

Wolfgang Borutzky

Bond Graph Methodology

Development and Analysis of
Multidisciplinary Dynamic System Models

 Springer

Bond Graph Methodology

Wolfgang Borutzky

Bond Graph Methodology

Development and Analysis of Multi-
disciplinary Dynamic System Models



Springer

Prof. Dr. Ing. Wolfgang Borutzky
Bonn-Rhein-Sieg University of Applied Sciences
53754 Sankt Augustin
Germany

ISBN 978-1-84882-881-0

e-ISBN 978-1-84882-882-7

DOI 10.1007/978-1-84882-882-7

Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data

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

Library of Congress Control Number: 2009939277

© Springer-Verlag London Limited 2010

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The original edition of this book was published by SCS Publishing House as *Bond Graphs: A Methodology for Modelling Multidisciplinary Dynamic Systems*. © SCS Publishing House, 2004

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Cover design: eStudioCalamar, Figueres/Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To my wife Heidrun and
to my son Lars*

Preface

Nowadays, engineering systems are of ever-increasing complexity and must be considered as *multidisciplinary* systems composed of interacting subsystems or system components from different engineering disciplines. Thus, an integration of various engineering disciplines, e.g. mechanical, electrical and control engineering in a *concurrent* design approach is required. With regard to the systematic development and analysis of system models, *interdisciplinary* computer aided methodologies are becoming more and more important.

A graphical description formalism particularly suited for multidisciplinary systems are *bond graphs* devised by Professor Henry Paynter in as early as 1959 at the Massachusetts Institute of Technology (MIT) in Cambridge, Massachusetts, USA and in use since then all over the world.

This monograph is devoted exclusively to the bond graph methodology. It gives a comprehensive, in-depth, state-of-the-art presentation including recent results scattered over research articles and dissertations and research contributions by the author to a number of topics.

The book systematically covers the fundamentals of developing bond graphs and deriving mathematical models from them, the recent developments in methodology, symbolic and numerical processing of mathematical models derived from bond graphs. Additionally it discusses modern modelling languages, the paradigm of object-oriented modelling, modern software that can be used for building and for processing of bond graph models, and provides a chapter with small case studies illustrating various applications of the methodology.

In favour of presenting topics in some reasonable depth and to keep the size of the book manageable, the book refrains from scratching the surface of too many topics. For this reason, some topics, e.g., chemical reactions or links to qualitative reasoning, are not addressed. The compilation of the material in this book and its presentation has been motivated by the author's individual experiences in research and teaching for more than two decades and has been inspired by his personal interaction with many leading personalities in this area.

This monograph addresses students, lecturers, researchers, and practicing engineers in industry who want to learn more about the potential and the state-of-the-art

design of this powerful interdisciplinary graphical modelling methodology and who want to see how it can help them better understand physical processes in multi-energy domain engineering systems in order to develop proper models in their respective engineering field. As bond graph modelling is based on the intuitive consideration of energy exchange between system components from various energy domains, the methodology is particularly suited for modelling and design tasks in mechatronics.

Bond Graph Methodology – Development and Analysis of Multidisciplinary Models addresses the fundamentals as well as advanced topics. It has been designed to serve readers interested in what bond graph modelling is about, readers with expertise in related areas who want to see how bond graph modelling can help them in their projects as well as members of the international community of bond graph modellers. The book can be used as a supplementary text in master's programme courses on modelling, simulation and control, as well as a guide for self-study and as a reference.

The progress made in bond graph modelling is due to many people around the world. Without their research, this monograph would not have been possible.

Acknowledgements

I would like to take this opportunity to thank the following colleagues of mine for their review of chapters of the manuscript. Their valuable feedback, comments and suggestions are gratefully acknowledged.

Vjekoslav Damić	University of Dubrovnik, Croatia
Geneviève Dauphin-Tanguy	École Centrale de Lille, France
José Granda	California State University Sacramento, CA, USA
Sergio Junco	Universidad Nacional de Rosario, Argentina
Loucas Louca	University of Cyprus, Cyprus
Wilfrid Marquis-Favre	Institut National des Sciences Appliquées de Lyon, France
David Murray-Smith	University of Glasgow, Scotland, UK
Arun Samantaray	Indian Institute of Technology, Kharagpur, India
Jean Thoma	University of Waterloo, Ontario, Canada

Furthermore, I wish to thank the team of Springer-Verlag at London, UK, for supporting this project. In particular, I would like to thank Claire Protherough for her kind help.

Last but not least, I wish to express my sincere thanks and appreciation to my wife for her support, her tolerance and her patience with me.

Sankt Augustin
May 2009

Wolfgang Borutzky

About This Book

This book on Bond Graph Methodology is organised into 13 chapters. Its focus is mainly on concepts, modelling approaches, techniques and software tools that support the process of bond graph based physical systems modelling.

By covering a number of advanced topics, e.g., models of variable structure, multibody systems and open thermodynamic systems, the book aims at demonstrating the true interdisciplinary potential of bond graph methodology. The discussion of concepts makes use of many examples that, for pedagogical reasons, have been kept fairly small and easy to survey. A glossary supports the use of the terminology.

The introduction briefly recalls essential features of block diagrams, signal flow graphs, network representations and the paradigm of object-oriented modelling. The intention is to begin with well known graphical model representations and modelling techniques and to outline the context in which bond graph modelling is embedded.

Chapter 2 provides the fundamentals of bond graph based physical systems modelling, allowing the reader to systematically construct a so-called non-causal bond graph from a system schematic system representation. Non-causal bond graphs reflect the structure of the system to be modelled. In the conceptual phase of developing a hierarchical (top-down, or bottom-up) model, functional relations between physical quantities, given either by equations or by look-up tables are not important in the beginning. Consequently, at a conceptual model development stage, non-causal bond graphs are used to represent physical effects and relations between them in a *qualitative* manner. At this early stage of model development, a mathematical model derived from a non-causal bond graph could take only the form of a set of differential-algebraic equations including many redundant algebraic equations due to the interconnection of system component models.

With regard to the systematic derivation of a mathematical model from a bond graph, the concept of *computational causality* is explained in Chapter 3. The so-called *Standard Causality Assignment Procedure* (SCAP), introduced by Karnopp and Rosenberg, is presented and the choice of state space variables is addressed.

After an introduction of the basic notion of a *causal path*, it is shown how an ordered set of equations can be derived manually from causal bond graphs of moderate size by following causal paths back and forth. In the simplest case, the equations

obtained can be turned into a set of explicit state space equations by eliminating auxiliary algebraic variables.

Beyond the most simple case of an explicit state space form, several types of causal paths in a bond graph and their effect on the form of the mathematical model are considered in Chapter 4. It turns out that a mathematical model to be derived from a causal bond graph, in general, is a set of differential-algebraic Equations (DAE system). Therefore, important notions from the theory of DAE systems are provided, in particular the notions of the *index of nilpotency* and the *differential index* of a DAE system.

From a modelling point of view, the generation of a DAE system can be avoided by inserting small energy storage elements into the bond graph. This way, a model is slightly modified. This approach, quite common before numerically robust solvers for DAE systems of index < 2 came up, is briefly discussed. Moreover, two alternatives to the standard causality assignment procedure, viz., the *method of relaxed causalities*, introduced by Joseph and Martens, and Lagrange causalities (Karnopp 1977) are considered.

In Chapter 5, some aspects of solving DAE systems numerically by means of the standard multistep Backward Differentiation Formula (BDF) are discussed. Some modelling and simulation software packages can perform symbolic manipulation of the equations of the DAE system *before* it is passed on to a numerical solver. In this context, reduction of the differential index of a DAE system based on Pantelides' algorithm is considered. Moreover, it is demonstrated how tearing of algebraic constraints can be supported by adding controlled sources to a bond graph.

Chapter 6 shows that a bond graph can be considered a *core* model representation from which not only equations for simulation can be derived, but also transfer functions, information about structural properties, e.g., *structural* controllability and *structural* observability as well as equations of the inverse system, parameter sensitivities and forms of state equations for robustness study. Also, bond graphs can support model-based fault detection and isolation.

As is well known, engineering components (e.g., diodes, thyristors, hydraulic check valves, clutches) or physical effects (e.g., stick-slip friction or the stop of a piston at a limiting position) give rise to the modelling abstraction of an instantaneous discontinuous change of state. With regard to the overall system dynamics, this abstraction is appropriate and can help avoid numerical problems due to steep gradients. For these reasons, it is common practice to use this abstraction. Chapter 7 considers several approaches to a combination of this abstraction with bond graph modelling. The latter methodology is based on conservation laws from physics and requires that state transitions take place in a nonzero time period. It appears that so far, no standard bond graph approach to hybrid modelling (time continuous models including the description of discrete events) has formed.

Chapters 8, 9, and 10 on multibody systems, on lumped parameter bond graph model approximations of distributed parameter models, and on open thermodynamic systems show the potential of the interdisciplinary bond graph methodology. *Multibond graphs* as introduced by Breedveld allow for a concise representation of models of multibody systems with bodies assumed to be rigid.

In Chapter 8, multibond graph modelling of multibody systems as proposed by Bos is presented. That is, translational motion of bodies assumed to be rigid is referred to an inertial frame, whereas the rotation of a body is described with reference to a body fixed frame. Moreover, bond graph modelling of multibody systems can also be used for the joint coordinate method that is well known in mechanical engineering.

If some bodies of a multibody system are to be considered flexible and consequently are to be represented by a distributed parameter model, then the latter can be approximated by a lumped parameter bond graph model based either on modal analysis (Karnopp and Margolis) or on a finite element approach (Pelegay, Doblare and Buil).

Chapter 10 presents fundamentals of pseudo bond graph as well as true bond graph modelling of open thermodynamic systems. Moreover, some effects in hydraulic systems are addressed that have given rise to an ad hoc representation in some bond graph related research reported in the literature.

Chapter 11 on automated modelling discusses modern modelling languages, the paradigm of object-oriented modelling (OOM), modern software that can be used for systematic development and processing of bond graph models.

As bond graph models have been developed concurrently in various places all over the world, Chapter 11 also addresses the issue of how this engineering knowledge can be shared. Inspired by the success of XML in many computer science related areas, a bond graph markup language, BGML, is proposed that can support the exchange and the reuse of bond graph models.

Regarding software for bond graph modelling, some features of three state-of-the-art integrated modelling and simulation environments have been considered and illustrated by some screen shots. However, it must be pointed out that the presentation of software is not meant to be product oriented. The aim, rather, is to outline different approaches towards a software support of bond graph modelling.

Finally, in Chapter 12, small elaborated case studies illustrate various applications of the bond graph methodology.

Contents

Abbreviations	xxi
1 Introduction	1
1.1 A Historical Survey of Bond Graph Modelling	1
1.2 Some General Aspects of Modelling Dynamic Systems	4
1.3 Object-Oriented Physical Systems Modelling	5
1.4 Traditional Graphical Model Representations	8
1.4.1 Block Diagrams	9
1.4.2 Signal Flow Graphs	10
1.4.3 Networks	11
1.5 Conclusion	12
References	14
2 Bond Graph Based Physical Systems Modelling	17
2.1 Fundamentals	17
2.1.1 Physical System Structure	17
2.1.2 Physical Systems Modelling	18
2.1.3 Multidisciplinary Engineering Systems	19
2.1.4 Hierarchical and Recursive Modelling	20
2.2 Nodes and Edges in Bond Graphs	21
2.3 Bond Graph Variables and Physical Analogies	22
2.3.1 Power Variables	22
2.3.2 Analogies	23
2.3.3 Energy Variables	28
2.4 Orientation of Power Bonds	29
2.5 Basic Bond Graph Elements and Power Port Orientations	29
2.5.1 Power Conserving Junctions	31
2.5.2 Ideal Power Couplers and Power Transducers	35
2.5.3 Energy Storage Elements	42
2.5.4 Dissipators	49
2.5.5 Memristors	56

2.5.6	Ideal Energy Sources and Sinks	59
2.5.7	Sensors	62
2.6	Pseudo Bond Graphs	63
2.7	Systematic Construction of Bond Graphs	66
2.7.1	Construction of Bond graphs for Mechanical Subsystems (Translation and Fixed-axis Rotation)	66
2.7.2	Construction of Bond Graphs for Non-mechanical Subsystems	70
2.7.3	Simplification of Some Bond Graph Structures	81
2.8	Some Remarks on the Choice of Orientations in Bond Graphs	82
2.9	Conclusion	85
	References	87
3	Derivation of Mathematical Models from Bond Graphs	89
3.1	On the Form of a Mathematical Model	89
3.2	The Concept of Computational Causality	92
3.2.1	The Notion of Computational Causality	92
3.2.2	Representation of Computational Causalities in Bond Graphs	93
3.2.3	Activated Bonds	94
3.2.4	Rules for Causality Assignment at the Ports of Bond Graph Elements	96
3.3	Sequential Assignment of Computational Causalities	101
3.4	On the Choice of State Variables	104
3.5	Systematic Derivation of Equations from a Bond Graph	109
3.5.1	Procedure for the Manual Derivation of Equations from a Causal Bond Graph	112
3.5.2	Application of the Procedure to Some Examples	112
3.6	Independent State Variables	119
3.7	Determination of the Number of Independent State Variables	123
3.8	Conclusion	126
	References	128
4	Causal Bond Graphs and Forms of Mathematical Models	129
4.1	Causal Paths Between Resistive Ports	129
4.2	Some Fundamentals from the Theory of Differential-Algebraic Systems	134
4.3	Inserting Energy Stores into Causal Paths Between Resistive Ports	139
4.4	Causal Paths Between Storage Ports of the Same Type	142
4.5	Closed Causal Paths	145
4.6	Bond Graphs with Causal Paths from Different Classes	150
4.7	Causal Loops of Unity Loop Gain	153
4.8	Algebraic Loops due to Internal Modulation	158
4.9	The Method of Relaxed Causalities	162
4.10	Lagrange Causalities	166

4.10.1	Identification of Generalised Coordinates in a Bond Graph	167
4.10.2	Determination of Generalised Forces from a Bond Graph	167
4.10.3	Derivation of Lagrange Equations from a Bond Graph	168
4.11	Conclusion	171
	References	174
5	Computing Mathematical Models Derived from Bond Graphs	177
5.1	Numerical Solution of Differential-Algebraic Systems	177
5.1.1	The Backward Differentiation Formula	177
5.1.2	Problems with the Numerical Solution of DAEs by Means of the BDF	179
5.2	Reduction of the Index of a Differential-Algebraic System	182
5.3	Reduction of Hamiltonian Equations of Motion with Constraints	189
5.4	Tearing of Algebraic Constraints	195
5.4.1	Causal Paths Between Resistive Ports	198
5.4.2	Causal Conflicts at Junctions	203
5.4.3	Causal Paths Between Storage Ports of the Same Type	205
5.4.4	Causal Loops	209
5.5	The Perturbation Index of Differential-Algebraic Equations	215
5.6	Conclusion	217
	References	220
6	Analysis of Causal Bond Graph Models	223
6.1	Equations Determining the Steady-state of a Dynamic System	224
6.2	Transfer Functions	227
6.2.1	Transfer Functions from the State Space Model	228
6.2.2	Transfer Functions from a Signal Flow Graph	229
6.2.3	Transfer Functions Directly from a Causal Bond Graph	231
6.3	Equations of the Inverse System	235
6.4	Structural Controllability and Observability	240
6.4.1	Structural Controllability	240
6.4.2	Structural Observability	244
6.5	Parameter Sensitivities	249
6.5.1	Incremental Models of Linear Bond Graph Elements	250
6.5.2	Derivation of Parameter Frequency Domain Sensitivities from an Incremental Bond Graph	254
6.6	State Equations for Robustness Study	261
6.6.1	Incremental Models of Linear Bond Graph Elements Revisited	262
6.6.2	Derivation of the Canonical Form of State Equations from an Incremental Bond Graph	265
6.6.3	The Standard Interconnection Form	269
6.6.4	Outline of the Uncertainty Bond Graph Approach	271
6.7	Bicausal Bond Graphs	276
6.7.1	Bicausal Bond Graphs for Parameter Estimation	279

6.7.2	Bicausal Bond Graphs for System Inversion	280
6.7.3	Bicausal Bond Graphs for State Estimation	282
6.8	Bond Graph Model-based Fault Detection and Isolation	282
6.8.1	Analytical Redundancy Relations	283
6.8.2	Structural Fault Signature Matrices	286
6.8.3	Fault Isolation	287
6.8.4	Residual Sinks in Bond Graph Model-based Fault Detection	288
6.9	Reduction of Model Complexity	292
6.9.1	Model Partitioning	292
6.9.2	Model Reduction	293
6.9.3	Structural Model Simplification	294
6.10	Conclusion	297
	References	299
7	Models of Variable Structure	305
7.1	Bond Graph Models with Fixed Causalities	306
7.1.1	Extending Element Characteristics	306
7.1.2	Switching Between System Modes by means of Modulation	308
7.1.3	Switched Power Junctions	313
7.1.4	Switching Off Degrees of Freedom by Sinks of Invariant Causality	317
7.2	Variable Causality Bond Graphs	331
7.2.1	Ideal Switches as Another Basic Bond Graph Element	331
7.2.2	Controlled Junctions – Hybrid Bond Graphs	337
7.3	A Combined Petri Net – Bond Graph Representation	340
7.4	Conclusion	346
	References	349
8	Multibody Systems	353
8.1	Brief Survey of Bond Graph Modelling of Multibody Systems	353
8.2	Multibond Graphs	354
8.2.1	Multibonds and Arrays of Bond Graph Elements	355
8.2.2	Multiport Energy Storage Elements	356
8.2.3	Multiport Transformers and Gytrators	358
8.2.4	Rotation of a Rigid Body in Space Described by a Multiport Gytrator	359
8.2.5	Multiport Resistors	364
8.2.6	Splitting a Multibond	365
8.3	Bond Graph Modelling of the 3D Motion of Multibody Systems	366
8.3.1	Multibond Graph of a Freely Moving Rigid Body	366
8.3.2	Connecting Instances of the Rigid Body Model	369
8.3.3	Multibond Graph Model of a Revolute Joint	370
8.3.4	Multibond Graph Model of a Prismatic Joint	370

8.3.5	Multibond Graph of a Three Degrees of Freedom Robot . .	372
8.3.6	Causalities in Multibond Graphs	372
8.4	The Joint Coordinate Method	374
8.4.1	Formulation of a Reduced Set of Equations of Motion . . .	376
8.4.2	Reduction of the Equations of Motion: Transformation of I Stores in the Bond Graph	377
8.4.3	Deriving the Reduced Form of Equations of Motion from the Bond Graph	379
8.4.4	Application of the Procedure to a Planar Pendulum	380
8.5	Software for Modelling and Simulation of Multibody Systems . . .	385
8.6	Conclusion	385
	References	387
9	Bond Graph Approximation of Distributed Parameter Models	391
9.1	Approximation of a One-dimensional Distributed Parameter Model by an Oscillator Chain	392
9.2	Brief Survey of Bond Graph Approximations of Distributed Parameter Models	394
9.3	Modal Analysis	394
9.3.1	The Bernoulli-Euler Beam	394
9.3.2	A Modal Bond Graph Model of the Bernoulli-Euler Beam	397
9.3.3	State Space Approximation	398
9.3.4	Features of the Generic Modal Beam Bond Graph Model .	399
9.3.5	Further Aspects of the Generic Modal Beam Bond Graph Model	400
9.3.6	Flexible Mechanical Structures	405
9.4	Finite Element Method	407
9.4.1	Classical Finite Element Method Revisited	408
9.4.2	Bond Graph Representation of Finite Element Models . . .	412
9.5	Conclusion	418
	References	421
10	Bond Graph Modelling of Open Thermodynamic Systems	425
10.1	Modelling Thermodynamic Systems by Pseudo Bond Graphs	426
10.1.1	Pseudo Bond Graph of a Heated Stirred Tank	427
10.1.2	Pseudo Bond Graph of a Variable Pneumatic Control Volume	431
10.1.3	Pseudo Bond Graph of a Compressible Fluid Flow Through an Orifice	434
10.1.4	Pseudo Bond Graph of a Pneumatic Bridge Circuit	436
10.2	True Bond Graph Models of Thermodynamic Systems	438
10.2.1	True Bond Graph of a Variable Pneumatic Control Volume	438
10.2.2	True Bond Graph of a Pneumatic Outlet Orifice	445
10.2.3	Further True Bond Graph Approaches to the Modelling of Thermodynamic Systems	446

10.2.4	Bond Graph of a Double Acting Hydraulic Cylinder	450
10.2.5	Flow Forces in Hydraulic Spool Valves	456
10.3	Conclusion	464
	References	467
11	Automated Modelling	469
11.1	Continuous System Simulation Languages	470
11.2	Object-Oriented Modelling Languages	478
11.2.1	Connection of Submodels According to the Physical Structure of the System	479
11.2.2	Algebraic Loops	480
11.2.3	Algebraic Dependencies Between State Variables	480
11.3	Bond Graph Modelling from an Object-Oriented Point of View	482
11.4	Describing Bond Graphs in SIDOPS	484
11.5	Describing Bond Graphs in Modelica	490
11.5.1	Bond Graph Power Ports and Their Interconnection	490
11.5.2	Basic Bond Graph Elements	495
11.5.3	Computational Causality	498
11.5.4	Hierarchical Bond Graphs	500
11.6	Software for Bond Graph Modelling	505
11.6.1	ENPORT™	506
11.6.2	TUTSIM™	508
11.6.3	Bond Graph Preprocessors	511
11.6.4	Bond Graph Toolboxes	514
11.6.5	Integrated Modelling and Simulation Environments	515
11.6.6	Transformation Between Different Model Description Forms	530
11.7	Exchange and Reuse of Bond Graph Models	535
11.7.1	Useful XML Features for the Description and Processing of Bond Graph Models	536
11.7.2	Information that an Exchange Format for Bond Graphs Should Capture	537
11.7.3	A Schema for an XML Based Description of Combined Bond Graph and Block Diagram Models	538
11.7.4	Pseudo Bond Graphs in BGML	546
11.7.5	Controlled Junctions in BGML	546
11.7.6	Supporting the Exchange and Reuse of Submodels	547
11.7.7	Transforming the BGML Description of a Bond Graph Model into a Target Language	549
11.7.8	XML Based Bond Graph Component Model Libraries	551
11.8	Conclusion	554
	References	556

12 Applications 561

 12.1 Inverted Pendulum 561

 12.2 Shunt Motor 566

 12.3 A Machine with an Unbalanced Rotor 570

 12.4 An Electronic Balance with Displacement Compensation 575

 12.5 A Piezoelectric Seismometer 582

 12.6 Engagement of a Clutch 589

 12.7 Dry Friction in a Suspension Strut of a Car 593

 12.8 A Buck Converter 601

 12.9 A Two Degrees of Freedom Rotary Joint Manipulator 610

 12.10 Fluid Level Control in a Three Tank System 616

 12.11 Fault Detection in a Hydraulic Two Tank System 621

 12.12 Heated Stirred Tank 629

 12.13 A Counterflow Heat Exchanger 633

 12.14 Conclusion 638

 References 639

13 Overall Conclusion and Outlook 643

 References 648

Glossary 651

Index 657

Abbreviations

3D	Three dimensional
ARR	Analytical Redundancy Relation
BDF	Backward Differentiation Formula
BD	Block Diagram
BG	Bond Graph
BGML	Bond Graph Markup Language
CR	Constitutive Relation
CSSL	Continuous System Simulation Language
DAE	Differential-Algebraic Equation
DTD	Document Type Definition
DOF	Degree Of Freedom
EJS	Eulerian Junction Structure
FDI	Fault Detection and Isolation
FEM	Finite Element Method
FFT	Fast Fourier Transform
FSM	Finite State Machine
GUI	Graphical User Interface
HBG	Hybrid Bond Graph
I/O	Input-Output
IVP	Initial Value Problem
LFT	Linear Fractional Transformation form
LTE	Local Truncation Error
LTI	Linear Time Invariant system
MBS	Multibody System
MIMO	Multiple Input Multiple Output system
NP	Nondeterministic Polynomial
ODE	Ordinary Differential Equation
OOM	Object-Oriented Modelling
OOP	Object-Oriented Programming
PDE	Partial Differential Equation
PID	Proportional, Integral and Derivative

PRV	Pressure Relief Valve
SCAP	Sequential Causality Assignment Procedure
XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 A Historical Survey of Bond Graph Modelling

When designing a new dynamic system or analysing an existing one, it is common for designers and engineers to use graphical representations of their models in order to communicate with others, to exchange ideas, to express modelling assumptions and to exchange their models for reuse. This is not surprising, as graphical representations are far more suited to human perception than oral or textual ones. For instance, a schematic of a closed loop controlled system is clearly more easily understood than a verbal description given over the telephone. In engineering disciplines, e.g., network representations, block diagrams, or linear graphs have a long tradition. In addition, (domain specific) iconic diagrams have become popular. If graphical representations adhere to formal rules, not only do they avoid misunderstandings between human beings, but they also allow for an automatic transformation into an executable program by means of appropriate software programs.

Among several graphical representation means used in different application areas, bond graphs are a description formalism best suited for modelling physical processes and multidisciplinary dynamic engineering systems including effects or components from different energy domains, viz., the mechanical, the electrical, the thermal, and the hydraulic domain. Many technical systems, often termed *mechatronic* systems, integrate components from different disciplines and exploit interacting effects, e.g., sensors and electronically controlled actuators.

Bond graphs, to be formally introduced in the next chapter, were devised by Professor Henry Paynter¹ at the Massachusetts Institute of Technology (MIT) in 1959. His former Ph.D. students, Professor D. Karnopp and Professor D. Margolis (University of California at Davis), and Professor R. Rosenberg (Michigan State University, East Lansing, Michigan), elaborated the concept into a methodology for physical systems modelling that nowadays is used in academia and industry by many people around the world. In retrospect, Paynter wrote in 1992 ([35] p. 13):

¹ 1923–2002

So it was that on April 24, 1959, as the writer was about to give a seminar lecture at Case Institute (now Case-Western) on “Interconnected Engineering Systems”, he awakened earlier that morning with the 0,1-junctions somehow finally planted in his head! Thus on that date the BG² system was complete and constituted a formal discipline.

The first published books of these pioneers include Paynter’s historic lecture notes titled *Analysis and Design of Engineering Systems*, dating back to the year 1961 [34], the book titled *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics* by Karnopp and Rosenberg [22] published in 1968, as well as the first edition of the textbook *System Dynamics – A Unified Approach* [23]. This book has become a widely recognised standard. A second edition was published in 1990 and a third edition in 2000, both co-authored by D. Margolis. In 2006, the three authors published an even more mature fourth edition that reflects their experience over decades in teaching courses at universities and in industry. This textbook now is titled *System Dynamics - Modeling and Simulation of Mechatronic Systems* [25].

Early promoters of the new modelling technique were Professor J. Thoma (Professor Emeritus at the University of Waterloo, Ontario, Canada), Professor J. J. van Dixhoorn (University of Twente, Enschede, Netherlands who passed away in 2001), P. Dransfield, Professor at Monash University, Melbourne, Australia and S. Scavarda, Professor at Institut National des Sciences Appliquées de Lyon (INSA), France who passed away in 2008. These gentlemen significantly contributed to the promotion and dissemination of the bond graph modelling technique in Europe, Japan, India and China. Van Dixhoorn founded a Technical Committee on Bond Graph Modelling (TC 16) as part of the International Association for Mathematics and Computers in Simulation (IMACS) that was chaired by J. Thoma for many years.

Right from its beginning, the bond graph methodology was supported by the famous ENPORT™ simulation program developed by R. Rosenberg. Its probably best known version has been ENPORT-4™. At that time, bond graphs were entered in alpha-numerical form in a so-called line code. Entries in that line code separated by commas denoted the type of a bond graph vertex followed by the numbers of the edges attached to that node. In his Ph.D. thesis [20], Professor J. Granda (California State University at Sacramento) developed a bond graph preprocessor that transformed the line code into equations for input into widely used simulation programs such as ACSL™³. In a further obvious step, he replaced the line code entry by a graphical editor.

