Metamodelling

Prentice Hall International
Series in Systems and Control Engineering

M. J. Grimble, Series Editor

BAGCHI, A., *Optimal Control of Stochastic Systems*
BENNETT, S., *Real-time Computer Control: An introduction,* second edition
BENNET, B. S., *Simulation Fundamentals*
BROWN, M. and HARRIS, C. J., *Neurofuzzy Adaptive Modelling and Control*
COOK, P. A., *Nonlinear Dynamical Systems,* second edition
GAWTHROP, P. and SMITH, L., *Metamodelling: For bond graphs and dynamic systems*
GRIMBLE, M. J., *Robust Industrial Control*
ISERMANN, R., LACHMANN, K. H. and MATKO, D., *Adaptive Control Systems*
KUČERA, V., *Analysis and Design of Discrete Linear Control Systems*
MARTIN SANCHEZ, J. M. and RODELLAR, J., *Adaptive Predictive Control*
MARTINS DE CARVALHO, J. L., *Dynamical Systems and Automatic Control*
MATKO, D., ZUPANČIČ, B. and KARBA, R., *Simulation and Modelling of Continuous Systems: A Case Study Approach*
OLSSON, G. and PIANI, G., *Computer Systems for Automation and Control*
ÖZGÜLER, A. B., *Linear Multichannel Control*
PARKS, P. C. and HAHN, V., *Stability Theory*
SABERI, A., SANNUTI, P. AND CHEN, B., *H_2 Optimal Control*
SÖDERSTRÖM, T. D., *Discrete-time Stochastic Systems*
SÖDERSTRÖM, T. D. and STOICA, P., *System Identification*
WATANABE, K., *Adaptive Estimation and Control*

Metamodelling: For bond graphs and dynamic systems

Peter Gawthrop
Lorcan Smith

Prentice Hall
London New York Toronto Sydney Tokyo Singapore
Madrid Mexico City Munich

First published 1996 by
Prentice Hall Intemational (UK) Limited
Campus 400, Maylands Avenue
Hemel Hempstead
Hertfordshire, HP2 7EZ
A division of
Simon & Schuster Intemational Group

Printed and bound in Great Britain by
Hartnolls Limited, Bodmin, Comwall

Library of Congress Cataloging-in-Publication Data

Gawthrop, Peter.
Metamodelling: Bond graphs and dynamic systems /
Peter gawthrop and Lorcan Smith
p. cm. – (Prentice Hall intemational series in systems and
control engineering)
Includes bibliographical references and index
ISBN 0-13-489824-9
1. Mathematical models. 2. Computer simulation. I. Smith,
Lorcan. II. Title. III. Series
IV. Series
TA342.G39 1995
003–dc20 95–31857
CIP

British Library Cataloguing in Publication Data

A catalogue record for this book is available
from the British Library

ISBN 0-13-489824-9

1 2 3 4 5 00 99 98 97 96

Contents

Preface

With the increasing complexity of processes to be analysed, the modern control engineer often needs to develop a model of the system to be controlled. However, in many cases, there is limited time for detailed system analysis, and the engineer may not be an expert in that particular system domain. This book is aimed at graduate engineers (and postgraduate students) who wish to use a systematic approach to model development that is suited to computer-aided modelling environments.

The goal of this book is to support the use of modelling as a useful knowledge-enhancing exercise, and to propose corresponding modelling methodologies. The motivation for this is the widespread use of models in analysing and simulating systems for safe and cost-effective evaluation of new processes. The context is primarily that of control system design, due to the extensive use of models of the process, and its disturbances, in modern design methods.

We use the term metamodelling to describe the approach taken; i.e. a modelling methodology which transcends the accepted mathematical models for specific applications. This methodology abstracts general models from first principles, by employing an existing notation (bond graphs) as a metalanguage for describing physical systems. This book is, therefore, concerned with separating out the model development process from the functions for which the model is developed, in order to enhance understanding of the essentials of the real physical systems.

This book is organised in two parts, so that the reader may first understand the motivation and the basic concepts, and then have the proposed methodology illustrated by a variety of examples covering a wide selection of applications.

The first part describes general modelling principles, based on system decomposition, first using classical dynamical analysis and then via the energy bond graph notation. Bond graphs are shown to provide a powerful core model representation from which a variety of mathematical models may be derived. Bond graphs provide a useful means of illustrating causality which is shown to be a crucial aspect of system modelling.

The second part uses specific case studies to illustrate the application of this methodology to systematic generation of the most widely used mathematical models. Reference is made to a computer-aided modelling tool (MTT), which is a research modelling toolbox which uses bond graphs to support the modelling of dynamic processes.

Acknowledgements

Some of the work described here was supported by the Engineering and Physical Sciences Research Council. In particular, some of the ideas arose as part of the following grants:

- Feedback Control of Multi-drug Anaesthesia Using Qualitative and Quantitative Measurements (GR/F 57816)
- The Engineering Design Research Centre
- Metamodelling of Robots (GR/F 57779)
- Process Simulation Integration and Control(GR/F 07965)
- Physical-Model-Based Control (GR/F 57779)

In the course of writing this book we have had the pleasure of discussing ideas and concepts with many people, all of whom have contributed in some measure to refining the material contained herein. We thank our friends and colleagues at Glasgow and elsewhere in the world, including: Mohamed Abderrahim, John Asbury, Donald Ballance, Des Costello, Marisol Delgado, Bill Gray, John Howell, Dean Karnopp, Peter Larcombe, Derek Linkens, Richard Jones, Sheena Mackenzie, Yasmine Mather, Neil Marrison, Jack Ponton, David Roberts, Jean Thoma, George Worship and Arthur Whittaker. However all errors and misconceptions are our own responsibility.

We would like to express our thanks to our respective families: Rosemary, Katharine, Sylvia, Joni and Alexis for their support and encouragement. Some early drafts were completed by Peter Gawthrop at his parents' home by the sea in Cumbria; and he wishes to express his gratitude to them for providing such a stimulating environment for work.

Thanks are also due to Christopher Glennie and the team at Prentice Hall for their tolerance and encouragment thus ensuring that this work was completed.

1

Introduction

SUMMARY

This chapter gives a general overview of the concepts of modelling dynamic systems. The chapter has four sections:

- 1.1 *What this book is about.* This will tell you whether the book is going to be of interest to you, and its overall layout.
- 1.2 *Rationales for modelling.* Highlights the variety of reasons why models are developed and used.
- 1.3 *A motivational example.* This section takes a common industrial process as an example showing how models can be used, while indicating some functional and structural requirements for a modelling tool.
- 1.4 *Computer-based modelling tools.* A discussion of the need for computer-based modelling tools with reference to a prototype tool box: MTT – Model Transformation Tools.

1.1 WHAT THIS BOOK IS ABOUT

The main aim of this book is to support the use of modelling as a useful knowledge-enhancing exercise, and to propose corresponding modelling methodologies. As a result, the book is concerned with separating out the model development process from the functions for which the model is developed. We use the term *metamodelling* to emphasise that we are abstracting and describing the thought processes (and corresponding computer-based tools) which lie behind developing specific models of specific systems. Thus we are concerned to abstract the essentials of modelling and thus move attention away from the details of generating specific mathematical models or simulations towards an understanding of the essentials of modelling physical systems in general.

The *bond graph* notation was introduced by (Paynter, 1961); its principles and application have been developed since that time and have been expounded in a number of textbooks including (Karnopp and Rosenberg, 1975) (Thoma, 1975) (Wellstead, 1979) (Rosenberg and Karnopp, 1983) (Karnopp *et al.*, 1990) (Thoma, 1990) (Cellier, 1991). We have chosen bond graphs to describe systems and to act as the core model description for computer-based modelling. We hope to convince the reader that this is a good choice.

The book is divided into two parts:

- Part I: general modelling principles,
- Part II: specific modelling applications

Part II illustrates the wide range of physical domains that can be captured by the bond graph approach.

- Part I: Principles
  - Chapter 1: *Introduction.* This chapter. Discusses why models are needed and, using an example industrial process, develops a requirements specification for a modelling tool.
  - Chapter 2: *Representation of Elementary Systems.* The decomposition of a system into a structure linking elements representing its static and dynamic behaviour is reviewed, first via classical dynamical analysis and then via the energy *bond graph* notation. Bond graphs are shown to provide a powerful core model representation from which a variety of mathematical models may be derived. This chapter provides the basic ideas of bond graphs and, in so doing, motivates the more detailed work of the remainder of the book.
  - Chapter 3: *Causality.* Causality is discussed with particular reference to computational causality. The application of bond graphs to causality analysis is detailed. Although causality may, at first, appear to be an abstract notion, this chapter argues that causality is a crucial aspect of system modelling. Links to related areas such as constraint programming and qualitative modelling are drawn.
  - Chapter 4: *Derived models.* The use of computers to aid modelling is a central theme of this book. This chapter discusses the twin issues of *representation* and *transformation.* In particular, model transformations from the core (bond graph) representation to various derived mathematical models (such as differential-algebraic equation, non-linear state-space, linearised state-space, frequency response etc.) are given and illustrated.
  - Chapter 5: *System approximation.* The art of modelling is, to a large extent, the art of abstracting the simplest model for the required purpose. Chapter 5 shows how a bond-graph methodology for system approximation can aid the system modeller.

- Chapter 6: *System inversion.* System inverses are of intrinsic interest as well as relevant to the design of control systems. This chapter shows how to obtain the bond-graph of an inverse system from the bond-graph of the system itself.

- Part II: Modelling Applications
  - Chapter 8: *Process engineering.* A systematic approach to modelling process systems is developed and illustrated using a progression of examples. The use of systematic *approximation* is emphasised.
  - Chapter 7: *An extrusion process.* The process of insulating copper wire using a plasticating extruder, described in this introduction, is modelled using the hierarchical bond graph approach.
  - Chapter 9: *Pharmacokinetics.* Models for inhaled drug uptake, with particular relevance to anaesthesia, are derived based on physical principles encapsulated in bond graphs.
  - Chapter 10: *Mechanical systems and robotics.* Bond graphs are used to model the dynamics of a two-dimensional mechanical link. This basic building block is used to systematically create dynamic models for a number of simple systems including a pendulum, a double pendulum and a two-link manipulator. This process is repeated for three-dimensional systems, resulting in models for robotic manipulators, including the PUMA and Stanford arm architectures.
  - Chapter 11: *Control Systems.* Applications of modelling to control are given. In particular, the use of models in generating physical model-based controllers is emphasised.

1.2 RATIONALES FOR MODELLING

Models are normally constructed in order to solve a problem or, at least to test a proposed solution to a problem. A systems analysis view of modelling has been proposed (Schmidt, 1985), in which modelling is shown to be a significant part of the systems analysis process:

1. problem identification,
2. specification of objectives,
3. definition of the system,
4. model formulation,
5. model verification and validation,
6. model implementation,
7. model use,
8. solution identification,

9. solution implementation,
10. model revalidation.

In his paper, Schmidt acknowledges that not all problems warrant all these steps, whereas others may require several iterations between steps. For some problems a simple mental model of the system is sufficient to resolve the problem, while other more difficult problems may best be solved by more detailed modelling, but the time or skills are not available for this.

Schmidt's paper also categorises models into two types – those whose purpose is descriptive, and those which are prescriptive. Descriptive models have the function of aiding understanding, or are developed for communication of concepts. Common formats for such models are engineering documentation, including drawings, and scale models.

Prescriptive models are used to recommend a course of action, since they permit predictions of the real system behaviour to be made. Typical model formats to achieve this end are simulation models, and those used for experimentation and parameter optimisation. Simulation models alone have a variety of uses, not least of which are education and training. Mathematical models suited to specialised analysis tools may also be included in this category. An important function for mathematical models is control design, for which a large variety of tools are available – frequency domain analysis, stability and eigenvalue analysis all depend on different formulations of the system model.

In general we can see that, in the non-academic world at least, modelling is only performed if the risk and cost of failure outweighs the cost of building models and running appropriate experiments. The way forward is to provide tools which support and accelerate the model building and experimenting processes.

1.3 A MOTIVATIONAL EXAMPLE

In this section we show, by example, how a modelling tool must offer a range of functions in order to meet a variety of application requirements. The example used in this discussion is an industrial process for extruding polymer sheathing onto wire for manufacturing electrical cables (Figure 1.1). This process is analysed in greater detail in Part II (Chapter 7) of this book, but it is useful to consider at this point, in order to understand the problems in modelling such a process.

For the moment, it is sufficient to know that a plasticating extruder is a large metal barrel in which a screw rotates in order to meter out quantities of molten polymer through a die. The screw is typically driven by an electric (D.C.) motor which provides the mechanical energy necessary to overcome the shear friction against the polymer and generate sufficient hydraulic pressure to force the polymer through a filter at entrance to the die. The polymer is initially heated by electrical heater bands round the barrel, but when it is being extruded at normal production

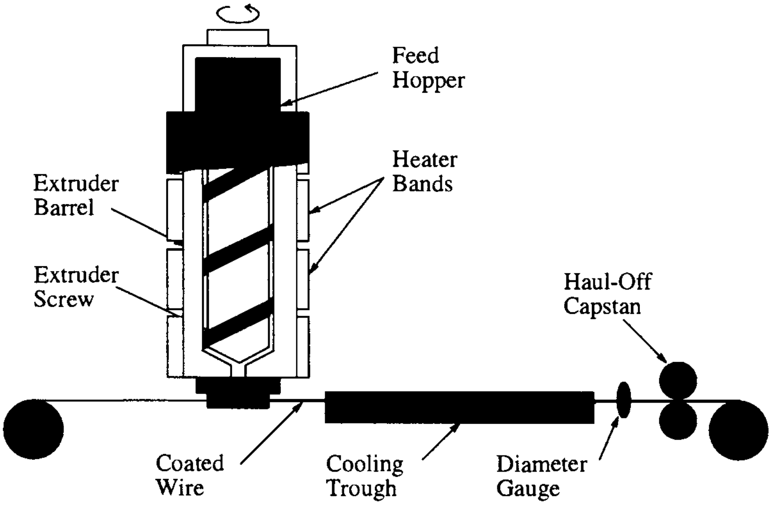


Figure 1.1 A plastic-on-wire extruder

rates, sufficient work heat is generated by the shear friction of the screw forcing the melt down the barrel and out of the die. Finally there are measurement systems on the extruder – measuring temperature and pressure – and also on the final product – measuring the outer diameter of the cable after it has been hauled through a cooling trough. This last measurement system is of greatest interest as it gives the main measure of product quality, although the measurement is subject to a long transport delay due to the cooling process.

Figure 1.1 is, in fact, our first model of the process and is well suited to the purpose of describing the process at an overview level. It is graphical and encapsulates the description in a very concise and understandable manner, but it also has some major disadvantages. In the first place, the drawing does not explicitly show all the sub-systems – the mechanical translation of the polymer through the barrel, and the associated hydraulics are assumed. The model is not complete and had to be supplemented by the written description in the above paragraph. Most important to the engineer, however, is the fact that even the combination of the figure and the written description is insufficient for any analysis or prediction of the performance of the process. The engineer needs some form of mathematical model to achieve these ends.

If our process engineer's purpose for modelling is just to achieve a relationship between the outer diameter of the coated cable and the screw speed or the haul-off speed, then he must find the static transfer function of the process. This is achieved by deriving a mass balance equation for the polymer flow into and out of the die. Intuitively one is not surprised to find that this transfer function shows that the diameter depends on the internal dimensions of the barrel and the screw, and on the ratio (screw speed/haul-off speed).

This transfer function is very useful if the engineer wants to judge the rate at which he can produce a given diameter of cable, but it has limited use if he wishes to design an automatic control system for this parameter. The problem is that this mathematical model only gives the steady-state gain of the process, whereas the dynamic transfer function is a more useful model for control design. In practice, some of the variables are often ignored at this stage in order to simplify the modelling exercise, but at the expense of reducing its usefulness in achieving an overall understanding of the process. A typical simplification is based on the fact that the temperature of the barrel wall is closely controlled by a multi-zone automatic control system. It is assumed that the melt temperature is approximately constant, or, at least, varies slowly with respect to the achievable changes in screw speed or line speed. An important feature lost by this assumption is the ability to predict the response of the diameter to large-scale changes in screw speed when the process ramps up to full speed and the generation of work heat changes rapidly.

It is important to be able to model the process behaviour during ramp-up to production speed (and ramp-down), because the diameter variation caused by this disturbance can mean that significant amounts of cable have to be scrapped. For this analysis, a simulation proves an invaluable tool, and, since the entire process forms rather a large model it is desirable to neglect some of the faster dynamics in order to run the simulation faster. In this case we require a mixed model which includes the dynamics of the slower sub-systems, and static models of the fast sub-systems.

The above discussion has shown that three different modelling requirements have resulted in three different mathematical models to provide each specific functionality. Modellers are not unused to this sort of problem, but it may explain why the benefits of process modelling are not as widely exploited in industry as they might be. The problem in industry is that the processes are subject to continuous change as market demands, financial constraints, and technology all change. The process engineer often cannot afford the time to generate more than the static model let alone keep several models up to date.

There is, therefore, a very strong incentive to provide one core model representation from which the variety of mathematical models described in the prece-ding paragraphs can automatically be generated. This is the aim of the book.

1.4 COMPUTER-BASED MODELLING TOOLS

In the context of software, it has been said that one good tool is worth many packages. UNIX is a good example of this philosophy: the user can put together applications from a range of ready made tools.

A recent paper Gawthrop (1995) describes the application of this philosophy to dynamic system modelling embodied in MTT – a set of Model Transformation Tools each of which implements a single transformation between system representations.

System representations have two attributes: a form: e.g. acausal bond graph, differential algebraic, linear state-space etc. a language: e.g. Fig, Matlab, LaTeX, Reduce, postscript etc.

Transformations are accomplished using appropriate software (e.g. Prolog, Reduce) encapsulated in UNIX Bourne shell scripts. The relationships between the tools are encoded in a Make File; thus the user can specify a final representation and all the necessary intermediate transformations are automatically generated.

Many of the equations and graphs appearing in this book have been automatically generated from the bond-graph representation using MTT. The theory behind MTT is given in Chapter 4.

Part I

Principles

2

Bond graph representation of elementary systems

SUMMARY

- The basic concepts of a generalised approach to modelling are discussed.
- Energy bond graphs are introduced as a generalised modelling technique which unifies models across energy domains.
- The concept of causality is introduced and shown to provide a systematic method of deriving mathematical models from bond graph models.

2.1 INTRODUCTION

Our aim in this chapter is to detail a generalised approach to modelling, which unifies physical systems of all energy domains. Thus, the emphasis in this chapter is on *physical* system modelling. A structured approach is to analyse the system in terms of its constituent parts, within a defined system boundary (a frame). This process requires the modeller to abstract the model to a structure of interacting sub-models in a hierarchical manner until at the lowest level each sub-model consists of a structure of elementary component behaviours (expressed as constitutive relations).

It is not our intention in this chapter to repeat the excellent text books already available in this area (for example those by Karnopp *et al.* (1990), Cellier (1991), Thoma (1990) and Wellstead (1979)) but rather to provide a short motivating introduction. A deeper discussion of some of the concepts is given in Chapter 3.

In this chapter, Section 2.2 describes an appropriate set of structural and constitutive relations for the primitive elements, while Section 2.3 describes how energy bond graphs provide a powerful notation for linking these elements and hence representing models using these concepts. Having captured such a representation of the system, it is necessary to transform the representation to a derived mathematical model suitable for analysis or simulation. Section 2.5 shows how various causal augmentations of bond graphs permit this to be systematically achieved, whilst providing deeper insights into the model and system. Section 2.6 describes the use of multi-port components, and Section 2.7 applies pseudo bond graphs to solve modelling problems for non-energy systems. The chapter is reviewed in Section 2.8.

2.2 STRUCTURE AND CONSTITUTIVE RELATIONS

As discussed in Chapter 1, the core model representation should include both the static and dynamic characteristics of the process. It should not be a set of mathematical equations, but should instead have a close mapping to the physical process, permitting the model to be extended to track modifications to this process. A natural way to achieve this aim is to subdivide the model into a set of standard elements and interconnect them in a structure appropriate to the process. This separation of structure and component behaviour is essential in order to permit the model to be interpreted easily by both humans and computers, thus facilitating modification in step with that of the process.

A popular method of modelling is to construct an electrical analogue of the actual process. A brief analysis of why this is the case may prove useful. Electrical schematics are quite concise, and unambiguously describe the structure (wiring) relating a set of idealised components - this energy domain is fortunate in having components that are close to ideal over a wide operating range. The schematic has the advantage of being easily understood by (trained) humans, and also, more recently, by CAD software. Unfortunately, the mapping between the electrical analogue and the process is not always one to one, so occasionally some confusion may arise. A more direct mapping also permits the modeller to evolve the model more easily to achieve a closer match to the real process. Another disadvantage of this circuit-based modelling approach is that it does not offer any direct insights into the workings of the real process, since it is purely an analogue. As discussed by Karnopp, mechanical systems in two or three dimensions are modelled using state-modulated transformers; these are not found in electrical systems.

Modelling using electrical analogues also tends to obscure the fact that, for processes covering multiple energy domains, the unifying variable is in fact energy. Much has been written by previous researchers in this field including Paynter (1961), Rosenberg and Karnopp (1983) and Wellstead (1979), exploiting this unification, which we can only summarise here. However, modelling energy transfers does provide a very useful focus for this part of our discussion of system representations. In practice, this turns out not to be a significant limitation, as most of the processes we are interested in modelling - general physical systems, mechanics and industrial processes - involve energy transfers. In later sections we will describe how the same techniques can be applied to developing models of processes where energy is not the exchange variable.

2.2.1 Energy transfer models

At this point it is necessary to give an overview of the basic concepts of system modelling based on *energy* as the variable manipulated by the system. For a more detailed exposition, we refer the reader to several excellent texts on this specific subject including those by MacFarlane (1964), Karnopp and Rosenberg (1975) and Wellstead (1979).

Choosing energy as the exchange variable for a model leads naturally to the use of two co-variables in each energy domain, which are conventionally called effort (e) and flow (f), where

$$\text{energy} \quad E = \int ef\, dt \tag{2.1}$$

At this point, it is worth commenting that an alternative pair of co-variables is in common use: the across and through variables. Across variables (transvariables) are spatially-extensive and are often described as those requiring a 2-point measurement (MacFarlane, 1964). Through variables (pervariables) are spatially-intensive and imply that the variable passes through the measurement instrument. This way of classifying variables results in voltage, pressure and velocity being grouped as across variables, while current, flow rate and force are the corresponding through variables. A discussion of the relative merits of the two approaches is given by Wellstead (1979).

In the effort-flow classification, voltage, pressure and force are effort variables, while current, flow rate and velocity are the corresponding flow variables. The consequence of this difference is that mechanical systems described using the effort-flow notation are duals of those using across-through notation. Each approach shows some inconsistencies, but since the effort-flow classification is most widely used in the context of bond graphs, this convention is adopted henceforth in this book.

Energy is exchanged through so-called *ports* on each element, where each port represents a single distinct energy interface. The energy model has four basic types of ideal elements:

Energy sources. Energy sources provide the system inputs which are a convenient way of defining the boundary of the modelled system, and hence for determining its reaction to effort or flow stimuli. The concept of an energy source includes that of an energy sink which can be regarded as a source with negative flow of energy.

Energy sources are ideal in the sense that either the effort or the flow variable is independent of the co-variable.

Energy stores. These elements accumulate either the effort or flow variable and are described as effort-accumulating or flow-accumulating stores, respectively. This accumulation (integration) of either effort or flow gives the system a state, and thus endows the system with dynamics.

Energy stores are ideal in the sense that they store, but do not dissipate, energy.

Energy dissipators. Energy dissipators are non-dynamic elements which dump energy out of the system into its environment, and which, for non-thermodynamic models, provide a convenient termination boundary to the model. This irreversible conversion of energy to the thermal domain results in non-dynamic elements.

Energy dissipators are ideal in the sense that they dissipate, but do not store, energy.

Energy transfer elements. These elements conserve energy, merely routing it through the model, between any other model elements. In some energy domains these elements are well-defined (e.g. parallel connections in electrical systems), while in others they are more abstract (common force points in mechanical systems). Included in this group of elements are couplers which neither store nor dissipate energy, but transform the effort and flow variables without energy loss.

It is recognised that it is also important to have system outputs (via sensors), but for analysis purposes outputs are signals and do not exchange energy. A sensor output may also exhibit dynamics, which may be either inherent or due to its location or relationship to the measured variable. Outputs will be dealt with in detail in the discussion of causality (Chapter 3).

The behaviour of a specific element is described by a physical law which is expressed as its *constitutive relation* , and the form of this relationship determines which of the above groupings is appropriate to a given element. Specific constitutive relations will be discussed further in Section 2.2.3, after we have looked in more detail at the energy transfer elements.

2.2.2 Model structure

The energy transfer elements actually represent the model structure, and are called multi-ports, indicating that they have two or more ports for transferring energy. The constitutive relation which is common to these elements is that the sum of all the energy flows into the junction is zero, i.e.

$$e_1f_1 + e_2f_2 + \cdots + e_nf_n = 0 \tag{2.2}$$

where subscripts $1, 2, \ldots n$ indicate the ports through which energy is flowing into the element. Note that a sign convention must be chosen which is consistent; for example, (as here) all energy flows being measured *into* the element.

There are four basic elements within this category, two of which maintain one of the variables constant through the element, and two of which perform a transformation.

Junctions

The first type is termed a junction element where either effort or flow is fixed and the co-variables must sum to zero. Electrical engineers will recognise this as a more general formulation of Kirchoff's Laws. There are two such laws for each energy domain, since either the effort or the flow may be fixed at a specific junction. Thus at an effort junction (also termed a parallel junction from its electrical domain equivalent) the following relations must hold

$$e_1 = e_2 = \cdots = e_n \tag{2.3}$$

$$\text{and } f_1 + f_2 + \cdots + f_n = 0 \tag{2.4}$$

Conversely, for a flow (series) junction the flow is fixed for each path into or out of the junction while the efforts must sum to zero, i.e.

$$f_1 = f_2 = \cdots = f_n \tag{2.5}$$

$$\text{and } e_1 + e_2 + \cdots + e_n = 0 \tag{2.6}$$

The direction of energy flow is generally assumed to be from input sources and into stores and dissipators. With a complex junction structure it is sometimes not obvious which way energy may be flowing, so our structural conventions must be able to unambiguously represent the chosen sign of the energy flows.

Transformers and gyrators

If the energy transfer element also transforms one of the effort or flow variables then the co-variable must also be transformed such that the energy conservation relationship (Equation 2.2) is still valid. The most widely used elements of this type have just two ports, so these will be described here, although the description can be applied more generally to n ports.

There are two elements of this type - the transformer and the gyrator. A two-port transformer has a relationship where the efforts on the two ports are constrained by the relationship

$$e_2 = ke_1 \tag{2.7}$$

where the transformer ratio, k, is either a constant or may be dependent on some other system variable, resulting in a modulated transformer. For energy conservation to hold at any instant

$$\begin{aligned} e_1 f_1 &= -e_2 f_2 \\ \text{so } f_1 &= -k f_2 \end{aligned} \tag{2.8}$$

The direction of power flows is normally defined such that one port is an input and the other an output, resulting in the transformer ratio being positive for both effort and flow relations.

Typical physical examples of transformer elements are a frictionless lever in the mechanical domain, or a two port transformer in the electrical domain. The reason that the latter example only transforms A.C. signals will be used to show how energy bond graphs can provide deeper insight into system behaviour.

The gyrator constitutive relation occurs when the relation is constrained by

$$f_2 = ge_1 \tag{2.9}$$

where g is referred to as the mutual conductance.

Substituting (2.9) back into the energy conservative relation (2.2) for a two-port, gives the complementary form of the gyrator constitutive relation

$$f_1 = -ge_2 \tag{2.10}$$

As for the transformer ratio, the mutual conductance (g) may be either a constant or dependent on some other system variable as long as both relations are simultaneously true.

Physical instances of gyrators are less easily recognised than transformers, as they occur most often when transformation from one energy domain to another is modelled. A typical example is the fixed field DC motor where the back e.m.f. generated by the armature rotation is proportionally related to the shaft speed – by the motor gyrator constant, and the input current is related to the load torque by the same constant. If the field current is derived by placing the field winding in series with the armature winding, then the mutual conductance becomes a function of this current resulting in a modulated gyrator.

2.2.3 Constitutive relationships of energy nodes

Energy sources, stores and dissipators have been identified as the basic elements which may be used to emulate the range of system behaviours required for a comprehensive energy model. A fuller understanding of these elements can be gained by studying their constitutive relations (CR) . These constitutive properties of an element will generally be expressed as a an algebraic equation relating the effort and to the integrated flow or vice versa, although this behaviour could equally be described by a graph. In many real physical systems the relationship between the effort and flow variables may be non-linear, and thus it is important that any modelling technique adopted must be able to handle constitutive relations which are non-linear.

Energy sources

The system inputs can be either effort sources or flow sources, where the type of source defines the variable controlled by the source, which, for an ideal source, is independent of the co-variable: The value of the co-variable is defined by the system which the source supplies. Thus using an electrical example once again, a battery is an effort source and if the system consists of a resistor across the battery terminals, then this resistor determines the current (flow) from the battery. Sources can also be modulated by another system variable, as is often the case with control systems, and in the electrical domain, an amplifier providing a low impedance voltage output may be modelled as a modulated effort source.

The constitutive relation for an effort source is

$$e = e_0 \tag{2.11}$$

and for a flow source

$$f = f_0 \tag{2.12}$$

Where e_0 and f_0 are (possibly modulated) constants.

Energy stores

Energy stores are a little more complicated, but again there are two basic types - those that accumulate effort and those that accumulate flow [1].

Dealing first with effort-accumulating stores, the general constitutive relation has the form

$$f = \phi(p) \tag{2.13}$$

where $\phi(p)$ is a (possibly nonlinear) function of the *integrated effort* or *generalised momentum* p given by

$$p = \int e\ dt \tag{2.14}$$

In the *linear* case, Equation 2.13 can be rewritten as

$$f = \frac{p}{I} \tag{2.15}$$

where the proportional constant I is called the *inertance*. Equation 2.14 is shown in integral form as this best indicates the storage mechanism and is physically realisable.

[1] There is the possibility of confusion here. Some authors use 'effort store' to refer to a store with effort output, that is a flow-accumulating store, and 'flow store' to refer to a store with flow output, that is an effort-accumulating store

However, Equations 2.14 and 2.13 can be rewritten in derivative form as

$$e = \frac{dp}{dt} \tag{2.16}$$

$$p = \phi^{-1}(f) \tag{2.17}$$

where ϕ^{-1} is the inverse of ϕ.

Example

An example of an effort store from the mechanical domain occurs when the effort variable, *force*, is applied for a time to a *mass*, resulting in a change in the flow variable, *velocity*. i.e.

$$velocity = \frac{1}{mass} \int force\,dt$$

the energy imparted to the mass has been stored as kinetic energy and the accumulated energy is given by

$$E = \frac{mass}{2} velocity^2$$

In a similar way, the flow-accumulating store has a general constitutive relation of the form

$$e = \phi(q) \tag{2.18}$$

where $\phi(q)$ is a (possibly nonlinear) function of the *integrated flow* or *generalised displacement* q given by

$$q = \int f dt \tag{2.19}$$

In the *linear* case, Equation 2.18 can be rewritten as

$$e = \frac{q}{C} \tag{2.20}$$

where the proportional constant C is called the *capacitance.*

Example

An easily visualised example of a flow store is a tank filled with incompressible fluid by a flow source at the bottom of the tank. The flow variable in this case is the volume flow rate of fluid into the tank, and the effort variable is the resulting pressure at the bottom of the tank. Simple hydraulics indicate that this pressure is given by

$$\begin{aligned} Pressure &= \frac{volume \times density \times g}{area} \\ &= \frac{density \times g}{area} \int volumeflowrate\,dt \end{aligned}$$

Hence the capacity, C, is $area/(density.g)$ and in this case, the energy is stored as potential energy in the head of water in the tank.

Energy dissipators

Energy dissipators are not divided into effort or flow types because their constitutive relations can generally be expressed in either form,

$$e = \phi(f); \; f = \phi^{-1}(e) \tag{2.21}$$

or, in the linear case,

$$\text{i.e. } e = Rf \quad \text{or} \quad f = e/R \tag{2.22}$$

These Equations (2.22) are seen to be general forms of Ohm's law in the electrical engineering domain, where R represents an electrical resistance, and the energy dissipated in the linear case may be expressed as

$$E = \int f^2 R\, dt = \int e^2/R\, dt \tag{2.23}$$

Mechanical and hydraulic dissipators are not necessarily linear, however, and thus their constitutive relations may be more easily calculated when expressed in one particular form. Dissipators in these domains exert forces which always oppose the direction of motion imposed upon them and vary according to a variety of laws. The effort (pressure drop) generated by incompressible flow through an orifice is typically given by

$$e = Rf|f|$$

thus giving two possible values of flow if this is expressed as a function of the effort variable.

As a final comment on dissipators, it should be realised that when modelling thermodynamic systems one is often specifically interested in calculating the dissipation of thermal energy into the environment, and so the environment itself contributes to the system variables. Thermodynamic systems will be dealt with in more detail in Section 2.3.

Due to the conflicting variable names used in each energy domain, and since the point of using energy as the manipulated variable is to unify the approach to all these domains, the designations used in this section will be used throughout the remainder of the text. The correspondence of these variables to individual energy domains is shown in Table 2.1.

To summarise this brief overview of modelling systems as energy manipulators, we have identified four basic element types which can be differentiated by the form of their constitutive relations . Elements which conserve energy and distribute it between other elements are seen to define the structure of the system. The remaining elements have constitutive relations which either put energy into the system (sources), remove energy from the system (dissipators), or store either potential or kinetic energy (stores). These energy stores accumulate all the history of the system and thus can be used to derive state variables for dynamical models.

Domain	Effort	e	Flow	f	Momentum	p	Displacement	q
Electric	EMF (voltage)	e V	Current	i	Lines	λ	Charge	q
Magnetic	MMF	M	Flux rate	$\dot{\phi}$	-	-	Flux	ϕ
Hydraulic	Pressure	P	Volume flow rate	$\dot{V}$	Pressure momentum	p	Volume	V
Mechanics (trans)	Force	F	Velocity	V	Momentum	p	Displacement	x
Mechanics (rotation)	Torque	T	Angular velocity	ω	Angular momentum	h	Angle	α
Thermo-dynamics	Temperature	T	Entropy flow rate	$\dot{S}$	–	–	Entropy	S

Table 2.1 Effort and flow variables for each energy domain

2.3 ENERGY BOND GRAPH MODELS

The bond graph notation is a graphical language designed specifically for the description of processes which manipulate energy. In consequence, the language includes elements which model all the requirements analysed in the preceding discussion on structure and constitutive relations . A graphical notation is necessary in order to provide a concise description of the entire process at a higher level of abstraction than the equations describing the energy transfers between elements. In addition, bond graphs also highlight the structure of the model, making the mapping between the model and the system somewhat more intuitive.

If it were just the case that bond graphs provide 'the acceptable face of energy equations' to improve their palatability to engineers, the notation would have less value than it actually provides. It is hoped that the following discussion will show how bond graphs not only represent the process in a form with which the user can easily interact, but also help to improve understanding of process fundamentals and yet permit unambiguous interpretation of the graph by software for transformation to a variety of derived models.

The remainder of this section describes bond graph syntax, with special emphasis on the interpretation of computational causality. Finally, the use of multi-port elements is described with, hopefully, a fresh view on their application.

2.3.1 Energy bonds

Bond graphs have the effect of shifting the user's attention away from the element which manipulates energy and towards its interaction with the rest of the system in which it exists. The energy bond carries all the information about this interaction, which notionally occurs through a 'port' on the element.

The bond is represented as a half arrow (Figure 2.1a) indicating the (supposed) direction of energy flow, between the ports to which it is attached. The bond may be annotated by symbols representing the effort (above the bond) and flow (below the bond) subscripted with the bond identification, which is typically the same as the identification of the attached energy node.

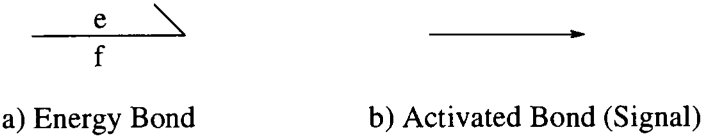


Figure 2.1 Representation of bonds and signals

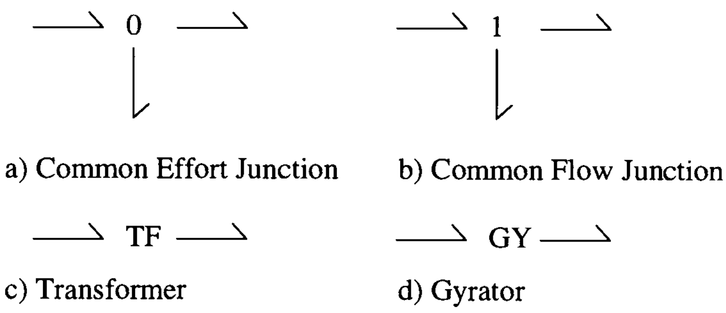
An energy transfer is implicit in every bond, so an equivalent symbol is required to indicate the transfer of zero energy signals (or information). The symbol for a signal is the full arrow (Figure 2.1b) borrowed from block diagram notation. The signal may convey either an effort or a flow, or alternatively the value of a state variable. By convention, a signal pointing towards an energy node implies that the constitutive relation of that element is modulated by the value conveyed by the signal. A shorthand notation has arisen where a signal directed at a junction implies a combination of a signal modulating the appropriate energy source on that junction, without having any effect on the source junction, i.e. a buffered signal. For this reason, signals are also called activated bonds, although these are distinguished from modulating signals in bond graphs given in this text by representing modulating signals as dashed arrows.

2.3.2 Junction structure

The need for the four structural elements provided by bond graphs has been outlined in Section 2.2, and these are illustrated in Figure 2.2.

The (common) effort junction is conventionally called a '0' junction, and has at least two ports, but more typically three or more. The constitutive relation of the '0' junction ensures that the effort is identical at each port and that the algebraic sum of the flows on each port is zero. The (common) flow junction is called a '1' junction and conserves energy by defining the flows on each port to be identical while the efforts sum to zero. Since the '0' and '1' junctions are generalisations of parallel and series electrical junctions, a convention has arisen labelling these as 'p' and 's' junctions respectively (Thoma, 1990). Since this is meaningless for mechanical systems and the '0' and '1' convention is most widely used, this book uses the latter henceforth.

Transformers are designated by 'TF' nodes in bond graphs and are again power conserving although the effort on the output port is scaled by the transformer ratio



a) Common Effort Junction

b) Common Flow Junction

c) Transformer

d) Gyrator

Figure 2.2 Junction structure elements

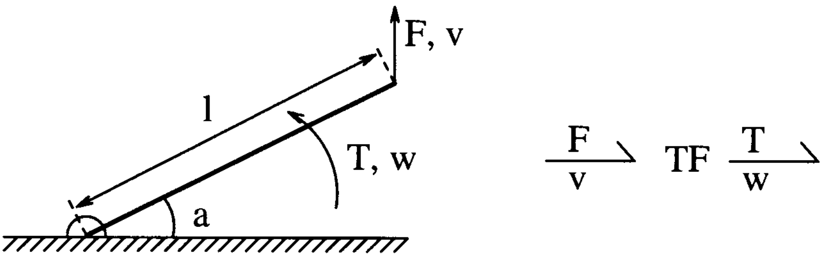


Figure 2.3 Transformation between mechanical domains

to the effort on the input port. In Section 2.2.2 we noted that the transformer ratio can be modulated by another system variable, which is indicated graphically by directing a signal toward the 'TF' node from a node carrying the relevant system variable. An example of a modulated mechanical transformer is shown in Figure 2.3 where a rigid bar pivoted at its end converts the translational force F to a torque T with a transformer ratio $(l\ \cos(\alpha))$ dependent on the angle of the bar.

$$\begin{aligned} T &= (l\ \cos(\alpha))F \\ \text{and}\ \ (l\ \cos(\alpha))w &= v \end{aligned}$$

Figure 2.3 is also an example of the use of a transformer to convert between energy domains - in this case, between the translational and rotational mechanical domains.

Gyrators (designated 'GY') are also energy conserving, but directly relate the input effort to the output flow - they most frequently occur when representing transducers between energy domains. A further example of this is shown in Figure 2.4 where an electrical coil wound on a magnetic core is modelled as a gyrator between the electrical and magnetic energy domains.

In this case, the coil gyrates electrical effort (e.m.f.) to magnetic flow (rate of change of flux), with a gyrator ratio equal to 1/N, where N is the number of turns in the coil (Faraday's Law). Since the gyrator is energy conserving, the electrical flow is related to the magnetic effort variable (m.m.f.) by the same ratio. Whereas

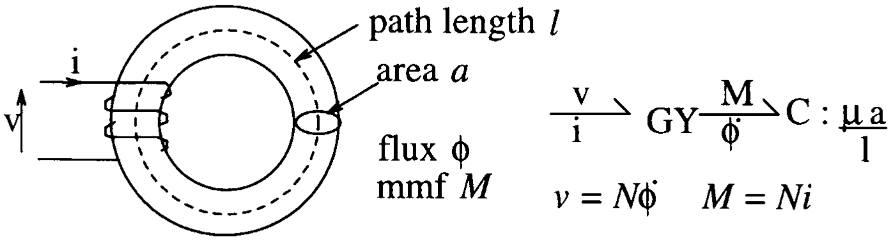

path length l
area a
flux ϕ
mmf M
i
v
$\frac{v}{i}$ GY $\frac{M}{\dot{\phi}}$ C : $\frac{\mu a}{l}$
$v = N\dot{\phi}$ $M = Ni$

Figure 2.4 Gyration between electrical and magnetic domains

the coil appears to the electrical system to which it is connected to be an effort store, this model indicates that the energy is stored in a flux store in the magnetic domain. The capacitance of the magnetic circuit (normally called the permeance) can be shown to be given by

$$C = \frac{\mu A}{l} \tag{2.24}$$

Hence the magnetic effort generated by the flow into this capacitance is given by

$$\begin{aligned} M &= \frac{1}{C}\int \frac{d\phi}{dt}\,dt \\ &= \frac{1}{C}\int \frac{e}{N}\,dt \\ &= \frac{l}{\mu A N}\int e\,dt \end{aligned} \tag{2.25}$$

The gyrator relation gives

$$i = \frac{M}{N} = \frac{l}{\mu A N^2}\int e\,dt \tag{2.26}$$

i.e. the electrical inductance is $\mu A N^2/l$

2.3.3 Energy nodes

In Section 2.2.2 we divided energy nodes into three categories: energy sources, stores and dissipators. Table 2.2 shows the bond graph representations for each of these elements and standard (linear) forms for their associated constitutive relations. Non-linear constitutive relations are of course possible, and may be represented within a bond graph model. Each node is illustrated with one associated energy bond, indicating that these are representations of single port elements. The effort and flow sources are shown supplying energy while for the remaining elements the nominal direction of the energy flow is toward each element.

At this point, it is useful to consider what elements or behaviours these symbols represent in the context of specific energy domains.

Symbol	Element type	Constitutive relation
SF $\rightharpoonup$	Flow source	$f = f_{in}$
SE $\rightharpoonup$	Effort source	$e = e_{in}$
$\rightharpoonup$ C	Flow store	$e = 1/C \int f\, dt = q/C$
$\rightharpoonup$ I	Effort store	$f = 1/I \int e\, dt = p/I$
$\rightharpoonup$ R	Dissipator	$e = Rf$ or $f = e/R$

Table 2.2 Bond graph elements

Electrical Elements

Since this domain has relatively ideal components, their behaviours can be mapped exactly onto those listed in Table 2.2. Voltage and current sources are represented by 'SE' and 'SF', respectively, and these can be modulated by some other system variable to model perfect amplifiers. 'C' and 'I' energy stores represent capacitors and inductors which store energy either as electric charge or magnetic flux. Finally, electrical resistors dissipate energy from the system and can be represented by 'R' nodes in the bond graph.

Magnetic Elements

Magnetomotive force (m.m.f.) can occur as a fixed effort source, when modelling the remanent magnetism in a permanent magnet, or as an effort source when produced by an electric current in a wire. In our discussion of gyrators, in Section 2.3.2, it was noted that the magnetic flow (rate of change of flux) is proportional to the voltage across the coil, so a magnetic flow source is created whenever a voltage is applied to an electrical path.

It was also seen that the energy is stored in the magnetic path, due to the accumulation of flux resulting in a magnetic 'C' element. There is no equivalent magnetic element to the effort store - 'I' node - this will be discussed further at the end of this section. Magnetic circuits can only dissipate energy when the m.m.f. is changing - this is due to the hysteresis loss of a magnetic core, and can be modelled by an 'R' node. Eddy current losses can also occur in metal cores, but these are due to a gyration back to the electrical domain.

Hydraulic Elements

When dealing with incompressible hydraulics, pumps can be represented by 'SE' (pressure sources) or 'SF' nodes depending on the type of pump. A tank capacity is readily seen to be an accumulator of flow, and is represented by a 'C' node.

An ideal pressure source is a large reservoir - effectively an infinite capacitance. The kinetic energy associated with mass flowing through a pipe is the result of accumulated effort and represented by an 'I' node.

Energy may be dissipated in two basic ways in hydraulic systems, either due to viscous forces between the fluid and static objects or viscous forces between fluid particles. Both are represented by an 'R' node. Laminar flow results in a linear constitutive relationship, but whenever turbulent flow exists this becomes highly non-linear.

Mechanical Elements

Although translational and rotational mechanics are deemed to be separate domains, they are dealt with together here as so much terminology is common. Imposed forces and torques are effort sources, the most common constant 'SE' node being gravity. Imposed (linear or angular) velocities are also possible, represented by the 'SF' node.

Mechanical engineers make the distinction between potential and kinetic energy, according to whether it is stored in a 'C' or 'I' element respectively. Springs are flow stores ('C' nodes), while mass accumulates effort and is represented by an 'I' node. Friction dissipates energy from the mechanical system and is represented by an 'R' node (often with a non-linear constitutive relation).

Thermodynamic Elements

Thermodynamic systems are often analysed using the variables temperature and heat flow rate (Q), but the latter cannot be used as the flow variable in an energy bond graph, as it is an energy rate variable. One can use heat flow rate in some bond graph representations (called pseudo bond graphs), but then care has to be exercised in interfacing with other energy domains.

Energy bond graphs for thermodynamic systems use entropy flow rate ($\dot{S}$) as the flow variable and absolute temperature (T) as the effort variable, thus satisfying the requirement that the product of effort and flow is instantaneous power. Effort sources are therefore models of elements which can force the temperature at one point in the system - a standard 'SE' input to thermodynamic systems is the ambient temperature.

Although entropy flow sources do exist they rely on inputs from other energy domains - cf. flow sources in the magnetic domain. In this case, energy lost through a dissipator in the other energy domain is conserved in the thermodynamic domain and emerges as a defined entropy flow rate. Since this is such a common mechanism for sourcing entropy flows, bond graphers have added the 'RS' node (Figure 2.5) to the terminology.

The constitutive relation of the 'RS' node is energy conservative as indicated

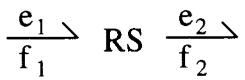


Figure 2.5 An 'RS' element

in Equation 2.27

$$\begin{aligned} f_2 &= \frac{e_1 f_1}{e_2} \\ \text{i.e. } \dot{S}_2 &= \frac{e_1 f_1}{T_2} \end{aligned} \tag{2.27}$$

It can be seen from this constitutive relation that this is a modulated 'SF' node in that the flow is dependent on the effort variable (temperature), as well as the energy imparted from the other domain.

The argument applied to dissipators from other energy domains conserving energy by passing it into the thermal domain, implies that thermodynamic dissipator cannot exist. Thermal resistance is not a dissipator but rather a dual entropy flow source which is also represented by an 'RS' node. The constitutive relation of a thermal resistance is given by

$$\dot{E} = T_1\dot{S}_1 = T_2\dot{S}_2 = H.(T_1 - T_2) \tag{2.28}$$

where H is the heat transfer coefficient.

Thermodynamic systems have flow stores in the form of thermal capacity, represented by a 'C' element. The constitutive relation of a thermal capacity is

$$T = T_0 \exp(S/C) \tag{2.29}$$

where T_0 is the initial (absolute) temperature. Equation (2.29) approximates to

$$T = T_0(1 + S/C) \tag{2.30}$$

for small differences between T and T_0.

Like magnetic systems, there is no effort store ('I') in thermodynamic systems, which has led Breedveld (1984*b*) to the conclusion that such stores are not fundamental. It was shown, in the discussion of gyrators, that the electrical effort store (inductance in a coil) is fundamentally a gyrated version of a magnetic flow store. Thus, it is always possible to use the gyrator's ability to make duals of elements to remove the need for the effort store. They are, however conceptually convenient, and in the mechanical domain, at least, neither the 'I' nor 'C' element appears more fundamental. Breedveld (1984*b*) has proposed a generalised bond graph theory, where inertances only exist when gyrated from 'C' elements, thus requiring dual (potential and kinetic) mechanical domains.

2.4 BOND GRAPH EXAMPLES

2.4.1 An Electrical Second Order Lag

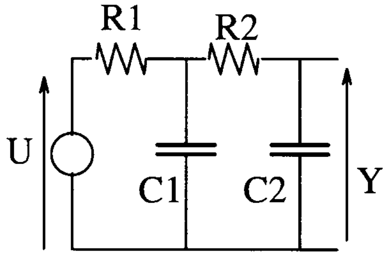


Figure 2.6 An electrical second order lag: schematic

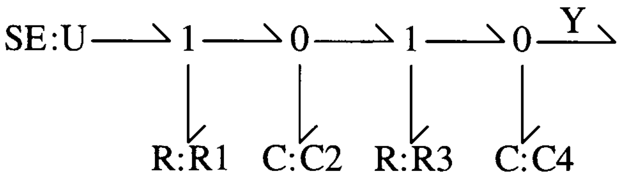


Figure 2.7 An electrical second order lag: bond graph

The electrical schematic for a second order lag is given in Figure 2.6, while the bond graph equivalent is shown in Figure 2.7.

Since electrical schematics provide an unambiguous representation of the real system, it is possible to give precise rules for transforming such schematics to bond graph notation:

1. Draw a '0' junction for each point in the schematic where parallel paths coincide.
2. Draw a '1' junction for each component on a series path, and attach the appropriate bond graph component by a bond to that junction. The arrowhead on each bond indicates the assumed direction of power flow, i.e. from sources and towards stores and dissipators.
3. Draw bonds between adjacent junctions, again indicating notional direction of power flow.
4. Remove the '0' junction representing the reference point (typically the 0 Volt rail) and remove all bonds attached to this junction.
5. Remove any remaining two-port junctions and move attached nodes to the adjacent junction.

This procedure converts even the most complex electrical schematics to bond graph form, for further analysis using bond graph techniques. The 'SS' element at the end of the graph shown in Figure 2.7 has been added to represent a sensor, since for this second order lag, we are interested in monitoring the output voltage across capacitor C4.

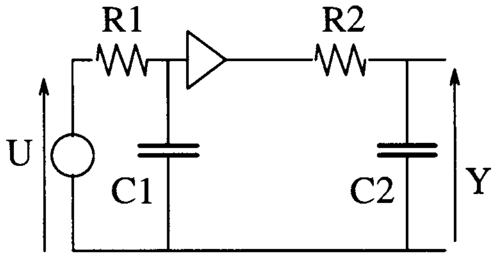


Figure 2.8 An electrical second order lag with buffer: schematic

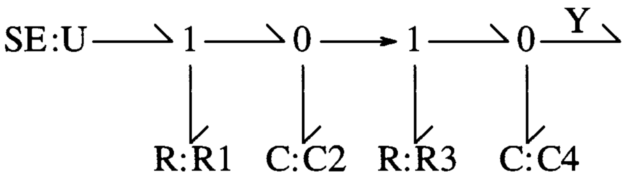


Figure 2.9 An electrical second order lag with buffer: bond graph

Figure 2.8 shows a modified version of the circuit of 2.6 containing a buffer amplifier (of unit voltage gain) connecting the two halves of the circuit. In this case there is no current flow from the '0' junction to the '1' junction and so the corresponding bond in Figure 2.9 is replaced by a signal.

2.4.2 A hydraulic brake system

Figure 2.10a shows a simplified schematic of an automobile braking system with a hydraulic system connecting the foot pedal to two brake pads, pressing against the brake disc.

The system is shown first as a word bond graph (Figure 2.10b) to better illustrate the components of the system, while Figure 2.10c shows the complete bond graph of this system. A force is applied (by the effort source SE1) to the brake pedal, which is coupled by an end-pivoted lever, represented by TF2, to a return spring with compliance C3. Since the piston rod is connected to a third point on the lever a further transformer (TF4) is required to couple the resultant of the applied and spring forces to the piston rod. Frictional force imposed on the piston rod is represented by R5 which is attached to the '1' junction representing the velocity of the piston rod.

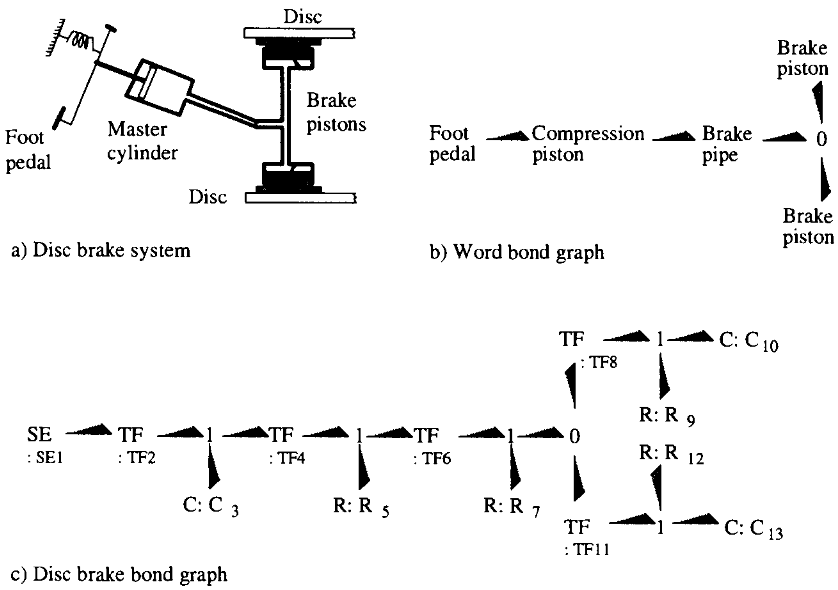


Figure 2.10 A disc brake system

The master cylinder (TF6) transforms the force on the piston to a hydraulic pressure. This pressure is measured at the outlet of the master cylinder into the brake pipe, which is assumed to have a small resistance (R7) to fluid flow. The brake fluid is assumed to be incompressible, as is the case for normal safe operation of the system; in a faulty system, however, air in the fluid can make it appear compressible.

The split into pipes for each brake is modelled by a '0' junction where the common pressure is applied to each brake piston in its caliper cylinder. These cylinders transform (TF8, TF11) the hydraulic pressure to forces on the brake pads which firstly overcome the frictional forces and the compliance due to the pad retainers. The reaction force from the brake disc may be modelled in several ways, but here it has been chosen to model this by modulating the dissipator parameters (R9 and R12) according to the position of the pads (i.e. the states of C10 and C11). The modulation causes the 'friction' to become infinite when the pads meet the disc thus giving zero velocity. A more detailed model of this system could employ an 'RS' element (Section 2.3.3) to indicate that the force of the pad on the disc converts energy into heat which can effect the pad friction parameters and cause the brake fluid to expand.

2.4.3 A DC motor

The bond graph model of a DC motor is developed from first principles, by considering the force F on a current carrying wire perpendicular to a uniform magnetic field B. If the length of the wire is l and the current is i, then Faraday's law gives

$$F = Bli \tag{2.31}$$

Assuming the wire is free to move across the magnetic field with velocity u the e.m.f. generated in the wire is

$$e = Blu \tag{2.32}$$

Since we have defined voltage and translational force as effort variables, and current and velocity as flow variables, we can see that Equations (2.31) and (2.32) represent gyrator action between the electrical and mechanical energy domains. The power passed through the motor is $Blui$, and the gyrator ratio is Bl.

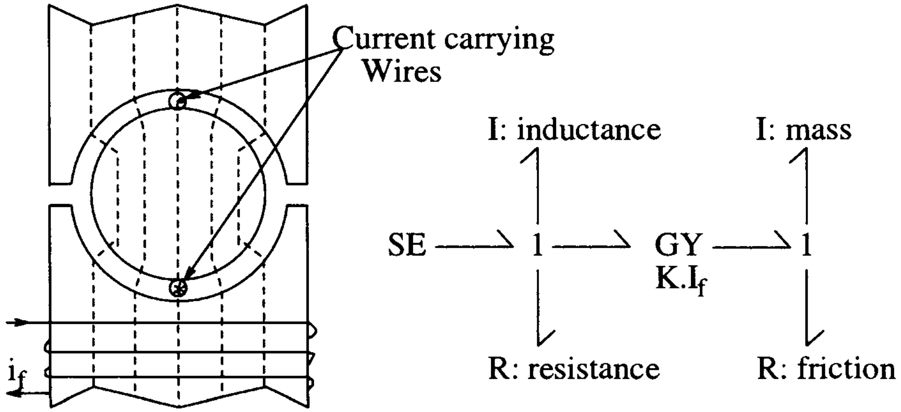


Figure 2.11 Models of a DC motor

Figure 2.11a schematically shows how this is implemented in a DC motor, where each turn of the armature wiring experiences a force $2F$ due to the two lengths per turn. In practice, the armature winding has significant resistance and inductance since many turns are required. The armature mass also results in rotational inertia, while friction losses occur in the bearings. The bond graph model is shown in Figure 2.11b, indicates that the electrical resistance and inductance are in series with the EMF required to drive the motor, and the armature inertia and friction losses are on a common velocity junction.

It can be seen that the gyrator ratio is proportional to both the number of active turns on the armature (n), and to the magnetic flux density, B. Since the magnetic field is often generated by a separate field winding, the gyrator ratio is then dependent on the field current, since

$$B = \frac{\phi}{A} = \frac{\mu N i_f}{l_e} \tag{2.33}$$

where μ is the permeability of the field core, N is the number of turns on the field winding, l_e is the effective magnetic path length, and i_f is the field current. Thus for a given motor the gyrator ratio is Ki_f,

$$\text{where} \quad K = \frac{2\, ln\ N\mu}{l_e} \tag{2.34}$$

Hence the motor gyrator ratio is actually modulated by the field current, if this is not constant.

2.4.4 An electric heater

For this example we will develop an energy bond graph model of an electrical heater rather than the more common model which uses heat flow rate as the flow variable. Figure 2.12 models the electro-thermal conversion as an energy conservative RS element which sources an entropy flow to the thermal capacitance (C3) of the heater, and to a thermal resistance (RS4) representing heat loss to the ambient (SE5).

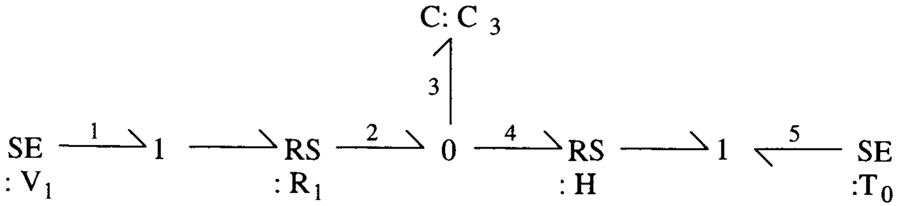


Figure 2.12 Bond graph of an electrical heater

The input power from the electrical source is V_1^2/R_1, where R_1 is the electrical resistance of the heater. The thermal power generated is therefore

$$e_2 f_2 = V_1^2/R_1 \tag{2.35}$$

where e_2 is the absolute temperature and f_2 is the entropy flow generated. This entropy flow splits at the '0' junction between that into the thermal capacitance, causing the rise in temperature, and that passing through RS4 to ambient. The (linearised) rise in temperature is approximated $T_0 S/C_3$ where T_0 is the initial (ambient) temperature and S the integrated entropy flow into C3, giving

$$e_3 = T_0(1 + S/C_3) \tag{2.36}$$

The heat flow rate to ambient is

$$Q_4 = e_4 f_4 = H(e_4 - T_0) \tag{2.37}$$

where H is the thermal conductance, between heater and ambient.

Since the efforts are common at a '0' junction

$$e_4 = e_2 = e_3 = T_3 \tag{2.38}$$

and the flows split at this junction

$$f_2 = f_3 + f_4 \tag{2.39}$$

Therefore

$$\begin{aligned} f_3 &= [V_1^2/R_1 - H(T_3 - T_0)]/T_3 \\ &= [V_1^2/R_1 - H(T_0(1 + S/C_3) - T_0)]/T_0(1 + S/C_3) \end{aligned} \tag{2.40}$$

For $S/C_3 < 1$ we can approximate $(1 - S/C_3) = 1/(1 + S/C_3)$, giving the state equation

$$\begin{aligned} f_3 &= [V_1^2/R_1 - HT_0S/C_3].(1 - S/C_3)/T_0 \\ \text{i.e. } \dot{S} &= [V_1^2/R_1 - HT_0S/C_3 - V_1^2S/(C_3R_1)]/T_0 \end{aligned} \tag{2.41}$$

ignoring terms in $(S/C_3)^2$.

2.5 CAUSAL AUGMENTATION OF BOND GRAPHS

The concept of (computational) causality is central to the systematic resolution of bond graphs into the mathematical form chosen by the modeller. Due to the importance of this concept we have devoted Chapter 3 to exploring this in more depth. This section explains causality in the context of bond graph analysis.

Assigning the causal orientation of a given bond in the graph implies that specifically either the effort or flow variable on that bond is known, and this known value (or expression) may then be propagated through the graph to arrive at a complete mathematical model. The rules for causally augmenting the bond graph permit the system equations to be ordered automatically for solution either by hand or by computer software.

In keeping with the concise graphical approach, causality is indicated on the bond graph by a causal stroke at one end of a bond joining two nodes on the graph. This stroke is drawn at the end of the bond nearest the node to which the effort is directed - the flow by implication is directed toward the node at the other end. The only elements that can force causality are effort or flow sources, and the structural elements - Figure 2.13a shows this notation applied to sources and to dissipators - the indicated direction of the energy flow is seen to be irrelevant to causality.

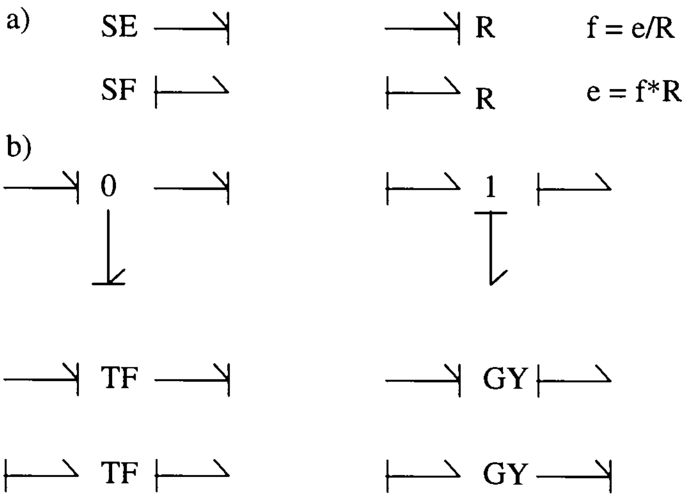


Figure 2.13 Permissible causalities

Figure 2.13b shows how causality is propagated through the bond graph by the structural elements; '0', '1', 'TF' and 'GY'. Since the effort at a '0' junction is common to all the bonds on that junction, only one bond can define the effort on that junction, the remaining bonds impose flows on the junction, while propagating the known effort to attached nodes. In contrast, only one bond determines the flow at a '1' junction, while the remaining bonds impose efforts on the junction. The transformer ('TF' node) passes causality on directly (thus a bond can be considered as a transformer with ratio 1), while gyrators have the effect of inverting causality - hence the application of gyrators to achieve the dual of an element.

Elements which are energy stores or dissipators do not impose causality on the system, although they may have preferred causality for computational reasons. In general, therefore, *the causality assignment of a given bond graph is not unique,* being dependent on the modeller's choice of mathematical model. In particular, systems having a large proportion of dissipators could be described as 'under-causal' since the modeller may have to make one or more arbitrary choices of causality in order to complete causal assignment on the bond graph. The consequence of under-causal systems is that some intermediate variable has to be eliminated by the solution of an algebraic loop, before the complete mathematical model can be derived.

In such cases, the modeller's choice of causality assignment may not be entirely arbitrary, but rather as preferred to improve ease of computation and minimise the number of algebraic loops (Lorenz and Wolper, 1985). For example, it is more convenient to calculate the effort variable from the flow for a dissipator representing the turbulent flow through a pipe where pressure drop (e) is given by the following function of flow rate (f)

$$e = Rf|f|^{\frac{3}{4}} \tag{2.42}$$

It has been noted that activated bonds can be used to represent sources modulated by a signal. In such cases, the activated bond has the causality of the modulated source and is drawn with a causal stroke in this text. For example, an activated bond from a '0' junction transmits the effort from that junction and is drawn with a causal stroke at the destination end of the signal.

2.5.1 Integral or derivative causality?

Table 2.2 shows that the constitutive relations of the energy stores contain information about the system inputs and state variables p and q, thus permitting the system dynamics to be fully represented. The emphasis in bond graph literature has been on the transformation of graphs to state equations - choosing alternative causality assignment rules results in different forms of mathematical model. When one is transforming the bond graph into its state equation form, the causality of interest for energy stores is termed integral causality, where the constitutive relations of the energy stores are in the form given in Table 2.2. The ability to assign integral causality also implies that the system is physically realistic, thus providing a deeper level of analysis of system constraints than would be possible without the concept of causality. A mixture of integral and derivative causality may then be forced by the causality propagation in real physical systems, but it implies that at least two of the energy stores are not dynamically independent - only those exhibiting integral causality result in state variables. This causal conflict can be considered as 'over-causal' by comparison with largely dissipative systems, since the consequence is also an algebraic loop - this time relating the interdependent energy stores.

Applying derivative causality to the energy stores in a bond graph results in the derivative form of mathematical model for the system. The resulting mathematical model is then in the most general form - a set of differential and algebraic equations (DAEs), although in some cases ordinary differential equations (ODEs) may result.

Derivative causality may also be applied to energy stores in order to facilitate static analysis of systems, without modifying the fundamental structure of the bond graph model. Since the derivative forms of the constitutive relations for energy store are

$$e = I\frac{df}{dt} \quad \text{and} \quad f = C\frac{de}{dt} \tag{2.43}$$

it can be seen that the static model is given either when the constitutive parameters I and C are zero, or when the effort and flow derivatives are all zero, i.e. the stationary state.

Thus, assigning derivative causality to stores and propagating this through the bond graph permits a derivative-based model to be generated, and substituting zero for all energy store components then results in the steady-state mathematical model. Similar bond graph solutions to the problem of deriving the steady-state model have been proposed by Breedveld (1984*a*), but the method described here has the advantage of retaining an invariable bond graph core model regardless of the transformation required to obtain the desired mathematical model.

2.5.2 Model reduction

The subject of model order reduction is covered in more detail in Chapter 5, but it is useful to overview the uses to which causal analysis can be put, in this section. The modeller may also wish to investigate the effect on the process when a component is removed. This can be done by removing the element from the graph, or, more conveniently, changing its parametric value to zero. This can have fundamental effects on the system states, due to changes in causality. For example, if the element is a dissipator whose causality was initially defined such that the constitutive relation was evaluated as

$$f = e/R \tag{2.44}$$

then defining R to be zero gives a computational problem, unless the opposite causality is forced by the modeller, with consequent changes in the causal augmentation of the model. This may have the effect of turning a 'stiff' model of the complete system into a reduced order model with interdependent energy stores.

2.5.3 Rules for assigning causality to a bond graph

The rules listed here give a systematic method fǫr causally augmenting a bond graph such that a state equation model may be derived. The corresponding procedure is known as the Sequential Causality Assignment Procedure (SCAP); see, for example, Karnopp *et al.* (1990) Cellier (1991) Thoma (1990) and Wellstead (1979).

1. Assign causality to any known effort or flow, such as activated bonds (signals) derived from junctions. For example, a signal from a '0' junction transmits the effort from that junction while the flow from this junction into the signal is, by definition, zero.
2. Assign causality to bonds linking directly to each source and propagate these causalities as far as possible through the junction structure by applying the causality constraints for structure elements (0, 1, TF, GY).
3. Assign integral causality to each energy store in turn and propagate this throughout the junction structure. Any conflict between the causality due to the store must be resolved by reassigning derivative causality on that store and propagating the new causality through the bond graph.

4. If any unassigned bonds remain, then assign a causality either arbitrarily or for computational convenience and again propagate this through the junction structure. Assigning causality to an unassigned interjunction bond (internal bond), such that this bond forces causality on both attached nodes, minimises the number of algebraic loops. Repeat for any remaining unassigned bonds.

Rule 4 is reconsidered in Section 3.4 of Chapter 3 and by Gawthrop and Smith (1992).

2.5.4 Examples of causally augmented bond graphs

This section considers two examples of causally augmented bond graphs

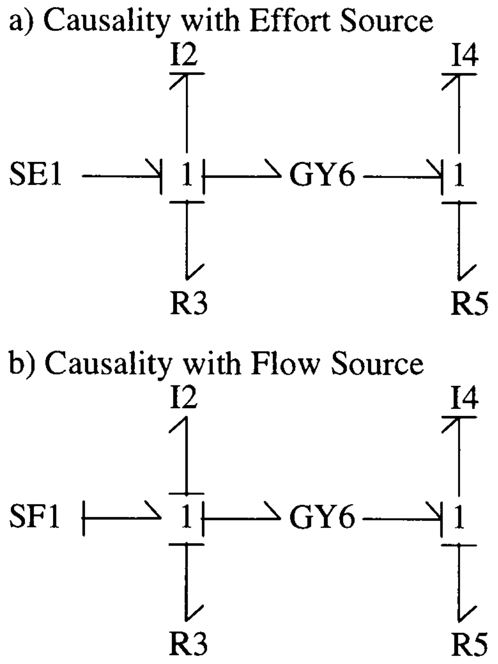
- a fixed field DC motor and
- an electrical operational amplifier circuit.

A fixed field DC motor

A DC motor with a voltage source applied to the armature indicates the potential of bond graphs for unambiguously representing a mixed energy domain system (Figure 2.14a).

Applying the causality rules, with integral causality on stores, results in a model with two state variables p_2 and p_4 and a single input e_1. A bond graph model of the same motor, driven by an electrical current source is shown in Figure 2.14b. Applying the causality rules to this bond graph indicate that I2 now has derivative causality imposed on it, and the system reduces to a first order model since p_4 is the only state variable and the input is f_1. The physical implication of derivative causality on the inductance I2 is that the current source, SF1, must be able to supply the very high voltages which will occur for step changes in motor loading.

Figure 2.15 shows the effect of adding a gearbox to the voltage-driven DC motor. In this case, derivative causality is forced on the motor shaft inertia or the load inertia, since these are not independent, being linked by the transformer ratio of the (non-compliant) gearbox. In practice, the shaft linking the motor to the gearbox will have some compliance resulting in a 'C' element between the motor inertia and the gearbox, which solves the causality problem and introduces another state variable. The likelihood is, however, that this compliance is very small, resulting in a 'stiff' system where the time constant due to the compliance is significantly smaller than those due to the inertias. This may give numerical resolution problems when simulating the system using this mathematical model.



a) Causality with Effort Source

b) Causality with Flow Source

Figure 2.14 Causality variations on a DC motor

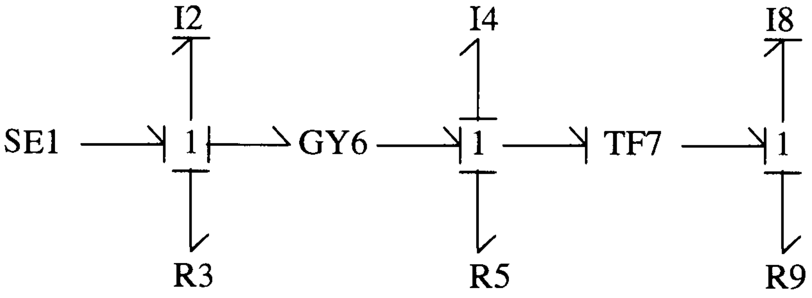


Figure 2.15 Causal conflict due to interdependent inertias

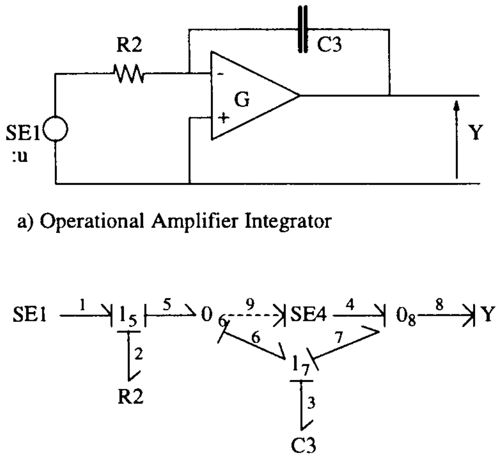
An electrical operational amplifier circuit

A simple inverting integrator is shown in electrical schematic form in Figure 2.16a, where the operational amplifier is assumed to have infinite input impedance, zero output impedance and a large negative gain $(-G)$. The causally augmented bond graph for this circuit is shown in Figure 2.16b, where the operational amplifier is modelled by the modulated effort source SE4, and the modulating signal defines its value as $-G^*$(effort on node 6). This example is analysed in detail to illustrate the systematic manner in which computer manipulation can be applied following

Bond	Effort		Flow	
1	u	3	f_2	13
2	$e_1 - e_5$	11	e_2/R	12
3	q_3/C	7	f_6	17
4	$-Ge_9$	4	$f_8 - f_7$	18
5	e_6	9	f_2	14
6	$e_3 + e_7$	8	$f_5 - f_9$	15
7	e_4	6	f_6	16
8	e_4	5	0	2
9	e_6	10	0	1

Table 2.3 Causally ordered equations

causal augmentation of the bond graph.



a) Operational Amplifier Integrator

b) Integrator Bond Graph

Figure 2.16 An electrical circuit model

Table 2.3 is derived by following causality in the order which it propagates through the graph - the second column under each of the effort and flow columns indicates this ordering. The state equation may be evaluated by selecting the derivative (f_3) of the state variable and working backwards through the ordered equations, i.e. $f_3 = f_6 = f_5 - f_9$ and so on. This process gives

$$f_3 = (e_1 - e_6)/R - f_9 = (e_1 - q_3/C + G * e_6)/R - f_9 \tag{2.45}$$

An algebraic loop (in e_6) has occurred due to the modulation on SE4 in the physical loop, which may be solved as

$$e_6 = \frac{1}{(1+G)}\frac{q_3}{C} \tag{2.46}$$

and hence the full state equation may be derived

$$f_3 = \frac{u}{R} - \frac{1}{(1+G)}\frac{q_3}{CR} \tag{2.47}$$

The system transfer function may be derived from this state equation, giving

$$\frac{y}{u} = \frac{-G}{1+sCR(1+G)} \tag{2.48}$$

which reduces to $-1/sCR$ as expected for $G >> 1$.

2.6 MULTI-PORT ENERGY NODES

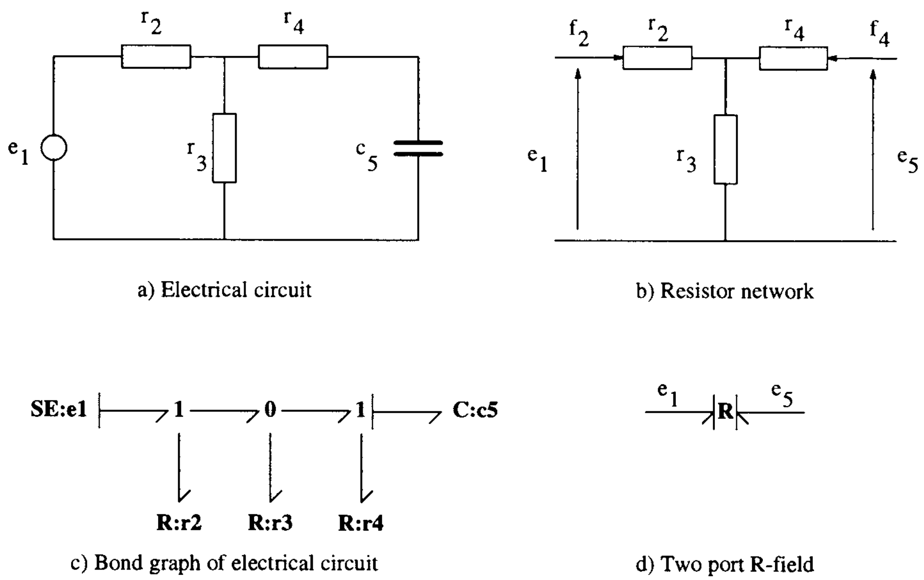
In the preceding descriptions of bond graph elements, all those representing component behaviour, i.e. sources, stores and dissipators, have had only one port through which energy is exchanged with the rest of the system. In general, these elements can be multi-ports (alternatively called N-ports or fields) in the same way that the structure elements, discussed in Section 2.2.2, have more than one interface to the system. This section gives examples of multi-port elements in a variety of energy domains, and their application on bond graph models.

2.6.1 R-fields

In the electrical domain it is often convenient to group a network of resistors together into one multi-port resistor (or R-field) represented by a matrix of resistive (or conductive) elements. Figure 2.17a shows a simple electrical circuit where the dissipators may be grouped together as a two-port 'R-field', as represented in the augmented bond graph of Figure 2.17d. The circuit is also shown represented by one-port 'R' elements in the partially-augmented bond graph of Figure 2.17c. Figure 2.17c indicates that there are several options for completing causality on this bond graph; choosing to assign f_7 as 'known' permits causality to be completed, resulting in only one algebraic loop and the shortest computation.

Alternatively, simple circuit analysis of the resistors as a separate network (Figure 2.17b) gives the constitutive relations of the R-field in the resistive form which may be inverted to give the conductance matrix form required by the given causality

$$\begin{bmatrix} f_2 \\ f_4 \end{bmatrix} = \frac{1}{d}\begin{bmatrix} (R_3+R_4) & -R_3 \\ -R_3 & (R_2+R_3) \end{bmatrix}\begin{bmatrix} e_1 \\ e_5 \end{bmatrix} \tag{2.49}$$



a) Electrical circuit

b) Resistor network

c) Bond graph of electrical circuit

d) Two port R-field

Figure 2.17 Applications of R-fields

where denominator $d = (R_2R_3 + R_2R_4 + R_3R_4)$. It is then a trivial substitution, $f_5 = -f_4$ and $q_5/C = e_5$, to obtain the state equation

$$f_5 = \frac{1}{d}\left(R_3e_1 - (R_2 + R_3)\frac{q_5}{C}\right) \tag{2.50}$$

The same result is obtained for similar computational effort, including the algebraic loop, by following the completed causal assignments on the one-port bond graph. Hence, it can be seen that the R-field has been used to solve the algebraic loop while calculating the matrix coefficients - in such cases, the choice of one-port or multi-port representation is purely the modeller's preference. It can be seen that R-fields can also be defined as having mixed causality, i.e. the dependent vector may be a mixture of efforts and flows.

