


Figure 5.2  Model of a flat-plate collector

depends upon several factors such as climatic conditions (ambient temperature, wind), number of covers and their radiative properties, incident solar angle, radiative properties of absorber plate, spacing of covers and absorber, fluid type and insulation of collector enclosure. The following assumptions were made before modeling the collector:

(a) The heat flow into the collector is basically a radiative heat flow, modeled by a flow source which is dependent on three factors, day of the year, time of the day and weather.

(b) There is loss from the collector to the surroundings which has conductive, convective and radiative elements with the first two more dominant. It is basically modeled as a temperature source and as a modulated conductance characterized by the absorber and environment.

(c) There exists conductive heat exchange between the collector space and the hydraulic spiral. And finally,

(d) There is convective heat transport in the spiral.

The water spiral introduced previously will be represented as a series of one-dimensional cells. The bond diagram and the DYMOLA model type of such a cell are depicted in Figures 5.3a and 5.3b, respectively, with the causalities correctly marked. The mGS element is a modulated conductive source modulated with temperature and, furthermore, it is modulated with the water velocity in the pipe as shown in Figure 5.3c. This element (one-dimensional cell) has been modeled through its conductance rather than through its resistance because the conductance changes linearly with the water velocity.

The bond graph of the one-dimensional cell favors heat flow from the left to the right; therefore, it is not symmetrical. Our decision to represent the heat (entropy) flow in such a way is just an approximation. If, for example, the mGS element is split into two equal parts, one turning left and the other right, this choice is not desirable because of the introduction of algebraic loops identified with the choices of causalities (Cellier, 1990a).

The next step is to develop the heat exchanger model being used to describe the exchange of heat across the border of two media. In this particular case, the heat exchanger is used to model the heat transport from the collector chamber to the water spiral. Its bond diagram as well as its DYMOLA model type are shown in the following two figures (5.3d and 5.3e).

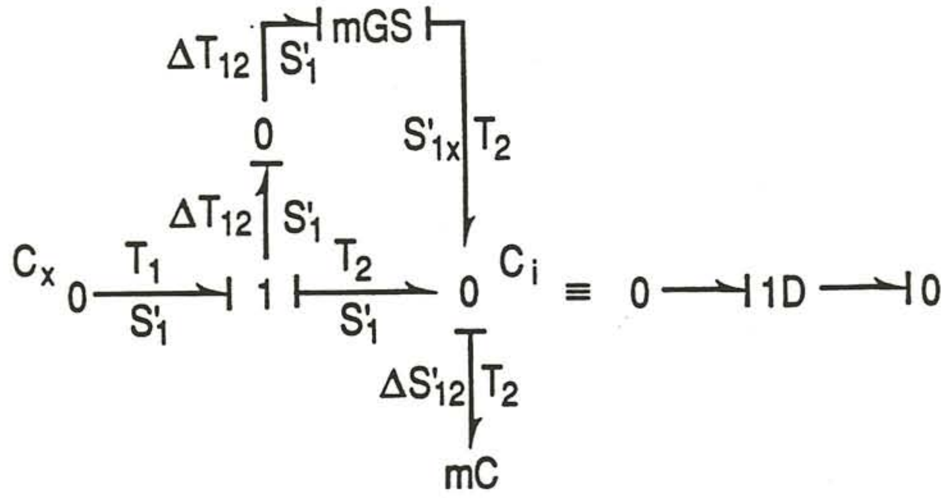


Figure 5.3a Bond diagram of a one-dimensional cell

```
{bond graph for one dimensional cell}

model type oneD
submodel(MGS) Gcell(a=1.5,b=0.72)
submodel(MC) Ccell(gamma=72.0)
submodel(bond) B1,B2,B3
node n1,n2

terminal vwater
cut Cx(ex/fx), Ci(ei/ -fi)
main path P<Cx - Ci>

connect B1 from Cx to n1
connect B2 from n1 to n2
connect Gcell from n2 to Ci
connect B3 from n1 to Ci
connect Ccell at Ci

Gcell.vel=vwater
end
```

Figure 5.3b DYMOLA model type of a one-dimensional cell

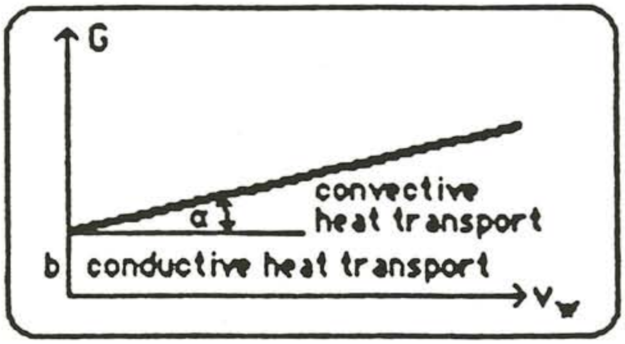


Figure 5.3c  Modulated conductive source

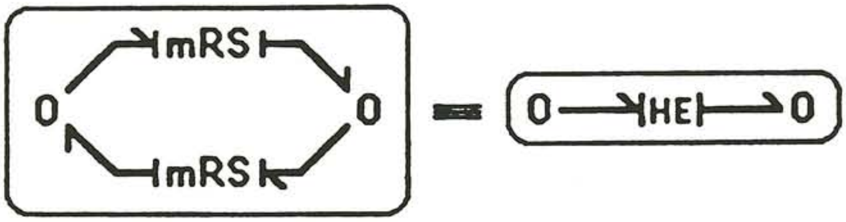


Figure 5.3d  Bond diagram of a heat exchanger

```
{bond graph for Heater(heat exchanger)}

model type HE
submodel(MRS) R1H(theta=8.0E+2),R2H(theta=8.0E+2)
cut A(e1/f1),B(e2/ -f2)
main cut C[A B]
main path P<A - B>
connect R1H from A to B
connect R2H from B to A
end
```

Figure 5.3e  DYMOLA model type of a heat exchanger

After modeling the one-dimensional cell model and the heater model, we are ready to proceed with the modeling of the water spiral which is a distributed parameter system. We have decided to represent the water spiral with three one-dimensional cells connected in series and heat exchangers attached in between. Obviously, our decision is an approximation of a process with distributed parameters. The bond graph of a water spiral which is a 3-port element is depicted next (see Figure 5.3f). Furthermore, its corresponding DYMOLA model type is shown in Figure 5.3g.

The final element to be developed for the complete collector model is the loss element from the collector chamber to the surroundings. This loss is partly conductive and partly convective and its bond diagram is shown in Figure 5.3h. It is a 1-port element. Its DYMOLA model type is also shown (Figure 5.3i). The effort source denotes the outside environment, whereas the mG element denotes the heat dissipation to the environment. The dissipated heat is proportional to the difference in temperatures between the inside and the outside. The mG element is a modulated conductance which is very similar to the mGS element found earlier. It is modeled with the temperature as well, but this time, the modulation is with respect to the wind velocity instead of the water velocity.