Nowadays, several facts demonstrate the worldwide acceptance and the success of bond graph modelling methodology. During the first years of bond graph methodology development, almost everyone concerned with the new technique knew almost all publications and moreover, knew many members of the small but worldwide community of bond graph modellers personally. Today, the number of bond

² BG means Bond Graph

³ ACSL, acsIX, and PowerBlock are registered trademarks of The AEGIS Technologies Group, Inc., 631 Discovery Drive, Huntsville, AL 35806, USA, <http://www.acslx.com>

graph related publications has grown tremendously so that a comprehensive survey is almost impossible. In 1977, V. Gebben [19] published the first *Bond Graph Bibliography* with the aim of recording the enormous increase in the number of bond graph related publications. Updates followed in 1985 [6] and 1991 [8], both published in the renowned *Journal of the Franklin Institute*. Around 1996, Professor F. Cellier, now with the Swiss Federal Institute of Technology, Switzerland, took on the tremendous burden of compiling references to bond graph related publications in a Bond Graph Compendium available on the World Wide Web [12]. Even this comprehensive compendium needs another update. For instance, in the year 2000 alone, several textbooks on bond graph modelling were published [3, 13, 24, 30, 41]. Some more recently published books on bond graph modelling are [25, 31, 37, 43].

In the past decades, bond graph researchers contributed many special sessions on bond graph modelling to international conferences. The author organised such special sessions as part of the ESM 1993, Lyon, France, of the 1994 and the 2003 Mathmod conference in Vienna, Austria, as part of the ESS 1997 in Passau, Germany and contributed to bond graph sessions organised by other members of the community, e.g., to CESA 1996 in Lille, France, and the CIFA 2008 in Bucarest, Romania. As general chairman, the author of this book organised the 2006 European Conference on Modelling and Simulation (ECMS 2006) held near Bonn, Germany [5]. This conference featured a well received track with three sessions devoted to bond graph modelling. Professor Cellier delivered a keynote speech by addressing his current research activities in bond graph modelling. During the last decade, many more conferences with papers, sessions, or even tracks addressing bond graph modelling took place. Space in this introduction does not allow for all of them to be listed.

Besides publications in international conferences, bond graph researchers have contributed to special journal issues, e.g., the 1999 special issue of *Simulation Practice and Theory* edited by J. U. Thoma and H. J. Halin [42], the 2002 special issue of the *Proceedings of the Institute of Mechanical Engineers* edited by P. Gawthrop and S. Scavarda [17], the 2006 special issue of the journal *Mathematical & Computer Modelling of Dynamical Systems* edited by I. Troch, W. Borutzky and P. Gawthrop [44], or the 2009 special issue of the journal *Simulation Modelling Practice and Theory* edited by the author [4]. The latter special issue also includes an introduction of bond graph modelling by the author of this book. In 2007, Gawthrop and Bevan published a tutorial introduction into bond graph modelling for control engineers in the IEEE Control Systems Magazine [16].

Last but not least, the biannual *International Conference on Bond Graph Modelling* (ICBGM) as part of the Western Multiconference (WMC) of the Society for Modelling and Simulation International (SCS) has to be mentioned.

In addition to research, bond graph based physical systems modelling has been considered more and more to be an important fundamental topic in engineering education and has become a regular part in the syllabi of many engineering programmes. In addition to the 2000 textbooks already mentioned above, other textbooks on bond graph modelling have been published in different languages during the past decades, e.g. [2, 7, 14, 18, 23, 32, 40, 46]. This list, which is not meant to

be comprehensive, shows that there has been a growing awareness and acceptance of the bond graph modelling technique worldwide during the past decades. In his contribution to the March, 1995 special issue on simulation in engineering of the journal *SIMULATION*, Professor F. Cellier points out why bond graphs are “the right choice for educating students in modelling continuous-time physical systems” [11].

Not only in research and education, but also in industry, bond graph modelling has become a useful approach in the everyday business of many engineers in small consulting firms as well as in big companies, especially the automobile industry, in aerospace and in automation. In the context of mechatronic system design, the appropriateness of a bond graph approach is particularly evident. On the other hand, it must be mentioned that bond graph modelling is one of a number of modelling methodologies that are equally well suited for a given design task. People in academia and industry do have their preferences. Consequently, bond graph modelling is not appreciated to the same extent in different places. Although bond graph modelling has spread from MIT to many places all over the world, there are engineering departments even in the USA where this technique is still not used.

In some countries, bond graph modellers founded national associations, e.g., the *Bondgraafclub* in the Netherlands, or the *Club de Bondgraphistes* with members in France and in Belgium.

In 2000, the first edition of this monograph was published as a first comprehensive presentation of bond graph methodology in the German language. The author of this monograph became aware of bond graphs in 1979 by a survey article by A. Schöne [38]. It was the starting point of an ever-continuing enthusiasm for this methodology that I share with friends in many places around the world.

1.2 Some General Aspects of Modelling Dynamic Systems

Building a model is an iterative procedure. It starts with the identification of essential features and of inherent mechanisms of a dynamic system to be designed. In a step by step refinement of the understanding of a dynamic physical system, different forms of representation are used. They are of graphical nature, especially during the conceptual phase as our eyes can easily perceive different information in parallel. Graphical representations are not only easy to grasp, they are also best suited for communication between individuals. This becomes evident if one considers the difficulty of transmitting all the information contained in a schematic of an electronic or hydraulic circuit by telephone, or if one receives the description of a dynamic system in a simulation language on many pages without any comments. The importance of graphical representations is not only essential in the modelling phase, but also for visualisation of simulation results. With the ever increasing computing power, not only is the graphical representation of numerical results as curves required, there is also a demand for 3D animation of system motion in a realistic fashion. With the development of languages such as Virtual Reality Modelling Language (VRML) and

appropriate freely available visualisation software tools, 3D presentations may be exchanged via the internet. Thus, graphical representations and visualisation play a vital role during the design of a dynamic system and the design of its control [36].

Graphical and textual model representations of dynamic systems are always associated with a certain view of a system, its properties and its inherent mechanisms. They reflect abstractions and modelling assumptions. The starting point of the modelling process always involve certain questions. Thus, features and system properties are assessed and are either taken into account or deliberately neglected. These considerations and decisions as well as the designer's experience have an impact on the choice of the graphical description. Answers to the question as to what purpose a model of a system to be designed shall serve, may give an indication to an appropriate choice of a description formalism from a repertoire of possible means. Often, the answer to the question concerning the most suitable form of representation is not unique. Several description formalisms may equally serve the requirements, or depending on the design task, a combination of some of them may be appropriate, e.g., a bond graph representation of a system and a block diagram of its control. Moreover, a graphical representation may be transformed into another one provided both are equivalent. For instance, a network could be transformed into a bond graph which in turn could be transformed into a block diagram. Consequently, a program for control systems could be used to simulate a system described as a network. There is a similar situation in computer science. For the software implementation of a solution to a given problem, several programming languages may be appropriate, whereby each one has its own flavour. On the other hand, software programs for automatic translation from one programming language to another are available.

1.3 Object-Oriented Physical Systems Modelling

The ever increasing performance of computers and simulation software has enabled one to model and to simulate problems of more and more complexity. While the simulation of a problem required hours of computation time in former times on computers called minicomputers, the same problem can be solved within minutes or less on a personal computer. The ever increasing complexity of problems to be analysed has had an essential impact on modelling methodologies and the software tools supporting them.

In order to cope with the increasing complexity of the system to be analysed, it is obviously necessary to pursue a hierarchical approach and to build libraries with reusable submodels as is known, for example, from the design of large integrated circuits. Moreover, for the development of large models, the need for a graphical representation becomes apparent. It is true that continuous system simulation languages (CSSLs), e.g., ACSL⁴ (*Advanced Continuous System Simulation Language*), offer macro features to build modularly structured models. However, instead of

⁴ ACSL, acslX, and PowerBlock are registered trademarks of The AEGIS Technologies Group, Inc., 631 Discovery Drive, Huntsville, AL 35806, USA, <http://www.acslx.com>

writing many thousands of lines of code in a more or less well structured model description language, it is more reliable to hierarchically develop a graphical model using model libraries and to have the result transformed automatically into simulation code. To that end, an easy to use graphical user interface (GUI) alone is not sufficient. What is needed is an appropriate model description language underlying the graphical model representation and its automatic transformation into a modelling language. However, as languages based on the CSSL standard established in 1967 [39] are more simulation languages than modelling languages, they show shortcomings in supporting the development of hierarchical modular structured models. Such a modelling language was described as early as 1979 in Elmqvist's dissertation [15].

On the other hand, in computer science, software projects of ever increasing size have led to the paradigm of Object-Oriented Programming (OOP). In their widely recognised 1991 book, *Object-Oriented Modelling and Design*, J. Rumbaugh and his co-authors present object-oriented modelling as a methodology for the design of large complex software systems that include the analysis of the problem, the design and the software implementation. They use the term *Object Modelling Technique* (OMT) and point out in the preface of their book that object orientation means more than merely a kind of programming. The attractiveness and the success of this fundamental concept has also had an impact on the terminology in physical systems modelling and the manner models of large systems are developed and described. Since about 1990, with the event of modelling languages such as Omola [1], the term *object-oriented physical systems modelling* has come into use. In contrast to classical control, describing a system's behaviour by functional input-output blocks and consequently focusing on the *computational structure*, object-oriented modelling emphasises the view of a model composed of *non-causal* submodels connected accordingly to the *physical structure* of the system. The new approach at that time was promoted by Anderson and Mattson [28] and F. Cellier [10]. In his dissertation [33], Otter used the modelling language Dymola⁵ [15], developed by Elmqvist in as early as 1978, for an object-oriented approach to mechatronic systems modelling. Without discussing details of the language Dymola or its successor Modelica⁶ [29], the characteristics of an object-oriented physical systems modelling approach are presented in the following.

Characteristics of Object-Oriented Physical Systems Modelling

According to principles of object-oriented programming in software engineering, object-oriented physical systems modelling may be characterised by the following features.

- **Objects**
In object-oriented physical systems modelling, models of components of engineering systems as well as models of physical processes are considered to be

⁵ Dymola[®] is a registered trademark of Dynasim AB, Ideon Science Park, SE-223 70 Lund, Sweden, <http://www.dymola.com>

⁶ Modelica[®] is a registered trademark of the Modelica Association, <http://www.modelica.org>

objects. Since basic models are comprised of inherent parameters, e.g., length, mass, moment of inertia, resistance, or data provided from the outside world through so-called *interfaces*, and constitutive equations, there is a correspondence to objects in the sense of object-oriented programming (OOP). Physical system models may be viewed as an aggregation of data and *methods* operating on them. In OOP, the term *method* means a function or a procedure that can process data of a defined type. For instance, the coordinates of a point, a length, and a function that returns the area of a circle around that point may be aggregated into a *class* called *circle*. A circle around a given point of given radius may be named *c1*. Then, *c1* is a particular object of the class *circle*. Likewise, the voltage across and the current through a resistor, parameters, and the constitutive (non-linear) relation between voltage and current may be considered a class *resistor* corresponding to the element type resistor. A copy of the class *resistor* with given values for the parameters in the constitutive relation called, for instance, *R5*, corresponds to a particular resistor in a circuit.

- Model Hierarchy

As it is well known, a model of a system may be composed of lower level submodels which in turn may contain submodels as well. That is, physical-system models are *hierarchical* in nature (in the sense of a membership relation).

- Model Classes and Instantiation

As explained above, physical system models and submodels of components can be viewed as particular objects of a certain class. In object-oriented modelling, the term *instantiation* is adopted from OOP. A model or submodel is called an *instance* of a model class. Models or submodels are *instantiated* from generic models or *model classes* in which more general properties common to the members of a model class are described. The members of a model class have got the same structure and exhibit the same general dynamic behaviour. For example, an instance of the model class *diode* is obtained by giving particular values to its parameters. The resulting instance corresponds to a particular diode in a circuit.

- Inheritance

If a submodel class is instantiated into a particular submodel, its properties are inherited by the submodel. That is, the particular submodel declaration is a special version of the more general submodel declaration. In Section 11.5.2, an incomplete model class is introduced that only captures the energetic property of a 1-port energy storage element in the sense that this type of element stores a physical quantity such as electrical charge. In that class, a relation between the state and the rate variable is not given. By adding this information, a particular class can be derived. The more special model class inherits all properties of the superclass from which it is derived. Consequently, it describes a particular type of an energy store. Again, by specifying parameters, a particular object or instance of this class is obtained associated with a particular energy store in the system. Another example is an incomplete model class *passiveOnePort* that accounts for the passivity of a 1-port element. A special class *diode* may be obtained from this incomplete superclass by specifying the constitutive relation as that of a diode. Obviously, the subclass inherits the passivity property of a 1-port from the su-

perclass. Using inheritance in the declaration of library model classes has the advantage that the potential of errors in the code is reduced. All features of the superclass are copied. Only features that characterise the more special class need to be added.

- Encapsulation

Knowledge contained in a model is encapsulated. Only a well defined part of it can be accessed in a well defined manner via interfaces to the outside world of an object. This means that the internal definition of a submodel is not affected by the connection of the submodel to other submodels. The part of a submodel describing its interfaces is separated from the part in which the behaviour is described by means of *non-causal* mathematical equations. The latter do not need to be known when submodels are connected in order to build a hierarchical model. In generalised networks for instance, an interface of a submodel is a pin. It is described by two power variables called *across* and *through* variables.

- Polymorphism

In a submodel definition, the description of its interfaces to the outside world is separated from the internal description of its dynamic behaviour. In this internal description, multiple possible cases may be taken into account. Consequently, depending on current conditions, the same submodel may exhibit different behaviour.

- Connection of Submodels

Submodel interfaces are connected accordingly to component interconnections in the real physical system, also called the *physical structure* of the system. In contrary to block diagrams, this means that equations of a submodel must be *non-causal*. The interconnection of submodels may require solving them for certain variables. In block diagrams, it is fixed a priori whether an interface variable of a block is an input or an output variable. This must be taken into account when connecting blocks.

Of course, it has been a tradition in physical systems modelling to build hierarchical models and to connect submodels according to the physical structure long before object-oriented modelling was introduced. In Chapter 11, aspects of object-oriented physical systems modelling will be picked up again when looking at bond graph modelling from an object-oriented modelling point of view.

1.4 Traditional Graphical Model Representations

There are many forms of graphical model representation in use in different engineering disciplines, e.g., free body schematics in mechanical engineering, circuit representations in electrical engineering and in hydraulics, linear graphs as well as block diagrams and signal flow diagrams in control theory. All of these forms are well known and have a long tradition. Therefore, only some aspects of block diagrams and signal flow graphs on the one hand, and network representations on the other hand, will be outlined. The aim is to show the context in which bond

graph methodology is embedded. In the following, aspects are discussed from a methodology point of view. In Chapter 11, links to corresponding software tools are considered.

1.4.1 Block Diagrams

Block diagrams have the following characteristics:

- They support the abstraction of unilateral signal flow through a system. Information flow is considered not to be bound to the transfer of energy or the transport of mass. A block in an oriented chain of blocks does not have a direct impact on its predecessor. Feedback is taken into account separately by signal feedback loops.
- The interface variables of a signal block are a priori discriminated into input and output signals, independent of the actual use of a signal block in a block diagram model. This must be taken into account when signal blocks are connected.
- Signal blocks represent functional relations between input and output signals. It is neither required nor ensured that relations comply with first principles of physics. Blocks may represent any linear, or nonlinear algebraic, or time dependent relation.
- Block diagrams display which variables must be known in order to compute others. They represent the structure of the mathematical model, or as van Dixhoorn [45] has termed it, the *computational structure*. They do not reflect the *physical structure* of a system. The reason is that feedback is represented by *separate* feedback loops. Signal blocks cannot be connected like corresponding system components. For instance, if two electrical devices are connected, then the voltages are set to be equal and at the same time, currents are added to zero. In block diagrams, however, a connection between two blocks represents only one signal. As will be pointed out later, in bond graphs, each edge is associated with two conjugate power variables. Consequently, connected bond elements always have a feedback to each other. If submodels in a block diagram are modified by neglecting effects or by taking into account additional ones, then small changes may have a considerable impact on the computational structure and thus on the structure of the block diagram. This disadvantage does not appear in networks or in bond graph graphs.
- As block diagrams represent signal flows and functional relations independent of the physical meaning of variables, they can be used in different engineering disciplines. They are used particularly in control since control systems are often designed in such a manner that there is no feedback between components. The computational structure then corresponds to the physical structure.
- Finally, block diagrams support a hierarchical decomposition into functional blocks.

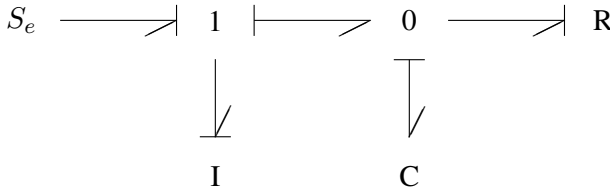


Fig. 1.1 Causally completed bond graph model of a second order system

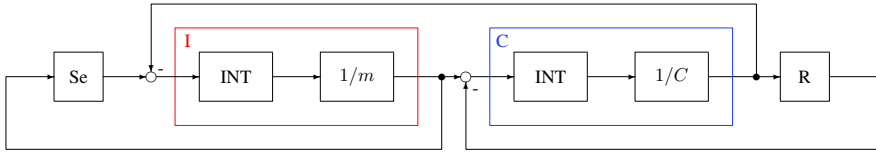


Fig. 1.2 Block diagram corresponding to the bond graph in Figure 1.1

As will become clarified, bond graphs reflect the physical structure of a system, as do networks. On the other hand, the computational structure may be superimposed on a bond graph by adding a perpendicular stroke to each edge, turning the initially non-causal model into a causal one. Moreover, such a causally completed bond graph can be systematically transformed into a block diagram if needed. During this transformation, information about the physical structure gets lost, as can be seen from Figures 1.1 and 1.2. As signal blocks in block diagrams can represent any functional relation, the converse does not hold. Not every block diagram can be transformed into a bond graph. Equations represented by bond graphs should comply with the first principles of physics.

1.4.2 Signal Flow Graphs

In signal flow graphs, the role of edges and vertices is essentially interchanged in comparison to block diagrams. Oriented edges represent functional relations between variables, while nodes are used to represent variables and the summation of variables. In that respect, they may be considered the dual of block diagrams. However, signal flow graphs are less general than block diagrams because besides the summation of variables, no functions with more than one input variable can be represented. Like block diagrams, signal flow graphs represent the computational, not the physical structure of a system. In the case of linear models, they can be used to derive the transfer function between two variables by applying Mason’s loop rule [27]. Bond graphs can also be transformed into signal flow graphs as shown in Fig-

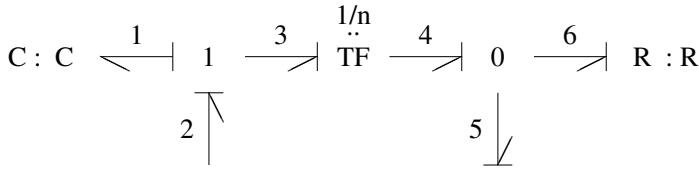


Fig. 1.3 Causally completed bond graph fragment

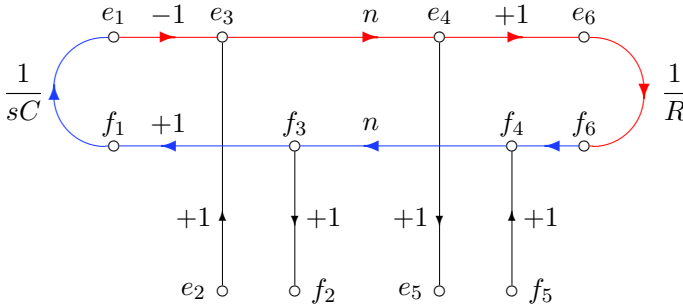


Fig. 1.4 Signal flow graph corresponding to the bond graph fragment in Figure 1.3

ures 1.3 and 1.4. However, as Brown has shown, Mason’s loop rule can be applied directly to a bond graph [9].

1.4.3 Networks

In electrical engineering, it is common to represent models as networks. However, this representation is not restricted to electrical systems. By relating the electrical power variables voltage and current to non-electrical quantities appropriately, networks may be used to represent models of systems in other energy domains. A unified approach to the modelling of engineering systems based on so-called *generalised networks* has been introduced by MacFarlane [26]. Networks have the following essential features.

- Contrary to block diagrams, networks represent the physical structure, but not the computational structure of an engineering system. Submodels are connected like corresponding components or devices in the real system. There is no need to decide whether an interface variable is an input or an output variable. The graphical representation of submodels is not uniform across energy domains as in bond graphs, but depends on the engineering discipline. For instance, hydraulic circuits use different icons for submodels than electrical circuits. In any case,

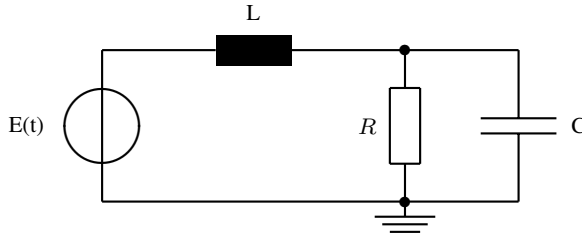


Fig. 1.5 Circuit with two energy stores

Kirchhoff's laws hold for electrical power variables as well as for corresponding variables in other energy domains.

- Networks are hierarchical in nature. Submodels can have a network structure.
- Networks account for energy flows in a system. Circuit nodes connecting pins of submodels comply with power conservation. Moreover, physical quantities, e.g., charge, are conserved if properly taken into account in the development of submodels. Meyer's NMOS transistor model, for instance, does not ensure conservation of charge ([21], Section 3.4.4)

Bond graphs, which will be introduced formally in the next chapter, on the one hand, reflect the physical system structure like networks. On the other hand, a computational structure can be superimposed so that the causally completed bond graph can be considered a concise representation of a block diagram. As bond graphs have features in common with block diagrams and with networks, both representations have been briefly discussed in this introduction. Moreover, bond graphs can be considered a core model representation for the following reasons.

- Generalised networks can be systematically converted into bond graphs (cf. Section 2.7). If orientations of edges in a bond graph indicating the reference direction of the energy flow across a bond are chosen with care, then the directed bond graph is equivalent to the network from which it has been constructed (Section 2.8). For instance, if the circuit in Figure 1.5 is converted into a bond graph, then the result is the bond graph in Figure 1.3.
- From a causally completed bond graph, a block diagram as well as a signal flow graph can be derived.
- Finally, domain-specific iconic diagrams can be systematically converted into a bond graph if there are bond graph equivalents of basic icons.

1.5 Conclusion

Dynamic system modelling has a long tradition. On the other hand, the views of model developers, methods and corresponding software tools change, and new is-

issues and new applications have been tackled. Due to the ever increasing power of computers and even of mobile computers, imitation, or the prediction of reality is becoming steadily less expensive in more and more fields. Even if experiments with the real system under consideration are feasible, less expensive computer simulation will reduce them to some extent. In this context, many publications on dynamic systems modelling have already appeared and many more are to be expected in the future. This introduction has focused on some general aspects of dynamic systems modelling. Essential features of block diagrams and of networks have been briefly recalled because bond graphs combine characteristics of both graphical model representations. In summary, one can say:

- Graphical model representation is a means of communication between humans and between humans and computer programs. The choice of an appropriate model representation depends on the purpose of the modelling process, the questions it helps to answer.
- In the process of abstraction, some properties of a real system and some effects are taken into account in an idealised manner, while others are completely neglected. The assessment of properties and their selection is guided by the purpose of the modelling task and by the experience of the model developer. Consequently, graphical model representations always reflect only some aspects. Thus, block diagrams appropriate for representing signal processing represent the computational structure, but not the physical structure of a system, whereas on the other hand, electrical, or hydraulic networks display how corresponding real system components are connected at the expense of the computational structure.

Graphical model representations are particularly appropriate for some purposes while they are less suited for others. Some may be converted into others, e.g., a network of an electric circuit can be converted into a block diagram. A possible reason for such a conversion may be that the available modelling software does not support a combination of different model representations. The choice of a graphical model representation formalism in dynamic modelling is similar to the choice of a programming language in a software development project. Depending on the task, some programming languages are more suited than others. Often, several programming languages may serve the requirements. Moreover, some programming languages can be translated automatically into others. Since graphical model representation formalisms are not equally suited for all applications and purposes, it may be reasonable to use a combination of them.

Bond graphs fit into the spectrum of graphical model representations as a formalism that is particularly suited for engineering systems with effects from multiple energy domains. Consequently, they are an ideal representation for mechatronic and for micro-mechanical systems. Moreover, they not only support model development as some kind of visual language, they have also been proven as an appropriate core representation. That is, several other graphical representations can be converted into bond graphs, while from causal bond graphs graphical representations, e.g., block diagrams, equations, transfer functions and other information can be derived sys-

tematically. This motivates the introduction of bond graph based physical systems modelling in the next chapter.

References

- [1] M. Andersson. Omola – An Object Oriented Language for Model Representation. Master's thesis, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1990. Licentiate thesis TFRT-3208.
- [2] P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella, and I. Zambettakis. *Modélisation et Identification des Processus*, volume tome 2. Éditions Technip, Paris, 1992.
- [3] W. Borutzky. *Bond graphs – A Methodology for Modelling Multidisciplinary Dynamic Systems*. Frontiers in Simulation. SCS European Publishing House, Erlangen, Ghent, 2000. ISBN: 1-56555-183-4 (in German).
- [4] W. Borutzky, editor. *Special Issue Bond Graph Modelling*, volume 17/1 of *Simulation Modelling Practice and Theory*, 2009. Elsevier.
- [5] W. Borutzky, A. Orsoni, and R. Zobel, editors. *Proc. of the 20th European Conference on Modelling and Simulation*, 2006. European Council for Modelling and Simulation. Hard Copy: ISBN: 0-9553018-0-7, CD-ROM: ISBN: 0-9553018-1-15.
- [6] A.M. Bos and P.C. Breedveld. Update of the Bond Graph Bibliography. *Journal of the Franklin Institute*, 319(1/2):269–286, Jan./Feb. 1985.
- [7] P.C. Breedveld and J. van Amerongen. *Dynamische systemen: modelvorming en simulatie met bondgrafen*. Open universiteit, Heerlen, The Netherlands, 1994.
- [8] P.C. Breedveld, R.C. Rosenberg, and T. Zhou. Bibliography of Bond Graph Theory and Application. *Journal of the Franklin Institute*, 328(5/6):1067–1109, 1991.
- [9] F.T. Brown. Direct Application of the Loop Rule to Bond Graphs. *Journal of Dynamic Systems, Measurement and Control*, pages 253–261, September 1992.
- [10] F.E. Cellier. *Continuous System Modeling*. Springer-Verlag, New York, Berlin, Heidelberg, 1991. ISBN: 0-387-97502-0.
- [11] F.E. Cellier. Bond Graphs: The Right Choice for Educating Students in Modeling Continuous-Time Physical Systems. *SIMULATION*, 64(3):154–159, March 1995.
- [12] F.E. Cellier. World Wide Web – The Global Library: A Compendium of Knowledge About Bond Graph Research. In J.J. Granda and G. Dauphin-Tanguy, editors, *1997 International Conference on Bond Graph Modeling and Simulation (ICBGM '97)*, volume 29(1) of *Simulation Series*, pages 187–191. SCS, 1997. URL <http://www.inf.ethz/personal/fcellier/BondGraphs/bg.html>.
- [13] G. Dauphin-Tanguy. *Les bond graphs*. Hermes Science Europe Ltd., Paris, France, 2000. ISBN: 2-7462-0158-5.
- [14] P. Dransfield. *Hydraulic Control Systems – Design and Analysis of Their Dynamics*. Springer-Verlag, New York, 1981.
- [15] H. Elmqvist. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1978. Report CODEN: LUTFD2/(TFRT-1015)/1-226/(1978).
- [16] P.J. Gawthrop and G.P. Bevan. Bond Graph Modeling. *IEEE Control Systems Magazine*, pages 24–45, April 2007.
- [17] P.J. Gawthrop and S. Scavarda, editors. *Special Issue on Bond Graphs*, volume 216(1) of *Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering*, London, UK, 2002. Professional Engineering Publishing.
- [18] P.J. Gawthrop and L. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [19] V. D. Gebben. Bond Graph Bibliography. *Trans. ASME Journal of Dynamic Systems, Measurement, and Control*, 99(2):143–145, 1977.