2.6.2 I-fields

A more useful example of an electrical multi-port is that of an N-port inductance (I-field) representing an electrical transformer with multiple secondaries. For integral causality, the constitutive relations of this I-field are given by a symmetric matrix with self-inductances on the diagonal and mutual-inductances between windings as the off-diagonal elements.

Many mechanical components are also best represented by multi-ports - the dimensional constraints on the mass elements of rigid bodies implies that all such bodies are I-fields, and conceptually it is most appropriate to model such bodies using a single constitutive relation.

2.6.3 C-fields

Multi-port 'C' elements also have significant use in the analysis of mechanical systems - a common example is that of an elastic beam deformed by forces applied to two points along the beam. For such cases the elastic displacement of the beam at the two points is related to both the applied forces and to their relative positions along the beam.

C-fields can also be used to represent the behaviour of energy stores which span energy domains - some transducers operate by storing energy in one domain and later converting it (ideally without loss) into the other domain. An example of such a transducer is the condenser microphone, where a velocity (due to acoustic pressure) is imposed on a springy diaphragm (mechanical capacitance), which is also a plate on a pre-charged electrical capacitor. Movement of the diaphragm causes the electrical capacitance to vary (ideally as the inverse of the distance between the diaphragm and the fixed plate) thus resulting in a change of the voltage on the capacitor.

2.6.4 Multi-bonds

Multi-bonds (originally known as vector bonds) are a generalisation of the single bond used up to this point, and indicate multiple energy transfers between (multi-port) nodes on the bond graph. The multi-bond is drawn as a large arrow (Figure 2.18) to distinguish it from a single bond, and is treated as a vector of individual bonds.

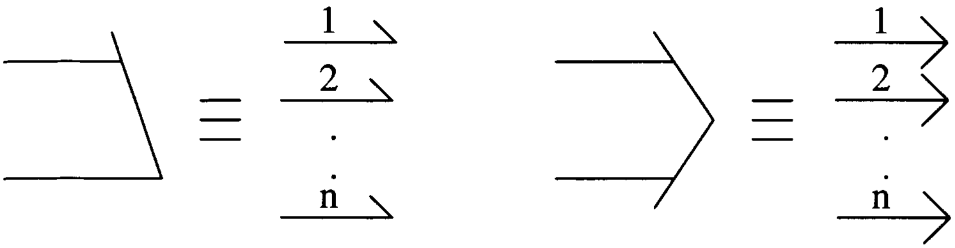


Figure 2.18 Multi-Bonds

Multi-bonds extend the advantage of conciseness and clarity when graphing systems with many multi-port components. A restriction is that all the bonds represented by the multi-bond must have the same causality i.e. the vector of dependent variables must consist entirely of efforts or entirely of flows. Similarly, an activated multi-bond must consist only of signals.

2.7 PSEUDO BOND GRAPHS

Throughout this chapter we have restricted our modelling to systems where energy is the exchange variable, accepting that this may limit the application of the resulting modelling technique. This restriction is overcome by the use of pseudo bond graphs, which provide a means of modelling systems in which the integrated product of the effort and flow variables is not energy. Two examples of the use of pseudo bond graphs are given in the remainder of this section, firstly for analysing manufacturing system dynamics, and then for a heated tank using heat flow as the flow variable (rather than entropy).

2.7.1 A manufacturing system model

Significant work has been done in the field of macro economic modelling using pseudo bond graphs by Brewer *et al.* (1982), where the effort variable is price/unit and the flow variable is the flow rate of a given commodity. The resulting exchange variable is the accumulated price of goods exchanged, i.e. the rate of movement of capital (value rate) is analogous to power in an energy bond graph. In economic systems, the analogy to energy conservation laws is Walras' Law, which states that the sum of the value rates into a port is zero.

Since we are attempting to achieve a continuous model of the system, it is necessary for the flow rate of the commodity to be large enough for aggregation of this flow to be statistically valid. This must be born in mind when modelling manufacturing systems, where the flow variable is typically the flow of produced items throughout the factory.

For this example, we consider a single manufacturing production line for electronic instrumentation consisting of a mechanical package, one basic printed circuit board (PCB), up to n option PCBs, and the associated documentation and packaging. We will assume that demand for the instrumentation is very variable, but delivery times must be low, resulting in the manufacturer building for stock. The pseudo bond graph for this system is shown in Figure 2.19, which combines both elemental components, and hierarchical sub-systems.

The sub-systems are represented as word bond graph nodes, which have specific dynamics associated with the underlying processes. The system input is a flow source representing the demand for the instruments from sales, which is supplied from the finished instrument stores, represented by a capacitance.

In economic bond graphs '1' junctions are used to describe points at which several incremental costs are added to give the overall cost of the item, but the

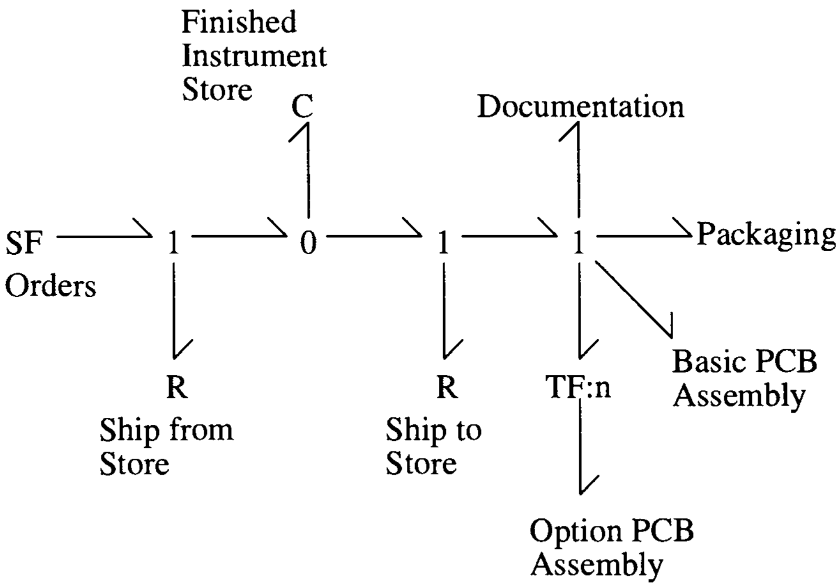


Figure 2.19 Bond graph model of a manufacturing system

flow of items on each attached bond is identical (by application of Walras' law). Thus we can see that the overall cost of the instrument before it passes to the stores is the sum of the costs of all its sub-assemblies, and the process of locating it in the store dissipates additional handling costs.

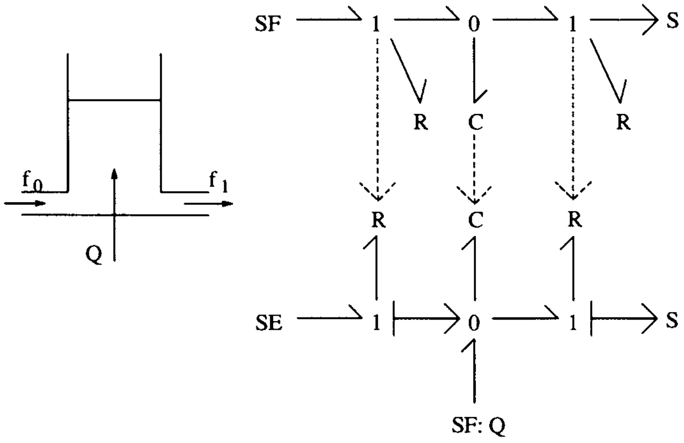
The store itself is linked to a '0' junction, at which the cost remains constant, but the flows into the junction must all add to zero, i.e. the store accumulates the difference between the supply from the production line and the output to sales. Again, the process of handling the instruments between stores and sales incurs a cost represented by the effort across a dissipator. Dissipators in such systems (representing valueless added activities) are typically highly non-linear, the constitutive 'resistance' having high values for small flows. The final unit cost to sales varies according to the demand, being dependent on both the finished instrument cost and the additional handling costs.

The addition of multiple option boards to the instrument is modelled using a transformer which scales the cost on the finished instrument side by n, and scales the flow rate demand on the option board PCB assembly sub-system by the same factor.

This model has not explicitly included an 'I' element, but these occur in macroeconomics, representing investment in capital equipment used to produce higher volumes of equipment more cheaply. The constitutive relation of this inertance results in rapid unit cost increases when the flow suddenly decreases, and vice versa, although the relationship is typically non-linear. Care should be exercised when modelling capital investment in individual manufacturing systems using inertance, since the low level of aggregation may invalidate the model. However, applying integral causality to manufacturing models using inertance to represent investment, does produce interesting qualitative insights whenever causal conflicts occur.

2.7.2 Heated tank model

This example has been chosen to illustrate that it is quite reasonable to model energy transfer systems using pseudo bond graphs. Further, it is possible to mix these with energy bond graphs, as long as the interface between these forms is consistent. Figure 2.20a illustrates the single tank system diagramatically, while Figure 2.20b shows the pseudo bond graph model of the system.



a) Heated tank b) Bond graph of heated tank

Figure 2.20 Heated tank system

The pseudo bond graph model is in two parts; the upper half corresponds to the hydraulic properties (pressure and mass flow), and the lower half corresponds to the thermal properties (temperature and enthalpy flow). These two parts are interfaced via modulated elements in the thermal system - the constitutive relation of the 'R' elements is

$$\dot{h} = (c_p\dot{m})T \tag{2.51}$$

That is, the enthalpy flow $\dot{h}$ is the product of the effort T and the 'conductance' $(c_p\dot{m})$, where the mass flow $\dot{m}$ derived from the hydraulic model modulates this relation. The integral of this relation also gives the constitutive relation of the thermal capacitance, in this case modulated by the state (m) of the hydraulic capacitor

$$h = (c_pm)T \tag{2.52}$$

The modulation for this thermal capacitance is thus shown on the bond graph as a signal from the hydraulic 'C' element, indicating (by the authors' convention) that the state is the modulating variable. The hydraulic capacitance is expressed in the constitutive relation between the effort variable (pressure at base of tank), and the state variable (mass in the tank)

$$p = \frac{gm}{a} \tag{2.53}$$

where g is gravity.

Although the bond graph is equally valid for non-linear relations, for simplicity it is assumed that:

- the fluid is incompressible,
- the tank has constant cross-section (a), and
- the pipes to and from the tank have constant flow resistance (r).

The hydraulic state equation can be derived, using the causality shown, as

$$\begin{aligned} \dot{m} &= f_{in} - f_{out} \\ &= f_{in} - \frac{gm}{ar} \end{aligned} \tag{2.54}$$

The thermal state equation is

$$\dot{h} = c_p f_{in} T_{in} + Q - \frac{h}{m} f_{out} \tag{2.55}$$

2.8 CONCLUSION

This chapter has highlighted the requirements for modelling elementary systems based on their energy transfer characteristics. The bond graph notation has been shown to meet all these requirements, while pseudo bond graphs may be used to model non-energy systems. Choosing energy as the unifying variable permits physical systems covering several energy domains to be modelled in a consistent manner, with pre-defined interfaces.

Separating the model structure from the elemental behaviours permits the model to be easily modified, due to its close mapping onto the actual system structure. This also allows non-linear and time-dependent behaviours to be handled separately in the constitutive relationships of the bond graph elements.

The application of a small set of causality rules permits bond graphs to be analysed systematically, either by hand or using a computer. Causality analysis has been shown to be a very powerful tool not only for deriving different forms of the mathematical model, but also for revealing conflicting system constraints. Due to the importance of this technique, causality concepts are discussed in greater detail in the following chapter.

3

Causality

SUMMARY

- The notion of *computational causality* is illustrated using a simple example.
- The notion of *computational causality* is examined in detail.
- The determination of component causality from the system bond graph is presented.
- The effect of modulated components on causality is discussed.
- Links with wider notions of causality, in particular that arising from the artificial intelligence community, are made.
- Links with wider notions of constraint programming are made.

3.1 INTRODUCTION

It can be argued that causality is fundamental to understanding and modelling systems and so, if such arguments are accepted, it follows that a system modelling methodology should provide a framework within which causality is clearly displayed. Bond graphs provide such a framework, together with an evocative notation.

The notion of "cause", though at first sight obvious, is in fact a slippery concept which has been the subject of much philosophical debate. However, following Simon (Simon, 1952), "we restrict ourselves to a logical, rather than ontological, discussion and hence avoid the rational versus empirical philosophical debate and the corresponding Humean controversy".

3.2 COMPUTATIONAL CAUSALITY: AN EXAMPLE

Summary

- The notion of *computational causality* is illustrated using a simple electrical circuit.
- Three possibilities are exemplified:
  - causal failure
  - causal completeness
  - causal incompleteness
- The relation between causal incompleteness and simultaneous algebraic equations is illustrated.
- Modulated components complicate computational causality.

3.2.1 Solving an electrical circuit

System models (should) imply system behaviour: that is it should be possible to deduce (numerically or symbolically) system variables from system inputs together with the system equations. Loosely speaking, then, system inputs should cause system outputs, given the system model.

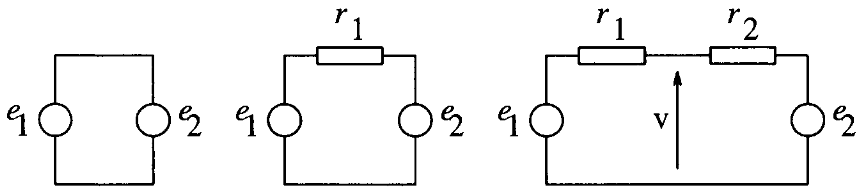


Figure 3.1 Electrical circuits

To fix ideas, consider the electric circuits in Figure 3.1 For the sake of argument, suppose that the two voltage sources are described by the equations

$$e'_1 = 1 \; ; e_2 = 2 \tag{3.1}$$

The left-hand circuit is clearly unsatisfactory: two voltage sources are simultaneously trying to cause the same voltage. This will be called an *over-causal* system. and is an example of causal failure. In algebraic terms, the three equations implied by the circuit

$$e_1 = 1 \; ; e_2 = 2 \; ; e_1 - e_2 = 0 \tag{3.2}$$

are inconsistent; either the system must be remodelled or one of the three equations relaxed.

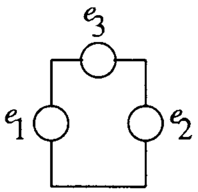


Figure 3.2 Relaxed electrical circuit

For example, a third voltage source could be introduced as in Figure 3.2. The third equation is then replaced by

$$e_1 - e_2 = e_3 \tag{3.3}$$

The middle circuit involves no causal failure and, moreover, it can be explicitly solved for the current i

$$i = \frac{e_1 - e_2}{r} = \frac{1}{r} \tag{3.4}$$

It is said to be *causally complete* or just *causal.*

Writing the equations for the three components in matrix form

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{r} & \frac{1}{r} & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ i \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \tag{3.5}$$

The matrix is lower triangular and thus the three variables (e_1, e_2 and i) can be explicitly computed without further manipulation. We can reasonably say the e_1 and e_2 cause i.

The right-hand circuit also involves no causal failure, but it is not possible to solve for i or v directly from the system equations without further manipulation. It is said to be *under-causal.*

We can express i in terms of v

$$i = \frac{e_1 - v}{r_1} \tag{3.6}$$

and, independently, v in terms of i

$$v = e_2 + r_2 i \tag{3.7}$$

but this pair of equations must be solved simultaneously to deduce i and v . In matrix terms:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{r_1} & 0 & 1 & \frac{1}{r_1} \\ 0 & -1 & 1 & -r_2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ i \\ v \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \end{pmatrix} \tag{3.8}$$

4. Conflicting causality: two components cause the same variable, but the system is physically possible. Defining new variables and additional constraints leads to an system.

3.2.2 Assignment statements and block diagrams

In view of the above example, computational causality is concerned with the order in which variables are computed. Following Breedveld (Breedveld, 1984*b*) this can be described by replacing equations of the form

$$x = y \tag{3.12}$$

with assignment statements of the form

$$x := y \tag{3.13}$$

if y causes x, or

$$y := x \tag{3.14}$$

if x causes y.

The middle circuit can thus be replaced by

$$e_1 := 1;\ e_2 := 2;\ i := (e_1 - e_2)/r \tag{3.15}$$

This algorithm could be executed by a computer as the right-hand sides of the assignment statements are evaluated in preceding statements.

However, the right-hand circuit cannot be directly expressed by assignment statements, but rather would need a numerical routine to solve the simultaneous Equations 3.8.

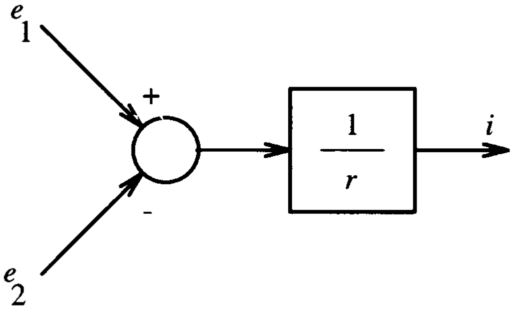


Figure 3.3 Causal electrical circuit: block diagram

Yet another view of computational causality is provided by block diagrams. In the same way as an assignment statement has a left-hand side and a right-hand

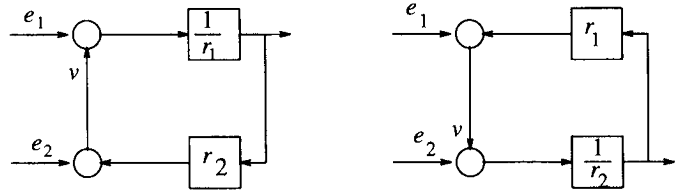


Figure 3.4 Electrical circuit with incomplete causality: block diagram

side, so does a block of a block diagram have an output and an input. Thus the middle circuit has the block diagram of Figure 3.3. This clearly shows the flow of computation.

In contrast, the right-hand circuit does not have complete causality. There is thus more than one possible block diagram - these are shown in Figure 3.4. Each of these contains an implicit algebraic loop which is *not* soluble without the simultaneous solution of algebraic equations.

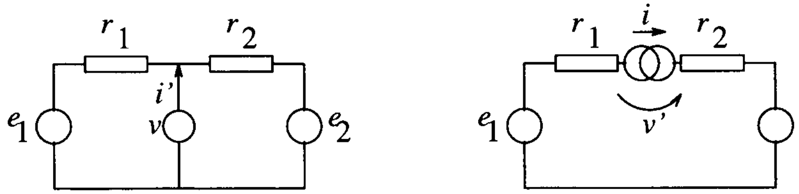


Figure 3.5 Electrical circuits

An alternative approach is to make the algebraic equations *explicit* by addition of appropriate sources. In particular, two possibilities are to

1. add a voltage source as in the left-hand part of Figure 3.5; this will not change the currents in the circuit as long as the current $i' = 0$.
2. add a current source as in the right-hand part of Figure 3.5; this will not change the voltages in the circuit as long as the voltage $v' = 0$.

In each case, the algebraic equation to be solved corresponds to the *constraint* that the additional source *input* is zero.

3.2.3 Modulated components

In the discussion so far, it has been assumed that the components have constant parameters: the source voltages are constant and the resistors have constant resistance. This section illustrates that if, in contrast, component parameters depend on other system variables, then the computational causality, and the corresponding equation solution, is not so obvious.

The matrix is not triangular and so must be manipulated to solve for i and v.

It is reasonable to say that e_1 and e_2 cause v and i, but we cannot say that v is caused by e_1, e_2 and i, as i is itself 'caused' by e_1, e_2 and v. Thus there is a causal loop.

Of course, in this simple case, this difficulty could be avoided by replacing the two resistors by a single equivalent resistor with resistance $r_1 + r_2$, but this would be a remodelling decision rather than a method of solution as such.

There are other ways to formulate the solution of the right-hand electrical circuit. For example, it could be useful to represent both of the two R components to give a current output.

$$\begin{aligned} i &= \frac{1}{r_1}(e_1 - v) \\ i &= \frac{1}{r_1}(v - e_2) \end{aligned} \tag{3.9}$$

(For example, this situation would arise if the two R components were diodes (Dijk and Breedveldt, 1995)) The pair of Equations 3.9 is said to be *over-causal*: each equation 'causes' the current i. This can be resolved by replacing Equations 3.9 by

$$\begin{aligned} i_1 &= \frac{1}{r_1}(e_1 - v) \\ i_2 &= \frac{1}{r_2}(v - e_2) \\ i_1 + i_2 &= 0 \end{aligned} \tag{3.10}$$

The complete set of equations can be written in matrix form as

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{r_1} & 0 & 1 & \frac{1}{r_1} & 0 \\ -\frac{1}{r_2} & 0 & 0 & \frac{1}{r_2} & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ i_1 \\ v \\ i_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.11}$$

Equations 3.10 are *under-causal* in the sense described above.

This example has highlighted four possible types of causal patterns associated with systems:

1. Causal failure: no solution is possible and the system model cannot represent reality. The system is said to be *over-causal.*
2. Complete causality: system variables can be explicitly computed from system inputs and constitutive relations without recourse to algebraic manipulation. The system is said to be *causal.*
3. Incomplete causality: system variables cannot be explicitly computed from system inputs and constitutive relations; simultaneous equations must be solved. The system is said to be *under-causal.*

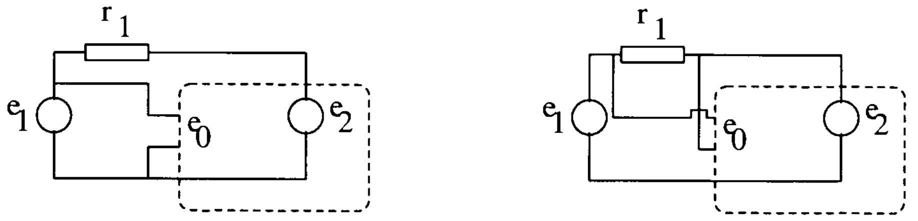


Figure 3.6 Electrical circuits with modulation

As an example, consider the second circuit of Figure 3.1 where the right-hand voltage source is replaced by the output stage of of a voltage amplifier with gain g and input e_0 as in Figure 3.6.

$$e_2 = ge_0 \tag{3.16}$$

Figure 3.6 shows two possible connections:

1. The amplifier input is $e_0 = e_1$, the input source voltage.
2. The amplifier input is $e_0 = e_1 - e_2 = ri$, the voltage across the resistor.

In the first case, this does *not* cause a problem as e_0, and hence e_2, can be directly computed from e_1 which is known. In particular, Equation 3.5 becomes

$$\begin{pmatrix} 1 & 0 & 0 \\ -g & 1 & 0 \\ -\frac{1}{r} & \frac{1}{r} & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ i \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{3.17}$$

This matrix is still lower triangular and so the current i can be directly computed as

$$i = \frac{(1-g)e_1}{r} \tag{3.18}$$

In the second case, however, this *does* cause a problem as e_0, and hence e_2, *cannot* be directly computed from e_1. In particular, Equation 3.5 becomes

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -gr \\ -\frac{1}{r} & \frac{1}{r} & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ i \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{3.19}$$

Equation 3.19 cannot be directly solved and needs algebraic manipulation.

As a further example, suppose that the two sources are fixed, but that the resistor is modulated. Once again, consider two cases:

1. The resistance is modulated by the *voltage* across it according to $r = \frac{\rho}{v_r} = \frac{\rho}{e_1 - e_2}$.
2. The resistance is modulated by the *current* through it according to $r = \frac{\rho}{i}$.

Component	Possible causalities
Effort source	S ⟞
Flow source	S ⟝
R component	R ⟝ or R ⟞
I (integral causality)	I ⟝
I (derivative causality)	I ⟞
C (integral causality)	C ⟞
C (derivative causality)	C ⟝
Transformer	⟝ TF ⟝ or ⟞ TF ⟞
Gyrator	⟞ GY ⟝ or ⟝ GY ⟞
Effort amplifier	⟞ AE ⟞
Flow amplifier	⟝ AF ⟝
Inter junction bond	⟝ or ⟞
Inter junction signal	0 ⟶ or 1 ⟼

Table 3.1 Component causality : summary

Both cases lead to non-linear equations; the first can be solved directly but the second cannot

$$i = \frac{e_1 - e_2}{r} = \frac{(e_1 - e_2)^2}{\rho} \tag{3.20}$$

in the second case

$$i = \frac{e_1 - e_2}{r} = \frac{e_1 - e_2}{i\rho} \tag{3.21}$$

In this particular case, Equation 3.21 may be solved to give

$$i = \sqrt{\frac{e_1 - e_2}{\rho}} \tag{3.22}$$

but, in general, such modulation would require numerical iterative solution.

3.3 BOND GRAPH COMPONENT CAUSALITY

Summary

- The causality constraints implied by bond graph components are considered in greater depth than in Chapter 2.
- These causality constraints are summarised in Table 3.1.

3.3.1 Discussion

Before considering the causality of system of components, it is necessary to consider the causality of individual components.

For the purposes of this book, each component has one or more ports each of which is associated with a bond and hence the two (effort and flow) variables. System causality is concerned with determining which of the two variables associated with each port on each component is the input and which the output. Component causality is concerned with determining what *constraints* on the causality of a component are imposed by the component itself.

In this book, we adopt the convention (Karnopp *et al.*, 1990) that each port is connected to a *junction* via a bond.

3.3.2 Bonds and the causal stroke

Summary

- If one of the two variables (one an effort and one a flow) associated with a bond connected to a component port is an input, then the other is an output
- The evocative notation of the *causal stroke* indicates whether the input is an effort or a flow.

Discussion

As discussed in Chapter 2, components are connected by bonds, and each component port has a bond associated with it.

In view of the discussion on computational causality in Section 3.2. it is important to have a notation to distinguish which of the two variables (one an effort and one a flow) associated with a port is an input and which an output. In principle, if each variable could be either an input or an output there would be four possibilities:

1. flow input, effort output
2. flow output, effort input
3. effort output, flow output
4. effort input, flow input

However, possibilities 3 and 4 will be excluded by making the following *modelling convention*

> *Of the two variables (one an effort and one a flow) associated with a component port, if one is a component input, then the other is a component output.*

There are thus two possibilities remaining: those corresponding to possibilities 1 and 2. This convention is justified for one-port components related by an invertible constitutive relation in Section 3.3.4. Alternatively, physical considerations naturally lead to this convention when using energy bonds (Karnopp *et al.*, 1990).

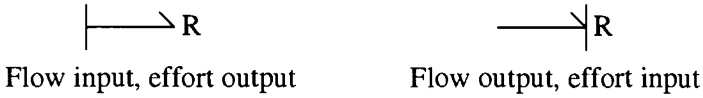


Figure 3.7 The causal stroke

The notation for these two possibilities is given in Figure 3.7, where an R component is used as an example.

3.3.3 Junctions

Summary

The junctions of a bond graph constrain the possible causalities of the components of the bond graph

- One, and only one, bond connected to an *0-junction* has an effort output.
- One, and only one, bond connected to an *1-junction* has a flow output.
- This is illustrated in Figure 3.8 for a four-port 0-junction and in Figure 3.9 for a four-port 1-junction .

Discussion

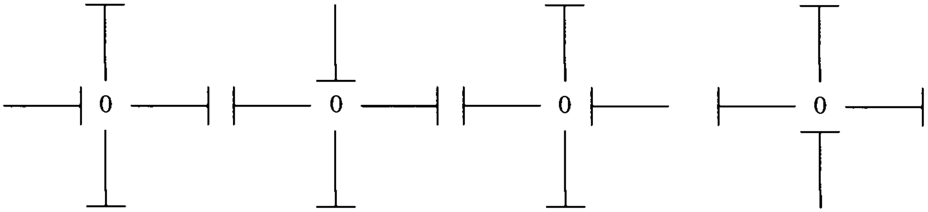


Figure 3.8 0-junction causality

The junctions of a bond graph interconnect the corresponding components. One aspect of this, discussed here, is that each junction constrains the possible causalities of the component ports connected to it. As discussed in Section 2.3.2, a 0-junction with n bonds connected to it constrains all the corresponding effort variables to be equal

$$e_1 = e_2 = \cdots = e_n \tag{3.23}$$

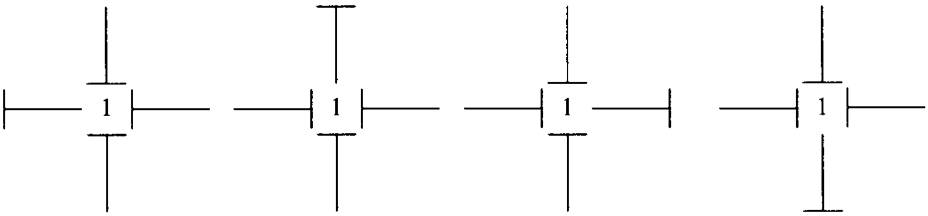


Figure 3.9 1-junction causality

and the signed sum of the flows to be zero

$$\sum_{i=1}^{n} \bar{f}_i = 0 \tag{3.24}$$

where $\bar{f}_i = f_i$ if the bond direction is into the junction and $\bar{f}_i = -f_i$ if the bond direction is out of the junction.

The latter (implicit) equation can be rewritten as an explicit equation in exactly n ways with one output variable $\bar{f}_j$.

$$\bar{f}_j = -\sum_{i \neq j} \bar{f}_i \tag{3.25}$$

Thus a 0-junction can only have one port (the jth) with flow output f_j; the other bonds must have effort output. In the same way, exactly one of the junction-ports (the jth) can have an effort input. As an effort output on the junction corresponds to an effort input on the port at the other end of the bond, the statement given in the summary (item 1) follows.

A similar argument holds for 1-junctions.

The four possible causal patterns for a four port 0-junction is given in Figure 3.8, and for a four-port 1-junction is given in Figure 3.9. (The half-arrows have been omitted to emphasise the independence of sign convention and causality). An n-port junction thus implies n equations: $n-1$ from equations 3.23 and one from Equation 3.25. An n-port junction thus has n possible causal patterns, and this restricts the number of causal patterns on the connected n-ports to n. If these ports were not causally constrained, there would be 2^n possibilities.

3.3.4 One-port components

Summary

- One port components may have either causality unless the constitutive relation does not exist for a particular causality.
- Dynamic one-port components (C and I) have a preferred causality – *integral causality* – leading to explicit state space equations. These preferred causalities are

- C: effort output
- I: flow output

In each case, the opposite causality is termed*derivative causality.*

Discussion

Non-dynamic components (for example Rs) (Section 2.2.3) have potentially, two constitutive relations associated with them relating effort to flow and vice versa

$$e = \phi_e(f)\ ; f = \phi_f(e) \tag{3.26}$$

If $\phi_e(f)$ is invertible

$$\phi_f(e) = \phi_e^{-1}(e) \tag{3.27}$$

and vice versa

If both ϕ_e and ϕ_f exist, there is no causality constraint; but if ϕ_e does not exist, the component cannot have effort output and vice versa.

The dynamic components, C and I, (Section 2.2.3) each have a relation relating effort to integrated flow, and flow to integrated effort respectively.

Thus for the C component

$$e = \phi_e(q); q = \phi_q(e) \tag{3.28}$$

Once again, if both ϕ_e and ϕ_q exist, either causality is permissible, but if either does not exist, then the corresponding causality does not exist.

Including the dynamics, the two causalities each correspond to a pair of equations

$$e = \phi_e(q);\ q = \int_0^t f\, dt + q_o \tag{3.29}$$

$$f = \dot{q};\ q = \phi_q(e) \tag{3.30}$$

From the form of the dynamic equations in each pair, the former (Equation 3.29) is termed *integral causality* and the latter (Equation 3.30) is termed *derivative causality.* This distinction is significant for the form of the dynamic equations arising from the system; this is discussed further in Section 4.6.

The following list displays components which do not have arbitrary causality

- An effort source (Section 2.2.3) has constitutive relation . (see Figure 3.10)

  $$\phi_e(f) = e_0 \tag{3.31}$$

  $\phi_f(e)$ does not exist and so this component must have effort output.

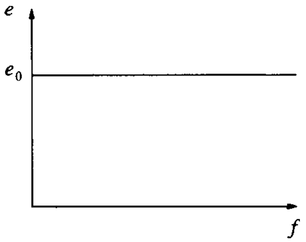


Figure 3.10 Effort source constitutive relation

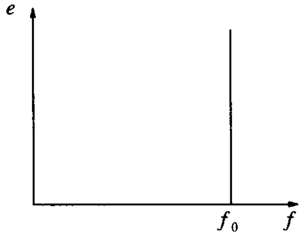


Figure 3.11 Flow source constitutive relation

- A flow source (Section 2.2.3 has constitutive relation (see Figure 3.11)

$$\phi_f(e) = f_0 \tag{3.32}$$

$\phi_e(f)$ does not exist and so this component must have flow output.

- A linear resistance has constitutive relation (see Figure 3.12)

$$\phi_e(f) = rf \tag{3.33}$$

if $r \neq 0$, ϕ_f exists

$$\phi_f(e) = \frac{e}{f} \tag{3.34}$$

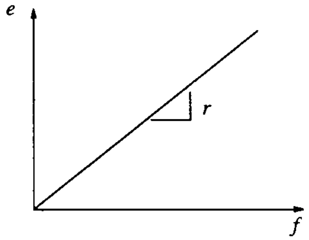


Figure 3.12 Linear resistance constitutive relation

and either causality is possible. But if $r = 0$ only an effort output is permissible, $e = 0$ and is thus independent of f the R component; it becomes a zero effort source.

Conversely, if

$$r = \frac{1}{\sigma} \tag{3.35}$$

then either causality is possible if $\sigma \neq 0$; but if $\sigma = 0$ then only a flow output is possible.

- A linear capacitance (Section2.2.3) has constitutive relation

$$\phi_e(q) = \frac{1}{c}q; \; \phi_q(e) = ce \tag{3.36}$$

If the capacitance $c = 0$ then only flow output is permissible; if $c = \infty$ then only effort output is permissible.

3.3.5 Multi-port components

Multi-port components have more than one energy port. In general, the causality associated with such ports is not independent and so the possible causal forms are not obvious. An algorithm (Section 3.3.5) is given for checking causal possibilities for *linear, non-dynamic* multi-port components.

Discussion

Multi-port components are necessary to model all but the simplest systems. Transformers and gyrators form an important class of two-ports; further examples appear in the rest of this section and in part II of this book.

Each port of an n-port component carries one output and one input (see Section 3.3.2) thus there are, potentially, 2^n possible causal patterns. In the *linear* case, each of these causal patterns potentially has a (matrix) constitutive relation of the form

$$y = \phi u \tag{3.37}$$

where ϕ is an $n \times n$ matrix and y and u are n-vectors of outputs and inputs respectively.

Given the constitutive relation for one causal pattern, we may wish to test whether the constitutive relation for another causal pattern exists and, if so, derive it.

In general, m of the outputs will be the same as before and we will put these into an m-vector y_1 and the rest of the outputs into the $n-m$-vector y_2. A similar decomposition of u is also constructed.

The new output and input vectors can, after rearrangement, be rewritten as

$$Y = \begin{pmatrix} y_1 \\ u_2 \end{pmatrix}; \; U = \begin{pmatrix} u_1 \\ y_2 \end{pmatrix} \tag{3.38}$$

We require the $n \times n$ matrix Φ such that

$$Y = \Phi U \tag{3.39}$$

The following algorithm tests if Φ exists and, if so, computes its value.

Algorithm

1. Choose the vector Y of n outputs and U of n inputs where

$$Y = \begin{pmatrix} y_1 \\ u_2 \end{pmatrix} ; U = \begin{pmatrix} u_1 \\ y_2 \end{pmatrix} \tag{3.40}$$

2. Rearrange the constitutive relation to be of the form:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{3.41}$$

where ϕ_{ij} is the ijth sub matrix of ϕ appropriately partitioned.

3. If ϕ_{22} is singular, then the desired causal form cannot exist.
4. If ϕ_{22} is not singular, then the desired causal form has a constitutive relation

$$\begin{pmatrix} y_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \phi_{11} - \phi_{12}\phi_{22}^{-1}\phi_{21} & \phi_{12}\phi_{22}^{-1} \\ -\phi_{22}^{-1}\phi_{21} & \phi_{22}^{-1} \end{pmatrix} \begin{pmatrix} u_1 \\ y_2 \end{pmatrix} \tag{3.42}$$

That is

$$\Phi = \begin{pmatrix} \phi_{11} - \phi_{12}\phi_{22}^{-1}\phi_{21} & \phi_{12}\phi_{22}^{-1} \\ -\phi_{22}^{-1}\phi_{21} & \phi_{22}^{-1} \end{pmatrix} \tag{3.43}$$

Example: Transformer

A transformer is a two-port device with two inputs and two outputs. There are thus potentially $\frac{4x3}{2} = 6$ ways of choosing the outputs. Two of these are dissallowed by the discussion of Section 3.3.2 leaving potentially four constitutive relation. One of these is

$$\begin{pmatrix} e_2 \\ f_1 \end{pmatrix} = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix} \begin{pmatrix} e_1 \\ f_2 \end{pmatrix} \tag{3.44}$$

We now use our algorithm (Section 3.3.5) to investigate which of the other three possible input-output pairings are possible.

Firstly, try the output combination:

$$Y = \begin{pmatrix} e_1 \\ f_2 \end{pmatrix} \tag{3.45}$$

Then, from Equations 3.38,

$$Y = u_2 = \begin{pmatrix} e_1 \\ f_2 \end{pmatrix} \tag{3.46}$$

and

$$U = y_2 = \begin{pmatrix} e_2 \\ f_1 \end{pmatrix} \tag{3.47}$$

Then

$$\phi_{22} = \phi = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix} \tag{3.48}$$

and

$$\Phi = \phi_{22}^{-1} = \begin{pmatrix} \frac{1}{k} & 0 \\ 0 & \frac{1}{k} \end{pmatrix} \tag{3.49}$$

Thus this constitutive relation exists and so does the corresponding causal form.

However, choosing the output combination

$$Y = \begin{pmatrix} e_2 \\ e_1 \end{pmatrix} \tag{3.50}$$

and input combination

$$U = \begin{pmatrix} f_2 \\ f_1 \end{pmatrix} \tag{3.51}$$

gives

$$y_1 = e_2, u_2 = e_1, u_1 = f_2, y_2 = f_1 \tag{3.52}$$

and so

$$\phi = \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix} \tag{3.53}$$

ϕ_{22} is zero so the corresponding causal form does not exist.

A similar argument holds for the remaining output combination

$$Y = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \tag{3.54}$$

Thus only two possible causalities are possible with this particular component. These are depicted in Figure 3.13. This agrees with the discussion in Section 2.5.

In a similar fashion, only two possible causalities are possible for a gyrator. These are depicted in Figure 3.14. This agrees with the discussion in Section 2.5.

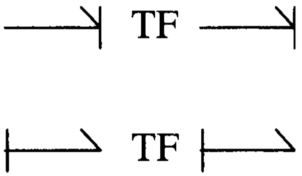


Figure 3.13 Transformer causalities

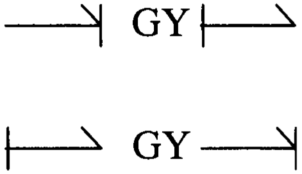


Figure 3.14 Gyrator causalities

Example: ideal amplifiers

An ideal effort amplifier draws zero flow at its input and gives an output effort proportional to its input effort. One of the four (see Section 3.3.5) possible constitutive relations is thus

$$\begin{pmatrix} e_2 \\ f_1 \end{pmatrix} = \begin{pmatrix} k & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} e_1 \\ f_2 \end{pmatrix} \tag{3.55}$$

As in Section 3.3.5, firstly, try the output combination

$$Y = \begin{pmatrix} e_1 \\ f_2 \end{pmatrix} \tag{3.56}$$

As in Section 3.3.5,

$$\phi_{22} = \phi \tag{3.57}$$

But, in this case

$$\phi = \begin{pmatrix} k & 0 \\ 0 & 0 \end{pmatrix} \tag{3.58}$$

which is singular. So this causal form does not exist. As in Section 3.3.5, the other two forms can also be eliminated.

Thus only *one* possible causality is possible with this particular component; it is depicted in Figure 3.15, where AE denotes the two-port effort amplifier component.

In a similar fashion, the two-port *flow* amplifier component has the causality depicted in Figure 3.16.

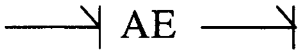


Figure 3.15 Effort amplifier causality

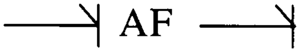


Figure 3.16 Flow amplifier causality

Example: two-port resistor

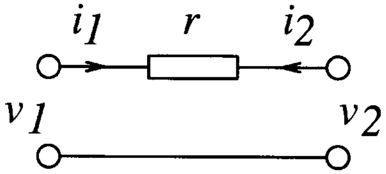


Figure 3.17 Two port resistor: schematic

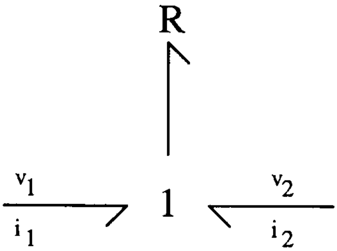


Figure 3.18 Two port resistor: bond graph

Typically, n-port components do not have an explicit internal structure. Nevertheless, n-port components with visible internal structure can be viewed as n-port components and the standard technique applies. Figure 3.17 shows a simple electrical circuit with two-ports, and Figure 3.18 shows the corresponding bond graph.

It is clear from bond graph analysis that one of the 4 potential forms is not possible (that with both currents as input), but the others are.

However let us regard this element as a black box, and start off with the constitutive relation

$$\begin{pmatrix} i_1 \\ i_2 \end{pmatrix} = \frac{1}{r}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{3.59}$$

Choosing

$$u_2 = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \tag{3.60}$$

$$y_2 = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{3.61}$$

gives

$$\phi_{22} = \frac{1}{r}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \tag{3.62}$$

This matrix is singular, so as predicted, the corresponding causal form does not exist.

3.3.6 Inter-junction bonds

Summary

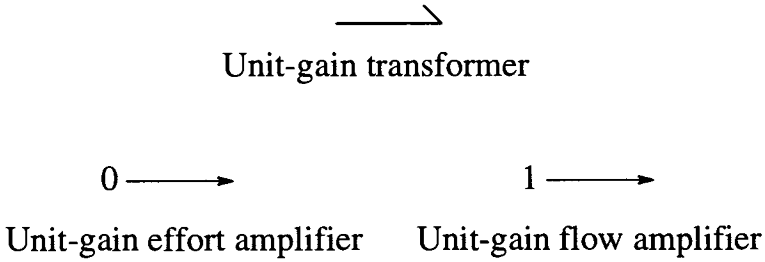


Figure 3.19 Inter-junction bonds

The two sorts of inter-junction bonds of Chapter 2 are interpreted as special cases of two-port components and thus inherit the same causality constraints.

Unit-gain transformers

A *unit-gain* transformer is a special case of the transformer discussed in Section 3.3.5 but with $k = 1$. That is

$$e_2 = e_1 \; ; f_1 = f_2 \tag{3.63}$$

The unit gain transformer is given the special notation of Figure 3.19. As there are only two possible causalities, the usual convention suffices. This is equivalent to a bond connecting two junctions.

Unit-gain gyrators

A *unit-gain* gyrator is a special case of the gyrator. It is sometimes called the *symplectic* gyrator or SGY element.

Unit-gain amplifiers

A *unit-gain* effort amplifier is a special case of the effort amplifier discussed in Section 3.3.5 but with $k = 1$. That is

$$e_2 = e_1 \; ; f_1 = 0 \tag{3.64}$$

The unit gain effort amplifier is assumed always to have its input connected to a 0-junction and is given the special notation of Figure 3.19. There is only *one* possible causality, so this notation is unambiguous.

A similar convention is used for a unit gain flow amplifier.

3.3.7 Sensors and source-sensors

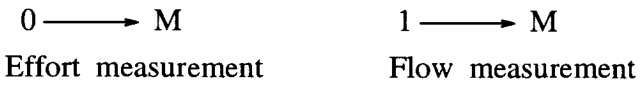


Figure 3.20 Sensors

An ideal sensor measures a system variable without otherwise affecting the system. This is rather like a unit-gain amplifier, and so, in this book, we adopt the notation of Figure 3.20.

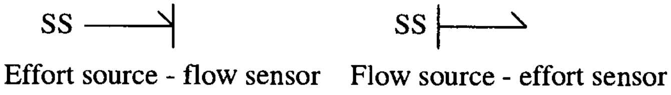


Figure 3.21 Source-sensors

In some cases, system sources and sensors are *collocated*: they correspond to an effort-flow pair. That is, the measured flow is that generated by the corresponding effort source and vice versa.

In this book, the SS (source-sensor) element is introduced with the notation shown in Figure 3.21. The left-hand SS element combines an *effort* source with a *flow* sensor; the right-hand SS element combines a *flow* source with an *effort* sensor.

The source and the sensor each has one of three possible attributes:

- external
- internal
- zero

These attributes have different connotations for the source (Table 3.2) and for the sensor (Table 3.3).

Attribute	Corresponding system input
external	Externally generated
internal	Internally generated for equation solution
zero	zero

Table 3.2 Source attributes

Attribute	Corresponding system output
external	Visible externally
internal	Invisible externally
zero	Constrained to be zero

Table 3.3 Sensor attributes

Internal sources are used with zero sensors as a means of handling under-causal systems (Section 3.4.1); the source-sensor concept is particularly useful for discussing *system inversion* (Chapter 6).

3.4 BOND GRAPH SYSTEM CAUSALITY

Summary

- The junction structure of a system constrains the causality of the system components attached to the structure.
- A system may be:
  - over-causal,
  - causal or
  - under-causal.
- Under-causal systems correspond to differential-algebraic equations.
- A two-stage algorithm for completing causality is given.

3.4.1 Causal constraints

The junction structure of a system (consisting of junctions and interjunction connections) constrains the causality of the system components attached to the structure.

As discussed in Section 3.3.3, each *junction* constrains the causality of the n bonds connected to it: a 1-junction must have precisely one flow output connected to it; a 0-junction must have precisely one effort output connected to it. The inter-junction bonds further constrain the causal possibilities.

For each system being modelled, the *components* can be divided into two sets:

1. those components for which the modeller wishes to impose causality. These will be called *causally prespecified* components.
2. all other components. These will be called *causally reversible* components.

The first class of causally prespecified components would typically contain:

- sources
- those C and I components which the modeller wishes specifically to have integral or derivative causality (Section 3.3.4).
- components whose constitutive relation precludes one form of causality (Section 3.3.4).

The second class of causally reversible components contains all the other components: the modeller does not care about the causality of these.

The constraints, together with the causalities chosen for the first class of components can have three consequences:

1. The causalities assigned to the causally prespecified components are incompatible with the constraints: the system is said to be *over-causal.*
2. The causalities assigned to the causally prespecified components are compatible with the constraints and, given the constraints, there is *only one* causality possible for each of the causally reversible components. The system is said to be *causally complete.*
3. The causalities assigned to the causally prespecified components are compatible with the constraints and, given the constraints, there is *more than one* causality possible for some of the causally reversible components. The system is said to be *under-causal.*

There are algorithms available for determining which of the three consequences holds. The classical Sequential Causality Assignment Procedure (SCAP) is discussed in the textbooks (Karnopp *et al.*, 1990; Wellstead, 1979; Thoma, 1990) and Section 2.5. More recent critical examination is given in a Thesis (van Dijk, 1994) and papers (van Dijk and Breedveld, 1991*a*) (van Dijk and Breedveld, 1991*b*) (van Dijk and Breedveld, 1993) (Borutzky, 1993).

Over-causal systems

An over-causal system is a sign to the modeller that the pattern of preassigned causality is not compatible with the system structure, and therefore that either the causality of these components must be rethought or that the system structure is not physically feasible.

Causal systems

A causal system is neither under nor over-causal. The causality of each component is implied by the prespecified causalities of source components. Such systems can be treated by the classical Sequential Causality Assignment Procedure (SCAP) is discussed in the textbooks (Karnopp *et al.*, 1990; Wellstead, 1979; Thoma, 1990) and Section 2.5 and lead to *ordinary differential equations* (Section 4.9).

Under causal systems

Under causal systems are not described by *ordinary differential equations* (ODEs), but rather by *differential-algebraic equations* (DAEs)(Section 4.7). The following discussion provides a constructive proof of this (Gawthrop and Smith, 1992).

Assuming that the bond graph is proper (all bonds impinge on a junction) then at least one junction (that to which a non-causal bond is attached) does not have causality imposed on it. That is, a causally incomplete 0-junction does not have an effort imposed on it; a causally incomplete 1-junction does not have a flow imposed on it.

An appropriate SS (Effort source for a 0-junction; flow source for a 1-junction) is then attached to the junction and causality propagated. There are then three possibilities for the resultant system:

1. the system is over-causal,
2. the system is causal or
3. the system is under-causal.

In the first case, the method fails and a more sophisticated approach must be used - see for example (van Dijk, 1994) (van Dijk and Breedveld, 1991*a*) (van Dijk and Breedveld, 1991*b*) (van Dijk and Breedveld, 1993) (Borutzky, 1993).

In the second case, the procedure successfully terminates. In the third case the procedure is repeated.

We now have a causal bond graph, but it corresponds to a new system with n new input sources. However, an effort source connected to a 0-junction has no effect on the system if the source effort is such that flow out of the source is zero: the junction is at its 'natural' effort. A similar statement may be made about a flow source on 1-junctions.

The result of this procedure is to add n additional sources, with output v_i and input w_i, to the system. The system thus has n additional inputs w_i which have no effect on the system if they are chosen such that the n implicit algebraic equations

$$w_i = 0 \tag{3.65}$$

are satisfied. These equations form the algebraic part of the system: the rest of the system is causally complete and thus has an ODE representation.

3.5 EXAMPLES

Summary

- The causal completion of some simple systems is illustrated.
- Other simple examples appear in Chapter 2, more complex examples appear in the remaining chapters of Part I and the applications chapters of Part II.
- The use of the bond graph to generate system equations and other models is deferred to Chapter 4.

3.5.1 Computational causality: an example (continued)

Summary

The example of Section 3.2 is re-examined in the light of completing causality via the bond graph algorithm.

Discussion

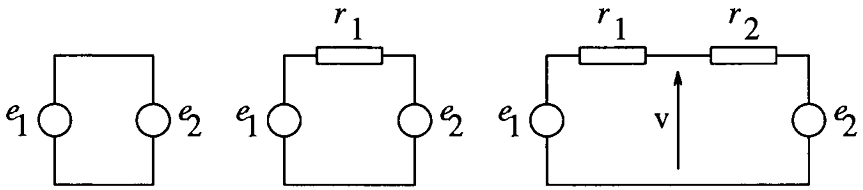


Figure 3.22 Electrical circuits

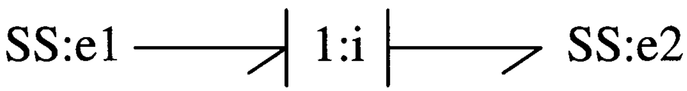


Figure 3.23 Electrical circuit: over-causal

Figure 3.22 repeats Figure 3.1 of Section 3.2. Figures 3.23, 3.24 and 3.25 show the corresponding bond graphs for the three circuits of Figure 3.22. SS components represent the two voltage sources in each case.

The bond graph of Figure 3.23 is over-causal; there is no flow impinging on the 1-junction. This can be relaxed following Figure 3.2; the corresponding bond graph of Figure 3.26 is causally complete.

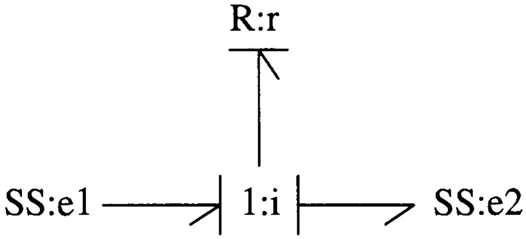


Figure 3.24 Electrical circuit: causal

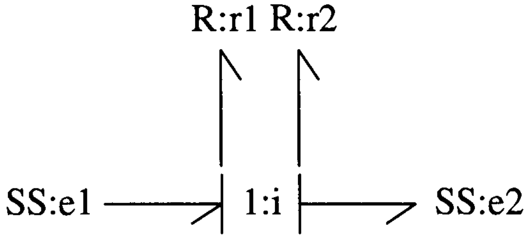


Figure 3.25 Electrical circuit: under-causal

In contrast, the bond graph of Figure 3.24 can be causally completed by assigning flow causality to the single R component.

The bond graph of Figure 3.25 is not causally complete; there is no way to uniquely assign causality to the two R components. This is resolved by adding an additional flow source (with collocated sensor measuring the voltage e_s) as in Figure 3.27. In this case, the algebraic equation

$$e_s = 0 \tag{3.66}$$

has to be solved.

Figure 3.28 corresponds to the alternative formulation of Equation 3.9 where both R components have a current output. The bond graph of Figure 3.28 is over-causal; this is resolved in Figure 3.29 by creating an under-causal diagram and

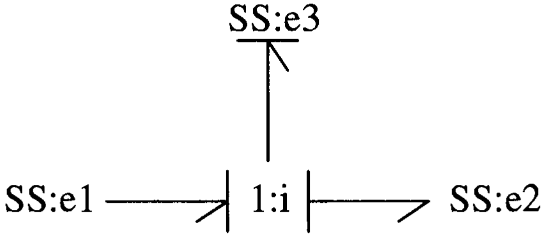


Figure 3.26 Electrical circuit: relaxed constraint

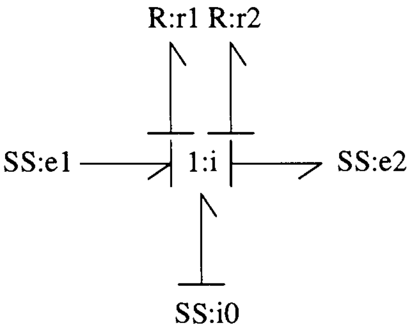


Figure 3.27 Electrical circuit: under-causal with current source

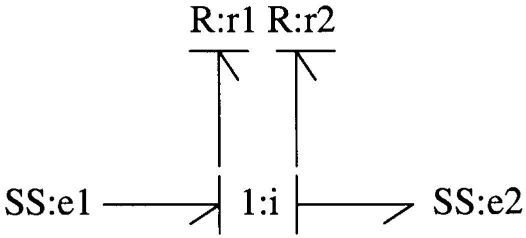


Figure 3.28 Electrical circuit: over-causal

adding a voltage source as in Equation 3.10. The corresponding sensed current $i_s = i_1 - i_2$ is constained by the algebraic equation

$$i_s = 0 \tag{3.67}$$

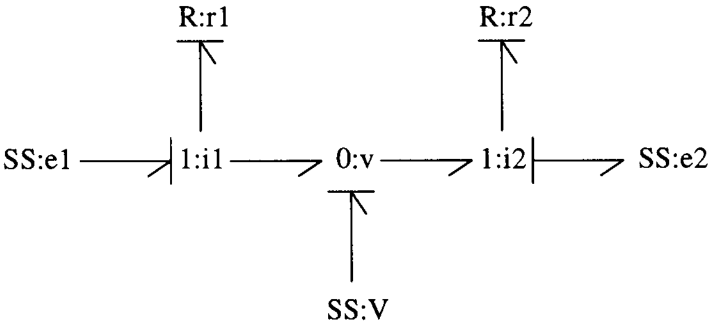


Figure 3.29 Electrical circuit: over-causal with voltage source

3.5.2 An electrical second-order lag

Summary

A simple RC circuit (see Section 2.4.1), and its variations, are used to illustrate causal completion for three cases:

1. a causal system with integral causality.
2. a causal system with mixed integral/derivative causality.
3. an under-causal system with integral causality.

Description

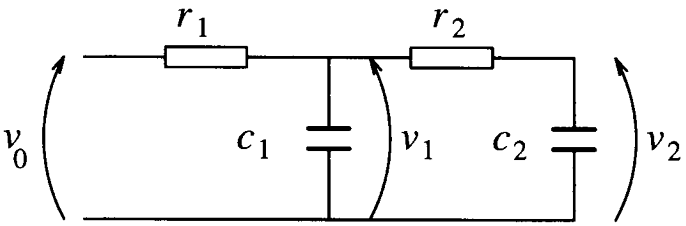


Figure 3.30 RC circuit: schematic

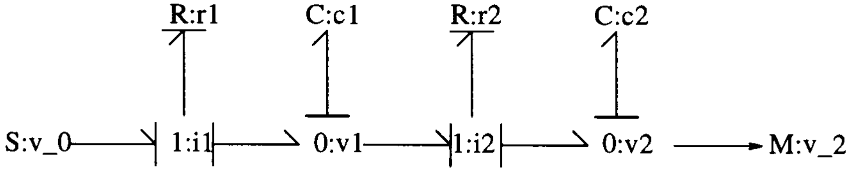


Figure 3.31 RC circuit: bond graph

Figure 3.30 shows a simple two-stage RC filter, with input voltage v_0 and output voltage v_2. The corresponding bond graph appears in Figure 3.31. Causal strokes have been added to the bond graph and causality is complete with integral causality on each of the C components. This can be interpreted as implying that the R elements 'cause' the currents in the 1-junctions and that the C elements 'cause' the voltage on the 0-junctions.

Figure 3.32 shows a modified version of Figure 3.30 where one of the resistors (r_2) has been removed. The corresponding bond graph appears in Figure 3.33. Integral causality is *not* possible on both C elements as the 1-junction on which

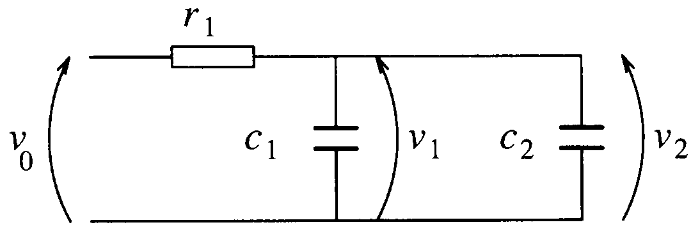


Figure 3.32 RC circuit with resistor removed: schematic

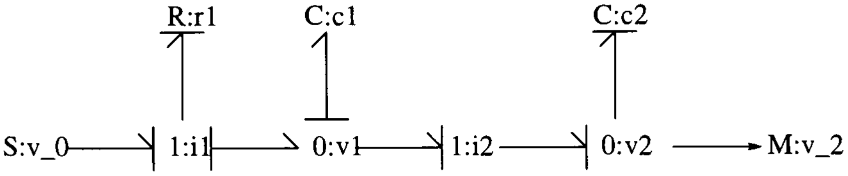


Figure 3.33 RC circuit with resistor removed: bond graph

r_2 used to live now must impose opposite causalities on the two neighbouring 0-junctions. Physically, they share the same voltage $v_1 = v_2$ and so the corresponding charges (states) cannot be independent.

Figure 3.33 shows one possibility, c_1 has integral causality and c_2 has derivative causality. Figure 3.34 shows the other possibility: c_2 has integral causality and c_1 has derivative causality.

Figure 3.35 shows a modified version of Figure 3.30 where one of the capacitors (c_1) has been removed. The corresponding bond graph appears in Figure 3.36. It is not possible to propagate causality further, so a voltage source is added as in Figure 3.37. Causality can now be completed - it is, of course the same pattern as that in Figure 3.30.

As discussed in Section 2.4.1, a buffer amplifier can be used to decouple the two stages of this RC circuit; this is depicted in Figure 3.38. The corresponding causal bond graph appears in Figure 3.39. The causal properties are identical to those of the system without the buffer amplifier.

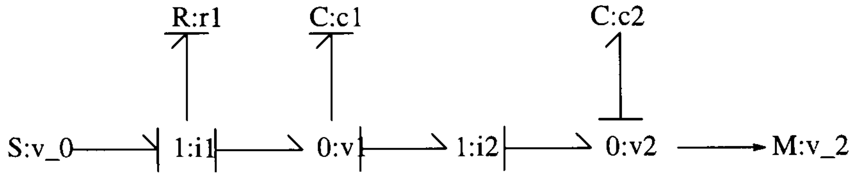


Figure 3.34 RC circuit with resistor removed: alternative bond graph

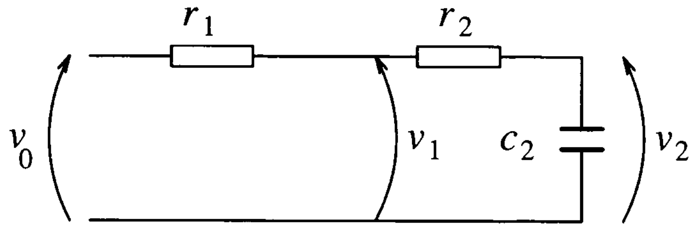


Figure 3.35 RC circuit with capacitor removed: schematic

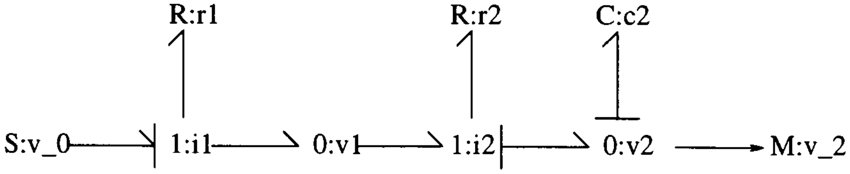


Figure 3.36 RC circuit with capacitor removed: bond graph

3.5.3 DC motor

Summary

A simple DC motor (see Section 2.4.3) is used to illustrate causal completion for two cases:

1. A voltage driven motor with integral causality.
2. A current driven motor with mixed integral/derivative causality.

Description

Figure 3.40 shows the bond graph corresponding to a simple DC motor with *voltage* drive; both I components have integral causality.

Figure 3.41 shows the bond graph corresponding to a simple DC motor with *current* drive; the I component labelled la (the armature inductance) is forced to have derivative causality. This example is aslo discussed in Section 2.5.4.

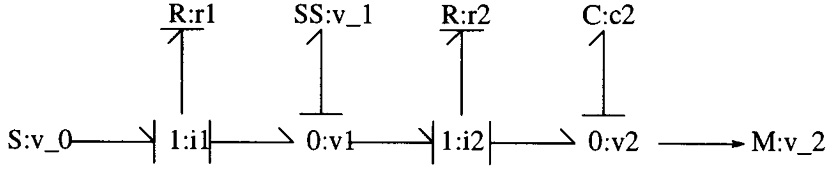


Figure 3.37 RC circuit with capacitor removed: modified bond graph

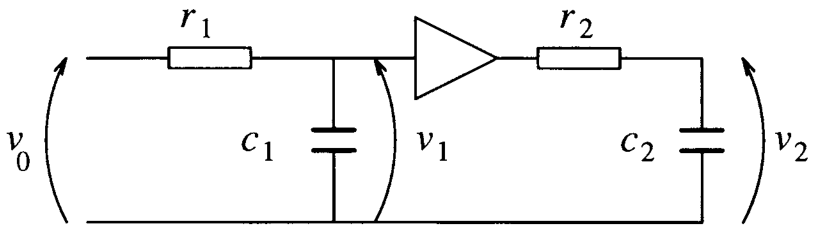


Figure 3.38 An electrical second-order lag with buffer: schematic

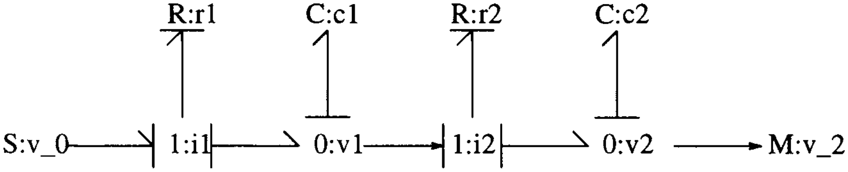


Figure 3.39 An Electrical second-order lag with buffer: bond graph

3.5.4 RLC circuit

Summary

A simple RLC circuit is used to illustrate the completion of causality when only one-port and junctions are involved. For this example, causality is completed in three ways:

1. All integral causality leads to incomplete causality. A voltage source is used to complete causality.
2. All derivative causality leads to a causally complete system.
3. Mixed causality leads to a causally complete system.

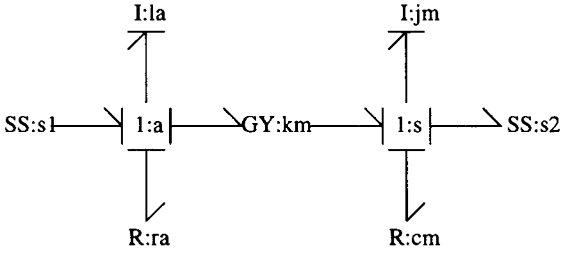


Figure 3.40 DC motor with voltage drive: bond graph

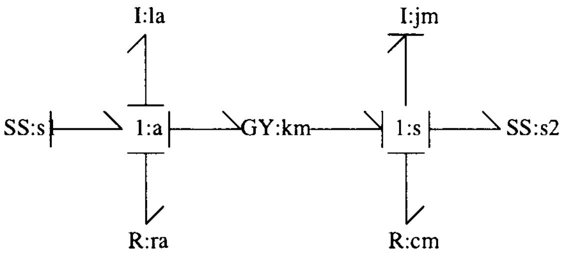


Figure 3.41 DC motor with current drive: bond graph

Description

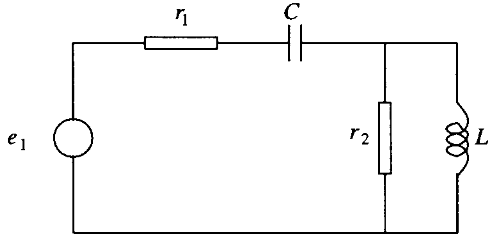


Figure 3.42 RLC circuit: schematic

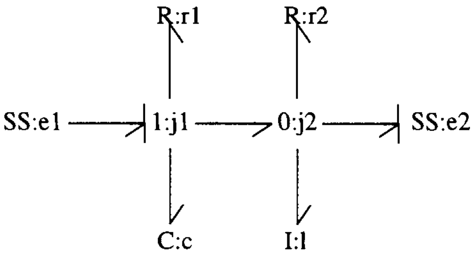


Figure 3.43 RLC circuit: bond graph

This example is taken from Karnopp, Margolis and Rosenberg (Karnopp *et al.*, 1990). The schematic diagram appears in Figure 3.42; the corresponding acausal bond graph appears in Figure 3.43. The source of the SS component e1 provides the input voltage; the sensor of the SS component e2 provides the output voltage.

All integral causality

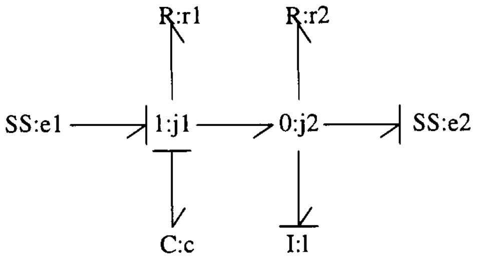


Figure 3.44 RLC circuit: bond graph with integral causality

Following Karnopp, Margolis and Rosenberg (Karnopp *et al.*, 1990) let us first attempt to complete causality with all integral causality. Noting that the effort source must have effort output, (section 3.3.4) the assumption of integral causality gives Figure 3.44.

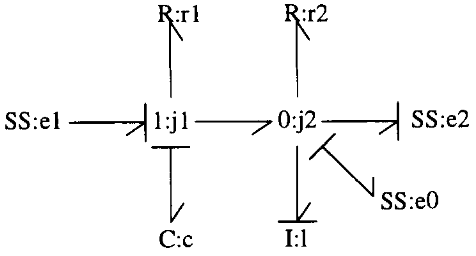


Figure 3.45 RLC circuit: bond graph with additional voltage source

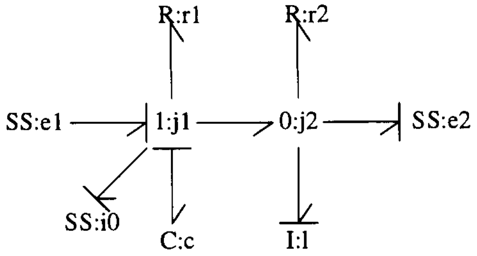


Figure 3.46 RLC circuit: bond graph with additional current source

The bond graph is not causally complete; the junctions do not constrain the causality. Hence the system corresponds to a DAE not an ODE. Following the procedure in Section 3.4, either a voltage source is appended to the right-hand junction or a current source is added to the left-hand junction. Making the former choice leads to Figure 3.45.

The diagram is now causally complete; the additional source indicates an explicit algebraic equation to be solved: the current into the voltage source must be zero.

Alternatively, a *current* source can be added to the left-hand junction as in Figure 3.46. Once again, the diagram is now causally complete; the additional source indicates an explicit algebraic equation to be solved: the voltage into the current source must be zero.

All derivative causality

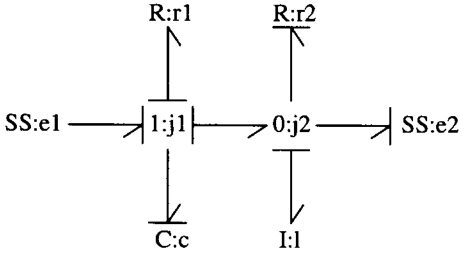


Figure 3.47 RLC circuit: bond graph with derivative causality

As an alternative, let us try to causally complete the system with the derivative causality. As the capacitor now imposes a current onto the 1-junction, and the inductor imposes a voltage onto the 0-junction, the diagram (Figure 3.47) is now causally complete.

It follows that the corresponding set of equations is a set of integral equations.

Mixed causality

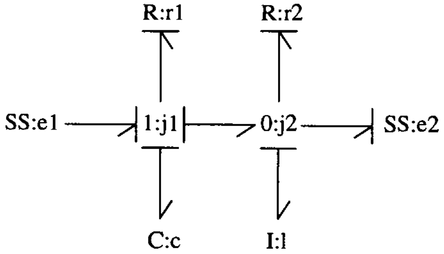


Figure 3.48 RLC circuit: bond graph with mixed causality

As a final permutation, let us try mixed causality with integral causality on the C and derivative causality on the I. This gives Figure 3.48 and, once again, causality is complete.

3.6 QUALITATIVE CAUSALITY

Summary

- Links between bond graph causality and the qualitative notions of causality, arising from the artificial intelligence community, are made.

3.6.1 Discussion

There has been an upsurge of interest in the past few years in the qualitative, rather than the quantitative, description of systems. It can be argued that a qualitative analysis of general system properties should precede quantitative analysis pertaining to specific system parameters.

Bond graphs clearly distinguish between system structure and constitutive relationships and the former aspect is essentially a qualitative description. Bond graphs also provide a framework for discussing causality and so in principle provide a context for the discussion of causality.

As stated in the introduction to this chapter, causality has much broader connotations than just computational causality. The purpose of this Section is to place causality in a broader context and thus make a link with more general causal notions. In particular, we make links with the work of Simon and Rescher (Simon and Rescher, 1966) (based on the earlier work of Simon(Simon, 1952)) and the more recent artificial intelligence work on causality in device behaviour, for example Iwasaki and Simon (1986).

3.6.2 The Causality of Simon and Rescher

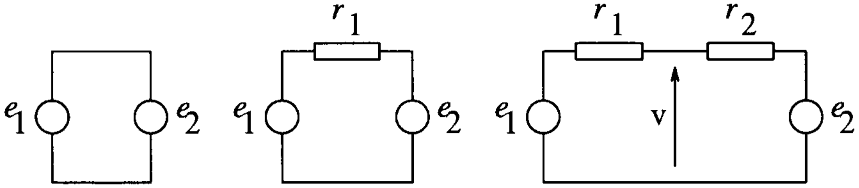


Figure 3.49 Electrical circuits

Simon and Rescher (1966) make two insightful statements about causality:

1. Causality is not logically the same as implication: from "a implies b" it follows that "not b implies not a" but "a causes b" does not lead to "not b causes not a".
2. Causality is not a relation between values of variables, but rather a functional relationship between the variables.

Simon and Rescher (1966) go on to give an explanation of causality in terms of the so-called *structure matrix.* The structure matrix has one row for each system output and one column for each system input. If there is a causal relation between the jth input and the ith output, then the ijth element of the structure matrix is 1, otherwise, the ijth element of the structure matrix is 0. A structure matrix can be analysed in terms of *self-contained* submatrices. We shall illustrate this matrix in terms of the example of Section 3.2, the relevant figure of which is repeated here as Figure 3.49.

The centre circuit has a structure matrix obtained from the matrix on the right-hand side of Equation 3.5 by replacing all non-zero elements by 1

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{3.68}$$

The structure matrix is then decomposed into 'self-contained' sub structures – self-contained meaning that there are exactly enough equations to solve (in principle) for the unknowns. The first and second rows are clearly self-contained: the first gives e_1, the second e_2. The knowns are now substituted into the final row giving the value for the remaining unknown i.

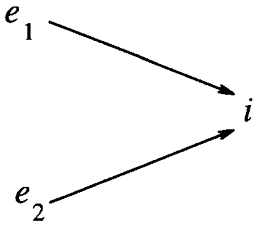


Figure 3.50 Causality diagram

Simon and Rescher (Simon and Rescher, 1966) use a *causality diagram* to depict system causality. An arrow from A to B is to be read as "A causes B". The structure matrix of Equation 3.68 leads to the causality diagram of Figure 3.50. This should be compared topologically with the corresponding block diagram of Figure 3.3.

Turning now to the right-hand circuit, the structure matrix corresponding to

Equation 3.8 is given by Equation 3.69

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \tag{3.69}$$

Proceeding as before, the first two rows each form a complete structure, but neither of the last two rows do. However, having eliminated the first two columns (determined by the first two rows) the last two rows have the structure matrix

$$s_{34} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \tag{3.70}$$

This is complete as it corresponds to two simultaneous equations in two variables. Thus i and v are jointly determined by e_1 and e_2.

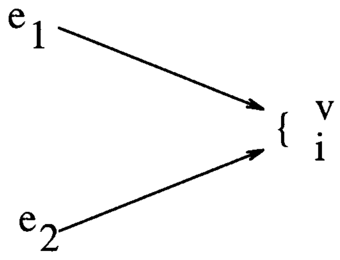


Figure 3.51 Causality diagram

The causality diagram is given in Figure 3.51.

As illustrated by these examples, then, the structure diagram approach to causality gives precisely the same causality diagrams as causal completion of a bond graph. The bond graph thus gives an alternative framework for discussing causality for these systems which do have a bond graph representation. Thus, in this sense, bond graph causality has a wider interpretation than just computation and thus may be used for causally reasoning about systems.

3.6.3 Device causality

The causality of a bathtub has become a standard test case for discussions in the artificial intelligence community. For example, Iwasaki and Simon (1986), gives a detailed discussion of causality for various versions of the bathtub including equilibrium and with various levels of sophistication in representing the dynamics.

This Section presents the bathtub, and its causality from the bond graph point of view. As with the previous Section, this provides a link with other notions of causality as well as providing an illustration of the utility of bond graphs in providing a context for discussion of causality.

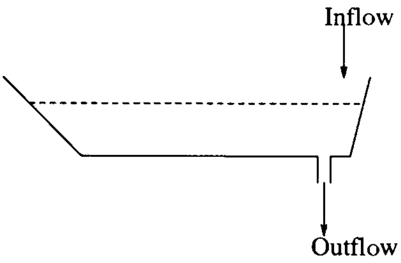


Figure 3.52 A bathtub

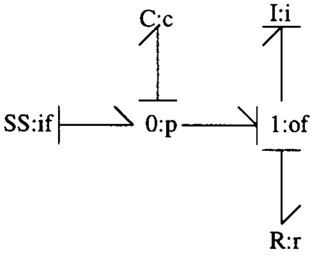


Figure 3.53 A bathtub: bond graph

The bathtub depicted in Figure 3.52 is modelled by the bond graph of Figure 3.53.

The tap has been modelled as a flow source – the flow is independent of the bathtub level.

The bath has been modelled as a C element (not necessarily linear, the bath may not have vertical sides) thus neglecting the velocity of liquid in the bath itself.

The outflow has been modelled as a combination of a (non-dynamic) resistance to flow together with an inertia representing the mass of liquid in the pipe that needs to be accelerated. Figure 3.53 has been completed with integral causality. Following the causal strokes gives the causality diagram of Figure 3.54, where, as in Figure 3.50, an arrow from A to B is to be read as "A causes B". In the linear case, corresponding equations are

$$\dot{x}_1 = \frac{(iu_1 - x_2)}{i} \tag{3.71}$$

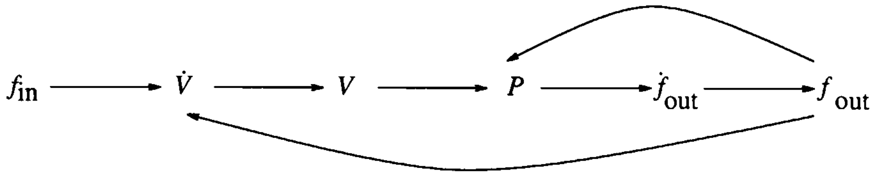


Figure 3.54 Bathtub causality

$$\dot{x}_2 = \frac{(-(crx_2 - ix_1))}{(ci)} \tag{3.72}$$

$$y_1 = \frac{x_1}{c} \tag{3.73}$$

where $x_1 = v$ is the volume of liquid in the bath, $x_2 = q$ is the outflow rate, $u_1 = q_0$ is the inflow, $y = p$ is the pressure at the base of the bathtub and c, i and r are the corresponding coefficients of the C, I and R components. The transfer function relating pressure to inflow is

$$G(s) = \frac{r + is}{1 + crs + cis^2} \tag{3.74}$$

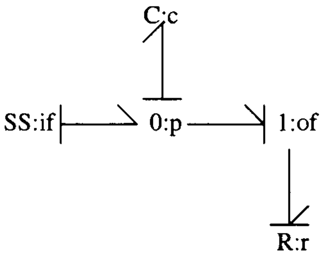


Figure 3.55 Simplified bathtub: bond graph

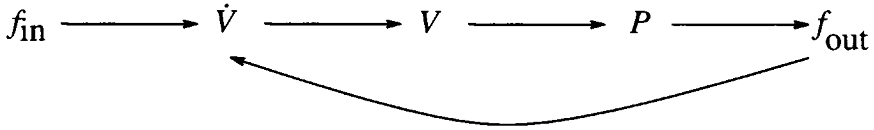


Figure 3.56 Simplified bathtub causality

A simplified version of this model could neglect the outflow dynamics by assuming that q_{out} responds rapidly to changes in pressure. In bond graph terms, the I component is deleted from Figure 3.53 to give Figure 3.55. The causality on the outflow resistance now changes: the outflow is caused by the pressure in the bathtub acting across the R element. In the linear case, corresponding equations are

$$\dot{x}_1 = \frac{(cru_1 - x_1)}{(cr)} \tag{3.75}$$

$$y_1 = \frac{x_1}{c} \tag{3.76}$$

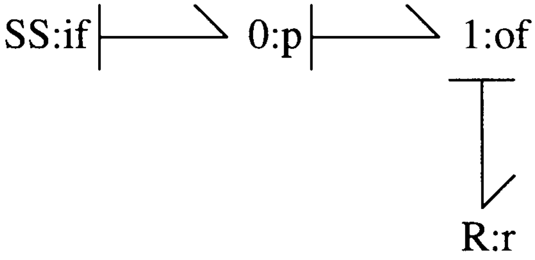


Figure 3.57 Equilibrium bathtub: bond graph

$$\begin{tikzcd} f_{\text{in}} \arrow[r] & f_{\text{out}} \arrow[r] & P \arrow[r] & V \end{tikzcd}$$

Figure 3.58 Equilibrium bathtub causality

The transfer function relating pressure to inflow is

$$G(s) = \frac{r}{1 + crs} \tag{3.77}$$

The causal ordering is now given by Figure 3.56

Finally, following Iwasaki and Simon (1986), it is of interest to consider the equilibrium (or steady-state) situation. This is essentially equivalent to removing the C, (as well as the I) component. Figure 3.57 shows the corresponding bond graph. Note that, as the C no longer imposes causality on its 0-junction, the overall system causality is completely changed: the flow throughout the system is determined by the inflow, and thus the pressure is determined by the outflow.