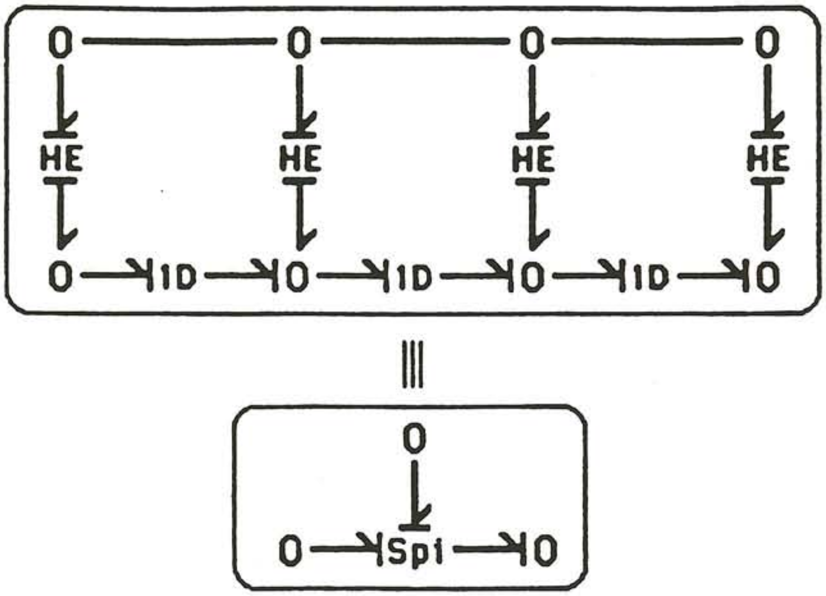


Figure 5.3f  Bond diagram of a water spiral

```
{bond graph Spiral}

model type Spi
submodel(HE) HE1,HE2,HE3,HE4
submodel(oneD) oneD1,oneD2,oneD3
node n1,n2
terminal vwater
cut inwater1(e1/f1),outwater1(e2/ -f2),C(e3/f3)
main cut D[inwater1 outwater1]
main path P<inwater1 - outwater1>

connect HE1 from C to inwater1
connect oneD1 from inwater1 to n1
connect HE2 from C to n1
connect oneD2 from n1 to n2
connect HE3 from C to n2
connect oneD3 from n2 to outwater1
connect HE4 from C to outwater1

oneD1.vwater=vwater
oneD2.vwater=vwater
oneD3.vwater=vwater

end
```

Figure 5.3g DYMOLA model type of a water spiral

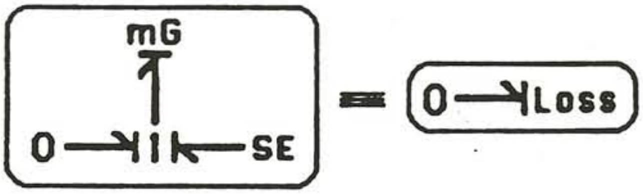


Figure 5.3h  Bond diagram of thermic loss

```
{bond graph for Loss}

model type Loss
submodel (SE) outtemp
submodel (MG) Gloss(a=1.5, b=0.72)
submodel (bond) B1,B2,B3
node n,nG,nS

main cut A(e/f)
terminal Tout, vwind

connect B1 from A to n
connect B2 from n to nG
connect Gloss at nG
connect B3 from n to nS
connect outtemp at nS

outtemp.E0 = Tout
Gloss.vel = vwind
Gloss.Tout = Tout
end
```

Figure 5.3i  DYMOLA model type of thermic loss

For the one-dimensional cell, the mGS element is used because the energy is not lost. The energy is simply transported right away to the next node. This is the reason that a new bond graph element called a resistive source (RS) has been introduced (Thoma, 1975). Obviously the GS element is 1/RS. On the other hand, the mG element is used in the loss because the behavior is like the electrical case where the resistances (conductances) dissipate heat and lose energy. Notice that in thermodynamics, the RS (R) and C elements are nonlinear. In DYMOLA, they are modeled by two new bond graph elements, mRS and mC, respectively.

The overall bond diagram for the collector can now be drawn as shown in the Figure 5.3j. The mC element which is modulated with temperature is the heat capacitance of the collector chamber. The SF element is the heat input from solar radiation which must be modeled separately.

We can use the hierarchical cut concept of DYMOLA to combine the two cuts into an aggregated bond graph representation pictorially represented by a double bond. The two cuts can be named as inwater and outwater and the hierarchical cut can be named as water. The disadvantage of the double bond representation is that causalities cannot be shown any longer.

Finally, the DYMOLA model type of the collector can be developed as depicted in the next figure (Figure 5.3k).

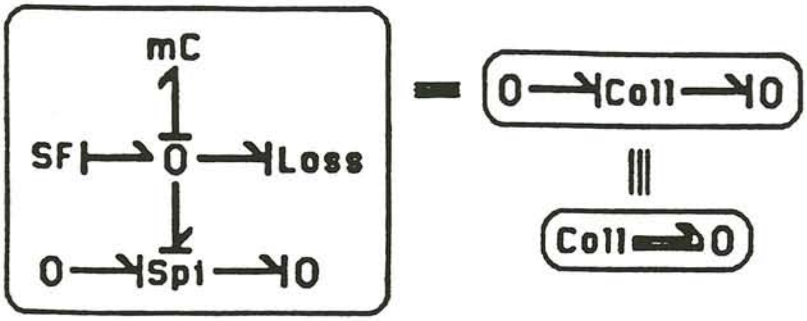


Figure 5.3j  Bond graph of the collector

```
{bond graph for Collector}

model type COLL
submodel(MC) Ccoll
submodel(SF) SDOT
submodel(Spi) CollSpiral
submodel(Loss) CollLoss
terminal S0,Tout,vwind,vwater

cut inwater(el/fl), outwater(e2/ -f2)
main cut water[inwater outwater]

connect SDOT at CollSpiral:C
connect Ccoll at CollSpiral:C
connect CollLoss at CollSpiral:C

SDOT.F0=S0
CollLoss.Tout = Tout
CollLoss.vwind = vwind
CollSpiral.vwater = vwater
end
```

Figure 5.3k  DYMOLA model type of the collector

In using the four elements, i.e., mC, mG, mGS and mRS, there are some physical concepts which must be mentioned. Cellier provides a very comprehensive analysis for these physical concepts and, therefore, it is used in this thesis (Cellier, 1990a).