- [20] J.J. Granda. *Computer-aided modelling program (CAMP): a bond graph preprocessor for computer-aided design and simulation of physical systems using digital simulation languages*. PhD thesis, Univ. of California, Davis, 1982.
- [21] E.-H. Horneber. *Simulation elektrischer Schaltungen auf dem Rechner*. Springer-Verlag, 1985.
- [22] D.C. Karnopp and R.C. Rosenberg. *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics*. MIT Press, Cambridge, MA, 1968.
- [23] D.C. Karnopp and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1975.
- [24] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics - Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons Inc., Third edition, 2000. ISBN 0-471-33301-8.
- [25] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics - Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons Inc., Fourth edition, 2005. ISBN: 0-471-70965-4.
- [26] A.G.J. MacFarlane. *Engineering Systems Analysis*. Harrap, London, UK, 1964.
- [27] S. J. Mason. Feedback Theory: some properties of signal flow graphs. In *Proc. IRE 41*, pages 1144–1156, 1953.
- [28] S.E. Mattsson. Object-Oriented Modelling of a Real Continuous-Time System. In J. Stephenson, editor, *Proc. of the 1992 European Simulation Multiconference*, pages 241–245, San Diego, CA, 1992. SCS. York, UK, June 1992.
- [29] Modelica Association. Modelica and the Modelica Association. URL <http://www.modelica.org>.
- [30] A. Mukherjee and R. Karmakar. *Modelling and Simulation of Engineering Systems through Bondgraphs*. Narosa Publishing House, New Delhi, India, 2000. ISBN: 81-7319-279-0.
- [31] A. Mukherjee, R. Karmakar, and A.K. Samantaray. *Bond Graph in Modeling, Simulation and Fault Identification*. I.K. International Publishing House, New Delhi, India, 2006. ISBN: 81-88237-96-5.
- [32] S.T. Nannenberg. *Bondgraaftechniek*. Delta Press, Overberg, Niederlande, 1987.
- [33] M. Otter. *Objektorientierte Modellierung mechatronischer Systeme am Beispiel geregelter Roboter*. PhD thesis, Ruhr-Universität Bochum, 1994.
- [34] H.M. Paynter. *Analysis and Design of Engineering Systems*. M.I.T. Press, Cambridge, Massachusetts, USA, 1961.
- [35] H.M. Paynter. An Epistemic Prehistory of Bond Graphs. In P.C. Breedveld and G. Dauphin-Tanguy, editors, *Bond Graphs for Engineers*, pages 3–17. North-Holland, 1992.
- [36] G. Romero, J. Félez, J. Maroto, and J.M. Mera. Efficient simulation of mechanism kinematics using bond graphs. *Simulation Modelling Theory and Practice*, 17/1:293–308, 2009.
- [37] A.K. Samantaray and B. Ould Bouamama. *Model-based Process Supervision – A Bond Graph Approach*. Advances in Industrial Control. Springer, London, 2008. ISBN 978-1-84800-158-9.
- [38] A. Schöne. Systembeschreibung durch simultane Diagramme. *Regelungstechnik*, 24(2):37–72, 1976.
- [39] J.C. Strauss, D.C. Augustin, M.S. Fineberg, B.B. Johnson, R.N. Linebarger, and F.H. Sanson. The SCi Continuous System Simulation Language (CSSL). *SIMULATION*, pages 281–303, Dec. 1967.
- [40] J.U. Thoma. *Simulation by Bondgraphs – Introduction to a Graphical Method*. Springer-Verlag, New York, 1990.
- [41] J.U. Thoma and B. Ould Bouamama. *Modeling and Simulation in Thermal and Chemical Engineering (A Bond Graph Approach)*. Springer-Verlag, Berlin, 2000.
- [42] J.U. Thoma and H.J. Halin, editors. *Special Issue Bondgraphs for Modeling and Simulation*, volume 7(5–6) of *Simulation Practice and Theory*, 1999. Elsevier.
- [43] J.U. Thoma and G. Mocellin. *Simulation with Entropy in Engineering Thermodynamics*. Springer, Berlin, Heidelberg, New York, 2006. ISBN -10 3-540-32798-3.

- [44] I. Troch, W. Borutzky, and P.J. Gawthrop, editors. *Mathematical & Computer Modelling of Dynamical Systems, Special Issue: Bond Graph Modelling*, volume 12 (2–3). Taylor & Francis, April-June 2006.
- [45] J.J. van Dichoorn. Bond graphs and the challenge of a unified modelling theory of physical systems. In F.E. Cellier, editor, *Progress in Modelling and Simulation*, pages 207–245. Academic Press, New York, 1982.
- [46] M. Vergé and D. Jaume. *Modélisation structurée des systèmes avec les Bond Graphs*. Edition Technip, 2003. ISBN: 2-7108-0838-2.

Chapter 2

Bond Graph Based Physical Systems Modelling

2.1 Fundamentals

Block diagrams represent signal flows and their processing. In contrast, bond graph modelling starts from the intuitive and physical approach that a dynamic system is composed of subsystems, components, or basic elements that interact by exchanging energy. From this basic description, a first (preliminary) definition of a bond graph can be derived and some important conclusions can be drawn which show that bond graph modelling fundamentally differs from block diagram modelling.

Definition 2.1 (*Undirected bond graph*). An undirected bond graph is an undirected graph whose vertices denote subsystems, components, or basic elements, while the edges called (power) bonds represent the instantaneous energy transfer between nodes.

Remark 2.1. An undirected bond graph displays the components of a system and their energetic interconnection. Each vertex in a bond graph has a certain number of connection points called *power ports* (cf. Definition 2.3). Bonds connect power ports of two different nodes. Each power port of a vertex must be connected to a power port of another vertex. That is, the number of bonds connected to a bond graph node equals its number of power ports. Power ports are not explicitly marked on a bond graph. The graphical editor of some software packages supporting bond graph modelling can make ports visible on demand.

An instantaneous energy transfer between two power ports means that energy is neither generated, stored, or dissipated in a port to port connection represented by a power bond.

2.1.1 Physical System Structure

In general, the transfer of energy between subsystems is enabled by means of engineering links, e.g., mechanical shafts, electrical wires, hydraulic conduits, hoses,

or glass fibre optics cables. Since subsystems, components and elements are represented by bond graph vertices and their energetic interaction by power bonds, bond graphs reflect the *physical structure* of a system, the way real engineering system components are connected. As long as bond graphs are constructed according to certain rules and are not simplified, they exhibit a strong topological affinity to the initial schematic of a mechanical system, an electrical circuit, or a cross sectional representation of a hydraulic device. Therefore, topological connections in a system schematic can guide the construction of a bond graph model. This could be achieved by drawing a bond graph directly on top of a schematic. For instance, if two dead volumes in a hydraulic valve are connected by a conduit, their models will be connected by the model of the conduit. The latter most often reduces simply to a power bond. The same strategy is applied to rigid bodies connected by a joint, or to integrated sub-circuits connected by transmission lines. For illustration, Figure 2.1 shows a cross sectional view of the magnetic circuit inside the torque-motor of an electrohydraulic servovalve. The magnetic circuit is composed of an upper and a lower pole-shoe connected by two permanent magnets. The one in the front has been removed as well as the mechanical flexible tube on which the armature is mounted. Although details of bond graph modelling have not yet been introduced, it can be seen how the bond graph is superimposed on the schematic. In block diagram representations, the information about the physical structure of the systems gets lost. They rather represent the *computational* structure, which may change significantly with small model modifications. On the contrary, the structure of a bond graph model is derived from the physical structure of the system and is retained if the computational structure changes.

2.1.2 Physical Systems Modelling

The exchange of energy between subsystems is associated with the exchange of physical quantities, e.g., momentum, mass, electrical charge, or entropy. For these physical quantities and for power conservation, principles should be reflected in a bond graph representation. Since there are no such constraints for information flows, signal processing blocks in block diagrams may represent any functional relation between signals. In bond graphs, however, constitutive relations of nodes must comply with conservation laws from physics. In his short article titled “System Graphing Concepts”, Paynter [28] stresses at the beginning:

Models of physical systems must be compatible with the conservation of mass, momentum, and energy. Functional models must be compatible with causality, such that a present state depends only on the past states (or: no effect in the absence of cause).

If one assumes that subsystems exchange energy when interacting, then processes must take place in the subsystems by which energy received from a subsystem is either transferred to another subsystem, distributed among others, transformed into other forms, or stored. That is, bond graphs represent physical processes in an engineering system. This is one reason why in bond graph related literature the term

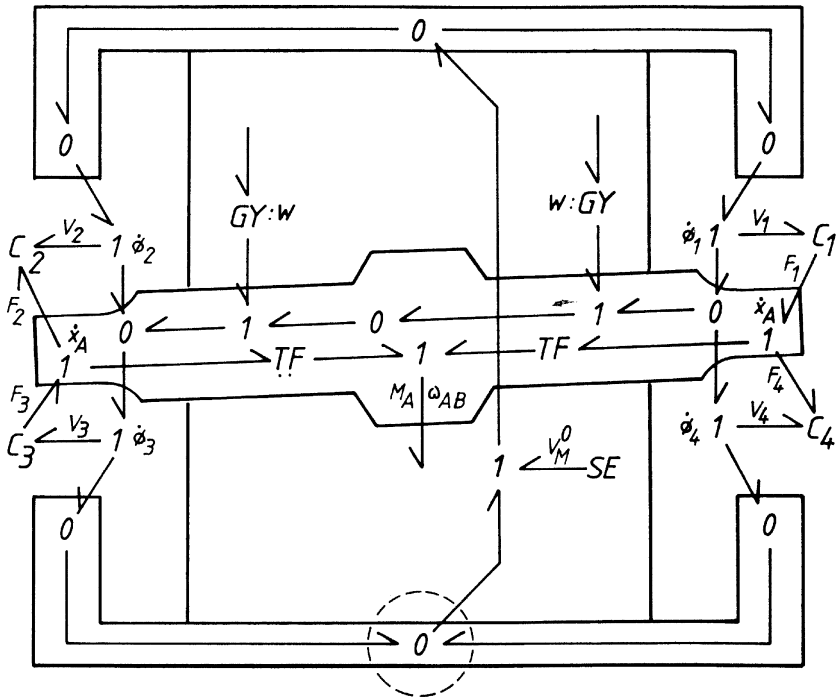


Fig. 2.1 Schematic of the magnetic circuit inside the torque motor of an electrohydraulic valve and a bond graph superimposed on it

physical systems modelling is in use [8]. Since physical processes are continuous with respect to time (and space), bond graphs are particularly suited for modelling time continuous systems. Chapter 7 discusses how the abstraction of a discontinuous description of continuous fast state transitions can be included into the framework of bond graph modelling. In this context, the observation can be made that only during the last 10 years, bond graphs have been increasingly used for models of varying structure. For many years, bond graph modelling was predominantly applied to systems in which energy exchange between subsystems is bound to time invariant real physical links, e.g., mechanical shafts.

2.1.3 Multidisciplinary Engineering Systems

The focus on energy exchange between subsystems and the transformation of energy from one form into another implies that bond graphs, from a conceptual point of view, are particularly suited for modelling *multidisciplinary* engineering systems, or

mechatronic systems in which effects from different energy domains interact with each other. This is an essential feature of bond graphs.

2.1.4 Hierarchical and Recursive Modelling

Like networks, bond graphs support a recursive top-down decomposition of a system into subsystems. That is, models of subsystems are represented by bond graphs until submodels are identified that represent basic physical processes described by equations. In this top-down approach, submodels of subsystems or components are denoted by words or alphanumeric symbols (enclosed by an ellipse). On the lowest hierarchy level the mnemonic code of elements is fixed. It indicates the behaviour of the element with respect to energy or power. As words are used for components or subsystem models, such bond graphs are called *word bond graphs* [21]. In a bottom-up approach, nodes in a word bond graph denote submodels described either by a bond graph or by a set of equations. Word bond graphs are not only a way to clearly represent large models in a hierarchical manner. They are important because they support the first steps of the conceptual modelling phase. Word bond graphs visualise the energy exchange between different parts of a system (subsystems, components, elements) that have been identified without the need for specifying all details of the system parts. At the beginning of the decomposition process, a word is sufficient that *qualitatively* indicates the behaviour of a system part with respect to power processing. Even on the lowest hierarchy level, when nodes describe physical processes and fixed mnemonic codes are used, a 1-port R-element, for instance, *qualitatively* describes dissipation of free energy. With regard to a mathematical description, there must be a relation between two so-called power conjugate variables. The decision on the form of the constitutive equation, however, can be postponed to a later phase of the modelling process.

Definition 2.2 (*Word bond graph*). A bond graph is called a word bond graph if its vertices represent subsystems or components and are denoted by a word or an alphanumeric symbol.

Remark 2.2. Mnemonic codes in a word bond graph may be enclosed by an ellipse.

As an example, Figure 2.2 shows a word bond graph of a hydrostatic plant. In the early phase of the modelling process, properties of the components are not yet specified.

Definition 2.3 (*Power port*). The connection points of a bond graph node that enable the energy exchange with other nodes across a power bond are called power ports.

Remark 2.3. Power ports may be viewed as places where energy can enter or leave a subsystem. In this context, the notion of a place is not limited to locations in space. Gawthrop and Smith also use the notion *energy interface* in their book [18]. Power

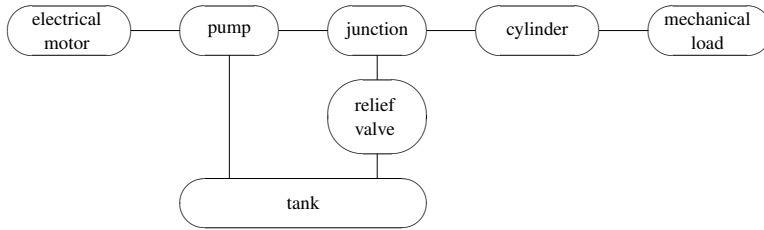


Fig. 2.2 Word bond graph of a hydrostatic plant

ports can be considered as energetic interfaces of an object (subsystem, component, element) to its outside world.

An energy flow between two power ports also involves a transfer of information. If the amount of power in a bond can be neglected with regard to other bonds and if only the flow of information shall be taken into account, which is appropriate if, for instance, a measuring instrument is connected to a system part port, then the port is called a *signal port*. In that case, the bond connecting the signal port of the system part with the instrument signal port can be replaced by an oriented edge as it is used in block diagrams. In that context, the notion of an *activated bond* is also used (cf. Definition 3.3). In the modelling language SIDOPS, ports have several properties. There is not only a distinction between power ports and signal ports, but also with respect to the energy domain. That is, an electrical power port can be only connected directly to an electrical power port of another submodel. In other words, the energy domain is an attribute of a power port among others.

Definition 2.4 (Multiport). A bond graph node is called a *multiport* if it has more than one port.

Figure 2.3 depicts an example of a general word bond graph in which S denotes the model of the overall system and S_i ($i = 1, \dots, 6$) the model of the i^{th} subsystem. According to definitions 2.1 and 2.4, models S_3, S_4 are one-ports, S_1, S_6 are two-ports, while S_5 is a three-port and S_2 is a four-port.

2.2 Nodes and Edges in Bond Graphs

As mentioned, on the lowest hierarchy level, bond graph models are called elements. They represent basic physical processes in which energy is

- distributed,
- transferred from one power port to another,
- transformed in the same energy domain,
- converted into another energy form, in particular into heat, or
- stored.

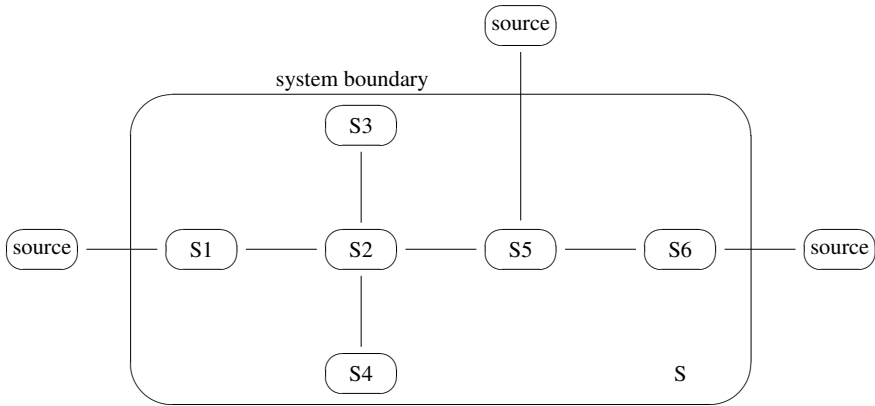


Fig. 2.3 Example of a general word bond graph

As in physical systems modelling based on networks, bond graph modelling also adopts the abstraction of spatially lumped physical properties. That is, mechanical elasticity, for instance, or friction in a fluid may be approximated by assuming that these effects are spatially concentrated in certain locations. Consequently, the above basic physical processes may be located in space and represented by a node in a bond graph. Since these elementary physical processes are encountered in all energy domains, it is reasonable to represent them by means of a unique mnemonic code that indicates the type of the process and that is the same for all energy domains. For instance, the symbol R always denotes irreversible transformation of energy into heat. Spatial concentration of physical properties means that bond graphs represent so-called *lumped parameter models*. The above processes are represented by basic bond graph elements that will be discussed when variables used in bond graphs have been introduced.

2.3 Bond Graph Variables and Physical Analogies

2.3.1 Power Variables

According to Definition 2.1, bonds in a bond graph represent an instantaneous energy flow, i.e., power between power ports of different bond graph nodes. It is a general observation that in each energy domain, the amount of power transferred equals the product of two physical quantities. Thus, contrary to block diagrams, in bond graphs two power conjugated variables are assigned to each edge. They are called *effort* and *flow* and are denoted by the letters *e* and *f*.

$$\text{Power} = \text{Effort} \times \text{Flow}$$

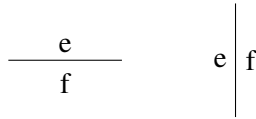


Fig. 2.4 Historical convention of annotating a bond with power variables effort and flow

This representation of power between two ports by the product of two variables, however, is not a physical law, but a convenience. The amount of power transferred can also be decomposed differently by means of so-called wave-power variables [27]. However, it is remarkable that two variables are sufficient for describing the power across a connection.

Regarding the annotation of bonds with two power conjugated variables, it is a historical convention to write the effort variable on the left side of a vertical bond and the flow variable on its right side. For horizontal bonds, the effort is written above and the flow below the edge (Figure 2.4). None of the two power conjugated variables is discriminated in its role against the other. They are only characterised by the fact that their product equals the energy flow $\dot{E}(t)$ between two ports at a given time t .

2.3.2 Analogies

As both power variables play an equal role for each energy domain, it must be decided which of them shall be the effort, while the other becomes the flow. As a result, two different analogies between mechanical and electrical systems have emerged in the literature.

One option is to let forces in mechanical systems be the efforts. Consequently, translational velocities become flows. At the same time, voltages in electrical systems may be considered efforts and currents as flows. This force-voltage analogy is known as direct or classical analogy (Table 2.1).

The other equally valid option is to denote velocities and voltages as efforts and forces and currents as flows. This velocity-voltage assignment is called *dual* or *mobility analogy* (Table 2.2). It was proposed by Firestone [17] in 1933, while the force-voltage analogy has been known for more than 100 years. This may justify the notion of a classical analogy.

Table 2.1 Classical force-voltage analogy

Force	\triangleq	Effort	\triangleq	Voltage
Velocity	\triangleq	Flow	\triangleq	Current

Table 2.2 Mobility (Firestone) analogy

Velocity \triangleq Effort \triangleq Voltage
Force \triangleq Flow \triangleq Current

The mobility analogy is suggested by the observation that Kirchhoff's current law (The sum of all currents into an electrical node equals zero.) is equivalent to D'Alembert's principle in mechanics. The latter says that for a system of n mass points, the sum of inertial forces and imposed forces is equal to zero. The mobility analogy also appears reasonable from a measurement point of view. For the measurement of an electrical voltage and of mechanical velocities, two points in space are needed, while for measuring electrical currents and forces, one point in space is sufficient. The measurement point of view has led to the two notions *across*- and *through*-variable. The mobility analogy is commonly used when non-electrical subsystems are represented by a generalised network or an iconic diagram.

An essential feature of the mobility analogy is that it conserves the interconnection structure of a system when mechanical power variables are replaced by electrical ones and vice versa. That is, a parallel connection of two elements remains a parallel connection. The same holds for series connections. If the network representations of systems from different energy domains have the same structure, then these systems are called structurally analogue [37].

Another consequence of the mobility analogy is that a mechanical inertia corresponds to an electrical capacitor. This can be seen by comparing the equation for the momentum of a point mass with that for the charge of an electrical capacitor. Assuming that the initial values of momentum and charge vanish, both equations read:

$$\int_0^t F(\tau) d\tau = m \times v(t) \quad (2.1)$$

$$\int_0^t i(\tau) d\tau = C \times u(t) . \quad (2.2)$$

Similarly, one can show that a mechanical spring corresponds to an electrical inductance.

In the community of bond graph modellers, it has become common to use the direct analogy. With regard to basic elements, this analogy entails that a mechanical inertia corresponds to an electrical inductance. This can be seen by comparing Newton's third law to Faraday's law.

$$F = \frac{dp}{dt} = m \times \frac{dv}{dt} \quad (2.3)$$

$$u_L = L \times \frac{di}{dt} \quad (2.4)$$

Likewise, a mechanical spring corresponds to an electrical capacitor. If F_{sp} denotes the spring force, k , the stiffness of a spring with a linear characteristic, and u_C , the voltage drop across a capacitor, then the electrical capacitance parameter, C , corresponds to the compliance, $1/k$, of the spring.

$$F_{sp}(t) = k \times \int_0^t v(\tau) d\tau + F_{sp}(t=0) \quad (2.5)$$

$$u_C(t) = \frac{1}{C} \times \int_0^t i(\tau) d\tau + u_C(t=0) \quad (2.6)$$

It is obvious that a mechanical dashpot corresponds to an electrical resistor. The constitutive equation of an electrical resistor algebraically relates the voltage drop u across its terminal to the current i through the two pin element. The linear case is given by Ohm's law

$$u = R \times i, \quad (2.7)$$

where R denotes the resistance. The constitutive equation of an ideal dashpot is an algebraic relation between the damping force F_D and the velocity v

$$F_D = b \times v, \quad (2.8)$$

where b is the dashpot constant. As the classical analogy relates a force to a voltage and a velocity to a current, the two elements correspond to each other.

Contrary to the mobility analogy, the classical analogy does not preserve the model structure. That is, when mechanical power variables are exchanged by those of the electrical domain, a parallel connection of mechanical elements becomes a series connection of corresponding electrical elements. For instance, if a mechanical spring and a damper are connected in parallel (Figure 2.5), then the force acting in each of the two hinge points (on the wall and on the moving body) is obviously the sum of the spring force and the damper force.

If electrical currents are assigned to mechanical forces, the electrical analogue is a parallel connection of an inductance and a resistor. If however, voltages are assigned to forces, then according to Kirchhoff's voltage law, the electrical analogue is a series connection of a capacitor and a resistor (cf. Figure 2.6).

$$F_{sp} + F_d + m \frac{dv}{dt} = F(t) \quad (2.9)$$

Table 2.3 Correspondence of mechanical and electrical energy stores according to the classical analogy

mechanical inertia	\triangleq electrical inductance
mechanical (rotational) spring	\triangleq electrical capacitor
mechanical dashpot	\triangleq electrical resistor

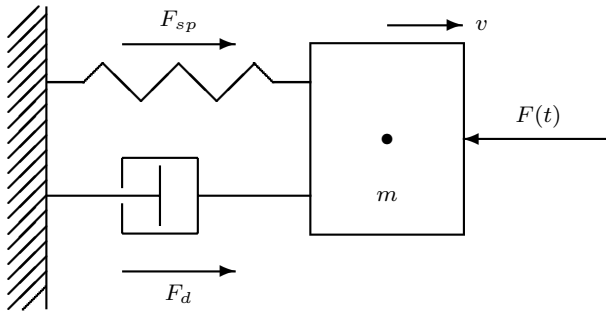


Fig. 2.5 Mechanical parallel connection

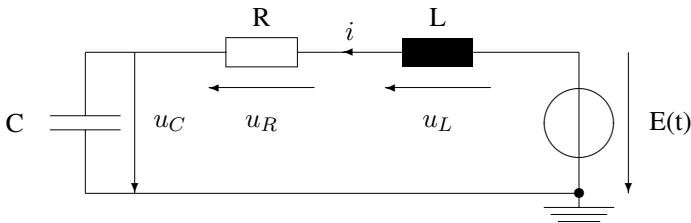


Fig. 2.6 Corresponding electrical series connection according to the classical analogy

$$u_C + u_R + L \frac{di}{dt} = E(t) \quad (2.10)$$

With regard to the derivation of a mathematical model from the bond graph, it does not matter which analogy has been chosen. As the classical analogy is common in bond graph related literature, this convention is also adopted throughout this monograph. Regarding the mobility analogy, see [16, 25, 37].

Table 2.4 shows which physical quantities are commonly chosen as effort and flow variables in the different energy domains. Regarding the electromagnetic energy domain, the horizontal line separating electrical quantities from magnetic ones does not go from left to right because the magnetic domain is not an energy domain independent from the electrical domain. The reader may notice that it is not the magnetic flux that can be chosen as a flow variable, but the magnetic flux rate. The product of magnetomotive force and magnetic flux is not the magnetic power transferred between two points in a magnetic material. The magnetic flux becomes conceivable by looking at paths built by the spatial orientations of tiny magnets inside the magnetic material. The flow is the variation of their orientations with time.

Finally, taking into account that for generalised networks or iconic diagram representations, the mobility analogy is commonly used, while in bond graph modelling, the classical analogy is most widely adopted, across- and through variables in networks can be related directly to efforts and flows in bond graphs, as shown in Table 2.5.

Table 2.4 Bond graph variables used in the various energy domains

Energy domain	Effort e	Flow f	Generalised momentum p	Generalised displacement q
Translational mechanics	Force F [N]	Velocity v [m/s]	Momentum p [Ns]	Displacement x [m]
Rotational mechanics	Angular moment M [Nm]	Angular velocity ω [rad/s]	Angular momentum p_ω [Nms]	Angle θ [rad]
Electro-magnetic domain	Voltage u [V]	Current i [A]	Linkage flux λ [Vs]	Charge q [As]
	Magnetomotive force V [A]	Magnetic flux rate $\dot{\Phi}$ [Wb/s]	–	Magnetic flux Φ [Wb]
Hydraulic domain	Total pressure p [N/m ²]	Volume flow Q [m ³ /s]	Pressure momentum p_p [N/m ² s]	Volume V_c [m ³]
Thermodynamic	Temperature T [K]	Entropy flow \dot{S} [J/K/s]	–	Entropy S [J/K]
Chemical domain	Chemical potential μ [J/mole]	Molar flow \dot{N} [mole/s]	–	Molar mass N [mole]

Table 2.5 Relation between across- and through variables in networks and efforts and flows in bond graphs

Energy domain	Effort	Flow
Mechanical domains	Through variable	Across variable
Non-mechanical domains	Across variable	Through variable

2.3.3 Energy Variables

Table 2.4 introduces two additional physical quantities used in bond graph modelling. They are called *generalised momentum* (p) and *generalised displacement* (q) and are obtained by integration of the power variables with respect to time.

$$p(t) = p(t_0) + \int_0^t e(\tau) d\tau \quad (2.11)$$

$$q(t) = q(t_0) + \int_0^t f(\tau) d\tau \quad (2.12)$$

It has been a convention since the beginning of bond graph modelling to use the notions generalised momentum and generalised displacement. However, these terms are not fully convincing since their roots are obviously in mechanical engineering. This may be considered inappropriate in other engineering disciplines. In electrical engineering, for instance, the charge, q , of a capacitor is the integral with respect to time of the current (flow). However, it is rather unusual to consider the electrical charge a generalised displacement, while in mechanical engineering displacements, in general, are not denoted by the letter q . The additional physical quantities introduced in Table 2.4 are called *energy variables* since they quantify the energy transferred in a time period and accumulated in an ideal energy store (cf. Section 2.5.3, Equations 2.38 and 2.46).