In the linear case, corresponding equations are

$$y_1 = ru_1 \tag{3.78}$$

The transfer function relating pressure to inflow is

$$G(s) = r \tag{3.79}$$

The corresponding causality diagram appears in Figure 3.58.

In the light of this example, it can be seen that when systems can be modelled by bond graphs, the causality implied by the bond graph corresponds to the notions of causality espoused by the artificial intelligence community.

3.7 CONSTRAINTS AND CONSTRAINT PROPAGATION

Summary

- Links are made betwen causality and the notion of *constraints* and the corresponding *constraint programming languages* arising from the artificial intelligence and computing science communities.

3.7.1 Discussion

Constraints(Winston, 1984; Leler, 1988), and their corresponding constraint programming languages (Leler, 1988; Fattah, 1992), form a well-developed part of computing science. As in Section 3.6.3, the aim of this Section is to make links with the notion of bond graph causality and the corresponding notion of constraint programming. This has recently been discussed by El Fattah (Fattah, 1995).

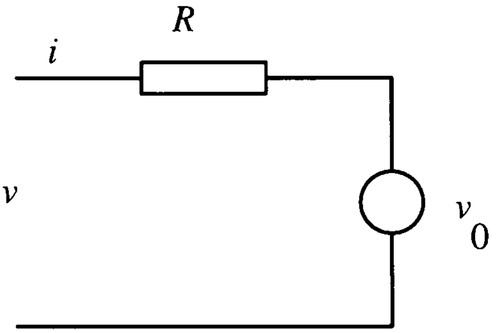


Figure 3.59 An electrical circuit

To fix ideas, consider the electrical circuit of Figure 3.59, consisting of a resistor and a voltage source. From the constraint point of view, this circuit can be seen as imposing a *global constraint* relating v and i

$$v = Ri + v_0 \tag{3.80}$$

made up from the *local constraints* arising from the individual components and their interconnections

$$\begin{aligned} v_r &= Ri_r \\ v_s &= v_0 \\ i &= i_r \\ v &= v_r + v_s \end{aligned} \tag{3.81}$$

These equations are *constraints* in the sense that equality, not assignment, is implied and the order of the equations is immaterial. Thus the constraints could, for example, be rearranged as

$$\begin{aligned} v_r - Ri_r &= 0 \\ v_s - v_0 &= 0 \\ i - i_r &= 0 \end{aligned}$$

$$v - v_r - v_s = 0 \tag{3.82}$$

The global constraint Equation 3.80 can be used to:

- compute v (given i, v_0 and R) from the assignment

$$v := Ri + v_0 \tag{3.83}$$

- compute i (given v, v_0 and R) from the assignment

$$i := \frac{v - v_0}{R} \tag{3.84}$$

- compute v_0 (given v, i and R) from the assignment

$$v_0 := v - Ri \tag{3.85}$$

- compute R (given v, i and v_0) from the assignment

$$R := \frac{v - v_0}{i} \tag{3.86}$$

That is, the single constraint 3.80 can be used in four different ways to compute unknown variables from known variables.

Global constraints such as 3.80 can be deduced, in this case, from *local* constraints such as those listed in equations 3.81. As discussed in the references (Winston, 1984; Leler, 1988), *local propagation* is the simplest technique to deduce global information from local constraints. Briefly the algorithm is as follows:

```
REPEAT
  Pick a constraint with exactly one unknown
  Compute the single unknown using the appropriate assignment statement
  Propagate the computed value to all other constraints
UNTIL desired value found
```

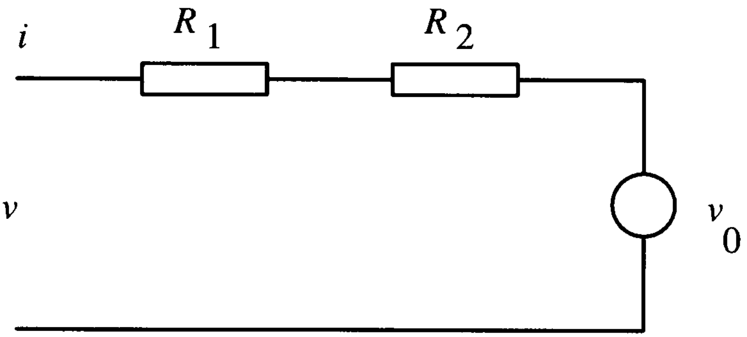


Figure 3.60 An electrical circuit where local propagation fails

Unfortunately, this algorithm may sometimes fail. The circuit of Figure 3.60 is equivalent to the local constraints

$$\begin{aligned} v_1 &= R_1 i_1 \\ v_2 &= R_2 i_2 \\ v_s &= v_0 \\ i &= i_1 \\ i &= i_2 \\ v &= v_1 + v_2 + v_s \end{aligned} \tag{3.87}$$

For example, if i, v_0 and R are all known, then v can be deduced from the following series of assignment statements

$$\begin{aligned} v_s &:= v_0 \\ i_1 &:= i \\ i_2 &:= i \\ v_1 &:= R_1 i \\ v_2 &:= R_2 i \\ v &:= v_1 + v_2 + v_s \end{aligned} \tag{3.88}$$

But if v, v_0 and R, i *cannot* be deduced by this local constraint propagation. The problem is that the following set of constraints

$$\begin{aligned} v_1 &= R_1 i \\ v_2 &= R_2 i \\ v &= v_1 + v_2 + v_s \end{aligned} \tag{3.89}$$

each has *two* unknowns and so the algorithm fails at this stage. Note, however, that these three equations in three unknowns (i, v_1 and v_2) do have a solution, but simultaneous algebraic equations must be solved.

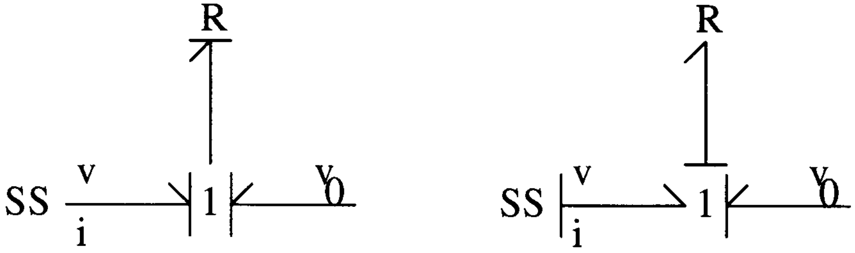


Figure 3.61 An electrical circuit bond graph

To see the relationship of constraint propagation to bond graph causality, it is illuminating to consider the bond graphs of the two preceding examples.

Figure 3.61 shows the bond graph of the electrical circuit of Figure 3.59; the left-hand circuit corresponds to known voltage (the SS output) and unknown current (the SS input) and the right-hand circuit corresponds to known current (the SS output) and unknown voltage (the SS input). Because each bond graph is causally complete, the equations of each component can be written down as assignment statements in such a way that the appropriate output can be computed.

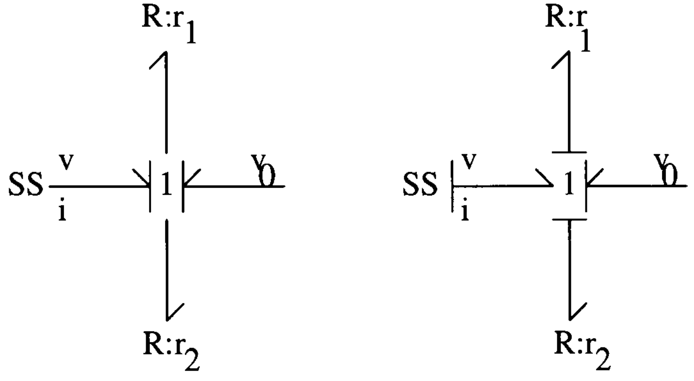


Figure 3.62 An electrical circuit bond graph

Figure 3.62 shows the bond graph of the electrical circuit of Figure 3.60; the left-hand circuit corresponds to known voltage (the SS output) and unknown current (the SS input) and the right-hand circuit corresponds to known current (the SS output) and unknown voltage (the SS input). In this case, the left-hand bond graph *cannot* be causally completed and so the equations of each component *cannot* be written down as assignment statements in such a way that the appropr'ate output can be computed.

A standard example in the constraint literature is the conversion of degrees Celsius to degrees Farenheit using the constraint equation

$$C = \alpha F + \beta \tag{3.90}$$

where α and β are the conversion constants and C and F are temperatures in Celsius and Farenheit respectively. This constraint equation is algebraically the same as Equation 3.80. Thus, in principle, an equivalent bond-graph may be constructed - but it does not have a clear physical meaning. Indeed, there seems to be no systematic way of generating the bond graph corresponding to arbitrary sets of constraint equations.

From these examples, it follows that there is a close connection between the constraint programming technique and bond graphs when applied to physical systems. In particular:

- The interpretation of an equation as a constraint implying a number of possible assignment statements, the choice of which is not prespecified, is equivalent to the acausal bond graph representation.

- If variables (as opposed to parameters) are regarded as system outputs then the set of ordered assignment statements needed to deduce an output value from known inputs and parameters is equivalent to the causal bond graph representation.
- The local constraint propagation algorithm is equivalent to the algorithm for causally completing a bond graph. Both algorithms get stuck for precisely the same reasons.

There are, however, some differences:

- Constraint programming applies to sets of equations which do not necessarily arise from physical systems.
- Constraint programming can regard parameters as unknown 'outputs'; standard bond graphs cannot.
- Bond graphs give a nice representation and a clear interpretation of the causally incomplete situation.

4

System representations and transformations

SUMMARY

- Various *system representations* can be *automatically* derived from the system bond graph including
  - Ordered elementary system equations
  - Differential-algebraic equations (DAE) including special cases:
    - Algebraic equations
    - Ordinary differential equations (ODE)
    - Constrained-state equations
    - Semi-explicit differential-algebraic equations
  - Linearised descriptor equations
  - Transfer functions
  - Simulation code
- Different types of causality give rise to different equation formulations.
- Each such representation has a *use* to which it is appropriate.

4.1 INTRODUCTION

Derived system representations can be automatically derived from system bond graphs. They provide alternative ways of looking at a system and, as such, are useful for specific aspects of analysis and synthesis of dynamic systems.

Each derived system representation has a *use*: for example,

- control design,
- system design,
- system simulation and
- system understanding.

- Physical system
- $Transformation_1 \Rightarrow Representation_1$
- $Transformation_2 \Rightarrow Representation_2$
- ...
- $Transformation_N \Rightarrow$ Core representation
- $Transformation_{N+1} \Rightarrow Representation_{N+1}$
- $Transformation_{N+2} \Rightarrow Representation_{N+2}$
- ...
- $Transformation_{N+M} \Rightarrow$ Model

Figure 4.1 System modelling: transformations

System modelling, the procedure for arriving at an appropriate (for its use) model can be viewed as a sequence of *transformations* between system *representations* as indicated in Figure 4.1. The start of this chain of transformations is the physical system; an intermediate representation is the *core* representation; the final representation is the system model in an appropriate form. The core representation used in this book is the system bond graph.

With reference to Figure 4.1, this Chapter discusses a number of derived representations together with the appropriate transformations. In particular, the following derived representations are considered:

- acausal bond graph: graphical representation (Section 4.2)
- acausal bond graph: list representation (Section 4.3);
- causal bond graph: graphical representation (Section 4.4);
- causal bond graph: list representation (Section 4.5);
- ordered elementary system equations (Section 4.6);
- differential-algebraic equations (DAE) (Section 4.7) including the special cases:
  - algebraic equations (Section 4.8)
  - ordinary differential equations (ODE) (Section 4.9)
  - constrained-state equations (Section 4.10)
  - semi-explicit differential-algebraic equations (Section 4.11);
- linearised descriptor equations (Section 4.12);
- transfer functions (Section 4.13);
- simulation code (Section 4.14).

A set of modelling transformation tools (MTT) have been developed (Gawthrop *et al.*, 1991*a*; Gawthrop, 1995). The discussion in this chapter is based on some of the MTT implementation; but does not constitute a full description of MTT. Rather, it gives an idea of the sort of implementation issues that are involved in computer manipulation of bond graphs.

An example, described in Section 4.1.1, is used throughout to illustrate the various representations. Further examples, drawn from Chapter 3, are collected in Section 4.15.

4.1.1 Example

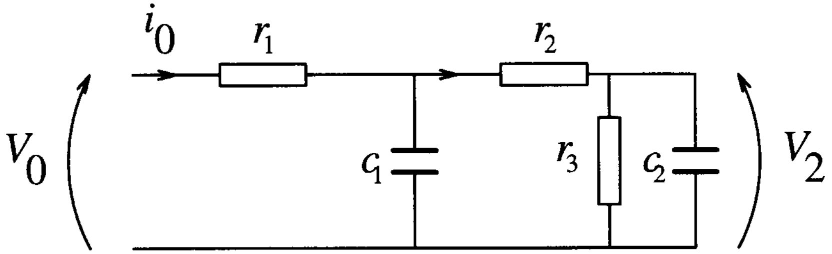


Figure 4.2 Electrical circuit

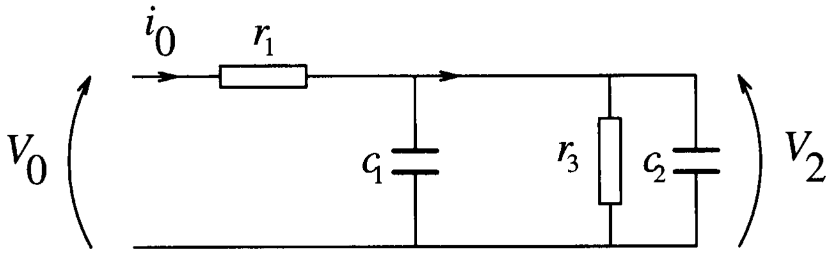


Figure 4.3 Electrical circuit: resistor removed

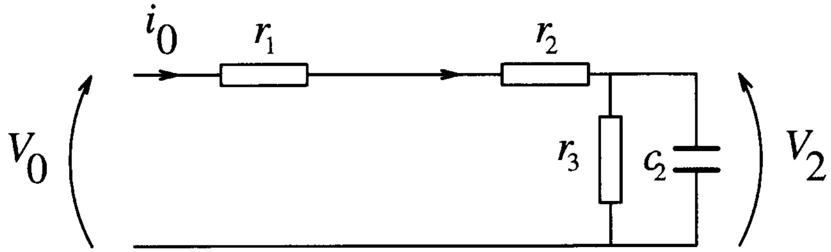


Figure 4.4 Electrical circuit: capacitor removed

A simple, linear, example is chosen to illustrate the various representations discussed in this chapter. It corresponds closely to the example of Section 3.5.2. The remaining examples from Chapter 3 are collected together at the end of this chapter in Section 4.15. More complex examples are considered in subsequent chapters.

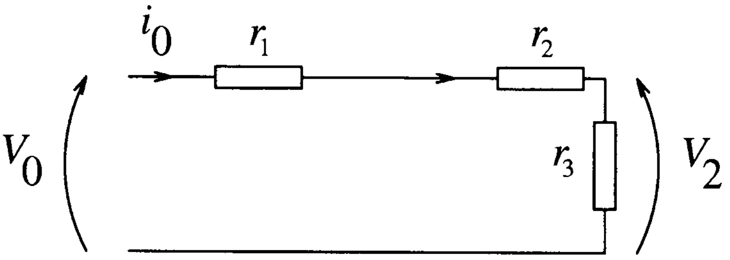


Figure 4.5 Electrical circuit: two capacitors removed

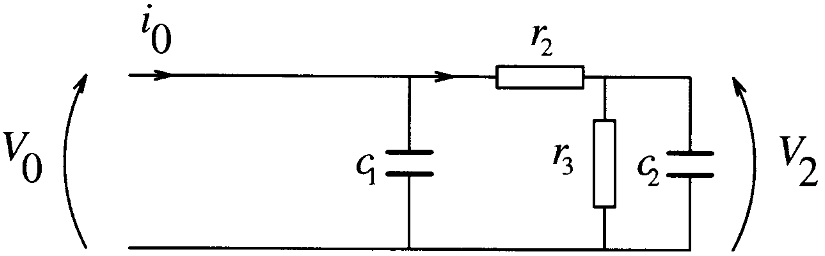


Figure 4.6 Electrical circuit: input resistor removed

Figure 4.2 represents a simple electrical circuit comprising two capacitors and three resistors. It is driven by a voltage source v_0, with corresponding current i_0 and the output voltage is v_2. Thus this system has one input and two outputs

$$u = (v_0)\,;\; y = \begin{pmatrix} i_0 \\ v_2 \end{pmatrix} \tag{4.1}$$

To illustrate various points in this chapter, a number of similar circuits, but with components removed, are also considered:

- Figure 4.3 has r_2 removed,
- Figure 4.4 has c_1 removed,
- Figure 4.5 has c_1 and c_2 removed and
- Figure 4.6 has r_1 removed.

4.2 ACAUSAL BOND GRAPH: GRAPHICAL REPRESENTATION

As discussed in the preceding, and subsequent, chapters and elsewhere (Wellstead, 1979; Karnopp and Rosenberg, 1975; Thoma, 1975; Rosenberg and Karnopp, 1983; Karnopp *et al.*, 1990), The schematic diagrams of many physical systems can be translated into bond graph form by a suitably experienced system modeller. Such an (acausal) bond graph is the starting point of this chapter. Because such a representation is precise and unambiguous, further transformations to give alternative representations can be largely automated.

MTT describes bond graphs using the graphical software xfig together with its associated description language. The figures on the following examples are from postscript versions of xfig representations.

This graphical description does not fully define the bond graph; in particular constitutive relations are not specified. In MTT, the additional information is provided in two text files:

- a label file and
- a CR file.

These are illustrated in the following example.

4.2.1 Example

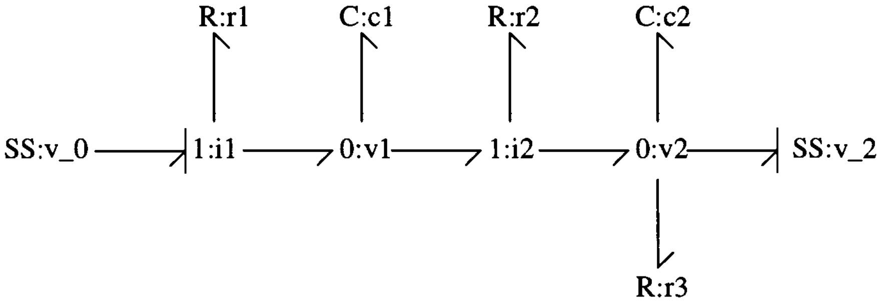


Figure 4.7 Electrical circuit: acausal bond graph

Figure 4.2 of Section 4.1.1 shows a simple electrical circuit. The corresponding bond graph (with integral causality) appears in Figure 4.7.

Causal strokes have been added to the SS components to indicate that the SS component labelled v_0 is imposing an effort whereas the SS component labelled v_2 is measuring an effort.

The *label* file appears in Figure 4.8. The first column corresponds to the labels appended to each component and junction in Figure 4.7. The second column gives a name to each component; this may be the same as the label. In the case of R, C and I components, the third and fourth column give the name and arguments respectively of the corresponding CR. In this case, all components are linear and use the CR labelled lin. The notation

```
[effort,r_1]
```

means that the corresponding component has a gain of r_1 when the component output is an effort.

```
elag2
% System elag2
% Electrical second-order lag
% File elag2_lbl.txt

%Junctions
i1 current1
i2 current2
v1 voltage1
v2 voltage2

%R components
r1      r1      lin      [effort,r_1]
r2      r2      lin      [effort,r_2]
r3      r3      lin      [effort,r_3]

%C components
c1      c1      lin      [state,c_1]
c2      c2      lin      [state,c_2]

%SS components
v_0 ss0      [external,external]
v_2 ss2      [zero,external]
```

Figure 4.8 Electrical circuit: label file

The *CR* file appears in Figure 4.9. This is written in the algebraic manipulation language REDUCE (Rayna, 1987). It defines two CRs: 'lin' and 'unity'. The CR 'lin' implements a linear CR as follows:

- If the component output has the same causality (Causality) as specified in the label file (DefaultCausality), then the component output is 'Gain*Input'.
- If, on the other hand, the component output has a different causality (Causality) to that specified in the label file (DefaultCausality), then the component output is '(1/Gain)*Input';

4.2.2 Example: r_2 removed

This example corresponds to Figure 4.3 of Section 4.1.1. The bond graph is given as Figure 4.10.

The text files associated with this example are the same as those in Figures 4.8 and 4.9 except that the the line corresponding to r_2 is deleted from the label file of Figure 4.8.

```
% System elag2
% Electrical second-order lag
% File elag2_cr.r


%Generic linear operator
OPERATOR Lin;
FOR ALL Causality, DefaultCausality, Gain, Input
SUCH THAT Causality = DefaultCausality
LET Lin(Causality, DefaultCausality, Gain, Input) = Gain*Input;

FOR ALL Causality, DefaultCausality, Gain, Input
SUCH THAT Causality NEQ DefaultCausality
LET Lin(Causality, DefaultCausality, Gain, Input) = (1/Gain)*Input;

%Generic unit operator
OPERATOR Unity;
FOR ALL Causality, Input
LET Unity(Causality, Input) = Input;
```

Figure 4.9 Electrical circuit: cr file

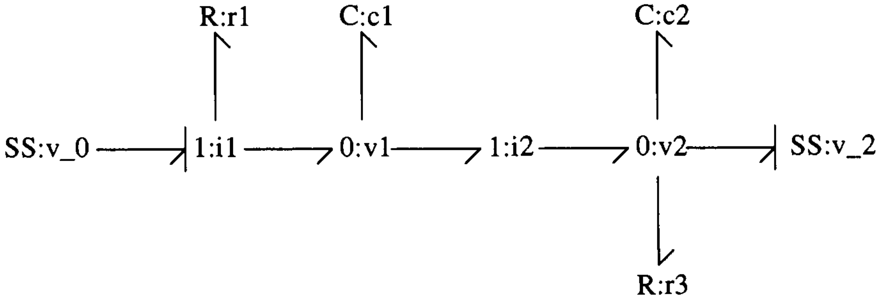


Figure 4.10 Electrical circuit (resistor removed): acausal bond graph

4.2.3 Example: c_1 removed

This example corresponds to Figure 4.4 of Section 4.1.1. The bond graph is given as Figure 4.11.

The text files associated with this example are the same as those in Figures 4.8 and 4.9 except that the the line corresponding to c_1 is replaced by a corresponding source-sensor statement.

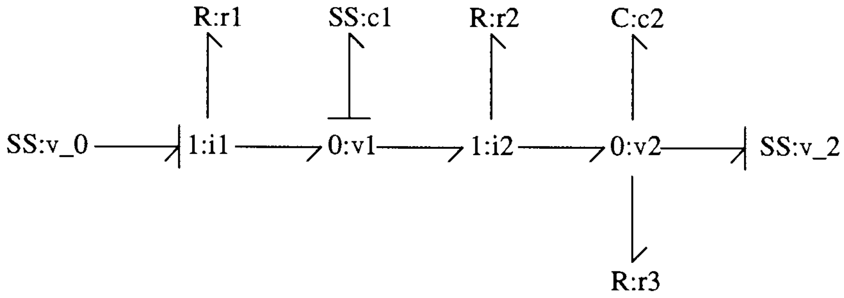


Figure 4.11 Electrical circuit (capacitor removed): acausal bond graph

4.2.4 Example: c_1 and c_2 removed

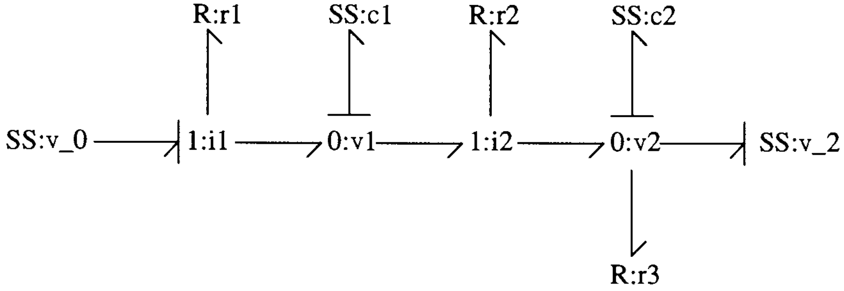


Figure 4.12 Electrical circuit (two capacitors removed): acausal bond graph

This example corresponds to Figure 4.5 of Section 4.1.1. The bond graph is given as Figure 4.12.

The text files associated with this example are the same as those in Figures 4.8 and 4.9 except that the the lines corresponding to c_1 and c_2 are replaced by corresponding source-sensor statements.

4.2.5 Example: r_1 removed

This example corresponds to Figure 4.6 of Section 4.1.1. The bond graph is given as Figure 4.13.

The text files associated with this example are the same as those in Figures 4.8 and 4.9 but with a line deleted.

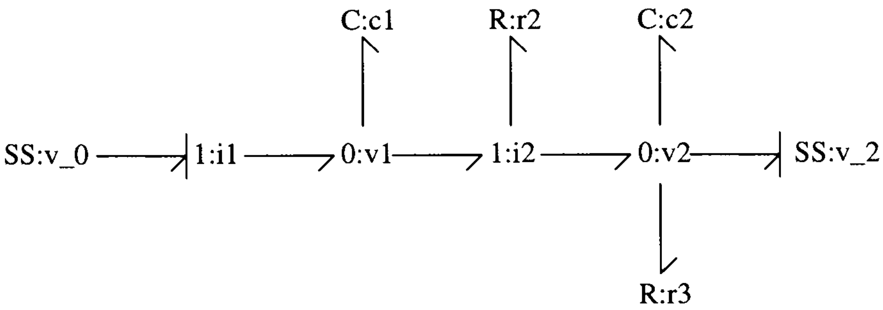


Figure 4.13 Electrical circuit - resistor removed): acausal bond graph

4.3 ACAUSAL BOND GRAPH: LIST REPRESENTATION

Bond graphs are essentially a *diagrammatic* representation of system dynamics. This section considers an equivalent representation more appropriate in the context of system transformations. For simplicity, systems without modulated components will be considered.

From this point of view, a system bond graph is a *list* of junctions, and with each junction is associated a list of bonds impinging on that junction. Each bond is itself a data structure containing information about the bond.

The following Prolog-like representation (as used by MTT) is one possibility:

System

```
System = system(SystemName, Junctions).
```

- 'SystemName' is the name of the system.
- 'Junctions' is a list of junctions.

Junctions

Junctions is a list of junctions of the form

```
Junctions = [Junction1,
             Junction2,
             .....
             JunctionN
            ].
```

Each *junction* is of the form

```
Junction(JunctionName, JunctionType, Bonds).
```

- 'JunctionName' is the junction name.

- 'JunctionType' is either 0 or 1.
- 'Bonds' is a list of bonds.

Standard name	MTT name
R	Dissipator
C	Estore
I	Fstore
SS	Source-sensor
TF	Transformer
GY	Gyrator
Interjunction bond	Junction
Signal bond	Signal

Table 4.1 MTT bond names

Bonds

'Bonds' is a list of bonds of the form

```
Bonds = [Bond1,
         Bond2,
         .....
         BondN].
```

Each *bond* is itself a structure of the form

```
Bond = bond(BondName, Component, BondDirection, BondCausality, CR).
```

- 'BondName'
  - If the bond is connected to a one-port component, then 'BondName' is the name associated with the component
  - If the bond is connected to a two-port component, then 'BondName' is the name of the junction to which the other end is attached.
- 'Component' is one of the standard bond-graph components renamed as in Table 4.1.
- 'BondDirection' is either 'in' or 'out' . This indicates direction of power flow *with respect to the junction.* Thus 'out' indicates that power is flowing out of the junction; this would usually be the case for a one-port component.
- 'BondCausality' is one of 'effort', 'flow' or '_' , the first two indicate whether the *component* output is an effort or a flow, the last indicates that causality is as yet unassigned.
- 'CR' denotes the constitutive relationship of the component.

4.3.1 Example

Figure 4.2 of Section 4.1.1 shows a simple electrical circuit. The corresponding bond graph (with integral causality) appears in Figure 4.7. In particular, (regarding inter-junction bonds as unit-gain transformers), the causally complete bond graph of Figure 4.7 can be rewritten in this list structure (see below).

The other circuits have a similar representation, the only difference is that some components are deleted from the list.

Example: acausal bond graph as a list

```
system(elag2,
[
  junction(current1, flow,
  [
    bond(ss0,source_sensor,in,effort,[external,external]),
    bond(voltage1,junction,out,_,[unity, [], [] ]),
    bond(r1,dissipator,out,_,[lin,[effort,r_1],[]])
  ]),

  junction(current2, flow,
  [
    bond(voltage1,junction,in,_,[unity, [], [] ]),
    bond(voltage2,junction,out,_,[unity, [], [] ]),
    bond(r2,dissipator,out,_,[lin,[effort,r_2],[]])
  ]),

  junction(voltage1, effort,
  [
    bond(current1,junction,in,_,[unity, [], [] ]),
    bond(current2,junction,out,_,[unity, [], [] ]),
    bond(c1,estore,out,_,[lin,[state,c_1],[]])
  ]),

  junction(voltage2, effort,
  [
    bond(current2,junction,in,_,[unity, [], [] ]),
    bond(c2,estore,out,_,[lin,[state,c_2],[]]),
    bond(ss2,source_sensor,out,flow,[zero,external]),
    bond(r3,dissipator,out,_,[lin,[effort,r_3],[]])
  ])
]).
```

4.4 CAUSAL BOND GRAPH: GRAPHICAL REPRESENTATION

There are often many ways to complete the causality of an acausal bond graph to give a causally complete bond graph. As the modeller is often interested in generating differential equations, it is conventional to use integral causality (Section 3.3) with dynamic components. If causality is left unassigned, then MTT tries to complete causality with the maximum number of components with integral causality. However, the modeller must be free to override this default. This is easily done by adding additional causal strokes to the acausal bond graph; MTT never overrides prespecified causality, it just checks that it is correct.

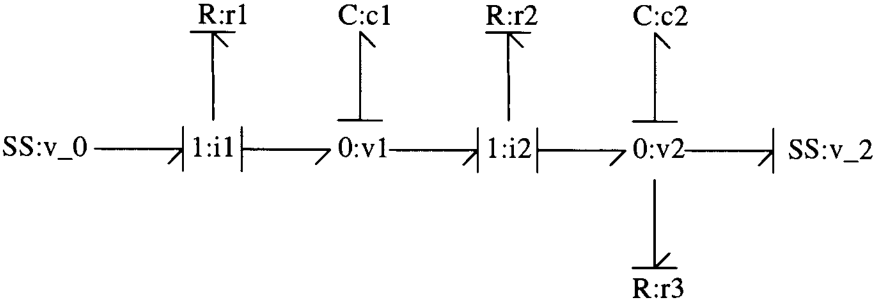


Figure 4.14 Electrical circuit: causal bond graph

Figure 4.14 shows the causal bond graph corresponding to Figure 4.7 where causality has been completed by hand to give integral causality.

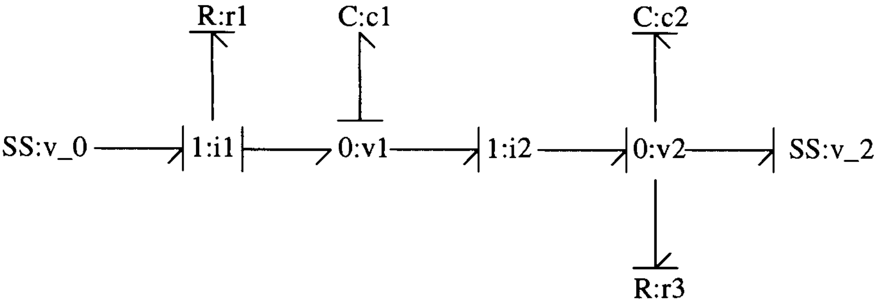


Figure 4.15 Electrical circuit- resistor removed: causal bond graph

Figure 4.15 shows the causal bond graph corresponding to Figure 4.10 where causality has been completed by hand to give integral causality on c_1 but derivative causality on c_2.

4.5 CAUSAL BOND GRAPH: LIST REPRESENTATION

The list representation used by MTT is identical to that described in Section 4.3 except that, by definition, all causalities are assigned.

4.5.1 Example

The lists corresponding to the examples of Section 4.1.1 appear in the following sections.

Integral causality

The following list representation corresponds to Figure 4.14

```
%%% File created by MTT (Version of 4 July 1994).
%%% This file contains all possible causally complete systems
%%% with the minimum number of non-states
%%% System name: elag2
%%% File: elag2_cbg.pl

%%% There is only 1 possible causal completion
%%% with 0 non-states

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% States(2):
%        [c1,effort,[lin,[state,c_1],[]],voltage1]
%        [c2,effort,[lin,[state,c_2],[]],voltage2]
%% Non-states(0):
%% Inputs(1):
%        [ss0,effort,[external,external],current1]
%% Outputs(2):
%        [ss0,flow,[external,external],current1]
%        [ss2,effort,[zero,external],voltage2]
%% Zero outputs(0):

system(elag2,
[
  junction(current1, flow,
  [
        bond(r1, dissipator, out, flow,
          [lin,[effort,r_1],[]]),
        bond(ss0, source_sensor, in, effort,
          [external,external]),
        bond(voltage1, junction, out, effort,
          [unity,[],[]])
  ]),

  junction(current2, flow,
  [
        bond(r2, dissipator, out, flow,
          [lin,[effort,r_2],[]]),
        bond(voltage1, junction, in, effort,
          [unity,[],[]]),
        bond(voltage2, junction, out, effort,
          [unity,[],[]])
  ]),

  junction(voltage1, effort,
  [
        bond(c1, estore, out, effort,
          [lin,[state,c_1],[]]),
        bond(current1, junction, in, flow,
          [unity,[],[]]),
        bond(current2, junction, out, flow,
          [unity,[],[]])
  ]),

  junction(voltage2, effort,
  [
        bond(c2, estore, out, effort,
          [lin,[state,c_2],[]]),
        bond(current2, junction, in, flow,
          [unity,[],[]]),
        bond(ss2, source_sensor, out, flow,
          [zero,external]),
        bond(r3, dissipator, out, flow,
          [lin,[effort,r_3],[]])
  ])

]).
```

r_1 *removed, mixed integral and derivative causality*

The following list representation corresponds to Figure 4.15.

```
%%% File created by MTT (Version of 4 July 1994).
%%% This file contains all possible causally complete systems
%%% with the minimum number of non-states
%%% System name: elag2r1
%%% File: elag2r1_cbg.pl
%%% There is only 1 possible causal completion
%%% with 1 non-states

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% States(1):
%        [c1,effort,[lin,[state,c_1],[]],voltage1]
%% Non-states(1):
%        [c2,flow,[lin,[state,c_2],[]],voltage2]
%% Inputs(1):
%        [ss0,effort,[external,external],current1]
%% Outputs(2):
%        [ss0,flow,[external,external],current1]
%        [ss2,effort,[zero,external],voltage2]
%% Zero outputs(0):

system(elag2r1,
[
  junction(current1, flow,
  [
        bond(r1, dissipator, out, flow,
          [lin,[effort,r_1],[]]),
        bond(ss0, source_sensor, in, effort,
          [external,external]),
        bond(voltage1, junction, out, effort,
          [unity,[],[]])
  ]),

  junction(current2, flow,
  [
        bond(voltage2, junction, out, flow,
          [unity,[],[]]),
        bond(voltage1, junction, in, effort,
          [unity,[],[]])
  ]),

  junction(voltage1, effort,
  [
        bond(c1, estore, out, effort,
          [lin,[state,c_1],[]]),
        bond(current1, junction, in, flow,
          [unity,[],[]]),
        bond(current2, junction, out, flow,
          [unity,[],[]])
  ]),
  junction(voltage2, effort,
  [
        bond(current2, junction, in, effort,
          [unity,[],[]]),
        bond(c2, estore, out, flow,
          [lin,[state,c_2],[]]),
        bond(ss2, source_sensor, out, flow,
          [zero,external]),
        bond(r3, dissipator, out, flow,
          [lin,[effort,r_3],[]])
  ])

]).
```

4.6 ORDERED ELEMENTARY SYSTEM EQUATIONS

The equations describing a system comprise:

- the component constitutive relationships (including modulation),
- the differential equations associated with C and I components,
- and the interconnection constraints via 0-junctions and 1-junctions

The causally complete system bond graph (of a system that is neither under-causal or over-causal) provides a structure for writing these equations in a *systematic* manner. In particular, they can be written as *causally ordered* assignment statements where the right-hand side of each assignment statement contains terms that have been computed by preceding assignment statements or are assumed to be known.

The system dynamics arise from the C and I components within the system. As discussed in Section 3.3.4, such components have either integral or derivative causality.

Integral causality gives rise to *differential* equations of the form

$$\begin{aligned} \dot{x} &= u \\ y &= \phi(u) \end{aligned} \tag{4.2}$$

where, for a C component, x is the integrated flow, y the effort output, u the flow input. ϕ is the constitutive relation giving effort output y in terms of x. For an I component, x is the integrated effort, y the flow output and u the effort input. ϕ is the constitutive relation giving flow output y in terms of x.

According to common usage, x is called the *state* of the component and its corresponding differential equation is Equation 4.2.

Differential causality gives rise to *integral* equations of the form

$$z = \phi^{-1}u$$

$$y = \int z \, dt \tag{4.3}$$

In this book, z is called the *non-state* of the component; it is not a state as such because it is directly dependent on the component input.

These equations can be systematically written in four groups of equations with the following outputs

1. State derivatives $\dot{x}$: *inputs* of C and I components with *integral* causality,

2. Non-states z,

3. Measurements (corresponding to SS components) which are constrained to be zero w, and

4. System outputs y.

Each of these four groups of equations is expressed in terms of four groups of variables:

1. *states* x of C and I components with *integral* causality

2. *outputs* of S and SS *without* zero input constraints u (that is, system inputs).

3. *outputs* of SS components *with* zero input constraints v

4. *non-state derivatives* of C and I components with *derivative* causality $\dot{z}$.

Different sets of equations will arise for the same system if different causal patterns are applied. Thus, for example, assigning differential or integral causality can give rise to quite different sets of equations as discussed by Karnopp (1983).

4.6.1 Transformation

In the context of bond graph modelling, the transformation from a causal bond-graph to a set of *ordered* assignment statements is important.

The following *recursive* algorithm is used to generate equations for a particular variable (defined according to the data structure in Section 4.2 as that variable with given 'Causality' associated with a 'Bond' attached to a 'Junction ') in terms of a list of known variables contained in 'old_list'. A 'new_list' is produced of the variables in 'old_list' together with those discovered in generating the equations needed to deduce the particular variable.

The algorithm is presented in pseudo-Pascal form below. (MTT actually uses a Prolog implementation - but this is not readable without a working knowledge of Prolog.)

The algorithm is applied in turn to each variable in the first set of four groups. 'old_list' is initialised to contain the names of all variables in the second set of four groups, and thereafter updated to 'new_list' after each application of the algorithm.

Algorithm

```
PROCEDURE write_equation_for(Junction, Bond, Causality, old_list, new_list)
  BEGIN{write_equation_for}
  IF NOT already_known(Junction.Name,Bond.Name,Causality,old_list)
    THEN
    BEGIN{Write equations}
    IF Causality = Bond.Causality
    THEN
      BEGIN{Component output}
      IF Bond.Component = r
      THEN {Write its input equation}
        write_equation_for(Junction, Bond, OtherCausality,
                           old_list, intermediate_list)
      ELSEIF  Bond.Component = tf
      THEN {Write its input equation}
        write_equation_for(Bond.Name, Bond, OtherCausality,
                           old_list, intermediate_list)
      ELSEIF  Bond.Component = gy
      THEN {Write its input equation}
        write_equation_for(Bond.Name, Bond, Causality,
                           old_list, intermediate_list)
      {Write out the component equation}
      write_component_equation(Junction, Bond);
      END{Component output}
    ELSE
      BEGIN{Component input from junction}
      IF Junction.Causality = Bond.Causality
      THEN
        BEGIN{Same causality as junction}\\
        find_causing_bond(Junction,Causing_Bond);
        write_equation_for(Junction, Causing_Bond, OtherCausality,
                           old_list, intermediate_list);
        write_equality(Bond,Causing_Bond,OtherCausality);
        END{Same causality as junction}
      ELSE
        BEGIN{Opposite causality to junction}
        find_causing_bonds(Junction,Causing_Bonds);
        write_equations_for(Junction, Causing_Bonds, OtherCausality,
                            old_list, intermediate_list);
        write_summation(Bond,Causing_Bonds,OtherCausality);
        END{Opposite causality to junction}
      END{Component input}
   append(Junction, Bond, Causality, intermediate_list, new_list);
   END{Write equations}
  END{write_equation_for}
```

Remarks

1. The line beginning 'IF NOT already_known' prevents redundant repetition of equations. It requires the function already_known to check against a list of already known equations.
2. The line beginning 'IF Causality = Bond.Causality' checks whether the *output* of a component is reqired. If so, the statements between 'BEGINComponent output' and 'ENDComponent output' are executed.
3. The lines beginning 'IF Bond.Component = ' and 'ELSEIF Bond.Component =' perform the operations appropriate to each component. If the components have an input, then the procedure 'write_equation_for' is *recursively* executed to write the necessary equations.
4. The line beginning 'write_component_equation' writes out the appropriate component CR.
5. If the statement 'IF Causality = Bond.Causality' is false the statements between 'BEGINComponent input' and 'ENDComponent input' are executed. In this case, the component input must be computed from the appropriate junction equation. Regarding a 0 junction as an effort junction and a 1 junction as a flow junction, there are two possibilities:
   (a) The variable causality is the same as that of the junction
   (b) The variable causality is *not* the same as that of the junction
6. In the former case, the single bond imposing the common variable onto the junction must be found (the line beginning 'find_causing_bond (Junction, Causing_Bond)' does this).
7. In the latter case, all the other bonds must be found (the line beginning 'find_causing_bonds (Junction, Causing_Bond)' does this).
8. In the former case, a single equality must be written - using the line beginning 'write_equality' does this.
9. In the latter case, a summation (having regard to direction) must be written - using the line beginning 'write_summation' does this.

4.6.2 Example

Continuing the example of Section 4.1.1 the elementary equations can be written down as in Section 4.6.2 below. Intermediate variables associated with components are automatically labelled as

```
junction_component_E
```

or

```
junction_component_F
```

to indicate effort or flow respectively associated with a component on a junction. Similarly intermediate variables associated with interjunction bonds are labelled as

```
junction1_junction2_E
```

or

```
junction1_junction2_F
```

In this case, only the first and last sets of equations are needed.

Full circuit

```
OFF echo;
%%% File created by MTT (Version of 14th Dec 1993).
%File: elag2.req

% States(2):
%        [c1,effort,[lin,[state,c_1],[]],voltage1]
%        [c2,effort,[lin,[state,c_2],[]],voltage2]
% Non-states(0):
% Inputs(1):
%        [ss0,effort,[external,external],current1]
% Zero Inputs(1):
%        [ss2,flow,[zero,external],voltage2]
% Internal Inputs(0):
% Outputs(2):
%        [ss0,flow,[external,external],current1]
%        [ss2,effort,[zero,external],voltage2]
% Zero Outputs(0):

% Set up the system input vector
matrix MTTU(1,1)$
MTTU(1,1) := MTTU1$
current1_ss0_E := MTTU1$

% Set up the system zero input vector
voltage2_ss2_F := 0$

% Set up the system internal input vector

% Set up the system state;matrix MTTX(2,1)$
MTTX(1,1) := MTTX1$
voltage1_c1_S  := MTTX1$
MTTX(2,1) := MTTX2$
```

```
voltage2_c2_S  := MTTX2$


% Equations generating state:
  % Junction: voltage1
  % Component: c1

voltage1_c1_E := lin(effort,
        state,
        c_1,
        voltage1_c1_S)$
voltage1_current1_E :=  + (
         + voltage1_c1_E)$
current1_voltage1_E := unity(effort,
        voltage1_current1_E)$
current1_r1_E :=  + (
         + current1_ss0_E
         - current1_voltage1_E)$
current1_r1_F := lin(flow,
        effort,
        r_1,
        current1_r1_E)$
current1_voltage1_F :=  + (
         + current1_r1_F)$
voltage1_current1_F := unity(flow,
        current1_voltage1_F)$
voltage1_current2_E :=  + (
         + voltage1_c1_E)$
current2_voltage1_E := unity(effort,
        voltage1_current2_E)$
voltage2_c2_E := lin(effort,
        state,
        c_2,
        voltage2_c2_S)$
voltage2_current2_E :=  + (
         + voltage2_c2_E)$
current2_voltage2_E := unity(effort,
        voltage2_current2_E)$
current2_r2_E :=  + (
         + current2_voltage1_E
         - current2_voltage2_E)$
current2_r2_F := lin(flow,
        effort,
        r_2,
        current2_r2_E)$
current2_voltage1_F :=  + (
         + current2_r2_F)$
voltage1_current2_F := unity(flow,
        current2_voltage1_F)$
voltage1_c1_F :=  + (
         + voltage1_current1_F
         - voltage1_current2_F)$

% Equations generating state:
  % Junction: voltage2
  % Component: c2

current2_voltage2_F :=  + (
         + current2_r2_F)$
voltage2_current2_F := unity(flow,
        current2_voltage2_F)$
voltage2_r3_E :=  + (
         + voltage2_c2_E)$
voltage2_r3_F := lin(flow,
        effort,
        r_3,
        voltage2_r3_E)$
voltage2_c2_F :=  + (
         + voltage2_current2_F
         - voltage2_ss2_F
         - voltage2_r3_F)$

% Set up the system state derivative;matrix MTTdX(2,1)$
MTTdX(1,1) := voltage1_c1_F$
MTTdX(2,1) := voltage2_c2_F$

% Set up the system output vector;
matrix MTTY(2,1)$
% Output equation: ss0;
current1_ss0_F :=  + (
         + current1_r1_F)$

% Output equation: ss2;
voltage2_ss2_E :=  + (
         + voltage2_c2_E)$

MTTY(1,1) := current1_ss0_F$
MTTY(2,1) := voltage2_ss2_E$

% Set up the system zero output vector;
```

```
;END;

OFF echo;
%%% File created by MTT (Version of 14th Dec 1993).
%File: elag2c.req

% States(1):
%        [c2,effort,[lin,[state,c_2],[]],voltage2]
% Non-states(0):
% Inputs(1):
%        [ss0,effort,[external,external],current1]
% Zero Inputs(1):
%        [ss2,flow,[zero,external],voltage2]
% Internal Inputs(1):
%        [c1,effort,[internal,zero],voltage1]
% Outputs(2):
%        [ss0,flow,[external,external],current1]
%        [ss2,effort,[zero,external],voltage2]
% Zero Outputs(1):
%        [c1,flow,[internal,zero],voltage1]

% Set up the system input vector
matrix MTTU(1,1)$
MTTU(1,1) := MTTU1$
current1_ss0_E := MTTU1$

% Set up the system zero input vector
voltage2_ss2_F := 0$

% Set up the system internal input vector
voltage1_c1_E := MTTUI1$

% Set up the system state;matrix MTTX(1,1)$
MTTX(1,1) := MTTX1$
voltage2_c2_S  := MTTX1$


% Equations generating state:
  % Junction: voltage2
  % Component: c2

voltage1_current2_E :=  + (
         + voltage1_c1_E)$
current2_voltage1_E := unity(effort,
        voltage1_current2_E)$
voltage2_c2_E := lin(effort,
        state,
        c_2,
        voltage2_c2_S)$
voltage2_current2_E :=  + (
         + voltage2_c2_E)$
current2_voltage2_E := unity(effort,
        voltage2_current2_E)$
current2_r2_E :=  + (
         + current2_voltage1_E
         - current2_voltage2_E)$
current2_r2_F := lin(flow,
        effort,
        r_2,
        current2_r2_E)$
current2_voltage2_F :=  + (
         + current2_r2_F)$
voltage2_current2_F := unity(flow,
        current2_voltage2_F)$
voltage2_r3_E :=  + (
         + voltage2_c2_E)$
voltage2_r3_F := lin(flow,
        effort,
        r_3,
        voltage2_r3_E)$
voltage2_c2_F :=  + (
         + voltage2_current2_F
         - voltage2_ss2_F
         - voltage2_r3_F)$

% Set up the system state derivative;matrix MTTdX(1,1)$
MTTdX(1,1) := voltage2_c2_F$

% Set up the system output vector;
matrix MTTY(2,1)$
% Output equation: ss0;
voltage1_current1_E :=  + (
         + voltage1_c1_E)$
current1_voltage1_E := unity(effort,
        voltage1_current1_E)$
current1_r1_E :=  + (
         + current1_ss0_E
         - current1_voltage1_E)$
current1_r1_F := lin(flow,
        effort,
        r_1,
        current1_r1_E)$
current1_ss0_F :=  + (
         + current1_r1_F)$

% Output equation: ss2;
voltage2_ss2_E :=  + (
         + voltage2_c2_E)$

MTTY(1,1) := current1_ss0_F$
MTTY(2,1) := voltage2_ss2_E$

% Set up the system zero output vector;
% Output equation: c1;
current1_voltage1_F :=  + (
         + current1_r1_F)$
voltage1_current1_F := unity(flow,
        current1_voltage1_F)$
current2_voltage1_F :=  + (
         + current2_r2_F)$
voltage1_current2_F := unity(flow,
        current2_voltage1_F)$
voltage1_c1_F :=  + (
         + voltage1_current1_F
         - voltage1_current2_F)$

MTTYz1 := voltage1_c1_F$


;END;
```

4.7 DIFFERENTIAL-ALGEBRAIC EQUATIONS

The ordered elementary system equations of Section 4.6 can be compressed by eliminating the intermediate variables in each block of assignment statements.

After this process, there will be *four* groups of equations giving:

1. state derivatives $\dot{x}$,
2. non-states z,
3. zero internal system outputs $w = 0$ and
4. system outputs y

all in terms of

1. system states x,
2. system inputs u,

3. internal inputs v,

4. and non-state derivatives $\dot{z}$.

Mathematically, this can be expressed in functional form as

$$\begin{aligned} \dot{x} &= f_x(x,u,v,\dot{z}) \\ z &= f_z(x,u,v,\dot{z}) \\ w=0 &= f_w(x,u,v,\dot{z}) \\ y &= f_y(x,u,v,\dot{z}) \end{aligned} \tag{4.4}$$

These equations represent a set of *differential-algebraic* equations (Gear and Petzold, 1984; Brenan *et al.*, 1989; Mattsson, 1989; Pantelides *et al.*, 1988).

This set of equations may be rewritten in a more compact form by defining the *descriptor* vector

X where

$$X = \begin{pmatrix} x \\ z \\ \dot{z} \\ v \end{pmatrix} \tag{4.5}$$

and

1. x is the $N_x \times 1$ vector of *state* variables associated with C and I elements with *integral* causality.

2. z is the $N_z \times 1$ vector of *non-state* variables associated with C and I elements with *derivative* causality.

3. $\dot{z}$ is the $N_z \times 1$ vector containing the corresponding derivatives.

4. v is the $N_v \times 1$ vector of additional inputs.

The system equations then become

$$\begin{aligned} E\dot{X} &= F(X,u) \\ y &= G(X,u) \end{aligned} \tag{4.6}$$

Where

$$E = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \tag{4.7}$$

Where E is a square matrix of dimension $N_x + 2N_z + N_v$ and I is the unit matrix of dimension $N_x + N_z$. For simplicity, we will denote this particular form of E by

$$E = I_0(N_x + N_z, N_z + N_v) \tag{4.8}$$

where $I_0(N_1, N_2)$ is an $N_1 + N_2 \times N_1 + N_2$ matrix with unit elements on the first N_1 diagonal elements and zeros elsewhere.

In general, E is singular (unless $N_z = N_v = 0$), and so Equations 4.6 cannot be written as an ordinary differential Equation 4.14. Such equations are variously called *differential-algebraic* equations (Gear and Petzold, 1984; Brenan *et al.*, 1989; Mattsson, 1989; Pantelides *et al.*, 1988) *descriptor* equations (Luenberger, 1977), *singular* equations (Campbell, 1980; Campbell, 1982), or *generalised state-space* equations (Verghese *et al.*, 1981).

In general, such equations are hard to handle, so particular forms of these will be discussed in the following sections:

- Algebraic equations
- Ordinary differential equations
- Constrained-state equations
- Semi-explicit differential-algebraic equations

4.8 ALGEBRAIC EQUATION

A special case of the general equations 4.4 arises when:

- there are no system states x and
- there are no system non-states z.

In this case, the general equations 4.4 are purely algebraic and are given by

$$\begin{aligned} w = 0 &= f_w(u,v) \\ y &= f_y(u,v) \end{aligned} \tag{4.9}$$

This form arises when the bond graph contains no C or I components: the system has no dynamics. The solution of Equation 4.9 involves the algebraic or numerical solution of N_w algebraic equations for the N_w unknowns v. This is particularly easy in the linear non-singular case where

$$\begin{aligned} w = 0 &= Bu + B_w v \\ y &= Du + D_w v \end{aligned} \tag{4.10}$$

where B_w is a non-singular $N_w \times N_w$ matrix. In particular

$$y = Du + D_w B_w^{-1} Bu \tag{4.11}$$

4.8.1 Example

Figure 4.5 of Section 4.1.1, which corresponds to the bond graph of Figure 4.12, contains no dynamic elements and therefore represents a set of *algebraic equations.*

Eliminating the intermediate variables gives the algebraic equations

$$0 = \frac{(-((r_2 + r_1)v_1 - v_2 r_1 - r_2 u_1))}{(r_2 r_1)}$$

$$0 = \frac{(-((r_2 + r_3)v_2 - v_1 r_3))}{(r_2 r_3)} \tag{4.12}$$

$$\begin{aligned} y_1 &= \frac{(-(v_1 - u_1))}{r_1} \\ y_2 &= v_2 \end{aligned} \tag{4.13}$$

4.9 ORDINARY DIFFERENTIAL EQUATIONS

A special case of the general equations 4.4 arises when:

- there are no additional inputs v and
- there are no system non-states z.

In this case, the general equations 4.4 become

$$\dot{x} = f_x(x, u) \tag{4.14}$$

This is an ordinary differential equation (ODE), and is perhaps the simplest conceptual dynamic system. This form arises when the bond graph is causally complete and contains no C or I components with *derivative causality.*

The solution is particularly easy in the linear case where

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{4.15}$$

4.9.1 Example

Figure 4.2 of Section 4.1.1 corresponds to the bond graph of Figure 4.7. This is causally complete, has integral causality and has no SS components with constrained measurements.

Eliminating the intermediate variables in Section 4.6.2 gives the ordinary differential equation

$$\begin{aligned} \dot{x}_1 &= \frac{(-(r_1 c_2 x_1 - r_1 c_1 x_2 - c_2 r_2 c_1 u_1 + c_2 r_2 x_1))}{(r_1 c_2 r_2 c_1)} \\ \dot{x}_2 &= \frac{(-((r_2 + r_3)c_1 x_2 - c_2 r_3 x_1))}{(c_2 r_2 c_1 r_3)} \end{aligned} \tag{4.16}$$

$$\begin{aligned} y_1 &= \frac{(c_1 u_1 - x_1)}{(r_1 c_1)} \\ y_2 &= \frac{x_2}{c_2} \end{aligned} \tag{4.17}$$

4.10 CONSTRAINED-STATE EQUATIONS

A special case of the general Equations 4.4 arises when:

- there are no additional inputs v,
- the system non-state z is a function of the state x and the system input u only and
- the state and output equations are *linear* in the non-state z.

Equations of this form are associated with

- system approximation (Chapter 5),
- system inversion (Chapter 6) and
- mechanical systems (Chapter 10).

In this case, the general Equations 4.4 become

$$\begin{aligned} \dot{x} &= f_x(x,u) + F_x(x,u)\dot{z} \\ z &= f_z(x,u) \\ y &= f_y(x,u) + F_y(x,u)\dot{z} \end{aligned} \tag{4.18}$$

The non-states arise from C and I components whose state is determined directly by states of other C and I components with integral causality and/or system inputs. The discussion is restricted to the special case where $\dot{z}$ appears *linearly* in the state and output equations.

It is termed the *constrained-state* form as the second equation indicates that the non-states are constrained in terms of the states and inputs by an algebraic equation.

A standard technique associated with differential algebraic equations is to use differentiation followed by substitution to reduce the differential algebraic equation to an ordinary differential equation (Gear and Petzold, 1984; Brenan *et al.*, 1989; Mattsson, 1989; Pantelides *et al.*, 1988). Because of the special structure of the constrained-state Equation 4.18, arising from the bond-graph formulation, this technique is easily applied to Equations 4.18.

In particular, the equation constraining z in terms of x and u can be differentiated with respect to time

$$\dot{z} = G_x(x,u)\dot{x} + G_u(x,u)\dot{u} \tag{4.19}$$

where the Jacobian matrices $G_x(x,u)$ and $G_u(x,u)$ are given by

$$G_x(x) = \frac{\partial G(x,u)}{\partial x};\ G_u(x) = \frac{\partial G(x,u)}{\partial u} \tag{4.20}$$

Equations 4.18 can be rearranged to give

$$\begin{aligned} E(x,u)\dot{x} &= f_x(x,u) + E_{xu}\dot{u} \\ y &= f_y(x,u) + E_{yx}\dot{x} + E_{yu}\dot{u} \end{aligned} \tag{4.21}$$

where

$$E(x,u) = I - E_{xx}(x,u) \tag{4.22}$$

I is the appropriate unit matrix and

$$\begin{aligned} E_{xx}(x,u) &= F_x(x)G_x(x) \\ E_{xu}(x,u) &= F_x(x)G_u(x) \\ E_{yx}(x,u) &= F_y(x)G_x(x) \\ E_{yu}(x,u) &= F_y(x)G_u(x) \end{aligned} \tag{4.23}$$

It is convenient to rewrite Equation 4.21 as

$$\begin{aligned} \dot{\chi} &= f_x(x,u) + E_{xu}\dot{u} \\ y &= f_y(x,u) + E_{yx}\dot{x} + E_{yu}\dot{u} \end{aligned} \tag{4.24}$$

where

$$\dot{\chi} = E(x)\dot{x} \tag{4.25}$$

As will be seen in the subsequent example and in Chapters 5, 6 and 10, this form can give useful physical insights into system behaviour. Indeed, in Section 10.8, the standard robot-form equations are shown to be directely linked to constrained-state equations.

Note that E is *not* usually singular in this case (unlike Section 4.7) so, in principle, Equations 4.24 can be rewritten as the ODE

$$\begin{aligned} \dot{x} &= E^{-1}(x,u)\left[f_x(x,u) + E_{xu}\dot{u}\right] \\ y &= f_y(x,u) + E_{yx}E^{-1}(x,u)\left[f_x(x,u) + E_{xu}\dot{u}\right] + E_{yu}\dot{u} \end{aligned} \tag{4.26}$$

4.10.1 Example: r_2 removed

Figure 4.3 of Section 4.1.1 corresponds to the bond graph of Figure 4.10. This is causally complete, but has one C component with derivative causality and one with integral causality.

Eliminating the intermediate variables gives the DAE

$$\dot{x}_1 = \frac{(-(r_1r_3c_1\dot{z}_1 + r_1x_1 - r_3c_1u_1 + r_3x_1))}{(r_1r_3c_1)} \tag{4.27}$$

$$z_1 = \frac{(c_2x_1)}{c_1} \tag{4.28}$$

$$\begin{aligned} y_1 &= \frac{(c_1u_1 - x_1)}{(r_1c_1)} \\ y_2 &= \frac{x_1}{c_1} \end{aligned} \tag{4.29}$$

which can be rewritten in *constrained-state* form as

$$\dot{\chi}_1 = \frac{(-((r_1 + r_3)x_1 - r_3 c_1 u_1))}{(r_1 r_3 c_1)} \tag{4.30}$$

$$y_1 = \frac{(c_1 u_1 - x_1)}{(r_1 c_1)} \tag{4.31}$$

$$y_2 = \frac{x_1}{c_1} \tag{4.32}$$

$$E = \left(\frac{(c_1+c_2)}{c_1} \right) \tag{4.33}$$

Note that E reflects the fact that the two capacities c_1 and c_2 add to form an equivalent capacity. Further examples of constrained-state equations appear in Chapters 5 and 10.

The corresponding ODE is

$$\dot{x}_1 = \frac{(-((r_1 + r_3)x_1 - r_3 c_1 u_1))}{((c_1 + c_2) r_1 r_3)} \tag{4.34}$$

$$y_1 = \frac{(c_1 u_1 - x_1)}{(r_1 c_1)} \tag{4.35}$$

$$y_2 = \frac{x_1}{c_1} \tag{4.36}$$

4.10.2 Example: r_1 removed

Figure 4.6 of Section 4.1.1 corresponds to the bond graph of Figure 4.13. Capacitor c_1 has its state directly determined by the input voltage u_1.

The DAE is

$$\dot{x}_1 = \frac{(-((r_2 + r_3)x_1 - r_3 c_2 u_1))}{(r_2 r_3 c_2)} \tag{4.37}$$

$$z_1 = c_1 u_1 \tag{4.38}$$

$$\begin{aligned} y_1 &= \frac{(r_2 c_2 \dot{z}_1 + c_2 u_1 - x_1)}{(r_2 c_2)} \\ y_2 &= \frac{x_1}{c_2} \end{aligned} \tag{4.39}$$

which can be rewritten in *constrained-state* form as

$$\dot{\chi}_1 = \frac{(-((r_2 + r_3)x_1 - r_3 c_2 u_1))}{(r_2 r_3 c_2)} \tag{4.40}$$

$$y_1 = \frac{(r_2 c_1 c_2 \dot{u}_1 + c_2 u_1 - x_1)}{(r_2 c_2)} \tag{4.41}$$

$$y_2 = \frac{x_1}{c_2} \tag{4.42}$$

$$E_{11} = 1 \tag{4.43}$$

As $E = 1$, this equation is already in ODE form.

4.11 SEMI-EXPLICIT DIFFERENTIAL-ALGEBRAIC EQUATIONS

In this case, the general Equations 4.4 become

$$\begin{aligned} \dot{x} &= f_x(x,u,v) \\ w = 0 &= f_w(x,u,v) \\ y &= f_y(x,u,v) \end{aligned} \tag{4.44}$$

This is an ordinary differential equation (ODE) in x, together with an algebraic equation relating x and v. This form arises when the bond graph is causally *incomplete* and contains no C or I components with *derivative causality.*

4.11.1 Example

Figure 4.4 of Section 4.1.1 corresponds to the bond graph of Figure 4.11. This is causally complete, has integral causality but has one SS component c_1 with a constrained measurement.

Eliminating the intermediate variables of the corresponding elementary system equations gives the Semi-explicit differential-algebraic equation

$$\dot{x}_1 = \frac{(-((r_2+r_3)x_1 - v_1 r_3 c_2))}{(r_2 r_3 c_2)} \tag{4.45}$$

$$0 = \frac{(-((r_2+r_1)v_1 c_2 - r_2 c_2 u_1 - r_1 x_1))}{(r_2 c_2 r_1)} \tag{4.46}$$

$$\begin{aligned} y_1 &= \frac{(-(v_1 - u_1))}{r_1} \\ y_2 &= \frac{x_1}{c_2} \end{aligned} \tag{4.47}$$

4.12 LINEARISED DESCRIPTOR EQUATIONS

In general, the system Equations 4.6, repeated here as

$$\begin{aligned} E\dot{X} &= F(X,u) \\ y &= G(X,u) \end{aligned} \tag{4.48}$$

will be nonlinear. For analysis purposes, it is useful to linearise these equations about a given steady-state condition. There are two stages to this process:

- finding the steady-state
- performing the linearisation.

The first stage is the hardest, and it may not be possible to obtain an explicit algebraic solution to the problem. By definition, a steady-state solution to Equation 4.48 implies that $\dot{X} = 0$. Thus a steady-state solution corresponds to a value of $X = X_s$ and $u = u_s$ such that

$$F(X_s, u_s) = 0 \tag{4.49}$$

The second step involves expanding F and G of Equation 4.48 in a first order Taylor series about the steady state. X, Y and u are written as

$$\begin{aligned} X &= X_s + \tilde{X} \\ Y &= Y_s + \tilde{Y} \\ u &= u_s + \tilde{u} \end{aligned} \tag{4.50}$$

and

$$\begin{aligned} F(X_s + \tilde{X}, u_s + \tilde{u}) &\approx F(X_s, u_s) + A\tilde{X} + B\tilde{u} \\ G(X_s + \tilde{X}, u_s + \tilde{u}) &\approx G(X_s, u_s) + C\tilde{X} + D\tilde{u} \end{aligned} \tag{4.51}$$

where the *Jacobian* matrices A, B, C and D are given by

$$\begin{aligned} a_{ij} &= \frac{\partial f_i}{\partial x_j} \\ b_{ij} &= \frac{\partial f_i}{\partial u_j} \\ c_{ij} &= \frac{\partial g_i}{\partial x_j} \\ d_{ij} &= \frac{\partial g_i}{\partial u_j} \end{aligned} \tag{4.52}$$

where a_{ij} is the ijth element of A (similarly for B, C and D), f_i is the ith (function) element of the vector F x_i is the ith element of the vector X and u_i is the ith element of the vector u.

Substituting into Equation 4.48, it follows that

$$\begin{aligned} E\dot{X} &\approx F(X_s, u_s) + A\tilde{X} + B\tilde{u} \\ Y &\approx G(X_s, u_s) + C\tilde{X} + D\tilde{u} \end{aligned} \tag{4.53}$$

Noting Equation 4.49 and the fact that X_s is constant

$$\begin{aligned} E\dot{\tilde{X}} &\approx A\tilde{X} + B\tilde{u} \\ \tilde{Y} &\approx C\tilde{X} + D\tilde{u} \end{aligned} \tag{4.54}$$

If the underlying system is linear, there is no approximation involved and so the tildes can be removed and $\approx$ replaced by $=$ in the linearised Equations 4.54 to give

$$\begin{aligned} E\dot{X} &= AX + Bu \\ Y &= CX + Du \end{aligned} \tag{4.55}$$

Noting the special structure of Equation 4.4 the linearised equations have the special form

$$\dot{x} = A_{xx}x + A_{xz}\dot{z} + A_{xw}v + B_x u$$

$$\begin{aligned}
\dot{z} &= \dot{z} \\
0 &= -z + A_{zx}x + A_{zz}\dot{z} + A_{zw}v + B_z u \\
0 = w &= A_{wx}x + A_{wz}\dot{z} + A_{ww}v + B_w u \\
y &= A_{yx}x + A_{yz}\dot{z} + A_{yy}v + B_y u
\end{aligned} \tag{4.56}$$

Thus A, B , C and D are given by

$$A = \begin{pmatrix} A_{xx} & 0 & A_{xz} & A_{xw} \\ 0 & 0 & I & 0 \\ A_{zx} & -I & A_{zz} & A_{zw} \\ A_{wx} & 0 & A_{wz} & A_{ww} \end{pmatrix} \tag{4.57}$$

$$B = \begin{pmatrix} B_x \\ 0 \\ B_z \\ B_w \end{pmatrix} \tag{4.58}$$

$$C = \begin{pmatrix} A_{yx} & 0 & A_{yz} & A_{yv} \end{pmatrix} \tag{4.59}$$

$$D = \begin{pmatrix} B_y \end{pmatrix} \tag{4.60}$$

In the special case that $E = 1$, the descriptor vector X becomes the state vector x and Equation 4.55 becomes the *linear state-space equation*:

$$\begin{aligned}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{aligned} \tag{4.61}$$

4.12.1 Example

All the examples in this chapter are linear, and so they do not have to be linearised as such. Nevertheless, it is convenient to rewrite the system equations in the systematic form of Equation 4.51. In particular, the five matrices A, B , C , D and E fully describe the systems. This is done for the various versions of the electrical circuit discussed in the previous examples.

Full circuit

$$A = \begin{pmatrix} \frac{(-(r_1+r_2))}{(r_1 r_2 c_1)} & \frac{1}{(r_2 c_2)} \\ \frac{1}{(r_2 c_1)} & \frac{(-(r_2+r_3))}{(r_2 c_2 r_3)} \end{pmatrix} \tag{4.62}$$

$$B = \begin{pmatrix} \frac{1}{r_1} \\ 0 \end{pmatrix} \tag{4.63}$$

$$C = \begin{pmatrix} \frac{(-1)}{(r_1 c_1)} & 0 \\ 0 & \frac{1}{c_2} \end{pmatrix} \tag{4.64}$$

$$D = \begin{pmatrix} \frac{1}{r_1} \\ 0 \end{pmatrix} \tag{4.65}$$

$r_2$removed

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.66}$$

$$A = \begin{pmatrix} \frac{(-(r_1+r_3))}{(r_1 r_3 c_1)} & 0 & -1 \\ 0 & 0 & 1 \\ \frac{c_2}{c_1} & -1 & 0 \end{pmatrix} \tag{4.67}$$

$$B = \begin{pmatrix} \frac{1}{r_1} \\ 0 \\ 0 \end{pmatrix} \tag{4.68}$$

$$C = \begin{pmatrix} \frac{(-1)}{(r_1 c_1)} & 0 & 0 \\ \frac{1}{c_1} & 0 & 0 \end{pmatrix} \tag{4.69}$$

$$D = \begin{pmatrix} \frac{1}{r_1} \\ 0 \end{pmatrix} \tag{4.70}$$

c_1 removed

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{4.71}$$

$$A = \begin{pmatrix} \frac{(-(r_2+r_3))}{(r_2 r_3 c_2)} & \frac{1}{r_2} \\ \frac{1}{(r_2 c_2)} & \frac{(-(r_2+r_1))}{(r_2 r_1)} \end{pmatrix} \tag{4.72}$$

$$B = \begin{pmatrix} 0 \\ \frac{1}{r_1} \end{pmatrix} \tag{4.73}$$

$$C = \begin{pmatrix} 0 & \frac{(-1)}{r_1} \\ \frac{1}{c_2} & 0 \end{pmatrix} \tag{4.74}$$

$$D = \begin{pmatrix} \frac{1}{r_1} \\ 0 \end{pmatrix} \tag{4.75}$$

c_1 and c_2 removed

$$E = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{4.76}$$

$$A = \begin{pmatrix} \frac{(-(r_2+r_1))}{(r_2 r_1)} & \frac{1}{r_2} \\ \frac{1}{r_2} & \frac{(-(r_2+r_3))}{(r_2 r_3)} \end{pmatrix} \tag{4.77}$$

$$B = \begin{pmatrix} \frac{1}{r_1} \\ 0 \end{pmatrix} \tag{4.78}$$

$$C = \begin{pmatrix} \frac{(-1)}{r_1} & 0 \\ 0 & 1 \end{pmatrix} \tag{4.79}$$

$$D = \begin{pmatrix} \frac{1}{r_1} \\ 0 \end{pmatrix} \tag{4.80}$$

r_1 *removed*

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.81}$$

$$A = \begin{pmatrix} \frac{(-(r_2+r_3))}{(r_2 r_3 c_2)} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \tag{4.82}$$

$$B = \begin{pmatrix} \frac{1}{r_2} \\ 0 \\ c_1 \end{pmatrix} \tag{4.83}$$

$$C = \begin{pmatrix} \frac{(-1)}{(r_2 c_2)} & 0 & 1 \\ \frac{1}{c_2} & 0 & 0 \end{pmatrix} \tag{4.84}$$

$$D = \begin{pmatrix} \frac{1}{r_2} \\ 0 \end{pmatrix} \tag{4.85}$$

4.13 TRANSFER FUNCTIONS

Much of the linear analysis and design of systems and control system is concerned with *transfer functions.* The reader is referred to the many standard texts for further information on this; this section just gives an outline of the main ideas.

A basic formula of Laplace transform theory states that if $\bar{X}(s)$ is the Laplace transform of $\tilde{X}(t)$ then if $\tilde{X}(0) = 0$ the Laplace transform of $\dot{\tilde{X}}$ is

$$s\bar{X}(s) \tag{4.86}$$

Thus using the overbar notation to indicate Laplace transform, the linearised descriptor equations 4.54 can be rewritten in transformed form as

$$\begin{aligned} sE\bar{X}(s) &= A\bar{X}(s) + B\bar{u}(s) \\ \bar{Y}(s) &= C\bar{X}(s) + D\bar{u}(s) \end{aligned} \tag{4.87}$$

Eliminating $\bar{X}(s)$ gives

$$G(s) = \frac{\bar{Y}(s)}{\bar{u}(s)} = C\left[sE - A\right]^{-1} B + D \tag{4.88}$$

In the special case that $E = I$ (the unit matrix)

$$G(s) = \frac{\bar{Y}(s)}{\bar{u}(s)} = C\left[sI - A\right]^{-1} B + D \tag{4.89}$$

4.13.1 Example

The transfer functions corresponding to the descriptor matrices of Section 4.12.1 are:

Full circuit

$$G_{11}(s) = \frac{1 + (r_2c_1 + r_3c_1 + r_3c_2)s + r_2r_3c_1c_2s^2}{(r_1 + r_2 + r_3) + (r_1r_2c_1 + r_1r_3c_1 + r_1r_3c_2 + r_2r_3c_2)s + r_1r_2r_3c_1c_2s^2} \tag{4.90}$$

$$G_{21}(s) = \frac{r_3}{(r_1 + r_2 + r_3) + (r_1r_2c_1 + r_1r_3c_1 + r_1r_3c_2 + r_2r_3c_2)s + r_1r_2r_3c_1c_2s^2} \tag{4.91}$$

r_2*removed*

$$G_{11}(s) = \frac{1 + (r_3(c_1 + c_2))s}{(r_3 + r_1) + (r_3r_1(c_1 + c_2))s} \tag{4.92}$$

$$G_{21}(s) = \frac{r_3}{(r_3 + r_1) + (r_3r_1(c_1 + c_2))s} \tag{4.93}$$

c_1 *removed*

$$G_{11}(s) = \frac{1 + r_3c_2s}{(r_3 + r_2 + r_1) + (r_3c_2(r_2 + r_1))s} \tag{4.94}$$

$$G_{21}(s) = \frac{r_3}{(r_3 + r_2 + r_1) + (r_3c_2(r_2 + r_1))s} \tag{4.95}$$

c_1 *and* c_2 *removed*

$$G_{11}(s) = \frac{1}{(r_2 + r_1 + r_3)} \tag{4.96}$$

$$G_{21}(s) = \frac{r_3}{(r_2 + r_1 + r_3)} \tag{4.97}$$

r_1*removed*

$$G_{11}(s) = \frac{1 + (r_2c_1 + r_3c_1 + r_3c_2)s + r_2r_3c_1c_2s^2}{(r_2 + r_3) + r_2r_3c_2s} \tag{4.98}$$

$$G_{21}(s) = \frac{r_3}{(r_2 + r_3) + r_2r_3c_2s} \tag{4.99}$$

4.14 SIMULATION CODE

Simulation is an important tool for analysing systems. Much simulation is done using special-purpose simulation languages or, even worse, general purpose programming languages. An important theme of this book is that systems should be described at a high level using bond graphs: a representation suitable for simulation should thus be a lower level description *derived* from the higher level description.

One possible route to this is to use the 'Differential-algebraic' equation representation as a basis for simulation code generation.

```
function [sys,x0] = elag2_ode(t,x,u,flag,xinitial, ...
r_1,...
r_2,...
r_3,...
c_1,...
c_2)
%function [sys,x0] = elag2_ode(t,x,u,flag,xinitial, ...
%r_1,...
%r_2,...
%r_3,...
%c_1,...
%c_2)
%ode in simulab form for system elag2
%file elag2_ode.m
%generated by mtt
if nargin<4; flag=0; end;
if (abs(flag) == 1) | (abs(flag) == 3);
% set up the state variables;
X1 = x(1);
X2 = x(2);
% set up the input variables;
u1 = u(1);
end;
if abs(flag) == 1 %state derivative;
ans1=-X1*r_1*c_2-X1*r_2*c_2+X2*r_1*c_1+u1*r_2*c_2*c_1;
dX(1,1)=ans1/(r_1*r_2*c_2*c_1);
dX(2,1)=(X1*c_2*r_3-X2*r_2*c_1-X2*c_1*r_3)/(r_2*c_2*c_1*r_3);
sys = dx;
elseif abs(flag) == 3 %outputs;
y(1,1)=(-X1+u1*c_1)/(r_1*c_1);
y(2,1)=X2/c_2;
sys = y;
elseif abs(flag) == 0 %structure;
sys = [2,0,2,1,0,0];
if nargin<5; xinitial = zeros(2,1); end;
x0  = xinitial;
end;
```

Table 4.2 Simulink code

As an example of this, the Simulink code, automatically generated for the electrical second-order lag is given in Table 4.2.

Another set of important tools is based on transfer function representations and frequency-domain analysis; many of these have been implemented in the Matlab environment. Once again, MTT can generate appropriate representations in the Matlab

```
function [num, comden] = elag2_tf(...
r_1,...
r_2,...
r_3,...
c_1,...
c_2)
%function [num, comden] = elag2_mtf(...
%r_1,...
%r_2,...
%r_3,...
%c_1,...
%c_2)
%transfer function in mv toolbox form for system elag2
%file elag2_mtf.m
%generated by mtt
comden = zeros(1,3);
num =    zeros(2,3);
comden(1,1)=r_1*r_2*r_3*c_1*c_2;
comden(1,2)=r_1*r_2*c_1+r_1*r_3*c_1+r_1*r_3*c_2+r_2*r_3*c_2;
comden(1,3)=r_1+r_2+r_3;
num(1,1)=r_2*r_3*c_1*c_2;
num(1,2)=r_2*c_1+r_3*c_1+r_3*c_2;
num(1,3)=1;
num(2,3)=r_3;
```

Table 4.3 Matlab code

language. For example, transfer function representations appropriate to the Multivariable Frequency Domain Toolbox can be generated as displayed in Table 4.3.