We can write the capacity of a body transporting heat in a dissipative manner as

$$\Delta T = \theta \cdot \frac{dQ}{dt} = (\theta \cdot T) \frac{dS}{dt} \tag{5.5}$$

where

$\Delta T$ = temperature difference

$\theta$ = thermal resistance

S = entropy

Q = heat

The above equation looks like Ohm's law and it can be written also as

$$\Delta T = R \frac{dS}{dt}, \; R = \theta \cdot T \tag{5.6}$$

The thermal resistance can now be written

$$\theta = \left(\frac{1}{\lambda}\right) \cdot \left(\frac{\ell}{A}\right) \tag{5.7}$$

where

$\lambda$ = specific thermal conductance of the material

A = area of cross section

$\ell$ = length

The three elements mG, mGS and mRS, are modeled based on the above concepts and their DYMOLA model types are illustrated at the end of this section along with the mC element.

Now, the capacity of the body to store heat can be written as

$$\Delta Q' = \gamma \frac{dT}{dt} \tag{5.8}$$

where $\gamma$ = thermal capacitance.

The previous equation can also be written as:

$$\Delta S' = C \frac{dT}{dt} \tag{5.9}$$

where $C = \frac{\gamma}{T}$.

The terms "thermal resistance" and "thermal capacitance" are introduced because of the traditional relationship between temperature and heat although throughout this thesis entropy is used extensively.

Continuing, the thermal capacitance of a body can be described as

$$\gamma = c \cdot m \tag{5.10}$$

where

m = mass of the body

c = specific thermal capacitance of the material

Mass can further be written as

$$m = \rho \cdot V \tag{5.11}$$

where

$\rho$ = density

V = volume

and

$V = A \cdot dx$

Now from (5.10) we have

$$C = \frac{\gamma}{T} = \frac{c \cdot \rho \cdot A \cdot \Delta x}{T} \tag{5.12}$$

The time constant can now be determined:

$$T = R \cdot C = \theta \cdot \gamma = \frac{c \cdot \rho}{\lambda} (\Delta x)^2 \tag{5.13}$$

The last equation provides us with the capability of determining the dimensions of both the resistive and capacitive elements in our bond graph.

Let us illustrate the modeling by means of the i-th computational cell. The equations describing such a cell were developed to be:

$$\frac{dT_i}{dt} = \frac{1}{C} \Delta S'_i \tag{5.14}$$

$$\Delta T_i = T_{i-1} - T_i \tag{5.15}$$

$$S'_{i-1} = \frac{1}{R} \Delta T_i \tag{5.16}$$

$$S'_{ix} = S'_{i-1} \frac{\Delta T_i}{T_i} \tag{5.17}$$

$$\Delta S'_i = S'_{i-1} + S'_{ix} - S'_i \tag{5.18}$$

where $T_{i-1}$ is the temperature of the computational cell to the left and $S'_i$ is the entropy flow to the computational cell to

```
{bond graph modulated conductance}

model type MG
main cut A(e/f)
terminal vel,Tout
parameter a=1.0, b=1.0
local G, Gl
Gl = a*vel + b
G = Gl/Tout
G*e = f
end
```

Figure 5.31 DYMOLA model type of mG
(modulated with T and Vwind)

```
{bond graph conductance source for one dimensional cell}

model type MGS
cut A(el/fl),B(e2/-f2)
main cut C[A B]
main path P<A - B>
terminal vel
parameter a=1.0,b=1.0
local G,Gl
Gl=a*vel+b
G=f2*Gl
G*el=fl
fl*el=f2*e2
end
```

Figure 5.3m DYMOLA model type of mGS
(modulated with T and Vwater)

```
{ Bond Graph of a heat modulated resistive source }

model type MRS
  cut A(el/fl), B(e2/ -f2)
  main cut C[A B]
  main path P<A - B>
  parameter theta=1.0
  local R
  R = e2*theta
  R*fl = el
  el*fl = e2*f2
end
```

Figure 5.3n DYMOLA model type of mRS (modulated with T and $\theta$)

```
{ Bond Graph modulated capacitor/compliance }

model type MC
  cut A (e / f)
  parameter gamma=1.0
  local C
  C=gamma/e
  C*der(e) = f
end
```

Figure 5.3o DYMOLA model type of mC (modulated with T and $\gamma$)

```
{bond graph for a resistive source}

model type RS

  cut A(el/fl) B(e2/-f2)
  main cut C[A B]
  main path P<A - B>
  parameter R=1.0

  R*fl = el
  el*fl = e2*f2

end
```

Figure 5.3p DYMOLA model type of RS (electrical primary side)

the right (please see the bond diagram on Figure 5.3a to follow the above equations).

One clarification should be made which is the following: The RS elements may have both sides, primary and secondary, as thermal ones and they are modeled as shown in Figure 5.3n. On the other hand, if their primary side is electrical then their DYMOLA model type is different and is shown in Figure 5.3p. We are going to meet this case when designing the electrical backup device for the storage tank (see next section).

### 5.5 Heat Storage Tank Modeling

After the collector model was made available, the immediate next step is to model the heat storage tank.

Frequently, the storage tank is realized as a large and well insulated water heater. However, in order to model a water heater correctly, we must take the mixing thermodynamics into account. This makes the modeling procedure difficult. Therefore, a solid body storage tank was used together with another water spiral so that the water from the collector loop and from the heater loop never mix. One water spiral deposits heat in the storage tank, while the other picks it up again.

Inside the storage tank there is a second water spiral which represents the heater water loop picking up the heat from the storage tank. Whenever the storage tank temperature falls below a critical value, an installed electrical heater heats the storage tank electrically up to the minimum maintenance temperature.

Another pump drives the heater water loop and this pump is switched on whenever the room temperature falls below 20°C during the day or 18°C during the night. It is switched off whenever the room temperature raises beyond 22°C during the day or 20°C during the night.

Summarizing, the storage tank contains two water spirals, one belonging to the collector water loop and the other one belonging to the heater water loop. This is depicted in Figure 5.4a. Moreover, an electrical backup device is installed and it is turned on only if the temperature in the storage tank falls below its critical value. It is used only during evening hours when the price of electricity is lower. The overall bond diagram for the storage tank is shown in Figure 5.4b. The mC element represents the heat capacity of the storage tank, whereas the flow source together with the RS element denote the backup device. The primary side of this resistive source is electrical while the secondary side is thermic.