2.4 Orientation of Power Bonds

First, the energy exchange between power ports of submodels can be represented by non-directed edges in a bond graph. If, however, equations are given for all basic submodels, a sign convention is needed for the derivation of a set of equations from a bond graph in which variables are consistent with respect to their sign. This is achieved in bond graph modelling by adding a so-called *half arrow* to each bond indicating the positive *reference direction* of the energy flow. A half arrow is chosen to distinguish between energy flows and signal flows. The latter are commonly represented by edges with a full arrow. This orientation of a bond graph edge does not represent the actual direction of the energy flow which can vary with time. Rather it means a time-invariant reference direction which coincides with the energy direction at time t , if for the amount of power, P , holds: $P(t) = e(t) \times f(t) > 0$. Aside from energy sources, the reference direction of power is assumed positive if a bond connected to a power port of a submodel is oriented toward the power port. That is, if $P(t) = e(t) \times f(t) > 0$, energy flows towards the port. If, in addition, $f(t) > 0$, then the flow is directed towards the port at time t . This convention is motivated by the fact that energy stores and resistors consume energy whereas energy stores give up the accumulated energy at a later time, resistors irreversibly convert it into heat. For a bond connected to a power port of an energy source, the positive reference direction of the power is oriented away from the port. That is, if $P(t) = e(t) \times f(t) > 0$, energy flows out of the source corresponding to the fact that sources usually supply energy to a system. These considerations of a positive reference direction for energy flows allow for a refinement of Definition 2.1 and the introduction of directed bond graphs.

Definition 2.5 (*Directed bond graph*). A bond graph is called a directed bond graph if a half arrow has been added to each bond indicating the positive reference direction of the energy flow across the bond.

By convention, the half arrow is added on that side of the bond where the flow variable is annotated (cf. Figure 2.7). In the case the inclination of a bond is not a multiple of 90° , the question as to where to mark the half arrow doesn't have a unique answer. It depends on how the bond is virtually rotated into a vertical or horizontal position. In such cases, the convention is adopted that the flow variable is always on that side of a bond where the half arrow has been attached [6].

2.5 Basic Bond Graph Elements and Power Port Orientations

The fundamental physical processes, already mentioned in Section 2.2, suggest the introduction of the following classes of basic multiport elements used for an idealised description of physical processes.

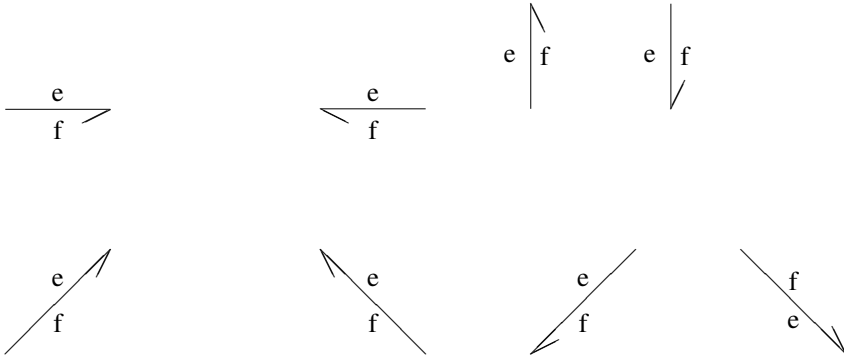


Fig. 2.7 Conventions for adding the half arrow to power bonds

- Energy sources and sinks (= negative energy sources),
- Energy stores,
- Dissipators converting energy irreversibly into heat,
- Power couplers and transducers, and
- Power nodes that instantaneously distribute power.

Energy sources deliver energy into a system, whereas sinks consume energy flowing out of the system. As depicted in Figure 2.3, sources and sinks do not belong to a system. They rather represent boundary conditions of a system embedded into the surroundings. In other words, sources and sinks model the impact of the surrounding on a system. In [10], Cellier points out:

A source as drawn in the circuit example above is actually a non-physical element. Power cannot be generated, only transported and converted. However, a “system” never denotes the whole of the universe. It denotes a piece of the universe. Sources are interfaces between the system and the universe around it.

Definition 2.6 (*Junction structure*). A bond graph in which bonds connect only nodes that instantaneously transfer or distribute power (without energy storage or conversion into heat), is called *Junction Structure* (JS).

The above classification indicates that basic elements are an idealised description of physical processes. That is, only *one* effect is represented, while other simultaneous effects are not taken into account. If several effects are involved in a physical process, different elements will have to be composed in a bond graph submodel representing the process. Besides the 0-junction and the 1-junction that are introduced in the next section, bond graph elements are similar to those used in generalised networks. This is not surprising since generalised networks also start from basic physical processes.

If the elements of a bond graph of a system model S are combined into submodels according to the classification discussed above, and if all power bonds between

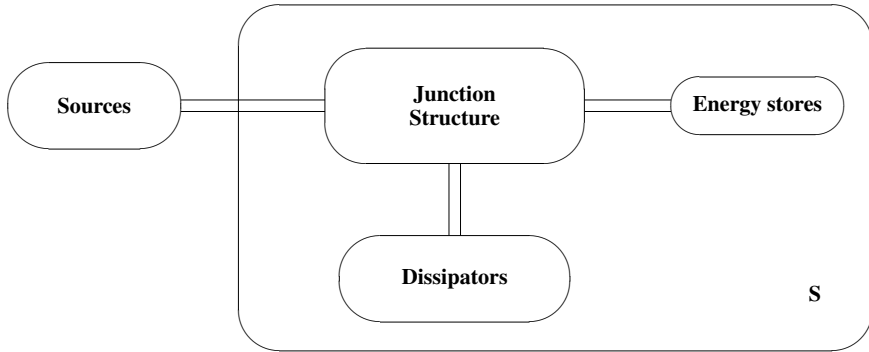


Fig. 2.8 General structure of bond graphs

the submodels introduced this way are represented by two parallel edges (undirected multibonds), then all bond graphs exhibit the general structure depicted in Figure 2.8. An essential feature of bond graphs is that the structure of the interconnections is a separate model part. The junction structure contains multiport elements that distribute or transfer power without storage or conversion into heat. As will be considered subsequently, the junction structure includes a type of an interconnection node that has no equivalent in networks. The reason is that networks are *terminal* oriented, whereas bond graphs are power port oriented.

In the following, the above classes of bond graph multiport elements will be discussed. Before starting with the bond graph elements that compose the general junction structure, the notion of *constitutive equations*, often used in the following section, is introduced. The constitutive equations describe the behaviour at the ports of a multiport element by relating power port variables. In the case of a 1-port element, its characteristic is a graphical representation of its constitutive equation.

2.5.1 Power Conserving Junctions

An ideal node that instantaneously distributes energy without storing it or converting it into heat must comply with the principle of *power conservation*. If P_{in} denotes the power entering the node and P_{out} the power leaving the node, then

$$P_{in} - P_{out} = 0 . \quad (2.13)$$

If such a node has $n \geq 3$ power ports and if one assumes without loss of generality that power enters at port 1 while leaving the node at all others ports simultaneously, then power conservation reads

$$e_1 f_1 - e_2 f_2 - \dots - e_n f_n = 0 . \quad (2.14)$$

Zero Junctions

As has been pointed out by F. Cellier in [11], among all possibilities to comply with the power balance, the simplest ones are to assume either equal efforts or equal flows. Assuming that all efforts are equal, the constitutive equations of an element are obtained that Paynter termed 0-junction (*zero junction*). In bond graphs, its type is denoted by the symbol 0.

Definition 2.7 (0-junction). A 0-junction is a multiport element defined by the following equations

$$e_1 = e_2 = \dots = e_n \quad (2.15a)$$

$$f_1 = f_2 + \dots + f_n. \quad (2.15b)$$

According to Equation 2.15a the element is also called *common effort junction*. Equation 2.15b has given rise to the notion of a *flow junction*.

Equations 2.15a and 2.15b are well known in electrical engineering. Let e_1, e_2, \dots, e_n denote the voltage drops across the n ports of a subnetwork and f_1, f_2, \dots, f_n the port currents, then Equation 2.15b is just Kirchhoff's current law. Thus, parallel connections in electrical networks can be represented by a 0-junction in a corresponding bond graph (cf. Figure 2.9).

If in Figure 2.9 node 2 is the common ground node, then the 0-junction just represents the voltage of node 1 with respect to ground. In hydraulics, a pipe tee junction can be represented by a 0-junction if dynamic pressures can be neglected with regard to hydrostatic pressures. The reader may notice that the number of ports of a 0-junction is not fixed.

According to what has been said regarding to the orientation of bonds and according to Equation 2.14, bond 1 must point toward the 0-junction while all other edges are directed away from the node (Figure 2.10).

Equation 2.15b indicates that power reference directions correspond to the sign of the flow variables. That is, if a power bond points towards a 0-junction, it is

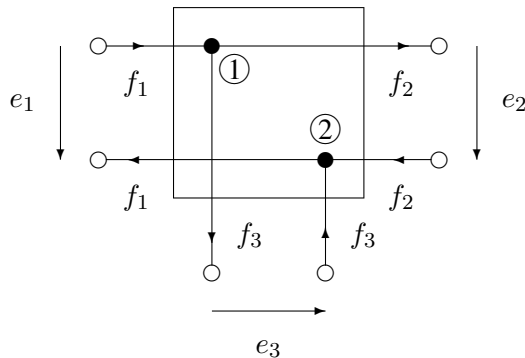


Fig. 2.9 Example of a 0-junction in an electrical network

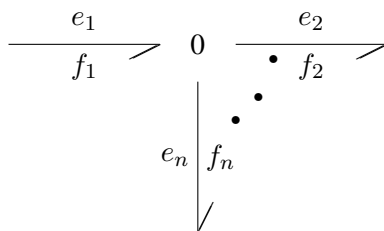


Fig. 2.10 Orientation of the bonds connected to a 0-junction according to Equation 2.14

assumed that its flow variable has positive sign, otherwise its sign is negative. If in a bond graph of a *mechanical* system one of the power ports of a 0-junction is connected to a power port of an energy store (spring) or to the port of a dissipator (dash-pot) by means of bond 2 then one of the remaining bonds connected to the 0-junction must point towards the junction while another bond must be directed away from the junction (cf. Figure 2.41). This rule reflects that physical connections between two bodies become effective if there is a relative velocity between them. It is the *difference* of velocities, not a sum of velocities, that gives rise to an action. On the other hand, a real physical link between two bodies, e.g., a spring, or a dashpot impose one and the same force, or a moment on both bodies. Consequently, in bond graphs of mechanical systems, 0-junctions are used for describing links.

It should be stressed that Equation 2.15b is not an assignment statement. It may be solved for any of the flow variables as needed. Which of the flow variables is a dependent one depends on the element ports, or submodels ports the 0-junction is connected to.

One Junctions

Let us now assume that all flow variables are equal in Equation 2.14. Thus, the sum of all efforts must vanish.

$$f_1 = f_2 = \dots = f_n \quad (2.16a)$$

$$e_1 - e_2 - \dots - e_n = 0 \quad (2.16b)$$

Equations 2.16a and 2.16b characterise an element that apparently is the dual to the 0-junction. For this reason, Paynter called it the 1-junction (one junction). In bond graphs, it is denoted by the symbol 1. According to its constitutive equations, it is also known as *common flow junction*, or as *effort junction*.

Definition 2.8 (1-junction). A 1-junction is a multiport element, for which power port variables comply with Equations 2.16a and 2.16b.

Equations 2.16a and 2.16b are also well known in electrical engineering. If the flow variables denote the current through elements with two terminals connected in series and if the effort variables are the voltage drops across the elements, then

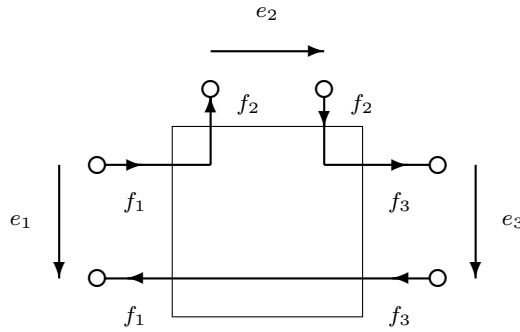


Fig. 2.11 Example of a 1-junction in an electrical network

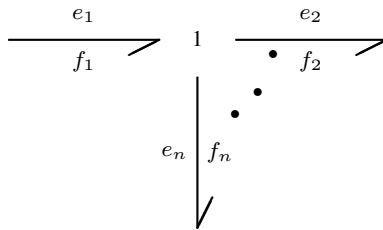


Fig. 2.12 Multiport 1-junction with power bond orientations according to Equation 2.14

Equation 2.16b is just Kirchhoff’s voltage law (Figure 2.11). Figure 2.12 depicts a multiport 1-junction with power bond orientations according to Equation 2.14.

In contrast to networks, bond graphs use an additional node to allow for series interconnection of elements in non-mechanical systems. There is a need for such a node since in bond graphs power ports are to be connected, whereas in networks, terminals are connected.

If in bond graphs of non-mechanical systems one bond of a 1-junction is oriented towards to the port of an adjacent energy store or a resistor, then one of its other bonds must point towards the 1-junction while another edge must be directed away from the 1-junction. This rule takes into account that it is a voltage drop across an electrical element with two terminals that gives rise to the current through the element and not the sum of electrical voltages with respect to ground. Likewise, it is a pressure difference across a hydraulic line and not the sum of pressures that is related to a fluid flow through the line. As a consequence of this rule, half arrows pointing to and away from the 1-junction also indicate the flow, e.q. electrical current, volume flow, or mass flow through the element from the higher to the lower potential. In bond graphs of mechanical systems, 1-junctions indicate the (angular) velocity at a point and at the same time, the sum of all forces (moments) acting on that point (D’Alembert’s principle).

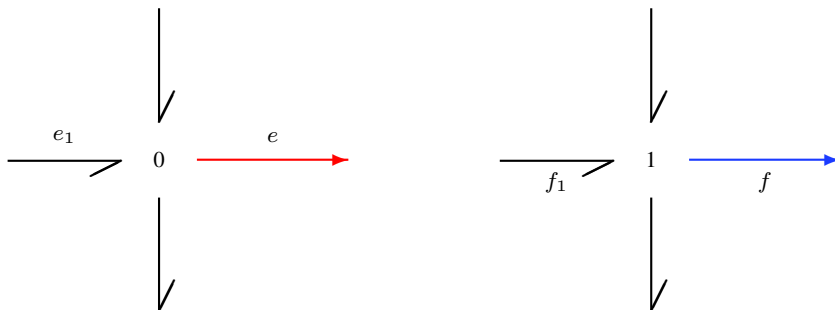


Fig. 2.13 Signal arrows taking up the *information* of the effort of a 0-junction or of the flow of 1-junction

Since both types of junction indicate a power variable that is common for all adjacent bonds, they may have one signal port in addition to their power ports. From this signal port, the *information* of the common power variable can be taken up and fed into a block diagram for signal processing. Ideal measuring of the common power variable does not affect the power balance of the junction and can be represented by attaching a signal arrow to the junction pointing away from it (Figure 2.13). Clearly, the information taken up at the signal port must be provided at one and only one of the power ports (cf. Section 3.2 on computational causalities, Figure 3.5).

The introduction of 0- and 1-junctions gives rise to further definitions.

Definition 2.9 (*Internal bond*). A bond is called an internal bond if it connects a 0- or 1-junction to another 0- or 1-junction.

Definition 2.10 (*Simple junction structure*). A bond graph is called a simple junction structure, or Kirchhoff junction structure if each node is either a 0- or a 1-junction.

Definition 2.11 (*External bond*). A bond is called an external bond if it connects a 0- or 1-junction to a power port of an element that does not belong to the simple junction structure.

2.5.2 Ideal Power Couplers and Power Transducers

The assumption that a power coupler or power transducer neither stores energy nor converts it irreversibly into heat means that the constitutive equations of such a device must comply with the principle of power conservation. First, two-port elements are considered. The general case of n -port elements will be dealt with in Chapter 8

in the context of multibond graphs. In the case of a two-port element, power conservation means

$$e_1 f_1 = e_2 f_2 . \quad (2.17)$$

Transformers

Assuming a constraint between the two efforts

$$e_1(t) = m \times e_2(t) , \quad (2.18)$$

where m is a non-negative real parameter and by substituting Equation 2.18 into the power balance yields

$$f_2(t) = m \times f_1(t) \quad (2.19)$$

for the flow variables. Another possible constraint between the two efforts is

$$k \times e_1 = e_2 . \quad (2.20)$$

That is, both parameters, $m, k \in \mathbb{R}$, $m, k \geq 0$, are constrained by the relationship $m = 1/k$.

If the two efforts are considered to be the voltages across the ports of an electrical two-port element and the flows are the currents flowing into and out of the ports, then Equations 2.18 and 2.19 describe an ideal electrical transformer. These observations have led to a bond graph element called transformer. In bond graphs, it is denoted by the symbol TF.

Definition 2.12 (*Two-port transformer*). A two-port transformer is an element with constitutive Equations 2.18 and 2.19. Its modulus m may be a constant of non-negative real value, a function of some other power variable or conserved physical quantity, or a function of time.

In bond graphs, the symbol TF may be annotated by the modulus m separated from the symbol TF by a colon. If the modulus is not constant, the symbol TF is prefixed by the letter M (Figure 2.14). In that case, the transformer is called a *modulated* transformer. Moreover, modulation may be emphasised by connecting the output of a signal processing block and the transformer node via a signal arrow. In that case, the two-port transformer becomes a 3-port element with two power ports and an additional signal port. Thus, the graphical representation of a model becomes a combination of a bond graph and a block diagram. In a slider crank mechanism for instance, the angular velocity ω is transformed into the translational velocity v_p of the piston. The transformer modulus is a (complicated) function of the angle ϕ that is derived from the geometric constraints (cf. Figure 4.6).

Since an ideal two-port transformer does not store energy, it is appropriate to have one bond pointing towards the element while the other one is directed away from the

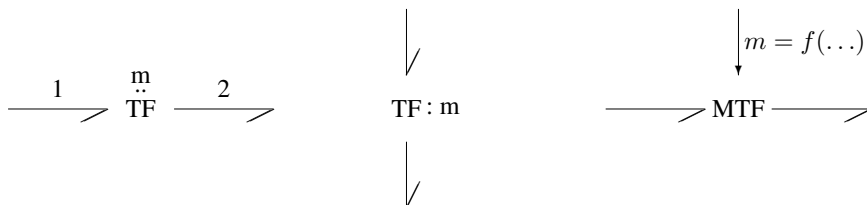


Fig. 2.14 Representation of two-port transformers

elements. This convention supports the view that energy flows *through* the element¹. The transformer modulus is defined unambiguously because the power variables e_1 and f_1 are always associated with the bond pointing *towards* the transformer [9].

Example: Mechanical Gear Pair

In mechanics, an ideal transformer can be used to capture the main function of a gear pair (Figure 2.15). Both gears have the same tangential velocity v_t .

$$r_1 \times \omega_1 = v_t = r_2 \times \omega_2 \tag{2.21}$$

Substitution of this kinematic relationship into the power balance yields for the moments

$$r_2 \times M_1 = r_1 \times M_2 . \tag{2.22}$$

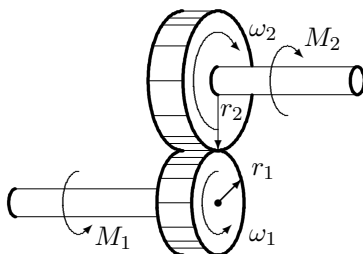


Fig. 2.15 Gear pair

¹ If an inward orientation of the bonds is adopted, then a minus sign would result in Equation 2.19 for the flows due to the rule that power is assumed positive if the bond is directed toward the element. $e_1 f_1 + e_2 f_2 = e_2 (m f_1 + f_2) = 0$

Example: Hydraulic Cylinder

In hydraulics or pneumatics, the core function of a cylinder can be represented by an ideal transformer. If the piston has cross section areas A_1 and A_2 and a translational velocity v , and if Q_1 and Q_2 denote the inlet and the outlet volume flow, then the velocity of an *incompressible* fluid reads

$$\frac{Q_1}{A_1} = v = \frac{Q_2}{A_2} . \quad (2.23)$$

If it is assumed that hydraulic power is approximately the product of hydrostatic pressure times volume flow, then the power balance for the hydrostatic pressures yields

$$p_1 A_1 = p_2 A_2 . \quad (2.24)$$

Another device that may be approximately described by an ideal transformer is a hydraulic flow pump. In this case, the transformer represents the instantaneous transformation of mechanical power into hydraulic power. Losses and storage effects in real pumps are accounted for by further bond graph elements.

Gyrators

If Faraday's law is applied to a conductor of length l moving at velocity v in a magnetic field of magnetic flux density B , then there is a relation between the electrical effort, the voltage u across the conductor, and the mechanical flow, the velocity v

$$u = (B \times l) \times v . \quad (2.25)$$

Substitution of this relation into the power balance yields a relationship for the power conjugated variables, the Lorentz force, F , acting on the conductor and the current, i , through the conductor

$$F = (B \times l) \times i . \quad (2.26)$$

Both relations assume that the vectors \mathbf{B} , \mathbf{i} , \mathbf{v} are perpendicular to each other. In contrast to the constitutive equations of a two-port transformer, these equations relate efforts to flows. Moreover, they describe a power conversion between electrical and mechanical energy.

Definition 2.13 (*Two-port gyrator*). An ideal two-port gyrator is a power conservative element defined by the two constitutive relations

$$e_1 = r \times f_2 \quad (2.27a)$$

$$e_2 = r \times f_1 . \quad (2.27b)$$

The parameter, $r \in \mathbb{R}$, $r > 0$, is called gyrator ratio.

In bond graphs, gyrators are denoted by the symbol GY. The gyrator ratio r may be attached to the symbol separated from the symbol by a colon. If the ratio is not

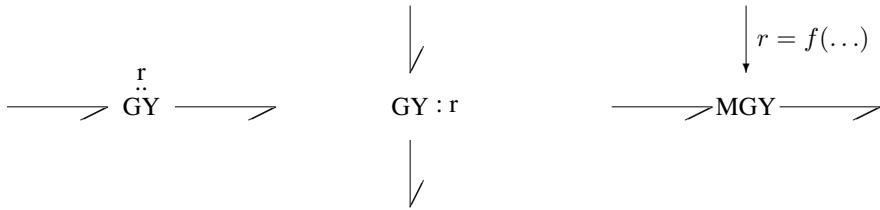


Fig. 2.16 Gyrator and modulated gyrator

a constant, the symbol MGY is used and the gyrator is called a *modulated* gyrator (Figure 2.16).

As to the orientation of the two adjacent bonds, a through direction is adopted like for transformers. The gyrator ratio is unambiguously defined by relating the effort of the bond pointing towards the two-port gyrator to the flow of the other bond pointing away from the node. Consequently, unlike the transformer modulus, the gyrator ratio always has a physical dimension. Since the constitutive equations of a gyrator are symmetric, it does not matter which of the two bonds has an inward orientation. A gyrator is mostly used to approximately describe transducers that transform energy from one form into another. Examples are electrical DC motors, electrodynamic loudspeakers, mass accelerometers, or centrifugal pumps.

Example: Energy Conversion in an Electrical Coil

A phenomenon that can be represented by a gyrator with *constant* ratio is the conversion of electrical energy into magnetic energy that happens in an electrical coil wound on a magnetic core. The voltage u across the terminals of a coil with n turns is related to the rate of the magnetic flux Φ according to Faraday's Law

$$u = n \times \frac{d\Phi}{dt}. \quad (2.28)$$

Its substitution into the power balance

$$u \times i = V \times \frac{d\Phi}{dt} \quad (2.29)$$

yields for the magnetomotive force V setting up the magnetic field

$$V = n \times i. \quad (2.30)$$

Both equations can be represented by a gyrator (Figure 2.17).

Storage effects, e.g., the self-inductance of the coil, storage of magnetic energy in the ferromagnetic material, losses, e.g., due to eddy currents in the magnetic core are to be accounted for separately.

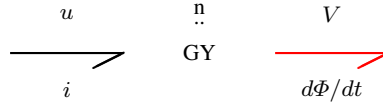


Fig. 2.17 Conversion between electrical and magnetic power in a coil

Example: DC Motor

The power conservative conversion of electrical energy into mechanical energy in a DC motor can be described by a *modulated* gyrator (MGY). Its ratio is a non-linear function of the field current due to saturation. Further effects, e.g., the self-inductance of the field and armature coils, the mechanical inertia of the rotor, electrical resistances and mechanical bearing losses will have to be accounted for by additional bond graph elements.

Although only bond graph elements belonging to the junction structure have been introduced so far, Figure 2.19 depicts an entire bond graph model of a shunt motor shown in Figure 2.18. The core bond graph element in this model is the modulated gyrator (MGY) representing ideal instantaneous lossless conversion from electrical into mechanical energy. One of its two constitutive equations relates the current, i_a , through the armature of the motor to the torque, M , driving the mechanical load. The second equation takes into account that the angular velocity, ω , of the load causes an induced voltage u_a .

$$M = \Psi \times i_a \quad (2.31a)$$

$$u_a = \Psi \times \omega \quad (2.31b)$$

The 0-junction represents the voltage, E , delivered by the voltage source. Simultaneously, it indicates that the current, i , through the voltage source is the sum of the current, i_a , through the armature and the current, i_f , through the field winding.

$$i = i_a + i_f \quad (2.32)$$

The left-hand side 1-junction represents the current i_a and the sum of all voltage drops along mesh II.

$$-E + u_{R_a} + u_{L_a} + u_a = 0 \quad (2.33)$$

The upper 1-junction indicates the current, i_f , through the field winding and that the voltage, E , of the voltage source is the voltage across the resistor of the field winding according to the schematic of the motor.