This can be used within the matlab environment to generate, for example, the Nyquist diagram of Figure 4.16. For this example, all system parameters have been set to unity.

4.15 EXAMPLES

This section contains two further illustrative examples:

- a DC motor and
- an RLC circuit.

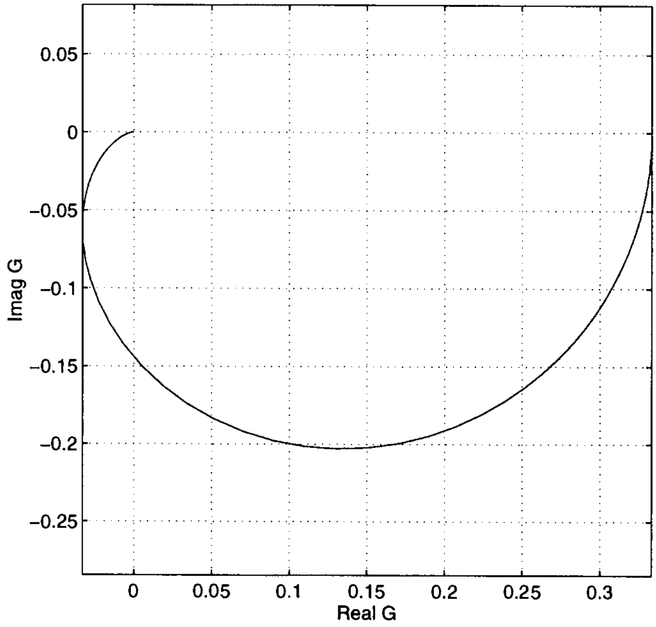


Figure 4.16 Electrical circuit: Nyquist diagram

4.15.1 DC motor

Summary

The simple DC motor of Sections 2.4.3 and 3.5.3 is used to illustrate the derived equations for two cases:

1. A voltage driven motor with integral causality.

2. A current driven motor with mixed integral/derivative causality.

Description

The bond graphs (Figures 3.40 and 3.41) of Section 3.5.3 are repeated here as Figures 4.17 and 4.18.

Both systems have two inputs u and two outputs y. In the case of the DC motor with voltage drive

$$u = \begin{pmatrix} v_a \\ \tau \end{pmatrix} ; \; y = \begin{pmatrix} i_a \\ \Omega \end{pmatrix} \tag{4.100}$$

where i_a is the armature current, v_a the armature voltage, τ the (external) applied shaft

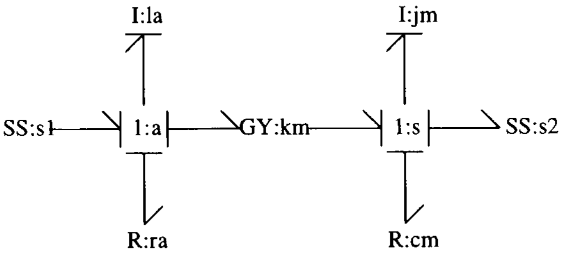


Figure 4.17 DC motor with voltage drive: bond graph

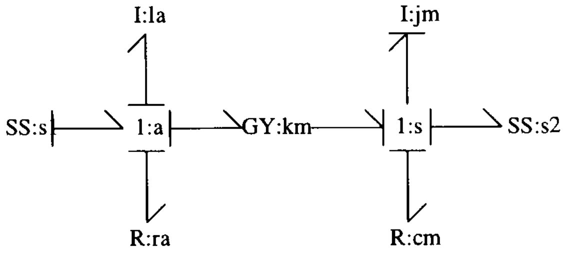


Figure 4.18 DC motor with current drive: bond graph

torque and Ω the shaft angular velocity. In the case of the DC motor with current drive

$$u = \begin{pmatrix} i_a \\ \tau \end{pmatrix} ; \; y = \begin{pmatrix} v_a \\ \Omega \end{pmatrix} \tag{4.101}$$

Differential-algebraic equation representation

The DC motor with voltage drive (Figure 4.17) is causal with no derivative causality and two components with integral causality. It therefore has an ODE representation with two states

$$\begin{aligned} \dot{x}_1 &= \frac{(-((r_a x_1 - l_a u_1) j_m + l_a k_m x_2))}{(j_m l_a)} \\ \dot{x}_2 &= \frac{(-((l_a u_2 - k_m x_1) j_m + l_a c_m x_2))}{(j_m l_a)} \end{aligned} \tag{4.102}$$

$$\begin{aligned} y_1 &= \frac{x_1}{l_a} \\ y_2 &= \frac{x_2}{j_m} \end{aligned} \tag{4.103}$$

where

$$x = (\, \lambda h \,) \tag{4.104}$$

and λ is the inductor state (flux density) and h the angular momentum of the motor.

On the other hand, the DC motor with current drive (Figure 4.18) has one component with derivative causality the system therefore has one state

$$x = (\, h \,) \tag{4.105}$$

and one non-state

$$z = (\, \lambda \,) \tag{4.106}$$

and the DAE is

$$\dot{x}_1 = \frac{((k_m u_1 - u_2) j_m - c_m x_1)}{j_m} \tag{4.107}$$

$$z_1 = l_a u_1 \tag{4.108}$$

$$\begin{aligned} y_1 &= \frac{((r_a u_1 + \dot{z}_1) j_m + k_m x_1)}{j_m} \\ y_2 &= \frac{x_1}{j_m} \end{aligned} \tag{4.109}$$

Constrained-state equations

The DC motor with current drive (Figure 4.18) has one component with derivative causality due to the imposition of the input current. Equation 4.107 has the appropriate form and can therefore be rewritten as a constrained-state equation

$$\dot{\chi}_1 = \frac{((k_m u_1 - u_2) j_m - c_m x_1)}{j_m} \tag{4.110}$$

$$y_1 = \frac{((r_a u_1 + l_a \dot{u}_1) j_m + k_m x_1)}{j_m} \tag{4.111}$$

$$y_2 = \frac{x_1}{j_m} \tag{4.112}$$

$$E = (\, 1 \,) \tag{4.113}$$

This is already in ODE form.

Descriptor/state-space matrices

The DC motor with voltage drive has an ODE representation and therefore has a State-space representation

$$A = \begin{pmatrix} \frac{(-r_a)}{l_a} & \frac{(-k_m)}{j_m} \\ \frac{k_m}{l_a} & \frac{(-c_m)}{j_m} \end{pmatrix} \tag{4.114}$$

$$B = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4.115}$$

$$C = \begin{pmatrix} \frac{1}{l_a} & 0 \\ 0 & \frac{1}{j_m} \end{pmatrix} \tag{4.116}$$

$$D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{4.117}$$

On the other hand, the DC motor with current drive has a singular E matrix.

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.118}$$

$$A = \begin{pmatrix} \frac{(-c_m)}{j_m} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \tag{4.119}$$

$$B = \begin{pmatrix} k_m & -1 \\ 0 & 0 \\ l_a & 0 \end{pmatrix} \tag{4.120}$$

$$C = \begin{pmatrix} \frac{k_m}{j_m} & 0 & 1 \\ \frac{1}{j_m} & 0 & 0 \end{pmatrix} \tag{4.121}$$

$$D = \begin{pmatrix} r_a & 0 \\ 0 & 0 \end{pmatrix} \tag{4.122}$$

Transfer function

The transfer function matrix of the DC motor with voltage drive contains four second-order transfer functions

$$G_{11}(s) = \frac{c_m + j_m s}{(r_a c_m + k_m^2) + (j_m r_a + l_a c_m)s + j_m l_a s^2} \tag{4.123}$$

$$G_{12}(s) = \frac{k_m}{(r_a c_m + k_m^2) + (j_m r_a + l_a c_m)s + j_m l_a s^2} \tag{4.124}$$

$$G_{21}(s) = \frac{k_m}{(r_a c_m + k_m^2) + (j_m r_a + l_a c_m)s + j_m l_a s^2} \tag{4.125}$$

$$G_{22}(s) = \frac{-r_a + -l_a s}{(r_a c_m + k_m^2) + (j_m r_a + l_a c_m)s + j_m l_a s^2} \tag{4.126}$$

In contrast, the DC motor with current drive has four first order transfer functions and one, relating i_a to v_a is an *improper* transfer function.

$$G_{11}(s) = \frac{(c_m r_a + k_m^2) + (j_m r_a + l_a c_m)s + j_m l_a s^2}{c_m + j_m s} \tag{4.127}$$

$$G_{12}(s) = \frac{-k_m}{c_m + j_m s} \tag{4.128}$$

$$G_{21}(s) = \frac{k_m}{c_m + j_m s} \tag{4.129}$$

$$G_{22}(s) = \frac{-1}{c_m + j_m s} \tag{4.130}$$

4.15.2 RLC circuit

Summary

The simple RLC circuit of Section 3.5.4 can be causally completed in a number of ways; each of these results in a different equation formulation for the same system.

This section illustrates this by displaying the DAE, the descriptor matrices and the transfer function for each causal pattern. Not surprisingly, the transfer function is the same in each case.

Description

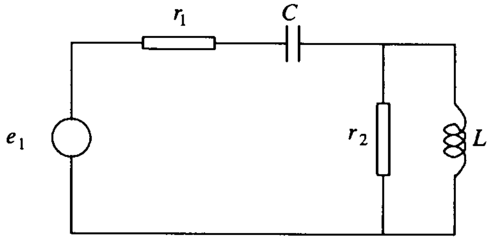


Figure 4.19 RLC circuit: schematic

The schematic diagram of Section 3.5.4 reappears as Figure 4.19; the corresponding acausal bond graph appears in Figure 4.20. The source of the SS component e1 provides the input voltage; the sensor of the SS component e2 provides the output voltage.

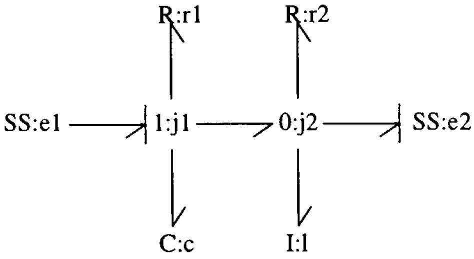


Figure 4.20 RLC circuit: bond graph

Integral causality: extra voltage source

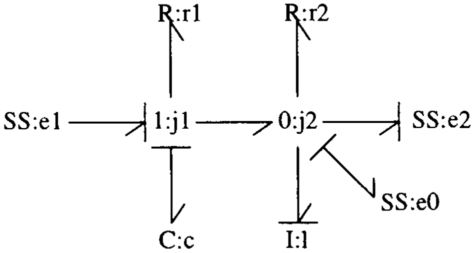


Figure 4.21 RLC circuit: bond graph with additional voltage source

Following Section 3.5.4, an extra voltage source is appended to the right-hand junction to give Figure 3.45 repeated here as Figure 4.21.

The corresponding bond graph is causal (with integral causality) and so represents an ODE. However, an additional algebraic equation arises from the constraint that the extra voltage source must have a corresponding zero current. These two equations form a DAE where the unknown source output v_1 is the solution of an algebraic equation

$$\begin{aligned} \dot{x}_1 &= \frac{(-(cx_1 + v_1 - u_1))}{r_1} \\ \dot{x}_2 &= v_1 \end{aligned} \tag{4.131}$$

$$0 = \frac{(-((r_1 + r_2)v_1 + cr_2x_1 + lr_1r_2x_2 - r_2u_1))}{(r_1r_2)} \tag{4.132}$$

$$y_1 = v_1 \tag{4.133}$$

This can be expressed in descriptor matrix form as

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.134}$$

$$A = \begin{pmatrix} \frac{(-c)}{r_1} & 0 & \frac{(-1)}{r_1} \\ 0 & 0 & 1 \\ \frac{(-c)}{r_1} & -l & \frac{(-(r_1+r_2))}{(r_1 r_2)} \end{pmatrix} \tag{4.135}$$

$$B = \begin{pmatrix} \frac{1}{r_1} \\ 0 \\ \frac{1}{r_1} \end{pmatrix} \tag{4.136}$$

$$C = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \tag{4.137}$$

$$D = \begin{pmatrix} 0 \end{pmatrix} \tag{4.138}$$

and as a transfer function

$$G(s) = \frac{+r_2 s^2}{clr_2 + (c + lr_2 r_1)s + (r_2 + r_1)s^2} \tag{4.139}$$

Note that the system has relative order zero.

Integral causality: extra current source

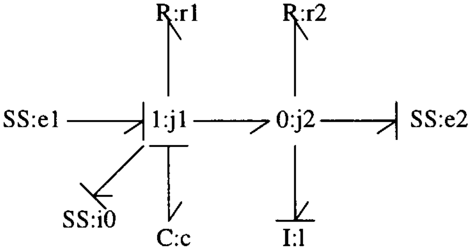


Figure 4.22 RLC circuit: bond graph with additional voltage source

Following Section 3.5.4, an extra current source is appended to the left-hand junction to give Figure 3.46 repeated here as Figure 4.22.

The corresponding bond graph is causal (with integral causality) and so represents an ODE. However, an additional algebraic equation arises from the constraint that the extra current source must have a corresponding zero voltage. These two equations form a DAE where the unknown source output v_1 is the solution of an algebraic equation

$$\begin{aligned} \dot{x}_1 &= v_1 \\ \dot{x}_2 &= -(lx_2 - v_1)r_2 \end{aligned} \tag{4.140}$$

$$0 = -((r_2 + r_1)v_1 + cx_1 - lr_2 x_2 - u_1) \tag{4.141}$$

$$y_1 = -(lx_2 - v_1)r_2 \tag{4.142}$$

This can be expressed in descriptor matrix form as

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.143}$$

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -lr_2 & r_2 \\ -c & lr_2 & -(r_2 + r_1) \end{pmatrix} \tag{4.144}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{4.145}$$

$$C = \begin{pmatrix} 0 & -lr_2 & r_2 \end{pmatrix} \tag{4.146}$$

$$D = \begin{pmatrix} 0 \end{pmatrix} \tag{4.147}$$

Although both of these representations are different to those of the previous section, because the system is the same, so is the transfer function and is given by Equation 4.139.

All derivative causality

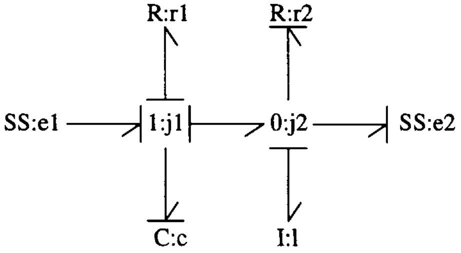


Figure 4.23 RLC circuit: bond graph with derivative causality

The bond graph is causally complete and has all derivative causality It follows that the corresponding set of equations is a set of integral equations

$$\begin{aligned} z_1 &= \frac{(-(r_1\dot{z}_1 - u_1 + \dot{z}_2))}{c} \\ z_2 &= \frac{(r_2\dot{z}_1 - \dot{z}_2)}{(lr_2)} \end{aligned} \tag{4.148}$$

$$y_1 = \dot{z}_2 \tag{4.149}$$

which can be expressed in descriptor matrix form as

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.150}$$

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & \frac{(-r_1)}{c} & \frac{(-1)}{c} \\ 0 & -1 & \frac{1}{l} & \frac{(-1)}{(lr_2)} \end{pmatrix} \tag{4.151}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{c} \\ 0 \end{pmatrix} \tag{4.152}$$

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.153}$$

$$D = \begin{pmatrix} 0 \end{pmatrix} \tag{4.154}$$

Once again, the transfer function is unchanged by the different causal pattern and is given by Equation 4.139.

Mixed causality

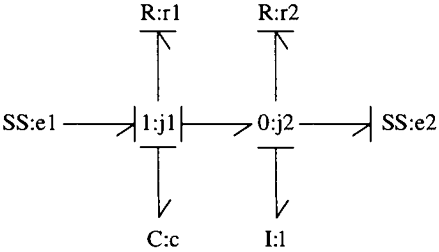


Figure 4.24 RLC circuit: bond graph with mixed causality

Figure 4.24 shows the version with mixed causality, both derivative and integral. The corresponding DAE is

$$\dot{x}_1 = \frac{(-(cx_1 - u_1 + \dot{z}_1))}{r_1} \tag{4.155}$$

$$z_1 = \frac{(-((r_1 + r_2)\dot{z}_1 + cr_2x_1 - r_2u_1))}{(lr_1r_2)} \tag{4.156}$$

$$y_1 \quad = \quad \dot{z}_1 \tag{4.157}$$

which can be expressed in descriptor matrix form as

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.158}$$

$$A = \begin{pmatrix} \frac{(-c)}{r_1} & 0 & \frac{(-1)}{r_1} \\ 0 & 0 & 1 \\ \frac{(-c)}{(lr_1)} & -1 & \frac{(-(r_1+r_2))}{(lr_1r_2)} \end{pmatrix} \tag{4.159}$$

$$B = \begin{pmatrix} \frac{1}{r_1} \\ 0 \\ \frac{1}{(lr_1)} \end{pmatrix} \tag{4.160}$$

$$C = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \tag{4.161}$$

$$D = \begin{pmatrix} 0 \end{pmatrix} \tag{4.162}$$

Once again, the transfer function is unchanged by the different causal pattern and is given by Equation 4.139.

5

System approximation

SUMMARY

- A systematic approach to model-based system approximation using bond graphs is presented.
- The causal implications of approximation are discussed.
- The derivation of the steady-state system from the bond graph is discussed.
- An extended example is given.

5.1 INTRODUCTION

System *approximation* is concerned with finding a simpler approximation to a complex model. This process has also been called *aggregation* by Simon and Ando (1961) and *model reduction.*

System modelling and system *approximation* go hand in hand: a good system model includes enough, but no more than enough, detail for the purpose for which it is required. This is essentially a tradeoff between having sufficient detail for accuracy and "seeing the wood for the trees". There are two possible approaches to system approximation.

1. Creating a differential equation model of the system and then approximating the differential equation model. This could be called a *black-box* approach.
2. Working with the bond graph model itself to remove or combine physical components. This could be called a *model-based* approach.

The latter approach is discussed here.

Thus a system modeller often has to make the decision as to which parts of the system may be approximated. Approximating system components may have far-reaching consequences on the resultant equation formulation: in bond graph terms, the *causality* of the approximate system may be quite different from that of the approximated system. Bond graphs provide a systematic way of exploring the causal implications of system approximation *before* the equations themselves are formulated.

5.2 CAUSALITY OF APPROXIMATING COMPONENTS

Component	Effort output	Flow output
R	$e = rf$	$f = \frac{1}{r}e$
I	$p = if$	$f = \frac{1}{i}p$
C	$e = \frac{1}{c}q$	$q = ce$

Table 5.1 Component constitutive relations

In this context, systematic system approximation involves examining each component and deciding whether to approximate it. In this chapter, we will consider the effect of approximating the three basic bond graph components:

- resistances (R-components),
- capacitances (C-components),
- inertias (I-components).

For simplicity, the discussion is limited to *linear* components, but the conclusions also hold for appropriate *non-linear* components. The corresponding constitutive relations appear in Table 5.1; each written in the two possible forms. A component would be approximated

- if the constitutive relation coefficient is small ($r = \epsilon$, $c = \epsilon$ or $i = \epsilon$ where ϵ is small)
- or if the inverse constitutive relation coefficient is large ($\frac{1}{r} = \epsilon$, $\frac{1}{c} = \epsilon$ or $\frac{1}{i} = \epsilon$ where ϵ is small).

In either case, system approximation involves replacing the component with by an *approximating component* of the same type but with

$$\epsilon = 0 \tag{5.1}$$

5.2.1 R components

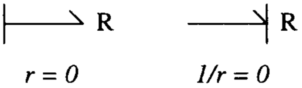


Figure 5.1 Causality of approximating R components

The assumption that infinite values of effort or flow variables are disallowed implies that only one of the two possible forms of the R equation in Table 5.1 is possible: that with finite (in fact zero) output. Thus, in these circumstances, the causality of an R component is *not* arbitary: if $r = 0$ the output is a (zero) effort; if $\frac{1}{r} = 0$, the output is a (zero) flow.

This is summarised in Figure 5.1. In effect, the R component is approximated by a *source* of appropriate causality and zero output.

5.2.2 C components

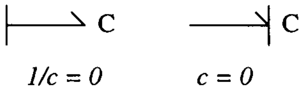


Figure 5.2 Causality of approximating C components

There are two cases to consider: $c = 0$ and $\frac{1}{c} = 0$. These will be considered in turn.

If $c = 0$, and assuming that the effort e is finite, as $q = ce$ it follows that $q = 0$ and thus $f = \dot{q} = 0$. In this case, then, the C component has a flow output. On the other hand, the effort associated with the C component is not determined by the component. The causality is summarised in Figure 5.2.

Unlike the R component, there is an *integrated* variable q associated with the C component. The restriction to finite flow f variables does *not* restrict the integral of flow q to be finite. Indeed, it makes sense to allow infinite states where this does not imply infinite inputs or outputs. For example, a large reservoir of water, with a finite water level, may be approximated by an infinite capacity containing an infinite amount of water.

In the case that $\frac{1}{c} = 0$, the appropriate form of the constitutive relationship is $e = \frac{1}{c}q$. The causality is summarised in Figure 5.2. Like the R component, the C component is approximated by a *source* of appropriate causality. Unlike the R component, the source corresponding to the the left-hand component of Figure 5.2 (integral causality) may have non-zero output.

5.2.3 I components

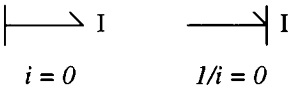


Figure 5.3 Causality of approximating I components

As with the C component, it makes sense to allow infinite states where this does not imply infinite inputs or outputs. For example, a large mass, with finite velocity, may be approximated by an infinite mass with infinite momentum.

A similar discussion leads to Figure 5.3. Once again, the I component is approximated by a *source* of appropriate causality, and the source corresponding to integral causality may have non-zero output.

5.3 REMOVING APPROXIMATING COMPONENTS

Under some circumstances, to be discussed in this section, approximating components may be removed from the bond graph; this has the benefit of leading to a simplified bond graph.

Components may be removed from a bond graph if two requirements are satisfied:

1. The component does *not* impose causality on the junction to which it is attached; that is, if connected to a *0-junction* it has a *flow* output and if connected to a *1-junction* it has an *effort* output.

2. The output of the component is zero.

These two criteria can be applied to the approximating components.

The first criterion depends on the system bond graph together with the causalities of Figures 5.1, 5.2 and 5.3.

The second criterion is automatically satisfied for R components and C and I components with derivative causality, but *not* for C and I components with integral causality (see Sections 5.2.2 and 5.2.3).

5.4 REPLACING APPROXIMATING COMPONENTS BY SOURCE-SENSORS

The approximating components discussed in Section 5.2 have two characteristics in common:

- the causality is fixed and

- the output is not dependent on the input.

These characteristics are shared with source-sensor components. It follows that an approximating component may be replaced by an SS component. There are two possibilities:

- the causality of the SS is the same as the *approximating* component

- the causality of the SS is *not* the same as the *approximating* component.

It is emphasised that the causality of the approximating component may be different from the causality of the approximated component.

In the first case, the SS source imposes a constant value (possibly zero) (effort or flow) on the system; the corresponding sensor output is the value of the other variable (flow or effort) on the SS bond.

In the second case, the sensor is constrained to have a fixed output; this is equivalent to an additional constraint equation.

5.5 CAUSAL IMPLICATIONS OF APPROXIMATING COMPONENTS

As will be seen in the example of Section 5.7, approximation changes, but does not necessarily simplify, system equations. It is helpful to the system modeller if the implications of system approximation appear at the bond graph, rather than the equation, level. This section introduces a terminology for discussing system approximation at the bond graph level. This will be used in the Example of Section 5.7.

For each component, approximation can have three possible outcomes:

1. The causality of the approximated component is the same as that of the approximating component – this will be called a *causally neutral* approximation.
2. The causality of the approximated component is not the same as that of the approximating component.
    - (a) The causal changes do not propagate beyond the junction to which the component is attached – this will be termed a *causally local* approximation.
    - (b) The causal changes do propagate beyond the junction to which the component is attached.

5.6 STEADY-STATE SOLUTIONS

A related, but different form of approximation arises from the approximation of a dynamic system by a non-dynamic system corresponding to the steady-state of the dynamic system. The steady state of a dynamic system is characterised by constant integrated effort and flow variables associated with all C and I components; it follows that the corresponding efforts and flows (respectively) must be zero.

As in Section 5.4, these (dynamic) C and I elements are replaced by (non-dynamic) SS elements.

There are four causal possibilities to consider:

1. The causality of the SS is the same as the dynamic component:
    - (a) the dynamic component had integral causality and
    - (b) the dynamic component had derivative causality.
2. The causality of the SS is *not* the same as the dynamic component:
    - (a) the dynamic component had integral causality and
    - (b) the dynamic component had derivative causality.

The two cases 1a and 1b leave the causal structure of the system unchanged; thus a causally complete system will remain so; whereas the two cases 2a and 2b change the causal structure of the system, thus a causally complete system will not necesarily remain so.

The two cases 1b and 2a give SS components with constrained (to zero) sensor signals which thus represent additional system constraints not represented on the bond graph; whereas the two cases 1a and 2b give SS components with zero source output.

The best choice of 1 or 2 depends on the system. The simplest situation arises if 1a/2b leads to a causally complete system; if it does not then it is probably simplest to stick with 1a/1b throughout. Examples appear in Section 5.7.8.

5.7 EXAMPLE: COUPLED TANKS

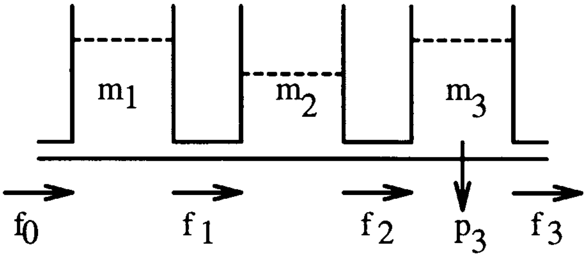


Figure 5.4 Three coupled tanks

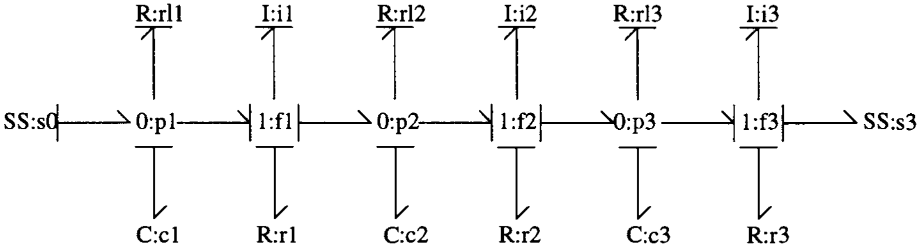


Figure 5.5 Three coupled tanks: bond graph

This section gives an illustrative example of model-based approximation using the techniques of this chapter.

Figure 5.4 shows three coupled tanks. f_0 is the inflow, f_1 and f_2 the intertank flows and f_3 is the outflow. p_i is the pressure at the base of the ith tank.

The following effects are included:

1. capacity of liquid in each tank c_i;
2. loss of liquid from each tank r_{li};
3. resistance to flow of each connecting pipe r_i;
4. inertia of liquid in each connecting pipe i_i.

For clarity, all components are assumed linear with corresponding coefficients 0, 1 or ∞.

5.7.1 Full system

Figure 5.5 shows the corresponding bond graph with causal strokes corresponding to all coefficients having value 1. The system causality is completely implied by the sources and stores, and all stores have integral causality.

The system equations are

$$x = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ m_1 \\ m_2 \\ m_3 \end{pmatrix} ;\ y = (\, s_3 \,);\ u = (\, s_0 \,) \tag{5.2}$$

$$\begin{aligned} \dot{x}_1 &= -(x_1 - x_4 + x_5) \\ \dot{x}_2 &= -(x_2 - x_5 + x_6) \\ \dot{x}_3 &= -(x_3 - x_6) \\ \dot{x}_4 &= -(x_1 + x_4 - u_1) \\ \dot{x}_5 &= x_1 - x_2 - x_5 \\ \dot{x}_6 &= x_2 - x_3 - x_6 \end{aligned} \tag{5.3}$$

$$y_1 = x_3 \tag{5.4}$$

and the transfer function is

$$G(s) = \frac{1}{13 + 38s + 51s^2 + 40s^3 + 20s^4 + 6s^5 + s^6} \tag{5.5}$$

In these equations, m_i is the mass of liquid in the ith tank and h_i the momentum of the fluid passing beween the tanks.

In terms of this example, various approximations are possible. For example:

1. Approximating leakage is causally neutral: setting $r_i = \infty$ does not change the causality of the component.

2. Approximating flow resistance is causally neutral (assuming that inertia is not approximated): setting $r_i = 0$ does not change the causality of the component.

3. Approximating inertia is not causally neutral, but (assuming that flow resistance is not approximated) the effect is local: setting $i_i = 0$ causes the flow resistance on each 1 junction to change causality.

4. Approximating both inertia and flow resistance is not causally neutral and the effects are not local; setting $r_i = 0$ and $i_i = 0$ makes the causality changes which propagate to adjacent 0-junctions.

5. Approximating capacities is not causally neutral and the effects are not local; setting $c_i = 0$ makes causality changes which propagate to adjacent 0-junctions.

5.7.2 Causally neutral approximation: approximating leakage

As noted above, approximating the tank leakage is causally neutral. Moreover, the corresponding components do not impose causality on their junctions and can therefore be removed.

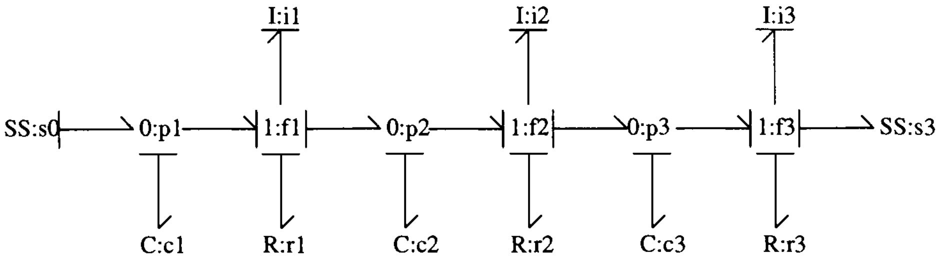


Figure 5.6 Three coupled tanks - no leakage: bond graph

Figure 5.6 shows the corresponding bond graph. The system equations are

$$x = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ m_1 \\ m_2 \\ m_3 \end{pmatrix} ; \; y = (\, s_3 \,) ; \; u = (\, s_0 \,) \tag{5.6}$$

$$\begin{aligned} \dot{x}_1 &= -(x_1 - x_4 + x_5) \\ \dot{x}_2 &= -(x_2 - x_5 + x_6) \\ \dot{x}_3 &= -(x_3 - x_6) \\ \dot{x}_4 &= -(x_1 - u_1) \\ \dot{x}_5 &= x_1 - x_2 \\ \dot{x}_6 &= x_2 - x_3 \end{aligned} \tag{5.7}$$

$$y_1 = x_3 \tag{5.8}$$

and the transfer function is

$$G(s) = \frac{1}{1 + 6s + 11s^2 + 11s^3 + 8s^4 + 3s^5 + s^6} \tag{5.9}$$

5.7.3 Causally neutral approximation: approximating flow resistance

As noted above, approximating the flow resistance is causally neutral (if the inertias are not approximated). Moreover, the corresponding components do not impose causality on their junctions and can therefore be removed. This example approximates the leakage terms as well.

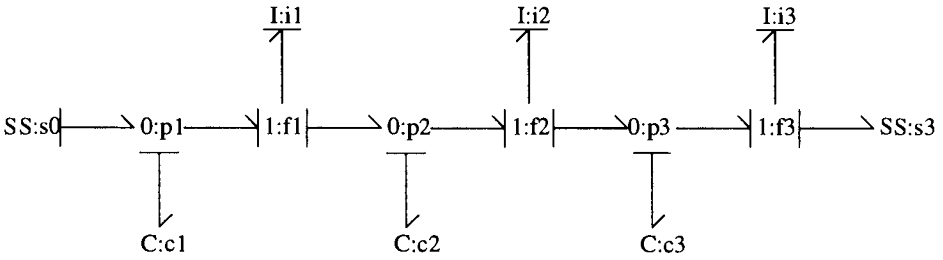


Figure 5.7 Three coupled tanks - no leakage or flow resistance: bond graph

Figure 5.7 shows the corresponding bond graph. The system equations are

$$x = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ m_1 \\ m_2 \\ m_3 \end{pmatrix} \; ; \; y = (\, s_3 \,) \; ; \; u = (\, s_0 \,) \tag{5.10}$$

$$\begin{aligned} \dot{x}_1 &= x_4 - x_5 \\ \dot{x}_2 &= x_5 - x_6 \\ \dot{x}_3 &= x_6 \\ \dot{x}_4 &= -(x_1 - u_1) \\ \dot{x}_5 &= x_1 - x_2 \\ \dot{x}_6 &= x_2 - x_3 \end{aligned} \tag{5.11}$$

$$y_1 = x_3 \tag{5.12}$$

and the transfer function is

$$G(s) = \frac{1}{1 + 6s^2 + 5s^4 + s^6} \tag{5.13}$$

5.7.4 Causally local approximation: approximating flow inertia

Assuming that flow resistance is not approximated, neglecting flow inertia is a causally local approximation. As Figure 5.8 shows, the causality of the flow resistances change, but the causal pattern is otherwise unchanged. The inertias no longer impose causality on the 1-junctions and thus the approximating zero sources have been removed.

The system equations are

$$x = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \; ; \; y = (\, s_3 \,) \; ; \; u = (\, s_0 \,) \tag{5.14}$$

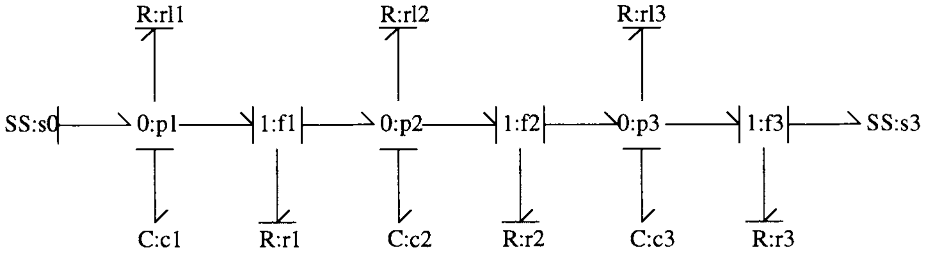


Figure 5.8 Three coupled tanks - no flow inertia: bond graph

$$
\begin{aligned}
\dot{x}_1 &= -(2x_1 - x_2 - u_1) \\
\dot{x}_2 &= x_1 - 3x_2 + x_3 \\
\dot{x}_3 &= x_2 - 3x_3
\end{aligned} \tag{5.15}
$$

$$
y_1 = x_3 \tag{5.16}
$$

and the transfer function is

$$
G(s) = \frac{1}{13 + 19s + 8s^2 + s^3} \tag{5.17}
$$

Because there are now three less energy stores, the system order has reduced from 6 to 3.

5.7.5 Causally non-local approximation: approximating flow inertia and resistance

In the absence of flow inertia, removing a flow resistance has a non-local effect on causality. In this example, the resistance labelled r_2 is removed. Figure 5.9 shows the corresponding bond graph. It is no longer possible to have integral causality on the two capacities joined via r_2: the pressures (and in this case the levels) are constrained to be equal. In Figure 5.9, the right-hand tank is given derivative causality.

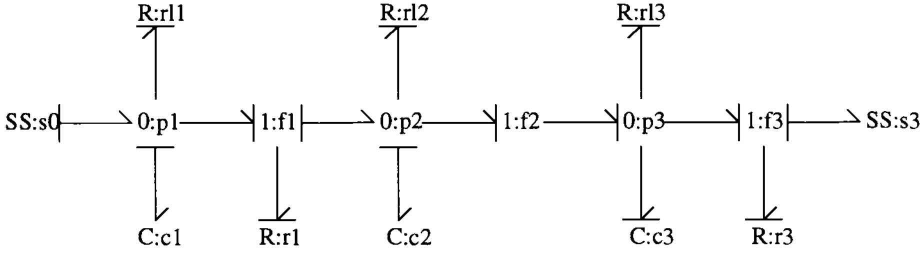


Figure 5.9 Three coupled tanks - zero flow resistance: bond graph

The system equations are

$$x = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix};\ z = (\,m_3\,);\ y = (\,s_3\,);\ u = (\,s_0\,) \tag{5.18}$$

$$\begin{aligned} \dot{x}_1 &= -(2x_1 - x_2 - u_1) \\ \dot{x}_2 &= x_1 - 4x_2 - \dot{z}_1 \end{aligned} \tag{5.19}$$

$$z_1 = x_2 \tag{5.20}$$

$$y_1 = x_2 \tag{5.21}$$

and the transfer function is

$$G(s) = \frac{1}{7 + 8s + 2s^2} \tag{5.22}$$

As there is now one less store with integral causality, the system order is reduced by 1 to 2.

5.7.6 Causally non-local approximation: approximating a capacity

Removing a capacity has a non-local effect on causality. In this example, the capacity labelled c_2 is removed. Figure 5.10 shows the corresponding bond graph. The capacity has been replaced by a source-sensor SS component, the output of which corresponds to the pressure, but the input of which is the zero flow corresponding to the zero capacity. It is not possible to replace the C element by a flow output SS element and achieve a causally complete bond graph.

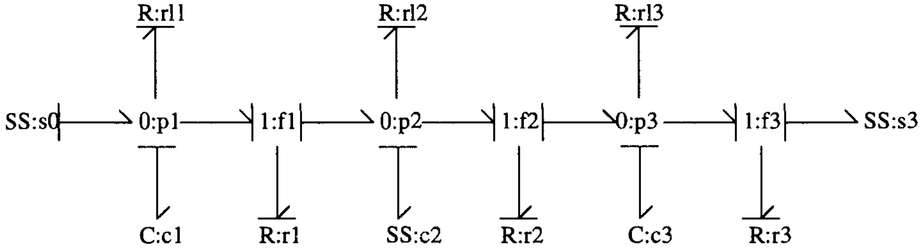


Figure 5.10 Three coupled tanks - zero capacity: bond graph

The system equations are

$$x = \begin{pmatrix} m_1 \\ m_3 \end{pmatrix};\ w = (\,c_2\,);\ y = (\,s_3\,);\ u = (\,s_0\,) \tag{5.23}$$

$$\dot{x}_1 = v_1 - 2x_1 + u_1$$

$$\dot{x}_2 = v_1 - 3x_2 \tag{5.24}$$

$$0 = -(3v_1 - x_1 - x_2) \tag{5.25}$$

$$y_1 = x_2 \tag{5.26}$$

and the transfer function is

$$G(s) = \frac{1}{13 + 13s + 3s^2} \tag{5.27}$$

As there is now one less store with integral causality, the system order is reduced by 1 to 2.

5.7.7 Causally non-local approximation: approximating a capacity and a resistance

This example looks at the combined effect of removing both an R and a C. The same capacity (the centre tank) is removed and, in addition, the R component labelled r_2 is regarded as being small. As the R component labelled r_2 can no longer impose the flow on to the corresponding junction, and it has zero (effort) output, it can be removed from the diagram.

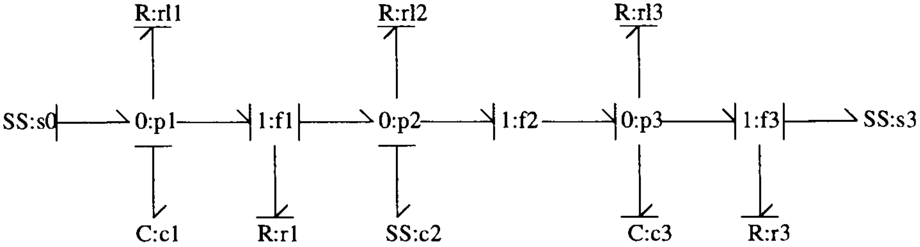


Figure 5.11 Three coupled tanks - zero capacity and resistance: bond graph

Figure 5.11 shows the corresponding bond graph. Note that derivative causality has now been imposed on the C element labelled c3. The system equations are

$$x = (\, m_1 \,)\,;\; z = (\, m_3 \,)\,;\; w = (\, c_2 \,)\,;\; y = (\, s_3 \,)\,;\; u = (\, s_0 \,) \tag{5.28}$$

$$\dot{x}_1 = v_1 - 2x_1 + u_1 \tag{5.29}$$

$$z_1 = v_1 \tag{5.30}$$

$$0 = -(4v_1 - x_1 + \dot{z}_1) \tag{5.31}$$

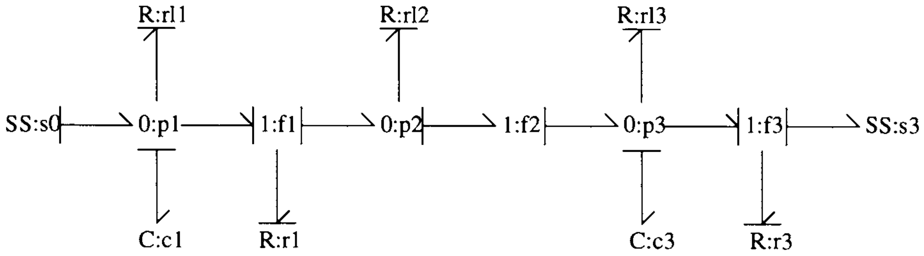


Figure 5.12 Three coupled tanks - zero capacity and resistance: alternative bond graph

$$y_1 = v_1 \tag{5.32}$$

and the transfer function is

$$G(s) = \frac{1}{7+6s+s^2} \tag{5.33}$$

As the R element labelled r2 has been removed, it is possible to complete causality even if the source replacing c2 has flow output - and this source can then be removed as in Figure 5.12. The resulting equations are

$$x = \begin{pmatrix} m_1 \\ m_3 \end{pmatrix};\ y = (\,s_3\,);\ u = (\,s_0\,) \tag{5.34}$$

$$\begin{aligned} \dot{x}_1 &= -(2x_1 - x_2 - u_1) \\ \dot{x}_2 &= x_1 - 4x_2 \end{aligned} \tag{5.35}$$

$$y_1 = x_2 \tag{5.36}$$

and the transfer function is

$$G(s) = \frac{1}{7+6s+s^2} \tag{5.37}$$

5.7.8 Steady-state solutions

The system of Section 5.7.2, with bond graph given in Figure 5.6, has integral causality throughout. With reference to Section 5.6, there is thus a choice between option 1a and 2a.

Figure 5.13 shows the corresponding bond graph where all SS causalities are reversed with respect to the six storage elements. The system causality is changed, but is causally complete. From physical considerations, it is clear that the inflow, outflow and all intermediate flows are the same.

The corresponding (steady-state) transfer function is

$$G(s) = 1 \tag{5.38}$$

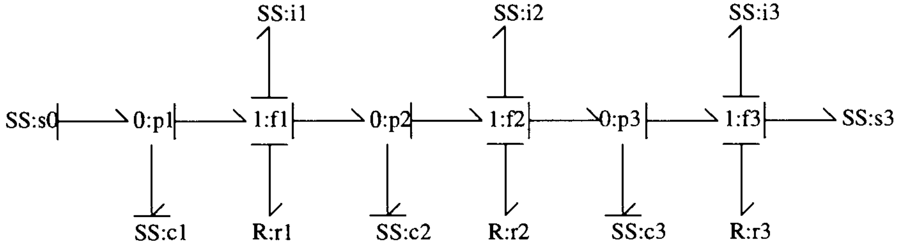


Figure 5.13 Three coupled tanks - no leakage: steady-state bond graph

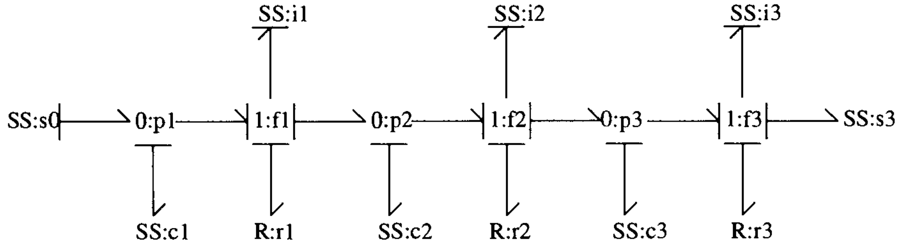


Figure 5.14 Three coupled tanks - no leakage: steady-state bond graph

Alternatively, Figure 5.14 shows the corresponding bond graph where all SS causalities remain the same with respect to the six storage elements. The system causality is unchanged, but the six additional SS elements impose six additional constraint equations. The six unknown souce outputs are labelled $v_1, ..., v_6$. The system equations are

$$\begin{aligned} 0 &= -(v_1 - v_4 + v_5) \\ 0 &= -(v_2 - v_5 + v_6) \\ 0 &= -(v_3 - v_6) \\ 0 &= -(v_1 - u_1) \\ 0 &= v_1 - v_2 \\ 0 &= v_2 - v_3 \end{aligned} \tag{5.39}$$

$$y_1 = v_3 \tag{5.40}$$

The six unknown souce outputs are labelled $v_1 ... v_6$. The corresponding (steady-state) transfer function remains the same.

A more complicated example is based on the full model of Section 5.7.1. Figure 5.15 shows the corresponding bond graph with the same causal strokes as Figure 5.5 of Section 5.7.1. There are thus six constraint equations (again in $v_1 ... v_6$) leading to the system equations

$$0 = -(v_1 - v_4 + v_5)$$

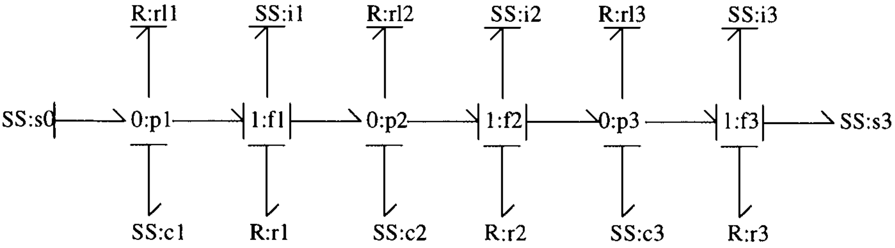


Figure 5.15 Three coupled tanks: steady-state bond graph

$$
\begin{aligned}
0 &= -(v_2 - v_5 + v_6) \\
0 &= -(v_3 - v_6) \\
0 &= -(v_1 + v_4 - u_1) \\
0 &= v_1 - v_2 - v_5 \\
0 &= v_2 - v_3 - v_6
\end{aligned}
\tag{5.41}
$$

$$
y_1 = v_3 \tag{5.42}
$$

The corresponding (steady-state) transfer function is

$$
G(s) = \frac{1}{13} \tag{5.43}
$$

Notice that this corresponds to Equation 5.5 but with $s = 0$.

In this case, the corresponding system with reversed causality on the six additional SS components is not causally complete and so algebraic equations need to be solved.

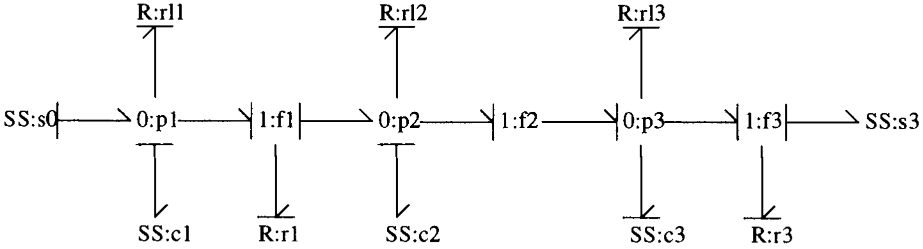


Figure 5.16 Three coupled tanks: steady-state bond graph

In Section 5.7.7 an approximate system (Figure 5.11) containing C components with both integral and derivative causality is discussed. The steady-state bond graph appears in Figure 5.16. The three C components $c_1 ... c_3$ have all been replaced by SS components. The SS components c_1 and c_2 correspond to C components with integral causality and therefore are associated with external constraint equations; the SS component c_3 corresponds to the C component with derivative causality and therfore imposes a zero flow.

There are thus two constraint equations (in v_1 and v_2) leading to the system equations

$$\begin{aligned} 0 &= -(2v_1 - v_2 - u_1) \\ 0 &= v_1 - 4v_2 \end{aligned} \tag{5.44}$$

$$y_1 = v_2 \tag{5.45}$$

The corresponding (steady-state) transfer function is

$$G(s) = \frac{1}{7} \tag{5.46}$$

Notice that this corresponds to Equation 5.33 but with $s = 0$.

6

System inversion

SUMMARY

- A system *inverse* gives the system input required to generate a given system output.
- The bond graph of a system inverse can be derived from the system bond graph.
- There is an important distinction between *collocated* and *non-collocated* sensor/actuator pairs.
- Feedback and inversion are closely related; this has a nice bond graph interpretation.

6.1 INTRODUCTION

For a given input (as a function of time), a dynamic system generates a corresponding output (as a function of time). The bond graph representation and its corresponding methodology (Karnopp *et al.*, 1990; Thoma, 1990; Wellstead, 1979) provide a systematic and insightful way of generating the equations describing such dynamic systems and, as such, support the *analysis* of dynamic systems.

For the purposes of design and *synthesis* of control systems , it is sometimes useful to find the system input (as a function of time), which will produce a given system output (as a function of time). This leads to the idea of the *inverse* of a dynamic system : that system which, given the system output at its input, will reproduce the system input at its output. Control design based on such inverse system models are used in a number of application domains. In robotics, the *computed-torque* and *feedforward* manipulator control techniques (An *et al.*, 1988; Craig, 1989) use inverse models to give the joint torques required to give a prespecified manipulator trajectory. In process engineering, *internal model control* (Morari and Zafiriou, 1989), *generic model control* (Lee and Sullivan, 1988) and *exact linearisation* strategies (Henson and Seborg, 1991) implicitly use the notion of inverse models and the corresponding ideal control. In flight control, non-linear inverse dynamics have been used as a basis for control design (Lane and Stengel, 1988). The inverse of a system has independent interest for revealing properties of the system itself (Kailath, 1980).

The usual approach to finding a system inverse is to model the system itself, derive its dynamic equations and then invert these equations. In contrast, a purpose of this chapter is to provide a way of deriving the bond graph of the inverse system from the bond graph of the system itself. Thus the inverse system is derived at the modelling, rather than the derived equation stage. This provides all the insights concerning the inverse system associated with the bond graph approach, thus allowing modelling decisions to be made in advance of equation formulation.

The *exact linearisation* strategies of Isidori (1989), reviewed in the process engineering context by Henson and Seborg (1991) augment the non-linear system with state-dependent (but non-dynamic) non-linear functions to give a linear augmented system which consists of multiple integrators. This augmented system can then be controlled using standard techniques; because the non-linear augmenting functions are state-dependent, this method requires all system states to be available. In contrast, this chapter considers system inverses which are in general dynamic and do not require (but may utilise) state information.

In the context of linear systems described by transfer functions, the inverse system may be described by the (algebraic) inverse of the system transfer function. This immediately shows some of the peculiar properties that may arise when dealing with inverse systems. For example, a strictly proper rational transfer function (more poles than zeros) will have an improper inverse (more zeros than poles). It follows that the usual representations (such as state equations) associated with proper dynamic systems will not be appropriate in the context of system inverses.

The differential-algebraic equation (or descriptor, or generalised state space, or singular system) formulation of system equations (see Section 4.7) has been suggested some time ago in the context of bond graphs by Brewer and Craig (1982) and has recently seen an upsurge of interest (van Dijk and Breedveld, 1991*a*; van Dijk and Breedveld, 1991*b*; Borutzky, 1993; van Dijk and Breedveld, 1993). In these papers, the descriptor formulation has proved useful for representing systems where causal considerations lead to derivative causality or algebraic loops. In contrast, the emphasis in this chapter is on the use of the descriptor formulation to represent inverse linear systems which are improper and non-linear systems with the corresponding property. The fundamental issue is that whereas the inverse of a state-space system is *not* a state space system (except in special circumstances) the inverse of a descriptor system is (except in special circumstances) another descriptor system. Another purpose of this chapter is, therefore, to explore the idea of using the *descriptor* system representation in this context. In the appropriate circumstances (see Section 4.10) the constrained-state equations are also useful in this context.

A further relation between control theory and inverse systems is associated with the notion of negative feedback. This chapter explores the relation between feedback and approximate system inversion in the context of bond graphs.

The chapter is organised as follows.

- Section 6.3 considers a special class of systems where the inputs (sources) and outputs (sensors are) *collocated*: they form an effort-flow pair.

- Section 6.4 considers the general N-input, N-output system.

- Section 6.6 discusses the relationship between feedback and inversion.

6.2 INVERSES AND PARTIAL INVERSES

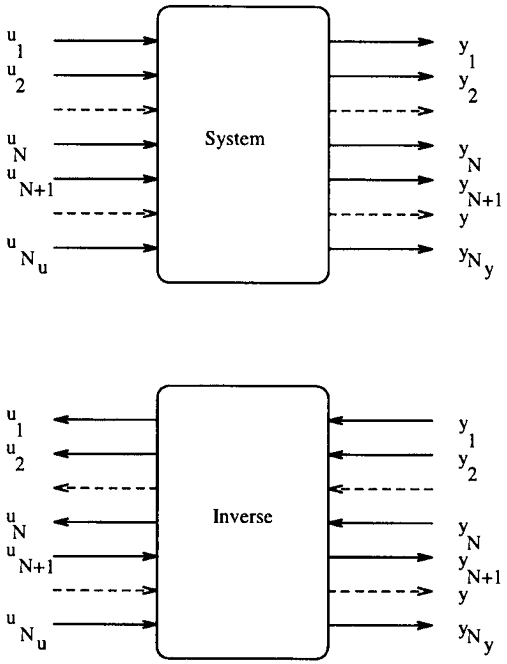


Figure 6.1 A system and its (partial) inverse

Consider the system represented in the left-hand part of Figure 6.1 with N_y outputs y_i and N_u inputs u_i. The first N inputs u_i ; $i = 1..N$ are associated with outputs y_i ; $i = 1..N$ to form the *input-output* pairs $p_i = [y_i, u_i]$. Thus the system output y and input u are

$$y = \begin{pmatrix} y_1 \\ y_2 \\ .. \\ y_N \\ y_{N+1} \\ .. \\ y_{N_y} \end{pmatrix} ; u = \begin{pmatrix} u_1 \\ u_2 \\ .. \\ u_N \\ u_{N+1} \\ .. \\ u_{N_u} \end{pmatrix} \tag{6.1}$$

This pairing is *not* an inherent feature of the particular system but rather represents the choice of the person modelling the system. The right-hand part of Figure 6.1 shows the *partial inverse* system with respect to the these input-output pairs. The output y_I

and input u_I of this partial inverse system are

$$y_I = \begin{pmatrix} u_1 \\ u_2 \\ .. \\ u_N \\ y_{N+1} \\ .. \\ y_{N_y} \end{pmatrix} ; u_I = \begin{pmatrix} y_1 \\ y_2 \\ .. \\ y_N \\ u_{N+1} \\ .. \\ u_{N_u} \end{pmatrix} \qquad (6.2)$$

When $N_y = N_u = N$ the right-hand system becomes the *inverse* of the left-hand system.

6.2.1 Example: two coupled tanks

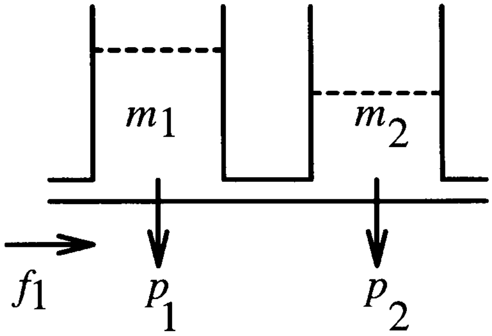


Figure 6.2 Two coupled tanks

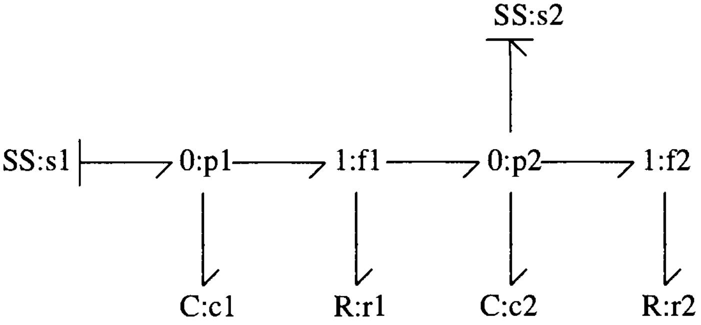


Figure 6.3 Two coupled tanks: bond graph

The system in Figure 6.2 comprises two uniform tanks of cross sections a_1 and a_2 respectively containing incompressible liquid of density ρ and coupled by short pipes of (linear) resistance r_1 and r_2 respectively. The (controlled) inflow is f_1; the pressures at the base of the tanks are p_1 and p_2 respectively; these pressures are regarded as system outputs. Making the usual approximations, the system bond graph appears in Figure 6.3.

In this case $N_u = 1$ and $N_y = 2$; and the system output and input vectors are

$$y = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}; u = \begin{pmatrix} f_1 \end{pmatrix} \tag{6.3}$$

This is a linear system and and is conveniently represented in linear state-space form (Section 4.12, Equation 4.61)

$$A = \begin{pmatrix} \frac{(-g)}{(a_1 r_1)} & \frac{g}{(r_1 a_2)} \\ \frac{g}{(a_1 r_1)} & \frac{(-(r_1+r_2)g)}{(r_1 a_2 r_2)} \end{pmatrix} \tag{6.4}$$

$$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{6.5}$$

$$C = \begin{pmatrix} \frac{g}{a_1} & 0 \\ 0 & \frac{g}{a_2} \end{pmatrix} \tag{6.6}$$

$$D = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{6.7}$$

or in transfer function matrix form (Section 4.13) as

$$G(s) = \begin{pmatrix} G_{11} \\ G_{21} \end{pmatrix} \tag{6.8}$$

where

$$G_{11}(s) = \frac{(g^2(r_1 + r_2)) + g r_1 r_2 a_2 s}{g^2 + (g(r_1 a_1 + r_2 a_2 + r_2 a_1))s + r_1 r_2 a_2 a_1 s^2} \tag{6.9}$$

$$G_{21}(s) = \frac{g^2 r_2}{g^2 + (g(r_1 a_1 + r_2 a_2 + r_2 a_1))s + r_1 r_2 a_2 a_1 s^2} \tag{6.10}$$

6.2.2 Example: two coupled rods

The system in Figure 6.4 comprises two uniform rigid rods of mass m, length $2l$ (and hence angular inertia about the mass centres of $J = \frac{ml^2}{3}$. A torque is applied at each joint. There is no gravity.

The input torque at joint 1 is τ_1 and the input torque at joint 2 is τ_2. The (relative) angular velocities associated with each joint are Ω_1 and Ω_2 respectively, and the corresponding angles are θ_1 and θ_2. The tip velocity components in absolute Cartesian coordinates are V_x and V_y, and the absolute angles of each rod are α_1 and α_2 respectively.

The system bond graph appears in Figure 6.5. (The derivation of this bond-graph is unimportant at this stage – Chapter 10 gives a detailed discussion.)

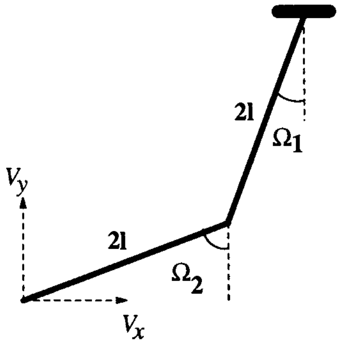


Figure 6.4 Two coupled rods

In this case, the torques are regarded as inputs and the velocities as outputs. Two of the possible pairings are described by

$$y = \begin{pmatrix} \Omega_1 \\ \Omega_2 \\ V_x \\ V_y \end{pmatrix} ; u = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \tag{6.11}$$

$$y = \begin{pmatrix} V_x \\ V_y \\ \Omega_1 \\ \Omega_2 \end{pmatrix} ; u = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \tag{6.12}$$

The rest of this section considers the first (collocated) pairing.

This non-linear system does not have a transfer function representation. It can, however be described by a *differential-algebraic* equation (Section 4.7)

$$\begin{aligned}
\dot{x}_1 &= (\dot{z}_1 + 2\dot{z}_3)\cos(x_2)l - (\dot{z}_2 + 2\dot{z}_4)\sin(x_2)l + u_1 - u_2 \\
\dot{x}_2 &= \frac{x_1}{j} \\
\dot{x}_3 &= -(\sin(x_4)l\dot{z}_4 - \cos(x_4)l\dot{z}_3 - u_2) \\
\dot{x}_4 &= \frac{x_3}{j}
\end{aligned} \tag{6.13}$$

$$\begin{aligned}
z_1 &= \frac{(-\cos(x_2)lmx_1)}{j} \\
z_2 &= \frac{(\sin(x_2)lmx_1)}{j} \\
z_3 &= \frac{(-(2\cos(x_2)x_1 + \cos(x_4)x_3)lm)}{j} \\
z_4 &= \frac{((2\sin(x_2)x_1 + \sin(x_4)x_3)lm)}{j}
\end{aligned} \tag{6.14}$$

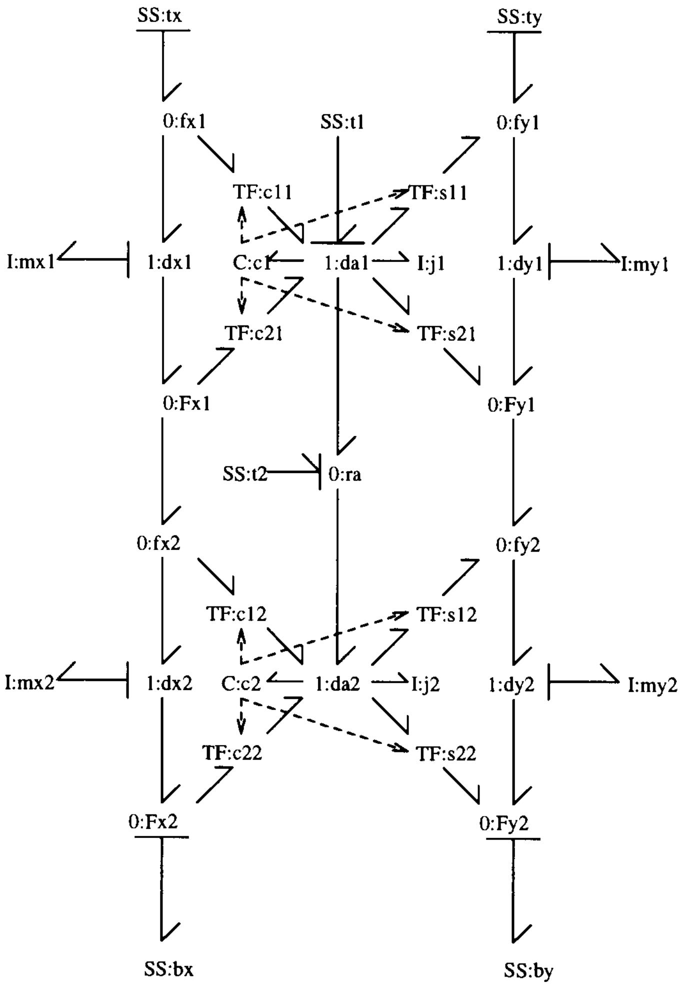


Figure 6.5 Two coupled rods

$$
\begin{aligned}
y_1 &= \frac{x_1}{j} \\
y_2 &= \frac{(-(x_1 - x_3))}{j} \\
y_3 &= \frac{(-2(\cos(x_2)x_1 + \cos(x_4)x_3)l)}{j} \\
y_4 &= \frac{(2(\sin(x_2)x_1 + \sin(x_4)x_3)l)}{j}
\end{aligned} \tag{6.15}
$$

Following Section 4.10 these equations are in *constrained-state* form and can be rewritten as the ODE

$$\dot{x}_1 = \frac{(3(ju_2 + 6sx_1^2)c - 2(u_1 - u_2)j + 12sx_3^2)}{(2(3c+4)(3c-4)j)} \tag{6.16}$$

$$\dot{x}_2 = \frac{x_1}{j} \tag{6.17}$$

$$\dot{x}_3 = \frac{(3((u_1 - u_2)j - 6sx_3^2)c - 8ju_2 - 48sx_1^2)}{(2(3c+4)(3c-4)j)} \tag{6.18}$$

$$\dot{x}_4 = \frac{x_3}{j} \tag{6.19}$$

$$y_1 = \frac{x_1}{j} \tag{6.20}$$

$$y_2 = \frac{(-(x_1 - x_3))}{j} \tag{6.21}$$

$$y_3 = \frac{(-2(cos(x_2)x_1 + cos(x_4)x_3)l)}{j} \tag{6.22}$$

$$y_4 = \frac{(2(sin(x_2)x_1 + sin(x_4)x_3)l)}{j} \tag{6.23}$$

The following substitutions have been made to simplify the equations

$$\begin{aligned} J &= \frac{ml^2}{3} \\ c &= \cos\theta_2 \\ s &= \sin\theta_2 \end{aligned} \tag{6.24}$$

6.3 COLLOCATED SENSORS AND SOURCES

Systematic modelling, including bond graphs, is based on the idea of *pairs* of variables catagorised into effort and flow (Karnopp *et al.*, 1990; Thoma, 1990; Wellstead, 1979) (the alternative across and through convention (Wellstead, 1979) will not be used here).

For the purposes of this chapter, a system has *collocated* sensors and sources if each sensor, or system output, measures the other member of the pair associated with the corresponding source, or system input. By definition, such systems have the same number of inputs and outputs.

In bond graph terms, a collocated source-sensor pair can be regarded as a source element with the corresponding measurement being the source input: the SS element of Section 3.3.7.

In such a system, then, the system inputs are the source outputs (as indicated by the causal stroke), and the system outputs are the source inputs. The inverse system, that is the system where the inputs and outputs are interchanged, is simply obtained by reversing the causality on each of the source sensors.

The algorithm for deriving the bond graph of the inverse system of a system with N *collocated* sensors and sources is then as follows.

Algorithm

1. Represent the N system input/output pairs by source-sensor (SS) elements.
2. The bond graph of the inverse system is obtained by reversing the causality of the N source-sensor (SS) elements.

6.3.1 Example: two coupled tanks

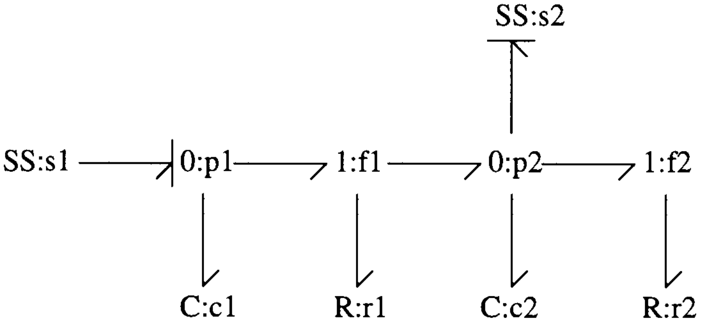


Figure 6.6 Two coupled tanks: bond graph of inverse

Continuing the example of Section 6.2.1, the first pairing corresponds to the collocated input-output pairing $u = f_1; y = p_1$. Inverting with respect to this pair gives the bond graph of Figure 6.6; the source-sensor element s1 has reversed causality.

The bond graph corresponding to the inverse system, Figure 6.6, (with collocated source-sensor) has one input and two outputs

$$u_I = p_1; y_I = \begin{pmatrix} f_1 \\ p_2 \end{pmatrix} \tag{6.25}$$

It has one state $x = m_2$ (the mass of liquid in tank 2) and one non-state $z = m_1$ (the mass of liquid in tank 1). As the sytem has a collocated source-sensor, there are no additional SS elements ($N_v = 0$). The corresponding descriptor vector is thus

$$X = \begin{pmatrix} m_2 \\ m_1 \\ \dot{m}_1 \end{pmatrix} \tag{6.26}$$

The descriptor matrices (Section 4.12) describing the (linear) inverse system are

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{6.27}$$

$$A = \begin{pmatrix} \frac{(-(r_1+r_2)g)}{(r_1 r_2 a_2)} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \tag{6.28}$$

$$B = \begin{pmatrix} \frac{1}{r_1} \\ 0 \\ \frac{a_1}{g} \end{pmatrix} \tag{6.29}$$

$$C = \begin{pmatrix} \frac{(-g)}{(r_1 a_2)} & 0 & 1 \\ \frac{g}{a_2} & 0 & 0 \end{pmatrix} \tag{6.30}$$

$$D = \begin{pmatrix} \frac{1}{r_1} \\ 0 \end{pmatrix} \tag{6.31}$$

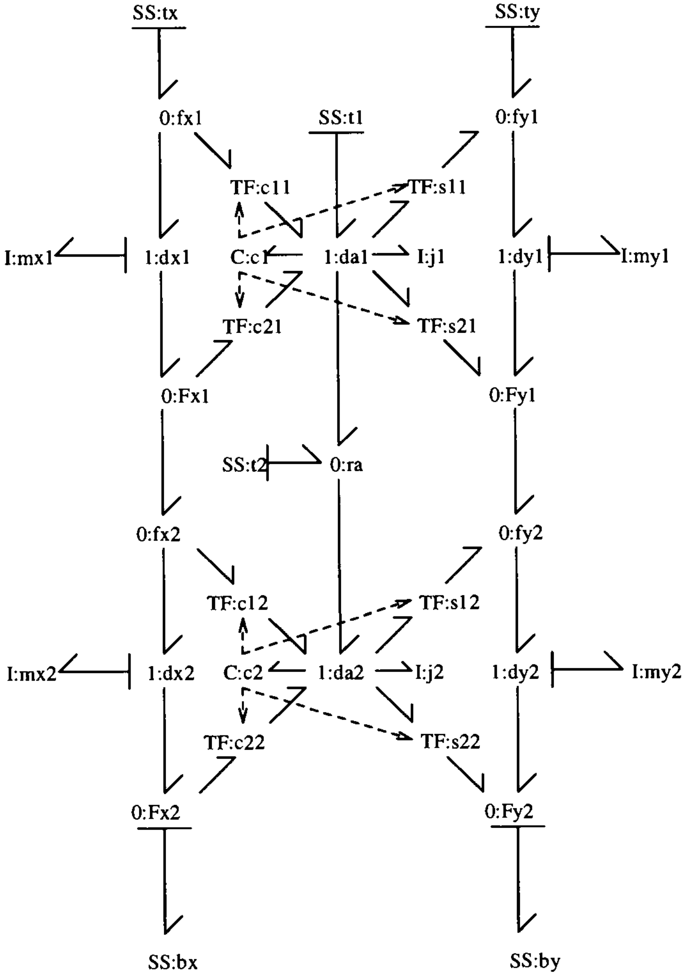


Figure 6.7 Two coupled rods: bond graph of inverse

The inverse system is conveniently represented in transfer function matrix form (Section 4.13) as

$$G(s) = \begin{pmatrix} G_{11} \\ G_{21} \end{pmatrix} \tag{6.32}$$

where

$$G_{11}(s) = \frac{g^2 + (g(r_1a_1 + a_1r_2 + r_2a_2))s + r_1a_1r_2a_2s^2}{(g^2(r_1 + r_2)) + gr_1r_2a_2s} \tag{6.33}$$

$$G_{21}(s) = \frac{gr_2}{(g(r_1 + r_2)) + r_1r_2a_2s} \tag{6.34}$$

6.3.2 Example: two coupled rods

Continuing the example of Section 6.2.2, the first pairing corresponds to the collocated input-output pair

$$y = \begin{pmatrix} \Omega_1 \\ \Omega_2 \end{pmatrix}; u = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \tag{6.35}$$

Inverting with respect to this pair gives the bond graph of Figure 6.7; the **SS** elements labeled **t1** and **t2** have reversed causality.

The bond graph corresponding to the inverse system of Figure 6.7 (with collocated source-sensors) has two inputs and two outputs

$$y_I = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}; u_I = \begin{pmatrix} \Omega_1 \\ \Omega_2 \end{pmatrix} \tag{6.36}$$

From the bond graph of Figure 6.7, it has two states α_1 and α_2 and six non-states (the angular and two translational momenta of each rod). There are no additional source sensors $N_v = 0$.

The corresponding descriptor vector is thus

$$X = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ p_{x_1} \\ p_{y_1} \\ h_1 \\ p_{x_2} \\ p_{y_2} \\ h_2 \\ \dot{h}_1 \\ \dot{p}_{x_1} \\ \dot{p}_{y_1} \\ \dot{h}_2 \\ \dot{p}_{x_2} \\ \dot{p}_{y_2} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \end{pmatrix} \tag{6.37}$$

The inverse system differential-algebraic equation is

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_1 + u_2 \end{aligned} \tag{6.38}$$

$$\begin{aligned} z_1 &= -\cos(x_1) l m u_1 \\ z_2 &= \sin(x_1) l m u_1 \\ z_3 &= j u_1 \\ z_4 &= -((u_1 + u_2)\cos(x_2) + 2\cos(x_1) u_1) l m \\ z_5 &= ((u_1 + u_2)\sin(x_2) + 2\sin(x_1) u_1) l m \\ z_6 &= (u_1 + u_2) j \end{aligned} \tag{6.39}$$

$$\begin{aligned} y_1 &= -((\dot{z}_1 + 2\dot{z}_4)\cos(x_1) l - (\dot{z}_2 + 2\dot{z}_5)\sin(x_1) l - \sin(x_2) l \dot{z}_5 + \cos(x_2) l \dot{z}_4 - \dot{z}_3 - \dot{z}_6) \\ y_2 &= \sin(x_2) l \dot{z}_5 - \cos(x_2) l \dot{z}_4 + \dot{z}_6 \\ y_3 &= -2((u_1 + u_2)\cos(x_2) + \cos(x_1) u_1) l \\ y_4 &= 2((u_1 + u_2)\sin(x_2) + \sin(x_1) u_1) l \end{aligned} \tag{6.40}$$

The tip velocities y_3 and y_4 are functions of the two joint velocities u_1 and u_2 and joint angles only.

In particular:

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = Q(x) \begin{pmatrix} \Omega_1 \\ \Omega_2 \end{pmatrix} \tag{6.41}$$

where

$$Q(x) = \begin{pmatrix} -2(\cos(x_1) + \cos(x_2)) l & -2\cos(x_2) l \\ 2(\sin(x_1) + \sin(x_2)) l & 2\sin(x_2) l \end{pmatrix} \tag{6.42}$$

Following Section 4.10 these equations are in *constrained-state* form and can be rewritten as the ODE

$$\dot{x}_1 = u_1 \tag{6.43}$$

$$\dot{x}_2 = u_1 + u_2 \tag{6.44}$$

$$y_1 = 2((3(2u_1 + u_2) s u_2 + 10\dot{u}_1 + 2\dot{u}_2) + 3(2\dot{u}_1 + \dot{u}_2) c) j \tag{6.45}$$

$$y_2 = -2((3 s u_1^2 - 2\dot{u}_1 - 2\dot{u}_2) - 3 c \dot{u}_1) j \tag{6.46}$$

$$y_3 = -2((u_1 + u_2)\cos(x_2) + \cos(x_1) u_1) l \tag{6.47}$$

$$y_4 = 2((u_1 + u_2)\sin(x_2) + \sin(x_1) u_1) l \tag{6.48}$$

Once again, Equations 6.24 have been used to simplify the system equations.

6.4 NON-COLLOCATED SENSORS AND SOURCES

In many systems, for example that discussed in Section 6.2.1, the sensors and sources are neither collocated nor quasi-collocated. Constructing the bond graph of the inverse of such systems is not as straightforward as for systems with collocated (or quasi-collocated) sensors and sources.

When the N input output pairs (Section 6.2) of a system are collocated, they can be represented by N SS elements. This section considers the more general case where some (or all) of the N input output pairs of a system are *not* collocated. In particular, it is assumed that the system has N_n sensors without corresponding collocated sources and precisely N_n sources without corresponding collocated sensors. In addition, the system may have N_c collocated source-sensor pairs. Thus the number of inputs (and corresponding outputs) is given by

$$N = N_n + N_c \tag{6.49}$$

The method for deriving the inverse system bond graph for systems containing non-collocated sensors and sources has two stages:

1. Convert the system bond graph into that of an equivalent system with collocated sensors and sources.
2. Apply Algorithm 6.3.

Part 1 of the algorithm is as follows.

6.4.1 Algorithm

1. Replace all *sources* without corresponding collocated sensors by source-sensor elements. This introduces N_n additional system *outputs* internal to the system model. The ith such internal system output is denoted v_i.
2. Replace all *sensors* without corresponding collocated sources by source-sensor elements. This introduces N_n additional system *inputs* internal to the system model. The ith such internal system input is denoted w_i
3. Impose the additional N_n *constraint equations*

$$w_i = 0 \;\; i = 1, ..., N_n \tag{6.50}$$

The additional inputs w_i, together with the additional outputs v_i, play a crucial role in determining the system inverse as detailed below.

1. Equations 6.50 ensure that the N_n additional inputs to the system have no effect on the system.
2. The internal system *inputs* w_i become internal *outputs* of the inverse system.
3. The internal system *outputs* v_i become internal *inputs* of the inverse system.

4. The N_n *constraint equations* 6.50 applied to the inverse system *outputs* w_i, implicitly imply the values of the N_n inverse system *inputs* v_i.

5. This procedure is related to that described elsewhere by Gawthrop and Smith (1992) for solving algebraic loops.

6.4.2 Example: two coupled tanks

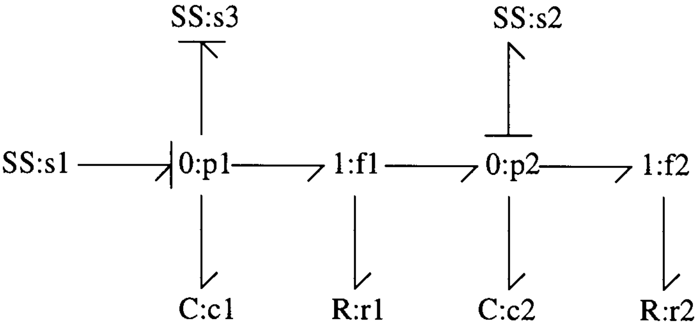


Figure 6.8 Two coupled tanks: bond graph of inverse

Continuing the example of sections 6.2.1 and 6.3.1, the second pairing corresponds to the non-collocated input-output pairing $u = f_1; y = p_2$.