As we notice, the storage tank is a 4-port element. When writing its DYMOLA model type, we combine the cut inwater1 with the cut outwater1 to the hierarchical cut

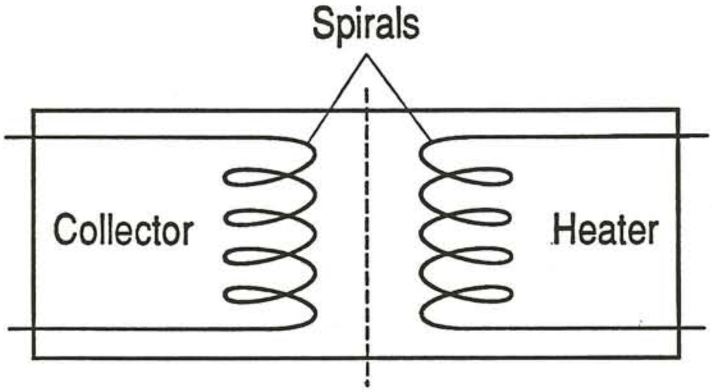


Figure 5.4a  The storage tank with the collector water loop and heater water loop

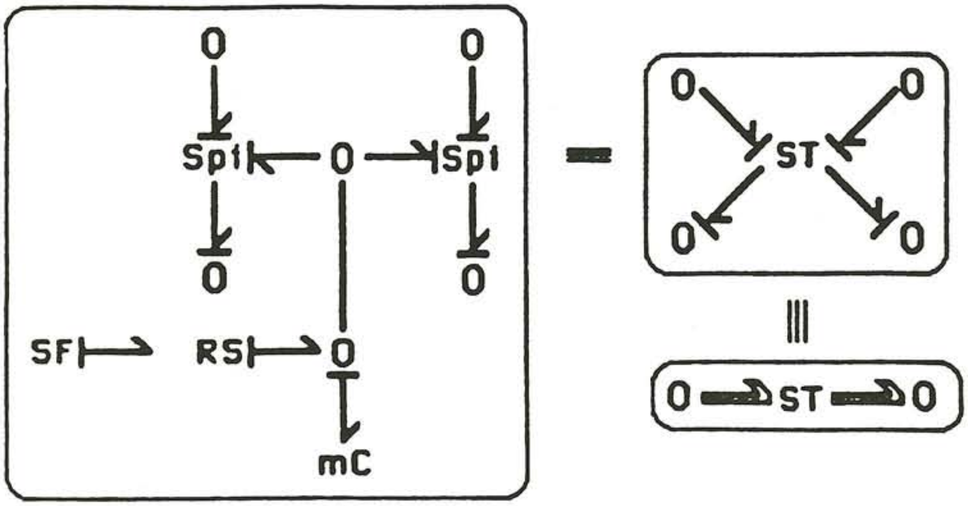


Figure 5.4b  Bond graph of the storage tank

inwater, whereas we combine the cut outwater2 with the cut inwater2 to the hierarchical cut outwater. By declaring a main path water, a logical bridge has been created from the hierarchical cut inwater to the hierarchical cut outwater. This is depicted in the following figure (Figure 5.4c).

## 5.6 Water Loop Modeling

An example of a convective mechanism in our solar house is the transport of the heat from the solar collector to the water heater connected by a pipe containing water. The water is circulated from the water heater (storage tank) to the collector and back by a pump.

We have already seen two water loops, the collector water loop and the heater water loop, and both are modeled exactly in the same way.

Each of the pipes is modeled by three one-dimensional cells connected in series as illustrated in the bond diagram (see Figure 5.5a). The one-dimensional cell has been developed previously. We shall assume that the pipes are thermically well insulated, that is, there is not any lost heat to the surroundings on the way. As shown, it is another 4-port element. In developing its DYMOLA model type, we shall combine the cut inwater1 with the cut outwater2 to the hierarchical cut inwater. In addition, we shall combine the cut outwater1 with the cut inwater2 to the hierarchical cut

```
{bond graph storage tank}

model type ST
submodel(SF) SDOT
submodel(RS) R1(R=10.0)
submodel(MC) Ctank(gamma=9.0E+4)
submodel(Spi) Spitank1,Spitank2
terminal S0,vwater

cut inwater1(e1/f1), outwater1(e2/ -f2)
cut inwater2(e3/f3), outwater2(e4/ -f4)
main cut inwater[inwater1 outwater1]
main cut outwater[outwater2 inwater2]
main path water<inwater - outwater>

connect Spitank1:C at Spitank2:C
connect Ctank at Spitank1:C
connect R1 from SDOT to Spitank1:C
SDOT.F0=S0
Spitank1.vwater = vwater
Spitank2.vwater = vwater
end
```

Figure 5.4c  DYMOLA model type of the storage tank

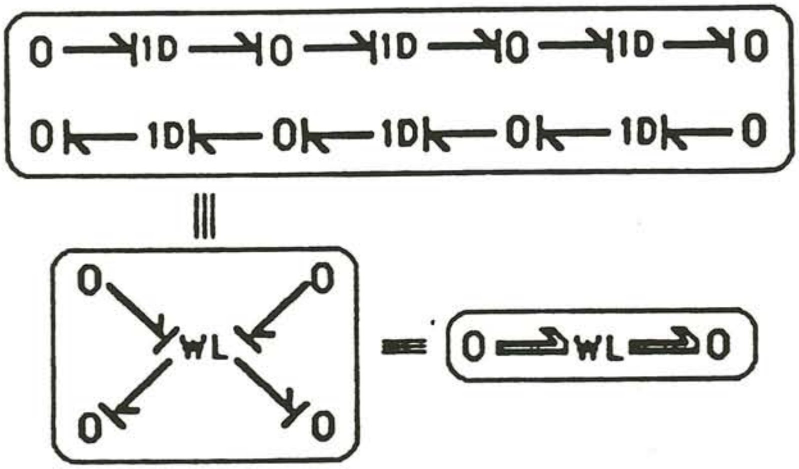


Figure 5.5a  Bond diagram for water loop

```
{bond graph water loop (heater+collector)}

model type WL
submodel(oneD) oneD1,oneD2,oneD3,oneD4,oneD5,oneD6
node n1,n2,n3,n4
terminal vwater

cut inwater1(e1/f1), outwater1(e2/ -f2)
cut inwater2(e3/f3), outwater2(e4/ -f4)
main cut inwater[inwater1 outwater2]
main cut outwater[inwater2 outwater1]
main path water<inwater - outwater>

connect oneD1 from inwater1 to n1
connect oneD2 from n1 to n2
connect oneD3 from n2 to outwater1
connect oneD4 from inwater2 to n3
connect oneD5 from n3 to n4
connect oneD6 from n4 to outwater2

oneD1.vwater=vwater
oneD2.vwater=vwater
oneD3.vwater=vwater
oneD4.vwater=vwater
oneD5.vwater=vwater
oneD6.vwater=vwater
 end
```

Figure 5.5b  DYMOLA model type for water loop

outwater. Moreover, we declare a main path water creating a logical bridge from the hierarchical cut inwater to the hierarchical cut outwater (see Figure 5.5b).