Finally, the right-hand side 1-junction represents the angular velocity, ω , and, simultaneously, the sum of all torques acting on the flywheel with the moment of inertia J_m .

$$M + M_{\text{load}} - M_R - J_m \dot{\omega} = 0 \quad (2.34)$$

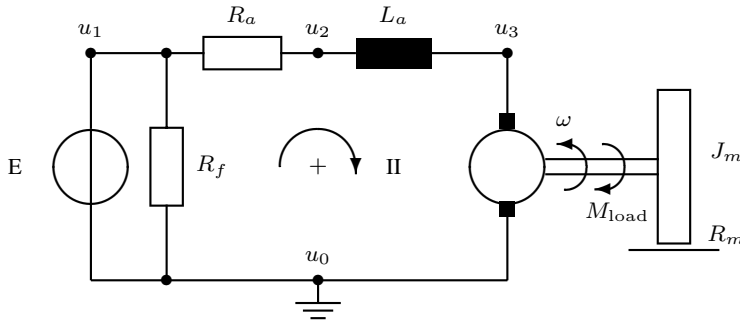


Fig. 2.18 Shunt motor

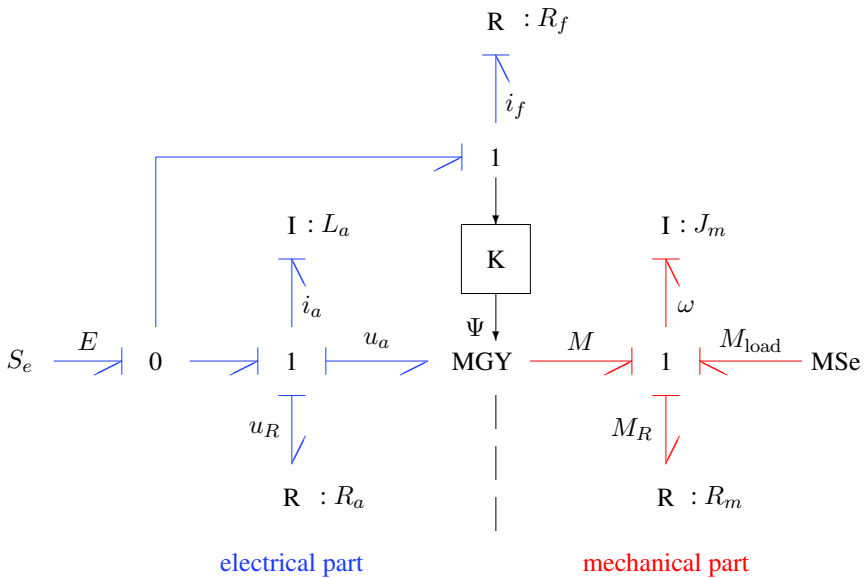


Fig. 2.19 Bond graph model of a shunt motor

The further bond graph elements, the energy storing elements denoted by the symbol I, the resistors (R elements) and the effort sources (Se) can be related to the elements in the schematic of the motor. They will be introduced in subsequent Sections 2.5.3, 2.5.4 and 2.5.6. The perpendicular strokes at the bonds in Figure 2.19 will be introduced later in Section 3.2. The bond graph model of the shunt motor is used in Chapter 11 as a reference in order to illustrate various aspects.

Now, having introduced the bond graph nodes of type 0,1, TF, GY, a formal definition of the notion *General Junction Structure* can be given.

Definition 2.14 (*General junction structure*). A bond graph with nodes of type 0,1, (M)TF, (M)GY is called General Junction Structure (GJS) [31].

A special case is the so-called *weighted* junction structure.

Definition 2.15 (*Weighted junction structure*). A bond graph with nodes of type 0,1, (M)TF is called a Weighted Junction Structure (WJS) [31].

Remark 2.4. Since the general junction structure is a bond graph of which all nodes are power conservative, it is a power conservative multiport.

Definition 2.16 (*Environmental elements*). All elements that do not belong to the general junction structure are called *environmental elements* [31, 36].

2.5.3 Energy Storage Elements

Like in generalised networks [25], in bond graphs, also two types of energy stores are used to describe energy storage in all energy domains except in the magnetic domain, the thermodynamic and in the chemical domain. In bond graphs, these two types of stores are designated by the symbols C and I. Although energy storage elements are multiport elements in the general case, in the following, the discussion will first consider 1-port energy storage elements. This limitation is justified since for many real problems, bond graph models can be developed that only use 1-port energy stores. After the discussion of further aspects of bond graph modelling in the subsequent chapters, we will come back to multiport energy stores. These are also called energy storage fields in the context of modelling rigid multibody systems (see Chapter 8).

Definition 2.17 (*1-port C energy store*). An ideal 1-port element of type C is defined by a one-to-one function $\Phi_C : \mathbb{R} \rightarrow \mathbb{R}$ relating the effort variable, e , of the power port to the generalised displacement q

$$q(t) = \Phi_C(e(t)), \quad (2.35)$$

where $t \in \mathbb{R}$ and $t \geq 0$. The function Φ_C must have a unique single valued inverse Φ_C^{-1}

(cf. [22]). According to its definition (see Equation 2.12), the rate of the generalised displacement equals the flow variable of the power port

$$\dot{q} = f. \quad (2.36)$$

Remark 2.5. This definition of a C element is independent of the form in which its constitutive equation is written. Due to the existence of a single valued inverse Φ_C^{-1} , the constitutive equation could also be written in the form $e = \Phi_C^{-1}(q)$.

The existence of a single valued inverse Φ_C^{-1} is required to ensure that the constitutive Equation 2.35 can be solved for the effort variable, if needed. It is not acceptable that the solution of a set of coupled equations depends on how elements are interconnected. An element should be independent of the surroundings in which it is embedded. For instance, if two point masses are connected via two springs in series, the position of the interconnection point of both springs is used in the constitutive equations of both springs. That is, the springs cannot act independently of each other. In order to eliminate the position of the point connecting both springs and to replace the two springs by a single one, imposing a force F on both point masses requires the existence of a single valued inverse function for one of the two constitutive relations.

From the non-uniqueness of the inverse relation of an element, Beaman and Rosenberg conclude that multiple physical effects have been accounted for improperly by the same element. They call such elements *composite* [2]. The requirement of constitutive relations having a unique inverse does not mean a limitation to modelling. Since non-unique inverse relations can cause problems, both authors suggest to avoid composite elements and to try to capture effects in separate elements that have a single valued inverse characteristic.

The relationship defining an ideal C energy store is well known from modelling mechanical springs, torsion springs, electrical capacitors, or volumes that store a compressible fluid. For instance, if q designates the charge and u the voltage drop across the terminals of an electrical capacitor, its constitutive relations reads

$$q = C \times u . \quad (2.37)$$

The capacitance C may depend on the voltage drop u . Following the example of the electrical capacitor, the bond graph element is called a (1-port) capacitor.

If an initially empty capacitor is assumed and if the direction towards the element is chosen as a positive reference direction of the energy flow, then the energy stored at time instant t is

$$\begin{aligned} E(t) &= \int_0^t e(\tau) \times f(\tau) d\tau \\ &= \int_0^q \Phi_C^{-1}(\tilde{q}) d\tilde{q} . \end{aligned} \quad (2.38)$$

Equation 2.38 indicates that the integral of the flow with respect to time, the conserved quantity, q , can be considered to be a so-called *energy variable*. It is a measure for the stored energy.

In the case of a linear characteristic

$$q = C \times e , \quad (2.39)$$

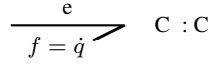


Fig. 2.20 1-port C element

the capacitance, C , may be added to the symbol C indicating the element type. The parameter C is separated from the symbol of the element by a colon (Figure 2.20).

The half arrow of the adjacent bond should point towards the element meaning that the energy store is accumulating energy. If the half arrow is added such that it points away from the element, this might be compensated by a negative parameter. This, however, is a source for potential sign errors if variables and parameters are not used consistently. Moreover, real devices are usually characterised by positive parameters. A mechanical spring, for instance, has a positive stiffness parameter or an electrical capacitor has a positive capacitance. For an electrical capacitor the parameter of the bond graph element equals the capacitance. In the case of a mechanical spring, it equals the inverse of the stiffness parameter. For a hydraulic fluid store, the parameter of the bond graph C element is the ratio of the bulk modulus of the fluid (and the wall of the container) and the container volume.

In the case of a linear characteristic, Equation 2.38 takes the well known form

$$E(t) = \frac{q(t)^2}{2C} . \quad (2.40)$$

If x designates the elongation of a linear spring of stiffness k , then its stored potential energy is expressed as

$$E_{pot}(t) = \frac{1}{2} k x^2(t) . \quad (2.41)$$

For nonlinear characteristics, we will have to distinguish between the stored energy and the so-called co-energy E^*

$$E^*(e) = \int_0^e \Phi_C(\tilde{e}) d\tilde{e} . \quad (2.42)$$

There is a relation between both energies

$$E^* = e \times q - E . \quad (2.43)$$

The co-energy can be represented by the area above the characteristic $e = \Phi_C^{-1}(q)$.

Definition 2.18 (*1-port I energy store*). An ideal 1-port element of type I is defined by a one-to-one function $\Phi_I : \mathbb{R} \rightarrow \mathbb{R}$ relating the flow variable, f , of the power port to the generalised momentum p

$$p(t) = \Phi_I(f(t)) . \quad (2.44)$$

The function Φ_I must have a unique single valued inverse Φ_I^{-1}

(cf. [22]). According to its definition (see Equation 2.11), the rate of the generalised momentum equals the effort variable of the power port

$$\dot{p} = e . \quad (2.45)$$

An element defined by Equations 2.44 and 2.45 is called an *inertia*, or an I element in bond graph methodology. In non-relativistic mechanics, Equation 2.44 is a linear relation. Its parameter designated by the letter I may be the mass or the moment of inertia of a rigid body, whereas in electrical engineering, Equation 2.44 relates the flux linkage λ and the current i . In the latter case, Equation 2.11 is a nonlinear relation in general. That is, in mechanics, a bond graph I element represents the storage of kinetic energy of a rigid body. In electrical engineering, the I element accounts for the storage of magnetic energy in a coil.

If we assume that the initial momentum vanishes, that is, the I energy store is empty at $t = 0$, then the amount of stored energy is

$$E = \int_0^p \Phi_I^{-1}(\tilde{p})d\tilde{p} . \quad (2.46)$$

In hydraulics, the use of a generalised momentum is uncommon. However, Newton's third law can be applied to an incompressible fluid in a volume of length l . Assuming a 1-dimensional flow and by replacing mechanical quantities by their corresponding hydraulic ones in the two equations

$$p = m \times v \quad (2.47)$$

$$\dot{p} = F \quad (2.48)$$

yields

$$\Delta p = I \times \dot{Q} \quad (2.49)$$

and

$$I = \int_0^l \frac{\rho}{A(x)} dx . \quad (2.50)$$

In these equations, Δp designates the pressure drop across the length l and $A(x)$, the cross section area of the volume at the position x ($0 \leq x \leq l$). That is, the parameter I represents the inertia of a 1-dimensional incompressible fluid in a volume of length l . As can be seen, the smaller the cross section area of a pipe, the higher the inertia of the fluid.

As to the orientation of the bond connected to a port of an I element and regarding the annotation of the parameter, the same rules hold as for a C store (Figure 2.21).

Table 2.6 lists some examples of devices and physical effects in various energy domains that can be approximately described by a C energy store or by an I energy store. The missing entries indicate that there are no inertia elements in the mag-

$$\frac{e = \dot{p}}{f} \quad I : I$$

Fig. 2.21 1-port I element

Table 2.6 Capacitor and inertia in various energy domains

Energy domain	C store	I store
Translational mechanics	Spring	Rigid body
Rotational mechanics	Torsion spring	Flywheel
Electro-magnetic domain	Capacitor	Coil
	Ferromagnetic material	—
Hydraulics	Fluid compressibility	Fluid inertia
Thermodynamics	Lump of material	—

netic and in the thermal domain, as it is well known. This is also the reason why in Table 2.4, generalised momenta for these energy domains are not given.²

² Occasionally, it has been argued in the literature that a thermal inertia would violate the second law of thermodynamics. In order to prove this statement, a *hypothetical* thermal inertia with the constitutive equation

$$\frac{d\dot{S}(t)}{dt} = \frac{1}{I} \times (\Delta T)(t)$$

is assumed. Apparently, such a relation implies a constant entropy flow, $\dot{S} = \text{const.} = \dot{S}(t = t_0)$ in the case of a vanishing temperature difference $\Delta T = 0$. It has been argued that this would contradict the second law of thermodynamics. Consequently, thermal inertia could not exist. However, since this proof, constructing a contradiction does not provide a constant > 0 and the latter could be zero as well. In this case, there would be no contradiction and nothing has been proven.

The Thermal Capacitor

In Table 2.6, a lump of material is considered a thermal capacitor. The constitutive equation of such an energy storage element is

$$S = \Phi_S(T) \quad (2.51)$$

relating entropy³, S , to temperature, T , can be derived in the following manner. First, we assume that the expansion of the volume can be neglected when the lump of material under consideration is heated. That is, the mechanical work it performs can be neglected. Otherwise, an additional port would be needed at which mechanical energy may enter or leave. From physics, it is known that the thermal capacitance is defined as the stored amount of heat, Q , divided by the temperature increase ΔT . Differentiation of the equation

$$Q = C \times \Delta T \quad (2.52)$$

with respect to time and observing that $\dot{Q} = T \dot{S}$ results in

$$T \dot{S} = C \dot{T} . \quad (2.53)$$

Integration with respect to time and defining $T_0 := T(t = t_0)$, $S_0 := S(t = t_0)$ gives the required constitutive relation

$$S - S_0 = C \ln \frac{T}{T_0} , \quad (2.54)$$

relating the effort T to the displacement S .

Equivalent Representation of the I Store

The introduction of a second type of an energy store indicates that this type is simply obtained by interchanging the role of effort and flow. For this reason, the I energy store is called *dual* to the C energy store. An I element can be replaced by combining a C energy store with a gyrator of unity ratio. Such a gyrator has been termed *symplectic gyrator* [5]. This equivalence can be seen by taking the following steps. First, Equation 2.27b of the gyrator is differentiated with respect to time. Then, \dot{e}_2 is replaced by f_2 using the constitutive relation of a C store. Finally, f_2 is replaced by e_1 using Equation 2.27b of the gyrator.

$$\dot{f}_1 = \frac{1}{r} \dot{e}_2 = \frac{1}{r} \frac{1}{C} f_2 = \frac{1}{r} \frac{1}{C} \frac{1}{r} e_1$$

³ Around 1855, Clausius introduced entropy by the equation

$$dS = \frac{dQ}{T} ,$$

where dQ denotes the change of heat.

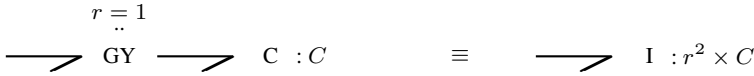


Fig. 2.22 Equivalence of an I energy store to a gyrator–C energy store combination

$$= \frac{1}{r^2 C} e_1 = \frac{1}{I} e_1, \tag{2.55}$$

where $I := r^2 C$ (Figure 2.22).

The equivalent representation of an I energy store by means of a symplectic gyrator and a C energy store could be used to give up the introduction of a second store in order to overcome the discrepancy that the second type of an energy store does not exist in all energy domains. For this reason, P. Breedveld introduced a general uniform bond graph concept that uses only the C type store. First, he called such bond graphs *thermodynamic* bond graphs [3]. Later, he changed the term into *Generalised Bond Graphs* [4]. Most bond graph modellers, however, prefer to keep two types of stores for convenience, although this is not fully satisfying with regard to a general uniform theory. In this book, we will follow the long lasting tradition of using two types of energy stores.

Are There Controlled Energy Stores?

In contrast to ideal power conservative couplers or transducers, energy storage elements cannot be controlled by a signal (Figure 2.23) because this would violate the principle of energy conservation.

Consider, for instance, a capacitor with movable plates. For such devices, the voltage e across the two terminals not only depends on the charge q of the plates, but also on their distance x . If $C(x)$ designates the capacitance, then

$$e(t) = C^{-1}(x(t)) \times q(t). \tag{2.56}$$

Consequently, the stored energy is a function of the charge q and the distance x of the plates

$$E(q, x) = \frac{q^2}{2C(x)}. \tag{2.57}$$

Thus, a change of the stored energy is



Fig. 2.23 Non-existing signal controlled energy C store

$$P(t) = \dot{E}(t) = \frac{\partial E}{\partial q} \dot{q}(t) + \frac{\partial E}{\partial x} \dot{x}(t). \quad (2.58)$$

If a *signal* controlled energy store of variable capacitance is assumed, then the second term in that sum must vanish because the energy flow associated with a controlling signal is neglected. For this reason, there is a distinction between power ports and signal ports in bond graphs. This, however, means that the stored energy and hence the voltage e cannot be a function of the modulating signal. Thus, modulated stores cannot exist [2].

In other words, if the stored energy, E , is a function of q and x , then Equation 2.58 must hold. However, if $P(t) = e(t) \times \dot{q}(t)$, then the second term in Equation 2.58 must be zero. As $\dot{x} \neq 0$, the factor $\partial E / \partial x$ must vanish. This means that

$$\frac{\partial}{\partial q} \left(\frac{\partial E}{\partial x} \right) = 0. \quad (2.59)$$

On the other hand, as E is assumed to be a function of q and x ,

$$\frac{\partial}{\partial x} \left(\frac{\partial E}{\partial q} \right) \neq 0. \quad (2.60)$$

That is, the value of the line integral

$$\int_{\mathcal{C}} \left(\frac{\partial E}{\partial q} \right) dq + \left(\frac{\partial E}{\partial x} \right) dx \quad (2.61)$$

along the path \mathcal{C} is not independent of the path \mathcal{C} , which means that the *modulated* 2-port C element is not energy conservative.

This does not mean that the stored energy cannot depend on a variable distance of the plates. However, it must be taken into account that a change of their distance is combined with an energy flow. Accordingly, a modulated C energy store is to be replaced by a multiport store. Such elements will be considered in Chapter 8.

2.5.4 Dissipators

Since there is no loss of energy, the notion of *free* energy is introduced.

Definition 2.19 (*Free energy*). If E denotes the total energy in a system and E_{th} the thermal energy, then the free energy, E_f , is the difference

$$E_f := E - E_{th} \quad (2.62)$$

(see, for instance, [9]).

The loss of *free* energy due to friction in mechanical and hydraulic systems, due to heat production in electrical circuits, or due to thermal conduction is modelled in bond graphs (like in generalised networks) by means of resistors. In bond graphs,

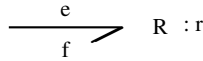


Fig. 2.24 1-port R element

the type of these elements, which may be multiport elements, is designated by the symbol ‘R’. In the following, 1-port resistors will be considered.

Definition 2.20 (*1-port resistor*). A 1-port resistor is defined by one of the two constitutive equations relating the power port variables e and f

$$e(t) = \Phi_R(f(t)) \quad (2.63)$$

or

$$f(t) = \Phi_G(e(t)) , \quad (2.64)$$

where $t \in \mathbb{R}$ and $t \geq 0$. Both functions, Φ_R and Φ_G , must be one-to one and have a single valued inverse. Their characteristics must fall into the first and third quadrants of the e-f plane.

In case of a 1-port resistor with a linear characteristic

$$e = r \times f , \quad (2.65)$$

the parameter r may be attached to the symbol R denoting the type of the element (Figure 2.24). In the general case of a (non)linear (multiport) resistor or (multiport) energy store, the symbol X (C or R) may be annotated by a string (Figure 2.25). In terms of object-oriented modelling, $X : s$ denotes an instantiation of the class X. The string s is the name of the instantiation that allows one to distinguish it from others. In case of a single element parameter, the string may be the name of that parameter. For more than one parameter, s may be the name of the set of parameters used in the constitutive equations. This is similar to the way in which elements in a circuit diagram are distinguished. There is a standardised graphical symbol for an element of type resistor. Different resistors have different names. The name R1 associated with one of them may be the name of its parameter having a value of 10Ω , viz., $R1 = 10 \Omega$. Resistors are also called dissipators, or R elements.

The *free* energy entering an R element is lost in the system. It is dissipated by the element. However, according to the first law of thermodynamics, energy cannot be conserved. In R elements, energy is converted irreversibly into heat. It is appropriate to define the direction towards the element as the positive reference direction of the

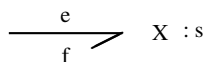


Fig. 2.25 Nonlinear 1-port element of type X (C or R) with s being the name of its parameter set

energy flow. Since true resistors irreversibly convert energy into heat, the reference direction is also the actual direction of the energy flow. Thus, in addition to the constitutive relation of an R element, there is the constraint $e(t) \times f(t) > 0 \quad \forall t > 0$.

Remark 2.6. The requirement of single valued inverse functions Φ_R^{-1} and Φ_G^{-1} shall ensure that the constitutive relation of an R element can be solved for the power variable in the function's argument list if necessary due to the connection of the resistor with other elements. Consider, e.g., an R element and a C energy store both of which are connected to a 0-junction as depicted in Figure 2.26.

Assume that the R element is defined by a nonlinear relation $e_2 = \Phi_R(f_2)$, whereas the characteristic of the C energy store is linear for simplicity. Observing the equations of the 0-junction, we have the following equations

$$e_1 = e_2 = e_3 \quad (2.66a)$$

$$e_2 = \Phi_R(f_2) \quad (2.66b)$$

$$C \dot{e}_3 = f_1 - f_2 . \quad (2.66c)$$

Now, suppose the effort e_1 is to be computed for a given flow f_1 . This is only possible if Φ_R has a unique inverse.

If the constitutive relation of an R element does not have a unique inverse relation, then Beaman and Rosenberg assume that the characteristic describes a composite element that captures more than one physical effect [2] and they make the conjecture that with more detailed modelling, such composite elements could be avoided. Despite the problem with the guaranteed uniqueness of the solution, it is common and convenient in electronics to model real devices by resistors with a characteristic that exhibit a negative gradient in some region. This modelling approach is less common in other disciplines. A well known example in mechanics is the approximation of dry friction by a `sign` (signum) function neglecting stiction.

In contrast to stores, R elements may be modulated by a signal.

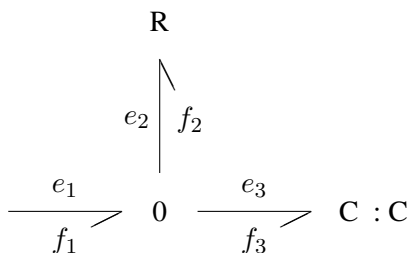


Fig. 2.26 Connection of a nonlinear R element and a linear C store

The RS Element

If isothermal conditions cannot be assumed, e.g., the impact of temperature on the operating point of an electronic circuit, or on the elasticity of the oil in a hydraulic hose must be taken into account, then instead of free energy, total energy including thermal energy must be considered. Since total energy is conserved in a closed system, dissipators become energy transducers converting non-thermal energy into heat. For that reason, a closer look reveals that 1-port R elements have an additional port, i. e., a thermal port. Since the conversion into heat is irreversible, dissipators can be considered to be heat sources in the thermal domain. For that reason, Thoma introduced the symbol ‘RS’ [34]. It accounts for the loss of free energy at the non-thermal port and the production of entropy at the thermal port (Figure 2.27). The letter ‘S’ means source and expresses the source character in the thermal domain. Since this special transducer does not store energy, the principle of power conservation must hold.

$$e f = T \dot{S} \tag{2.67}$$

According to the second law of thermodynamics, $\dot{S} > 0$. Moreover, since $T > 0$, power conservation implies that the power port variables of a dissipator must hold the constraint $e(t) \times f(t) > 0, \forall t > 0$, as stated previously.

Since the power conversion is unidirectional from non-thermal energy into heat, it cannot be represented by a transformer with constant modulus or by a gyrator with constant ratio. The orientation of bonds connected to the ports of the RS element introduced by Thoma indicates the actual direction of the irreversible energy flow.

From Equation 2.67, it can be seen that the additional constitutive relation of an RS element is always nonlinear, even if the relation between the non-thermal power port variables is linear. Assuming $\Delta e = R \times f$, then power conservation yields for the thermal port

$$\dot{S} = \frac{R f^2}{T} . \tag{2.68}$$

Since the absolute temperature T cannot be negative, it follows from the second law of thermodynamics that the resistance parameter must be positive, $R > 0$. Or, the other way round, given a positive resistance, R , a resistor, in fact, produces entropy in accordance with the second law of thermodynamics.

Transfer of the generated heat may take place either via thermal conduction, convection, or radiation.

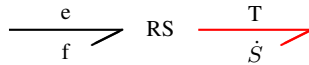


Fig. 2.27 Extension of a 1-port R element according to Thoma

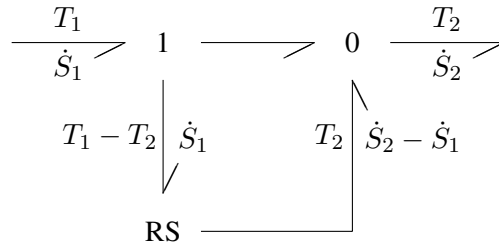


Fig. 2.28 Bond graph model of heat conduction using the RS element (Thoma 1975)

Example: Thermal Conduction

Thermal conduction between two points with absolute temperatures T_1 and T_2 , and $T_1 > T_2 > 0$ may be modelled by means of an RS element as depicted in Figure 2.28 [33].

For the RS element in Figure 2.28 power conservation reads

$$(T_1 - T_2) \dot{S}_1 = T_2 (\dot{S}_2 - \dot{S}_1) . \quad (2.69)$$

With \dot{Q} denoting the heat flow from bond 1 to bond 2, it follows

$$\dot{Q} = T_1 \dot{S}_1 = T_2 \dot{S}_2 . \quad (2.70)$$

That is, the model for heat transfer from point 1 to point 2 is power conservative. From Equation 2.69, it follows that

$$\dot{S}_2 - \dot{S}_1 > 0 \quad (2.71)$$

because the heat flow into the RS element is positive, $(T_1 - T_2) \times \dot{S}_1 > 0$, and temperature $T_2 > 0$. That is, at the point where the heat flow leaves the lump, entropy is higher than the entropy at the point where the heat flow enters the lump. Thus, there is an entropy flow from the higher to the lower temperature and entropy increases in accordance with the second law of thermodynamics.

Heat conduction is also described by Fourier's law in the form

$$\dot{Q} = K (T_1 - T_2) . \quad (2.72)$$

K denotes a constant that depends on the thermal conductivity and the geometry of the heat conductor. Fourier's law and Equation 2.70 for power conservation yield for the two entropy flows

$$\dot{S}_1 = K \frac{T_1 - T_2}{T_1} \quad (2.73a)$$

$$\dot{S}_2 = K \frac{T_1 - T_2}{T_2} . \quad (2.73b)$$

For the irreversible entropy production during heat transfer, it follows

$$\dot{S}_2 - \dot{S}_1 = K \frac{(T_1 - T_2)^2}{T_1 T_2} > 0. \quad (2.74)$$

Thus, the constitutive equations of the RS element read:

$$\dot{S}_1 = K \frac{T_1 - T_2}{T_1} \quad (2.75a)$$

$$\dot{S}_2 - \dot{S}_1 = K \frac{(T_1 - T_2)^2}{T_1 T_2}. \quad (2.75b)$$

Remark 2.7. For an RS element, relating a non-thermal energy domain to the thermal domain, energy that disappears at the non-thermal port *reappears* at the thermal port. In the case of thermal conduction, the amount of heat entering into the R-port remains in the thermal domain. It is fed back to the system via the source port of the RS element. Hence, the flow of produced entropy must be added to a junction in the model. In Figure 2.28, this is done by adding it to the outward power port of the model. Without affecting the port behaviour of the heat conduction model, the flow of produced entropy can also be added to the model's inward power port. In the alternative model, the locations of the 1- and the 0-junction are interchanged.

The heat conduction model in Figure 2.28 assumes $T_1 > T_2$. Index 1 denotes the inward power port and index 2 the outward power port. In the case $T_2 > T_1$, heat conduction is represented by the slightly modified bond graph model in Figure 2.29 [33].

Instead of the RS element, a modulated transformer may also be used for modelling heat conduction as depicted in Figure 2.30. The transformer expresses power conservation (Equation 2.70). The variable transformer modulus results from Equations 2.73a and 2.73b. Finally, the modulation of the transformer correctly reflects the production of entropy.