Figure 6.8 shows the SS elements **s1** and **s2** with reversed causality. The additional constraint is that

$$w = 0 \tag{6.51}$$

where w is the flow associated with the additional SS element (labeled **s2**). This flow is an *output* of the *inverse* system and serves to determine p_1, the pressure associated with the SS element **s1**.

An additional SS element has been added to display the pressure p_1.

The matrices describing the (linear) inverse system are

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.52}$$

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & \frac{a_1}{g} \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & \frac{1}{r_1} \end{pmatrix} \tag{6.53}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{a_2}{g} \\ \frac{(-(r_1+r_2))}{(r_1 r_2)} \end{pmatrix} \tag{6.54}$$

$$C = \begin{pmatrix} 0 & 0 & 1 & 0 & \frac{1}{r_1} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.55}$$

$$D = \begin{pmatrix} \frac{(-1)}{r_1} \\ 0 \end{pmatrix} \tag{6.56}$$

The inverse system is conveniently represented in transfer function matrix form as

$$G(s) = \begin{pmatrix} G_{11} \\ G_{21} \end{pmatrix} \tag{6.57}$$

where

$$G_{11}(s) = \frac{g^2 + (g(a_1 r_1 + a_1 r_2 + r_2 a_2))s + a_1 r_1 r_2 a_2 s^2}{g^2 r_2} \tag{6.58}$$

$$G_{21}(s) = \frac{(g(r_1 + r_2)) + r_1 r_2 a_2 s}{g r_2} \tag{6.59}$$

6.5 QUASI-COLLOCATED SENSORS AND SOURCES

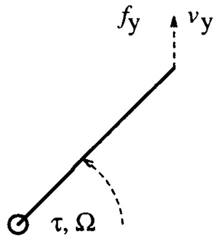


Figure 6.9 A lever with source-sensor pair

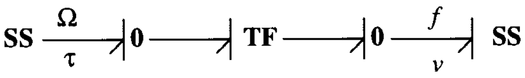


Figure 6.10 A lever with source-sensor pair: bond graph

A source-sensor pair may be spatially separated yet behave as if collocated. An example of such a pair is shown in Figure 6.9 where a torque τ acts at the pivot of a rigid rod of unit length and a velocity sensor v_y measures the vertical velocity of the free end. Now v_y is determined by Ω together with the angle θ by the equation

$$v_y = \Omega \cos \theta \tag{6.60}$$

and Ω is collocated with τ.

This suggests the following definition:

Definition 1

Given a vector of n inputs u and n *collocated* outputs y_c, the vector of n outputs y is said to be *quasi-collocated* with u iff

$$y = Q(x)y_c \tag{6.61}$$

and $Q(x)$ is an $n \times n$ non-singular matrix which is dependent only on system states.

The situation is also described by the bond graph of Figure 6.10. In particular, if the left-hand **SS** imposes the angular velocity Ω onto the system, then the causality of the right-hand **SS** is determined.

This leads to an alternative definition:

Definition 2

Given an acausal bond graph including set of n **SS** elements representing n collocated source-sensor pairs and a further set of n **SS** elements; then the outputs corresponding to the latter set are *quasi-collocated* with the inputs corresponding to the former if the inverted causality of the former set implies the causality of the latter set *without choosing the causality of any I and C elements.*

The following algorithm can be used for inverting systems with quasi collocated sensor-actuator pairs

6.5.1 Algorithm

1. Create the corresponding system with *collocated* sensors.

2. Find the inverse of this new system using Algorithm 6.3 with input $u_{cI} = y_c$ and output $y_I = u$

3. The inverse of the quasi-collocated system is then given by substituting

$$u_{cI} = Q(x)^{-1}u_I \tag{6.62}$$

where $u_I = y$

6.5.2 Example: two coupled rods

In this example, the tip xy velocities are quasi-collocated with the joint torques as the joint velocities determine the tip xy velocities. The matrix $Q(x)$ is given by Equation 6.42

6.6 FEEDBACK AND INVERSION

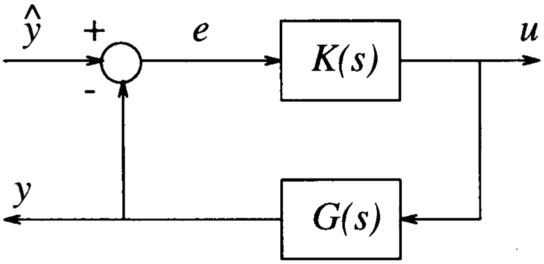


Figure 6.11 A feedback system

The notion of feedback and the notion of a system inverse are intimately connected. Consider the block diagram of Figure 6.11; the lower block $G(s)$ represents the (N-input, N-output) linear system to be inverted; the upper block $K(s)$ represents a high-gain controller. y, u and $\hat{y}$ represent the (N-dimensional) system output, input and desired output respectively.

In the particular case that

$$K(s) = \frac{1}{\epsilon}H(s)^{-1} \tag{6.63}$$

where $H(s)$ is an invertible $N \times N$ matrix, the closed-loop system can be written as

$$\frac{u(s)}{w(s)} = G_c(s) = [\epsilon H(s) + G(s)]^{-1} \tag{6.64}$$

If ϵ is small then

$$G_c(s) \approx G(s)^{-1} \tag{6.65}$$

Thus, in this rather naive sense, feedback can be used to approximately invert a system.

Another way of looking at the inverting property of feedback is to note that the negative feedback control system is designed to make the error signal $\hat{e}$ (Figure 6.11) small. That is

$$\hat{e} \approx 0 \tag{6.66}$$

and so

$$y \approx \hat{y} \tag{6.67}$$

Thus u is the input signal required to make the output of G approximately follow the desired signal $\hat{y}$. Unlike the previous argument, this does not rely on the linearity of the system G.

Of course, for $\epsilon \neq 0$, $G_c(s)$ is not necessarily a stable approximation to $G(s)^{-1}$ in the sense that the finite poles of $G_c(s)$ approximating the infinite poles of $G(s)^{-1}$ do not necessarily have negative real parts. Choosing $H(s)^{-1}$ to give a stable closed-loop transfer function $G_c(s)$ is an important issue in multivariable control theory (Maciejowski, 1989); but further discussion of this is beyond the scope of this paper.

6.6.1 Bond graph proportional control

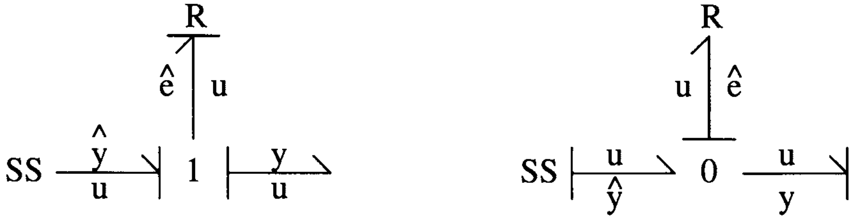


Figure 6.12 Bond graph proportional control

Instead, for the purposes of illustration, the special case where

$$H(s) = I \tag{6.68}$$

where I is the $N \times N$ unit matrix, is considered. In this case, the bond graph of the approximate inverse system corresponding to this particular feedback control, can be constructed. As expected, this procedure is simplest for a system with collocated source-sensor pairs.

In this case, the bond graph of the inverse system can be converted into that of the approximate inverse system for the particular controller given by Equation 6.68 where inputs and outputs are naturally paired. Each 'SS' element with effort output is replaced by the left-hand diagram of Figure 6.12; each 'SS' element with flow output is replaced by the right-hand diagram of Figure 6.12. In both cases, the R element has constitutive relation

$$u = \frac{\hat{e}}{\epsilon} = \frac{\hat{y} - y}{\epsilon} \tag{6.69}$$

though in the former case u is an effort, and in the latter a flow.

In the limiting case when $\epsilon = 0$, the causality of the R component reverses and the R component can be removed from the diagram. Each diagram then reverts to a single SS component, and the original inverse system is recreated.

The situation is more complicated for systems with non-collocated sensors. The diagrams in Figure 6.13 correspond to a system where each input and output pair (the pairing is not so obvious in this case, however) have opposite causality: the left-hand

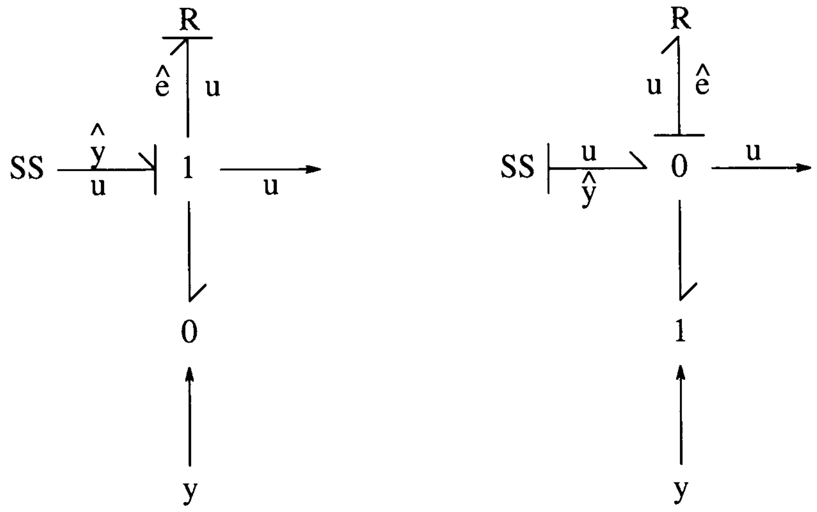


Figure 6.13 Bond graph proportional control - non collocated

diagram corresponds to an inverse system with effort input and flow output; the right-hand diagram corresponds to an inverse system with flow input and effort output. In each case, signals, or active bonds (Section 3.3.6), are used to buffer the system input and output. In this case, the limiting situation when $\epsilon = 0$ is not causally possible.

6.6.2 Example: two coupled tanks

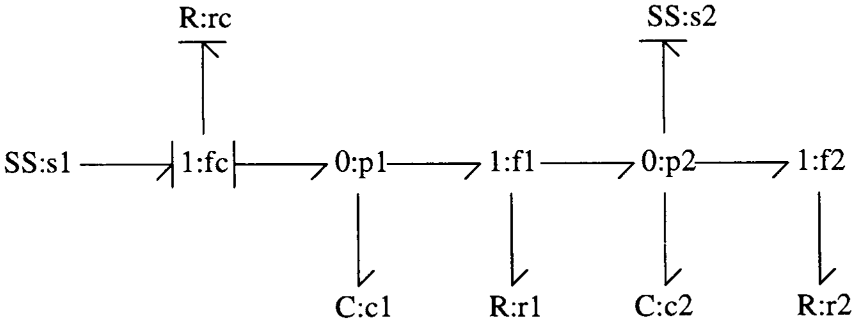


Figure 6.14 Two coupled tanks with collocated control: bond graph

A proportional controller for p_1 is appended to the two coupled tanks of Figure 6.3 to give Figure 6.14. As f_1 and p_1 are collocated, the controller of Figure 6.12 is used.

The controller gain corresponds to the conductance of the component labelled *rc* and is given by Equation 6.69, rewritten as

$$f_1 = \frac{\hat{e}}{\epsilon} = \frac{\hat{p}_1 - p_1}{\epsilon} \tag{6.70}$$

The corresponding closed-loop transfer function relating f_1 (as output) to $\hat{p}_1$ (as input) is

$$G_c(s) = \begin{pmatrix} G_{11} \\ G_{21} \end{pmatrix} \tag{6.71}$$

where

$$G_{11}(s) = \frac{g^2 + (g(r_1a_1 + a_1r_2 + r_2a_2))s + r_1a_1r_2a_2s^2}{(g^2(r_1 + r_2 + \epsilon)) + (g(r_1a_1\epsilon + r_1r_2a_2 + a_1r_2\epsilon + r_2a_2\epsilon))s + r_1a_1r_2a_2\epsilon s^2} \tag{6.72}$$

$$G_{21}(s) = \frac{g^2r_2}{(g^2(r_1 + r_2 + \epsilon)) + (g(r_1a_1\epsilon + r_1r_2a_2 + a_1r_2\epsilon + r_2a_2\epsilon))s + r_1a_1r_2a_2\epsilon s^2} \tag{6.73}$$

Notice that setting $\epsilon = 0$ gives the transfer function of the corresponding system inverse given by Equations 6.33 and 6.34.

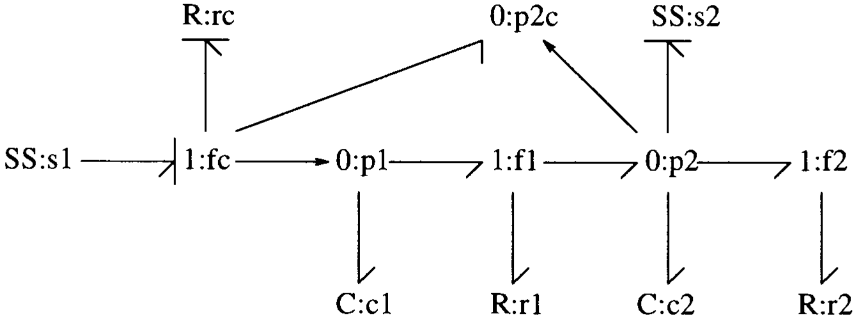


Figure 6.15 Two coupled tanks with collocated control: bond graph

A proportional controller for p_2 is appended to the two coupled tanks of Figure 6.3 to give Figure 6.15. In this case the controlled output p_2 and the input f_1 are not collocated, so the controller of Figure 6.13 is used.

The controller gain corresponds to the conductance of the component labelled *rc* and is given by Equation 6.69, rewritten as

$$f_1 = \frac{\hat{e}}{\epsilon} = \frac{\hat{p}_2 - p_2}{\epsilon} \tag{6.74}$$

The corresponding closed-loop transfer function relating f_1 (as output) to $\hat{p}_2$ (as input) is

$$G_c(s) = \begin{pmatrix} G_{11} \\ G_{21} \end{pmatrix} \tag{6.75}$$

where

$$G_{11}(s) = \frac{g^2 + (g(a_1r_1 + a_1r_2 + r_2a_2))s + a_1r_1r_2a_2s^2}{(g^2(r_2 + \epsilon)) + (g\epsilon(a_1r_1 + a_1r_2 + r_2a_2))s + a_1r_1r_2a_2\epsilon s^2} \tag{6.76}$$

$$G_{21}(s) = \frac{g^2r_2}{(g^2(r_2 + \epsilon)) + (g\epsilon(a_1r_1 + a_1r_2 + r_2a_2))s + a_1r_1r_2a_2\epsilon s^2} \tag{6.77}$$

Notice that setting $\epsilon = 0$ gives the transfer function of the corresponding system inverse given by Equations 6.58 and 6.59. This verifies the identity between control and inversion at $\epsilon = 0$.

Part II

Applications

7

An extrusion process

SUMMARY

- A procedure for developing a bond graph model from a hierarchical word bond graph is described.
- A combined energy and pseudo bond graph model of a plasticating extruder is developed.
- State equations and the steady state model of the process are derived from the bond graph.

7.1 INTRODUCTION

The motivational example described in Chapter 1 gave a descriptive model of an extrusion process used for manufacturing plastic coated cable. In this chapter, the modelling techniques described in Part I of this book are used to model the plasticating extruder which is the major sub-system of the plastic-on-wire extrusion process.

The extrusion process may be represented hierarchically, and, in this discussion, the plasticating extruder is modelled using two nested levels of hierarchy. A set of rules used for hierarchical modelling is described in Section 7.2 and these are used to build the models, and thence aggregate the complete model to a 'flat' bond graph.

A bond graph model of the process is developed as the core model from which the state equations may be derived for simulation purposes, together with the more common steady state model used for predicting machine throughput. In addition, a frequency-response analysis model may be easily derived either directly from the bond graph model or as a subsequent transformation from the state space model.

7.2 RULES FOR BUILDING HIERARCHICAL WORD BOND GRAPHS

The modelling concepts described in Part I of this book can be summarised into a set of rules which permit bond graph sub-models to be re-used in a hierarchical model. The rules are based on a 'top-down' analysis of a large system, although real modelling often iterates between 'top-down' and' bottom-up' approaches.

1. Generate the word bond graph representing the major components of the system to be modelled.
2. Decompose each complex node (sub-model) in the word bond graph into a further word bond graph, or 'flat' bond graph, as appropriate.
3. Repeat step 2 until the largest node in each word bond graph can be easily modelled using a bond graph.
4. For each sub-model so produced, formulate the acausal bond graph (or re-use sub-models from a library).
5. Define all constitutive relations in each sub-model using symbolic parameters.
6. Test each sub-model individually, to verify its behaviour.
7. Repeat steps 4 to 6 until all acausal sub-models have been generated and tested.
8. Aggregate all sub-models into a 'flat' bond graph of the complete hierarchical word bond graph.
9. Apply required inputs to the complete bond graph.
10. Apply the causal initiations appropriate to the required derived model.
11. Follow causal propagation rules to obtain the ordered equations for the derived model.

7.3 A PLASTICATING EXTRUDER

A descriptive model of this industrial application was developed in Section 1.3 (Chapter 1), to illustrate the need for a core model representation. In this case study, a hierarchical word bond graph is developed for this system following the development process described in Section 7.2. Due to the complexity of the extrusion process, significant effort could be expended developing a detailed model (Tadmor *et al.*, 1974; Reber *et al.*, 1973; Parnaby *et al.*, 1973), but in this example, the bond graph developed is a simple exploratory model suitable for understanding the basic processes. In particular, the model represents the final *metering* section of the extruder where all the polymer is molten, although similar models may be cascaded to represent the feed and transition sections of the extruder. The resulting 'flat' bond graph is then analysed, using the algorithm detailed in Chapter 3 to model the steady state performance, as would be required to predict the thickness of polymer extruded onto electrical wire.

7.3.1 Developing the hierarchical word bond graph

At the highest level of abstraction, a plasticating extruder consists of the following sub-systems:

- d.c. motor,
- heated extrusion barrel,
- extrusion screw,
- extruded polymer,
- die.

These sub-systems are inter-connected to give the word bond graph shown in Figure 7.1, where the 'SS' elements are drawn to indicate the major inputs and outputs of the system without imposing causal constraints.

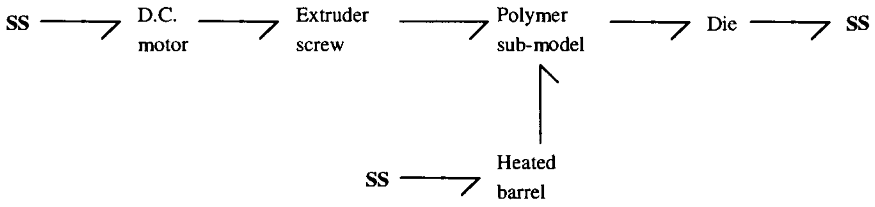


Figure 7.1 Word bond graph of plasticating extruder

Following the method of Section 7.2, shows that the only word bond node which can usefully be further decomposed into another word bond graph is the polymer sub-model. The polymer node may be modelled as two separate but interactive processes in the expanded word bond graph (Figure 7.2) for the full model.

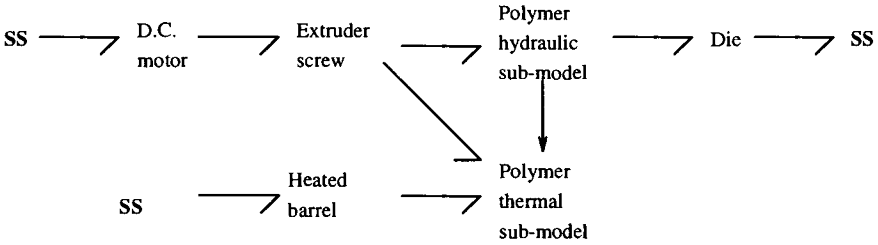


Figure 7.2 Expanded word bond graph of plasticating extruder

7.3.2 Combined energy and pseudo bond graph model

As for any engineering design, there are decisions to be made on trade-offs between alternative approaches, and in this case, the main decision is whether to use an energy bond graph or a pseudo bond graph.

For the d.c. motor, the energy bond graph model described in Section 2.4.3 is appropriate, since it describes the transduction between electrical and rotational energy domains most effectively. Since we are particularly interested in polymer *mass flow rates* through the extruder, and can linearly control the power input to the barrel heater, the pseudo bond graph appears most appropriate when modelling the polymer and heater sub-systems. Thus possible variables for the hydraulic sub-model are pressure (effort) and mass flow rate (flow), while those for the thermal sub-model are temperature and enthalpy flow rate respectively. The mass of polymer in the modelled section of the extruder is then the hydraulic state variable. The enthalpy state variable of the polymer results from enthalpy flows from the heated barrel sub-system, and from the viscous shearing action of the screw in the polymer, together with the nett enthalpy flows as polymer passes through the extruder. It can be seen that the hydraulic-enthalpic (heated tank) sub-model described in Section 2.7.2 is suitable for modelling the polymer sub-system, where the hydraulic capacitance is replaced by a capacitance representing the compressibility of the polymer. However, this would result in a stiff system model, and since most extruder models assume the polymer is incompressible, the chosen model drops this capacitance and assumes constant mass of polymer in the barrel control volume. Using this assumption the hydraulic model can revert to an energy bond graph where the bond variables are *pressure* and *volume flow rate.*

Since the extrusion screw sub-model interfaces between the d.c. motor energy bond graph and the polymer pseudo bond graph, this sub-model must contain the transformations between the variables on the input bond whose product is power, and those chosen for convenience in the pseudo bond graph. Figure 7.3 shows the acausal bond graph of the extruder screw.

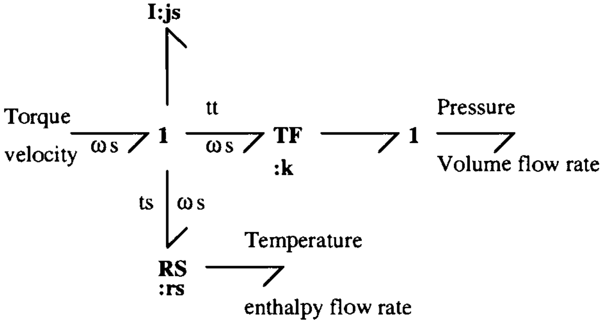


Figure 7.3 A causal sub-model of extruder screw

It can be seen that the bond variables on the bonds connected to the left-hand 1-junction are torque (effort) and angular velocity (flow). The energy conserving conversion to pressure and volume flow rate is modelled using the transformer TF:k, giving

$$\begin{aligned} pressure &= \frac{t_t}{k} \\ \text{and} & \qquad\qquad (7.1) \\ volumeflowrate &= w_s k \qquad\qquad (7.2) \end{aligned}$$

where the transformer ratio k is calculated from the internal barrel radius R, the inner screw radius r and the screw pitch P as

$$k = \frac{(R^2 - r^2)P}{2} \tag{7.3}$$

and w_s is given in radians/sec.

Polymer inertia is assumed insignificant at this point due to the low translational velocity of polymer through the extruder.

The second transformation to temperature and enthalpy flow rate provides the interface between the energy bond graph and the pseudo bond graph. This is achieved by an unconventional application of a two-port 'RS' element, where the enthalpy flow rate is given by

$$\frac{dh_s}{dt} = t_s w_s$$

and the temperature is determined by the sub-system into which the enthalpy flows.

The other external contribution of enthalpy flow to the polymer is from the electrical heaters around the barrel, as illustrated by the bond graph sub-model shown in Figure 7.4.

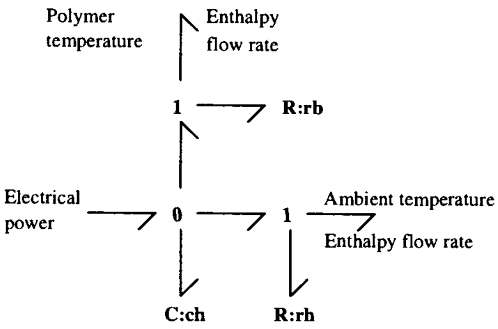


Figure 7.4 Sub-model of barrel heater sub-system

The input and output bonds on this bond graph have known causalities, as illustrated. The input flow source supplies electrical power - this is applied as a constant a.c. voltage to a resistive heater, with power controlled by linearly pulse-width modulating the on/off switching. The electrical power is sourced directly into the thermal capacitance c_h of the heater. There are two effort inputs, the polymer temperature and the ambient temperature, which are needed to calculate the enthalpy flows into the polymer and into the extruder environment, respectively. r_b models the thermal resistance of the barrel between the heater and the polymer, while r_h models the thermal resistance between the heater and its environment.

The last sub-system to be modelled is the polymer flow through the extruder die, and is shown in Figure 7.5.

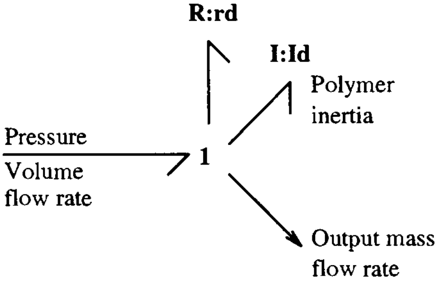


Figure 7.5 A causal bond graph of polymer extrusion through die

At the end of the barrel the molten polymer is forced through a screen filter and thence through the die resulting in a very high shear friction loss r_d.

Taking the control volume approach to calculating flow inertia (Karnopp and Rosenberg, 1975), the inertia of the polymer extruded from the die is given by: $I_d = \frac{\rho l}{a}$, where ρ is the density of the polymer, l is the length of the die channel, and a is the cross-sectional area of the die.

The large reduction in cross-sectional area of the polymer flow as it passes through the die results in a rapid increase in linear velocity, such that polymer inertia Id becomes a significant element. The output mass flow rate is a signal which is measured (by inference) from the cross-sectional area of the cooled polymer extrusion which may be subject to closed-loop control.

These five sub-systems have been aggregated in the bond graph illustrated in Figure 7.6, where causality has been completed as shown. The complete bond graph has been slightly simplified by including the motor armature friction and moment of inertia with the corresponding parameters r_s and j_s for the screw, since the latter are the dominant effects. The bonds on the graph have been numbered for reference purposes, e.g. $e_1 = e_m$.

The polymer (melt) temperature is shown as an additional output, as this variable is normally measured and automatically controlled by varying the electrical power into the barrel heaters.

In the bond graph of Figure 7.6, the dissipators r_s and r_d represent irreversible energy dissipation due to shearing of the polymer. For r_d, this energy passes out of the extruder and is dissipated in the environment, so a conventional R element is used. For r_v (the viscous friction of polymer moving through the barrel) and r_s, the energy dissipated becomes an enthalpy flow into the thermal capacitance of the polymer, via RS elements. The constitutive equations for r_s and r_d are given by Kurihara and Kimura (1985) as

$$\begin{aligned} e_7 = r_s(T_m)f_7 &= r_s(T_m)w_s\text{and} \\ e_{12} = r_d(T_m)f_{12} &= r_d(T_m)kw_s \end{aligned} \tag{7.4}$$

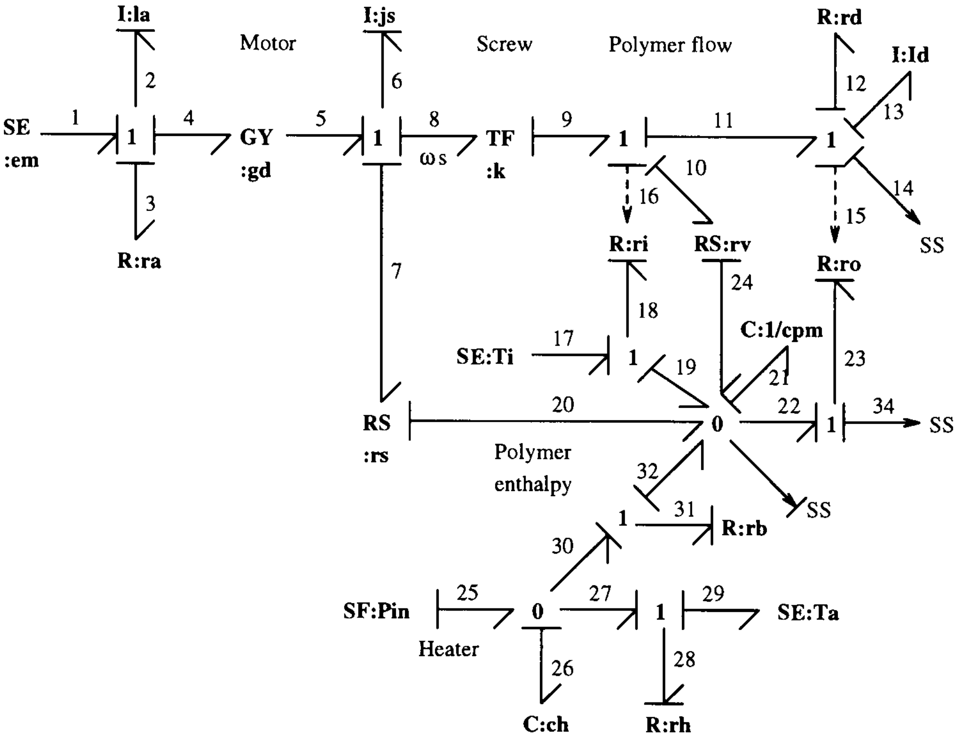


Figure 7.6 Bond graph of extruder metering section

since the polymer volume flow rate $v = kw_s$.

The viscous friction has the constitutive equation $e_{10} = r_v(T_m)f_{10} = r_v(T_m)kw_s$ where $r(T_m)$ indicates the dependence of the viscosity and shear dissipation on polymer temperature, and $T_m = T_{21} = h_{21}/(c_p m)$ where m is the mass of polymer melt in the barrel section, and c_p is the specific heat of the polymer.

The electrical heater sub-model gives equations relating enthalpy flow and temperature.

$$\begin{aligned}
f_{25} &= P_{in} \\
T_{26} &= T_h = h_{26}/(c_{ph}m_h) \\
f_{27} &= (T_h - T_a)/r_h \\
f_{30} &= (T_h - T_m)/r_b \\
f_{26} &= f_{25} - f_{27} - f_{30}
\end{aligned} \tag{7.5}$$

The enthalpy equations for the melt polymer are

$$\begin{aligned}
f_{19} &= e_{18}f_{16}/r_i = T_i v\rho c_p \\
f_{20} &= e_7 f_7 = r_s(T_m)w_s^2 \\
f_{22} &= e_{23}f_{15}/r_o = T_m v\rho c_p \\
f_{24} &= e_{10}f_{10} = r_v(T_m)v^2
\end{aligned}$$

$$\begin{aligned} f_{32} &= f_{30} \\ f_{21} &= f_{19} + f_{20} - f_{22} + f_{24} + f_{32} \end{aligned} \quad (7.6)$$

Hence the state equations can be obtained from the bond graph by propagating causality through the bond graph, first for the mechanical-hydraulic sub-model and then for the enthalpic syb-model

$$e_2 = e_m - i_a r_a - w_s/g_d \quad (7.7)$$

Based on our assumption of incompressibility of the polymer, the volume flow rate through the die is linearly related to the screw speed state variable w_s. Thus the non-state variable f_{13} resulting from the polymer inertia I_d results in a linear modification to the state equation for the screw torque

$$e_6(J_s + k^2 I_d)/J_s = i_a/g_d - w_s(r_s + k_2(r_v + r_d)) \quad (7.8)$$

The enthalpic state equation obtained from the heater equations 7.5 is

$$f_{26} = P_{in} - (T_h - T_a)/r_h - (T_h - T_m)/r_b \quad (7.9)$$

and from polymer equations 7.6

$$f_{21} = k w_s \rho c_p (T_i - T_m) + r_s w_s^2 + r_v k^2 w_s^2 + (T_h - T_m)/r_b \quad (7.10)$$

The dynamic response of the system (to changes in screw speed, for example) may be obtained by applying appropriate parameter values in these state equations.

7.3.3 Deriving the steady state model

For this example, we wish to derive the steady state equations for the system, in order to predict the output mass flow rate and the melt temperature for a given set of input conditions. This is achieved using the algorithms described in Chapter 3, where the first step - generating the dynamic model with integral causality - has been performed in the preceding section. Replacing energy stores by source-sensors u_1 to u_5, we get the steady state bond graph shown in Figure 7.7, where $u_5 = 0$ since I_d has derivative causality, and the outputs to source-sensors

$$w_1 = w_2 = w_3 = w_4 = 0$$

Since causality for the steady state bond graph is identical to that for the state Equation derivation, the steady state model re-uses the state Equations 7.7 to 7.10, with the source-sensor outputs set to zero

$$\begin{aligned} w_1 &= 0 = e_m - i_a r_a - w_s/g_d \text{and} \\ w_2 &= 0 = i_a/g_d - w_s(r_s + k_2(r_v + r_d)) \end{aligned} \quad (7.11)$$

Hence

$$w_s = \frac{e_m g_d}{(1 + r_a g_d^2(r_s + k^2(r_v + r_d))} \quad (7.12)$$

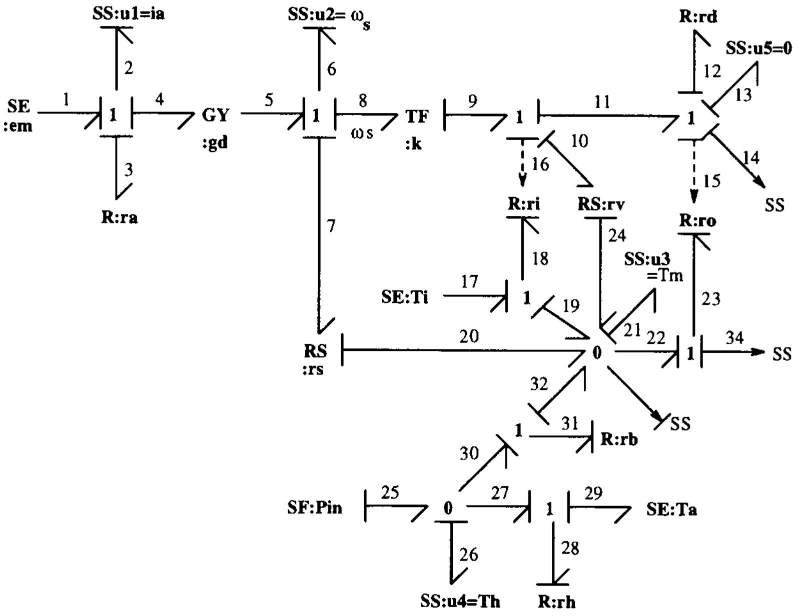


Figure 7.7 Steady state bond graph for extruder

From the heater state Equation 7.9

$$w_4 = 0 = P_{in} - (T_h - T_a)/r_h - (T_h - T_m)/r_b \tag{7.13}$$

and from the polymer state Equation 7.10

$$\begin{aligned} w_3 &= 0 \\ &= k w_s \rho c_p (T_i - T_m) + r_s w_s^2 + r_v k^2 w_s^2 + (T_h - T_m)/r_b \end{aligned} \tag{7.14}$$

Hence,

$$T_m = \frac{T_a + P_{in} r_h + (k w_s \rho c_p T_i + r_s w_s^2 + r_v k^2 w_s^2)(r_h + r_b)}{1 + k w_s \rho c_p (r_h + r_b)} \tag{7.15}$$

Note that since r_s, r_v and r_d relate to the polymer viscosity, they are all temperature dependent, and thus Equations 7.12 and 7.15 must be solved iteratively. In practice, the temperature T_m is maintained approximately constant by automatic control loops, and thus the polymer viscosity and, therefore, r_s, r_v and r_d are also approximately constant.

Equations 7.3, 7.12 and 7.15 are therefore used to optimise the extruder design for a given maximum throughput, such that the melt temperature is maintained by work heat, and the electrical input P_{in} is minimised.

The variable of interest for calculating the mass flow output of the extruder is the volume flow rate through the die

$$f_{14} = f_{11} = f_9 = k f_8$$

i.e.

$$outputmassflowrate = \rho(R^2 - r^2)Pw_s/2$$

where w_s is given by Equation 7.12.

Thus the extruded diameter of plastic on the wire may be calculated as a function of the output mass flow rate, and adjusted by controlling the angular velocity of the extruder screw.

This example has indicated how hierarchical bond graph models can be generated and how a variety of mathematical models can be derived using the procedures described in this book. In particular, the method given in Chapter 5 was used for deriving a steady state model. The model generation and analysis was performed entirely systematically, ensuring that the procedures are well suited to encoding as computer algorithms.

8

Process systems

SUMMARY

- The application of bond graphs to the modelling process engineering systems is discussed.
- The advantages of automatic model generation in this context are illustrated.
- Model-based approximation (Chapter 5) is illustrated in this context).

8.1 INTRODUCTION

Compared with other branches of engineering, process engineering is characterised by the complexity of the dynamic models associated with it and the difficulty of obtaining these models. It is all the more important, then, to have *systematic* techniques for describing, manipulating and communicating such models.

Rather than concentrate on developing specific models for specific situations, it is more worthwhile in the long run to develop *generic* techniques which can then be applied to develop specific models in a more efficient way. However, applying these techniques to process engineering systems has proved more troublesome than other domains and it has taken a concerted effort involving collaboration between chemical engineers and control engineers to make progress. Early results are reported in conference papers (MacKenzie *et al.*, 1991; MacKenzie *et al.*, 1993).

This chapter provides an introduction to the bond graph technique in the context of process engineering.

As discussed in more detail in Section 8.2, this chapter is careful to distinguish between simulation and modelling. Thus simulation code (in, for example ACSL or FORTRAN), is not regarded as being system model, but rather a *derived representation* of a system which can, in principle, be automatically generated from a higher-level system representation: the system bond graph.

The book by Franks (1972) is an important milestone along the road to systematic modelling techniques. Although Franks' book tends to equate modelling with simulation, it nevertheless makes important contributions in:

- *representing* system structure using diagrams to show the interconnection of system dynamics together with the flow of computation in system simulation;
- emphasising the importance of (computational) *causality*: the order in which simultaneous differential equations must be solved to accomplish effective numerical simulation;
- *classification* of component subsystems.

As noted in Chapter 5, bond graphs provide a basis for, *model-based* system approximation. This is illustrated here (Section 8.3) in the context of process engineering.

Section 8.2 introduces and surveys the bond graph technique in the context of automatic modelling of process engineering systems. The techniques are illustrated by a sequence of examples of increasing complexity:

- two open tanks of incompressible liquid connected by a flow resistance (Section 8.4)
- the previous example extended by the addition of a heater to each tank (Section 8.5)
- a liquid-liquid extraction process (Section 8.6)

In each case, additional derived models generated including time responses of the nonlinear and linearised systems and frequency responses of the linearised systems.

Some of this material has appeared in the literature (MacKenzie *et al.*, 1991; MacKenzie *et al.*, 1993).

8.2 MODELLING OF PROCESSES USING BOND GRAPHS

Using the standard bond graph approach (Karnopp and Rosenberg, 1975; Rosenberg and Karnopp, 1983; Karnopp *et al.*, 1990), variables describing quantities relevant to process engineering are divided into *effort* and *flow* variables. Some possibilities are given in Table 8.1

	Domain	Effort	*Units*	Flow	*Units*
1	Hydraulic	Pressure	Pa	Volume flow rate	m^3s^{-1}
2	Thermal	Temperature	K	Entropy flow rate	WK^{-1}
3	Hydraulic	Pressure	Pa	Mass flow rate	kgs^{-1}
4	Thermal	Temperature	K	Enthalpy flow rate	WK^{-1}

Table 8.1 Effort and flow variables in process engineering

The first two choices have the property that the product of the effort and flow variables is power and thus lead to true bond graphs (Karnopp *et al.*, 1990); the latter two choices do not have this property and thus lead to pseudo bond graphs (Karnopp and Rosenberg, 1975; Rosenberg and Karnopp, 1983; Karnopp *et al.*, 1990). The advantage of true bond graphs is that they can be readily coupled (via bond graph *transformers*) to other energy domains; the advantage of the latter is that they correspond to standard process engineering practice. Following Karnopp (Karnopp, 1978), the pairs 3 and 4 will be used in the rest of this chapter.

The standard bond graph junctions may be used in either case:

- At 0 junctions, the effort variables are common and the flows sum (algebraically) to zero.
- At 1 junctions, the flow variables are common and the efforts sum (algebraically) to zero.

8.2.1 Hydraulic domain

R elements

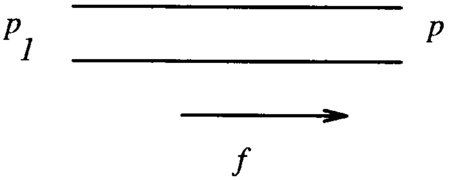


Figure 8.1 Short pipe: schematic

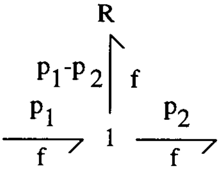


Figure 8.2 Short pipe: bond graph

R elements correspond to *flow resistance*. In this chapter, two simplifying assumptions are made:

1. the flow is dependent on pressure difference;
2. momentum effects can be neglected.

Thus the flow through the short pipe in Figure 8.1 is represented by the junction - R combination in Figure 8.2.

The assumptions leading to Figure 8.2 do *not* imply a linear CR but rather that

$$f = \phi(p_1 - p_2) \tag{8.1}$$

where ϕ is a (possibly non-linear) function of its argument. A linear CR is considered in Section 8.4.2, and a non-linear CR is considered in Section 8.4.5. More complex situations are given by Karnopp (1978).

C elements

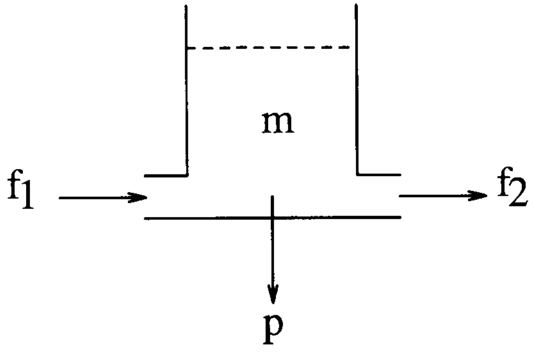


Figure 8.3 Tank: schematic

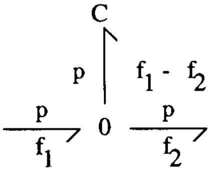


Figure 8.4 Tank: bond graph

The dynamics of chemical processes are, in this chapter, due to C elements whereby a *flow* is integrated to produce an effort. The simplest of these C elements treated here is the open-topped tank of Figure 8.3 with one inflow f_1 and one outflow f_2. The corresponding bond graph appears in Figure 8.4.

Using the mass conservation principle, the total tank mass holdup m at time t is described by

$$m(t) = \int_0^t f_1(\tau) - f_2(\tau)\, d\tau + m_0 \tag{8.2}$$

where m_0 is the holdup at time $t = 0$. Equation 8.2 is true by definition; but the constitutive relationship relating the mass holdup to the effort variable (pressure) is dependent on the tank geometry

$$p = p(m) \tag{8.3}$$

For example, as discussed in Section 8.4, a tank with uniform cross-section a would have a CR

$$p = \frac{g}{a}m \tag{8.4}$$

A more complex situation arises when the pressure is partially due to vapour trapped in the space above the liquid. This pressure depends on *temperature* and so the corresponding CR is *modulated* by the thermal domain.

8.2.2 Thermal domain

R elements

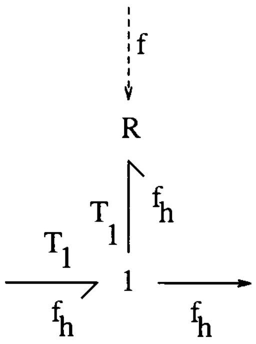


Figure 8.5 Thermal pipe: bond graph

In the context of the examples treated here, enthalpy flow is a transport phenomenon: heat is transported by the flow of a substance (Karnopp, 1978). Thus a well insulated pipe containing a substance with specific enthalpy h flowing at a rate of $fkgs^{-1}$ has an associated enthalpy flow of

$$f_h = \dot{H} = hf \tag{8.5}$$

The specific enthalpy may be written as

$$h = c_p T \tag{8.6}$$

where the specific heat c_p may be a function of temperature; so

$$f_h = \frac{T}{r_T} \tag{8.7}$$

where

$$r_T = \frac{1}{c_p f} \tag{8.8}$$

This situation is similar to that of the flow resistance: the flow variable (enthalpy flow (f_h) depends on the effort variable T; but there are two differences:

1. the flow variable depends on the upstream effort T rather than the difference between the up- and downstream efforts,
2. and the 'resistance' r_T depends on the flow from a different domain.

The corresponding bond graph in Figure 8.5 thus has two differences from that of Figure 8.2:

1. the right-hand power bond is replaced by a *signal* bond which does *not* transmit the upstream effort,
2. and the resistance is modulated from the hydraulic domain.

C elements

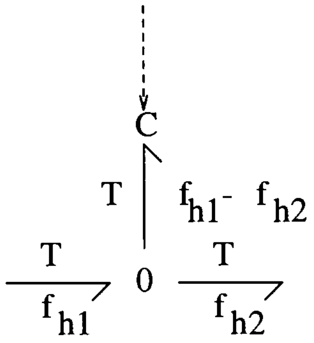


Figure 8.6 Thermal capacity: bond graph

In an analogous way to mass conservation, the energy conservation principle can be applied to a well-mixed substance to deduce that total enthalpy holdup H in the tank of Figure 8.3 at time t is described by

$$H(t) = \int_0^t f_{h1}(\tau) - f_{h2}(\tau) d\tau + H_0 \tag{8.9}$$

where f_{h1} is the inflow of enthalpy, f_{h2} is the outflow of enthalpy and H_0 is the holdup at time $t = 0$. As before, Equation 8.9 is true by definition; but the constitutive relationship relating the enthalpy holdup to the effort variable (temperature) is dependent on the mass of the substance and the thermodynamic relationships.

$$T = \frac{1}{c_p m} H \tag{8.10}$$

Once again, this is similar to the hydraulic case except that the C element is *modulated* from another domain. This is depicted in Figure 8.6. Section 8.5 expands on these concepts.

In general, some care should be taken when using *modulated* C components as energy and/or mass conservation is not automatically enforced. However, when used with care, this representation is useful as it explicitly shows the linkages between the C components in each domain.

8.2.3 Dissolved substances

	Domain	Effort	*Units*	Flow	*Units*
5	Transport	Concentration	$molkg^{-1}$	solute flow rate	kg/sec

Table 8.2 Effort and flow variables in process engineering: concentration

As mentioned in Section 8.2.2, enthalpy carried by a liquid is just one example of a transport phenomenon. Another example arises when dilute substances with concentration c_i are carried by a liquid. If m_i is the mass of the ith substance dissolved in mass M of the liquid then

$$c_i = \frac{m_i}{M} \tag{8.11}$$

The appropriate effort and flow variables are given in Table 8.2.

R elements

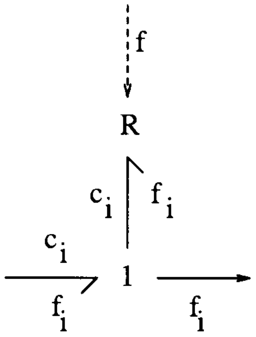


Figure 8.7 Concentration flow: bond graph

Similarly to Equation 8.5, the flow f_i of the mass of each dissolved substance is

$$f_i = \frac{1}{r_i} c_i \tag{8.12}$$

where

$$r_i = \frac{1}{f} \tag{8.13}$$

and f is the flow of the liquid. As in Section 8.2.2, the corresponding bond graph appears in Figure 8.7.

C elements

In a similar way to that in Section 8.2.2, mass conservation (of the ith dissolved substance) in the tank of Figure 8.3 gives

$$m_i(t) = \int_0^t f_{i1}(\tau) - f_{i2}(\tau)\, d\tau + m_{i0} \tag{8.14}$$

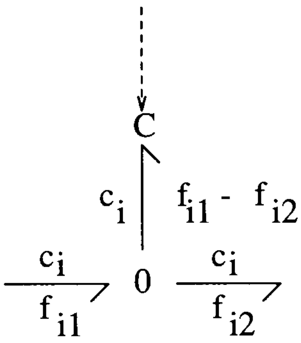


Figure 8.8 Concentration capacity: bond graph

where f_{i1} is the inflow of substance i, f_{i2} is the outflow of substance i and m_{i0} is the holdup at time 0. As before, Equation 8.14 is true by definition; but the constitutive relationship relating the (solute) mass holdup to the mass of the substance gives the CR

$$c_i = \frac{1}{M} m_i \tag{8.15}$$

Once again, this is similar to the hydraulic case except that the C element is *modulated* from another domain. This is depicted in Figure 8.8.

Section 8.6 expands on these concepts. The transport of gases in liquids is treated in Chapter 9.

8.3 SYSTEM APPROXIMATION

Although the 'correct' model of a system may be high-order, it may be useful to approximate the system by one of lower order for a number of reasons:

- to comprehend the system behaviour;
- to obtain better numerical properties - the high order system may be numerically stiff;
- to give simpler control system design.

Chapter 5 gives a more general discussion of system approximation; this chapter focuses on specifically process engineering aspects.

In the linear case, one approach is to derive the full state equations for the system and apply some standard model reduction algorithm to these equations; this is a black-box approach to model reduction. In contrast, this chapter considers a model-based approach to order reduction.

Two distinct approaches to simplifying systems are illustrated here:

- removing small capacities;
- removing small resistances.

8.3.1 Removing small capacities

As discussed in Section 5.3, if part of a process has a relatively small mass or thermal capacitance (holdup) it seem reasonable to remove it. As illustrated in Section 8.4.7, this can lead to an algebraic constraint between two variables. One approach is to just generate the corresponding equations and, in the case of simulation, allow the DAE solver to sort out any problems.

Alternatively, such algebraic constraints may be made explicit by replacing the capacity by a *source* of the same causality, and measuring the source input.

The resulting set of equations will have

- one additional input: the source output.
- one additional output: the source input.

By construction, the computational causality will be unchanged by this procedure.

The additional system input (source output) is then defined by the requirement that the corresponding system output (system input) is zero: there is no net flow into the capacity (holdup). This additional constraint forms an additional algebraic equation.

8.3.2 Removing small resistances

As discussed in Section 5.3, if part of a process has a relatively small mass or thermal resistance it seem reasonable to remove it. Once again, this leads to constraints between variables and typically constraints between states. In bond graph terms, a constraint between states leads to *derivative causality*: the corresponding component acts as a differentiator rather than an integrator. A example appears in Section 8.4.6.

As discussed in Chapter 4, the corresponding equations are *constrained-state* ordinary differential equations.

8.4 EXAMPLE: TWO COUPLED TANKS

8.4.1 Description

The system depicted in Figure 8.9 consists of two uniform open tanks containing an incompressible liquid. The fluid enters the left-hand tank, flows between the tanks, and leaves the second tank via short pipes which restrict the flow.

Assumptions

- The liquid inertia may be neglected;
- A unique pressure (relative to atmospheric pressure) can be defined at the base of each tank;

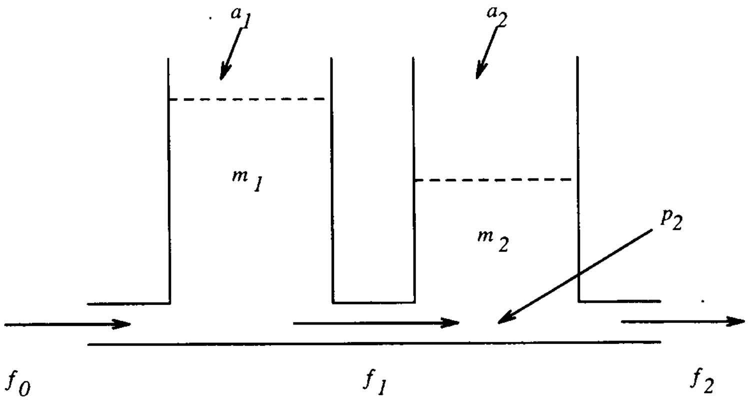


Figure 8.9 Coupled tanks: schematic

- The left-hand inflow is driven by an ideal flow source.

A number of variations on this theme are presented in the following subsections.

8.4.2 Linear flow resistance

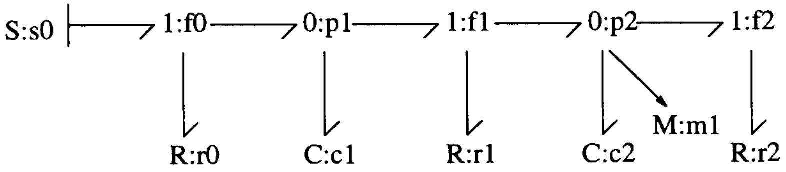


Figure 8.10 Coupled tanks (flow input): bond graph

This section presents the simplest version of the coupled tank system leading to linear system equations.

Assumptions

1. The left-hand inflow is determined by a flow source of strength $f_0 kgs^{-1}$;
2. The CR relating mass flow to pressure for each pipe is *linear.*

The first assumption leads to the causal bond graph of Figure 8.10. Notice that the left-hand flow resistance has no effect on the flow - it merely adds to the back pressure acting on the flow source.

CRs

1. Each tank, being uniform, has a linear CR relating state (mass) to pressure

$$p_i = \frac{g}{a_i} m_i \tag{8.16}$$

2. By assumption, each pipe acts as a linear resistance to flow

$$\Delta p_i = r_i f_i \tag{8.17}$$

   where Δp_i is the pressure drop across each pipe

Using the modelling tools MTT (Gawthrop, 1995) referred to in Chapter 4, a number of representations can be *automatically* derived from the bond graph of Figure 8.10. As illustration, the following representations are derived:

- differential-algebraic equations,
- descriptor matrices,
- step response,
- frequency response.

System differential-algebraic equations

$$x = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix};\ y = (\, p_2 \,);\ u = (\, f_0 \,) \tag{8.18}$$

$$\begin{aligned} \dot{x}_1 &= \frac{(-((a_2 x_1 - a_1 x_2)g - a_2 a_1 r_1 u_1))}{(a_2 a_1 r_1)} \\ \dot{x}_2 &= \frac{(-((r_1 + r_2)a_1 x_2 - a_2 r_2 x_1)g)}{(a_2 a_1 r_1 r_2)} \end{aligned} \tag{8.19}$$

$$y_1 = \frac{(g x_2)}{a_2} \tag{8.20}$$

Descriptor matrices of the linearised system

$$A = \begin{pmatrix} \frac{(-g)}{(a_1 r_1)} & \frac{g}{(r_1 a_2)} \\ \frac{g}{(a_1 r_1)} & \frac{(-(r_1+r_2)g)}{(r_1 a_2 r_2)} \end{pmatrix} \tag{8.21}$$

$$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{8.22}$$

$$C = (\, 0 \quad \tfrac{g}{a_2} \,) \tag{8.23}$$

$$D = (\, 0 \,) \tag{8.24}$$

Transfer function

$$G(s) = \frac{g^2 r_2}{g^2 + (g(a_1 r_1 + a_1 r_2 + a_2 r_2))s + a_1 a_2 r_1 r_2 s^2} \tag{8.25}$$

As expected, this system has second order transfer function (Equation 8.25). Because of the flow source, r_0 does not appear.

System properties

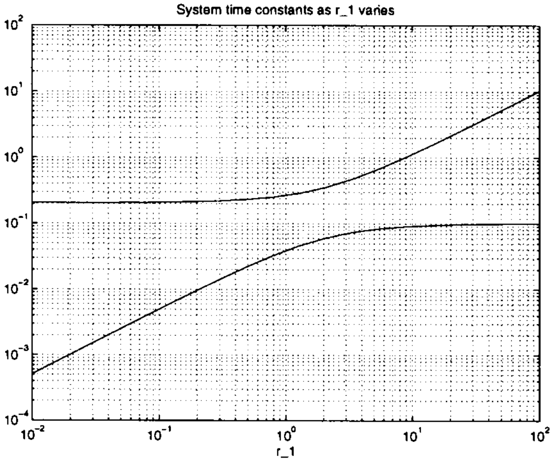


Figure 8.11 Coupled tanks: time constants

Having converted the system equation into Matlab state matrix form, the properties of the system can be analysed. For example, the effect of r_1 (the inter-tank flow resistance) is investigated. Figure 8.11 shows (on a logarithmic scale) the effect of r_1 on the two system time constants. *Decreasing* the value of r_1 gives one decreasing time constant and one time constant corresponding to *one* tank with area $a_1 + a_2$. An *increasing* value of r_1 gives one increasing time constant (tending to that of an integrator) and one tending to that of the second tank.

$$a_1 = a_2 = r_2 = 1 \tag{8.26}$$

The step response for $r_1 = 0.1, 1$ and 10 appears in Figure 8.12, and the corresponding frequency response (in Nyquist form) in Figure 8.13.

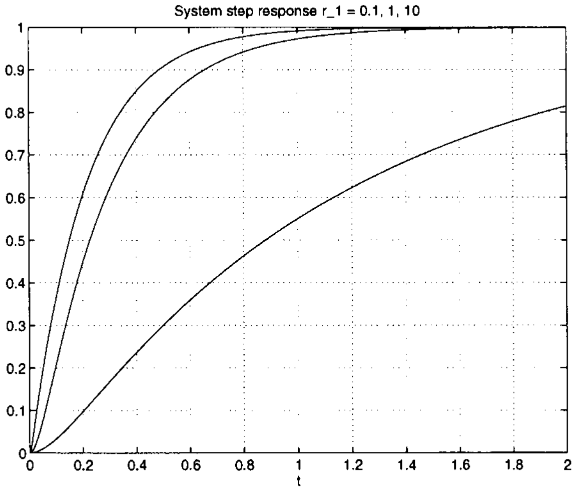


Figure 8.12 Coupled tanks: step response

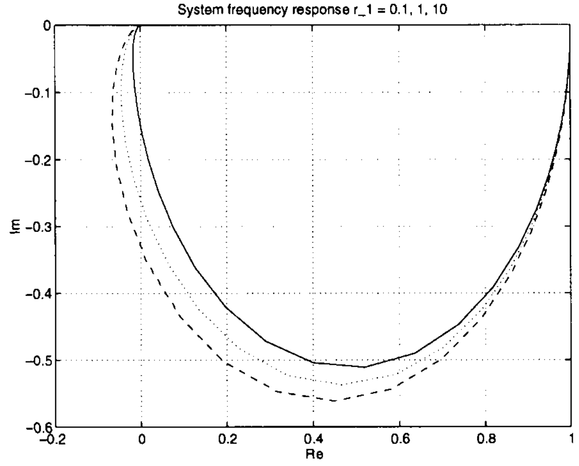


Figure 8.13 Coupled tanks: frequency response

Steady-state properties

As discussed in Section 5.6, steady-state properties can be deduced by replacing storage components (in this case **C** components) by appropriate **SS** components.

The corresponding bond graph appears in Figure 8.14. The corresponding steady-state transfer function is

$$G(s) = r_2 \tag{8.27}$$

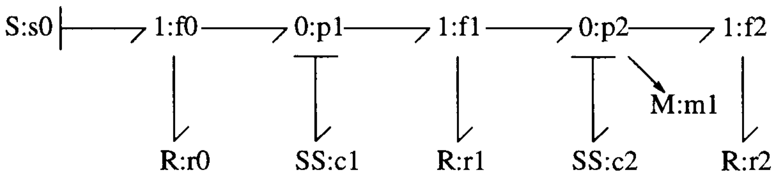


Figure 8.14 Coupled tanks - steady-state: bond graph

This could also be obtained by setting $s = 0$ in Equation 8.25.

8.4.3 No back pressure

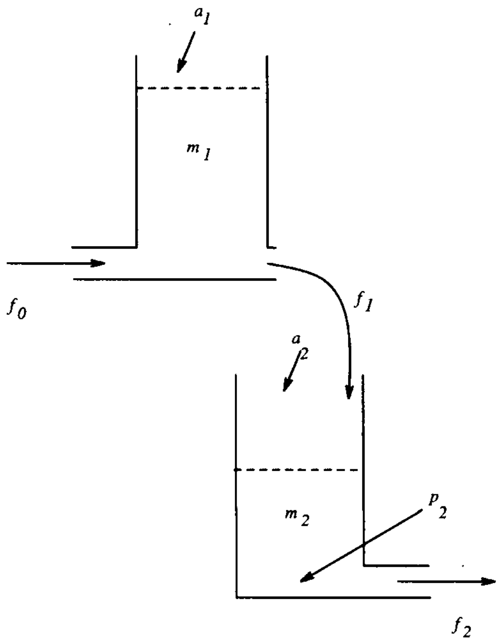


Figure 8.15 Coupled tanks (no back-pressure): schematic

The tanks appearing in Figure 8.15 are the same as those appearing in Figure 8.9, but they are connected in a different way. Unlike the tanks in Figure 8.9, the pressure in the second tank does *not* affect the inter-tank flow f_1.

The corresponding bond graph in Figure 8.16 has one change with respect to Figure 8.16: the appropriate bond (that between f1 and p2) is replaced by a signal.

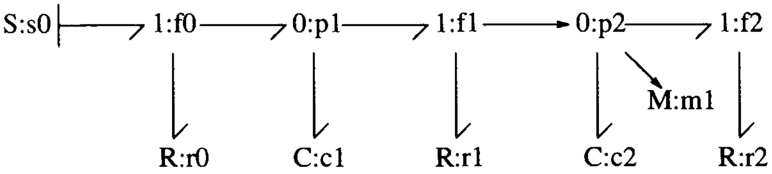


Figure 8.16 Coupled tanks (no back-pressure): bond graph

System differential-algebraic equations

$$x = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} ; \; y = (\, p_2 \,) ; \; u = (\, f_0 \,) \tag{8.28}$$

$$\begin{aligned} \dot{x}_1 &= \frac{(-(gx_1 - a_1 r_1 u_1))}{(a_1 r_1)} \\ \dot{x}_2 &= \frac{(-(a_1 r_1 x_2 - a_2 r_2 x_1) g)}{(a_1 r_1 a_2 r_2)} \end{aligned} \tag{8.29}$$

$$y_1 = \frac{(gx_2)}{a_2} \tag{8.30}$$

Descriptor matrices of the linearised system

$$A = \begin{pmatrix} \frac{(-g)}{(a_1 r_1)} & 0 \\ \frac{g}{(a_1 r_1)} & \frac{(-g)}{(a_2 r_2)} \end{pmatrix} \tag{8.31}$$

$$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{8.32}$$

$$C = (\, 0 \quad \tfrac{g}{a_2} \,) \tag{8.33}$$

$$D = (\, 0 \,) \tag{8.34}$$

Transfer function

$$G(s) = \frac{g^2 r_2}{g^2 + (g(r_2 a_2 + a_1 r_1))s + r_2 a_1 r_1 a_2 s^2} \tag{8.35}$$

8.4.4 Pressure input

This section illustrates an important feature of bond graphs: the representation does *not* depend on the causality of the input source although the resulting equations do change.

The system is identical to that of Section 8.4.2 except that

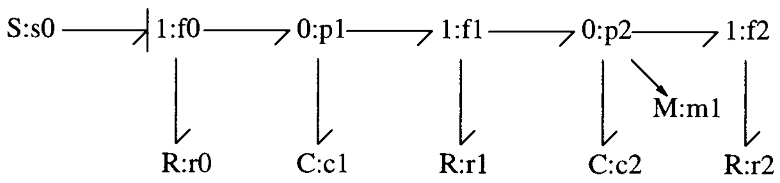


Figure 8.17 Coupled tanks (pressure input): bond graph

1. The left-hand on flow is driven by a *pressure* source.

The bond graph appears in Figure 8.10. The inflow is now imposed by the input flow resistance.

System differential-algebraic equations

$$x = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}; \; y = (\, p_2 \,); \; u = (\, f_0 \,) \tag{8.36}$$

$$\begin{aligned} \dot{x}_1 &= \frac{(-(((r_0 + r_1)a_2 x_1 - r_0 a_1 x_2)g - a_2 r_1 a_1 u_1))}{(r_0 a_2 r_1 a_1)} \\ \dot{x}_2 &= \frac{(-((r_1 + r_2)a_1 x_2 - a_2 r_2 x_1)g)}{(a_2 r_1 a_1 r_2)} \end{aligned} \tag{8.37}$$

$$y_1 = \frac{(g x_2)}{a_2} \tag{8.38}$$

Descriptor matrices of the linearised system

$$A = \begin{pmatrix} \frac{(-(r_0+r_1)g)}{(r_0 r_1 a_1)} & \frac{g}{(r_1 a_2)} \\ \frac{g}{(r_1 a_1)} & \frac{(-(r_1+r_2)g)}{(r_1 a_2 r_2)} \end{pmatrix} \tag{8.39}$$

$$B = \begin{pmatrix} \frac{1}{r_0} \\ 0 \end{pmatrix} \tag{8.40}$$

$$C = (\, 0 \quad \tfrac{g}{a_2} \,) \tag{8.41}$$

$$D = (\, 0 \,) \tag{8.42}$$

Transfer function

$$G(s) = \frac{g^2 r_2}{(g^2(r_2 + r_0 + r_1)) + (g(r_2 r_0 a_1 + r_2 r_0 a_2 + r_2 r_1 a_2 + r_0 r_1 a_1))s + r_2 r_0 r_1 a_1 a_2 s^2} \tag{8.43}$$

The transfer function, though still second order, now depends on r_0.

8.4.5 Non-linear flow resistance

The bond graph reflects structure, not properties. This is illustrated in this section by replacing the CR of the resistors in the example of Section 8.4.2 by a non-linear relationship; the bond graph is unchanged but the definition of flow resistance in the CR file changes.

CR

Equation 8.17 is replaced by

$$\Delta p_i = r_i f_i^2 \tag{8.44}$$

System differential-algebraic equations

$$x = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}; \; y = (p_2); \; u = (f_0) \tag{8.45}$$

$$\begin{aligned} \dot{x}_1 &= \frac{(-(\sqrt{((a_2x_1 - a_1x_2)g)} - \sqrt{(r_1)}\sqrt{(a_1)}\sqrt{(a_2)}u_1))}{(\sqrt{(r_1)}\sqrt{(a_1)}\sqrt{(a_2)})} \\ \dot{x}_2 &= \frac{(\sqrt{(r_2)}\sqrt{((a_2x_1 - a_1x_2)g)} - \sqrt{(x_2)}\sqrt{(r_1)}\sqrt{(a_1)}\sqrt{(g)})}{(\sqrt{(r_2)}\sqrt{(r_1)}\sqrt{(a_1)}\sqrt{(a_2)})} \end{aligned} \tag{8.46}$$

$$y_1 = \frac{(gx_2)}{a_2} \tag{8.47}$$

The states are unchanged, but the equations contain square roots due to the CR (Equation 8.44) being used with flow as output.

Steady-state properties

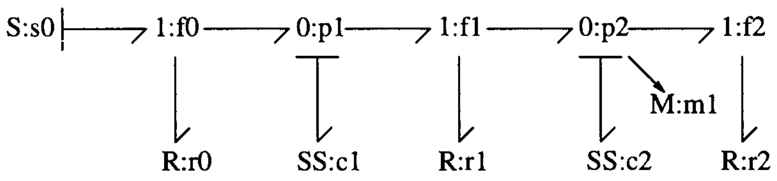


Figure 8.18 Non-linear coupled tanks - steady-state: bond graph

As discussed in Section 5.6, steady-state properties can be deduced by replacing storage components (in this case **C** components) by appropriate **SS** components.

The corresponding bond graph appears in Figure 8.14. The corresponding steady-states (with constant flow f_0) are

$$x_0 = \begin{pmatrix} \frac{(a_1 f_0^2 (r_1+r_2))}{g} \\ \frac{(a_2 r_2 f_0^2)}{g} \end{pmatrix} \tag{8.48}$$

$$u_0 = (\, f_0 \,) \tag{8.49}$$

$$y_0 = (\, r_2 f_0^2 \,) \tag{8.50}$$

Linearised system state matrices

$$A = \begin{pmatrix} \frac{(-g)}{(2a_1 r_1 f_0)} & \frac{g}{(2r_1 f_0 a_2)} \\ \frac{g}{(2a_1 r_1 f_0)} & \frac{(-(r_1+r_2)g)}{(2r_1 f_0 a_2 r_2)} \end{pmatrix} \tag{8.51}$$

$$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{8.52}$$

$$C = (\, 0 \quad \frac{g}{a_2} \,) \tag{8.53}$$

$$D = (\, 0 \,) \tag{8.54}$$

The non-linear state equations can be linearised about a steady state. In this case, the steady-state corresponds to a constant inflow of f_0 giving the same flow through each resistor. Differentiating the CR gives an equivalent resistance of

$$\frac{\partial \Delta p_i}{\partial f_i} = 2 f_0 r_i \tag{8.55}$$

This is reflected in the A matrix of the linearised system.

Linearised system transfer function

$$G(s) = \frac{2g^2 r_2 f_0}{g^2 + (2g f_0 (a_1 r_1 + a_1 r_2 + a_2 r_2))s + 4 a_1 a_2 r_1 r_2 f_0^2 s^2} \tag{8.56}$$

This is the same as that of Equation 8.25 except that each resistance is multiplied by $2f_0$.

System properties

Figure 8.19 shows the steady-state pressure output as a function of f_0. Figure 8.20 shows the linearised system time constants as a function of f_0. The system responses to a 10% change in flow are shown in Figure 8.21 for $f_0 = 0.1, 1, 10$. The corresponding frequency responses appear in Figure 8.22.

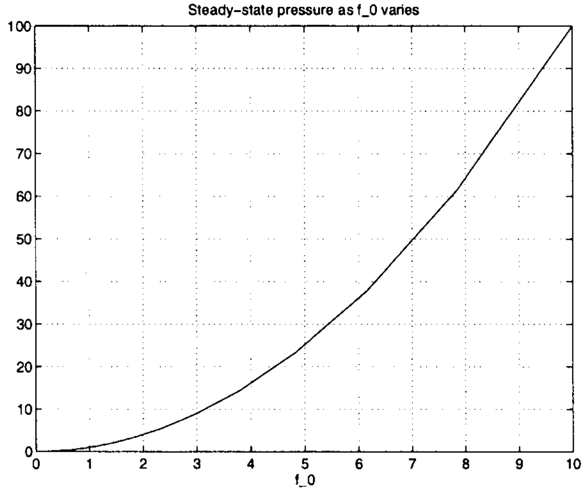


Figure 8.19 Coupled tanks (non-linear): steady-state pressure

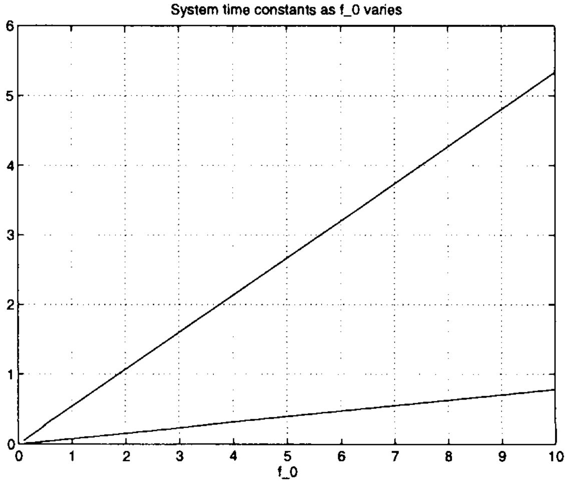


Figure 8.20 Coupled tanks (non-linear): time constants

8.4.6 Approximate system: zero resistance

As indicated in Section 8.4.2, a small valve of the inter-tank resistance r_1 leads to two widely different time constants. Such a system is said to be 'stiff' and is difficult to simulate numerically.

From the modelling point of view, it is then of interest to investigate the effect of removing the resistance between the tanks. As discussed in Chapter 5, a zero resistance must have effort output - this requirement is imposed by the appropriate causal stroke

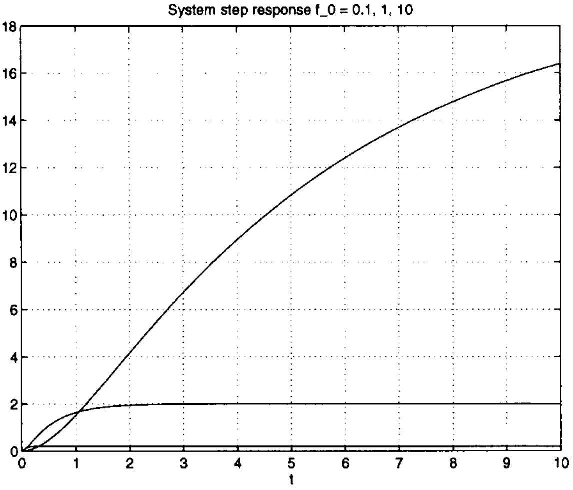


Figure 8.21 Coupled tanks (non-linear): step response

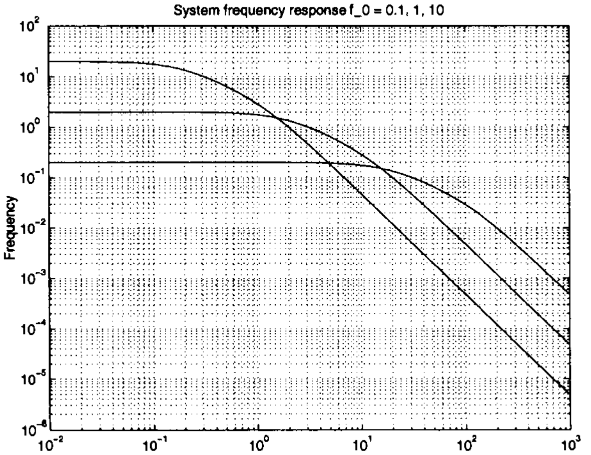


Figure 8.22 Coupled tanks (non-linear): frequency response

in Figure 8.23. The effect of removing the inter-tank resistance is that the causality of either the left or right tank C element must be reversed; Figure 8.23 constrains the left-hand tank to have integral causality by the appropriate causal stroke.

Essentially, removing the resistance constrains the two tank pressures, and hence states, to be algebraically related.

$$p_1 = p_2 \tag{8.57}$$

$$\frac{m_1}{a_1} = \frac{m_2}{a_2} \tag{8.58}$$

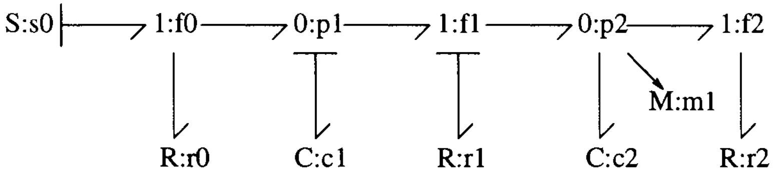


Figure 8.23 Coupled tanks (zero resistance): bond graph

System differential-algebraic equations

$$x = (\, m_1 \,);\; z = (\, m_2 \,);\; y = (\, p_2 \,);\; u = (\, f_0 \,) \tag{8.59}$$

$$\dot{x}_1 = \frac{((u_1 - \dot{z}_1)a_1 r_2 - g x_1)}{(a_1 r_2)} \tag{8.60}$$

$$z_1 = \frac{(a_2 x_1)}{a_1} \tag{8.61}$$

$$y_1 = \frac{(g x_1)}{a_1} \tag{8.62}$$

Descriptor matrices of the linearised system

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{8.63}$$

$$A = \begin{pmatrix} \frac{(-g)}{(a_1 r_2)} & 0 & -1 \\ 0 & 0 & 1 \\ \frac{a_2}{a_1} & -1 & 0 \end{pmatrix} \tag{8.64}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{8.65}$$

$$C = (\, \frac{g}{a_1} \;\; 0 \;\; 0 \,) \tag{8.66}$$

$$D = (\, 0 \,) \tag{8.67}$$

The result of removing the intertank resistance is to make one state into a non-state - one possibility is given in Equation 8.59.

Constrained-state equations

Constrained-state equations are considered in Chapter 4, Section 4.10. In this case the constrained-state equations are

$$\dot{\chi}_1 = \frac{(-(gx_1 - a_1 r_2 u_1))}{(a_1 r_2)} \tag{8.68}$$

$$y_1 = \frac{(gx_1)}{a_1} \tag{8.69}$$

$$E = \left(\frac{(a_1+a_2)}{a_1} \right) \tag{8.70}$$

The E matrix weights the state in proportion to the total area divided by the area of the tank corresponding to the system state. As E is non-singular, the system can also be rewritten as the ODE of Equation 4.26

$$\dot{x}_1 = \frac{(-(gx_1 - a_1 r_2 u_1))}{((a_1 + a_2) r_2)} \tag{8.71}$$

$$y_1 = \frac{(gx_1)}{a_1} \tag{8.72}$$

Transfer function

$$G(s) = \frac{g r_2}{g + (r_2(a_1 + a_2))s} \tag{8.73}$$

The transfer function has been reduced to that of a *first*-order system with time constant corresponding to *one* tank with area $a_1 + a_2$.

System properties

Figure 8.24 compares the step response if the reduced system with that of the original system with $r_1 = 0.1$. Figure 8.25 gives the corresponding frequency responses.

8.4.7 Approximate system: zero capacity

In contrast to Section 8.4.6, the coupled tanks are approximated by removing a capacity rather than a resistance. Thus the second tank is assumed to be small and the corresponding bond is removed from the diagram. Following Section 8.3, the capacity of the second tank is replaced by an effort pressure source.