Assuming that there is no air in the pipe and the water in it is totally incompressible, several implicit physical simplifications can be taken into consideration. This leads to the conclusion that the water flow via the whole pipe has a constant velocity $V_w$. The hydraulic flow is expressed in $m^3s^{-1}$ denoted by $\Phi_v$ and the volume of water in a one-dimensional cell is $V = A \cdot \Delta x$. Thus, the amount of entropy leaving the i-th cell per time unit to the right is given by

$$S'_{i\ \text{out}} = \Delta S_i \ \frac{\Phi_v}{V} \tag{5.19}$$

which can also be written as

$$S'_{i\ \text{out}} = \left( C \cdot \frac{\Phi_v}{V} \right) T_i \tag{5.20}$$

During the same time, a similar amount of heat is transported into the cell from its left neighbor being

$$S'_{i\ \text{in}} = \left( C \cdot \frac{\Phi_v}{V} \right) T_{i-1} \tag{5.21}$$

Combing the previous two equations, we conclude that

$$S'_{i\ \text{conv}} = G_{\text{conv}} \ \Delta T_i$$

and

$$G_{\text{conv}} = C \frac{\Phi_v}{V} \tag{5.22}$$

Consequently, convection is simply a second convective resistance being connected in parallel with the conductive resistance. Therefore, convection augments the thermal conductivity.

As mentioned before, several simplifications were made. In reality, there is friction between the liquid and the wall, and friction within the liquid. We would notice that in this case the liquid flows faster at the center of the pipe and slower near the wall. The hydraulic friction is a dissipative process producing more heat and thereby more entropy sources should be applied to the thermal unit.

Furthermore, if the assumption of incompressibility were not made, then the whole situation would be much more complicated. In this case, we would need to take into consideration the pneumatic process besides the thermal process. The pneumatic process generates a time- and space-dependent flow rate $\Phi_v(t,x)$ which can be used to modulate the convective resistance of the thermal unit. Overall, the whole situation becomes very involved.

### 5.7 Habitable Space Modeling

It was decided that the habitable space (house) would be a cube with dimensions 10m x 10m x 10m (see Figure 5.6c), mainly for reasons of simplicity.

However, before starting to model the house, a three-dimensional cell as well as a two-dimensional cell have been developed. The concept of a one-dimensional cell which has been developed previously can be extended to the two- and three-dimensional case. So, let us assume that each three-dimensional cell consists of one capacitor and three resistors, one to its left, one to its front and one below as depicted in Figure 5.6a together with its bond diagram. It can also be seen from the figure that the necessity to attach every element to 0-junctions for DYMOLA modeling has been taken into consideration. The DYMOLA model type "CEL" describing the three-dimensional cell has been developed as well and it is depicted in Figure 5.6b. The two-dimensional case consists of two resistors (xy, yz, xz directions) and a capacitor.

We shall assume that the entire house consists of one room only and that a single large radiator is used for heating purposes. The radiator exchanges heat with the room in a partly conductive and partly convective manner. It is attached to the left wall of the house somewhere close to the floor so that the heat input takes place at the left low outside center three-dimensional cell. Because the dimensions of the radiator are much smaller than those of the house, a decision was made not to model the radiator. Therefore, the outwater1 of the heater water loop has been simply connected with the inwater2 of the heater water loop. Moreover, another heat exchanger has been attached at this node with the responsibility of exchanging the heat between the heater water loop and the house.

From Figure 5.6c, it can be seen that the house has 27 nodes. At these nodes one-, two- and three-dimensional cells are placed with the exception of the first node. So, nodes 2 and 3 are one-dimensional in x direction, 4 and 7 are one-dimensional in y direction, 10 and 19 are one-dimensional in z direction. The two-dimensional cells are located at nodes 5, 6, 8, 9 (xy direction), 11, 12, 20, 21 (xz direction) and 13, 16, 22, 25 (yz direction). Finally three-dimensional cells are placed at the remaining nodes, i.e., 14, 15, 17, 18, 23, 24, 26, 27. These nodes are connected through their paths accordingly as depicted in Figure 5.6d. We need also to place our heat source and one additional capacitor at the first node which is actually the Cx point of the second node. The house loses heat through the four walls and through the roof, but not through the floor so that at nodes 5 and 14 there are no losses. The previously developed loss elements are attached to each of the 0-junctions as appropriate. In the case of a cell adjacent to two or three outside walls, we attach one combined loss element to the corresponding node only because of the emergence of algebraic loops. Therefore, nodes 2, 4, 6, 8, 11, 13, 15, 17, 23 have

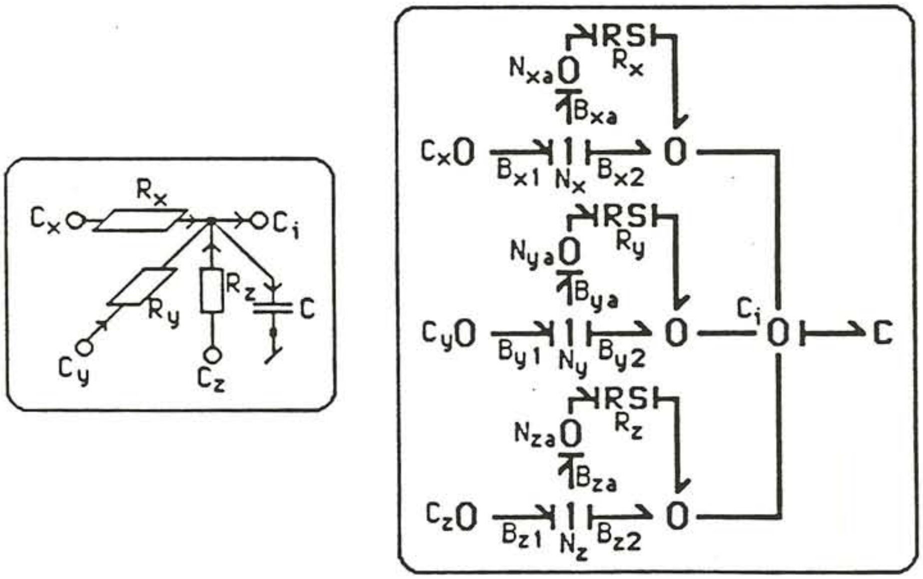


Figure 5.6a Three-dimensional diffusion cell
(RS elements are actually mRS elements)

```
{bond graph for a three dimensional cell}

model type CEL
   submodel (MRS) Rx(theta=0.5),Ry(theta=0.5),->
                  Rz(theta=0.5)
   submodel (MC) C(gamma=152310.0)
   submodel (bond) Bx1,Bx2,Bxa,By1,By2,Bya,Bz1,Bz2,Bza
   node Nx,Nxa,Ny,Nya,Nz,Nza

   cut Cx(ex/fx), Cy(ey/fy), Cz(ez/fz), Ci(ei/-fi)
   path Px<Cx - Ci>, Py<Cy - Ci>, Pz<Cz - Ci>

   connect Bx1 from Cx to Nx
   connect By1 from Cy to Ny
   connect Bz1 from Cz to Nz
   connect Bx2 from Nx to Ci
   connect By2 from Ny to Ci
   connect Bz2 from Nz to Ci
   connect Bxa from Nx to Nxa
   connect Bya from Ny to Nya
   connect Bza from Nz to Nza
   connect Rx from Nxa to Ci
   connect Ry from Nya to Ci
   connect Rz from Nza to Ci
   connect C at Ci
end
```