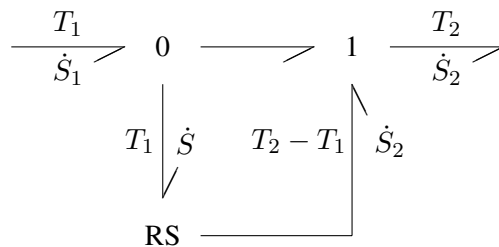


Fig. 2.29 Heat conduction model using an RS element in case $T_1 < T_2$

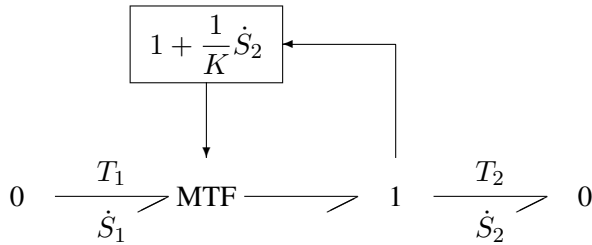


Fig. 2.30 Modelling heat conduction by means of a modulated transformer

$$\dot{S}_2 - \dot{S}_1 = \frac{1}{K} \dot{S}_1 \dot{S}_2 \geq 0 \tag{2.76}$$

Like the bond graph of Figure 2.29, its alternative of Figure 2.29 assumes that $T_1 > T_2$. In case $T_2 > T_1$, the modulus is determined by \dot{S}_2 instead of \dot{S}_1 .

Example: Electrical Current and Entropy Flow in an Electrical Conductor

Let us return to the irreversible conversion of non-thermal energy into heat and consider an electrical conductor of resistance R. The conductor gets heated while an electrical current is flowing. Thus, there is a flow of electrical energy and of entropy as well. The entropy flow can be accounted for by replacing the common electrical 1-port resistor by the submodel depicted in Figure 2.31. The result is an extended bond graph that uniformly accounts for the electrical as well as for the thermal energy flow.

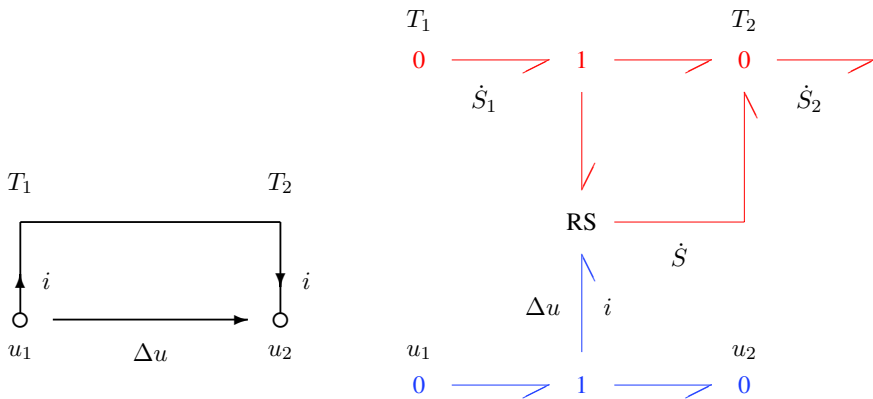


Fig. 2.31 Flow of electrical energy and of entropy in an electrical conductor

A closer look reveals that in an electrical conductor not only electrical energy flow and heat conduction take place. The material also stores thermal energy as illustrated in Table 2.6. The storage of thermal energy can be taken into account by adding a thermal capacitor to the 0-junction. That is, thermal energy storage distributed over the volume of the conductor has been lumped into the point with the lower temperature T_2 in this one-dimensional model.

In conclusion, the R element representing the loss of *free* energy in the non-thermal energy domains is to be replaced by the RS element, if thermal effects must be taken into account. Finally, a ‘dissipator’ does not exist in the thermal domain. The thermal energy leaving at the resistive port of an RS element reenters the system at its S-port.

2.5.5 Memristors

As early as 1961, in his class notes for M.I.T. course 2.751 [27], H. Paynter represented the possible functional relationships between the key variables, e , f , p , q by the so-called *tetrahedron of state*. In this graph, the vertices are associated with the power and the energy variables of a given system while the edges denote functional relationships between them (Figure 2.32).

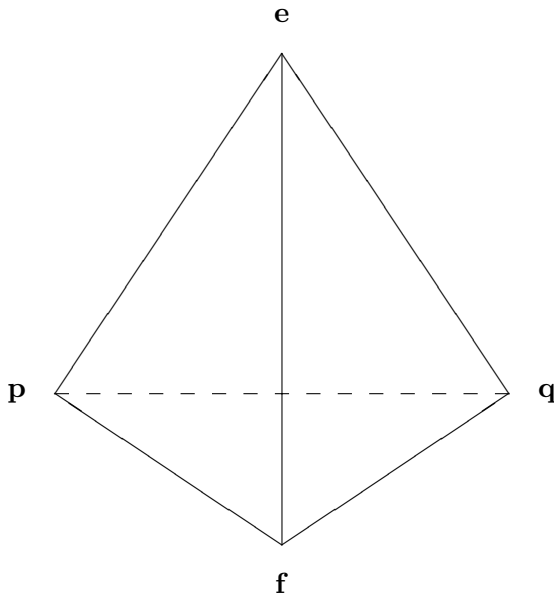


Fig. 2.32 Tetrahedron of state (H. M. Paynter, 1961)

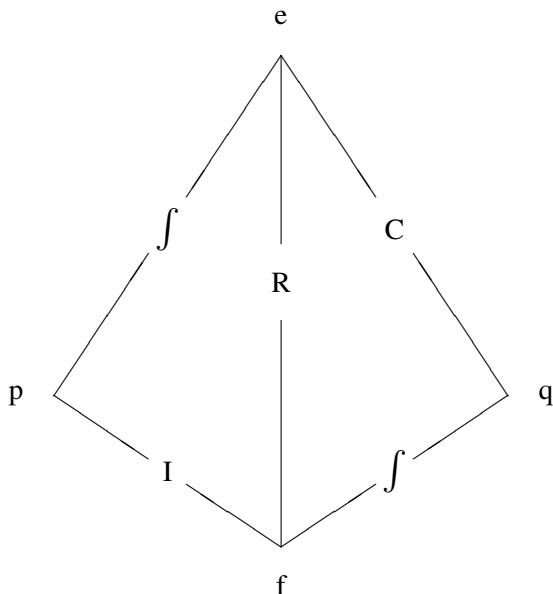


Fig. 2.33 Tetrahedron of state for the three 1-port elements R, C, and I (Karnopp, Margolis, Rosenberg, 2005)

In order to point out the nature of the relationships, other authors have added two integral operators and the basic elements R, C and I, as shown in Figure 2.33 [23].

Interestingly, in his diagram, Paynter has connected the vertices for the energy variables p and q by a dashed line indicating a possible functional relation between the energy variables. The symmetry in the tetrahedron of state may suggest the existence of another basic element besides the two types of energy stores and resistors. In 1971, L. O. Chua postulated the existence of a fourth basic two-terminal circuit element besides the capacitor, inductor and resistor that is characterised by such a relationship between flux linkage and electrical charge. He termed that element *memristor* (memory resistor) and presented an active circuit realisation containing many circuit elements [12]. Inspired by this introduction of a fourth basic circuit element, Oster and Auslander proposed the memristor as a new bond graph element. The abstract of their 1972 article in the *Journal of Dynamic Systems, Measurement and Control* [26] reads:

The “memristor”, firstly defined by L. Chua for electrical circuits, is proposed as a new bond graph element on an equal footing with R,L, & C, and having some unique modelling capabilities for nonlinear systems.

In [26], the authors call the integral of the effort with respect to time, $p = \int_0^t e(\tau) d\tau$, impulse and distinguish between a charge controlled memristor with a constitutive relation

$$p = G(q) \tag{2.77}$$

and an impulse controlled memristor with a constitutive relation of the form

$$q = F(p) , \quad (2.78)$$

where F, G are functions $\mathbb{R} \rightarrow \mathbb{R}$ with trajectories in the first and third quadrant. Differentiation of these two forms of constitutive relations with respect to time results in the two equations

$$e = M(q) \times f \quad (2.79a)$$

$$f = W(p) \times e , \quad (2.79b)$$

where $M(q)$ is called incremental *memristance* and $W(p)$ incremental *memductance*. The equations obtained after differentiation with respect to time show that a memristor turns into an ordinary resistor in case of a linear constitutive relation. Nevertheless, the memristor is a peculiar element. It is dissipative, but at the same time, it is a dynamic element requiring the specification of an initial value. Oster and Auslander considered a tapered dashpot and an electrochemical system with two oppositely charged membranes that can be modelled by using a memristor. In [13], Chua shows that the electrical behaviour with a hysteresis loop in the voltage-current plane observed from a two-terminal nanowire device driven by a low frequency periodic voltage signal can be explained by using a memristor.

The view that there are physical phenomena that justify the introduction of a memristor to be added to the small set of fundamental bond graph elements has not been shared by most members of the bond graph community. Even in the fourth edition of their renowned textbook [23], Karnopp, Margolis and Rosenberg note that “no element will relate p and q ”. This statement is annotated by the following footnote

One can, in fact, define an element corresponding to the hidden edge, the “memristor”. While interesting and occasionally useful, memristors can be represented in terms of other elements to be introduced later, so the memristor will not be considered to be a basic element. . . .

However, in May 2008, Strukov and his colleagues from a Hewlett-Packard laboratory in Palo Alto, California, USA reported in a communication published in the journal nature that they have been able to build an integrated nanoscale circuit device that behaves like a memristor [32]. The device is composed of a 5 nm titanium dioxide film with two layers of different resistivities that are connected to wire electrodes. One layer has a slight depletion of oxygen atoms which results in a lower resistance in comparison to the non-depleted layer. An electric field applied to the device lets the oxygen vacancies serving as charge carriers pass in one direction. As a result, the boundary between the two layers moves and by this way, the resistance of the device changes. This discovery by Stanley Williams and his team at HP Labs has received considerable attention worldwide. It opens up possibilities for various practical applications and gives rise for further research. As a memristor is a dynamic element that is described by an algebraic relation between effort and flow and an additional differential equation, it will affect standard state space modelling.

The dimension of the state space is not only determined by the storage elements, but also by memristors. An extension of the so-called port-based Hamiltonian formulation framework [35] including memristors is considered by Jeltsema and Maks in a paper presented at the Mathmod 2009 conference [19].

2.5.6 Ideal Energy Sources and Sinks

Having considered passive bond graph elements, in the following, we will briefly address sources. As mentioned, they are not part of a system model itself. Rather, they describe boundary conditions of the system, or in other words, the impact of the environment in which the system is embedded. Like in physical systems modelling by means of generalised networks, in bond graph modelling, there are also two types of sources. They are designated by the symbol ‘S’ (source). Their type is indicated either by a subscript ‘e’ or ‘f’ depending on whether the source imposes an effort or a flow on the system (Figure 2.34). Often, the letter characterising the type of a source is not a subscript. That is, the notations S_e and S_f are also used. For sources, it is appropriate to assume the outward orientation as the positive reference direction of the energy flow.

Sources may provide a power variable that is either constant or time dependent. Moreover, sources may be controlled by a signal. In this case, the symbol ‘S’ is prefixed by the letter ‘M’ as shown by the example of a modulated effort source on the right-hand side of Figure 2.34. Furthermore, as resistors, energy stores and all other bond graph elements, the symbol of an element of type source may be annotated by a string that distinguishes the source from other sources of the same type. The string may also be a constant parameter. This is convenient if, for instance, the source represents a vanishing boundary condition, or if the source provides a power variable of constant value (Figure 2.35).

Examples of Sources

With good approximation, gravity near the surface of the earth may be modelled by a constant effort source independent of the coordinates of the place where gravity is effective. Gravity acting on a rigid body of mass m can be represented as shown in Figure 2.35.

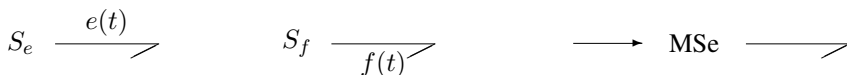


Fig. 2.34 Ideal sources in bond graphs

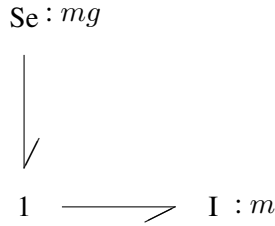


Fig. 2.35 Bond graph fragment accounting for gravity acting on a rigid body

If only the current is of interest that an amplifier feeds into an electrohydraulic servovalve then the amplifier may be captured by a flow source. Furthermore, isentropic boundary conditions of a system may be represented by a flow source imposing a vanishing entropy flow ($dS/dt = 0$).

Another example in which boundary conditions are to be represented by sources is a submodel of a rigid body with two hinge points. The model has two ports for translational velocities and two ports for angular velocities to the outside world. If the body as part of a manipulator is mounted on a basis that does not move, then the two ports of the hinge connected to the bed need to be connected to flow sources that provide zero values.

Controlled Sources

In modelling electronic circuits, it is appropriate and convenient to use controlled sources. The core of a functional model that captures the terminal behaviour of an operational amplifier, for instance, is often a voltage controlled voltage source. Furthermore, if the dynamic behaviour of an actuator in a controlled engineering system is not relevant, it may be represented by a source that is controlled by a signal provided by a signal processor. The latter serves as a controller. A D/A converter converts the signal into an analog signal of low power level, while it is the role of an actuator to provide an effort or a flow at a sufficient power level that affects the system behaviour. Therefore, controlled sources are also available in bond graph modelling. If a source is controlled by a signal, then this is often highlighted by a letter ‘M’ preceding the source symbol (cf. Figure 2.34).

However, one should be aware of potential risks. Controlled sources can be used to represent relations that do not comply with the principle of energy conservation. In [20], Karnopp and Rosenberg stress

... at once very powerful and at the same time hard to discipline, because the nature of the signal input that sets the modules appears to be quite arbitrary. It is possible to represent virtually any system using these modulated 2-ports and other elements in a variety of ways if the moduli are allowed to be dynamically related to system variables. When much of the dynamics of the system is put into signals setting the moduli, the organizational structure which bond graph techniques usually bring to the study of physical system dynamics may be obscured.

This statement is also recalled by Beaman and Rosenberg in [2]. In [1], Beaman and Breedveld conclude

Although the models with controlled sources are functionally correct, these active sources can violate the energetic basis of bond graphs. Hence, they should be avoided in physical models, whenever possible.

In other words, if given relations are essentially represented by means of controlled sources, then the virtue of bond graphs over block diagrams expressing the physical structure gets lost. Moreover, in bond graphs, the use of controlled sources in accordance with the principle of energy conservation, as required by Paynter for physical system models, is not evident. As mentioned in Section 1.4.1, in block diagrams, all kinds of functional relations may be represented that are not necessarily consistent conservation laws from physics. This is also possible in bond graphs if controlled sources are used. If, however, bond graphs are meant to be more than merely a graphical representation of equations, then they should at least comply with the principle of energy conservation as they represent the energy exchange between subsystems.

From power continuity of the RS element (Equation 2.67, it can be seen that the flow of entropy provided at the S-port of the element is not independent of the temperature T . Moreover, it depends on the power fed into the element from the non-thermal side. Hence, from the thermal side, the RS element may be considered a controlled source [18].

Energy Sinks

The environment of a system may be considered intuitively to be a reservoir of infinite capacity. Independent of the amount of entropy flow it receives from the system, the ambient temperature remains constant. A similar observation holds for a hydraulic return reservoir. Independent of the amount of returning hydraulic volume flow, the pressure in the tank remains constant (at atmospheric level). That is, the environment imposes a boundary condition on the system and at the same time it receives an energy flow that does not affect its impact on the system. Both aspects suggest the use of a *sink*, i.e., a source element with a positive reference direction of the energy flow *towards* the element. Isothermal boundary conditions, e.g., may be represented by an effort sink that imposes a constant temperature on the system independent of the amount of entropy it receives from the system. This view does not exclude the possibility that a sink may operate temporarily as a source. Likewise, sources may operate temporarily as sinks.

According to Equations 2.73a and 2.73b, heat conduction between two points in space may be represented by a doubly modulated entropy source (Figure 2.36). This is an alternative to the modulated transformer representation introduced in the previous section (Figure 2.30). While the element is power conservative, it produces entropy. If the reference direction in Figure 2.30 corresponds with the actual direction of energy flow, i. e., there is a heat flow from T_1 to T_2 , then the source receives an entropy flow at the left port while it provides a higher amount of entropy flow at

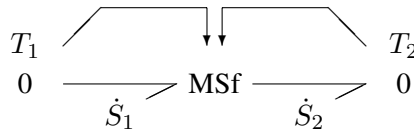


Fig. 2.36 Representation of heat conduction between two temperatures by a doubly modulated entropy flow source

its right port. If the actual direction of the energy flow is opposite to the reference direction, then it becomes a negative source at the port where the adjacent bond is pointing outward and a negative sink at the other port.

2.5.7 Sensors

For the control of engineering systems, sensors are clearly necessary to measure aspects of the system response, e.g., velocities and displacements in mechanical systems. Sensors, in general, perform a conversion of a non-electrical signal of low power into an electrical one, which is converted into a digital signal and fed into a controller via a feedback loop. A characteristic feature of sensors is that they sense a signal without affecting the system. The amount of power they take out of the system is very small and can be neglected. If the dynamic behaviour of a sensor can also be neglected, then the device can be modelled by an energy sink that provides a zero effort or a zero flow.

According to Table 2.7, an effort sensor can be represented by a zero flow sink and a flow sensor can be modelled by a zero effort sink. Table 2.7 also includes alternative representations of ideal sensors that are particularly popular in the community of bond graph modellers in France. The symbol for the type of element clearly indicates which power variable is measured and the signal arrow points out that the conjugate power variable vanishes. In order to distinguish between detectors representing real sensors of measurable variables and fictitious (virtual) detectors of non-measurable variables, a star, *, is added as a superscript to the latter (De* and Df*) [14].

Table 2.7 Representation of sensors in bond graphs

$0 : Sf \begin{matrix} e \\ \swarrow \\ f = 0 \end{matrix} 0$	Effort sensor, zero flow source	De ← 0	Effort detector
$0 : Se \begin{matrix} e = 0 \\ \swarrow \\ f \end{matrix} 1$	Flow sensor, zero effort source	Df ← 1	Flow detector

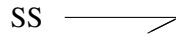


Fig. 2.37 Source-sensor element (Gawthrop and Smith, 1996)



Fig. 2.38 Connection of passive 1-port elements

When a source imposes a power variable on a system, then, at the same time, the conjugate power variable is the feedback of the system onto the source. Hence, the source can be a sensor of the conjugate power. Therefore, Gawthrop and Smith introduced a so-called *source-sensor* element, *SS*, that combines a source with a sensor. This element can be used for representing sources as well as sensors and has proven particularly useful in so-called bicausal bond graphs used for system inversion (cf. Sections 6.3 and 6.7). The *SS* element, shown in Figure 2.37, can represent either an effort source and simultaneously a flow sensor, or it can be a flow source combined with an effort sensor.

The two detector elements (*De* and *Df*) can be considered special cases of this *SS* element. An effort detector corresponds to a zero flow source-effort sensor, while a flow detector is a zero effort-flow sensor (cf. Section 6.7 on bicausal bond graphs, Table 6.2).

These considerations of sources close the introduction of basic ideal bond graph elements. The top-down decomposition of an initial word bond graph model comes to an end when incompletely specified multiports representing submodels have been replaced recursively until bond graphs of submodels are only composed of basic bond graph elements. We shall call a bond graph model at the bottom of the model hierarchy built by basic elements an *elementary* bond graph. Note that, due to the conventions for the reference directions of energy flows, power ports cannot always be connected directly by a bond. For instance, passive elements cannot be joint directly (Figure 2.38).

2.6 Pseudo Bond Graphs

In the previous section, basic elements have been introduced that enable a unified physical modelling approach for all energy domains on the basis of one of the two analogies discussed in Section 2.3. Both analogies, the classical force - voltage analogy as well as the dual mobility analogy, are in accordance with the general observation that in all energy domains, the amount of power transferred between two power ports may be expressed as the product of two power conjugated variables.

Table 2.8 Effort and flow variables in pseudo bond graphs

	Hydraulics	Thermodynamics	
Effort	Pressure	Temperature	Temperature
Flow	Mass flow	Heat flow	Enthalpy flow

This is essential in bond graph modelling. Nevertheless, in considering thermal or hydraulic systems, occasionally, it is convenient to choose effort and flow variables not as indicated in Table 2.4 and to accept that their product is *not* the power transferred between ports. Although the basic principle does no longer applies, again, basic elements describing physical effects in an idealised manner can be introduced. Moreover, the systematic construction of a bond graph from a schematic and, furthermore, the systematic derivation of equations from the bond graph still remain applicable. However, since such bond graphs do not represent energy flows, they are called *pseudo bond graphs* in the literature. In general, such pseudo bond graphs cannot be connected to true power bond graphs via transformers or gyrators. Their advantage is that modelling of thermodynamic systems may become easier (cf. Chapter 10). On the other hand, it may be considered a disadvantage that pseudo bond graph modelling of physical effects is sometimes not quite convincing. If, for instance, in the magnetic domain, the magnetic flux is chosen as a flow instead of the flux rate, then the magnetic capacitor with the capacitance parameter C becomes a resistor with the resistance $R_{mag} = 1/C$. Table 2.8 shows some choices of effort and flow variables in the hydraulic and in the thermal domain for which the product is not the power transferred between ports. The choice of mass flow and an enthalpy flow takes into account that in open systems, an energy flow between subsystems is accompanied by a flow of mass. It is common to consider a control volume and to set up balances for energy, matter and momentum. Whereas a correct true bond graph representation is not always easy, a pseudo bond graph approach is similar to common engineering practice, especially in modelling process engineering systems. Often, pseudo bond graph modelling is considered to be more intuitive. True bond graph modelling of open thermal systems is dealt with in Chapter 10.

Example: Heat Transfer Through a Slab of Material

In the following, pseudo bond graph modelling shall be illustrated by considering heat transfer through an insulating slab of material. Since in this case, there is no flow of matter, according to Table 2.8, the absolute temperature T may be chosen as a effort variable and the heat flow \dot{Q} as a flow variable (The latter variable already has the physical dimension of power).

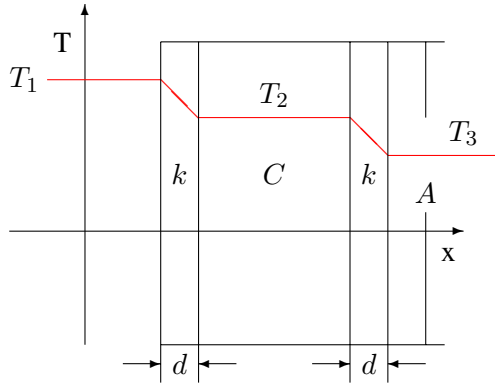


Fig. 2.39 Isolation between two temperatures

Consider two insulating parallel layers of thickness d and area A as depicted in Figure 2.39. The schematic may represent a piece of a wall separating a room of temperature T_1 from the colder surrounding at temperature T_3 .

Inside the wall, between the two layers, a uniform temperature distribution is assumed. If k denotes the thermal conductance coefficient of the two layers, then Fourier's law reads

$$\dot{Q}_1 = \frac{kA}{d} (T_1 - T_2). \quad (2.80)$$

According to the choice of effort and flow, Equation 2.80 may be represented by a combination of a resistor with the resistance parameter $R = d/(kA)$ and a 1-junction accounting for the temperature difference.

Due to the heat flows entering and leaving the lump of material between the two insulating layers, the spatially uniformly distributed temperature T_2 changes. The amount of heat stored in the wall is

$$Q_1(t) - Q_3(t) = C (T_2(t) - T_{20}), \quad (2.81)$$

where $T_{20} := T_2(t = 0)$. Since Equation 2.81 relates temperature T_2 to the displacements Q_1, Q_2 , it may be considered to be the constitutive equation of a 1-port capacitor with the capacitance parameter C . This capacitor model assumes that the change of the volume between the two layers can be neglected. Let c denote the specific heat at constant volume and m the mass of the material lump between the two layers, then $C = cm$. The difference of heat flows rates can be represented by a 0-junction connected to the 1-port capacitor. If heat production inside the room (temperature T_1) is represented by a temperature source and the surrounding, considered to be a reservoir of infinite capacity at temperature T_3 , is represented by a thermal effort sink, then the combination of the two resistors and the capacitor results in a coarse model of the wall displayed in Figure 2.40.

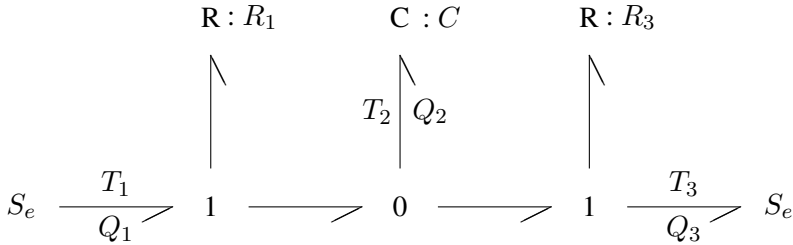


Fig. 2.40 Coarse pseudo bond graph model of a wall

2.7 Systematic Construction of Bond Graphs

Having introduced basic bond graph elements for representing fundamental physical processes and discussed reference directions for energy flows, the question now is how these elements can be combined in a systematic manner in order to come up with a bond graph model of a real physical process at the bottom of a model hierarchy. In other words, how can a non-hierarchical bond graph model be constructed in a systematic manner from a given system schematic. Procedures for each energy domain have been given by Karnopp, Margolis and Rosenberg in their textbook [23]. Breedveld gives a uniform formulation for all non-mechanical energy domains (electrical, magnetic, hydraulic, thermal domain). Since this method does not represent reference nodes in the bond graph and accounts for simplifications of structures in constructing the bond graph, the result does not exhibit the close topological affinity with the system schematic that an initial bond graphs shows if the procedure of Karnopp, Margolis and Rosenberg is applied. It may be a matter of personal preference, whether one or the other of the two procedures or a modification of them is used. At least for students, it might be useful if the construction of an initial bond graph is guided by the topology of the system schematic and if the initial result clearly resembles the system schematic. In subsequent steps, the initial bond graph can be simplified and completed following the given rules. In this book, we follow the procedure of Karnopp, Margolis and Rosenberg. In the following, the construction of bond graphs for planar motion of mechanical systems will be discussed first and then illustrated by an example.

2.7.1 Construction of Bond graphs for Mechanical Subsystems (Translation and Fixed-axis Rotation)

The starting point for a systematic construction of bond graphs for mechanical subsystems are distinct velocities and angular velocities.

1. Identify distinct inertial velocities and angular velocities and represent them by a 1-junction.

They should be annotated by a name in order to express which 1-junction represents which velocity. Ground (zero absolute velocity) is often represented by a 1-junction with a zero velocity (flow) source (Sf) attached. For the sake of a closer structural similarity between a given system schematic and an initial bond graph, the 1-junction representing zero absolute velocity may be repeatedly used in various places of the bond graph.

2. Connect 1-port C energy stores representing springs and 1-port resistors for dashpots to a 0-junction and insert them between proper pairs of 1-junctions.

Insert 2-port transformers, or 2-port gyrators between proper pairs of 1-junctions.

Springs and dampers react to a velocity *difference*. They provide a force or a moment acting equally on both velocity points. Therefore, their corresponding bond graph element is connected to a 0-junction that accounts for the velocity *difference*. The 0-junction is inserted between the two 1-junctions representing the velocities. In Figure 2.41, the symbol 'X' may be replaced either by a 'C' or an 'R' element. In order to ensure that there is a velocity difference at the port of the element, one bond must be oriented towards the 0-junction, the other one must point away from the 0-junction and the third one must be inward to the element port. Transformers couple velocities. For gyrators, two velocities are either inputs or both are outputs. Consequently, both types of elements are inserted between appropriate pairs of 1-junctions. The symbol 'TG' stands for 'TF' or 'GY' (Figure 2.42).