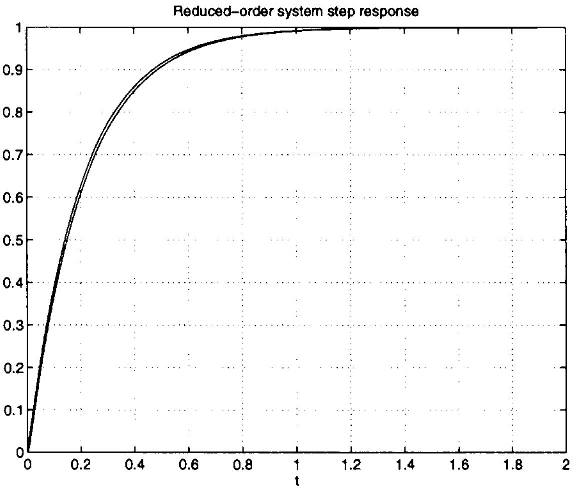


Figure 8.24 Coupled tanks (zero resistance): step response

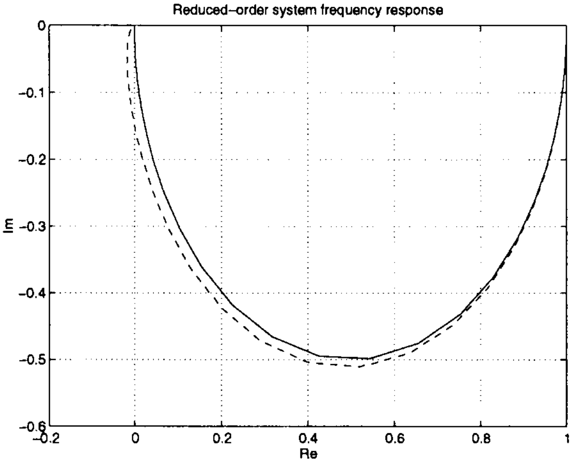


Figure 8.25 Coupled tanks (zero resistance): frequency response

System differential-algebraic equations

$$x = (m_1);\ w = (m_2);\ y = (p_2);\ u = (f_0) \tag{8.74}$$

$$\dot{x}_1 = \frac{(-(gx_1 - v_1 a_1 - a_1 r_1 u_1))}{(a_1 r_1)} \tag{8.75}$$

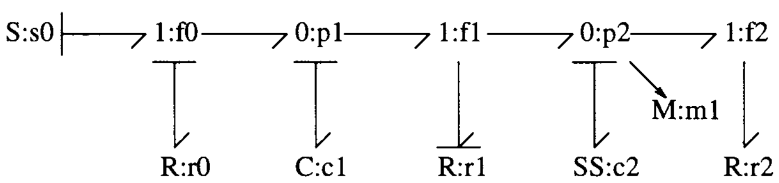


Figure 8.26 Coupled tanks (zero capacity): bond graph

$$0 = \frac{(-((r_1 + r_2)v_1 a_1 - g r_2 x_1))}{(a_1 r_1 r_2)} \tag{8.76}$$

$$y_1 = v_1 \tag{8.77}$$

Descriptor matrices of the linearised system

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{8.78}$$

$$A = \begin{pmatrix} \frac{(-g)}{(a_1 r_1)} & \frac{1}{r_1} \\ \frac{g}{(a_1 r_1)} & \frac{(-(r_1 + r_2))}{(r_1 r_2)} \end{pmatrix} \tag{8.79}$$

$$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{8.80}$$

$$C = \begin{pmatrix} 0 & 1 \end{pmatrix} \tag{8.81}$$

$$D = \begin{pmatrix} 0 \end{pmatrix} \tag{8.82}$$

Transfer function

$$G(s) = \frac{g r_2}{g + (a_1(r_2 + r_1))s} \tag{8.83}$$

The reduced system is thus first order with a time constant corresponding to the first tank capacity and the *sum* of the two flow resistances r_1 and r_2

8.5 EXAMPLE: TWO STIRRED-TANK HEATERS

8.5.1 Description

This example corresponds to that in Section 8.4.2 but with the addition of thermal effects. With reference to Figure 8.27, the first tank is heated by an element generating qW. The inlet temperature is t_0, and the specific heat of the liquid is taken as c_p Jkg^{-1}.

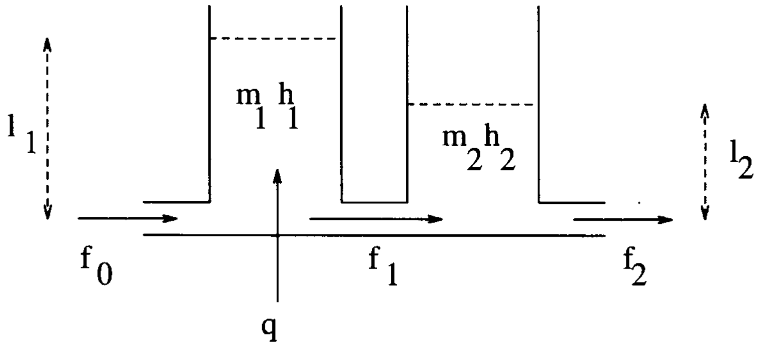


Figure 8.27 Stirred-tank heaters: schematic

Two tanks (Figure 8.27) of uniform cross-sections a_1 and a_2 contain masses m_1 and m_2 of water at temperatures t_1, and t_2. The manipulated inputs are the *mass* inflow f_0 and heat input q; the inflow temperature t_0 is a disturbance. The levels l_1 and l_2 in each tank and the temperatures t_1 and t_2 are the outputs. The state vector corresponds to the mass and enthalpy holdups:

$$X = \begin{pmatrix} m_1 \\ m_2 \\ h_1 \\ h_2 \end{pmatrix} \tag{8.84}$$

where m_i is the mass and h_i is the enthalpy stored in tank i. The input vector is

$$u = \begin{pmatrix} f_0 \\ t_0 \\ q \end{pmatrix} \tag{8.85}$$

and the output vector is

$$y = \begin{pmatrix} l_2 \\ t_2 \end{pmatrix} \tag{8.86}$$

The flow rates f_i are given by

$$f_i = k_i \Delta p_i \tag{8.87}$$

where Δp_i is the pressure across each pipe and f_i is the mass flow rate. The dotted signals indicate that the thermal resistors are modulated by mass flow rates and the thermal capacities by tank masses (states).

Assumptions

1. The flow is one way (left to right)
2. There is no heat loss from the tank.

The bond graph in Figure 8.28 represents this system. The bond graph junctions and components have been labelled as in Table 8.3.

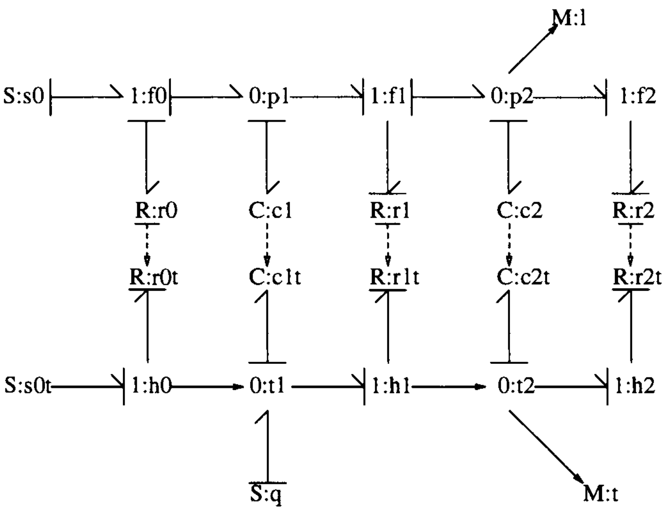


Figure 8.28 Stirred-tank heaters: bond graph

Label	Physical equivalent
f0	Common flow of mass: from source s0, through pipe, into tank
f1	Common flow of mass: to source s1, through pipe, from tank
h0	Common flow of enthalpy: from source s0, through pipe, into tank
h1	Common flow of enthalpy: to source s1, through pipe, from tank
p1	Common pressure at: end of input pipe, base of tank end of output pipe
t1	Common temperature at: end of input pipe, within tank, end of output pipe
s0	flow source: input *mass* flow rate f_i
s1	flow source: output *mass* flow rate f
s0t	effort source: input temperature T_i
q	flow source: input input heat flow q
r0, r0t	Flow and thermal resistances: input pipe
r1, r1t	Flow and thermal resistances: output pipe
c1, c1t	Flow and thermal capacities: tank
l	measurement: level in tank
T	measurement: temperature of liquid in tank

Table 8.3 Bond graph notation

8.5.2 The system equations

The bond graph of Figure 8.28 is causally complete and therefore leads to a set of ordinary differential equations. These are

$$\dot{x}_1 = \frac{(-((a_2x_1 - a_1x_2)gk_1 - a_2a_1u_1))}{(a_2a_1)}$$

$$\begin{aligned}
\dot{x}_2 &= \frac{(-((k_1+k_2)a_1x_2 - a_2k_1x_1)g)}{(a_2a_1)} \\
\dot{x}_3 &= \frac{(-((a_2x_1 - a_1x_2)gk_1x_3 - (c_pu_1u_2 + u_3)a_2a_1x_1))}{(a_2a_1x_1)} \\
\dot{x}_4 &= \frac{(((a_2k_1x_3 - a_1k_2x_4)x_1 - k_1a_1x_2x_3)g)}{(a_2a_1x_1)}
\end{aligned} \tag{8.88}$$

$$\begin{aligned}
y_1 &= \frac{x_2}{(a_2\rho)} \\
y_2 &= \frac{x_4}{(c_px_2)}
\end{aligned} \tag{8.89}$$

where

$$x = \begin{pmatrix} m_1 \\ m_2 \\ h_1 \\ h_2 \end{pmatrix}; \; y = \begin{pmatrix} l_2 \\ t_2 \end{pmatrix}; \; u = \begin{pmatrix} f_i \\ t_i \\ q \end{pmatrix} \tag{8.90}$$

Notice that the first two of equations 8.88 (representing hydraulics) are linear, whereas the second two of equations 8.88 (representing thermal effects) are *non-linear.*

8.5.3 Linearised system

Following the procedures in Section 4.12, the non-linear system of equations 8.88 can be linearised about the steady state given by

$$x_0 = \begin{pmatrix} \frac{(a_1f_0(k_1+k_2))}{g} \\ \frac{(a_2k_2f_0)}{g} \\ \frac{(a_1(k_1c_pf_0t_0+k_1q_0+k_2c_pf_0t_0+k_2q_0))}{g} \\ \frac{(a_2k_2(c_pf_0t_0+q_0))}{g} \end{pmatrix} \tag{8.91}$$

$$u_0 = \begin{pmatrix} f_0 \\ t_0 \\ q_0 \end{pmatrix} \tag{8.92}$$

$$y_0 = \begin{pmatrix} \frac{(k_2f_0)}{(g\rho)} \\ \frac{(c_pf_0t_0+q_0)}{(c_pf_0)} \end{pmatrix} \tag{8.93}$$

to give the descriptor (in this case state-space) matrices.

$$A = \begin{pmatrix} \frac{(-gk_1)}{a_1} & \frac{(gk_1)}{a_2} & 0 & 0 \\ \frac{(gk_1)}{a_1} & \frac{(-(k_1+k_2)g)}{a_2} & 0 & 0 \\ \frac{(-(c_pf_0t_0+q_0)gk_1k_2)}{((k_1+k_2)a_1f_0)} & \frac{((c_pf_0t_0+q_0)gk_1)}{(a_2f_0)} & \frac{(-gk_1^2)}{((k_1+k_2)a_1)} & 0 \\ \frac{((c_pf_0t_0+q_0)gk_1k_2)}{((k_1+k_2)a_1f_0)} & \frac{(-(c_pf_0t_0+q_0)gk_1)}{(a_2f_0)} & \frac{(gk_1^2)}{((k_1+k_2)a_1)} & \frac{(-gk_2)}{a_2} \end{pmatrix} \tag{8.94}$$

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ c_p t_0 & c_p f_0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{8.95}$$

$$C = \begin{pmatrix} 0 & \frac{1}{(a_2 \rho)} & 0 & 0 \\ 0 & \frac{(-(c_p f_0 t_0 + q_0) g)}{(a_2 k_2 c_p f_0^2)} & 0 & \frac{g}{(a_2 k_2 c_p f_0)} \end{pmatrix} \tag{8.96}$$

$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{8.97}$$

8.5.4 Simulation

This system can be simulated using the following numerical values expressed in suitably scaled units)

$$\begin{aligned} a_1 &= 5 \\ a_2 &= 10 \\ k_i &= 1 \\ \rho &= 1 \\ c_p &= 4.180 \\ g &= 9.81 \end{aligned} \tag{8.98}$$

The simulation diagram is:

The non-linear system was simulated with initial state

$$X = \begin{pmatrix} m_1 \\ m_2 \\ h_1 \\ h_2 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0 \\ 0 \end{pmatrix} \tag{8.99}$$

The system input was constant

$$u = \begin{pmatrix} f_0 \\ t_0 \\ q \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 100 \end{pmatrix} \tag{8.100}$$

The linearised version was simulated at the same time and with the same inputs. The system was linearised with respect to

$$u = \begin{pmatrix} f_0 \\ t_0 \\ q \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix} \tag{8.101}$$

Note that the system is set up in such a way that the linearised system (in state-space form) operates on *deviation* variables.

The system inputs where varied in a stepwise fashion as shown in Figure 8.30. The values of the input were

$$u = \begin{pmatrix} f_0 \\ t_0 \\ q \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix} \tag{8.102}$$

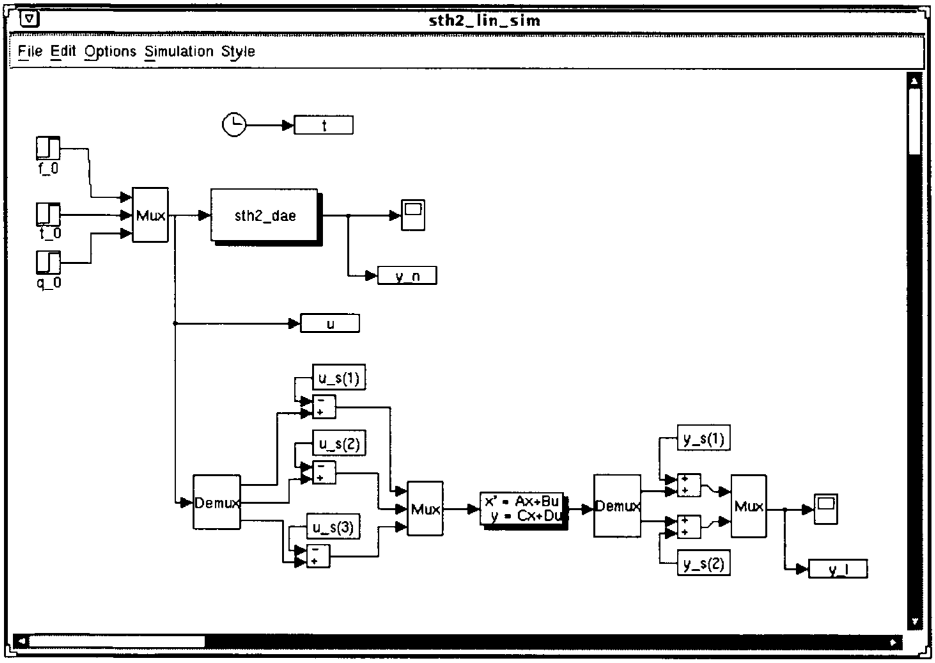


Figure 8.29 Two stirred-tank heaters: non-linear simulation diagram

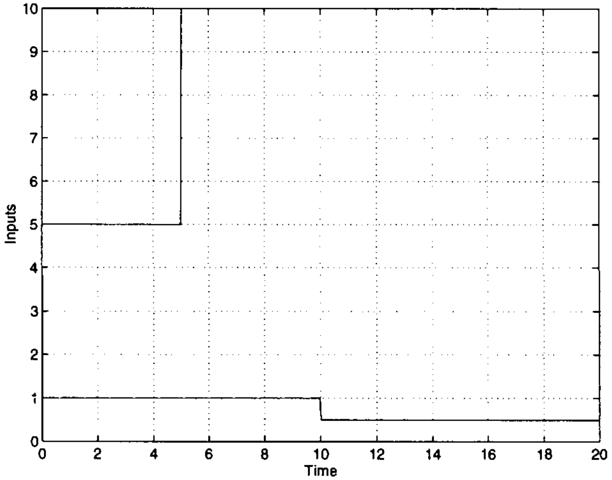


Figure 8.30 Two stirred-tank heaters: non-linear simulation - inputs

from time 0 to time 5;

$$u = \begin{pmatrix} f_0 \\ t_0 \\ q \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 10 \end{pmatrix} \tag{8.103}$$

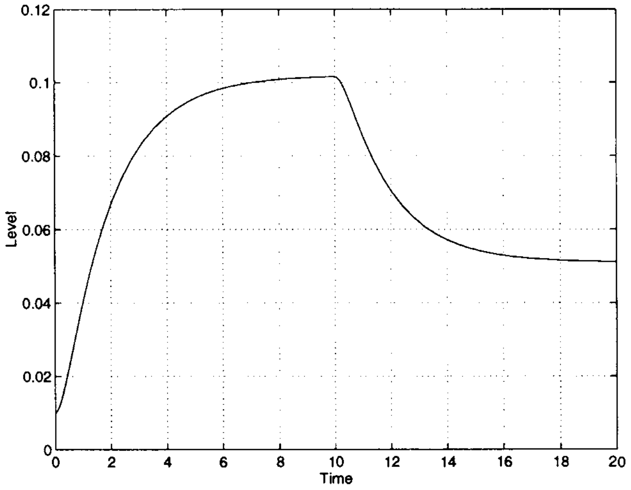


Figure 8.31 Two stirred-tank heaters: non-linear simulation - level

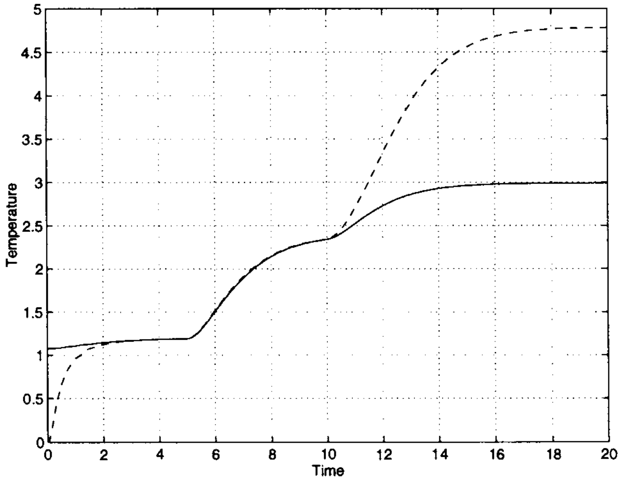


Figure 8.32 Two stirred-tank heaters: non-linear simulation - temperature

from time 5 to time 10; and

$$u = \begin{pmatrix} f_0 \\ t_0 \\ q \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0 \\ 10 \end{pmatrix} \tag{8.104}$$

thereafter.

The resulting responses of the two outputs

$$y = \begin{pmatrix} l_2 \\ t_2 \end{pmatrix} \tag{8.105}$$

appear in Figures 8.31 and 8.32.

In each case, the dashed line represents the non-linear result and the firm line the linearised response. The pressure response to flow is, of course linear, but the corresponding temperature response in non-linear. The non-linear response is a good approximation when the system is close to the steady-state corresponding to the linearised model; the approximation is poor away from these steady-states.

8.5.5 The approximate system

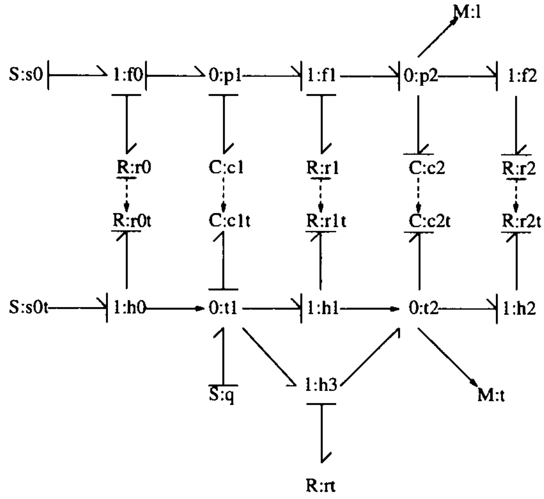


Figure 8.33 The approximate system: bond graph

As in Section 8.4.6, the system is approximated by one where the inter-tank *flow* resistance is zero. In Figure 8.33 this is accomplished by imposing effort causality on the inter-tank resistance (and setting the corresponding resistance to zero). This constrains the two mass states to be equal (as $a_1 = a_2$). Again, Figure 8.33 ensures that the state associated with the first tank has integral causality.

In addition, the two thermal states are constrained to be equal by including the additional resistance r_t into Figure 8.33. In the previous section, effectively $r_t = \infty$, in this section $r_t = 0$.

System differential-algebraic equations

$$x = \begin{pmatrix} m_1 \\ h_1 \end{pmatrix} ; \; z = \begin{pmatrix} m_2 \\ h_2 \end{pmatrix} ; \; y = \begin{pmatrix} l_2 \\ t_2 \end{pmatrix} ; \; u = \begin{pmatrix} f_i \\ t_i \\ q \end{pmatrix} \tag{8.106}$$

$$\dot{x}_1 = \frac{((u_1 - \dot{z}_1)a_1 - gk_2x_1)}{a_1}$$

$$\dot{x}_2 = \frac{((c_p u_1 u_2 + u_3 - \dot{z}_2)a_1 - g k_2 x_2)}{a_1} \tag{8.107}$$

$$\begin{aligned} z_1 &= \frac{(a_2 x_1)}{a_1} \\ z_2 &= \frac{(a_2 x_2)}{a_1} \end{aligned} \tag{8.108}$$

$$\begin{aligned} y_1 &= \frac{x_1}{(a_1 \rho)} \\ y_2 &= \frac{x_2}{(c_p x_1)} \end{aligned} \tag{8.109}$$

System constrained-state equations

As in Section 4.10, the system equations can be rewritten in constrained-state form as

$$\dot{\chi}_1 = \frac{(-(g k_2 x_1 - a_1 u_1))}{a_1} \tag{8.110}$$

$$\dot{\chi}_2 = \frac{((c_p u_1 u_2 + u_3)a_1 - g k_2 x_2)}{a_1} \tag{8.111}$$

$$y_1 = \frac{x_1}{(a_1 \rho)} \tag{8.112}$$

$$y_2 = \frac{x_2}{(c_p x_1)} \tag{8.113}$$

$$E = \begin{pmatrix} \frac{(a_1+a_2)}{a_1} & 0 \\ 0 & \frac{(a_1+a_2)}{a_1} \end{pmatrix} \tag{8.114}$$

As E is non-singular, the system can also be rewritten as the ODE

$$\dot{x}_1 = \frac{(-(g k_2 x_1 - a_1 u_1))}{(a_1 + a_2)} \tag{8.115}$$

$$\dot{x}_2 = \frac{((c_p u_1 u_2 + u_3)a_1 - g k_2 x_2)}{(a_1 + a_2)} \tag{8.116}$$

$$y_1 = \frac{x_1}{(a_1 \rho)} \tag{8.117}$$

$$y_2 = \frac{x_2}{(c_p x_1)} \tag{8.118}$$

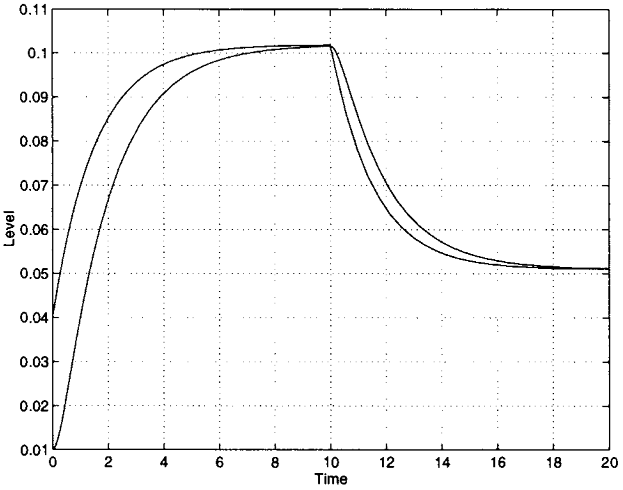


Figure 8.34 Two stirred-tank heaters: approximate simulation - level

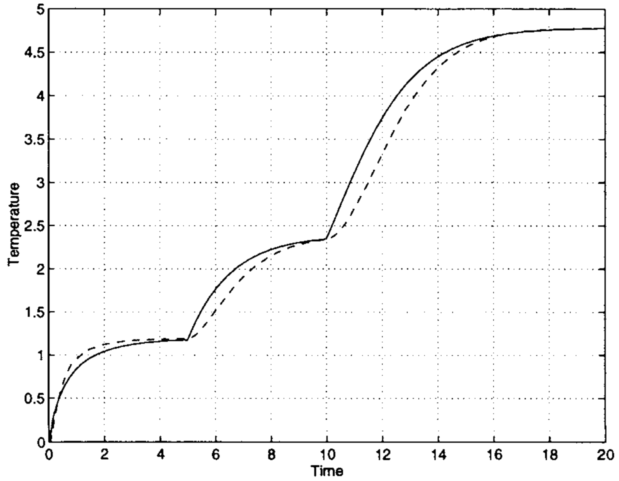


Figure 8.35 Two stirred-tank heaters: approximate simulation - temperature

8.5.6 Simulation

This system (in constrained-state form) can be simulated as in Section 8.5.4.

The approximate system was simulated with initial state

$$X = \begin{pmatrix} m_1 + m_2 \\ h_1 + h_2 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix} \tag{8.119}$$

and the same input as before.

The resulting response of the two outputs

$$y = \begin{pmatrix} l_2 \\ t_2 \end{pmatrix} \tag{8.120}$$

appear in Figures 8.34 and 8.35.

In each case, the dashed line represents the exact system and the firm line the approximate system.

8.6 EXAMPLE: LIQUID-LIQUID EXTRACTION

8.6.1 Description

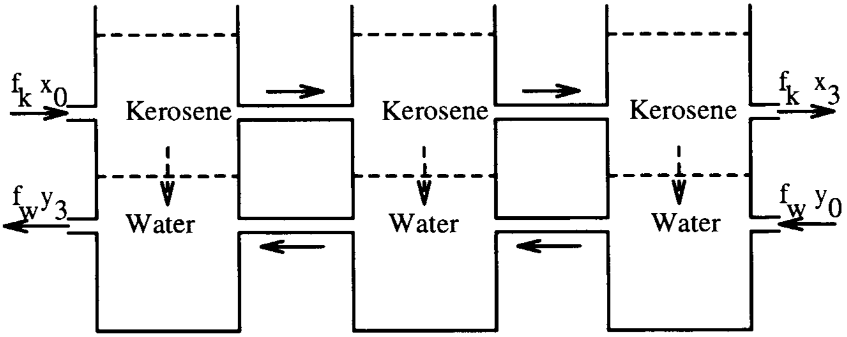


Figure 8.36 Liquid-liquid extraction: schematic

Figure 8.36 is a highly stylised representation of a three-stage liquid-liquid extraction process. Each of the three tanks contains the immiscible liquids water and kerosene, and each of these liquids has the same solute dissolved in it. The purpose of the process is to remove the solute from the kerosene and hence add it to the water. Kerosene flows at a constant rate f_k from left to right, and water flows at a constant rate f_w from right to left; it is assumed that the mass of water and kerosene in each tank does *not* vary with time. The input and output concentrations in the solute in kerosene are x_0 and x_3 respectively; the input and output concentrations of the solute in water are y_0 and y_3 respectively.

The solute diffuses from the kerosene to the water in each tank so that $x_0 > x_3$ and $y_3 > y_0$.

Assumptions

1. The holdups of water and kerosene are constant.

2. The solute diffuses from kerosene to water at a rate proportional to the concentration difference $f_{kwi} = \frac{k_i - w_i}{r_1}$

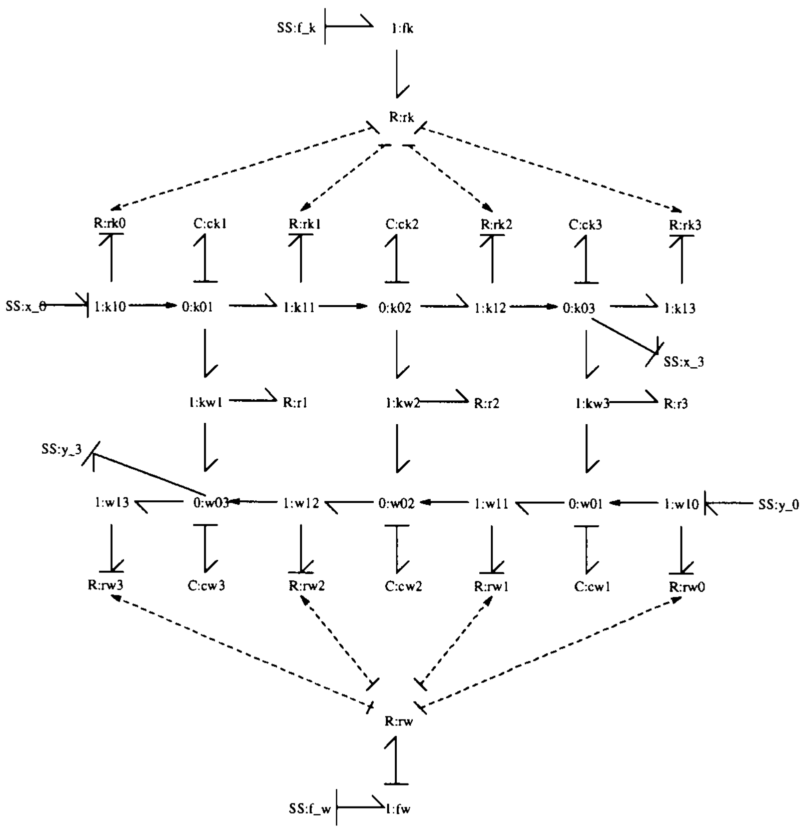


Figure 8.37 Liquid-liquid extraction: bond graph

The bond graph in Figure 8.37 represents this system. The bond graph junctions and components have been labelled as in Table 8.4. The flow of solute between the tanks is modelled as in Section 8.2.3; there is no need for hydraulic modelling as the water and kerosene holdups are assumed constant. The flow of solute between the water and kerosene is modelled using the usual R element.

8.6.2 The system equations

The bond graph of Figure 8.37 is causally complete and therefore leads to a set of ordinary differential equations. These are

$$\begin{aligned}
\dot{x}_1 &= \frac{((c_k u_2 - x_1) r r_k c_w u_1 - c_w x_1 + c_k x_6)}{(r c_w c_k)} \\
\dot{x}_2 &= \frac{((x_1 - x_2) r r_k c_w u_1 - c_w x_2 + c_k x_5)}{(r c_w c_k)}
\end{aligned}$$

Label	Physical equivalent
f_k	Kerosene flow
fk	Common flow of kerosene though system
x_0	Input concentration of solute dissolved in kerosene
rk0 , ... , rk3	R modulated by f_k
k10 , ... , k13	Flow of solute dissolved in kerosene
ck1 , ... , ck3	Holdup of solute dissolved in kerosene $k_1, ..., k_3$
k01 , ... , k03	Concentration of solute dissolved in kerosene
x_3	Output concentration of solute dissolved in kerosene
kw1 , ... , kw3	Flow of solute from kerosene to water
r1 , ... , r3	Resistance to flow of solute from kerosene to water
f_w	Water flow
wk	Common flow of water though system
y_0	Input concentration of solute dissolved in water
rw0 , ... , rw3	R modulated by f_w
ww10 , ... , k13	Flow of solute dissolved in water
cw1 , ... , cw3	Holdup of solute dissolved in water $w_1, ..., w_3$
w01 , ... , w03	Concentration of solute dissolved in water
y_3	Output concentration of solute dissolved in water

Table 8.4 Bond graph notation

$$\begin{aligned}
\dot{x}_3 &= \frac{((x_2 - x_3)rr_kc_wu_1 - c_wx_3 + c_kx_4)}{(rc_wc_k)} \\
\dot{x}_4 &= \frac{((c_wu_4 - x_4)rc_kr_wu_3 + c_wx_3 - c_kx_4)}{(rc_wc_k)} \\
\dot{x}_5 &= \frac{((x_4 - x_5)rc_kr_wu_3 + c_wx_2 - c_kx_5)}{(rc_wc_k)} \\
\dot{x}_6 &= \frac{((x_5 - x_6)rc_kr_wu_3 + c_wx_1 - c_kx_6)}{(rc_wc_k)}
\end{aligned} \tag{8.121}$$

$$\begin{aligned}
y_1 &= \frac{x_3}{c_k} \\
y_2 &= \frac{x_6}{c_w}
\end{aligned} \tag{8.122}$$

where

$$x = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix}; \; y = \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}; \; u = \begin{pmatrix} f_k \\ x_0 \\ f_w \\ y_0 \end{pmatrix} \tag{8.123}$$

Notice that equations 8.121 are non-linear.

9

Pharmacokinetics

SUMMARY

- The application of bond graphs to the modelling of human anaesthetic drug uptake is discussed.
- The advantages of automatic model generation in this context are illustrated.

9.1 INTRODUCTION

Models for the uptake of anaesthetic drugs, based on physiological data, were developed some 30 years ago by Mapleson (1964) and further developed Mapleson (1973) and Davis and Mapleson (1981). The strength of these models is that they are *not* empirical, but rather they are built on quantitative physiological information (for example the "standard man" (Mapleson, 1964)). They thus not only lead to accurate computer simulations but also enhance understanding and explanation in terms of the underlying quantitative physiological processes. The first version of Mapleson's model was based on a passive electrical analogue, but later versions were based on computer simulation code.

This chapter shows that bond graphs provide an alternative, and powerful, modelling technique. In particular, the modelling procedures discussed in Chapter 8 are appropriate here. For brevity, attention is focussed on inhaled, rather than injected, drugs.

A wider discussion of the use of bond graphs in the life sciences is given by LeFèvre (1995).

9.2 COMPONENT MODELS

9.2.1 Variables

As in process engineering (Chapter 8), *pseudo* bond graphs are used in this chapter; thus the variables used by the specialists in the field can be freely used.

As discussed by Mapleson (Mapleson, 1964) the *tension* of a gas dissolved in a liquid is defined as the *partial pressure* of that gas within a gas in *equilibrium* with the liquid.

Effort	*Units*	Flow	*Units*
Tension	atm	mass flow rate	m^3s^{-1}

Table 9.1 Effort and flow variables in pharmacokinetics

As he says '... there are certain advantages in working in terms of ... tension rather than concentration, because all tensions in the blood and tissues all tend toward the same value – the tension in the inspired tissue.' For these reasons, it is natural to use *tension* as the *effort variable.* Following (Mapleson, 1973) an appropriate unit is atmospheres - i.e. pressure expressed as a percentage of atmospheric pressure. Tension may then be re-expressed in any other units by multiplying by atmospheric pressure expressed in those units.

The concept of partial pressure is based on the approximation that the properties of one gas do not change when mixed with a second gas. In particular, the mass M of a gas with density (at a given temperature and pressure) ρ within a container of volume V and at a tension (partial pressure) T is

$$M = \rho VT \tag{9.1}$$

This approximation no longer holds when a gas is dissolved in a liquid. This leads to the notion of the *partition coefficient* λ between a gas and a liquid defined as 'the ratio of the concentrations in the two phases when they are in equilibrium' (that is at the same tension) (Mapleson, 1964). Thus Equation 9.1 becomes

$$M = \lambda\rho VT \tag{9.2}$$

It is convenient to define a normalised mass m as the ratio of the mass M to that of a unit volume of the gas (at the corresponding temperature and pressure)

$$m = \frac{M}{\rho} \tag{9.3}$$

Equation 9.2 then becomes

$$m = \lambda VT \tag{9.4}$$

As in in Chapter 8, the appropriate *flow* variable is mass flow. In this context, the inhaled gas dissolves in the blood and is then transported to the various body organs. If the volumetric blood flow is Q, the transported flow Q_g of dissolved gas is

$$Q_g = \lambda VTQ \tag{9.5}$$

9.2.2 Components

The human body contains organs and tissues perfused by a flow of blood. If the blood and tissue drug tensions differ, then the drug flows between the blood and the corresponding tissue; the tissues store the drug, the blood both stores and transports the drug. This quite complicated system may be approximated by dividing the body into *pools* , each containing drug at a given tension and *arteries* and *veins* which transport the drug between pools (Mapleson, 1973).

C elements

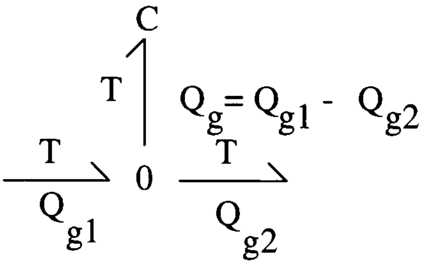


Figure 9.1 Pool model

As in Section 8.2.2, a pool is a C element storing the mass of the dissolved gas m as an integral of the net flow $Q_g = Q_{g1} - Q_{g2}$

$$m(t) = \int^t Q_g(\tau)d\tau \tag{9.6}$$

The corresponding effort variable, tension T, is then given by Equation 9.4 as

$$T = \frac{1}{c}m \tag{9.7}$$

where

$$c = \lambda V \tag{9.8}$$

The corresponding bond graph appears in Figure 9.1.

R elements

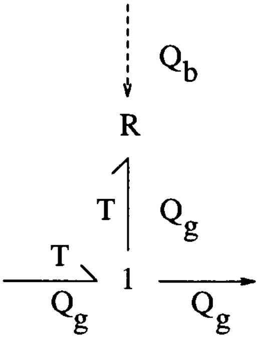


Figure 9.2 Artery/vein model

Similarly, arteries and veins can be modelled as in Section 8.2.2 as R elements where the flow Q_g is given in terms of the *upstream* tension T as

$$Q_g = \frac{1}{r}T \tag{9.9}$$

where the equivalent resistance r is given by

$$\frac{1}{r} = \lambda Q \qquad (9.0)$$

The corresponding bond graph appears in Figure 9.2; as in Section 8.2.2, the R elemen is modulated (by the blood flow Q) and the active bond renders the gas flow Q_g independent of the downstream tension.

9.3 SIMPLE PHARMACOKINETIC MODELS

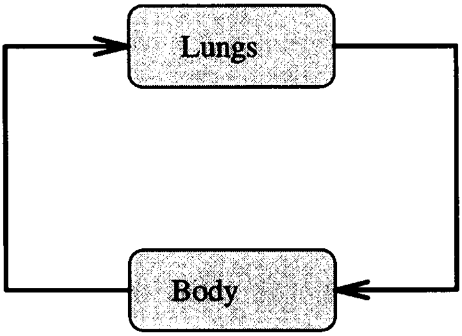


Figure 9.3 A simple pool model

As an initial illustration, consider the very simplified model of drug uptake outlinc in Figure 9.3. It is to be interpreted as follows:

- the lungs inhale and expire (but do not store) the drug;
- the arteries carry blood (containing drug at a tension; corresponding to the lungs) from the lungs to the body with flowrate Q;
- the body is composed of a single lump of tissue;
- the tension of the drug contained in the blood perfusing the body tissue is in equilibrium with that tissue, and also stores the drug ;
- the blood perfusing the body is lumped into one pool;
- blood in the vein and artery carry, but do not store the drug.

These interpretations are not obvious from Figure 9.3; in contrast the correspondng bond graph has a precise and unambiguous interpretation. Two bond graph models vill be considered:

- an *active* model containing active bonds;
- a *passive* model containing no active bonds.

9.3.1 Active model

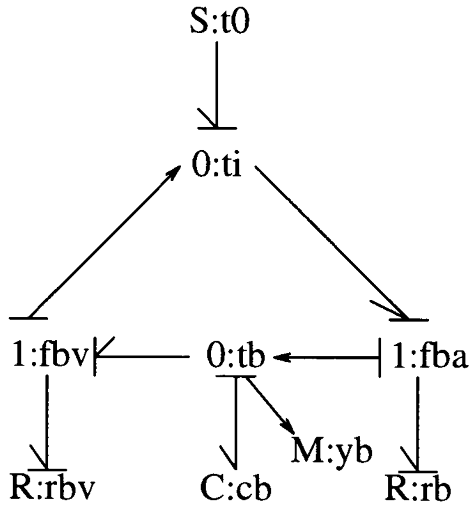


Figure 9.4 A simple pool model: bond graph

Component label	Component type	Associated physical variable
ti	Common tension junction	Lung tension T_L
tb	Common tension junction	Body tension T_B
t0	Source	Inspired tension $T_i = T_L$
cb	C	Quantity of drug stored in body
fba	Common flow junction	Arterial drug flow (lung-body)
rb	R	Arterial drug flow (lung-body)
fbv	Common flow junction	Venous drug flow (body-lung)
rbv	R	Venous drug flow (body-lung)

Table 9.2 A simple pool model: bond graph labels

The bond graph of Figure 9.4 gives a precise and unambiguous representation of Figure 9.3. The interpretation of the bond graph components is given in Table 9.2. Following Section 9.2.2, each of the two R components has an equivalent resistance (Equation 9.10)

$$r = \frac{1}{\lambda Q} \tag{9.11}$$

and the C component include the drug stored in both the perfusing blood and the tissue and thus has the equivalent capacity (Equation 9.8)

$$c = \lambda_b V_b + \lambda_t V_t \tag{9.12}$$

where λ_b and λ_t are the partition coefficients of the blood and tissue respectively and V_b and V_t are the volumes of the blood and tissue respectively.

An alternative approach to modelling would be to explicitly include separate C components for the blood and tissue pools; this would have the advantage of allowing examination of the effect of dropping the assumption of tissue/blood tension equilibrium.

This bond graphs represents a linear system; and the bond graph is causally complete and contains no C or I components with *derivative causality*. So the system has an ordinary differential equation representation (Section 4.9) on the form of Equation 4.14

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= CX + Du \end{aligned} \tag{9.13}$$

In this particular case the matrices are all scalar and given by

$$x = (\, t_i \,);\ y = (\, t_b \,);\ u = (\, t_0 \,) \tag{9.14}$$

$$A = \left(\frac{(-1)}{(c_b r_b)} \right) \tag{9.15}$$

$$B = (\, \frac{1}{r_b} \,) \tag{9.16}$$

$$C = (\, \frac{1}{c_b} \,) \tag{9.17}$$

$$D = (\, 0 \,) \tag{9.18}$$

As discussed in Section 4.13, the corresponding transfer function representation is

$$G(s) = \frac{1}{1 + c_b r_b s} \tag{9.19}$$

For this very simple case, then, the dynamic system relating inspired tension to body tension is a first-order lag with time-constant $\tau = cr_b = \frac{\lambda_b V_b + \lambda_t V_t}{\lambda_b Q}$.

9.3.2 Passive model

Component label	Component type	Associated physical variable
ti	Common tension junction	Lung tension T_L
tb	Common tension junction	Body tension T_B
t0	Source	Inspired tension $T_i = T_L$
cb	C	Quantity of drug stored in body
fb	Common flow junction	Net arterial/venous drug flow (lung-body)
rb	R	Net arterial/venous drug flow (lung-body)

Table 9.3 A simple pool model: passive bond graph labels

The bond graph of Section 9.3.1 (Figure 9.4) contains active components associated with the **R** components. However, in this simple case, an entirely passive (no active bond) model is possible. This is because the bond graph of Figure 9.4 has special properties:

- the two R elements (**rb** and **rbv**) have the same value;

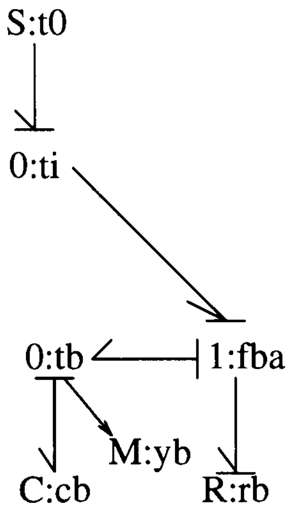


Figure 9.5 A simple pool model: passive bond graph

- the flows through the two resistors are connected between the *same* two zero junctions.

The net flow into the pool represented by **cb** is thus

$$Q_g = \frac{1}{r}(T_L - T_B) \tag{9.20}$$

The *passive* bond graph of Figure 9.5 thus has exactly the same properties as that of Figure 9.4: the flow through the R components is given by Equation 9.20.

9.4 A DETAILED PHARMACOKINETIC MODEL

A diagrammatic representation of how an inhaled drug perfuses the body tissue and organs appears in Figure 9.6. Like many domain-specific diagrams, Figure 9.6 is again ambiguous and imprecise without additional domain-specific information. It is to be interpreted as follows:

- the lungs inhale and expire the drug;
- the lungs also store the drug, and therefore have an associated (gaseous) pool with volume V;
- the arteries carry blood (containing drug at a tension corresponding to the lungs) from the lungs to the tissues with flowrate Q;
- the blood perfusing the tissues is lumped into four pools:
  - brain (volume V_b)

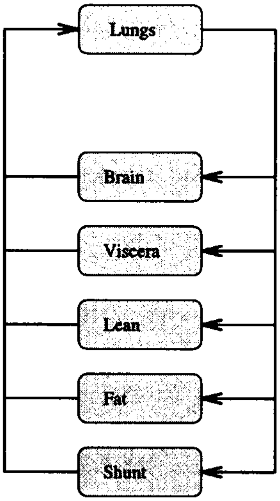


Figure 9.6 Inhaled drug perfusion

  - viscera (volume V_v)
  - lean (volume V_l)
  - fat (volume V_f).

- some blood does not perfuse any tissue; this is lumped into the pool labeled shunt with volume V_s;
- the flow of blood in the artery (and vein) associated with the five pools (four tissue and one shunt) is a fixed percentage of the blood flow in the lung artery;
- the veins carry blood (containing drug at a tension corresponding to each tissue) from the each tissue to the lungs;
- the tension of the drug contained in the blood perfusing each tissue is in equilibrium with that tissue, and also stores the drug;
- blood in the veins and arteries carry, but do not store, the drug.

The bond graph of Figure 9.7 shows a *passive* representation of the more complex system; it is an extension of the bond graph of Figure 9.5 and may be used in place of the corresponding *active* version for the same reasons as those given in Section 9.3. The components corresponding to the lung and brain are given in Table 9.4; the components corresponding to the viscera, lean, fat and shunt pools have suffix v,l,f and s respectively.

There are two main differences compared to the model of Section 9.3.2:

- the lung model includes the storage of drug in the lung tissue and
- the body is subdivided.

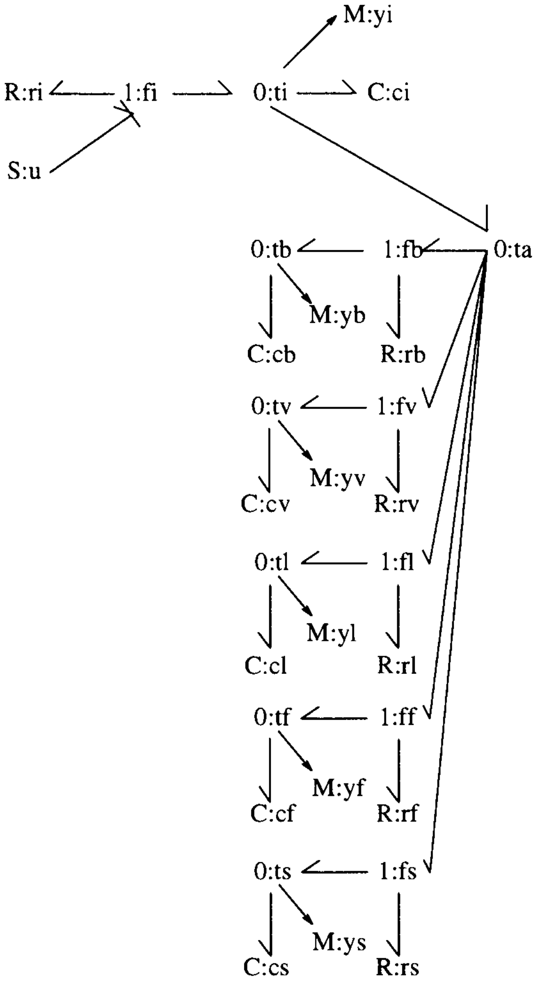


Figure 9.7 A detailed pharmacokinetic model: bond graph

As in Section 9.3.2, the system has an ordinary differential equation representation (Section 4.9) on the form of Equation 4.14

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= CX + Du \end{aligned} \tag{9.21}$$

Component label	Component type	Associated physical variable
u	Source	Inspired tension T_i
ti	Common tension junction	Lung tension T_L
fi	Common flow junction	Net drug flow into lung
ci	C	Quantity of drug stored in lung
tb	Common tension junction	Brain tension T_b
t0	Source	Inspired tension T_i
cb	C	Quantity of drug stored in brain
fb	Common flow junction	Net arterial/venous drug flow (lung-brain)
rb	R	Net arterial/venous drug flow (lung-brain)

Table 9.4 A detailed pharmacokinetic model: bond graph labels

But the state, output and input vectors are now

$$x = \begin{pmatrix} m_i \\ m_b \\ m_v \\ m_l \\ m_f \\ m_s \end{pmatrix} ; \; y = \begin{pmatrix} t_i \\ t_b \\ t_v \\ t_l \\ t_f \\ t_s \end{pmatrix} ; \; u = (\, t_0 \,) \tag{9.22}$$

and the corresponding matrices are given by

$$A = \begin{pmatrix} \frac{(-(k_b+k_v+k_l+k_f+k_s+k_i))}{c_i} & \frac{k_b}{c_b} & \frac{k_v}{c_v} & \frac{k_l}{c_l} & \frac{k_f}{c_f} & \frac{k_s}{c_s} \\ \frac{k_b}{c_i} & \frac{(-k_b)}{c_b} & 0 & 0 & 0 & 0 \\ \frac{k_v}{c_i} & 0 & \frac{(-k_v)}{c_v} & 0 & 0 & 0 \\ \frac{k_l}{c_i} & 0 & 0 & \frac{(-k_l)}{c_l} & 0 & 0 \\ \frac{k_f}{c_i} & 0 & 0 & 0 & \frac{(-k_f)}{c_f} & 0 \\ \frac{k_s}{c_i} & 0 & 0 & 0 & 0 & \frac{(-k_s)}{c_s} \end{pmatrix} \tag{9.23}$$

$$B = \begin{pmatrix} k_i \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{9.24}$$

$$C = \begin{pmatrix} \frac{1}{c_i} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{c_b} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{c_v} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{c_l} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{c_f} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{c_s} \end{pmatrix} \tag{9.25}$$

$$D = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{9.26}$$

The expressions have been simplified by substituting

$$k_x = \frac{1}{r_x} = \lambda_x \delta_x Q \tag{9.27}$$

where λ_x is the partition coefficient of the pool x, δ_x is the fraction of the blood flow perfusing tissue x and Q is the total blood flow

$$Q = \frac{V_s}{T_h} \tag{9.28}$$

where V_s is the heart stroke volume and T_h the heartbeat interval.

The capacity c_x of each pool is

$$c_x = \lambda_x V_x + \lambda_{blood} \gamma V \tag{9.29}$$

where V_x is the tissue volume and γ_x is the fraction of the total blood volume associated with each pool.

Where x is the appropriate subscript.

Pool	Volume V_x	Flow fraction δ_x	Volume fraction γ_x	Partition Coeff. λ_x
Lung	0.6	—	—	—
Brain	0.0007	0.000086	0.000055	0.46
Viscera	6.2	0.63	0.399	0.46
Lean	39.2	0.131	0.131	0.46
Fat	12.2	0.04	0.111	1.40
Shunt	0.0	0.199	0.126	0.46

Table 9.5 A detailed pharmacokinetic model: data

The total blood volume V, the heart stroke volume V_s and the heartbeat interval T_h are

$$V = 5.4 \text{ "}litres\text{"}; \; V_s = 0.108 \text{ "}litres\text{"}; \; T_h = 1 \text{ "}s\text{"}. \tag{9.30}$$

This linear system was simulated with the data in Table 9.5 corresponding to a 'standard man' (Mapleson, 1964) breathing 75% N_2O at atmospheric pressure of 760 mmHg for two minutes and air for the rest of the time.

Figures 9.8, 9.9 and 9.10 show the drug tension in the lung, brain and the other tissues respectively. These graphs closely resemble those given by Mapleson (1964).

The broad picture is that the tensions increase whilst N_2O is breathed in, and then they decrease. In this case, it is the brain tension which determines depth of anaesthesia; the fact that this takes a long time to decay is of concern in the context of post-operative recovery. The reason is that the time constants associated with two of the pools – the

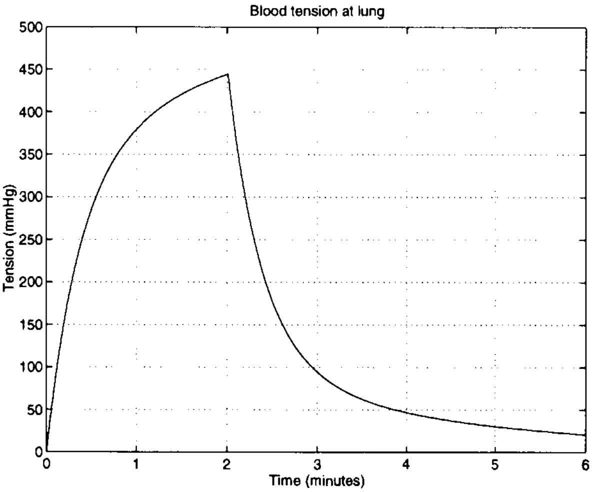


Figure 9.8 A detailed pharmacokinetic model: lung tension

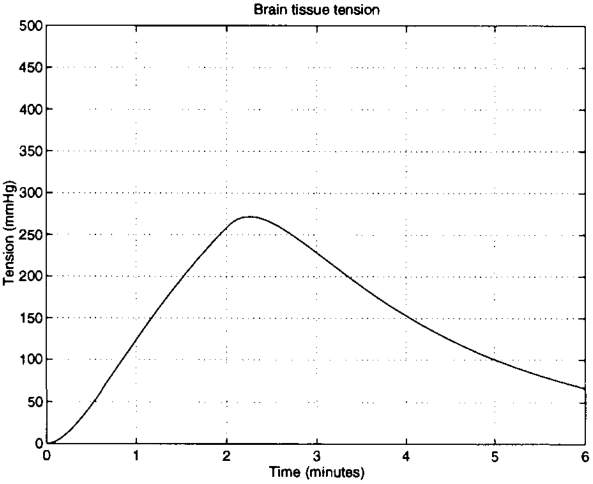


Figure 9.9 A detailed pharmacokinetic model: brain tension

fat and the lean tissues – are long; drug continues to leak out of these for an extended period.

Many other representations can be generated. For example, the magnitude and phase of the frequency-response relating the input tension to brain tension (Figures 9.11 and 9.12). For example, this could form the basis of a frequency-domain control design.

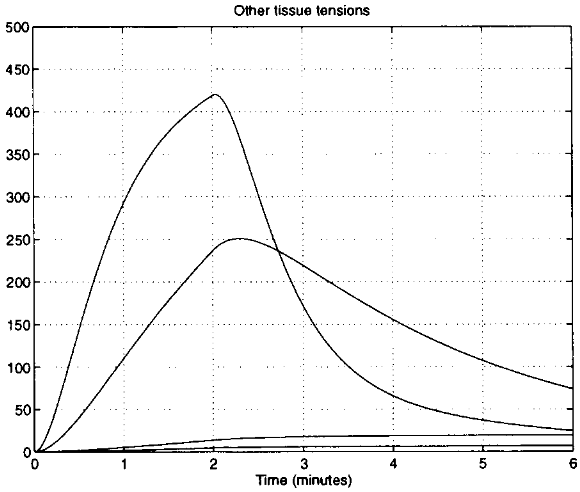


Figure 9.10 A detailed pharmacokinetic model: other tissue tensions

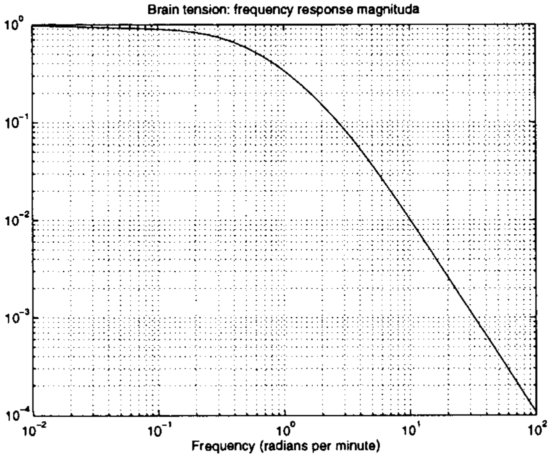


Figure 9.11 A detailed pharmacokinetic model: frequency-response magnitude

9.5 AN APPROXIMATE PHARMACOKINETIC MODEL

As discussed in Chapter 5, bond graphs provide a convenient way of approximating dynamic systems; this section shows how the six pool model may be reduced to a four pool model by lumping together the fat/lean and the viscera/shunt pools. This is achieved in Figure 9.13 by forcing the pools in each pair to have a common tension via the bonds associated with the two additional junctions.

In this case, the bond graph has some C elements with *derivative* causality. As in

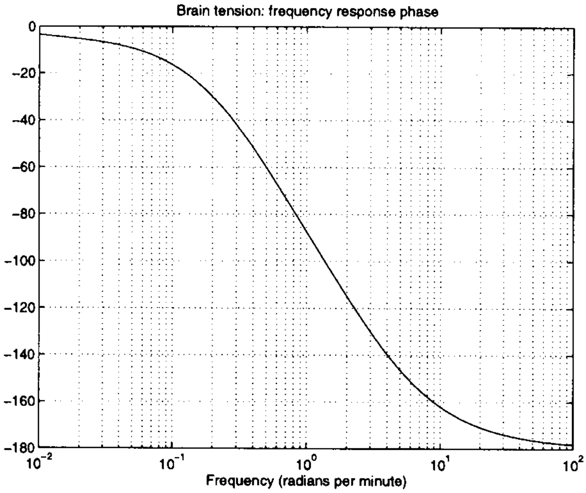


Figure 9.12 A detailed pharmacokinetic model: frequency-response phase

Section 4.10 (Equation 4.55), the system may be described by *constrained-state* equations of the form

$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{9.31}$$

where

$$x = \begin{pmatrix} m_i \\ m_b \\ m_v \\ m_f \end{pmatrix} \; ; \; z = \begin{pmatrix} m_l \\ m_s \end{pmatrix} ; \; y = \begin{pmatrix} t_i \\ t_b \\ t_v \\ t_l \\ t_f \\ t_s \end{pmatrix} \; ; \; u = \left(t_0 \right) \tag{9.32}$$

$$A = \begin{pmatrix} \frac{(-(k_b+k_v+k_l+k_f+k_s+k_i))}{c_i} & \frac{k_b}{c_b} & \frac{(k_v+k_s)}{c_v} & \frac{(k_l+k_f)}{c_f} \\ \frac{k_b}{c_i} & \frac{(-k_b)}{c_b} & 0 & 0 \\ \frac{(k_v+k_s)}{c_i} & 0 & \frac{(-(k_v+k_s))}{c_v} & 0 \\ \frac{(k_l+k_f)}{c_i} & 0 & 0 & \frac{(-(k_l+k_f))}{c_f} \end{pmatrix} \tag{9.33}$$

$$B = \begin{pmatrix} k_i \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{9.34}$$

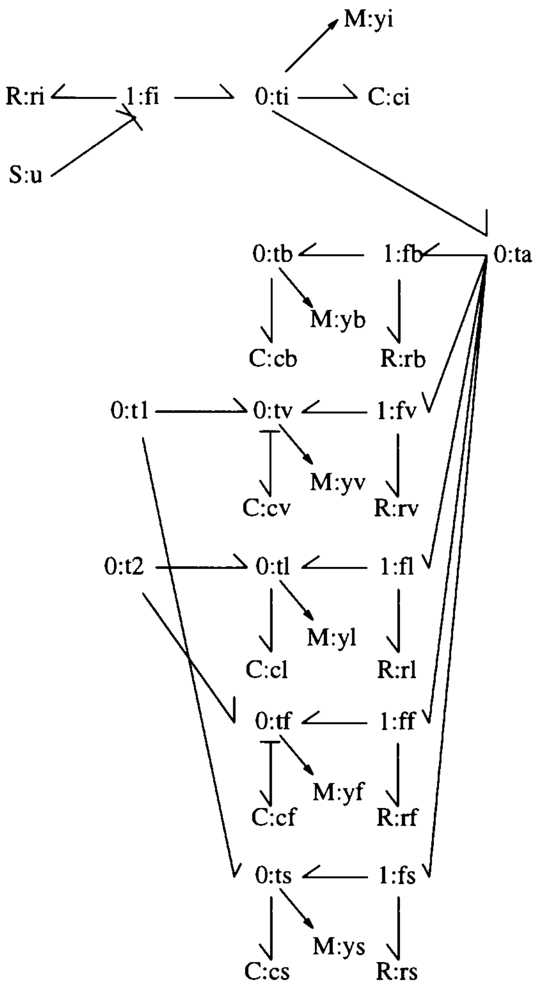


Figure 9.13 An approximate pharmacokinetic model: bond graph

$$C = \begin{pmatrix} \frac{1}{c_i} & 0 & 0 & 0 \\ 0 & \frac{1}{c_b} & 0 & 0 \\ 0 & 0 & \frac{1}{c_v} & 0 \\ 0 & 0 & 0 & \frac{1}{c_f} \\ 0 & 0 & 0 & \frac{1}{c_f} \\ 0 & 0 & \frac{1}{c_v} & 0 \end{pmatrix} \tag{9.35}$$

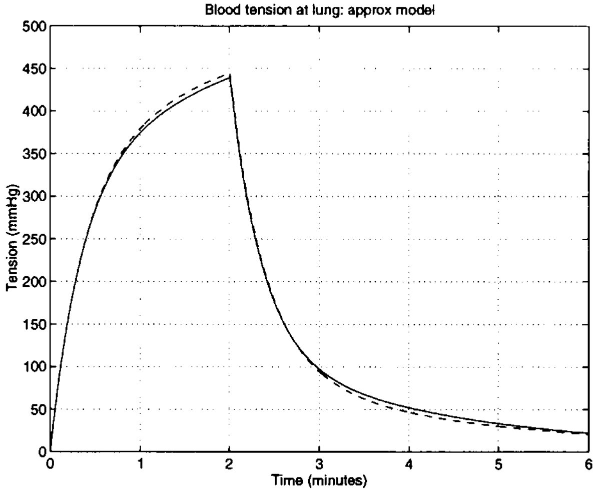


Figure 9.14 An approximate pharmacokinetic model: lung tension

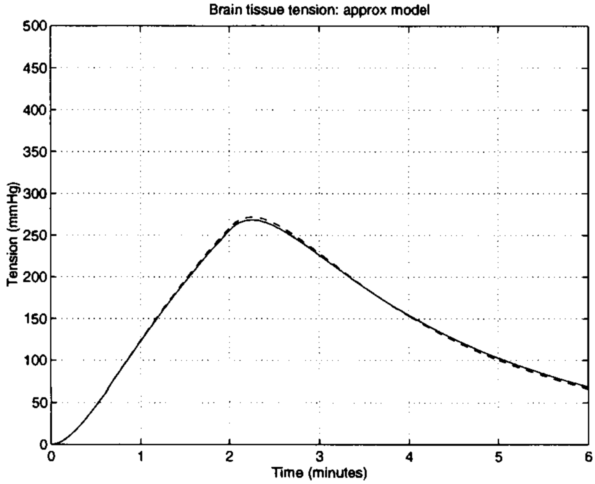


Figure 9.15 An approximate pharmacokinetic model: brain tension

$$D = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{9.36}$$

$$\underline{E} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{(c_v+c_s)}{c_v} & 0 \\ 0 & 0 & 0 & \frac{(c_f+c_l)}{c_f} \end{pmatrix} \tag{9.37}$$

The same effect could be obtained by creating a new model with the appropriate pools lumped together; however, the approach taken here is more direct and does not require remodelling.

Once again, this linear system was simulated with the data in Table 9.5 corresponding to a 'standard man' (Mapleson, 1964) breathing 75% N_2O at atmospheric pressure of 760 mmHg for 4 minutes and air for the rest of the time.

Figures 9.14 and 9.15 show the drug tension in the lung and brain, respectively shown in firm lines for the approximate model and dashed lines for the detailed model. As far as lung and brain tensions are concerned, the approximate model is adequate.

9.6 A MORE DETAILED PHARMACOKINETIC MODEL

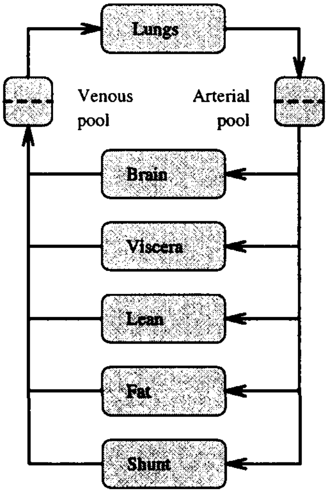


Figure 9.16 A more detailed pharmacokinetic model

The models considered in the previous sections assume that the storage of drug in the blood is associated with the individual tissue pools. The more complex model of Mapleson (1973) has separate pools for the blood. This can be important as there may be a significant delay introduced by the artery and vein. There are a number of ways of extending the model in this way, one of which appears in Figure 9.16. The arterial and the venous blood each have a separate pool; each of which is divided into two equal portions.

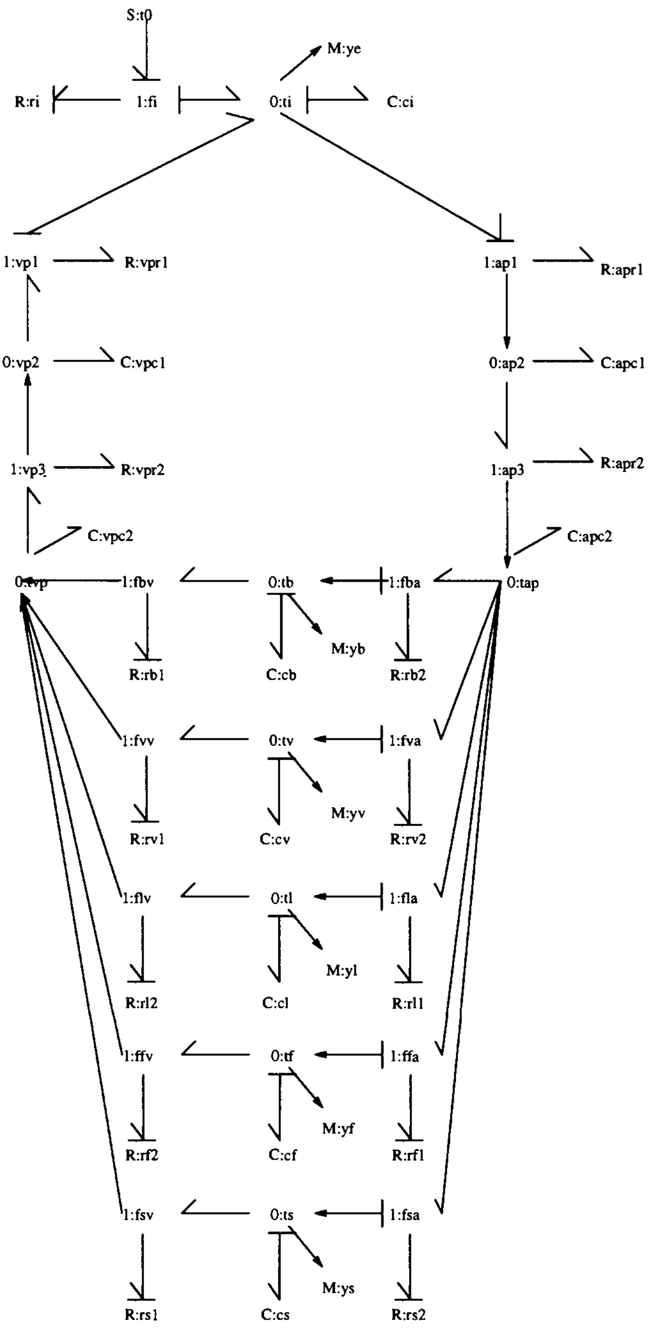


Figure 9.17 A more detailed pharmacokinetic model: bond graph

Because the tissue pools no longer connect to a single pool (the lung) it is no longer

Component label	Component type	Associated physical variable
u	Source	Inspired tension T_i
ti	Common tension junction	Lung tension T_L
ci	C	Quantity of drug stored in lung
apl	Common flow junction	Drug flow from lung
vpl	Common flow junction	Drug flow into lung
apcl, apc2	C	Quantity of drug stored in arterial pools
vpcl, vpc2	C	Quantity of drug stored in venous pools
ap3	Common flow junction	Drug flow in artery
vp3	Common flow junction	Drug flow in vein
tb	Common tension junction	Brain tension T_b
t0	Source	Inspired tension T_i
cb	C	Quantity of drug stored in brain
fb	Common flow junction	Arterial drug flow to brain
rb	R	Arterial drug flow to brain
fbv	Common flow junction	Venous drug flow from brain
rbv	R	Venous drug flow from brain

Table 9.6 A more detailed pharmacokinetic model: bond graph labels

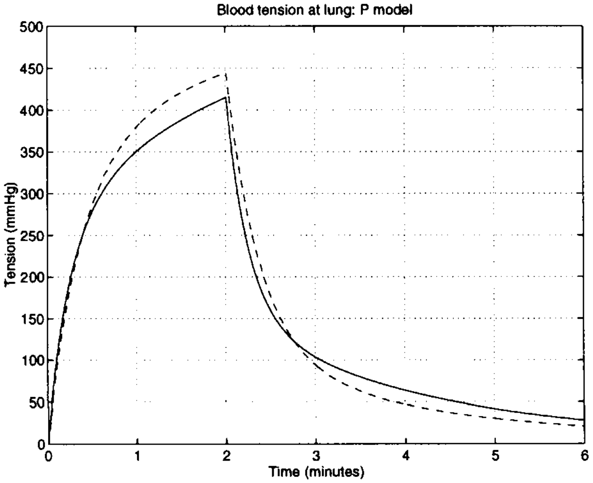


Figure 9.18 A more detailed pharmacokinetic model: lung tension

possible to use a passive bond graph.

The bond graph of Figure 9.17 shows an *active* representation of the more detailed system; it is an extension of the bond graph of Figure 9.7. It is topologically similar to Figure 9.16.

Table 9.4 gives a description of the bond graph components; the components corresponding to the viscera, lean, fat and shunt pools have suffix v,l,f and s respectively.

Once again, this linear system was simulated with the data in Table 9.5 corresponding

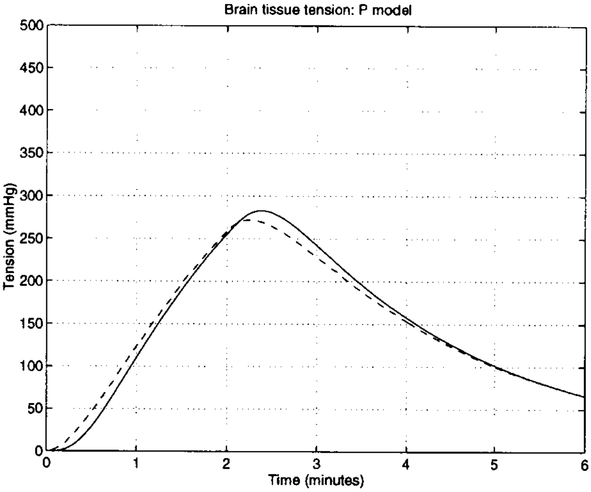


Figure 9.19 A more detailed pharmacokinetic model: brain tension

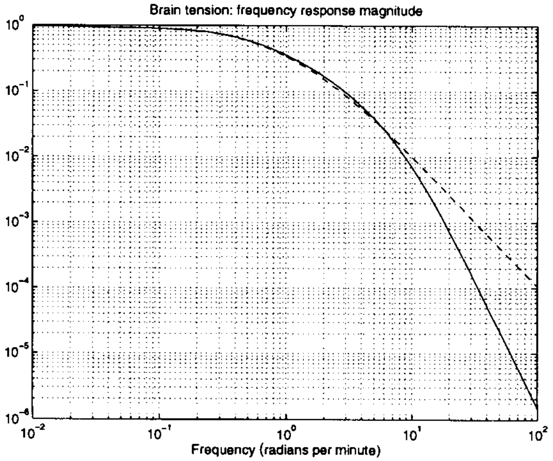


Figure 9.20 A more detailed pharmacokinetic model: frequency-response magnitude

to a 'standard man' (Mapleson, 1964) breathing 75% N_2O at atmospheric pressure of 760 mmHg for 4 minutes and air for the rest of the time. Figures 9.18 and 9.19 show the drug tension in the lung and brain, respectively shown in firm lines for the more detailed model and dashed lines for the detailed model. The effect of the redistribution of the blood in the model is to somewhat delay the responses.

Figures 9.20 and 9.21 show the magnitude and phase of the frequency-response relating the input tension to brain tension for the more detailed model. The increased phase lag in Figure 9.21 as compared to that in Figure 9.12 has implications for control system

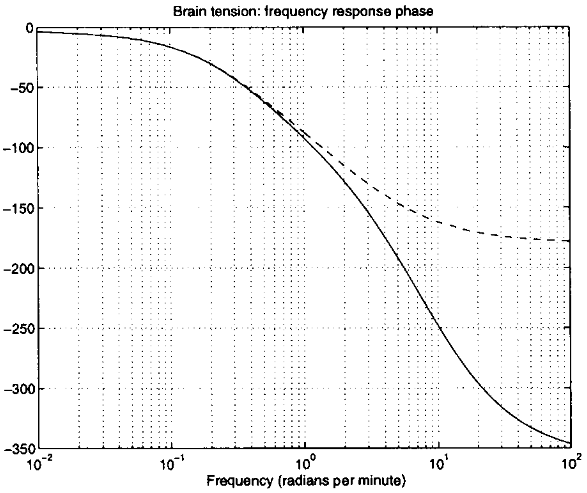


Figure 9.21 A more detailed pharmacokinetic model: frequency-response phase

design.

10

Mechanical and robotic systems

SUMMARY

- The systematic construction of dynamic models of mechanical systems composed of interconnected rods is introduced.
- The automated modelling of three-dimensional manipulators is discussed.

10.1 INTRODUCTION

There are a number of ways of obtaining the equations of motion of mechanical systems. The purpose of this chapter is to illustrate how bond graphs provide a *systematic* way of describing mechanical systems and their corresponding dynamics. The seminal paper in this area is that of Karnopp (Karnopp, 1969), and extensions appear in a recent text book (Karnopp *et al.*, 1990).

A key notion, introduced by Karnopp (Karnopp, 1969), is that of the representation of geometric *transformations* by bond graph (energy conserving) *transformers*. This enables bond graphs to be written down from *geometric* considerations; the power conservation automatically implies the forces corresponding to the velocities.

These ideas are illustrated by examples in which mechanical systems are built up from masses, springs and dampers connected by rigid rods. The exposition is initially restricted to two dimensional motion; but Section 10.6 extends the ideas to three dimensional motion.

10.2 TWO-DIMENSIONAL MOTION: THE RIGID ROD

The rigid rod of Figure 10.1, moving in the plane, is a standard component of mechanical systems. The three significant locations on the rod are the two tips and the centre of mass. The tips are significant because they define the connections to the rod; the centre of mass is significant because the equations of motion are most conveniently written there. The rigid rod thus acts as a *constraint* between these three spatial locations.

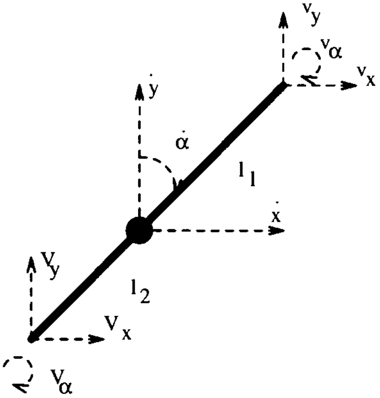


Figure 10.1 A rigid rod

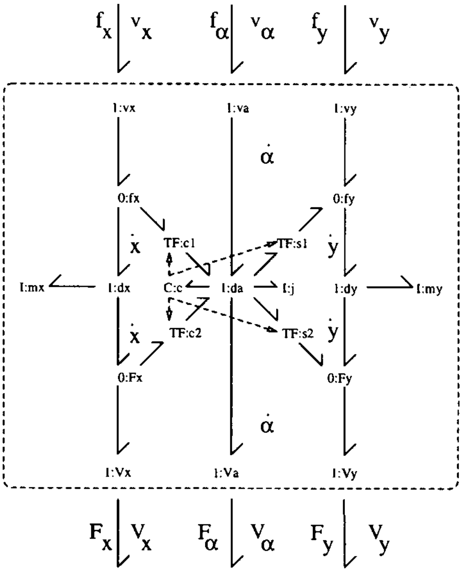


Figure 10.2 A rigid rod: bond graph

Motion is considered with respect to an absolute coordinate system: v_x and v_y are the components the velocity of one tip with respect to this coordinate system; V_x and V_y are the components of the velocity of the other tip; $\dot{x}$ and $\dot{y}$ are the components the velocity of the centre of mass. These three locations share the same angular velocity $\dot{\alpha} = v_a = V_a$. The distance from the first tip to the centre of mass is l_1, and the distance from the second tip to the centre of mass is is l_2.

The *kinematics* of the rod are expressed by the equations

$$\dot{x} = v_x - v_{x\alpha}$$

$$\begin{aligned} \dot{y} &= v_y + v_{y\alpha} \\ \dot{V}_x &= \dot{x} - V_{x\alpha} \\ \dot{V}_y &= \dot{y} + V_{y\alpha} \end{aligned} \tag{10.1}$$

Where $v_{x\alpha}$, $v_{y\alpha}$, $V_{x\alpha}$ and $V_{y\alpha}$ are the velocity components due to the angular velocity $\dot{\alpha}$ given by the *transformation* equations

$$\begin{aligned} v_{x\alpha} &= l_1 \cos\alpha \; \dot{\alpha} \\ v_{y\alpha} &= l_1 \sin\alpha \; \dot{\alpha} \\ V_{x\alpha} &= l_2 \cos\alpha \; \dot{\alpha} \\ V_{y\alpha} &= l_2 \sin\alpha \; \dot{\alpha} \end{aligned} \tag{10.2}$$

The *dynamics* of the rod are given by the three (Newton-Euler) equations

$$\begin{aligned} \ddot{x} &= \frac{\Delta f_x}{m} \\ \ddot{y} &= \frac{\Delta f_y}{m} \\ \ddot{\alpha} &= \frac{\Delta\tau}{J} \end{aligned} \tag{10.3}$$

where m and J are the mass and inertia of the rod respectively and Δf_x and Δf_y are the *net* forces acting in the x and y directions at the centre of mass and $\Delta\tau$ is the net torque acting at the centre of mass.

The corresponding bond graph appears in Figure 10.2. The area within the dotted box represents the rod itself, the six external bonds indicate how connections are made to the rod.

- There are three I components labelled 'mx', 'my', and 'j'; these implement the three *dynamic* Equations 10.3.
- There is one C component labelled 'c' this has zero stiffness and thus does not effect the system behaviour, but its corresponding integrated flow variable q is

  $$q = \int_0^t \dot{\alpha}(t')dt' = \alpha + q_0 \tag{10.4}$$

  If initialised in such a way that $q_0 = 0$, $q = \alpha$ and thus provides a modulating signal for the transformers.
- There are three 1-junctions labelled 'dx', 'dy', and 'da'.
  - These three 1-junctions carry the three velocities associated with the centre of mass: $\dot{x}$, $\dot{y}$, and $\dot{\alpha}$.
  - These three 1-junctions each compute the net effort acting on the corresponding I element (Δf_x, Δf_y and $\Delta\tau$).
- There are four 0-junctions labelled 'fx', 'fy', 'Fx', 'Fy'.
  - These four 0-junctions carry the x and y components of the force associated with the *upper* and *lower* parts of the rod.

- These four 0-junctions imply the four kinematic equations 10.1.

- There are four *transformers* labelled ‘c1’, ‘s1‘, ‘c2’ and ‘s2’.
  - These four *transformers* imply the four *transformation* Equations 10.2.
  - These four *transformers*, by power conservation, also imply the corresponding *force* transformations.
  - These four transformers are each *modulated* by α (generated by the zero-stiffness compilance labelled ‘c’).