Figure 5.6b  DYMOLA model type of a three-dimensional cell

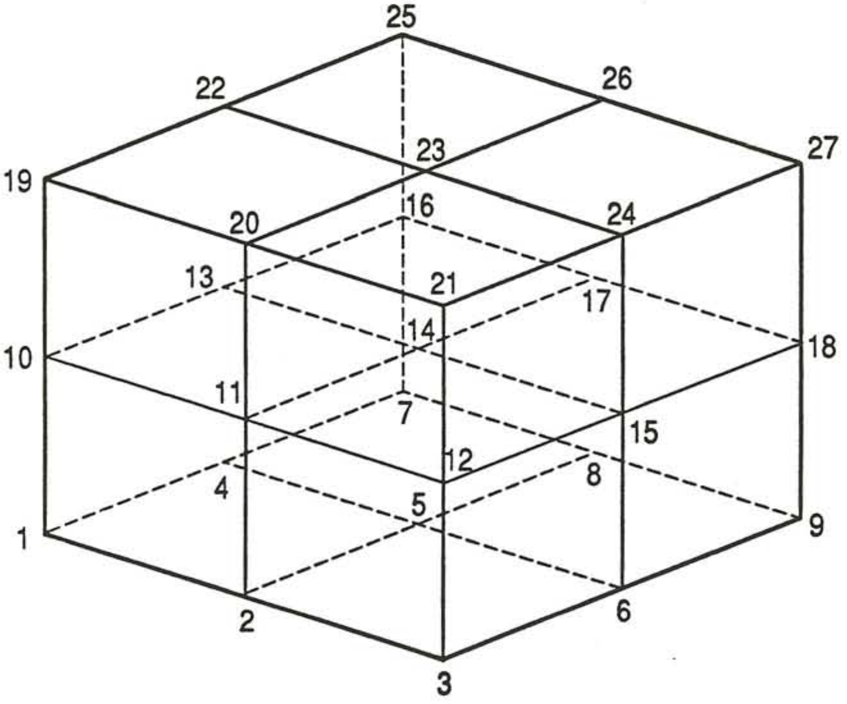


Figure 5.6c The house room represented as a 10x10x10 cube

```
model SPACE
submodel(SF) Tsp
submodel(MC) C(gamma=152310.0)
submodel(LS1) l2,l4,l6,l8,l11,l13,l15,l17,l23
submodel(LS2) l1,l3,l7,l9,l10,l12,l16,l18,l20,l22,l24,l26
submodel(LS3) l19,l21,l25,l27
submodel(CXD) d2,d3
submodel(CYD) d4,d7
submodel(CZD) d10,d19
submodel(XYC) d5,d6,d8,d9
submodel(XZC) d11,d12,d20,d21
submodel(YZC) d13,d16,d22,d25
submodel(CEL) d14,d15,d17,d18,d23,d24,d26,d27

input Tout, vwind, S0
output y1

connect (Px) d2-d3,d5-d6,d8-d9,->
             d11-d12,d14-d15,d17-d18,->
             d20-d21,d23-d24,d26-d27

connect (Py) d4-d7,d5-d8,d6-d9,->
             d13-d16,d14-d17,d15-d18,->
             d22-d25,d23-d26,d24-d27
connect (Pz) d10-d19,d11-d20,d12-d21,->
             d13-d22,d14-d23,d15-d24,->
             d16-d25,d17-d26,d18-d27

connect l1 at d2:Cx
connect l2 at d2:Ci
connect l3 at d3:Ci
connect l4 at d4:Ci
connect l6 at d6:Ci
connect l7 at d7:Ci
connect l8 at d8:Ci
connect l9 at d9:Ci
connect l10 at d10:Ci
connect l11 at d11:Ci
connect l12 at d12:Ci
connect l13 at d13:Ci
connect l15 at d15:Ci
connect l16 at d16:Ci
connect l17 at d17:Ci
connect l18 at d18:Ci
```

Figure 5.6d  DYMOLA model type of the SPACE (house)

```
connect l19 at d19:Ci
connect l20 at d20:Ci
connect l21 at d21:Ci
connect l22 at d22:Ci
connect l23 at d23:Ci
connect l24 at d24:Ci
connect l25 at d25:Ci
connect l26 at d26:Ci
connect l27 at d27:Ci

connect Tsp at d2:Cx
connect C at d2:Cx

connect d5:Cx at d4:Ci
connect d8:Cx at d7:Ci
connect d11:Cx at d10:Ci
connect d14:Cx at d13:Ci
connect d17:Cx at d16:Ci
connect d20:Cx at d19:Ci
connect d23:Cx at d22:Ci
connect d26:Cx at d25:Ci
connect d4:Cy at d2:Cx
connect d5:Cy at d2:Ci
connect d6:Cy at d3:Ci
connect d13:Cy at d10:Ci
connect d14:Cy at d11:Ci
connect d15:Cy at d12:Ci
connect d22:Cy at d19:Ci
connect d23:Cy at d20:Ci
connect d24:Cy at d21:Ci
connect d10:Cz at d2:Cx
connect d11:Cz at d2:Ci
connect d12:Cz at d3:Ci
connect d13:Cz at d4:Ci
connect d14:Cz at d5:Ci
connect d15:Cz at d6:Ci
connect d16:Cz at d7:Ci
connect d17:Cz at d8:Ci
connect d18:Cz at d9:Ci

l1.Tout = Tout
l2.Tout = Tout
l3.Tout = Tout
l4.Tout = Tout
l6.Tout = Tout
l7.Tout = Tout
l8.Tout = Tout
```

Figure 5.6d  (Continued)

```
l9.Tout = Tout
l10.Tout = Tout
l11.Tout = Tout
l12.Tout = Tout
l13.Tout = Tout
l15.Tout = Tout
l16.Tout = Tout
l17.Tout = Tout
l18.Tout = Tout
l19.Tout = Tout
l20.Tout = Tout
l21.Tout = Tout
l22.Tout = Tout
l23.Tout = Tout
l24.Tout = Tout
l25.Tout = Tout
l26.Tout = Tout
l27.Tout = Tout

l1.vwind = vwind
l2.vwind = vwind
l3.vwind = vwind
l4.vwind = vwind
l6.vwind = vwind
l7.vwind = vwind
l8.vwind = vwind
l9.vwind = vwind
l10.vwind = vwind
l11.vwind = vwind
l12.vwind = vwind
l13.vwind = vwind
l15.vwind = vwind
l16.vwind = vwind
l17.vwind = vwind
l18.vwind = vwind
l19.vwind = vwind
l20.vwind = vwind
l21.vwind = vwind
l22.vwind = vwind
l23.vwind = vwind
l24.vwind = vwind
l25.vwind = vwind
l26.vwind = vwind
l27.vwind = vwind

Tsp.F0 = S0
yl = d7.ei

end
```

Figure 5.6d (Continued)

a 1*MG loss element, nodes 1, 3, 7, 9, 10, 12, 16, 18, 20, 22, 24, 26 have a 2*MG loss element and nodes 19, 21, 25, 27 have a 3*MG loss element. (The reader might go back to Figure 5.3h and see the loss element.) The corresponding DYMOLA model type for the house has been named SPACE and it is depicted in Figure 5.6d.