3. Add 1-port inertia elements to their respective 1-junction.

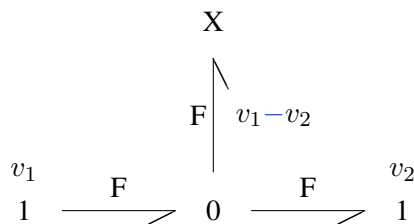


Fig. 2.41 Springs and dampers between two velocities



Fig. 2.42 2-port transformer or gyrator between two velocities

Since the motion of the centre of gravity of a rigid body, or the rotation with respect to a fixed axis is defined with reference to an inertial frame, stores of kinetic energy are attached directly to the 1-junction representing their velocity or angular velocity. The body's kinetic (co-)energy is a function of its velocity.

4. Add 1-port sources and 1-port sinks to appropriate 1-junctions.

Sources and sinks represent boundary conditions. For instance, each body is subject to gravity force, or a motor provides an angular velocity. Since 1-junctions not only represent velocities or angular velocities, but also the sum of forces or moments acting on a body, sources or sinks are attached directly to the proper 1-junction.

5. Orientation of bonds

Having inserted all elements in the bond graph, reference directions for energy flows are assigned to the bonds. For that purpose, it is useful to assume empty energy stores and an energy flow from the sources through the junction structure into energy stores, dissipators and sinks. Following this view, orientations of bonds must comply with rules already discussed for each bond graph element. In particular, orientations at 0-junctions connected to the power port of a C energy store or R element must ensure a velocity difference as depicted in Figure 2.41.

6. Simplification of the bond graph

Finally, all 1-junctions representing a velocity or angular velocity equal to zero are eliminated along with all adjacent bonds. Resulting 2-port junctions with a through orientation of adjacent bonds are replaced by a single bond (Figure 2.43a). The symbol 'J' may be a 0- or 1-junction. If both adjacent bonds of a 2-port junction have an inward orientation, the junction cannot be condensed into a single bond. Such node changes the sign of one of the two power conjugated variables. Consider the 0-junction shown in Figure 2.44. Power conservation and the equality of efforts entails $f_1 + f_2 = 0$. 2-port 1-junctions with inward oriented adjacent bonds may be used in bond graphs of mechanical systems to explicitly represent internal forces in a body that appear in a free body diagram.

Apparently, two junctions of the same type, either a 0- or a 1-junction, can be condensed into a single junction (Figure 2.43b).

A general recommendation finally added to the previously commented construction procedure is to add labels to elements so that they can be easily identified and distinguished from other elements of the same type. Typically, the label is a parameter if only one of the latter exists for a particular element. Otherwise, it can be either a function of the parameter, for example $\sin \alpha$ where α is an angle, a time depen-

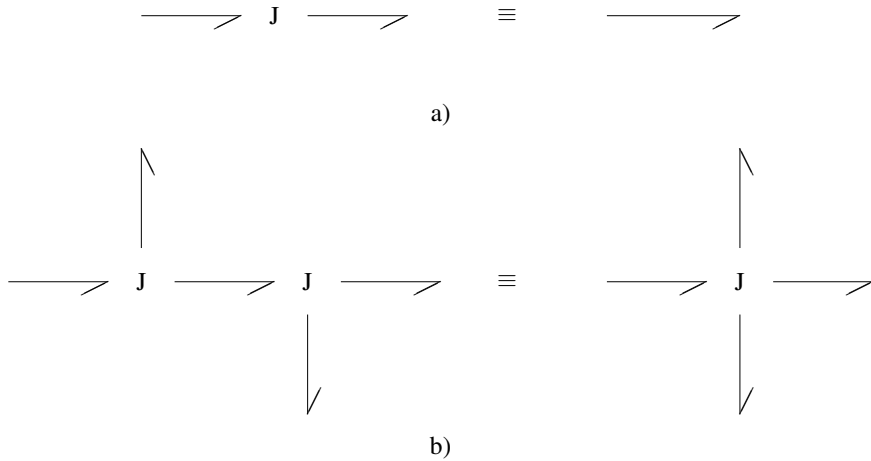


Fig. 2.43 Simplifications of junctions. **a** Replacing junction with two bonds by a single bond. **b** Condensing connected junctions of the same type into one junction

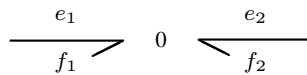


Fig. 2.44 Change in sign at a 2-port junction: $f_2 = -f_1$

dent function for a source, a name that may point to a set of parameters or just an identifier of the element.

Example: Rolling Cylinder on an Inclined Plane

For illustration of the systematic construction of a bond graph of a mechanical system in planar motion, consider the example depicted in Figure 2.45. In this example, a cylinder of radius r and mass m connected to a spring of stiffness k is rolling on a plane inclined at an angle α . It has a moment of inertia J with respect to an axis perpendicular to the paper plane through its centre of gravity. The centre of gravity of the rigid body moves at velocity $v(t)$ parallel to the inclined plane. At the same time, the cylinder is rotating with the angular velocity $\omega(t)$. At the contact point between the cylinder and the inclined plane, a viscous friction force is acting that is proportional to the relative velocity v_r . The contact point as part of the rolling cylinder has the velocity $v - v_t = v - r \times \omega$, whereas the velocity of the contact point as part of the plane is zero. Hence, the relative velocity v_r , effecting the viscous friction force, reads $v_r = v - r \times \omega$. The step by step construction of a bond graph model is shown in Figures 2.46 and 2.47.

For the sake of simplicity, linear characteristics have been assumed for the spring and the viscous friction. Furthermore, it is assumed that the inclined plane is station-

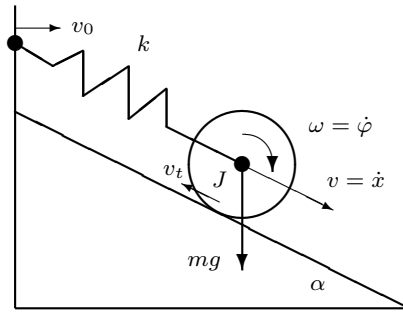


Fig. 2.45 Rolling cylinder on an inclined plane

ary ($v_0 \equiv 0$). The 2-port junctions are then condensed into bonds and the result is depicted in Figure 2.47.

We will come back to this example in the next chapter after having introduced the concept of *computational causality* (Section 3.2) and a procedure for systematically deriving equations from a bond graph.

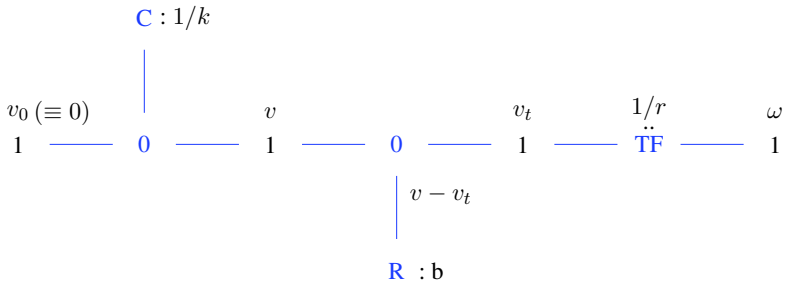
For modelling more complex planar mechanical motion, and in particular for the description of three-dimensional motion of rigid multibody systems, it is customary to use several moving body fixed reference frames. Since quantities are related to local reference frames, transformations between reference frames are needed. In bond graphs, such transformations can be represented by MTF elements. Their modulus depends on displacements. Three-dimensional motion of more complex mechanical systems is considered in Chapter 8. In the following, we will continue by considering the systematic construction of bond graph for non-mechanical subsystems. The notion of non-mechanical subsystems shall express that the modelling will focus on electrical, magnetic, hydraulic, acoustic, or thermal properties. Mechanical properties are assumed to be negligible. For instance, if a body is heated, it is assumed that the mechanical work due to its expansion can be neglected.

2.7.2 Construction of Bond Graphs for Non-mechanical Subsystems

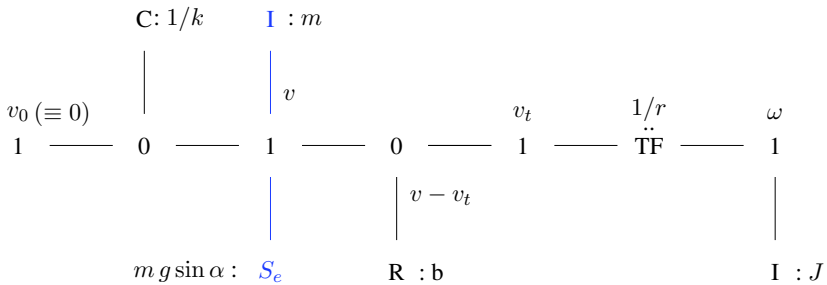
The starting point for the construction of a bond graph for non-mechanical systems are distinct *efforts*. Apart from the interchanged role of efforts and flows due to the classical force voltage analogy, the procedure is quite similar to the one for mechanical systems.



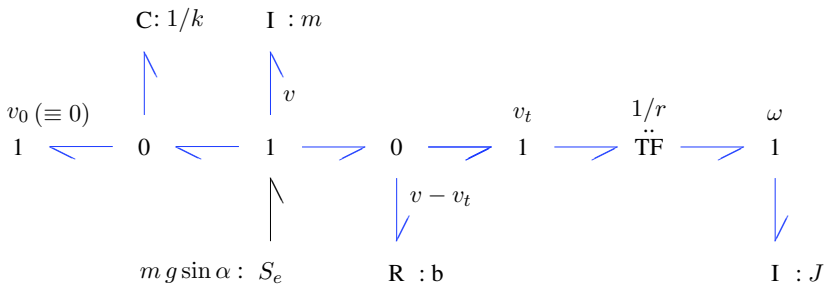
Step 1: Identification of distinct velocities and angular velocities



Step 2: Inserting C stores, dampers and TF elements

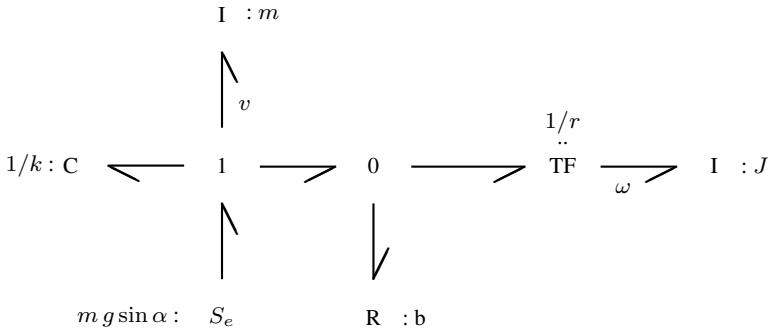


Step 3 u. 4: Adding inertias, sources and sinks



Step 5: Orientation of bonds

Fig. 2.46 Step by step construction of a bond graph model



Step 6: Simplification of the bond graph

Fig. 2.47 Figure 2.46 (continued)

1. Identify distinct efforts and represent them by 0-junctions.

This means that according to Table 2.4, electrical, or magnetic potentials, absolute pressures in hydraulic subsystems, or absolute temperatures in the thermal domain are represented by 0-junctions. Like 1-junctions in bond graphs of mechanical subsystems, they should be labelled by a name in order to distinguish them.

2. The non-mechanical power port of an energy store, a resistor, a 2-port transformer, a 2-port gyrator, or a source is connected to a 1-junction to be inserted between a proper pair of 0-junctions.

In case of an electrical transformer, the 1-junction at both ports of the TF element represent the currents through the coils of the transformer. In bond graphs of hydraulic systems, C elements are inserted via a 1-junction between the 0-junction of an absolute pressure and the 0-junction of the atmospheric pressure. In bond graphs of thermal systems, the thermal port of a C element is attached directly to the 0-junction of an absolute temperature.

In electrical circuits, for instance, there is a voltage drop across an element with two terminals. In order to ensure a difference of potentials, one bond must be oriented toward the 1-junction, a second one away from the junction, and the third one towards the power port of the element (either an energy store or a resistor). The reference direction through the 1-junction corresponds to the reference direction of the current through the element with two terminals. There must also be

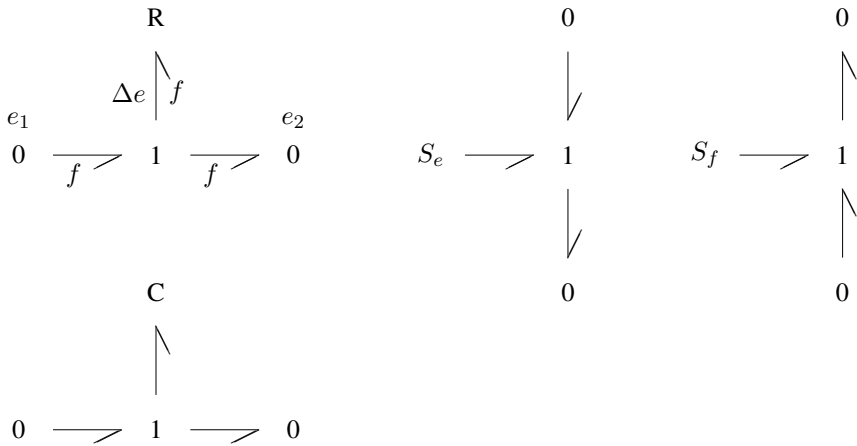


Fig. 2.48 Reference directions of energy flows for non-mechanical 1-port elements

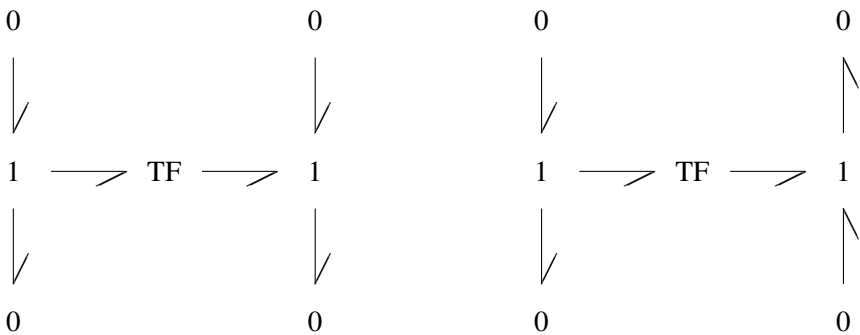


Fig. 2.49 Inserting a 2-port transformer

a through reference direction at a 1-junction if a source is connected to it (Figure 2.48). For each 1-junction connected to a port of a transformer or a gyrator, the reference direction of the energy flow must be through the junction according to Figure 2.49.

- Once all elements have been inserted, reference directions for energy flows are assigned assuming empty energy stores and energy flows from the sources through the junction structure into energy stores, dissipators and sinks. Following this view, orientations of bonds must comply with rules already discussed for each bond graph element.

4. In bond graphs of electrical systems, choose a potential as reference and eliminate its corresponding 0-junction along with all adjacent bonds. If two sub-circuits of an electrical circuit are connected via an isolating transformer, a reference potential must be chosen in each sub-circuit.

In hydraulic subsystems, it is common to choose the atmospheric pressure of the return reservoir. After elimination of its associated 0-junction along with all adjacent bonds, 0-junctions represent gage pressures. This results in a simplification of the construction of bond graphs for hydraulic systems. If 0-junctions represent gage pressures, then C elements are attached directly to a proper 0-junction. As TF elements in bond graphs of hydraulic systems relate a pressure to its associated mechanical force and, at the same time, the volume flow of an incompressible fluid flow to its associated translational velocity, the hydraulic port of the TF element is connected to a 0-junction of a gage pressure while its mechanical port is connected to the 1-junction of a velocity.

5. After elimination of reference nodes along with adjacent bonds, the bond graph is simplified, as has been discussed for bond graphs of mechanical subsystems.

Finally, it is good practice to add labels to elements so that they can be easily identified and distinguished from other elements of the same type. This has already been emphasised in the introduction of bond graph elements and as a general comment to the previous construction procedure for mechanical subsystems.

Example: Electrical Network with an Isolating Transformer

Consider the electrical network depicted in Figure 2.50. The conversion of the circuit diagram into a non-simplified bond graph is straightforward. Distinct nodes in the circuit are represented by 0-junctions. Bond graph elements corresponding to elements with two pins are inserted by means of a 1-junction. The result of the first three steps is shown in Figure 2.51. One can clearly see the topological affinity of the bond graph with the circuit diagram. The circled 0-junctions indicate reference potentials. After their elimination and subsequent simplification of the graph, the bond graph depicted in Figure 2.52 is obtained.

Before we follow the above procedure for the construction of a bond graph of a hydraulic subsystem, some preceding remarks have to be made.

Hydraulic Capacitance of an Oil Filled Volume

As indicated in Table 2.6, the compliance of a fluid can be represented by a C energy store. If E_{fluid} denotes its bulk modulus and Q_c the volume flow into a pressurised volume V filled with a fluid of density ρ assumed to be spatially uniform, then the increase of fluid in the volume due to its compression in the sealed volume results in a pressure increase (Equation 2.82)

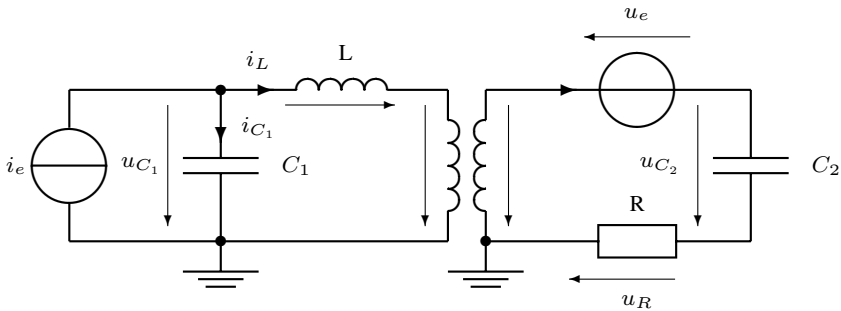


Fig. 2.50 Electrical circuit with an isolating transformer

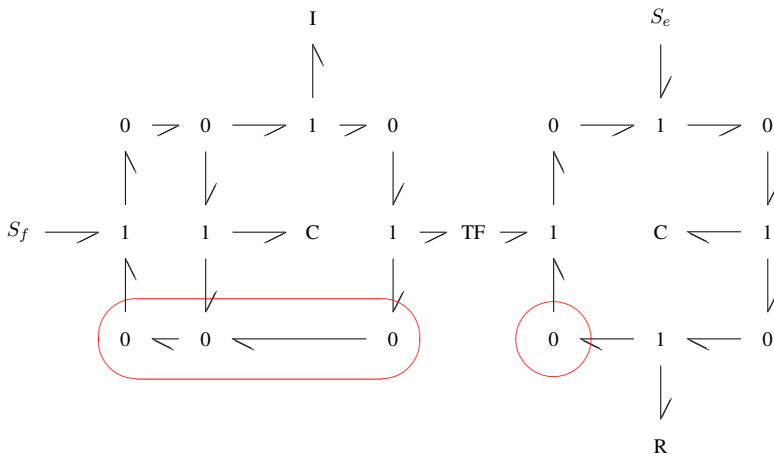


Fig. 2.51 Bond graph corresponding to the electrical circuit in Figure 2.50 after step 1) to 3)

$$p(t) - p(t = 0) = E_{fluid} \times \frac{Q_c dt}{V} . \tag{2.82}$$

That is, the integral of the volume flow Q_c with respect to time is related to the *absolute* pressure, not to a difference, $p_1(t) - p_2(t)$, of two pressures at points 1 and 2 in space represented by two 0-junctions in a bond graph. Consequently, hydraulic C energy storage elements are always inserted between the 0-junction of a pressure assumed to be uniformly distributed in a volume and the 0-junction representing the atmospheric pressure p_0 . Since the latter one is usually taken as reference, the corresponding 0-junction is removed from the bond graph with all adjacent bonds. Thus, a hydraulic C energy storage element is to be connected *directly* to the 0-junction representing the pressure, p , in the pressurised volume V . From Equa-

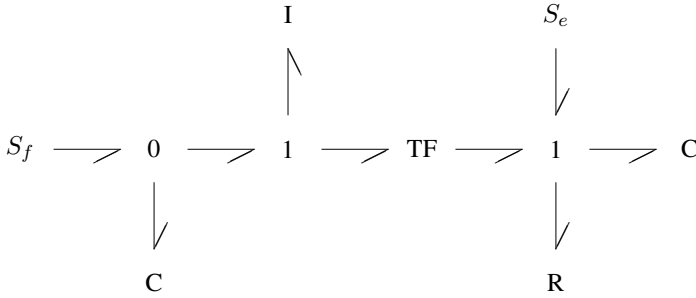


Fig. 2.52 Simplified bond graph corresponding to the circuit example in Figure 2.50

tion 2.82, we obtain for the capacitance parameter of a hydraulic C energy store $C_{hy} = V/E_{fluid}$.

Lossless Hydraulic Tee Junction

Moreover, we recall that the total pressure accounts for the hydrostatic pressure, the dynamic pressure and a gravitational term proportional to a difference of vertical heights. If the energy transported by the fluid is considered, then the kinetic energy and the internal energy of the fluid must be taken into account in addition to the hydraulic energy. It appears that bond graph modelling of thermofluid systems may be complicated. It becomes rather simple for so-called *hydrostatic* systems with low velocity flows and hydrostatic pressures of high values. Consequently, the total amount of power transferred between two ports may be approximated by the product of hydrostatic pressure and volume flow. Under this assumption, a lossless tee junction in a hydraulic circuit can be represented by a 0-junction like nodes in electrical networks. However, if fluid velocities cannot be neglected, dynamic pressures have to be taken into account. Consequently, the hydraulic tee junction can no longer be represented by a potential junction. Consider the hydraulic tee junction shown in Figure 2.53a). Assuming an incompressible fluid, the mass balance becomes

$$Q_1 + Q_2 + Q_3 = 0. \quad (2.83)$$

Let v_i ($i = 1, 2, 3$) denote the corresponding fluid velocities. With the assumption that the pipes have the same cross sectional area A and that the fluid is not accelerated inside the tee junction, the momentum balance reads

$$A(p_1 - p_2) = -(\rho Q_1 v_1 - \rho Q_2 v_2). \quad (2.84)$$

Finally, the energy balance is

$$(p_1 + \frac{1}{2}\rho v_1^2)Q_1 + (p_2 + \frac{1}{2}\rho v_2^2)Q_2 + (p_3 + \frac{1}{2}\rho v_3^2)Q_3 = 0. \quad (2.85)$$

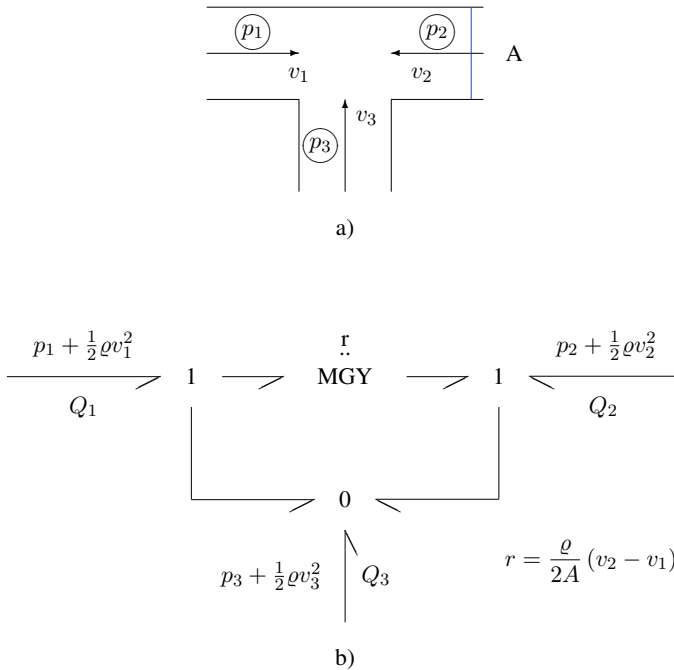


Fig. 2.53 Hydraulic tee junction. **a** Schematic. **b** Bond graph of a lossless hydraulic tee junction accounting for dynamic pressures (Breedveld, 1984)

All three balances can be represented by the bond graph depicted in Figure 2.53.

Apparently, the modulated gyrator disappears and the bond graph reduces to a 0-junction if $v_1 = v_2$.

Hydraulic Ram

At the piston of a hydraulic ram, the hydrostatic pressure, p , in the chamber is converted into a mechanical force, F , acting on the piston of cross sectional area A . At the same time, the piston’s displacement at (slow) velocity, v , entails a volume flow \dot{V} .

$$F = A \times p \tag{2.86a}$$

$$\dot{V} = A \times v \tag{2.86b}$$

As these equations relate efforts and simultaneously relate associated flows, they are the constitutive equations of a transformer, of which the hydraulic port is to be connected to the 0-junction of the gage pressure, p , and its mechanical port to the 1-junction representing the translational velocity v .

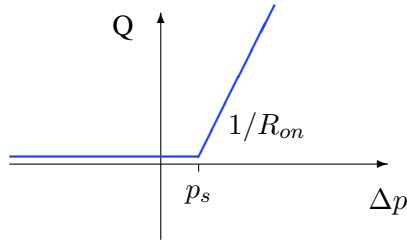


Fig. 2.54 Piecewise linear approximation of the static characteristic of a the pressure relief valve

Pressure Relief Valve

If the fast dynamics of a pressure relief valve are neglected, then it may be modelled as a switched resistor. If the pressure on the pump surmounts an allowed limit p_s , the valve is considered to open immediately and some volume flow returns to the tank passing an opening. In this view, the pressure relief valve may be compared to an electrical diode. Its characteristic may be approximated by the piecewise linear curve depicted on Figure 2.54, where Δp denotes the pressure drop across the valve and Q the volume flow through the valve.

Example: Hydrostatic Drive

With these remarks, a bond graph model of a hydraulic drive can be developed systematically by following the procedure for non-mechanical subsystems.

In the simple example of a hydrostatic drive (Figure 2.55), a constant volume flow pump, protected by a pressure relief valve, delivers a hydraulic volume flow into a double acting unsymmetrical cylinder with cross sectional areas A_1 and A_2 . Its piston moves a mechanical load (not depicted) against a spring of stiffness k . The initial word bond graph in Figure 2.56 directly corresponds with the circuit schematic. In the word bond graph, an effort source accounts for possible disturbances on the mechanical load.

Since the aim is to demonstrate the step by step construction of a bond graph and not to develop a model accounting for given requirements, it is sufficient to build a rather simple model. The constant flow pump may be represented by a flow source and the pressure relief valve (PRV) by a nonlinear resistor neglecting the valve's fast dynamics.

After the first three steps of the procedure for the systematic construction of bond graphs of non-mechanical subsystems, the bond graph depicted in Figure 2.57 is obtained. Like the initial bond graph of the electrical network, it shows a close topological affinity to the hydraulic circuit schematic. The fluid compliance in the trapped oil volumes on both sides of the piston has been accounted for by two C energy stores. If the return tank pressure is chosen as a reference, elimination of the corre-

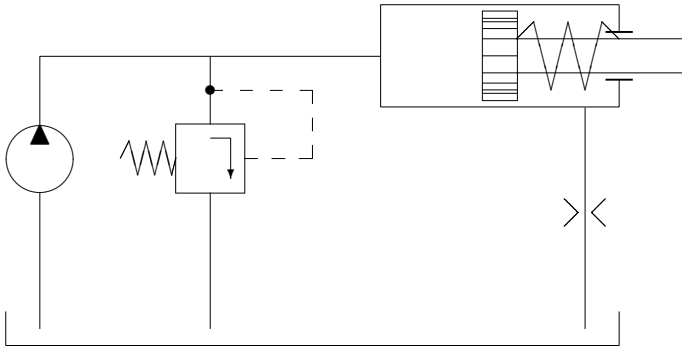


Fig. 2.55 Hydraulic drive

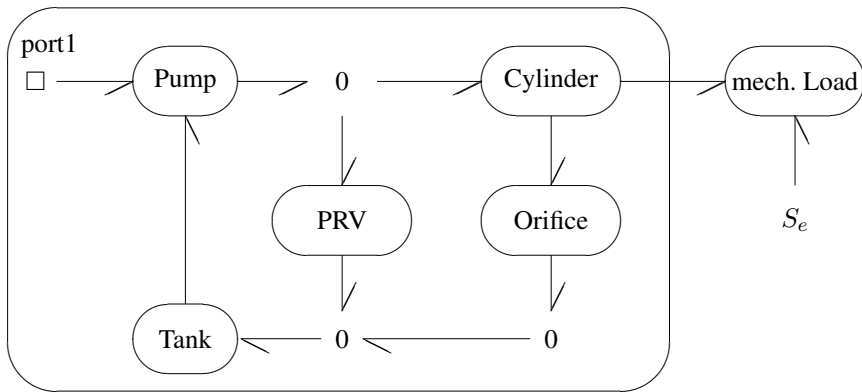


Fig. 2.56 Word bond graph of the hydraulic drive

sponding 0-junctions and subsequent simplification yields the bond graph shown in Figure 2.58.