This basic two dimensional building block may be used to construct dynamic systems in various ways as illustrated in the following examples.

10.3 A SIMPLE PENDULUM

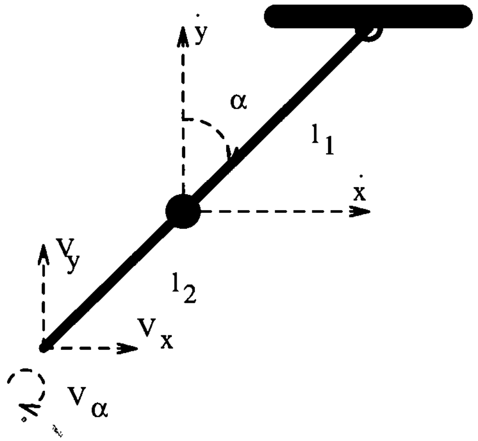


Figure 10.3 A simple pendulum

Figure 10.3 shows a simple pendulum made by attaching the upper end of the rod to a fixed rigid body via a frictionless hinge. A torque may be applied to the upper end of the pendulum. For simplicity in this example take $l_1 = l_2 = l$. A gravity force $u_1 = f_g$ acts at the centre of mass in a vertically downward direction.

Because the pendulum is a special case of the rod, the bond graph (Figure 10.4) is the same as that of the rod but with additional components to take account of the constraints implied by the attachment. In particular:

- Zero *velocity* sources are attached to junctions ‘vx’ and ‘vy’ to indicate that the upper tip is fixed in the horizontal and vertical directions.

- Zero *force* sources are attached to junctions ‘Vx’ and ‘Vy’ to indicate that the lower tip has no forces acting on it.

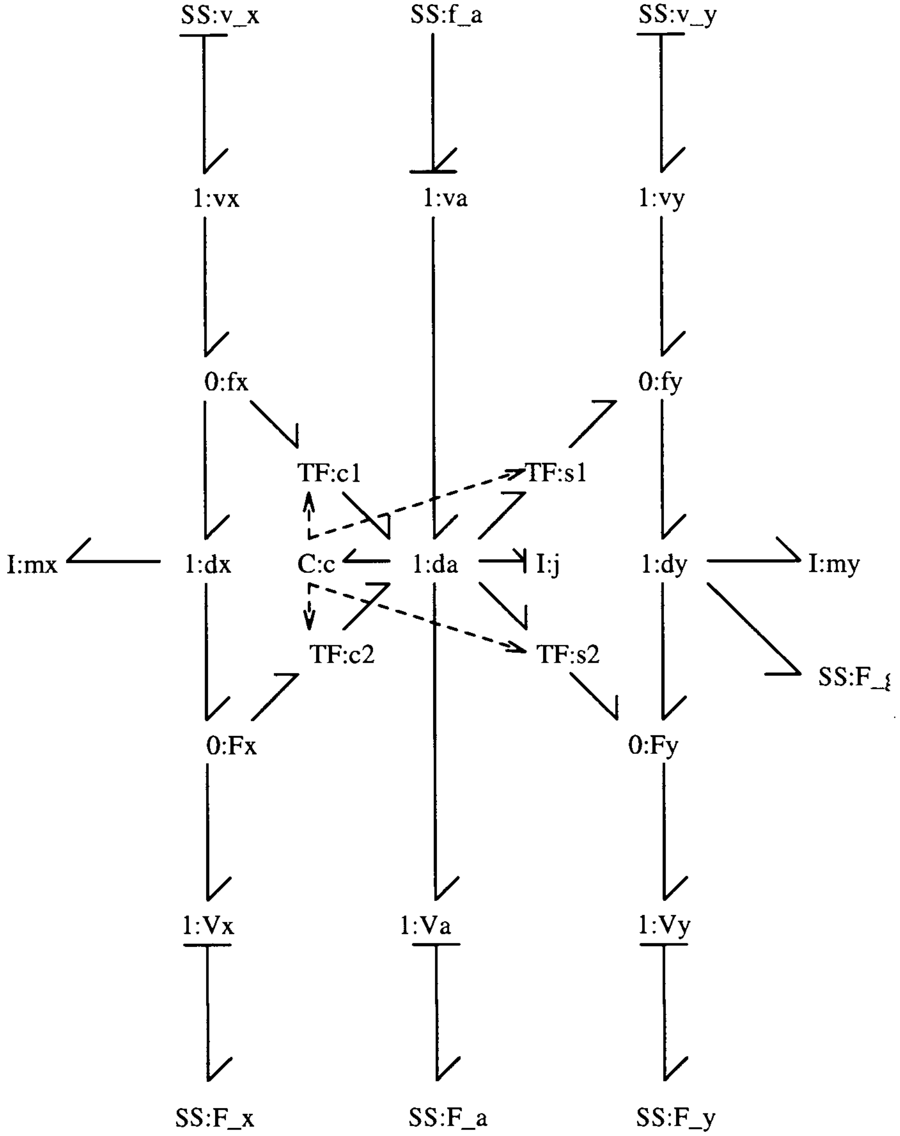


Figure 10.4 A simple pendulum: bond graph

- A *force* source (with half arrow pointing outwards) is attached to junction 'dy to indicate that gravity is acting in the negative y direction at the centre of mass.
- A *force* source is attached to junction 'va' to indicate the applied torque at he hinge.

Only one of the three I elements may have integral causality, the one correspondng to the rotation has been chosen in Figure 10.4.

The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} h_\alpha \\ \alpha \end{pmatrix};\ z = \begin{pmatrix} h_x \\ h_y \end{pmatrix};\ y = (\,\alpha\,);\ u = \begin{pmatrix} f_g \\ \tau \end{pmatrix} \tag{10.5}$$

$$\begin{aligned} \dot{x}_1 &= -((u_1 + \dot{z}_2)\sin(x_2)l - \cos(x_2)l\dot{z}_1 - u_2) \\ \dot{x}_2 &= \frac{x_1}{j} \end{aligned} \tag{10.6}$$

$$\begin{aligned} z_1 &= \frac{(-\cos(x_2)lmx_1)}{j} \\ z_2 &= \frac{(\sin(x_2)lmx_1)}{j} \end{aligned} \tag{10.7}$$

$$y_1 = \frac{x_1}{j} \tag{10.8}$$

These equations are linear in the non-state derivative $\dot{z}_i$ terms and so may be rewritten in *constrained-state* form (Section 4.12) as

$$\dot{\chi}_1 = -(\sin(x_2)lu_1 - u_2) \tag{10.9}$$

$$\dot{\chi}_2 = \frac{x_1}{j} \tag{10.10}$$

$$y_1 = \frac{x_1}{j} \tag{10.11}$$

$$E = \begin{pmatrix} \frac{(j+l^2m)}{j} & 0 \\ 0 & 1 \end{pmatrix} \tag{10.12}$$

Not surprisingly, the E matrix essentially converts the inertia j about the centre of mass to the inertia $j + ml^2$ about the tip. The equation is non-linear due to the gravity force u_1. The differential-algebraic equation may be linearised about $\alpha = 0$; $\dot{\alpha} = 0$ to give the linear descriptor equation (Section 4.10) with matrices

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{10.13}$$

$$A = \begin{pmatrix} 0 & -lu_1 & 0 & 0 & l & 0 \\ \frac{1}{j} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{(-lm)}{j} & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{pmatrix} \tag{10.14}$$

$$B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{10.15}$$

$$C = \begin{pmatrix} \frac{1}{j} & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{10.16}$$

$$D = \begin{pmatrix} 0 & 0 \end{pmatrix} \tag{10.17}$$

The corresponding transfer function is

$$G_{11}(s) = 0 \tag{10.18}$$

$$G_{12}(s) = \frac{+s}{lu_1 + (j + l^2 m)s^2} \tag{10.19}$$

Notice that the transfer function relating gravity to angle is zero; the gravity term u_1 appears as a modulation in the transfer function relating torque to angular velocity ($\dot{\alpha}$).

The bond graph of Figure 10.4 has integral causality on the angular momentum inertia. In fact, there are three possible causalities associated with the inertias: corresponding to integral causality on each of the three inertias in turn.

The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} h_x \\ \alpha \end{pmatrix}; \; z = \begin{pmatrix} h_y \\ h_\alpha \end{pmatrix}; \; y = (\, \alpha \,); \; u = \begin{pmatrix} f_g \\ \tau \end{pmatrix} \tag{10.20}$$

$$\begin{aligned} \dot{x}_1 &= \frac{((u_1 + \dot{z}_1)\sin(x_2)l - u_2 + \dot{z}_2)}{(\cos(x_2)l)} \\ \dot{x}_2 &= \frac{(-x_1)}{(\cos(x_2)lm)} \end{aligned} \tag{10.21}$$

$$\begin{aligned} z_1 &= \frac{(-\sin(x_2)x_1)}{\cos(x_2)} \\ z_2 &= \frac{(-jx_1)}{(\cos(x_2)lm)} \end{aligned} \tag{10.22}$$

$$y_1 = \frac{(-x_1)}{(\cos(x_2)lm)} \tag{10.23}$$

These equations are linear in the non-state derivative $\dot{z}_i$ terms and so may be rewritten in *constrained-state* form (Section 4.12) as

$$\dot{\chi}_1 = \frac{(\sin(x_2)lu_1 - u_2)}{(\cos(x_2)l)} \tag{10.24}$$

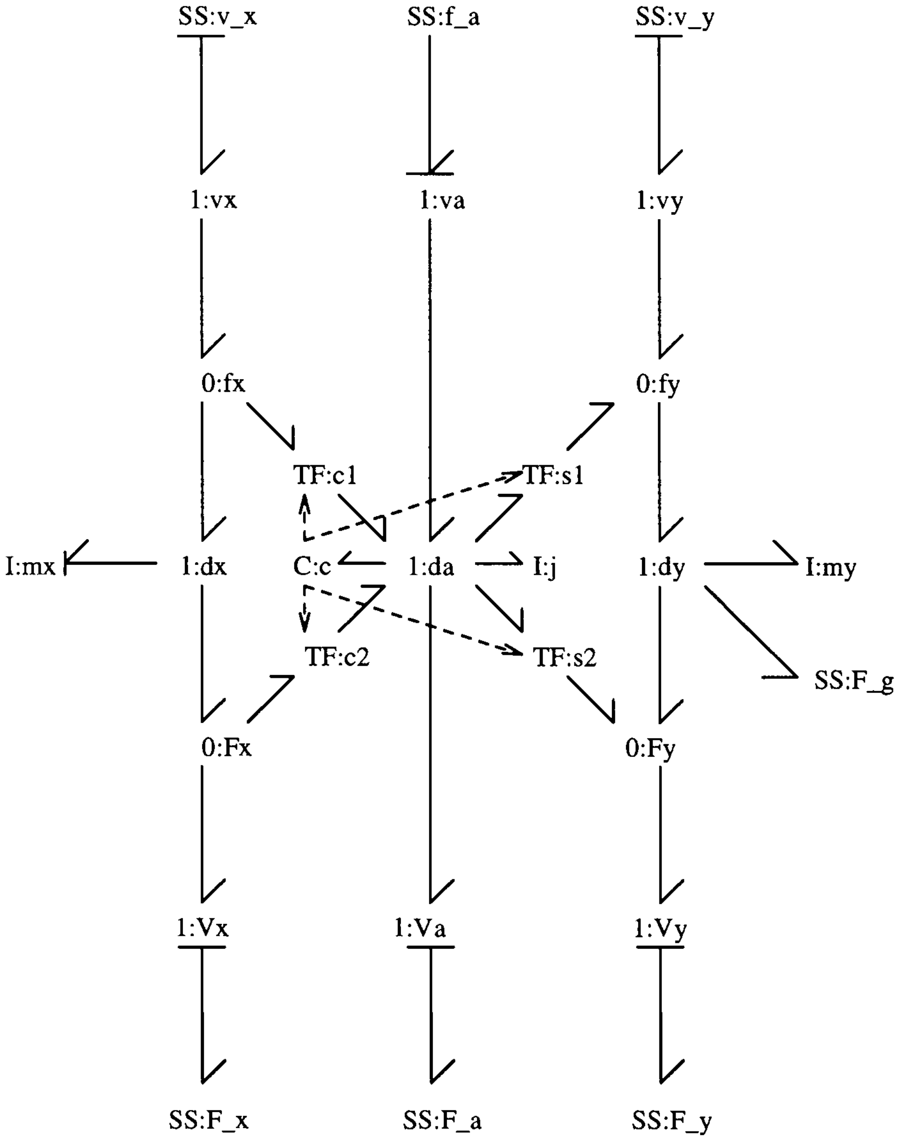


Figure 10.5 A simple pendulum: bond graph with different causality

$$\dot{\chi}_2 = \frac{(-x_1)}{(\cos(x_2)lm)} \tag{10.25}$$

$$y_1 = \frac{(-x_1)}{(\cos(x_2)lm)} \tag{10.26}$$

$$E = \begin{pmatrix} \frac{(j+l^2m)}{(\cos(x_2)^2 l^2 m)} & \frac{((j+l^2m)\sin(x_2)x_1)}{(\cos(x_2)^3 l^2 m)} \\ 0 & 1 \end{pmatrix} \tag{10.27}$$

Thus the change in causal pat ern gives a cartesian state-space description in place of the polar state-space description. The same system has quite different representations at this level. However, the resultant linearised transfer function is identical.

10.3.1 A simple pendulum with bob

There are variations on this theme. For example, a bob of mass m_b and small dimension (and thus zero inertia) could be added to the lower tip. Figure 10.6 is identical to Figure 10.4 except that the two lower force sources have been replaced by I components corresponding to the x and y velocities of the bob, and a third input $u_3 = f_b$ has been added in the form of an additional source labelled F_b in Figure 10.6.

The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} h_\alpha \\ \alpha \end{pmatrix}; \; z = \begin{pmatrix} h_x \\ h_y \\ mb_x \\ mb_y \end{pmatrix}; \; y = (\,\alpha\,); \; u = \begin{pmatrix} f_g \\ \tau \\ f_b \end{pmatrix} \tag{10.28}$$

$$\begin{aligned} \dot{x}_1 &= -((u_1 + 2u_3 + \dot{z}_2 + 2\dot{z}_4)\sin(x_2)l - (\dot{z}_1 + 2\dot{z}_3)\cos(x_2)l - u_2) \\ \dot{x}_2 &= \frac{x_1}{j} \end{aligned} \tag{10.29}$$

$$\begin{aligned} z_1 &= \frac{(-\cos(x_2)lmx_1)}{j} \\ z_2 &= \frac{(\sin(x_2)lmx_1)}{j} \\ z_3 &= \frac{(-2\cos(x_2)lm_bx_1)}{j} \\ z_4 &= \frac{(2\sin(x_2)lm_bx_1)}{j} \end{aligned} \tag{10.30}$$

$$y_1 = \frac{x_1}{j} \tag{10.31}$$

Again, these equations are linear in the non-state derivative $\dot{z}_i$ terms and so may be rewritten in *constrained-state* form (Section 4.10) as

$$\dot{\chi}_1 = -((u_1 + 2u_3)\sin(x_2)l - u_2) \tag{10.32}$$

$$\dot{\chi}_2 = \frac{x_1}{j} \tag{10.33}$$

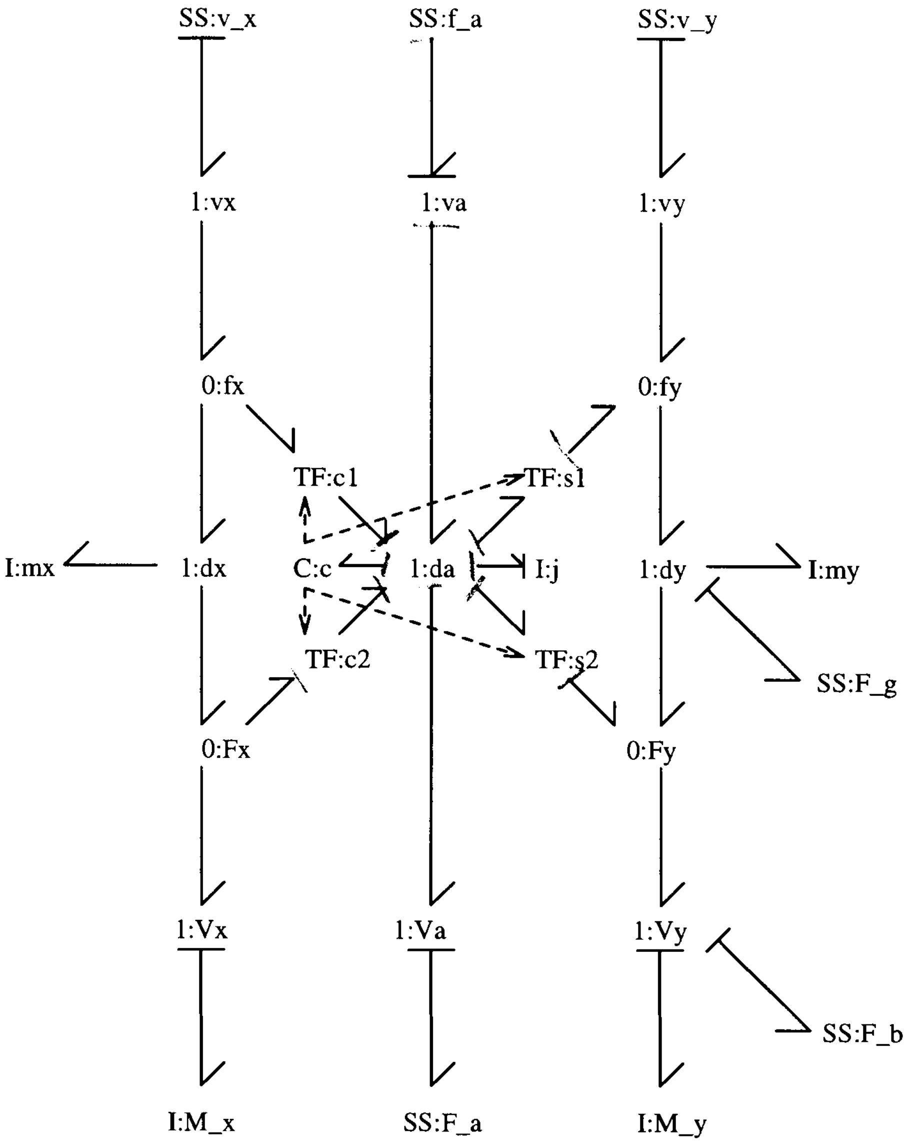


Figure 10.6 A simple pendulum with bob: bond graph

$$y_1 = \frac{x_1}{j} \tag{10.34}$$

$$E = \begin{pmatrix} \frac{((m+4m_b)l^2+j)}{j} & 0 \\ 0 & 1 \end{pmatrix} \tag{10.35}$$

This time, the equivalent inertia is $j + ml^2 + m_b(2l)^2$.

Linearised about $\alpha = 0$; $\dot{\alpha} = 0$, the system transfer function is

$$G_{11}(s) = 0 \tag{10.36}$$

$$G_{12}(s) = \frac{+s}{(l(u_1 + 2u_3)) + (j + l^2 m + 4l^2 m_b)s^2} \tag{10.37}$$

$$G_{13}(s) = 0 \tag{10.38}$$

Again, the transfer functions relating the gravity terms (u_1 and u_3) to angular velocity ($\dot{\alpha}$) are zero, but both $u_1 = gm$ and $u_3 = gm_b$ modulate the transfer function relating τ to angular velocity ($\dot{\alpha}$).

10.3.2 Inverted pendulum and cart

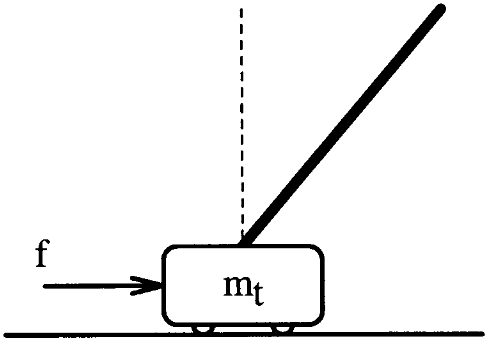


Figure 10.7 Inverted pendulum

Figure 10.7 shows an inverted pendulum comprising a uniform rod of mass m and length $2l$ attached to a cart of mass m_c via a frictionless pivot. Figure 10.8 shows the bond graph. The basis of this bond graph is that of the rod of Figure 10.1 with the bond graph of Figure 10.2. The x velocity of the lower end of the pendulum shares the velocity of the cart, so the I element m_c is added to the corresponding 1:junction V_x. The y velocity of the lower end of the pendulum is zero, so a zero flow source V_y is attached to the corresponding 1:junction V_y. The there is no torque acting at the lower end of the pendulum so a zero effort source F_a is added to the corresponding 1:junction V_a.

The upper end of the rod has no applied forces or torques - the appropriate zero effort sources are added to junctions v_x, v_a and v_y. As before, the inertia of the rod about its centre is $\frac{1}{3}ml^2$.

The expressions in the following equations have been simplified by substituting

$$\rho = \frac{m}{m_c} \tag{10.39}$$

The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} h_x \\ h_\alpha \\ \alpha \end{pmatrix} ; \; z = \begin{pmatrix} h_y \\ h_c \end{pmatrix} ; \; y = (\, \alpha \,) ; \; u = \begin{pmatrix} f_g \\ f \end{pmatrix} \tag{10.40}$$

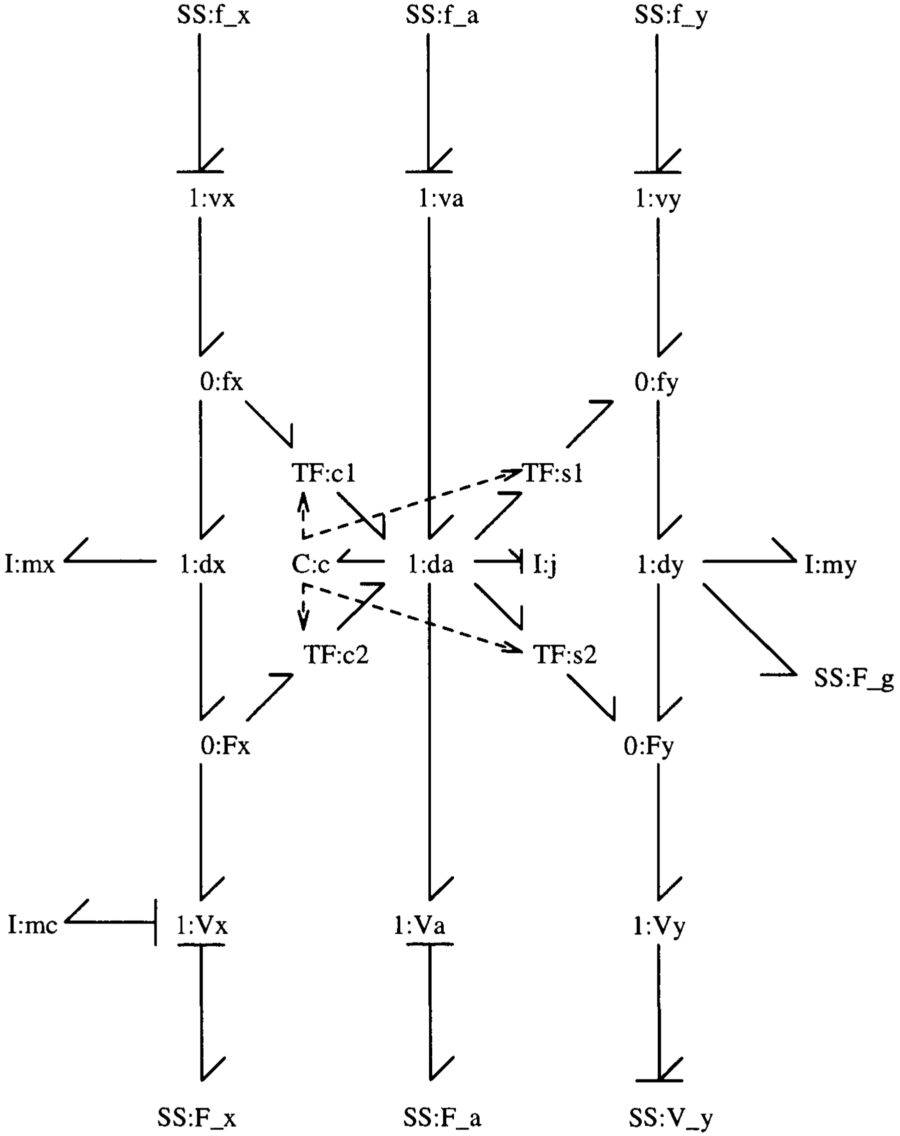


Figure 10.8 Inverted pendulum: bond graph

$$\begin{aligned}
\dot{x}_1 &= -(u_2 + \dot{z}_2) \\
\dot{x}_2 &= ((u_1 + \dot{z}_1)\sin(x_3) + (u_2 + \dot{z}_2)\cos(x_3))l \\
\dot{x}_3 &= \frac{(3x_2)}{(l^2 \rho m_c)}
\end{aligned} \tag{10.41}$$

$$\begin{aligned} z_1 &= \frac{(-3\sin(x_3)x_2)}{l} \\ z_2 &= \frac{(-(3\cos(x_3)x_2 - lx_1))}{(l\rho)} \end{aligned} \tag{10.42}$$

$$y_1 = \frac{(3x_2)}{(l^2\rho m_c)} \tag{10.43}$$

Again, these equations are linear in the non-state derivative $\dot{z}_i$ terms and so may be rewritten in *constrained-state* form (Section 4.10) as

$$\dot{\chi}_1 = -u_2 \tag{10.44}$$

$$\dot{\chi}_2 = (\sin(x_3)u_1 + \cos(x_3)u_2)l \tag{10.45}$$

$$\dot{\chi}_3 = \frac{(3x_2)}{(l^2\rho m_c)} \tag{10.46}$$

$$y_1 = \frac{(3x_2)}{(l^2\rho m_c)} \tag{10.47}$$

$$E = \begin{pmatrix} \frac{(\rho+1)}{\rho} & \frac{(-3\cos(x_3))}{(l\rho)} & \frac{(3\sin(x_3)x_2)}{(l\rho)} \\ \frac{(-\cos(x_3)l)}{\rho} & \frac{(-(3(\rho-1)\cos(x_3)^2-4\rho))}{\rho} & \frac{(3(\rho-1)\sin(x_3)\cos(x_3)x_2)}{\rho} \\ 0 & 0 & 1 \end{pmatrix} \tag{10.48}$$

Linearised about a vertical, stationary pendulum and cart, the system transfer function is

$$G_{11}(s) = 0 \tag{10.49}$$

$$G_{12}(s) = \frac{+-3s}{(3gm_c(\rho+1)) + (lm_c(-\rho-4))s^2} \tag{10.50}$$

10.4 A DOUBLE PENDULUM

Figure 10.9 shows a double pendulum made from the simple pendulum by attaching a further rod to the lower tip of the first rod using a frictionless revolute joint. This joint constrains the velocities of the lower tip of the upper rod and the upper tip of the lower rod (in both x and y directions) to be the same; there is no constraint on the relative angles of the two rods. For simplicity, the rods are taken to be uniform and of length $2l$. The inertia j about the centre is thus

$$j = \frac{ml^2}{3} \tag{10.51}$$

The relative angle θ_2 of the second rod with repect to the first is the difference of the absolute angles α_1 and α_2

$$\theta_2 = \alpha_2 - \alpha_1 \tag{10.52}$$

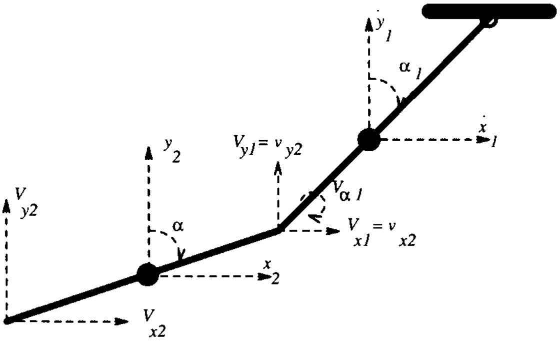


Figure 10.9 A double pendulum

Figure 10.10 shows the corresponding bond graph. A copy of the single rod bond graph has been appended to that of the simple pendulum and connected by two bonds which ensure that the velocities of the lower tip of the upper rod and the upper tip of the lower rod match in both x and y directions; a zero junction and corresponding (zero) torque source (f_a2) has been added

Again, zero-force sources have been added at the lower tip of the lower rod.

The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} h_1 \\ \alpha_1 \\ h_2 \\ \alpha_2 \end{pmatrix} ; \; z = \begin{pmatrix} h_x \\ h_y \\ h_x \\ h_y \end{pmatrix} ; \; y = (\,\alpha\,) ; \; u = \begin{pmatrix} f_1 \\ \tau_1 \\ f_2 \end{pmatrix} \tag{10.53}$$

$$\begin{aligned} \dot{x}_1 &= -((u_1 + 2u_3 + \dot{z}_2 + 2\dot{z}_4)\sin(x_2)l - (\dot{z}_1 + 2\dot{z}_3)\cos(x_2)l - u_2) \\ \dot{x}_2 &= \frac{x_1}{j} \\ \dot{x}_3 &= -((u_3 + \dot{z}_4)\sin(x_4) - \cos(x_4)\dot{z}_3)l \\ \dot{x}_4 &= \frac{x_3}{j} \end{aligned} \tag{10.54}$$

$$\begin{aligned} z_1 &= \frac{(-\cos(x_2)lmx_1)}{j} \\ z_2 &= \frac{(\sin(x_2)lmx_1)}{j} \\ z_3 &= \frac{(-(2\cos(x_2)x_1 + \cos(x_4)x_3)lm)}{j} \\ z_4 &= \frac{((2\sin(x_2)x_1 + \sin(x_4)x_3)lm)}{j} \end{aligned} \tag{10.55}$$

$$y_1 = \frac{x_3}{j} \tag{10.56}$$

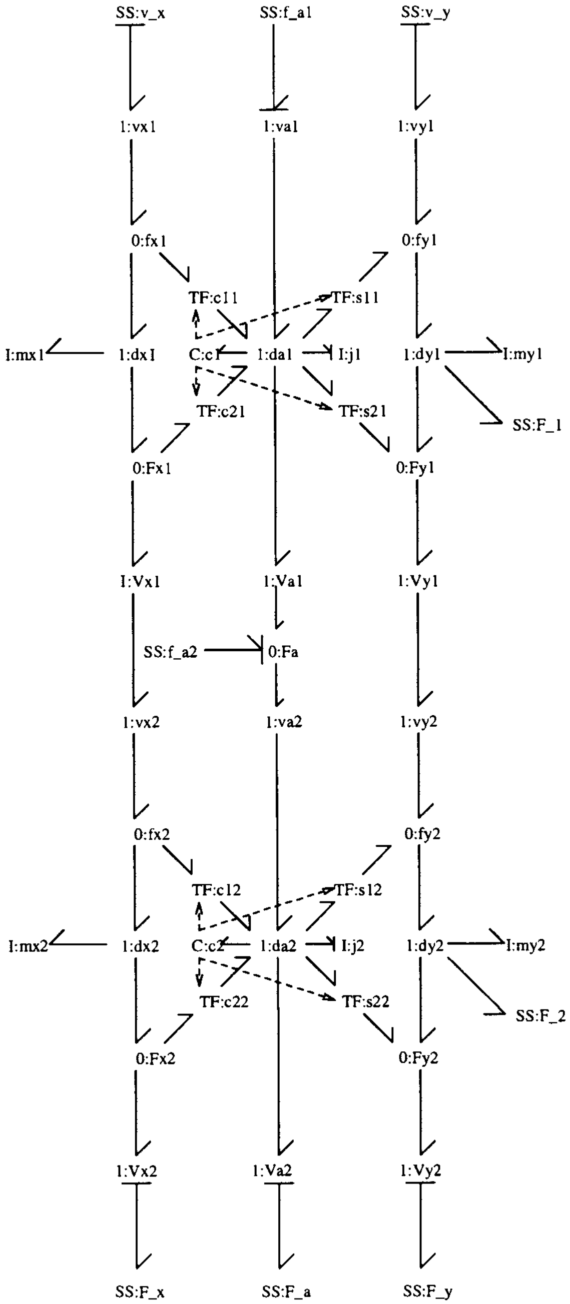


Figure 10.10 A double pendulum: bond graph

Again, these equations are linear in the non-state derivative $\dot{z}_i$ terms and so may be rewritten in *constrained-state* form (Section 4.10) as

$$\dot{\chi}_1 = -((u_1 + 2u_3)\sin(x_2)l - u_2) \tag{10.57}$$

$$\dot{\chi}_2 = \frac{x_1}{j} \tag{10.58}$$

$$\dot{\chi}_3 = -\sin(x_4) l u_3 \tag{10.59}$$

$$\dot{\chi}_4 = \frac{x_3}{j} \tag{10.60}$$

$$y_1 = \frac{x_3}{j} \tag{10.61}$$

$$E = \begin{pmatrix} 16 & 0 & 6c_1 & 6s_1x_3 \\ 0 & 1 & 0 & 0 \\ 6c_1 & -6s_1x_1 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10.62}$$

The following substitutions have been made to simplify the equations

$$\begin{aligned} J &= \frac{ml^2}{3} \\ \theta_2 &= \alpha_2 - \alpha_1 \\ c_2 &= \cos\theta_2 \\ s_2 &= \sin\theta_2 \end{aligned} \tag{10.63}$$

Linearised about $\alpha_1 = \alpha_2 = \dot{\alpha}_1 = \dot{\alpha}_2 = 0$, the system transfer function is

$$G_{11}(s) = 0 \tag{10.64}$$

$$G_{12}(s) = \frac{+ - 6s^3}{9g^2m + 28glms^2 + 28js^4} \tag{10.65}$$

$$G_{13}(s) = 0 \tag{10.66}$$

Again, the transfer functions relating the gravity terms (u_1 and u_3) to angular velocity ($\dot{\alpha}$) are zero, but both $u_1 = gm$ and $u_3 = gm_b$ modulate the transfer function relating τ to angular velocity ($\dot{\alpha}$).

A variation on this theme arises by fixing a rotational spring at the joint between the two rods; the corresponding bond graph is given in Figure 10.11. The 0-junction labelled 'f_a2' gives the *relative* angle $\theta = \alpha_2 - \alpha_1$ across the new C component labelled c representing the spring with compliance c.

This gives an additional state corresponding to the spring displacement θ. The three angular states θ, α_1 and α_2 are not, at first sight, independent. However, the initial conditions on the three corresponding compliances are *independent* so these three states are not, in fact, dependent. The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} h_1 \\ \alpha_1 \\ \theta \\ h_2 \\ \alpha_2 \end{pmatrix}; \; z = \begin{pmatrix} h_x \\ h_y \\ h_x \\ h_y \end{pmatrix}; \; y = (\,\alpha\,); \; u = \begin{pmatrix} f_1 \\ \tau_1 \\ f_2 \end{pmatrix} \tag{10.67}$$

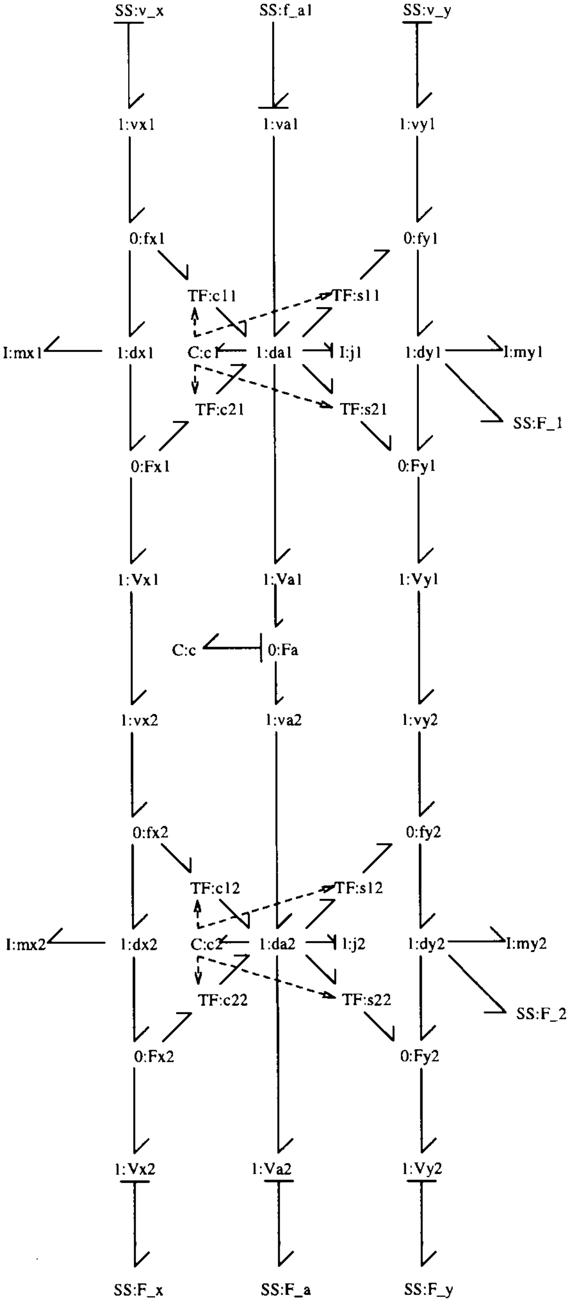


Figure 10.11 A double pendulum with spring: bond graph

$$\dot{x}_1 = \frac{(-((u_1 + 2u_3 + \dot{z}_2 + 2\dot{z}_4)\sin(x_2)cl - (\dot{z}_1 + 2\dot{z}_3)\cos(x_2)cl - cu_2 + x_3))}{c}$$

$$\begin{aligned}
\dot{x}_2 &= \frac{x_1}{j} \\
\dot{x}_3 &= \frac{(x_1 - x_4)}{j} \\
\dot{x}_4 &= \frac{(-((u_3 + \dot{z}_4)\sin(x_5)cl - \cos(x_5)cl\dot{z}_3 - x_3))}{c} \\
\dot{x}_5 &= \frac{x_4}{j}
\end{aligned} \tag{10.68}$$

$$\begin{aligned}
z_1 &= \frac{(-\cos(x_2)lmx_1)}{j} \\
z_2 &= \frac{(\sin(x_2)lmx_1)}{j} \\
z_3 &= \frac{(-(2\cos(x_2)x_1 + \cos(x_5)x_4)lm)}{j} \\
z_4 &= \frac{((2\sin(x_2)x_1 + \sin(x_5)x_4)lm)}{j}
\end{aligned} \tag{10.69}$$

$$y_1 = \frac{x_4}{j} \tag{10.70}$$

Again, these equations are linear in the non-state derivative $\dot{z}_i$ terms and so may be rewritten in *constrained-state* form (Section 4.10) as

$$\dot{\chi}_1 = \frac{(-((u_1 + 2u_3)sin(x_2)cl - cu_2 + x_3))}{c} \tag{10.71}$$

$$\dot{\chi}_2 = \frac{(3x_1)}{(l^2m)} \tag{10.72}$$

$$\dot{\chi}_3 = \frac{(3(x_1 - x_4))}{(l^2m)} \tag{10.73}$$

$$\dot{\chi}_4 = \frac{(-(sin(x_5)clu_3 - x_3))}{c} \tag{10.74}$$

$$\dot{\chi}_5 = \frac{(3x_4)}{(l^2m)} \tag{10.75}$$

$$y_1 = \frac{(3x_4)}{(l^2m)} \tag{10.76}$$

$$\begin{aligned}
E_{11} &= 16 \\
E_{14} &= 6(sin(x_2)sin(x_5) + cos(x_2)cos(x_5)) \\
E_{15} &= 6(sin(x_2)cos(x_5) - sin(x_5)cos(x_2))x_4 \\
E_{22} &= 1 \\
E_{33} &= 1 \\
E_{41} &= 6(sin(x_2)sin(x_5) + cos(x_2)cos(x_5))
\end{aligned}$$

$$
\begin{aligned}
E_{42} &= -6(sin(x_2)cos(x_5) - sin(x_5)cos(x_2))x_1 \\
E_{44} &= 4 \\
E_{55} &= 1
\end{aligned}
\tag{10.77}
$$

(10.78)

(10.79)

Linearised about $\alpha_1 = \alpha_2 = \theta = \dot{\alpha}_1 = \dot{\alpha}_2 = 0$, the system transfer function is

$$G_{11}(s) = 0 \tag{10.80}$$

$$G_{12}(s) = \frac{+s + -6cjs^3}{(gm(9cgj + 4l)) + (4j(7cglm + 8))s^2 + 28cj^2s^4} \tag{10.81}$$

$$G_{13}(s) = 0 \tag{10.82}$$

10.5 A TWO-LINK MANIPULATOR

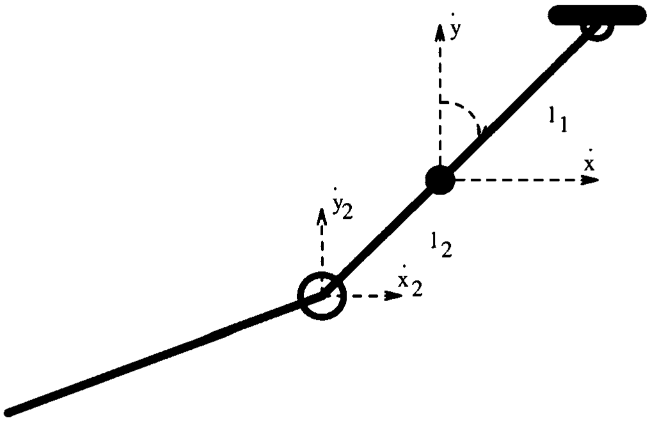


Figure 10.12 A two-link manipulator

Manipulators composed of rigid links connected by revolute joints are usually analysed via recursive Newton-Euler or Lagrange techniques (Paul, 1981; Fu *et al.*, 1987; Craig, 1989). However, bond graphs provide an alternative technique (Anex and Hubbard, 1984; Tiernego and Bos, 1985; Gawthrop, 1991) which is particularly attractive when actuator characteristics are to be modelled. The motion of manipulators in three dimensions is discussed in Section 10.6.

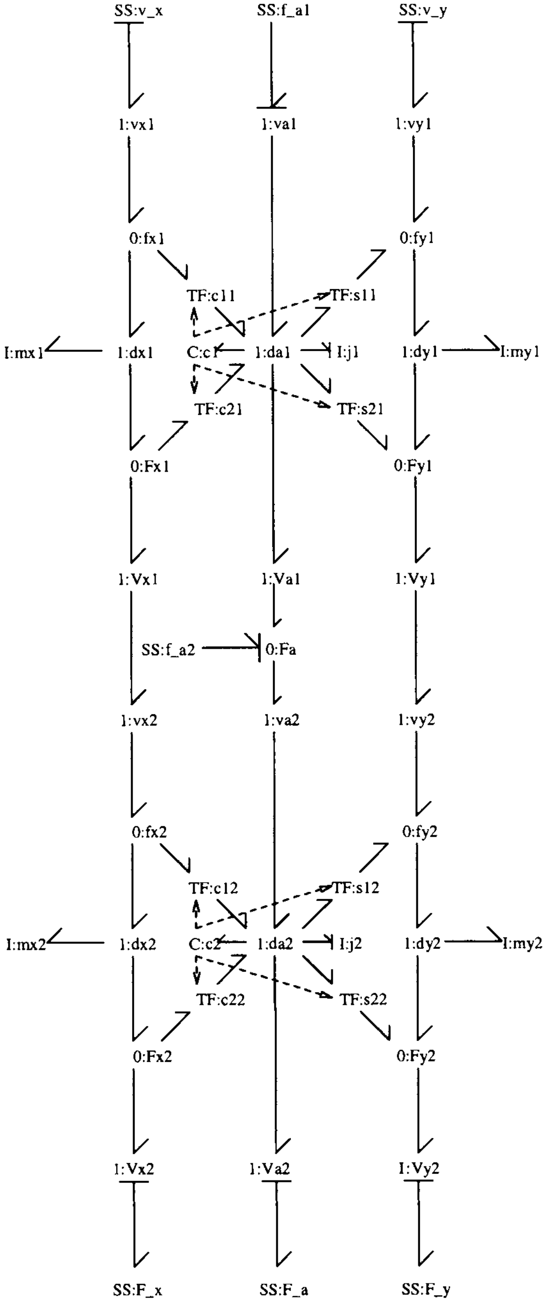


Figure 10.13 A two-link manipulator: bond graph

This Section considers the two-link manipulator depicted in Figure 10.12 composed of two uniform rigid links of length 21 moving in a horizontal plane. The *joint* angles θ_1 and θ_2 are

$$\begin{aligned} \theta_1 &= \alpha_1 \\ \theta_2 &= \alpha_2 - \alpha_1 \end{aligned} \tag{10.83}$$

The bond graph is similar to that of the double pendulum of Section 10.4 except that:

- there are no gravity terms (the manipulator moves in a horizontal plane),
- torques are applied at each joint and the corresponding joint velocities measured.

The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} h_1 \\ \alpha_1 \\ h_2 \\ \alpha_2 \end{pmatrix} ; \; z = \begin{pmatrix} h_x \\ h_y \\ h_x \\ h_y \end{pmatrix} ; \; y = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} ; \; u = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \tag{10.84}$$

$$\begin{aligned} \dot{x}_1 &= (\dot{z}_1 + 2\dot{z}_3)cos(x_2)l - (\dot{z}_2 + 2\dot{z}_4)sin(x_2)l + u_1 - u_2 \\ \dot{x}_2 &= \frac{x_1}{j} \\ \dot{x}_3 &= -(sin(x_4)l\dot{z}_4 - cos(x_4)l\dot{z}_3 - u_2) \\ \dot{x}_4 &= \frac{x_3}{j} \end{aligned} \tag{10.85}$$

$$\begin{aligned} z_1 &= \frac{(-cos(x_2)lmx_1)}{j} \\ z_2 &= \frac{(sin(x_2)lmx_1)}{j} \\ z_3 &= \frac{(-(2cos(x_2)x_1 + cos(x_4)x_3)lm)}{j} \\ z_4 &= \frac{((2sin(x_2)x_1 + sin(x_4)x_3)lm)}{j} \end{aligned} \tag{10.86}$$

$$\begin{aligned} y_1 &= \frac{x_1}{j} \\ y_2 &= \frac{(-(x_1 - x_3))}{j} \end{aligned} \tag{10.87}$$

Again, these equations are linear in the non-state derivative $\dot{z}_i$ terms and so may be rewritten in *constrained-state* form (Section 4.10) as

$$\dot{\chi}_1 = u_1 - u_2 \tag{10.88}$$

$$\dot{\chi}_2 = \frac{x_1}{j} \tag{10.89}$$

$$\dot{\chi}_3 = u_2 \tag{10.90}$$

$$\dot{\chi}_4 = \frac{x_3}{j} \tag{10.91}$$

$$y_1 = \frac{x_1}{j} \tag{10.92}$$

$$y_2 = \frac{(-(x_1 - x_3))}{j} \tag{10.93}$$

$$E = \begin{pmatrix} 16 & 0 & 6c_1 & 6s_1x_3 \\ 0 & 1 & 0 & 0 \\ 6c_1 & -6s_1x_1 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10.94}$$

The following substitutions have been made to simplify the equations

$$\begin{aligned} J &= \frac{ml^2}{3} \\ \theta_2 &= \alpha_2 - \alpha_1 \\ c_2 &= \cos\theta_2 \\ s_2 &= \sin\theta_2 \end{aligned} \tag{10.95}$$

As discussed in Section 10.8, the components of the E matrix may be related to the inertia, centrifugal and Coriolis matrices of conventional robotic theory (Paul, 1981; Fu *et al.*, 1987; Craig, 1989). Notice that E depends on the relative angle $\alpha_2 - \alpha_1 = \theta$ as well as the angular momenta h_1 and h_2(Gawthrop, 1991).

Linearised about $\alpha_1 = 0$; $\alpha_2 = \theta_0$, the corresponding transfer function is

$$G_{11}(s) = \frac{-1}{+(j(9\cos(\theta_0)^2 - 16))s} \tag{10.96}$$

$$G_{12}(s) = \frac{(3\cos(\theta_0) + 2)}{+(2j(9\cos(\theta_0)^2 - 16))s} \tag{10.97}$$

$$G_{21}(s) = \frac{(3\cos(\theta_0) + 2)}{+(2j(9\cos(\theta_0)^2 - 16))s} \tag{10.98}$$

$$G_{22}(s) = \frac{(-3\cos(\theta_0) - 5)}{+(j(9\cos(\theta_0)^2 - 16))s} \tag{10.99}$$

This depends on the angle θ_0 about which the system is linearised.

The notion of an *inverse* system is much used in robotics, for example in computed torque control (Paul, 1981; Fu *et al.*, 1987; Craig, 1989). The methods of Chapter 6 can be used in this context. The system inputs and outputs are collocated and repesented by the two SS elements. The inverse system is thus given by reversing the causality on these two components as in Figure 10.14. All of the I components now have derivative causality and the two remaining states correspond to the two angles α_1 and α_2.

The corresponding differential-algebraic equation (Section 4.7) is

$$x = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix};\ z = \begin{pmatrix} h_x \\ h_y \\ h_x \\ h_y \end{pmatrix};\ y = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix};\ u = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} \tag{10.100}$$

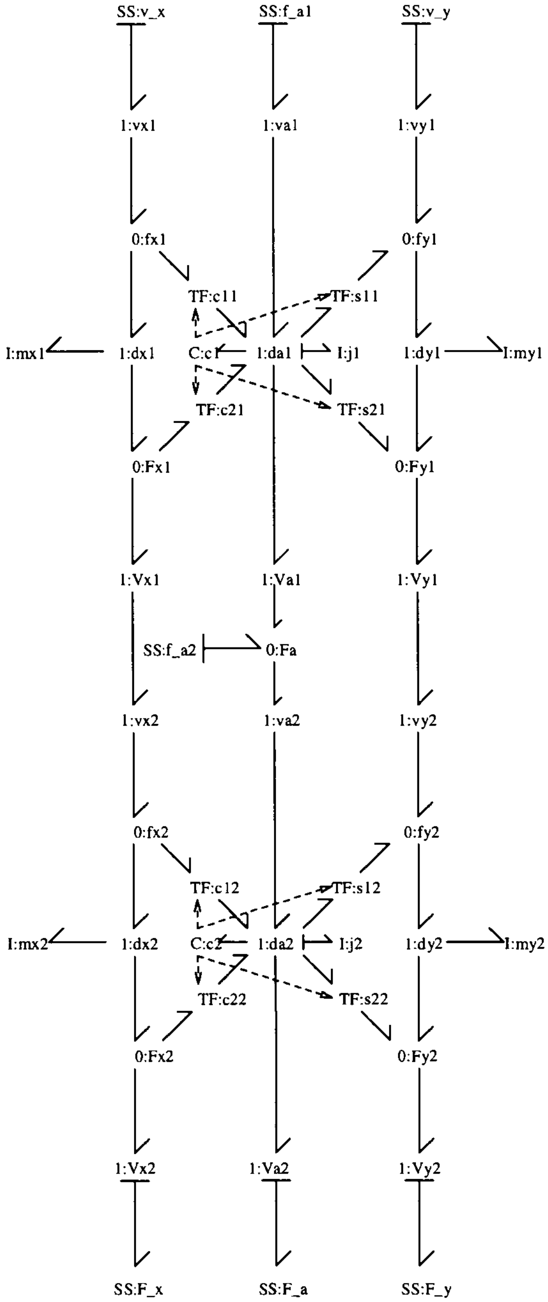


Figure 10.14 A two-link manipulator: bond graph of inverse

$$\dot{x}_1 = u_1$$

$$\dot{x}_2 = u_1 + u_2 \tag{10.101}$$

$$\begin{aligned}
z_1 &= -c_1 lmu_1\\
z_2 &= s_1 lmu_1\\
z_3 &= ju_1\\
z_4 &= -((u_1+u_2)c_2 + 2c_1u_1)lm\\
z_5 &= ((u_1+u_2)s_2 + 2s_1u_1)lm\\
z_6 &= (u_1+u_2)j
\end{aligned} \tag{10.102}$$

$$\begin{aligned}
y_1 &= -((\dot{z}_1 + 2\dot{z}_4)c_1 l - (\dot{z}_2 + 2\dot{z}_5)\sin(x_1)l - s_2 l\dot{z}_5 + c_2 l\dot{z}_4 - \dot{z}_3 - \dot{z}_6)\\
y_2 &= s_2 l\dot{z}_5 - c_2 l\dot{z}_4 + \dot{z}_6
\end{aligned} \tag{10.103}$$

Linearised about $\alpha_1 = 0$; $\alpha_2 = \theta_0$, the corresponding transfer function is

$$G_{11}(s) = -6\sin(\theta_0)ju_2 + (4j(3\cos(\theta_0)+5))s \tag{10.104}$$

$$G_{12}(s) = (-6\sin(\theta_0)j(u_1+u_2)) + (2j(3\cos(\theta_0)+2))s \tag{10.105}$$

$$G_{21}(s) = 6\sin(\theta_0)ju_1 + (2j(3\cos(\theta_0)+2))s \tag{10.106}$$

$$G_{22}(s) = +4js \tag{10.107}$$

This depends on the angle θ_0 about which the system is linearised. This transfer function matrix is of the form

$$G(s) = Js \tag{10.108}$$

where J is a 2×2 inertia matrix. The ijth element of this matrix corresponds to the torque on the ith joint needed to give unit acceleration on the jth joint when the other joint has zero velocity.

- J_{22} corresponds to the inertia of the outer member when the inner member is fixed; this is independent of θ_0 and corresponds to the inertia of a uniform rod of mass m and length l pivoted about its end ($\frac{4}{3}ml^2$).

- J_{11} corresponds to the inertia of both members about the first joint when the second joint is fixed. This is a function of the (fixed) second joint angle θ_0 and varies from a maximum of $8 \times \frac{4}{3}ml^2$ when $\theta_0 = 0$ to a minimum of $2 \times \frac{4}{3}ml^2$ when $\theta_0 = \pi$. The former corresponds to a uniform rod of length $4l$ and mass $2m$ pivoted about one end; the latter corresponds to a uniform rod of length $2l$ and mass $2m$ pivoted about one end.

10.6 THREE-DIMENSIONAL MOTION

This section describes the metamodelling of robots moving in *three* dimensions.

Within this context, two configurations are modelled: a three degree of freedom manipulator with three revolute joints approximating the PUMA, and a three degree of freedom manipulator with two revolute joints and one prismatic joint approximating the Stanford arm.

A paper (Gawthrop, 1991) described how robot equations in the standard form could be derived from bond graph models. This chapter provides a simpler approach. The chapter highlights the following aspects of the metamodelling approach in the context of robotics:

- systematic creation of a three-dimensional manipulator model;
- automatic generation of different derived models including:
  - simulation code,
  - the M, V and G matrices of the conventional robot equations,
  - transfer functions corresponding to dynamic models linearised about arbitary joint angles;
- symbolic, numeric and mixed symbolic/numeric models.

A bond graph gives a causality-free system representation in that, although it implies constaints on the possible causalities of the system components, it does not specify which of the possible causalities is to be used. This flexibility can be used to give alternative causal representations of the same system; the choice between these alternatives depends on the use to which the representation will be put. In this context (robotics) there are three considerations underlying this choice:

1. the need to obtain physical insight into the manipulator dynamics;
2. the need to generate effective simulation code;
3. the need to compare the equations with the standard robot equation (Craig, 1989):

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) = \tau \tag{10.109}$$

In this case, it turns out that the first consideration leads to a causal pattern satisfying the other two considerations. In particular, in the context of manipulators with DC drives, we believe that the correct intuitive view of such a manipulator is summarised as: *a manipulator is a set of DC motors coupled to a mechanism.*

There are a number of reasons for this view:

1. If the motors are coupled to the mechanism via a high-ratio gearbox the mechanism appears to have relatively small inertia as viewed from the motor (Craig, 1989); thus the dynamics are dominated by the motors.
2. It is natural to think of the system state in terms of the motor velocities and positions; in the rigid case, this state determines the position and velocity of each link of the mechanism.

3. From the control point of view, it is nice to view the system as a set of collocated sensors and actuators: the motor angular velocities and drive torques.

4. Control, simulation and understanding of a set of uncoupled motors is easy: the dynamics are simple linear ordinary differential equations corresponding to a set of double integrators. Adding the rest of the system (the mechanism) is thus a (nonlinear) perturbation about this simple case. As well as yielding a clear intuitive insight into robot dynamics, it also gives rise to well-posed simulation code.

10.7 UNCOUPLED MOTORS

As background to the rest of the chapter, consider a set of ideal motors viewed as ideal torque sources driving an inertia. Given the above discussion, each motor has two states associated with it: the ith motor has a state comprising the angle θ_i and the angular velocity $\dot{\theta}_i$. However, from the point of view of systematic system modelling, states should be either integrated effort or integrated flow variables. Thus an alternative state comprises the angle θ_i (an integrated flow) and the angular momentum $h_i = i_i\dot{\theta}_i$ (an integrated effort).

Component label	Component type	Associated physical variable
dt1, dt2, dt3	Common velocity junctions	Motor angular velocities $\dot{\theta}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$
i1, i2 i3	Inertia components	Motor inertias: states h_1, h_2, h_3
c1, c2, c3	Compliance component	Provides the joint angle θ_2
s1, s2, s3	Source-sensors	Torques τ_1, τ_2, τ_3 and velocities ω_1, ω_2, ω_3

Table 10.1 Motors: bond graph labels

Figure 10.15 gives the bond graph of three ideal uncoupled motors regarded as torque sources driving an inertia. The notation appears in Table 10.1. The C elements supply the angles of each motor as a state: as they have zero stiffness they do not affect the dynamics. The corresponding equations are

$$\dot{h} = \tau \tag{10.110}$$

$$\dot{\theta} = I_m^{-1} h \tag{10.111}$$

where I_m is a diagonal matrix and $I_{m_{ii}} = i_i$

10.8 ROBOT-FORM EQUATIONS

Given the view of a robot expressed above (*a manipulator is a set of DC motors coupled to a mechanism*), the next step in deriving a general form of robot equations is to consider the effect of attaching a mechanism to these otherwise uncoupled drives. Because of the

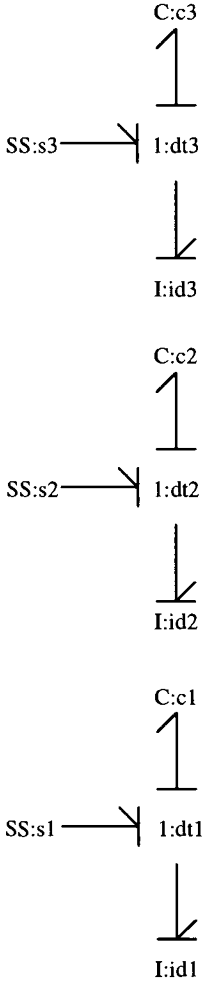


Figure 10.15 Uncoupled motors: Bond graph

assumption of rigidity, the motion of the entire mechanism is completely constrained by the motion of the motors: the position and velocity of each part of the mechanism, given its geometry, is completely specified by the position and velocity of each motor. In other words, the robot dynamics may be described by the same states (motor angles θ_i and angular momenta $h_i = i_i\dot{\theta}_i$) as the system of uncoupled motors. Thus the system state vector x can be written as

$$x = \begin{pmatrix} h \\ \theta \end{pmatrix} \tag{10.112}$$

where h and θ are vectors containing the angular momenta of the n motors and angles of the n joints respectively

$$h = \begin{pmatrix} h_1 \\ .. \\ h_i \\ .. \\ h_n \end{pmatrix} ; \; \theta = \begin{pmatrix} \theta_1 \\ .. \\ \theta_i \\ .. \\ \theta_n \end{pmatrix} \tag{10.113}$$

Defining a vector z containing the momenta associated with all other **I** elements (the *non-state*), it follows from the preceding discussion that

$$z = g(x) \tag{10.114}$$

where $g(x)$ is a (non-linear) function from the state to non-state. Taking derivatives, it follows that

$$\dot{z} = G(x)\dot{x} \tag{10.115}$$

where

$$G(x) = \frac{\partial g(x)}{\partial x} \tag{10.116}$$

As discussed in the subsequent sections and elsewhere (Karnopp *et al.*, 1990) (Gawthrop 1991), the bond graph of the rest of the mechanism comprises **I** elements coupled by transformers modulated by joint angle and gyrators modulated by angular momenta and velocities. Hence, the dynamic equations describing the motor momenta (which were given by Equation 10.110) are of the form

$$\dot{h} = \tau + f_z(x)\dot{z} + f_x(x) \tag{10.117}$$

The equation for the joint angle (Equation 10.111) remains the same.

It follows that the dynamic equations derived here are special cases of the *constrained state* form discussed in Section 4.10.

$$E(x)\dot{x} = f(x) + u \tag{10.118}$$

where

$$E = I_{2n\times 2n} - \begin{pmatrix} f_z \\ 0 \end{pmatrix} G(x) \tag{10.119}$$

Thus the $2n \times 2n$ matrix E is of the special form

$$E = \begin{pmatrix} E_{11} & E_{12} \\ 0 & I_{n\times n} \end{pmatrix} \tag{10.120}$$

This can be rewritten as the standard robot Equation 10.109

$$M(\theta) = E_{11}I_m \tag{10.121}$$

$$V(\theta, \dot{\theta}) = E_{12} - f_x \tag{10.122}$$

Following Craig (1989), the V matrix can be rewritten as

$$V(\theta,\dot{\theta}) = B(\theta)\left[\dot{\theta}\dot{\theta}\right] + C(\theta)\left[\dot{\theta}^2\right] \tag{10.123}$$

where the *Coriolis matrix* $B(\theta)$ $(n \times n(n-1)/2)$ and the *centrifugal matrix* $C(\theta)$ $(n \times n)$ depend only on θ. $\left[\dot{\theta}\dot{\theta}\right]$ is a $n(n-1)/2$ vector of velocity cross products

$$\left[\dot{\theta}\dot{\theta}\right] = \left[\dot{\theta}_1\dot{\theta}_2\ \dot{\theta}_1\dot{\theta}_3 \dots \dot{\theta}_{n-1}\dot{\theta}_n\right]^T \tag{10.124}$$

and $\left[\dot{\theta}^2\right]$ is an $n \times 1$ vector of squared angular velocities

$$\left[\dot{\theta}^2\right] = \left[\dot{\theta}_1^2 \dots \dot{\theta}_n^2\right]^T \tag{10.125}$$

In the simple case considered in Section 10.7, the only non-zero matrix entries are

$$\begin{aligned} M_{11} &= i_1 \\ M_{22} &= i_2 \\ M_{33} &= i_3 \end{aligned} \tag{10.126}$$

The rest of the models in this chapter can be considered as perturbations about this base case.

10.9 GRAVITY EFFECTS

Component label	Component type	Associated physical variable
vx, vy	Common velocity junctions	Velocities v_x, v_y of CM
mx, my	Inertia components	Momenta of mass centre p_x, p_y,
s3	Source-sensor	Force and dispacement of transduce
sg, ig	Source-sensor, inertia	Source and unit inertia emulating gravity

Table 10.2 Including gravity: bond graph labels

Gravity can be included by the simple expedient suggested by Craig (1989): that is, the gravity-free manipulator is supposed to be mounted in a lift accelerating upwards with an acceleration of g. As a simple example, consider the manipulator of Figure 10.16 with a simple prismatic joint (with parameter θ_3) set at a *fixed* angle θ_2 to the horizontal.

In bond graph terms, this can be represented as in Figure 10.17 with the notation of Table 10.2. The source **sg** drives a unit mass **ig** with an acceleration of g. This acceleration is transformed to x and y coordinates by the two transformers **tgx** (with gain $\sin(\theta_2)$) and **tgy** (with gain $\cos(\theta_2)$). The coupling is via signals so that the acceleration of the unit mass is not affected by the rest of the system.

The corresponding differential/algebraic equations are

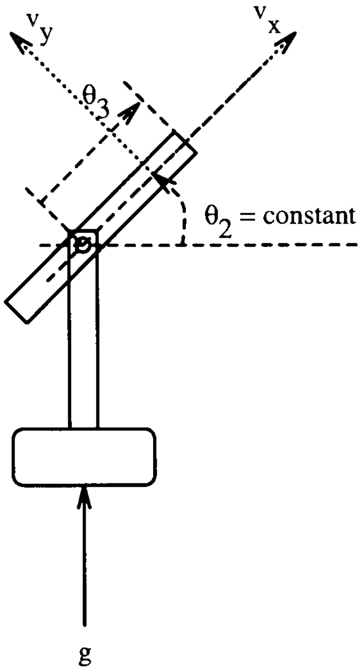


Figure 10.16 Including gravity: schematic

$$\begin{aligned}
\dot{x}_1 &= u_1 - \dot{z}_1 \\
\dot{x}_2 &= \frac{x_1}{i_3} \\
\dot{x}_3 &= u_2
\end{aligned} \tag{10.127}$$

$$\begin{aligned}
z_1 &= \frac{((\sin(\theta_2)i_3x_3 + x_1)m)}{i_3} \\
z_2 &= \cos(\theta_2)mx_3
\end{aligned} \tag{10.128}$$

$$y_1 = \frac{x_1}{i_3} \tag{10.129}$$

In constrained-state form they become

$$\dot{\chi}_1 = u_1 \tag{10.130}$$

$$\dot{\chi}_2 = \frac{x_1}{i_3} \tag{10.131}$$

$$\dot{\chi}_3 = u_2 \tag{10.132}$$

$$y_1 = \frac{x_1}{i_3} \tag{10.133}$$

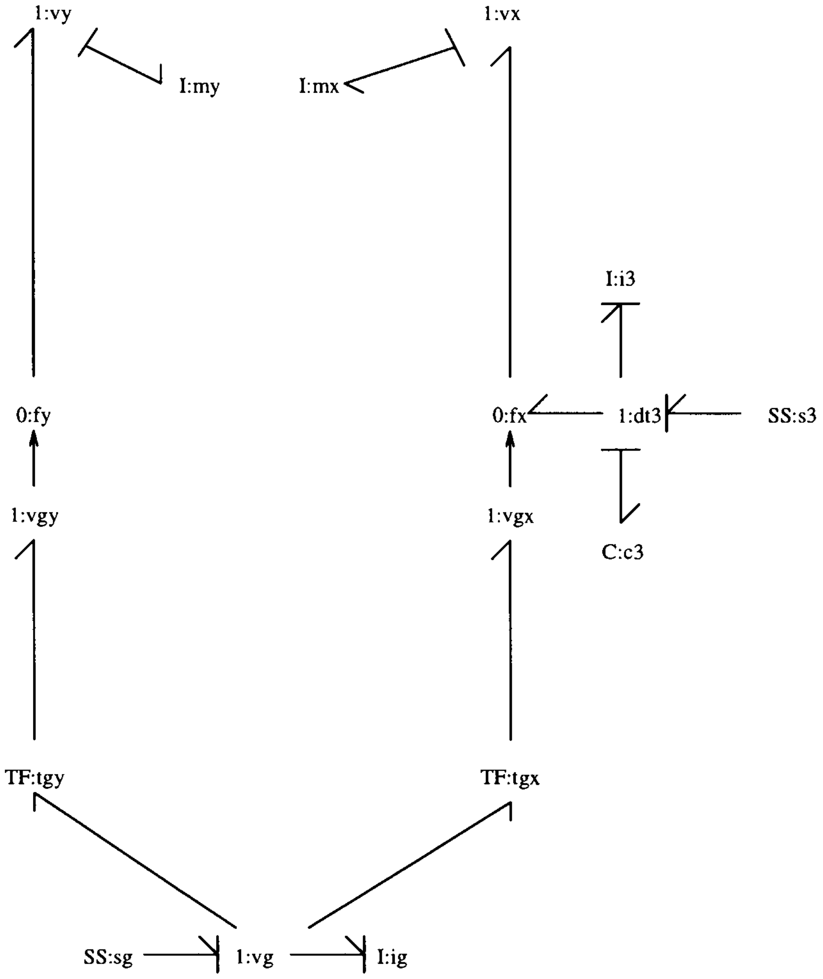


Figure 10.17 Including gravity: Bond graph

$$E = \begin{pmatrix} \frac{(m+i_3)}{i_3} & 0 & \sin(\theta_2)m \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{10.134}$$

where

$$x = \begin{pmatrix} h_3 \\ \theta_3 \\ h_g \end{pmatrix}; \ z = \begin{pmatrix} p_x \\ p_y \end{pmatrix}; \ y = (\, s_3 \,); \ u = \begin{pmatrix} s_3 \\ s_g \end{pmatrix} \tag{10.135}$$

Note that $x_3 = h_g = v_g$, the momentum of the unit mass.
Recalling Equation 10.118,

$$E(x)\dot{x} = f(x) + u \tag{10.136}$$

The first equation contains the term $E_{13}\dot{x}_3 = E_{13}\dot{v}_g = E_{13}g$. Thus E_{13}g is, in this case, the gravity matrix.

In general, following the notation developed previously, the state vector now becomes

$$x = \begin{pmatrix} h \\ \theta \\ v_g \end{pmatrix} \tag{10.137}$$

and thus

$$\dot{x} = \begin{pmatrix} \dot{h} \\ \dot{\theta} \\ g \end{pmatrix} \tag{10.138}$$

Hence E gains an additional row and column and can be repartitioned as

$$E = \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ 0 & I_{n\times n} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{10.139}$$

Thus

$$G(\theta) = E_{13}g \tag{10.140}$$

Returning to this example, then, the robot matrices are

$$M_{11} \quad = \quad m + i_3 \tag{10.141}$$

$$G_1 \quad = \quad \sin(\theta_2)m \tag{10.142}$$

10.10 THREE-DIMENSIONAL MOTION AND EULER'S EQUATIONS

The seminal paper on bond graph representation of three-dimensional rigid body mechanics is that of Karnopp (Karnopp, 1969); a more accessible account appears in a recent textbook (Karnopp *et al.*, 1990). The main idea is that Euler's equations can be represented by a triangular bond graph structure.

As discussed in the standard text books, the simplest version of Euler's equations occur when rigid body motion occurs about a *fixed* point. As discussed elsewhere (Karnopp, 1969; Karnopp *et al.*, 1990), this simple case can be represented by the bond graph of Figure 10.18. The main features of the bond graph appear in Table 10.3.

ω_x, ω_y and ω_z are the three components of the ***absolute angular*** velocity referred to the instantaneous body axes. The fact that this is a *moving* coordinate system leads to the coupling between the three dynamic equations. The three dotted arrows indicate that the gain of the gyrator at each tip is the angular momentum at each tail.

The Euler ring has been augmented with three source-sensors. With the causality shown in Figure 10.18, the *effort* (torque) sources are inputs (called u_1 – u_3) and the corresponding velocities regarded as outputs (called y_1 – y_3).

The corresponding dymamic equations are

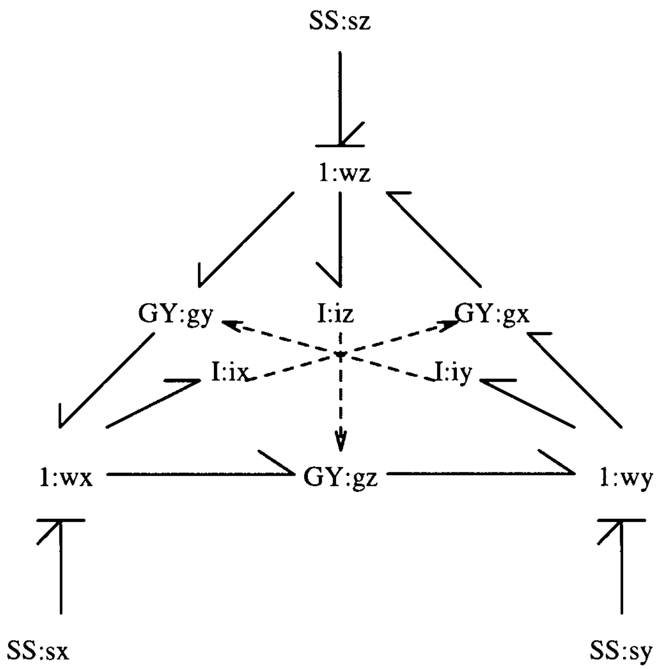


Figure 10.18 Euler ring: Bond graph

Component label	Component type	Associated physical variable
wx, wy, wz	Common velocity junctions	Angular velocities ω_x, ω_y, ω_z
ix, iy, iz	Inertia components	Angular momenta h_x, h_y, h_z,
gx, gy, gz	Gyrators	Coupling due to rotation
sx, sy, sz	Source-sensors	Torques τ_x, τ_y, τ_z, velocities ω_x, ω_y, ω_z

Table 10.3 Euler ring: bond graph labels

$$\begin{aligned}\dot{x}_1 &= \frac{(-((j_z - j_y)x_2x_3 - j_zj_yu_1))}{(j_zj_y)}\\ \dot{x}_2 &= \frac{((j_z - j_x)x_1x_3 + j_zj_xu_2)}{(j_zj_x)}\\ \dot{x}_3 &= \frac{(-((j_y - j_x)x_1x_2 - j_yj_xu_3))}{(j_yj_x)}\end{aligned} \tag{10.143}$$

$$\begin{aligned}y_1 &= \frac{x_1}{j_x}\\ y_2 &= \frac{x_2}{j_y}\\ y_3 &= \frac{x_3}{j_z}\end{aligned} \tag{10.144}$$

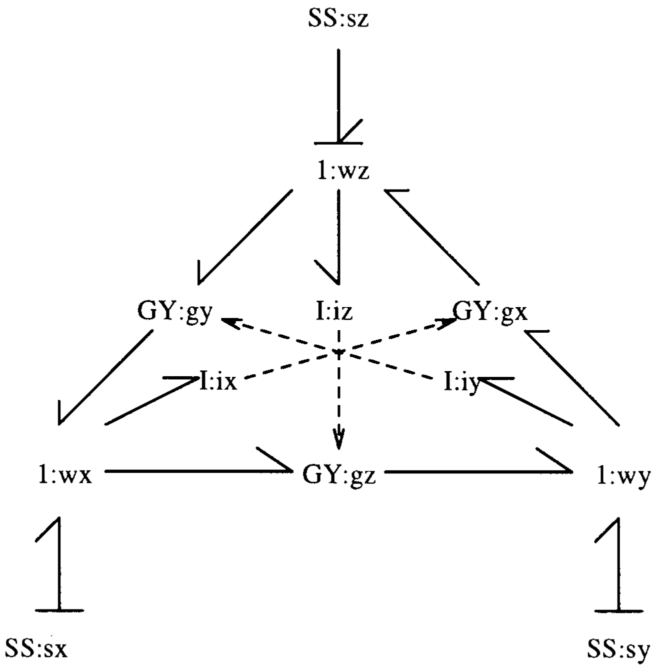


Figure 10.19 Euler ring - alternative causality: bond graph

In practice, a rigid body is normally connected to other objects as part of a system: thus the three angular motions are typically *constrained* and these equations would not have this causality. For example, Figure 10.19 shows another causal possibility: the angular velocities ω_x, ω_y are fixed (driven by velocity sources) whilst the z-axis is driven as before. The inputs and outputs are now

$$u = \begin{pmatrix} \omega_x \\ \omega_y \\ \tau_z \end{pmatrix} y = \begin{pmatrix} \tau_x \\ \tau_y \\ \omega_z \end{pmatrix} \tag{10.145}$$

The resultant equations are

$$\dot{x}_1 = (j_x - j_y)u_1u_2 + u_3 \tag{10.146}$$

$$\begin{aligned} z_1 &= j_x u_1 \\ z_2 &= j_y u_2 \end{aligned} \tag{10.147}$$

$$\begin{aligned} y_1 &= \frac{(-((j_y - j_z)x_1u_2 - j_z\dot{z}_1))}{j_z} \\ y_2 &= \frac{((j_x - j_z)x_1u_1 + j_z\dot{z}_2)}{j_z} \end{aligned}$$

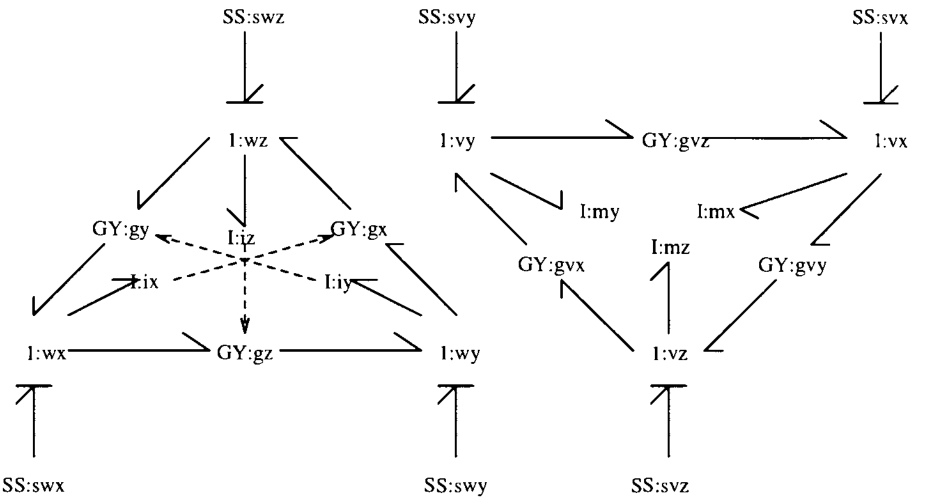


Figure 10.20 Euler ring - translational motion: bond graph

$$y_3 = \frac{x_1}{j_z} \tag{10.148}$$

The more complex case occurs when motion is considered with respect to the (non-stationary) centre of mass. The resultant bond graph (Karnopp *et al.*, 1990) is given in Figure 10.20. The Euler ring has been augmented with three *effort* (force) sources (inputs) u_1 – u_3 and the corresponding velocities regarded as outputs y_1 – y_3. The corresponding dymamic equations are

$$\begin{aligned}
\dot{x}_1 &= \frac{(-((j_z - j_y)x_2x_3 - j_zj_yu_1))}{(j_zj_y)} \\
\dot{x}_2 &= \frac{((j_z - j_x)x_1x_3 + j_zj_xu_2)}{(j_zj_x)} \\
\dot{x}_3 &= \frac{(-((j_y - j_x)x_1x_2 - j_yj_xu_3))}{(j_yj_x)} \\
\dot{x}_4 &= \frac{((j_yu_4 - x_2x_6)j_z + j_yx_3x_5)}{(j_zj_y)} \\
\dot{x}_5 &= \frac{((j_xu_5 + x_1x_6)j_z - j_xx_3x_4)}{(j_zj_x)} \\
\dot{x}_6 &= \frac{((j_xu_6 - x_1x_5)j_y + j_xx_2x_4)}{(j_yj_x)}
\end{aligned} \tag{10.149}$$

$$\begin{aligned}
y_1 &= \frac{x_1}{j_x} \\
y_2 &= \frac{x_2}{j_y} \\
y_3 &= \frac{x_3}{j_z}
\end{aligned}$$

$$\begin{aligned} y_4 &= \frac{x_4}{m} \\ y_5 &= \frac{x_5}{m} \\ y_6 &= \frac{x_6}{m} \end{aligned} \tag{10.150}$$

Once again, when embedded in a manipulator bond graph, the bond graph fragment of Figure 10.20 normally has a different causality imposed upon it due to the constraint of being attached to other components.