This concludes the modeling of all the parts of the solar house.

### 5.8 The Total Solar-Heated House

The overall system is a series connection of the previously presented aggregated bond graph elements, that is, the collector, the collector water loop, the storage tank, the heater water loop, the heat exchanger and the house. This is depicted in Figure 5.7.

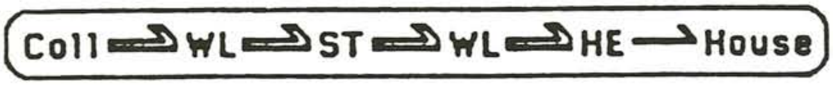


Figure 5.7 Aggregated bond graph of the overall system

## 5.9 Choosing Appropriate Parameters for Analyzing the Effectiveness of Our System

Until now, the modeling of the total solar house has been discussed. Now we are ready to test the versatility of DYMOLA to describe such a complex physical system after it has been modeled by the bond graph methodology.

The best approach to simulate is to start from the habitable space (house itself) as indicated in the total aggregated figure. We shall imagine that there is an arbitrary heat source heating the house, assuming that the initial temperature inside the house is 15°C (288°K). Our goal is to determine the time that it takes for the temperature to reach its steady-state value in various locations inside the house.

The DYMOLA model type SPACE (model SPACE in this case because the house is our main system now) has been used to generate the DESIRE program. The same procedure as described in Chapter 4 has been used. However, the PC was unable to generate the DESIRE program because it was exceeding its memory (heap) capability.

Therefore, a decision was made to use the VAX. Using the VAX, we were able to generate a SIMNON program (currently, the VAX version does not provide a DESIRE program generation capability yet). However, the conversion of the SIMNON program into a DESIRE program is not a difficult procedure. After doing that, we are ready to simulate the house (habitable space).

The following elements must be calculated before proceeding: the modulated resistive source (theta), the modulated capacitance (gamma) being inside the one-, two- and three-dimensional cells. These values will be the same everywhere in the house. Moreover, the modulated conductance in the loss elements (a and b) must be evaluated. Please refer to Figures 5.3n, 5.3o and 5.3l where the parameter values for these elements are specified in parentheses.

A logical and economical heat source (entropy source) is found to be 20 J/K. Our intuition was based on an average monthly utility bill that people spend for heating their house during winter time.

It is found that the a, b and theta parameter values affect the overall heating of the house. Figure 5.3c helps us to find a, b. The angle must be kept small around 25° and b is approximately one-half of the tangent of that angle. These a and b values determine how well the house is insulated. Moreover, the value of theta depends on the air inside the house. Formula 5.7 determines the value of theta but the physical constant $\lambda$ for air is not reliable. We have used some flexibility in deciding the value of theta which is about 0.5 sec.K/J.

Finally, the value of gamma can be found using formula 5.12 and it is found to be 152310 J/K (not to be confused with entropy).

Having found all the necessary parameters, the simulation can now be performed. We simulated the house, and displayed the temperature in the vicinity of the heat source, and also at farther away nodes. The farthest one, node 27, caused the most problems and did not give satisfactory results.

Various results are in the following table: (see Appendix A for graphs with y1, y2, y3, y4 and y5 corresponding to nodes 16, 20, 22, 9 and 26 respectively. Notice that y2 is the same as y3 because of symmetry of the nodes, therefore only one of their graphs is shown).

| Node # | $T_{steady-state}$ (°C) | Time (sec) |
|---|---|---|
| 3 | 77 | 1.40E+5 |
| 9 | 24 | 2.00E+5 |
| 14 | 63 | 1.50E+5 |
| 16 | 38 | 1.50E+5 |
| 17 | 29 | 1.80E+5 |
| 20 | 32 | 1.50E+5 |
| 22 | 32 | 1.50E+5 |
| 26 | 15 | 1.80E+5 |
| 27 | 10 | 1.70E+5 |

Figure 5.8 Table of some results

The outside temperature was assumed to be 0°C.

From the above results, we can observe that the temperature in most of the nodes of the house is not consistent. In the vicinity of the heat source, the temperatures are very high, whereas in the farthest nodes of the house, the temperatures are low. This makes us believe that the heat dissipation through the house (e.g. by means of convection) is not modeled correctly.

The temperature at every node reaches its steady-state value in a little over a day. This makes sense. It takes a long time to heat the house to its steady-state temperature with an economical heater such as the one we used.

The next step was to add the heater water loop and constant temperature source at the storage tank to produce the 20 J/K entropy (heat) source. Nevertheless, combining all the DYMOLA model types together, the whole program will become very large so we decided to stop the simulation analysis. It is true, however, that by having a computer with enough heap (memory) that can handle such large programs, the whole simulation analysis can be performed until we reach the collector. At the end, we will have a very large DYMOLA program with all the hierarchical structures of the solar house connected together.

## CHAPTER 6

## CONCLUSION

This thesis touches on a modern and advanced modeling-simulation technique applied to a large and complex physical system--the solar heated house.

The bond graph modeling methodology has been studied extensively as well as a software tool called DYMOLA designed to implement bond graphs. How well they work together was demonstrated in Chapter 4.

Bond graphs were successful in providing us with a complete and easily comprehensible model of the solar house, a relatively complicated system. Furthermore, DYMOLA proved to be a suitable software tool for implementing the hierarchical bond graphs encountered in the system. Both tools, like SPICE, can be combined together for studying the behavior of several less complex systems such as electrical and mechanical ones.

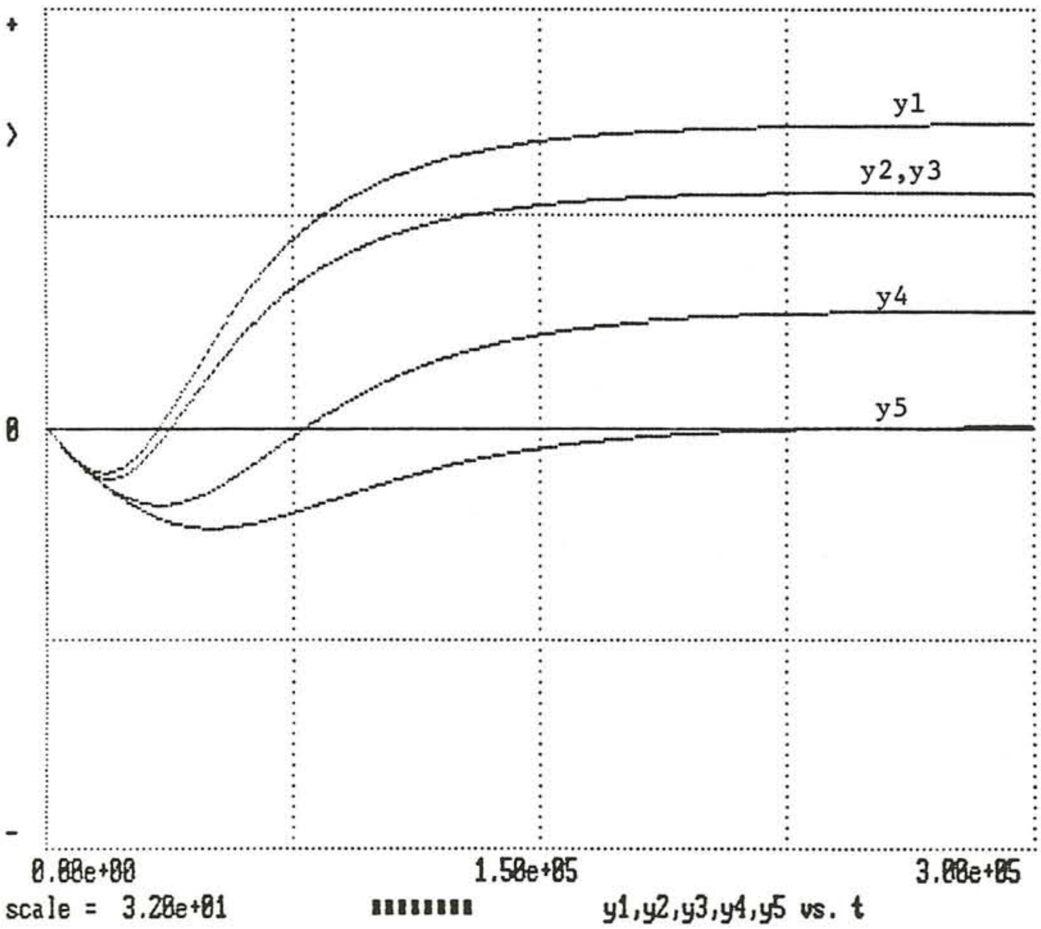
On the negative aspect, bond graphs as they are developed today, are not suitable for modeling distributed parameter systems in several space dimensions. As all other graphical techniques, bond graphs become clumsy when applied to distributed parameter problems in more than one space dimension.

There are many opportunities for research. Both bond graphs and DYMOLA can be further developed so that the study of complex systems can become more feasible and attractive to the researcher.

This thesis has provided new insight into the process of modeling complex physical systems. For the first time, the bond graph modeling technique was expanded to hierarchical model descriptions. It has been shown that the general purpose continuous-system modeling language DYMOLA can be effectively used to describe hierarchical nonlinear bond graphs.

# APPENDIX

## GRAPHS



Simulation results at various nodes

## REFERENCES

Blundell, Alan (1982), "Bond Graphs for Modeling Engineering Systems," Ellis Horwood Publishers, Chichester, United Kingdom, and Halsted Press, New York.

Cellier, Francois E. (1990a), "Continuous System Modeling and Simulation, Volume 1: Continuous System Modeling," Springer Verlag, New York.

Cellier, Francois E. (1990b), "Hierarchical Non-Linear Bond Graphs: A unified methodology for modeling complex physical systems," Proc. European Simulation Multiconference, SCS Publishing.

Coppas, C. H. (1975), "Simulation of Residential Solar Heating Systems," MS Thesis, Dept. of Electrical Engineering, University of Toledo, August 1975.

Deffie, John A. and Bechman, William A. (1980), "Solar Engineering of Thermal Processes," John Wiley, New York.

Elmqvist, Hilding (1975), "SIMNON - An Interactive Simulation Program for Non-Linear Systems - User's Manual," MS Thesis, Report CODEN:LUTFD2/(TFRT-7502), Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Elmqvist, Hilding (1978), "A Structured Model Language for Large Continuous Systems," PhD Dissertation, Report CODEN:LUTFD2/(TRFT-1015), Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Granda, J. Jose (1982), "Computer Aided Modeling Program (CAMP): A Bond Graph Preprocessor for Computer Aided Design and Simulation of Physical Systems Using Digital Simulation Languages," PhD Dissertation, Dept. of Mechanical Engineering, University of California, Davis.

Howell, John R., Bannerot, Richard B., and Vliet, Gary C. (1985), "Solar-Thermal Energy Systems," McGraw-Hill Book Company, New York.

Karnopp, Dean C. and Rosenberg, Ronald C. (1975), "Systems Dynamics, A Unified Approach," John Wiley, New York.

Karnopp, Dean C. and Rosenberg, Ronald C. (1972), "A Definition of the Bond Graph Language," Trans. ASME, Journal of Dynamic Systems, Measurement, and Control, September 1972, pp. 179-182.

Kass, Pierre (1978), "Simulation of a Solar House," MS Thesis, Dept. of Automatic Control, Swiss Federal Institute of Technology Zurich, December 1978.

Korn, Granino A. (1989a), "NEUNET/DESIRE User's Manual," 6801 E. Opata Rd., Tucson, AZ 85715.

Korn, Granino A. (1989b), "Interactive Dynamic-System Simulation," McGraw-Hill, New York.

Paynter, Henry M. (1961), "Analysis and Design of Engineering Systems," M.I.T. Press, Cambridge, MA.

RosenCode Associates, Inc. (1989), "The ENPORT Reference Manual," 200 North Capitol Building, Lansing, Michigan 48933.

Tipler, Paul A. (1982), "Physics," Volume 1, Worth Publishers Inc., New York.

Thoma, Jean U. (1975), "Entropy and Mass Flow for Energy Conversion," J. Franklin Institute, 299(2), pp. 89-96.

van Dixhoorn, J. Jan (1982), "Bond Graphs and the Challenge of a Unified Modelling Theory of Physical Systems," in: Progress in Modelling and Simulation (F. E. Cellier, ed.), Academic Press, London, pp. 207-245.

Wang, Qingsu (1989), "Management of Continuous System Models in DEVS-SCHEME: Time Windows for Event Based Control," MS Thesis, Dept. of Electrical and Computer Engineering, University of Arizona, December 1989.