10.11 MODELLING A TWO-DEGREES-OF-FREEDOM PUMA

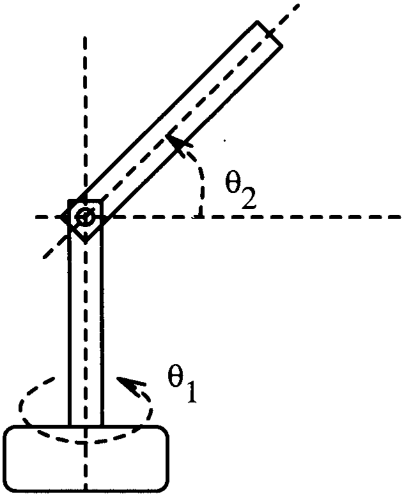


Figure 10.21 2 DOF PUMA: schematic

A simple two-degrees-of-freedom (but three-dimensional) manipulator appears in Figure 10.21. This can be regarded as a simplified PUMA with the elbow and wrist locked at appropriate angles and zero joint offset. It is also similar to the example of Karnopp (Karnopp, 1969) also discussed by Hubbard (Anex and Hubbard, 1984).

The second link, although moving in three dimensions, rotates about a *fixed* point: joint 2. Its dynamics are therefore determined by the Euler ring of Figures 10.18 and 10.19. The first link is a simple one-dimensional rotating inertia coupled to the second link by a joint. The angular velocities of the second link about the x and y axes ω_x and ω_y are entirely determined by that of the first link ω_1

$$\omega_x = \sin\theta_2\omega_1 \tag{10.151}$$

$$\omega_y = \cos\theta_2\omega_1 \tag{10.152}$$

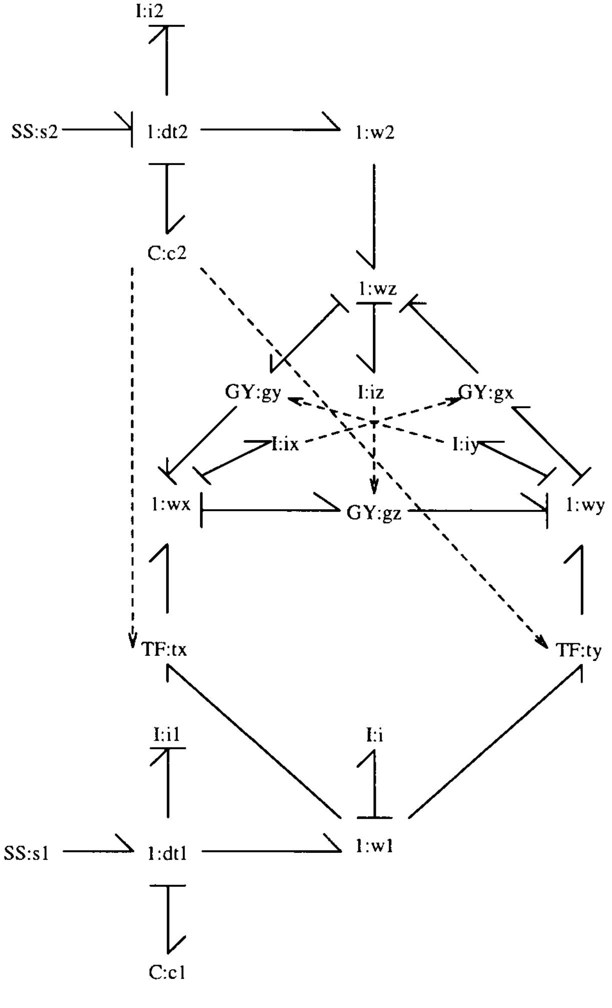


Figure 10.22 2 DOF PUMA: Bond graph

Hence the appropriate causality is that displayed in Figure 10.19. Notice that gravity *cannot* be included in this example using the method of Section 10.9 as the joint is assumed to be stationary.

The corresponding bond graph appears as Figure 10.22 with notation in Table 10.3.

The equations resulting from this Bond-Graph can be presented in many forms. Fistly, the equations can be derived in *differential-algebraic* form: a set of *differential* equations in the state x (Equation 10.112) and a set of *algebraic* equations giving the nonstate z in terms of the state x (Equation 10.114). These equations are

$$\dot{x}_1 = \frac{(-(((j_x - j_y)\cos(x_4)x_1x_3 + i_2i_1\dot{z}_2)\sin(x_4) - (u_1 - \dot{z}_1)i_2i_1 + \cos(x_4)i_2i_1\dot{z}_3))}{(i_2i_1)}$$

Label	Component type	Associated physical variable
dt1, dt2	Common velocity junctions	Joint angular velocities $\dot{\theta}_1$, $\dot{\theta}_2$
wx, wy, wz	Common velocity junctions	Angular velocities ω_x, ω_y, ω_z
i1	Inertia component	Inertia of link 1
ix, iy, iz	Inertia components	Principle angular momenta h_x, h_y, h_z,
gx, gy, gz	Gyrators	Coupling due to rotating coordinate system
s1, s2	Source-sensors	Torques τ_1, τ_2; Angular velocities $\dot{\theta}_1$, $\dot{\theta}_2$
tx,ty	Transformers	Transformations: eqns 10.151, 10.152
c	Compliance component	Provides the joint angle θ_2

Table 10.4 2 DOF PUMA: bond graph labels

$$\begin{aligned}\dot{x}_2 &= \frac{x_1}{i_1}\\ \dot{x}_3 &= \frac{((j_x - j_y)\sin(x_4)\cos(x_4)x_1^2 + (u_2 - \dot{z}_4)i_1^2)}{i_1^2}\\ \dot{x}_4 &= \frac{x_3}{i_2}\end{aligned} \tag{10.153}$$

$$\begin{aligned}z_1 &= \frac{(j_1 x_1)}{i_1}\\ z_2 &= \frac{(\sin(x_4)j_x x_1)}{i_1}\\ z_3 &= \frac{(\cos(x_4)j_y x_1)}{i_1}\\ z_4 &= \frac{(j_z x_3)}{i_2}\end{aligned} \tag{10.154}$$

$$\begin{aligned}y_1 &= \frac{x_1}{i_1}\\ y_2 &= \frac{x_3}{i_2}\end{aligned} \tag{10.155}$$

These differential-algebraic equations may be rewritten in *constrained-state* form (Equation 10.118 as

$$\dot{\chi}_1 = \frac{(-((j_x - j_y)\sin(x_4)\cos(x_4)x_1x_3 - i_2i_1u_1))}{(i_2i_1)} \tag{10.156}$$

$$\dot{\chi}_2 = \frac{x_1}{i_1} \tag{10.157}$$

$$\dot{\chi}_3 = \frac{((j_x - j_y)\sin(x_4)\cos(x_4)x_1^2 + i_1^2u_2)}{i_1^2} \tag{10.158}$$

$$\dot{\chi}_4 = \frac{x_3}{i_2} \tag{10.159}$$

$$y_1 = \frac{x_1}{i_1} \tag{10.160}$$

$$y_2 = \frac{x_3}{i_2} \tag{10.161}$$

$$E = \begin{pmatrix} \frac{(\sin(x_4)^2 j_x + \cos(x_4)^2 j_y + i_1 + j_1)}{i_1} & 0 & 0 & \frac{((j_x - j_y)\sin(x_4)\cos(x_4)x_1)}{i_1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{(i_2 + j_z)}{i_2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10.162}$$

Reordering the equations and using equations 10.121, 10.122 and 10.123 then gives the robot matrices as

$$M = \begin{pmatrix} \sin(\theta_2)^2 j_x + \cos(\theta_2)^2 j_y + i_1 + j_1 & 0 \\ 0 & i_2 + j_z \end{pmatrix} \tag{10.163}$$

$$C = \begin{pmatrix} 0 & 0 \\ -(j_x - j_y)\sin(\theta_2)\cos(\theta_2) & 0 \end{pmatrix} \tag{10.164}$$

$$B = \begin{pmatrix} 2(j_x - j_y)\sin(\theta_2)\cos(\theta_2) \\ 0 \end{pmatrix} \tag{10.165}$$

The mass matrix M has two non-zero terms. M_{11} is the inertia of motor 1 (i_i) + the inertia of link 1 about its axis (j_1) + two terms representing the effect of link 2 dependent on θ_2. The interpretation is clear when $\theta_2 = 0$ or $\theta_2 = \frac{\pi}{2}$. M_{22} is the inertia of motor 2 (i_2 + the inertia of link 2 about the joint 2 axis (j_z).

The centrifugal matrix C has one non-zero element C_{21} representing the additional torque acting at joint 2 due to the angular velocity of joint 1 $\dot{\theta}_1$. This is zero when either $\theta_2 = 0$ or $\theta_2 = \frac{\pi}{2}$. The coriolis matrix B has one non-zero element B_{11} representing the additional torque acting at joint 1 due to both angular velocities. This is again zero when either $\theta_2 = 0$ or $\theta_2 = \frac{\pi}{2}$.

10.12 MODELLING A STANFORD ARM

The Stanford arm (see, for example the description by Wolovich (1987)) is depicted in Figure 10.23. It has one additional degree of freedom compared to model 1; there is a translational joint added. From the dynamic point of view, this means that the simple Euler ring cannot be used as the second link no longer has a fixed point. So Figure 10.24 contains the more complex structure of Figure 10.20. In addition to the two rotational joints with angles θ_1 and θ_2 there is a *prismatic* joint with displacement θ_3.

In addition to the transformers **tx** and **ty**, the transformers **txy** and **tyx** are also modulated, but this time by the prismatic joint displacement θ_3.

Gravity is included as discussed in Section 10.9

The corresponding robot matrices are

$$M = \begin{pmatrix} (m\theta_3^2 + j_y)\cos(\theta_2)^2 + \sin(\theta_2)^2 j_x + i_1 + j_1 & 0 & 0 \\ 0 & m\theta_3^2 + i_2 + j_z & 0 \\ 0 & 0 & m + i_3 \end{pmatrix} \tag{10.166}$$

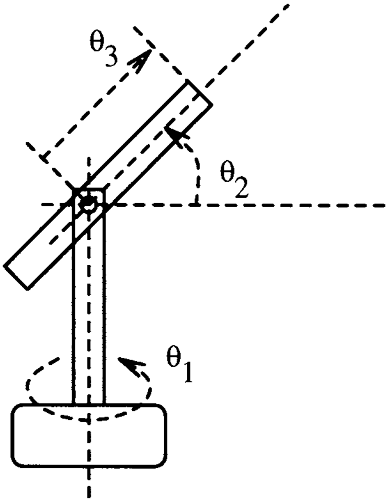


Figure 10.23 Stanford arm: schematic

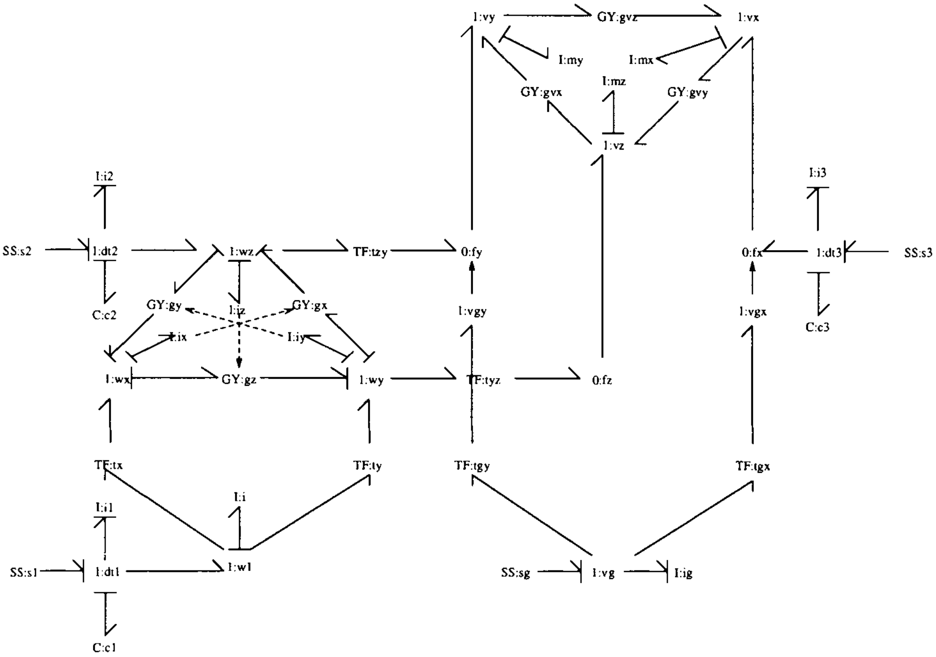


Figure 10.24 Stanford arm: Bond graph

$$C = \begin{pmatrix} 0 & 0 & 0 \\ -(m\theta_3^2 + j_x - j_y)\sin(\theta_2)\cos(\theta_2) & 0 & 0 \\ \cos(\theta_2)^2 m\theta_3 & -m\theta_3 & 0 \end{pmatrix} \tag{10.167}$$

$$B = \begin{pmatrix} 2(j_x - j_y)\sin(\theta_2)\cos(\theta_2) & \cos(\theta_2)^2 m\theta_3 & 0 \\ 0 & 0 & m\theta_3 \\ 0 & 0 & 0 \end{pmatrix} \tag{10.168}$$

$$G = \begin{pmatrix} 0 \\ \cos(\theta_2) m\theta_3 \\ \sin(\theta_2) m \end{pmatrix} \tag{10.169}$$

The first two elements M_{11} and M_{22} are similar to that of Section 10.11. The inertia is, however, with respect to the mass-centre and so the term $m\theta_3^2$ appears to augment j_y. The third element M_{33} is just the sum of the mass and the prismatic drive of link 3.

Gravity does not affect joint 1, so $G_1 = 0$. The second term G_2 gives the moment about joint 2 due to gravity and the third term corresponds to the example of Figure 10.16.

10.13 MODELLING A THREE-DEGREES-OF-FREEDOM PUMA

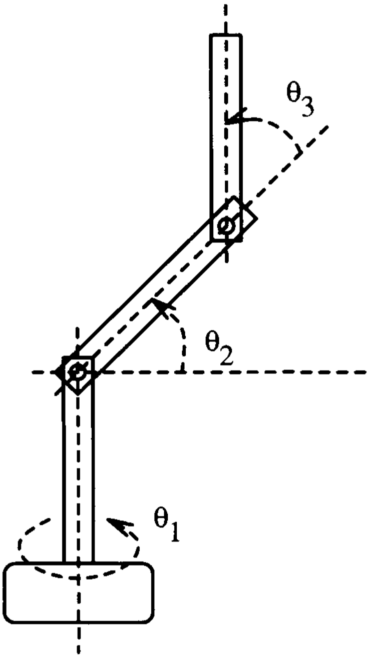


Figure 10.25 3 DOF puma: schematic

An approximation to a three-degrees-of-freedom PUMA appears in Figure 10.25, and the corresponding bond graph in Figure 10.26. The lower part of the diagram is similar to that in Figure 10.24 (but without the prismatic joint); the upper part contains the double

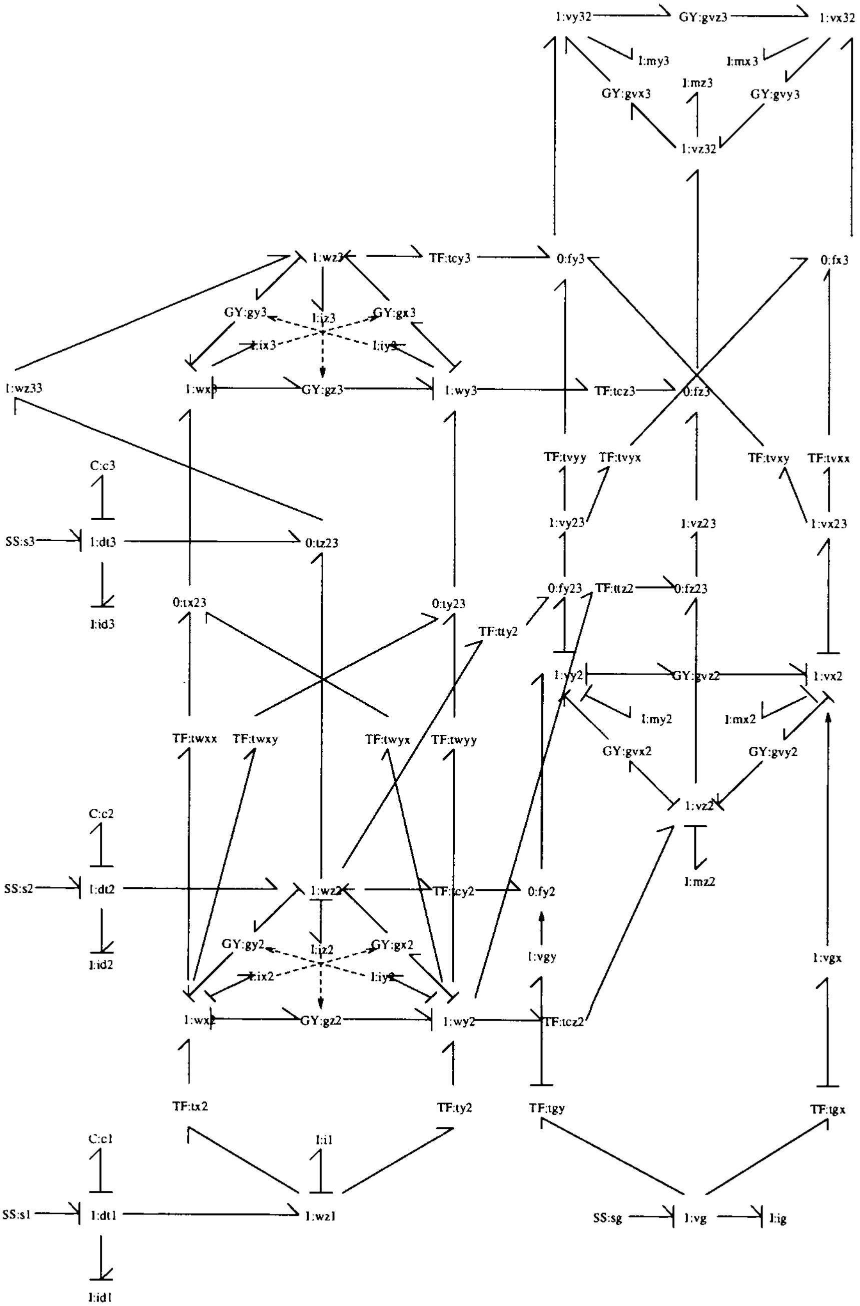


Figure 10.26 3 DOF puma: Bond graph

Euler ring structure of Figure 10.20 connected to the lower part by a set of transformers

Component	Value
r_2	0.068
r_3	0.143
j_1	0.35
j_2	0.53
j_3	0.07
i_1	1.14
i_2	4.71
i_3	0.83
m_2	17.4
m_3	6.04
a_2	0.4318
a_3	0.4331

Table 10.5 PUMA: numerical values

corresponding to the coordinate transformation occurring between links 2 and 3.

Numerical values due to Armstrong *et al.* (1986) are given in Table 10.5 and were substituted for the symbolic values. The corresponding robot matrices are

$$M = \begin{pmatrix} 0.194c_{23}^2 + 0.747c_{23} + 3.24 & 0 & 0 \\ 0 & 0.745c_3 + 6.65 & 0.374c_3 + 0.194 \\ 0 & 0.374c_3 + 0.194 & 1.03 \end{pmatrix} \tag{10.170}$$

$$C = \begin{pmatrix} 0 & 0 & 0 \\ ((0.373c_{23}^2 + 1.07c_{23} - 0.373) - (0.373c_{23} + 1.13)c_3)s_3 & 0 & 0 \\ -(0.0536c_{23} + 0.373)s_3 & 0.374s_3 & 0 \end{pmatrix} \tag{10.171}$$

$$B = \begin{pmatrix} b_{11} & -(0.194c_{23} + 0.373)s_{23} & 0 \\ 0 & 0 & -0.374s_3 \\ 0 & 0 & -0.374s_3 \end{pmatrix} \tag{10.172}$$

where

$$b_{11} = -((1.01c_{23} - 0.374)s_3 - 1.06c_3s_3 + 0.194c_{23}s_{23} + 0.373s_{23}) \tag{10.173}$$

$$G = \begin{pmatrix} 0 \\ 0.864c_{23} + 3.79 \\ 0.864c_{23} \end{pmatrix} \tag{10.174}$$

Because the full equations are somewhat complicated, the equations are displayed for the special case

$$\theta_2 = 0 \tag{10.175}$$

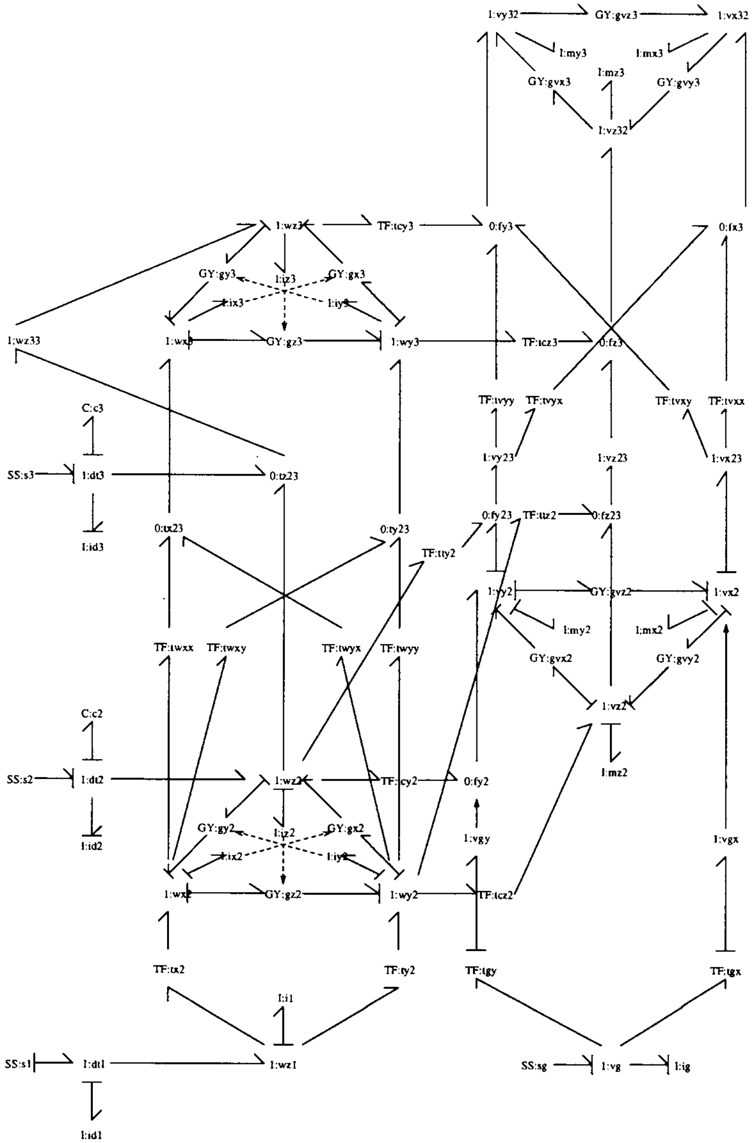


Figure 10.27 Craig's example: bond graph

10.13.1 Checking the models

One way to check these models is to compare results with those in the textbooks. This is done for three simplifications of these models.

If joint one of the three-degrees-of-freedom three-dimensional PUMA of Section 10.13 is locked, then the remainder of the mechanism becomes a planar two-degrees-of-freedom manipulator. This is studied in the standard textbooks (Craig, 1989; Fu *et al.*, 1987). Craig (1989) studies such a configuration where the mass of each link is concentrated at the distal end whereas Fu *et al.* (1987) consider two uniform links with the same length but different masses.

A joint can be locked by changing the causality of the corresponding source/sensor element to give a *velocity source*, and the corresponding velocity is then set to zero. This has been done in Figure 10.27. Using parameters corresponding to Craig (1989) gives

$$M = \begin{pmatrix} (m_3 + m_2)l_2^2 + 2\cos(\theta_2)l_2l_3m_3 + l_3^2m_3 + i_2 & (\cos(\theta_2)l_2 + l_3)l_3m_3 \\ (\cos(\theta_2)l_2 + l_3)l_3m_3 & l_3^2m_3 + i_3 \end{pmatrix} \tag{10.176}$$

$$C = \begin{pmatrix} 0 & -\sin(\theta_2)l_2l_3m_3 \\ \sin(\theta_2)l_2l_3m_3 & 0 \end{pmatrix} \tag{10.177}$$

$$B = \begin{pmatrix} -2\sin(\theta_2)l_2l_3m_3 \\ 0 \end{pmatrix} \tag{10.178}$$

$$G = \begin{pmatrix} (m_3 + m_2)\cos(\theta_1)l_2 + \cos(\theta_2 + \theta_1)l_3m_3 \\ \cos(\theta_2 + \theta_1)l_3m_3 \end{pmatrix} \tag{10.179}$$

and using parameters corresponding to Fu *et al.* (1987) gives

$$M = \begin{pmatrix} (m_3 + m_2)l_2^2 + 2cos(\theta_2)l_2l_3m_3 + l_3^2m_3 + i_2 & (cos(\theta_2)l_2 + l_3)l_3m_3 \\ (cos(\theta_2)l_2 + l_3)l_3m_3 & l_3^2m_3 + i_3 \end{pmatrix} \tag{10.180}$$

$$C = \begin{pmatrix} 0 & -sin(\theta_2)l_2l_3m_3 \\ sin(\theta_2)l_2l_3m_3 & 0 \end{pmatrix} \tag{10.181}$$

$$B = \begin{pmatrix} -2sin(\theta_2)l_2l_3m_3 \\ 0 \end{pmatrix} \tag{10.182}$$

$$G = \begin{pmatrix} (m_3 + m_2)cos(\theta_1)l_2 + cos(\theta_2 + \theta_1)l_3m_3 \\ cos(\theta_2 + \theta_1)l_3m_3 \end{pmatrix} \tag{10.183}$$

These correspond to the published results.

If joint one of the three-degrees-of-freedom three-dimensional Stanford arm of Section 10.12 is locked, then the remainder of the mechanism one again becomes a planar two-degrees-of-freedom manipulator but with one prismatic and one revolute joint. This has been done in Figure 10.28.

Using parameters corresponding to Craig (1989) gives

$$M = \begin{pmatrix} m_2\theta_2^2 + i_2 + j_z & 0 \\ 0 & m_2 + i_3 \end{pmatrix} \tag{10.184}$$

$$C = \begin{pmatrix} 0 & 0 \\ -m_2\theta_2 & 0 \end{pmatrix} \tag{10.185}$$

$$B = \begin{pmatrix} 2m_2\theta_2 \\ 0 \end{pmatrix} \tag{10.186}$$

Once again, these correspond to the published results.

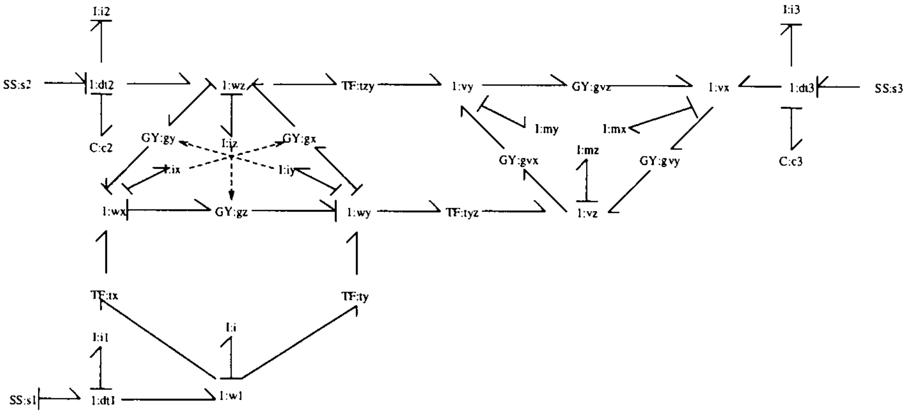


Figure 10.28 Craig's RP example: Bond graph

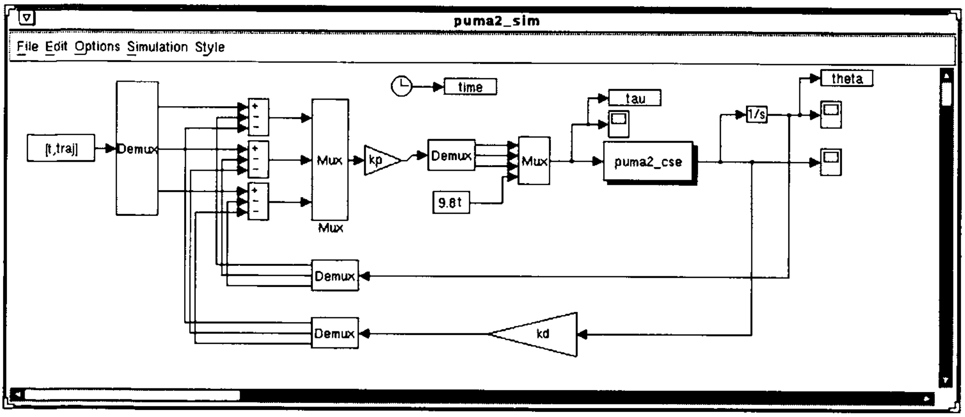


Figure 10.29 Simulation diagram

10.13.2 Linearised system equations

For the purposes of control design, a linearised model is useful. This has been done for the PUMA model of Section 10.13 using the *numerical* parameters of Table 10.5 about the *symbolic* joint angles θ_1, θ_2 and θ_3. The corresponding transfer function is

$$G_{11}(s) = \frac{5.17}{+(c_{23}^2 + 3.86c_{23} + 16.7)s} \tag{10.187}$$

$$G_{12}(s) = 0 \tag{10.188}$$

$$G_{13}(s) = 0 \tag{10.189}$$

$$G_{14}(s) = 0 \tag{10.190}$$

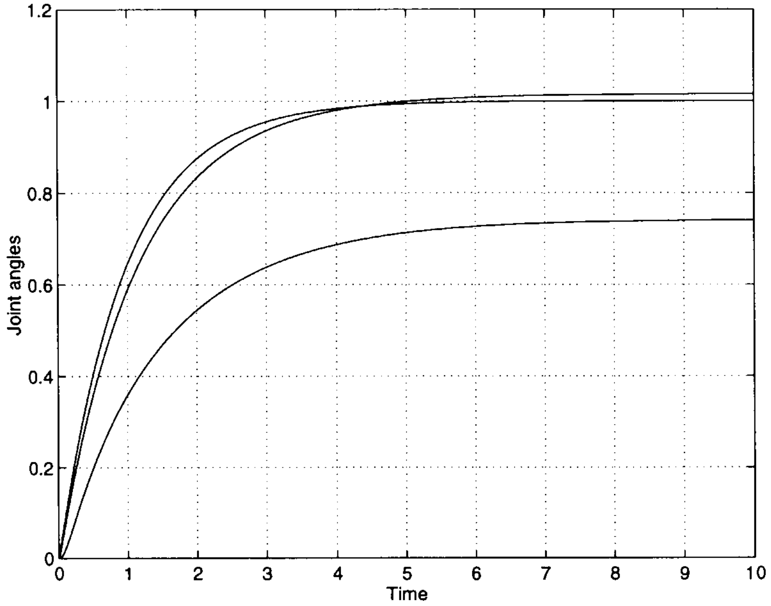


Figure 10.30 Joint angles

$$G_{21}(s) = 0 \tag{10.191}$$

$$G_{22}(s) = \frac{1.01}{+(c_3^2 + 0.61c_3 + 1.14s_3^2 + 5.53)s} \tag{10.192}$$

$$G_{23}(s) = \frac{(-0.367c_3 - 0.19)}{+(c_3^2 + 0.61c_3 + 1.14s_3^2 + 5.53)s} \tag{10.193}$$

$$G_{24}(s) = \frac{(-2.31c_3c_{23} - 2.63s_3^2 - 0.705c_{23} - 1.19)}{+(c_3^2 + 0.61c_3 + 1.14s_3^2 + 5.53)s} \tag{10.194}$$

$$G_{31}(s) = 0 \tag{10.195}$$

$$G_{32}(s) = \frac{(-0.366c_3 - 0.19)}{+(c_3^2 + 0.61c_3 + 1.14s_3^2 + 5.53)s} \tag{10.196}$$

$$G_{33}(s) = \frac{(1.11c_3^2 + 0.732c_3 + 1.11s_3^2 + 5.41)}{+(c_3^2 + 0.61c_3 + 1.14s_3^2 + 5.53)s} \tag{10.197}$$

G_{34} is too complex to be displayed here.

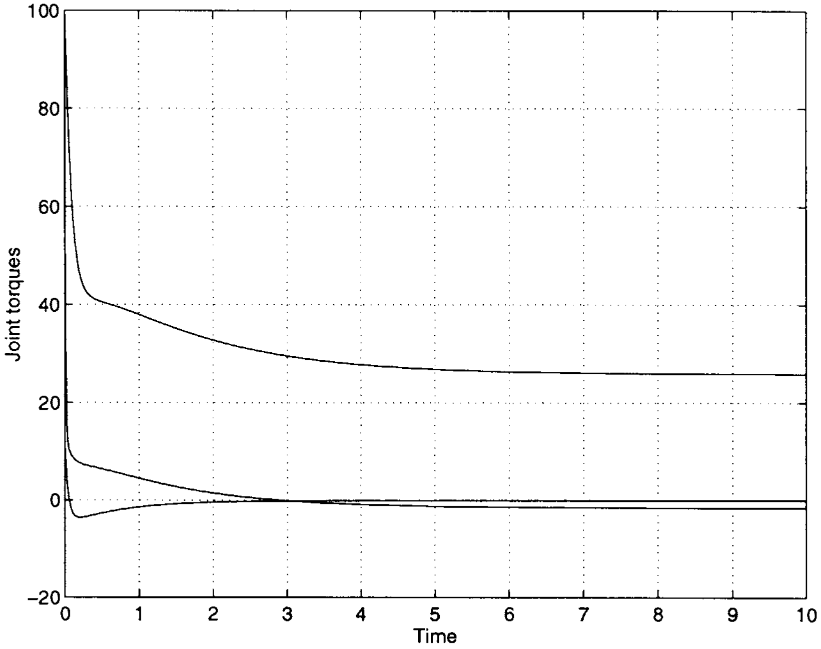


Figure 10.31 Torques

10.13.3 Simulation

Simulation code suitable for Simulab was automatically generated for the three-degrees-of-freedom PUMA model. An elementary simulation was created as in Figure 10.29. A step trajectory was applied, $k_p = 100$ and $k_d = 1$. The output angle θ and applied torque τ are plotted in Figures 10.30 and 10.31.

11

Control systems

SUMMARY

- The application of bond graphs to model-based control is discussed.
- An illustrative example is given.

11.1 INTRODUCTION

Control has always been concerned with generic techniques, that is techniques which can be applied across a range of physical domains.

The design of controllers, adaptive or non-adaptive, for linear systems requires a *representation* of the system to be controlled. For example, Bode and Nichols diagram designs requires a frequency-response representation; root-locus designs requires a pole-zero representation; observer/state-feedback designs require a state-space representation; polynomial designs require a transfer-function representation.

These representations are *generic* in the sense that they can represent (linearised) systems drawn from a range of physical domains including: mechanical, electrical, hydraulic and thermodynamic, and this is their strength. But, at the same time, these representations suffer from being ***abstractions*** of physical systems: the very process of abstracting the generic features of physical systems means that system-specific physical details are lost. Both the parameters and states of such representations are not easily related back to the original physical parameters of the system.

This loss of system-specific physical detail is, perhaps, acceptable at the two extremes of knowledge about the system: the system parameters are completely known, or the system parameters are entirely unknown. In the former case, the system can be translated into the representations mentioned above, and the physical system knowledge is translated into, for example, transfer function parameters. In the latter case, the system can be deemed to have one of the representations mentioned above and there is no physical system knowledge to be translated.

However, in the case of *partially-known* systems the situation is not satisfactory. None of the above representations is particularly suited to including *partial* physical system knowledge and, in doing so, representations become problem specific, not generic.

Thus much of the current body of control achieves a generic coverage of application areas by having a generic representation of the systems to be controlled which are, however, not well suited to partially-known systems. This situation is unsatisfactory because for many dynamic systems there is a wealth of quantitative information available.

This chapter proposes using an alternative approach which, whilst achieving a generic coverage of application areas, allows the use of particular representations for particular (possibly partially-known, possibly non-linear) systems. Instead of having a generic representation of systems, we propose a generic method for *automatically deriving* system-specific representations: this is based on metamodelling via bond graphs.

In this chapter we will illustrate the use of the bond graph representation by introducing one aspect of our approach to model-based control namely *model-based observer control.* The basic ideas behind this are demonstrated in the application to a process engineering example, namely the level control of three coupled tanks using inferential measurement. The motivation for this approach comes from three areas: linear state-space observer theory (Kwakernaak and Sivan, 1972; Kailath, 1980), inferential control (Joseph *et al.*, 1978) (Parrish and Brosilow, 1985), and the application of bond graphs to observer design (Karnopp, 1979).

The chapter is organised as follows.

- Section 11.2 introduces model-based observers and discusses how they can be used for model-based observer control.
- Section 11.3 gives the illustrative non-linear example from the field of process engineering.
- Section 11.4 continues the example and shows how to augment the observer to include unknown disturbances. and
- Section 11.5 concludes the chapter.

The chapter is based on a conference paper by Gawthrop *et al.* (1992).

11.2 MODEL-BASED OBSERVER (MBO) CONTROL

The general scheme of model-based observer control is outlined in Figure 11.1. The different aspects of the approach are outlined below.

- The block labelled 'model' is a dynamic simulation model of the system to be controlled (labelled 'system'). It will usually be non-linear.
- The model has the same control input u as the system. Other system inputs which can be measured may also be applied to the model in the same way.
- The measured outputs y of the system are compared with the corresponding model outputs $\hat{y}$ to create an error e.
- Additional inputs are provided to this model, one for each state. In the case of process systems, these inputs are typically flows of mass or energy.

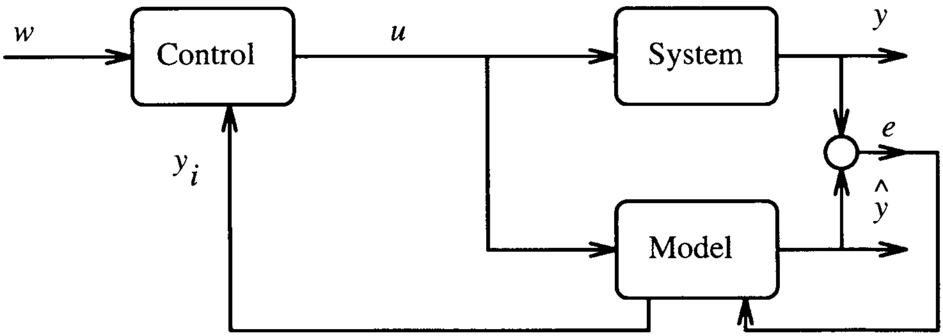


Figure 11.1 Model-based observer (MBO) control

- The model has *feedback* applied to these additional inputs in such a way as to drive the error e to zero.
- The controller generates the control signal u in terms of any internal signals generated within the model together with the setpoint.
- If the model-based observer is working well, these internal signals will be the same or close to the corresponding (but possibly unmeasurable) internal signals generated within the system itself.

The above desciptions are non-specific with regard to how individual objectives are achieved. There is the freedom to integrate within the structure any theory which will satisfy the control objectives. The process engineering example shows one particular implementation.

11.3 A NON-LINEAR EXAMPLE: THREE COUPLED TANKS

The MBO approach to model-based control is demonstrated by considering its application to control of liquid level in three coupled tanks. The level in the middle tank has to be regulated using only the measurement of the level in the third tank and by manipulating the flowrate into the first tank. In this particular case, level is numerically equal to pressure, so that pressure and level control are, in this context, the same.

11.3.1 The system

The system shown in Figure 5.4 consists of three uniform tanks of incompressible liquid connected by pipes. Neglecting inertia effects, the pipes are modelled as pure, but non-linear, resistances

$$f_i = k_i\sqrt{\Delta p_i} \tag{11.1}$$

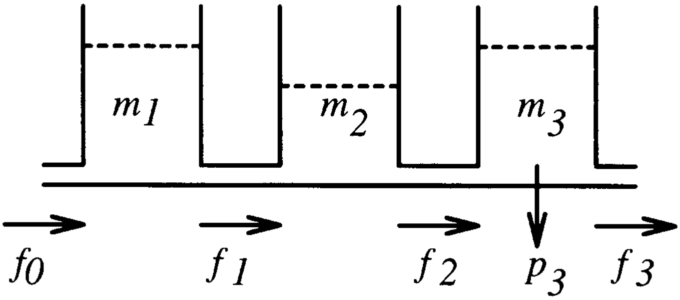


Figure 11.2 Three coupled tanks

where f_i is the mass flow in the ith pipe and Δp_i is the corresponding pressure drop. The tanks are modelled as pure capacities

$$\dot{m}_i = f_{i-1} - f_i \tag{11.2}$$

$$p_i = \frac{g}{a_i} m_i \tag{11.3}$$

where m_i is the liquid mass in the ith tank and p_i is the corresponding pressure.

It is assumed that

- all constants are unity (this is for simplicity);
- the only measurement is p_3, the pressure (proportional to the level) at the base of tank 3;
- the quantity to be controlled is p_2, the pressure (proportional to the level) at the base of tank 2;
- the only system control input is f_0, the system inflow. This is limited to the range $0 \leq f_0 \leq 10$.

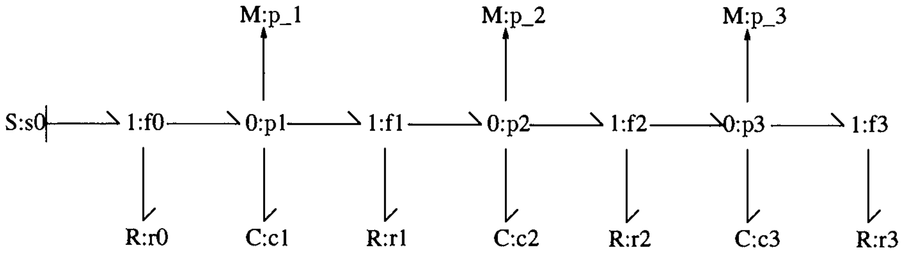


Figure 11.3 Three coupled tanks - bond graph

The system is thus non-linear in two ways: the non-linear flow resistances and the constrained control signal. It is modelled by the bond graph of Figure 11.3. All symbols on this bond graph are standard. On the other bond graphs the SS component denotes a collocated source-sensor pair.

The system (differential) equations are

$$x = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} ; \; y = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} ; \; u = (f_0) \tag{11.4}$$

$$\begin{aligned} \dot{x}_1 &= -(\sqrt{(x_1 - x_2)} - u_1) \\ \dot{x}_2 &= \sqrt{(x_1 - x_2)} - \sqrt{(x_2 - x_3)} \\ \dot{x}_3 &= \sqrt{(x_2 - x_3)} - \sqrt{(x_3)} \end{aligned} \tag{11.5}$$

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_2 \\ y_3 &= x_3 \end{aligned} \tag{11.6}$$

11.3.2 The model-based observer

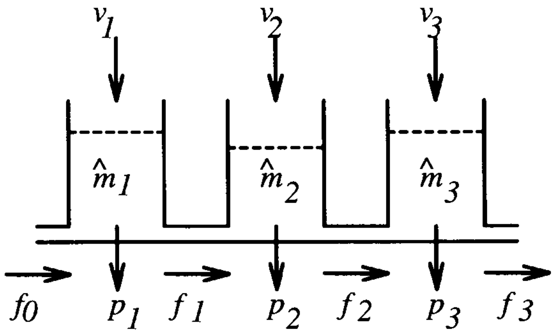


Figure 11.4 Three coupled tanks observer– schematic model

The physical system corresponding to the *model* is shown in Figure 11.4. There are two differences compared with the system itself:

- There are three additional inputs: the flows v_1, v_2 and v_3.

- All variables are available for measurement.

The additional inputs provide a means of adjusting the model states ($\hat{m}_i$) towards the corresponding system states (m_i).

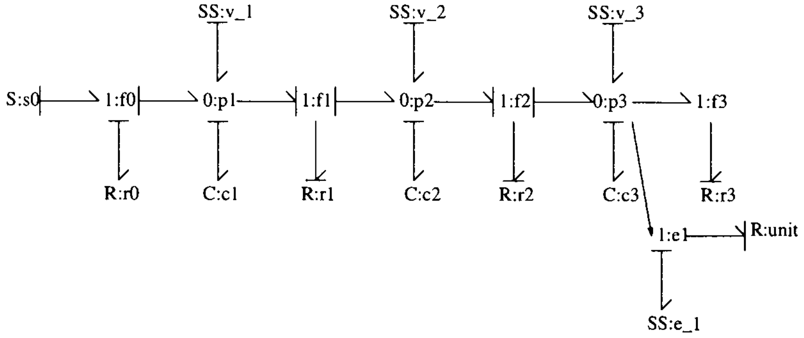


Figure 11.5 Three coupled tanks observer– bond graph model

The model bond graph is given by Figure 11.5. The corresponding model equations are

$$x = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} ; \; y = \begin{pmatrix} e_1 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} ; \; u = \begin{pmatrix} f_0 \\ e_1 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} \tag{11.7}$$

$$\begin{aligned} \dot{x}_1 &= -(\sqrt{(x_1 - x_2)} - u_1 - u_3) \\ \dot{x}_2 &= \sqrt{(x_1 - x_2)} - \sqrt{(x_2 - x_3)} + u_4 \\ \dot{x}_3 &= \sqrt{(x_2 - x_3)} - \sqrt{(x_3)} + u_5 \end{aligned} \tag{11.8}$$

$$\begin{aligned} y_1 &= x_3 - u_2 \\ y_2 &= x_1 \\ y_3 &= x_2 \\ y_4 &= x_3 \end{aligned} \tag{11.9}$$

The *model-based observer* is created from the model together with the feedback

$$v = K_o e \tag{11.10}$$

where the error signal e is the difference between system and model outputs

$$e = y - \hat{y} \tag{11.11}$$

This is implemented using the part of the bond graph of Figure 11.5 with junction labelled 'e1'.

There are many possibilities for designing the observer feedback gains. For example, the *observer gain* K_o may be designed (for example using pole-placement or LQ theory (Kwakernaak and Sivan, 1972)) using a *linearised* model about the steady-state corresponding to an inflow f_0

$$x_0 = \begin{pmatrix} 3f_0^2 \\ 2f_0^2 \\ f_0^2 \end{pmatrix} \tag{11.12}$$

$$u_0 = (\, f_0 \,) \tag{11.13}$$

$$y_0 = \begin{pmatrix} 3f_0^2 \\ 2f_0^2 \\ f_0^2 \end{pmatrix} \tag{11.14}$$

The linearised model is of the form

$$\dot{\tilde{X}} = A\tilde{X} + B\tilde{u} \tag{11.15}$$

$$\tilde{y} = C\tilde{X} + D\tilde{u} \tag{11.16}$$

Where $\tilde{X}$, $\tilde{y}$ and $\tilde{u}$ are the state, output and input deviation variable and the matrices A, B, C and D are

$$A = \begin{pmatrix} \frac{(-1)}{(2f_0)} & \frac{1}{(2f_0)} & 0 \\ \frac{1}{(2f_0)} & \frac{(-1)}{f_0} & \frac{1}{(2f_0)} \\ 0 & \frac{1}{(2f_0)} & \frac{(-1)}{f_0} \end{pmatrix} \tag{11.17}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{11.18}$$

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{11.19}$$

$$D = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{11.20}$$

x_i refers to the ith component of the state corresponding to a steady state of the system.

Note that although the feedback is *linear*, the observer itself is *non-linear*. Other designs, both linear and non-linear, would also be possible.

11.3.3 Control

The purpose of the model-based observer is to drive the *model* states towards the *system* states; the purpose of the MBO control is to drive appropriate *model* variables towards desired setpoints.

This latter objective may be achieved by any control law, linear or non-linear, acting on any variables available in the model (which are not necessarily available from system sensors).

For this example, the control objective is to drive the pressure (proportional to the level) in tank 2 (p_2) to 1.0. This is achieved using a cascade configuration making use of the observer variables p_1 and p_2 to generate the input flow f_0. Note that neither p_1 nor p_2 can actually be measured. For this example, a proportional control was used on the inner (tank 1) loop and a proportional + integral control was used on the outer (tank 2) loop.

11.3.4 Simulation

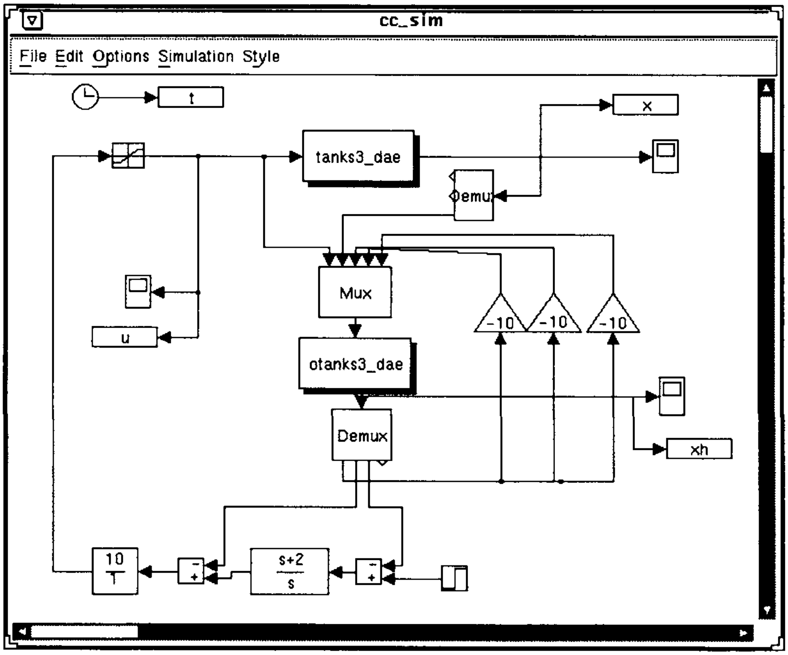


Figure 11.6 The simulation diagram

The bond graph toolbox MTT (Gawthrop *et al.*, 1991*a*) was used to generate both system and model equations in a form suitable for Simulab. The simulation diagram appears in Figure 11.6.

No control

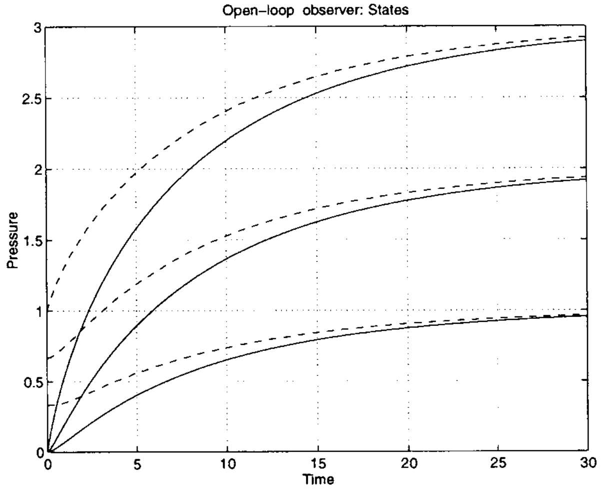


Figure 11.7 Open-loop observer, step input

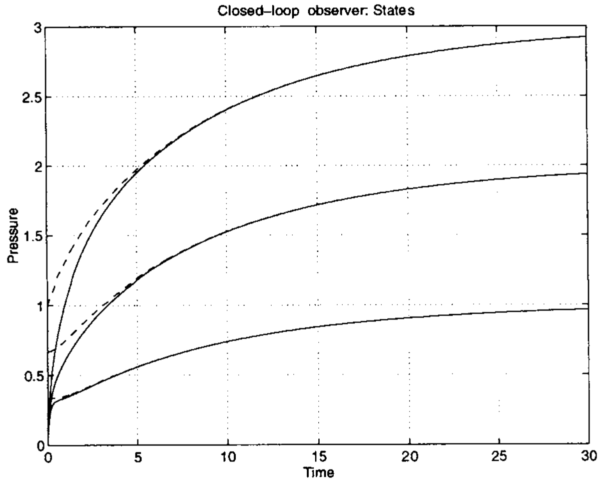


Figure 11.8 Closed-loop observer, step input

Figures 11.7 and 11.8 show observer states (firm lines) and system states (dashed lines). For the purposes of illustration, all system states were initialised at values corresponding to a unit level in the first tank (the other levels being $\frac{2}{3}$ and $\frac{1}{3}$ respectively). The model state was initialised to zero.

Figure 11.7 shows the results with an open-loop observer (no feedback to the model) and no control. The inflow was set to $f_0 = 1$. The model states eventually converge to those of the system because, in this particular case, the system is stable. Figure 11.8 shows the effect of closing the observer loop: the model states converge to those of the system more rapidly.

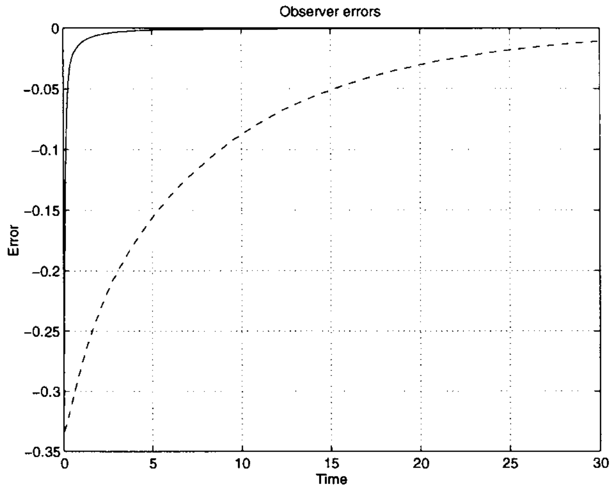


Figure 11.9 Closed- and open-loop errors

Figure 11.9 shows the observer error signal for the closed-loop (firm line), and open loop (dashed line).

Closed-Loop control

Figure 11.10 corresponds to an open-loop observer but with cascade control applied to the model. The second state of the observer (numerically equal to $\hat{p}_2$ in this case) is driven to the desired value of 1.0. But, because the observer is open-loop, this model state takes some time to converge to the corresponding system state.

Figure 11.11 corresponds to a closed-loop observer to give the complete model-based observer control. Not only is the model state corresponding to $\hat{p}_2$ driven to the setpoint, but also the model states are driven towards the system states.

Known disturbances

In the control of most process engineering systems we are primarily concerned with the disturbance problem. In this particular system we will examine the rejection of a disturbance on the level (or liquid pressure) in the middle tank caused by a unit step leakage from the first tank initiated at time $t = 10$. In this section, it is assumed that this disturbance in *known* and therefore available to the model. Figure 11.12 corresponds to

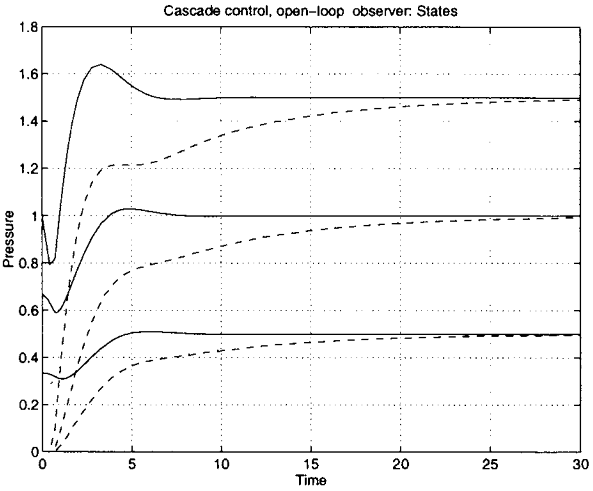


Figure 11.10 Open-loop observer, closed-loop control

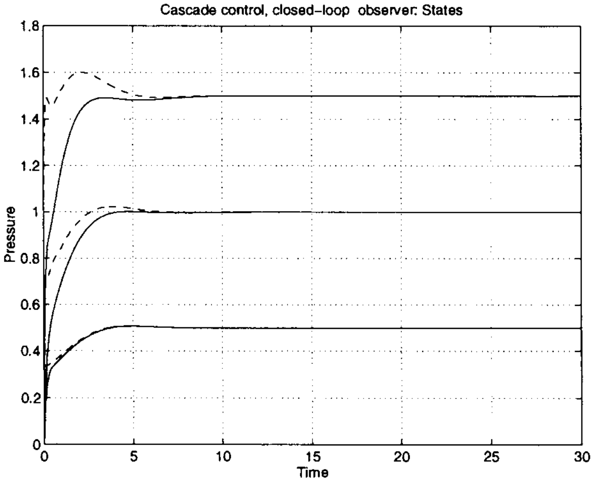


Figure 11.11 Closed-loop observer, closed-loop control

a closed-loop observer to give the complete model-based observer control. The controller is not making use of the measured disturbance signal, so there is a small tracking error at time 10. The control signal appears in Figure 11.13.

The controller structure could be changed to take account of the measured disturbance as a feedforward term; but it is hardly necessary here.

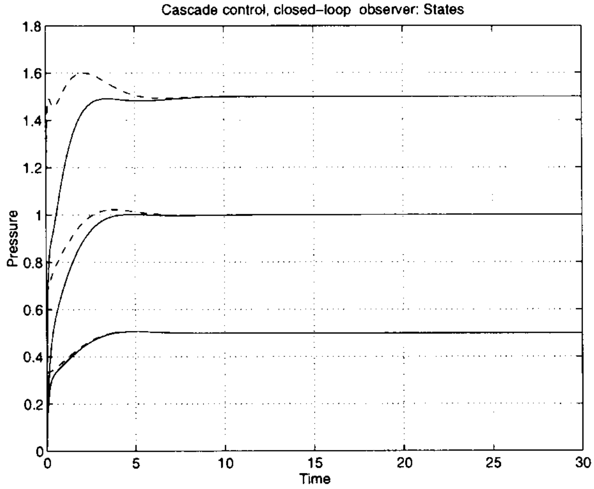


Figure 11.12 Known disturbance

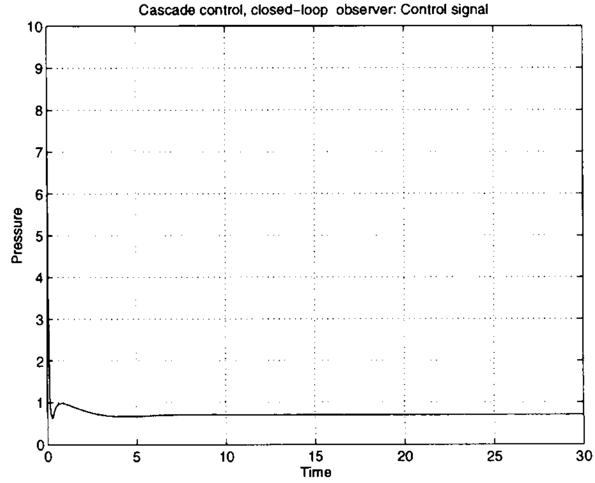


Figure 11.13 Control signal

11.4 UNKNOWN DISTURBANCES

The previous section assumed that the system disturbance was known; in this section it is assumed unknown and therefore not available to the model. Repeating the simulation depicted in Figure 11.12 but with the disturbance *not* supplied to the model gives Figure 11.14. The performance of the model-based observer is not good in the face of this *unmeasured* load disturbances; although the model state corresponding to $\hat{p}_2$ is driven towards the setpoint, the model states are too far from the system states to give satis-

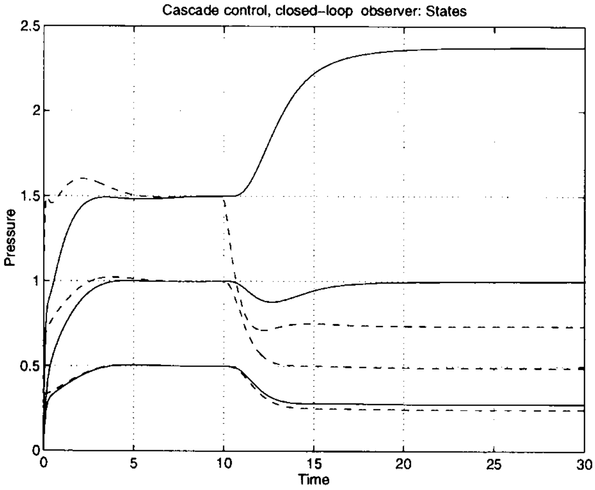


Figure 11.14 Closed-loop observer, closed-loop control

factory control. Perhaps the obvious reaction to this is to include an integral term in the MBO feedback. Though this will achieve our objective it is preferable to use a different approach if possible. Instead, we will use another well-known bit of control theory: the disturbance, as well as the process itself, must be modelled. In the case of the example used here, it is natural to model the disturbance as being caused by the level in a fourth tank which is subject to abrupt changes at unexpected times. The changes and the corresponding times are unknown and cannot be inputs to the model; nevertheless, they can be, in effect, deduced by the observer in the same way as unknown states. Moreover, because the *change* in the level of this additional level is transitory, the corresponding error will also be transitory.

The bond graph of the corresponding augmented model appears in Figure 11.15. The additional (as compared to Figure 11.5) components model the disturbance. This additional subsystem is connected to the rest of the model by a signal bond to indicate that the disturbance does not depend on the process itself.

The corresponding model equations are

$$x = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ m_d \end{pmatrix} ; \; y = \begin{pmatrix} e_1 \\ v_1 \\ v_2 \\ v_3 \\ v_0 \end{pmatrix} ; \; u = \begin{pmatrix} f_0 \\ e_1 \\ v_1 \\ v_2 \\ v_3 \\ v_0 \end{pmatrix} \tag{11.21}$$

$$\begin{aligned}
\dot{x}_1 &= -(\sqrt{(x_1 - x_2)} - x_4 - u_1 - u_3) \\
\dot{x}_2 &= \sqrt{(x_1 - x_2)} - \sqrt{(x_2 - x_3)} + u_4 \\
\dot{x}_3 &= \sqrt{(x_2 - x_3)} - \sqrt{(x_3)} + u_5
\end{aligned}$$

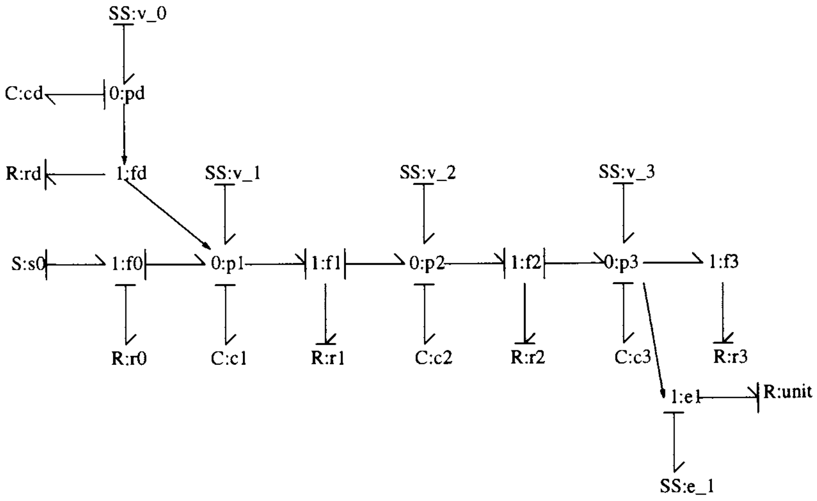


Figure 11.15 Three coupled tanks with disturbance - model bond graph

$$\dot{x}_4 = u_6 \tag{11.22}$$

$$\begin{aligned} y_1 &= x_3 - u_2 \\ y_2 &= x_1 \\ y_3 &= x_2 \\ y_4 &= x_3 \\ y_5 &= x_4 \end{aligned} \tag{11.23}$$

The result of using this augmented model with an unknown disturbance appears in Figure 11.16. Notice that, although the estimated state is in error immediately after the disturbance change at time 10, this is only transitory. The additional observer state tracks the disturbance.

11.5 CONCLUSION

A brief outline of the model-based observer approach to model-based control has been given. The focus has been on the observer rather than the controller, as control becomes relatively simple when all measurements are available.

More work needs to be done in the following areas:

- Use of more advanced control strategies such as GPC.
- Application to more complex processes such as distillation.
- A critical comparison, both theoretical and experimental, with other model-based approaches.

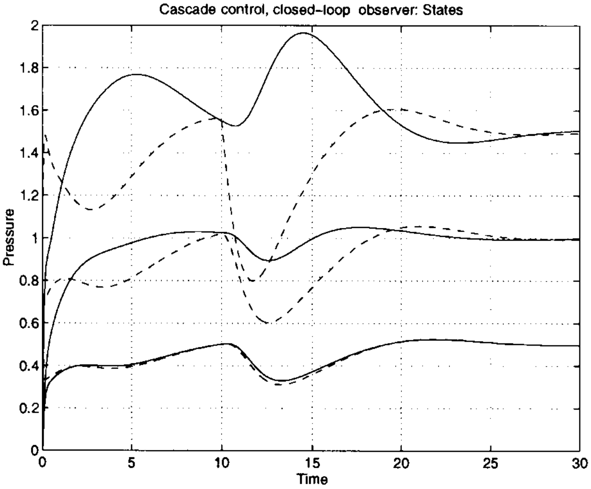


Figure 11.16 Unknown disturbance: augmented model

The use of Bond graphs for model-based identification have been independently presented by two groups (Nagy and Ljung, 1991; Gawthrop *et al.*, 1991*b*; Gawthrop *et al.*, 1993). When combined with the model-based observer control this provides the basis for automatic generation of adaptive control algorithms for partially-known systems (Gawthrop and Jones, 1992).

References

An, C. H., C. G. Atkeson and J. M. Hollerbach (1988). *Model-based Control of Robot Manipulators.* The MIT Press.

Anex, Jr, R. P. and M. Hubbard (1984). 'Modelling and adaptive control of a mechanical manipulator'. *Journal of Dynamic Systems, Measurement and Control.*

Armstrong, B., O. Khatib and J. Burdik (1986). The explicit dynamic model and inertial parameters of the puma 560 arm. In 'IEEE International Conference on Robotics and Automation'. pp. 510–518.

Borutzky, W. (1993). On interrelations between bond graph causal patterns and the numerical solution of the mathematical model. In J. J. Granda and F. E. Cellier (Eds.). 'Proceedings of the International Conference on Bond Graph Modelling (ICBGM 93)'. Vol. 25. Society for computer simulation. pp. 93–100.

Breedveld, P. C. (1984*a*). 'A bond graph algorithm to determine the equilibrium state of a system'. *J. Franklin Institute.*

Breedveld, P. C. (1984*b*). Physical Systems Theory in Terms of Bond Graphs. Phd thesis. Enschede.

Brenan, K. E., S. L. Campbell and L. R. Petzold (1989). *Numerical solution of Initial Value Problems in Differential-algebraic equations.* North-Holland. New York.

Brewer, J., P. Craig, M. Hubbard and K. Watt (1982). 'The bond graph methodology for technological forecasting and resource policy analysis'. *Energy.*

Brewer, J. W. and P. P. Craig (1982). 'Bilinear, dynamic single ports and bond graphs of economic systems'. *J. Franklin Institute* **313**(4), 185–196.

Campbell, S. L. (1980). *Singular Systems of Differential Equations.* Pitman. London.

Campbell, S. L. (1982). *Singular Systems of Differential Equations II.* Pitman. London.

Cellier, F. E. (1991). *Continuous system modelling.* Springer-Verlag.

Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control(2nd ed.).* Addison-Wesley.

Davis, N. R. and W. W. Mapleson (1981). 'Structure and quantification of a physiological model of the distribution of injected agents and inhaled anaesthetics'. *British Journal of Anaesthesia* **53**, 399.

Dijk, J. V. and P. C. Breedveldt (1995). Relaxed causality: A bond graph appoach to dae modelling. In F. E. Cellier and J. J. Granda (Eds.). 'Proceedings of 1995 International Conference on Bond Graph Modelling and Simulation'. Society for Computer Simulation. pp. 297–302.

Fattah, Y. E. (1992). 'Constraint logic programming: applications and implications'. *Artificial Intelligence in Engineering* **7**, 175–182.

Fattah, Y. E. (1995). Constraint-based programming for bond graph causality. In F. E. Cellier and J. J. Granda (Eds.). 'Proceedings of 1995 International Conference on Bond Graph Modelling and Simulation'. Society for Computer Simulation. pp. 89–94.

Franks, R. G. E. (1972). *Modelling and simulation in chemical engineering*. Wiley.

Fu, K. S., R. C. Gonzalez and C. S. G. Lee (1987). *Robotics: Control, Sensing, Vision and Intelligence*. McGraw-Hill.

Gawthrop, P. J. (1991). 'Bond graphs: A representation for mechatronic systems'. *Mechatronics* **1**(2), 127–156.

Gawthrop, P. J. (1995). Mtt: Model transformation tools. In 'Proceedings Of International Conference On Bond Graph Modeling And Simulation (ICBGM'95)'. Society for Computer Simulation. Las Vegas. pp. 197–202.

Gawthrop, P. J. and L. Smith (1992). 'Causal augmentation of bond graphs'. *Journal of the Franklin Institute* **329**(2), 291–303.

Gawthrop, P. J. and R. W. Jones (1992). Bond-graph-based adaptive control. In '4th IFAC Symposium on Adaptive systems in Control and Signal Processing'.

Gawthrop, P., J. Ježek, R. W. Jones and I. Sroka (1993). 'Grey-box model identification'. *Control-Theory and Advanced Technology*.

Gawthrop, P. J., N. A. Marrison and L. Smith (1991*a*). MTT: A bond graph toolbox. In 'Proceedings of the 5th IFAC/IMACS Symposium on Computer-aided Design of Control Systems:CADCS91, Swansea, Wales'. pp. 274–279.

Gawthrop, P. J., R. W. Jones and S. A. MacKenzie (1991*b*). Identification of partially-known systems. In '9th IFAC/IFORS symposium on identification and system parameter estimation. Budapest, Hungary'. pp. 1347–1352.

Gawthrop, P. J., R. W. Jones and S. A. MacKenzie (1992). Bond graph based control: A process engineering example. In 'American Control Conference'.

Gear, C. W. and L. R. Petzold (1984). 'Ode methods for the solution of differential/algebraic systems'. *SIAM J. Numer. Anal.* **21**, 716–728.

Henson, M. A. and D. E. Seborg (1991). 'Critique of exact linearisation strategies for process control'. *J. Process Control* **1**, 122–139.

Isidori, A. (1989). *Nonlinear Control Systems: An Introduction. 2nd Ed.*. Springer-Verlag. New York.

Iwasaki, Y. and H. A. Simon (1986). 'Causality in device behaviour'. *Artificial Intelligence* **29**, 3–32.

Joseph, B., C. B. Brosilow and M. Tong (1978). 'Inferential control of processes: Parts i-III'. *AIChE Journal* **25**, 485–509.

Kailath, T. (1980). *Linear Systems*. Prentice-Hall.

Karnopp, D. C. (1969). 'Power-conserving transformations: Physical interpretations and applications using bond graphs'. *J. Franklin Institute* **288**(3), 175–201.

Karnopp, D. C. (1978). 'Pseudo bond graphs for thermal energy transport'. *J. Dynamic Systems, Measurement and Control* **100**, 165–169.

Karnopp, D. C. (1979). 'Bond graphs in control: Physical state variables and observers'. *J. Franklin Institute* **308**(3), 221–234.

Karnopp, D. C. (1983). 'Alternative bond graph causal patterns and equation formulations for dynamic systems'. *Transactions of the ASME* **105**, 58–63.

Karnopp, D. C. and R. C. Rosenberg (1975). *System Dynamics: A Unified Approach.* John Wiley.

Karnopp, D. C., D. L. Margolis and R. C. Rosenberg (1990). *System Dynamics: A Unified Approach.* John Wiley.

Kurihara, F. and S. Kimura (1985). 'An experimental study of the viscosity of polymer melts at high shear rate'. *Polymer Journal* **17**(7), 863–868.

Kwakernaak, H. and R. Sivan (1972). *Linear Optimal Control Systems.* Wiley.

Lane, S. H. and R. F. Stengel (1988). 'Flight control design using non-linear inverse dynamics'. *Automatica* **24**(4), 471–483.

Lee, P. L. and G.R. Sullivan (1988). 'Generic model control (gmc)'. *Comput. Chem. Engng* **12**(9), 573–580.

LeFèvre, J. (1995). Why and how should we introduce bond graphs to life scientists?. In F. E. Cellier and J. J. Granda (Eds.). 'Proceedings of 1995 International Conference on Bond Graph Modelling and Simulation'. Society for Computer Simulation. pp. 297–302.

Leler, W. (1988). *Constraint Programming Languages.* Addison Wesley.

Lorenz, F. and J. Wolper (1985). 'Assigning causality in the case of algebraic loops'. *J. Franklin Institute* **319**(1/2), 237–241.

Luenberger, D. G. (1977). 'Dynamic equations in descriptor form'. *IEEE Trans. Autom. Control* **AC-22**, 312.

MacFarlane, A. G. J. (1964). *Engineering Systems Analysis.* G.G.Harrap. Cambridge, Mass.

Maciejowski, J. M. (1989). *Multivariable Feedback Design.* Addison-Wesley.

MacKenzie, S. A., P. J. Gawthrop and R.W. Jones (1993). Modelling chemical processes with pseudo bond graphs. In 'Proceedings of International Conference on Bond Graph Modelling and Simulation, San Diego'.

MacKenzie, S. A., P. J. Gawthrop, R. W. Jones and J. W. Ponton (1991). Systematic modelling of chemical processes. In 'IFAC symposium on Advanced Control of Chemical Processes, ADCHEM'91. Toulouse, France.'.

Mapleson, W. W. (1964). 'Mathematical aspects of the uptake, distribution and elimination of inhaled gases and vapours'. *British Journal of Anaesthesia* **36**, 129.

Mapleson, W. W. (1973). 'Circulation-time models of the uptake of inhaled anaesthetics and data for quantifying them'. *British Journal of Anaesthesia* **45**, 319.

Mattsson, S. E. (1989). 'On modelling and differential/algebraic systems'. *Simulation* pp. 24–32.

Morari, M. and E. Zafiriou (1989). *Robust Process Control.* Prentice-Hall. Englewood Cliffs.

Nagy, P. A. J. and L. Ljung (1991). System identification using bond graphs. In 'Proceedings of the First European Control Conference'. Hermes. Grenoble. pp. 2564–2569.

Pantelides, C. C., D. Gritsis, K. R. Morison and R. W. H. Sargent (1988). 'The mathematical modelling of transient systems usign differential-algebraic equations'. *Comput .chem. Engng* **12**(5), 449–454.

Parnaby, J., A. Kochhar, C. Chow and B. Wood (1973). Identification of process models for plastics extrusion systems - a preliminary study. In 'Proceedings of 3rd IFAC Conference'. Delft. pp. 299–306.

Parrish, J. R. and C. B. Brosilow (1985). 'Inferential control algorithms'. *Automatica* **21**(5), 527–538.

Paul, R. P. (1981). *Robot Manipulators: Mathematics, Programming, and Control.* The MIT Press series in artificial intelligence. Cambridge, Mass.

Paynter, H. M. (1961). *Analysis and design of engineering systems.* MIT Press. Cambridge, Mass.

Rayna, G. (1987). *Reduce: Software for Algebraic Computation.* Springer.

Reber, D., R.E. Lynn and E. Freeh (1973). 'A mathematical model for predicting dynamic behaviour of a plasticating extruder'. *Polymer Engineering and Science* **13**, 346–356.

Rosenberg, R. C. and D. C. Karnopp (1983). *Introduction to Physical System Dynamics.* McGraw-Hill.

Schmidt, J. W. (1985). Introduction to systems analysis, modelling and simulation. In S. D.Gantz, G.Blais (Ed.). 'Proc. 1985 Winter Simulation Conference'.

Simon, H. A. (1952). 'On the definition of the causal relation'. *The Journal of Philosophy* **XLIX**(16), 517–528.

Simon, H. A. and A Ando (1961). 'Aggregation of variables in dynamic systems'. *Econometrica* **29**(2), 111–138.

Simon, H. A. and N. Rescher (1966). 'Cause and conterfactual'. *Philosophy of Science* **33**, 323–340.

Tadmor, Z., S. D. Lipshitz and R. Lavie (1974). 'Dynamic model of a plasticating extruder'. *Polymer Engineering and Science* **14**(2), 112–119.

Thoma, J. (1975). *Introduction to Bond Graphs and their Applications.* Pergamon Press.

Thoma, J. U. (1990). *Simulation by bond graphs.* Springer-Verlag. Berlin.

Tiernego, M. J. L. and A. M Bos (1985). 'Modelling the kinematics and dynamics of mechanical systems with multibond graphs'. *J. Franklin Institute* **319**(1/2), 37–50.

van Dijk, J. (1994). On the role of bond graph causality in modelling mechatronic systems. PhD thesis. Universitiet Twente.

van Dijk, J. and P. C. Breedveld (1991*a*). 'Simulation of system models containing zero-order causal paths – i. classification of zero-order causal paths'. *J. Franklin Institute* **328**(5/6), 959–979.

van Dijk, J. and P. C. Breedveld (1991*b*). 'Simulation of system models containing zero-order causal paths – numerical implications of class i zero-order causal paths'. *J. Franklin Institute* **328**(5/6), 981–1004.

van Dijk, J. and P. C. Breedveld (1993). The structure of the semi-state space form derived from bond graphs. In J. J. Granda and F. E. Cellier (Eds.). 'Proceedings of the International Conference on Bond Graph Modelling (ICBGM 93)'. Vol. 25. Society for computer simulation. pp. 101–107.

Verghese, G. C., B. C. Levy and T. Kailath (1981). 'A generalised state-space for singular systems'. *IEEE Trans. Autom. Control* **AC-26**, 811–831.

Wellstead, P. E. (1979). *Introduction to Physical System Modelling.* Academic Press.

Winston, P. H. (1984). *Artificial Intelligence.* Vol. 2nd Edition. Addison-Wesley.

Wolovich, W. A. (1987). *Robotics: Basic Analysis and Design.* Holt, Rinehart and Winston.

Index