Example: Electromechanical System

For systems with subsystems in different energy domains, it is obvious to construct a bond graph model for each subsystem following the steps of the corresponding procedure and to connect the submodels by models of the energy transducers. For illustration, consider the mechanical slider crank mechanism driven by an electrical DC motor with constant excitation as shown in Figure 2.59. The armature winding has a self-inductance L and a resistance R . The rod connecting the disk of inertia J to the slider of mass m is assumed to be massless. The piston is moving against an external disturbance force. Figure 2.60 shows the corresponding bond graph of the electromechanical system.

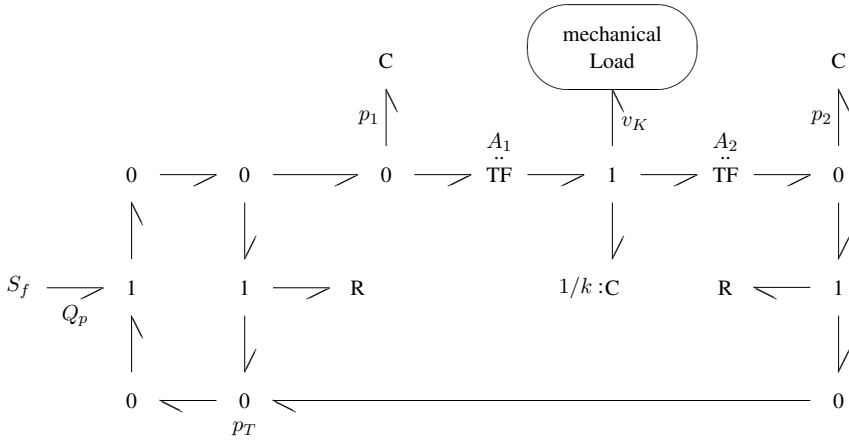


Fig. 2.57 Initial bond graph of the hydraulic drive

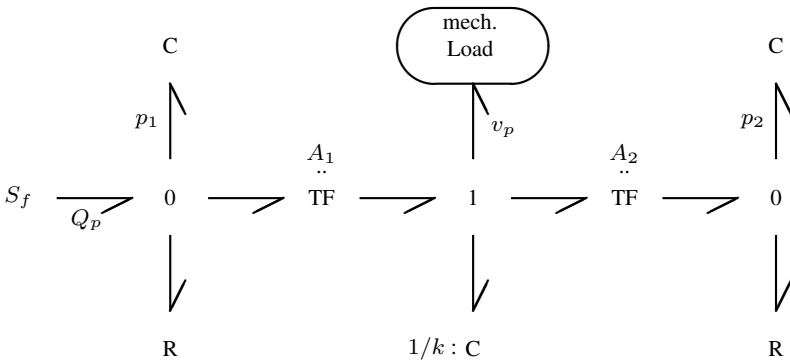


Fig. 2.58 Simplified bond graph of the hydraulic drive

The bond graph of the electrical subsystem shows a clear similarity to the topology of the circuit schematic. In the bond graph of the mechanical subsystem, the 1-junction of the translational velocity, v , of the piston is connected to the 1-junction of the angular velocity, ω , by a displacement modulated transformer. Its modulus, $T(\phi)$, is controlled by the angle, ϕ , of the crank given by the geometry. The motor converting electrical energy into mechanical energy has been simply modelled by the gyrator. It transforms the electrical current, i_a , through the armature winding into a mechanical moment and simultaneously converts the angular velocity, ω , into a voltage. Its ratio is the torque constant k_T . That is, the inertia of the rotor and bearing friction are neglected. Moreover, the shaft between the motor and the pump is considered an ideal power conservative connection represented by a bond.

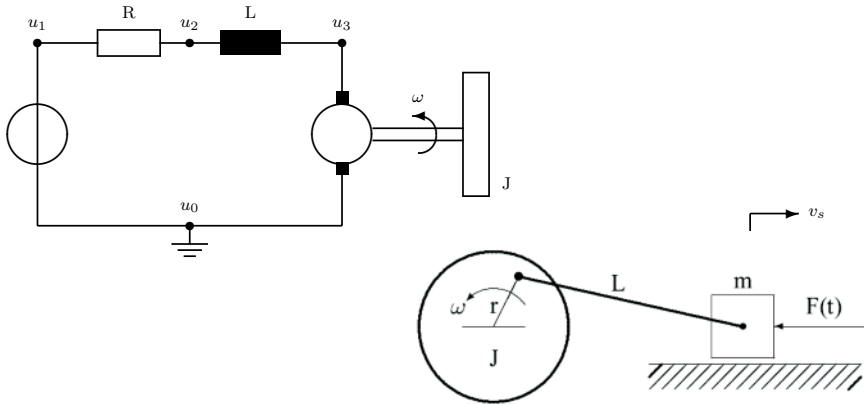


Fig. 2.59 Simple example of an electromechanical system

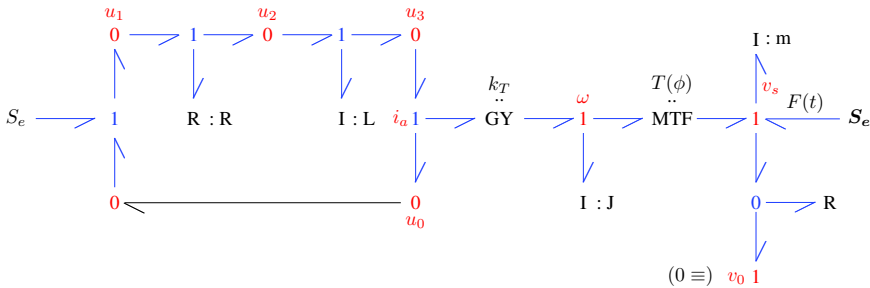


Fig. 2.60 Bond graph of the electromechanical system in Figure 2.59

Torsion and inertia effects are neglected. The friction between the piston and the plane is modelled by a resistor connected to a 0-junction that is inserted between the 1-junctions of the piston and the plane. An external disturbance force $F(t)$, acting on the piston, has been accounted for by an effort source. Mechanical inertias are connected directly to the two 1-junctions representing the angular velocity ω of the disk and the translational velocity v of the piston.

2.7.3 Simplification of Some Bond Graph Structures

Once a bond graph has been derived from a system schematic following the steps of the construction procedures discussed above, it might be appropriate to look for further simplifications. This is useful for visual analysis of the graph, but also for

its automatic processing by software packages. If rule based simplifications are not carried out automatically, then the storage of variables that are not needed and unnecessary operations can be avoided if simplifications of the graph are done manually prior to any further model processing. Figure 2.61 shows some sub-structures in bond graphs and their corresponding equivalent simplification. Equivalences may be proven by setting up the equations for all nodes and subsequent simplification. Figure 2.62 depicts how a transformer or a gyrator together with another element can be condensed into one element.

2.8 Some Remarks on the Choice of Orientations in Bond Graphs

In the previous sections, the basic bond graph elements and two procedures for systematically constructing a bond graph from a system schematic orientations of bonds have been introduced. The rules for reference directions of energy flows do not determine a *unique* orientation for all bonds in a bond graph. There may be several admissible pattern of orientations for the bonds of the junction structure. However, this does not mean that once an undirected bond graph has been constructed, subsequently orientations of bonds in the junction structure may be chosen almost arbitrarily [7]. The result may easily be an inconsistent choice of signs. As Perelson has shown in [29], a bond graph derived from an electrical circuit may no longer represent the network if there is no through orientation of the energy flow at 1-junctions representing the current through an element with two pins.

In 1993, Lamp, Asher and Woodall [24] considered the reverse question under which condition a given bond graph can be implemented by a network. In [24], they observe that the bond graph structure reproduced in Figure 2.63 can be implemented by a network with one transformer (Figure 2.64). A particular feature of this bond graph structure is that the bond loop includes an odd number of bonds having the same energy reference direction. As a generalisation of this observation, they express the conjecture that each bond graph can be implemented as a network if a certain number of transformers and gyrators is used. This, however, means that either the functionality of 1- and 0-junctions is extended or networks are considered realisations of bond graphs even if they contain elements with no correspondence in the bond graph (see Perelson, footnote in [29]). In [30], Perelson shows that if 1-junctions connected to a port of an element have an adjacent bond oriented towards the junction and another pointing away from it (cf. Figure 2.48), then the directed bond graph is equivalent to the network from which it has been constructed. This has been taken into account when the procedure for the construction of bond graphs for non-mechanical subsystems (cf. Figure 2.48) was introduced.

With respect to practical engineering problems, the transformation of a system schematic into a bond graph is considered to be of primary concern. This holds in particular for multidisciplinary systems for which graphical model representations are either not formalised or which use domain specific symbols. Consequently, a

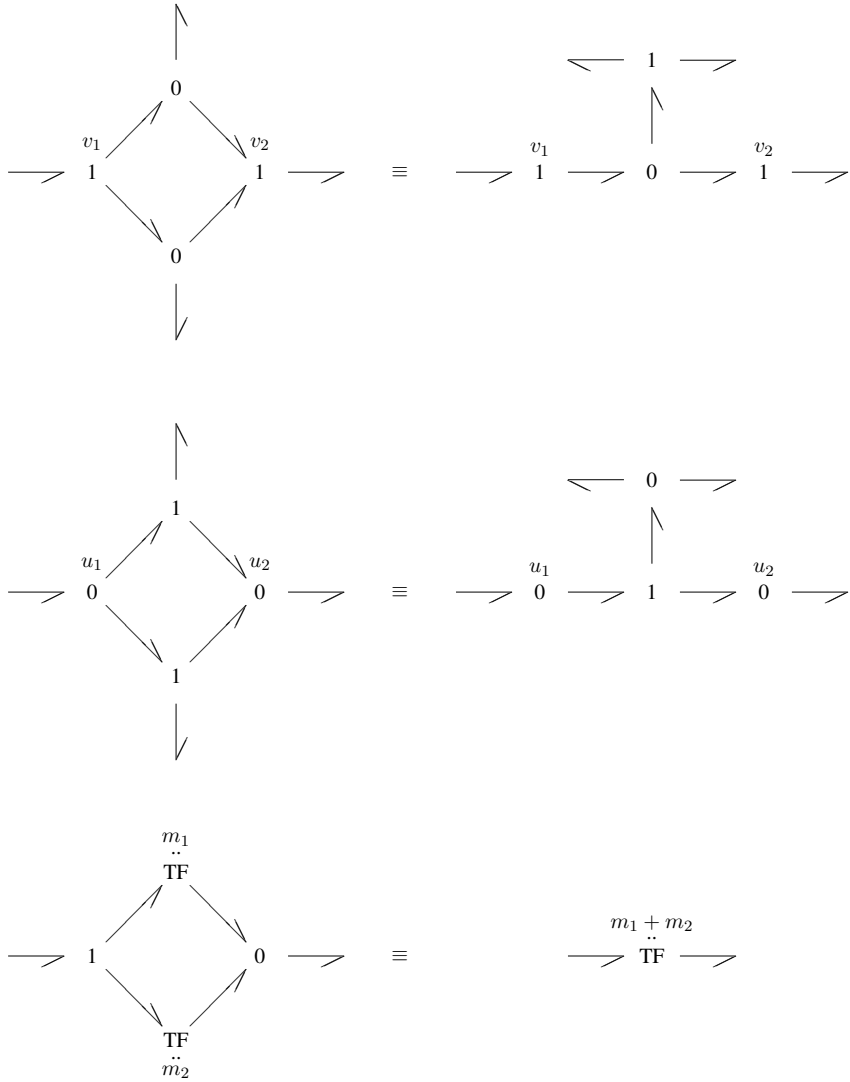


Fig. 2.61 Simplification of some sub-structures

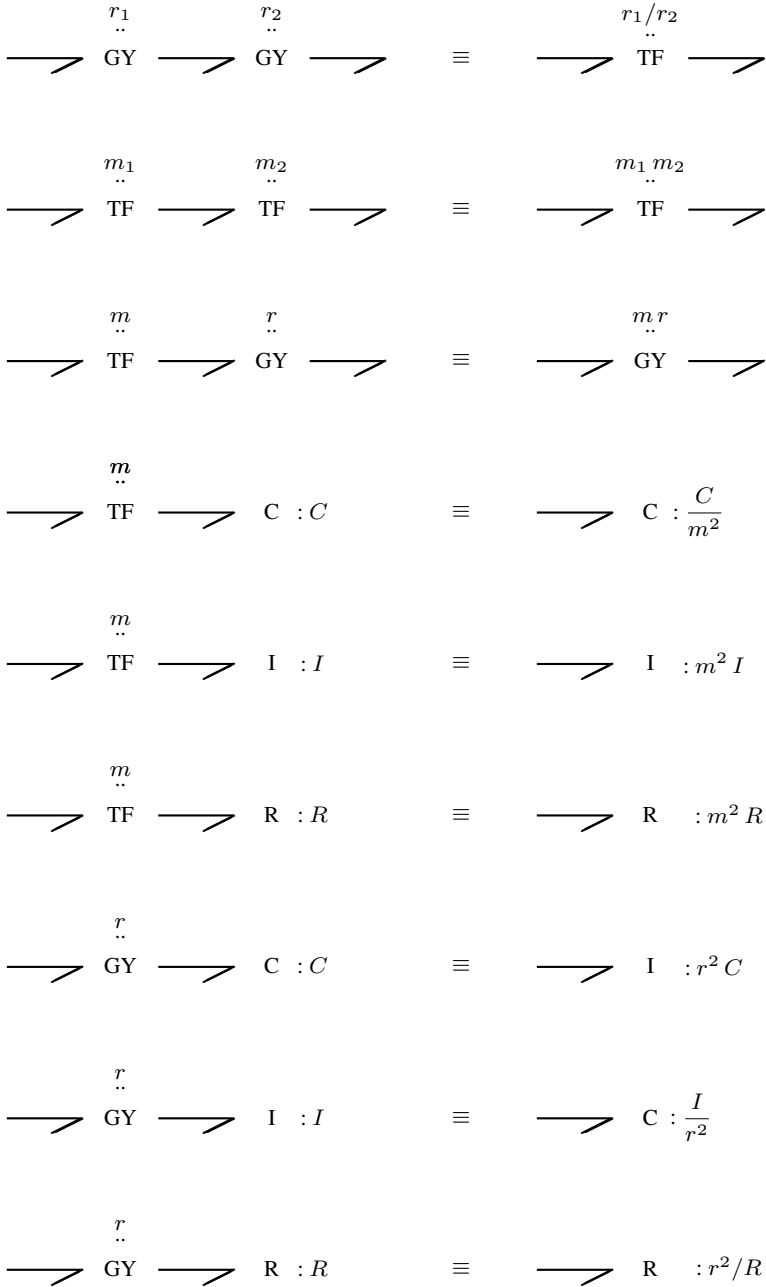


Fig. 2.62 Combination of a transformer or a gyrator with another element

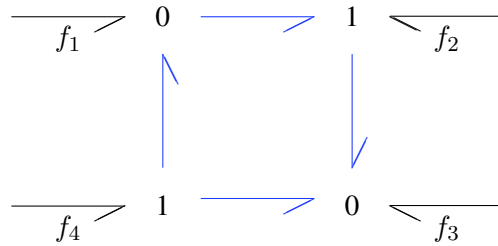


Fig. 2.63 Bond loop with an odd number of bonds having the same orientation

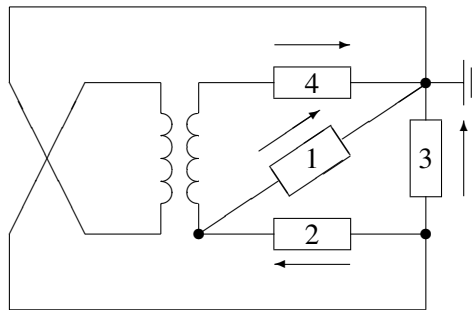


Fig. 2.64 Network realisation of the bond graph part in Figure 2.63 (Lamp, Asher and Woodall, 1993)

unified formalised graphical representation, e.g., a bond graph, is certainly a useful step towards a mathematical model. If the rules discussed above are taken into account, then the equivalence of the resulting bond graph with the initial representation is ensured. Moreover, since there are powerful software packages available for modelling and simulation that, besides other formalisms support bond graphs and are able to directly process them, there is little need to transform a bond graph of a multidisciplinary system into a network to have it processed by a program for network analysis. Therefore, the transformation of a given bond graph into a network is considered less important and will not be furthermore addressed in this book.

2.9 Conclusion

In this chapter, the fundamentals of bond graph based physical system modelling have been provided. The adjective *physical* emphasises that the intellectual decomposition of a system uses the view of an exchange of energy between subsystems and that there are elementary physical processes on the bottom of the model hierar-

chy for which fundamental balances, viz., conservation of energy, conservation of mass, or momentum hold.

Before starting the development of a bond graph model, it must be decided which physical effects are to be taken into account.⁴ In bond graphs, these effects are represented either by a word free of choice for a subsystem, e.g., a transducer or a sensor, or a reserved symbol denoting a basic physical process, e.g., storage of potential energy in a spring. Like networks, bond graphs use the abstraction of spatial concentration of physical properties. It is a characteristic feature of bond graph based or network based physical systems modelling that an initial graphical model is developed by accounting for physical effects in a *qualitative* manner. Moreover, as long as the initial bond graph is not simplified, it shows a clear resemblance to the topological structure in the system schematic.

In contrast to block diagrams, in general, it is not necessary to consider functional relations in order to construct a bond graph. There is no need to precisely know how the power conjugated variables of an element are related. It is sufficient to know which type of an element the constitutive equations define. Consideration and manipulation of equations can be reserved to subtle cases in which it is unclear how coupled physical effects can be modelled under given assumptions. The aim of bond graph based physical modelling is not only to graphically express functional relations, but to come up with a model that complies with conservation laws of physics. Such a task is not easy for open thermodynamic systems in which mass enters and leaves a control volume conveying momentum and energy.

Since the conceptual starting point of bond graph modelling is the energy exchange between subsystems, this modelling approach is particularly suited for multidisciplinary dynamic systems in which several energy domains are involved. Moreover, the introduction of two general power conjugated variables and their assignment to physical quantities in each energy domain enable a uniform description of basic physical process, viz., the distribution or the storage of energy, and energy conversion. The uniform representation is reflected by a few elements that are the same in all energy domains apart from some details, e.g., that there is no inertia in thermodynamics. Like in networks, there are sources, energy stores and resistors. In contrast to networks, the range of symbols is limited to those representing fundamental effects. For instance, a diode can be considered to be a resistor with a special characteristic.

The choice of effort and flow variables is based on an analogy between mechanical and electrical quantities. There are two such analogies, the classical force voltage analogy and the mobility analogy. Both analogies lead to equivalent mathematical models. Following a widely used convention in the bond graph literature, the classical analogy is adopted throughout this monograph.

Especially for thermodynamic and process engineering systems, it is also common to use physical quantities as effort and flow, although their product does not have the physical dimension of power. With such choices, it is convenient to develop

⁴ The proper choice of physical effects that need be included in a model clearly has an impact on the complexity of a model. This subject is briefly addressed in Section 6.9. A recent review of proper modelling techniques is given in the 2008 article [15] by Ersal and his co-authors.

pseudo bond graphs. They do have an intuitive meaning, but cannot be coupled to true bond graphs via transformers or gyrators in general.

True bond graphs as well as pseudo bond graphs can be systematically constructed from a system schematic. For that purpose, there are two procedures, one for mechanical and one for non-mechanical subsystems. Regarding the orientation of bonds, some rules are crucial in order to ensure that a bond graph complies with requirements from physics and is equivalent to the initial system representation.

The next chapter discusses how bond graphs can be augmented so that they are equivalent to a block diagram and how a mathematical model can be derived from a so-called *causally* augmented bond graph in a systematic manner. Often, causal information is added to a bond graph following a basic procedure introduced by Karnopp and Rosenberg. Once this procedure is available, some interesting aspects and approaches can be discussed.

References

- [1] J.J. Beaman and P.C. Breedveld. Physical Modeling with Eulerian Frames and Bond Graphs. *Journal of Dynamic Systems, Measurement and Control*, 110(2):182–188, 1988.
- [2] J.J. Beaman and R.C. Rosenberg. Constitutive and Modulation Structure in Bond Graph Modeling. *Journal of Dynamic Systems, Measurement and Control*, 110:395–402, 1988.
- [3] P.C. Breedveld. Thermodynamic bond graphs: a new synthesis. *Int. J. of Modelling and Simulation*, 1:57–61, 1981. Acta Press, Anaheim (Cal.).
- [4] P.C. Breedveld. A Bond Graph Algorithm to Determine the Equilibrium State of a System. *Journal of the Franklin Institute*, 318(2):71–75, 1984.
- [5] P.C. Breedveld. Multibond Graph Elements in Physical Systems Theory. *Journal of the Franklin Institute*, 319(1/2):1–36, 1985.
- [6] P.C. Breedveld. A Definition of the Multibond Graph Language. In S.G. Tzafestas and P. Borne, editors, *Complex and Distributed Systems: Analysis, Simulation and Control*, pages 69–72. Elsevier Science Publishers, 1986.
- [7] P.C. Breedveld. A Systematic Method to Derive Bond Graph Models. In *Proc. of the 2nd European Simulation Congress*, pages 38–44, Sept. 1986. Antwerp, Belgium.
- [8] P.C. Breedveld. Physical System Modeling: From Components to Elements. In J.J. Granda and F.E. Cellier, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM' 93*, pages 3–4. SCS, SCS Publishing, Jan. 17–20 1993. La Jolla, CA.
- [9] F.E. Cellier. *Continuous System Modeling*. Springer-Verlag, New York, Berlin, Heidelberg, 1991. ISBN: 0-387-97502-0.
- [10] F.E. Cellier. Hierarchical nonlinear bond graphs: a unified methodology for modeling complex physical systems. *SIMULATION*, 58(4):230–248, April 1992.
- [11] F.E. Cellier, H. Elmqvist, and M. Otter. Modeling from Physical Principles. In W.S. Levine, editor, *The Control Handbook*, pages 99–108. CRC Press, Boca Raton, Florida, 1996.
- [12] L.O. Chua. Memristor – the missing circuit element. *IEEE Trans. on Circ. Theory*, CT-18(5): 507–519, 1971.
- [13] L.O. Chua. Nonlinear Circuit Foundations for Nanodevices, Part I: The Four-Element Torus. *Proc. of the IEEE*, 91/11:1830–1859, 2003.
- [14] G. Dauphin-Tanguy. *Les bond graphs*. Hermes Science Europe Ltd., Paris, France, 2000. ISBN: 2-7462-0158-5.

- [15] T. Ersal, H.K. Fathy, D.G. Rideout, L.S. Louca, and J.L. Stein. A Review of Proper Modeling Techniques. *Journal of Dynamic Systems, Measurement and Control*, 130:061008–1–061008–13, 2008.
- [16] A.C. Fairlie-Clarke. Impact of Physical Analogies on Choice of Power Co-Variables. In I. Troch and F. Breitenecker, editors, *Proc. of the 3rd MATHMOD VIENNA, IMACS Symposium on Mathematical Modelling*, pages 729–732, 2000. ARGESIM Report No. 15, volume 2, ISBN: 3-901608-15-X.
- [17] F.A. Firestone. A new analogy between mechanical and electrical system elements. *Journal of the Acoustic Society of America*, 3:249–267, 1933.
- [18] P.J. Gawthrop and L. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [19] D. Jeltsema and J.G. Maks. Port-Hamiltonian formulation of systems containing memristors. In I. Troch and F. Breitenecker, editors, *Proceedings MATHMOD 09 Vienna - Full Papers CD Volume*, number 35 in ARGESIM-Reports, pages 617–628. ARGESIM, 2009. ISBN 978-3-901608-35-3.
- [20] D.C. Karnopp and R.C. Rosenberg. *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics*. MIT Press, Cambridge, MA, 1968.
- [21] D.C. Karnopp and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1975.
- [22] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics - Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons Inc., Third edition, 2000. ISBN 0-471-33301-8.
- [23] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics - Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons Inc., Fourth edition, 2005. ISBN: 0-471-70965-4.
- [24] J.D. Lamb, G.M. Asher, and D.R. Woodall. Network Realisation of Bond Graphs. In J. J. Granda and F. E. Cellier, editors, *1993 International Conference on Bond Graph Modeling, and Simulation (ICBGM 1993)*, pages 85–90. SCS Publishing, 1993. Simulation Series, volume 25, Number 2, ISBN: 1-56555-019-6.
- [25] A.G.J. MacFarlane. *Engineering Systems Analysis*. Harrap, London, UK, 1964.
- [26] G. F. Oster and D. M. Auslander. The memristor: A new bond graph element. *Journal of Dynamic Systems and Control*, 94(3):249–252, 1973.
- [27] H.M. Paynter. *Analysis and Design of Engineering Systems*. M.I.T. Press, Cambridge, Massachusetts, USA, 1961.
- [28] H.M. Paynter. System Graphing Concepts. *Instruments and Control Systems*, pages 77–78, 1970.
- [29] A.S. Perelson. Bond graph sign conventions. *Trans. ASME, Journal of Dynamic Systems, Measurement, and Control*, pages 184–187, 1975.
- [30] A.S. Perelson. Description of Electrical Networks Using Bond Graphs. *International Journal on Circuit Theory and Applications*, 4:107–123, 1976.
- [31] R.C. Rosenberg and A.N. Andry. Solvability of Bond Graph Junction Structures with Loops. *IEEE Trans. on Circuits and Systems*, CAS-26(2):130–137, February 1979.
- [32] D.B. Strukov, G.S. Snider, D. R. Stewart, and R.S. Williams. The missing memristor found. *Nature*, 453:80–83, 2008.
- [33] J.U. Thoma. *Introduction to Bond Graphs and their Applications*. Pergamon Press, Oxford, 1975.
- [34] J.U. Thoma. Entropy and Mass Flow for Energy Conversion. *Journal of The Franklin Institute*, 299(2):89–96, 1975.
- [35] A.J. van der Schaft. Port-Hamiltonian Systems: An Introductory Survey. In *Proceedings of the International Congress of Mathematicians ICM*, volume 3, pages 1339–1365. European Mathematical Society Publishing House, 2006. ISBN 978-3-03719-022-7.
- [36] J. van Dijk. *On the role of bond graph causality in modelling mechatronic systems*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1994.
- [37] P.E. Wellstead. *Introduction to Physical System Modelling*. Academic Press, London, 1979.

Chapter 3

Derivation of Mathematical Models from Bond Graphs

3.1 On the Form of a Mathematical Model

Bond graph based physical systems modelling starts from a *qualitative* consideration of physical effects. In contrast to block diagrams, model development is not guided by setting up equations. The approach is rather object-oriented as explained in Section 1.3. This aspect is essential for bond graph based physical systems modelling. Due to its conceptual approach, bond graph modelling can help better understand the interacting physical processes in a system. Moreover, bond graphs, as an interdisciplinary graphical description language, can support communication between experts from different engineering disciplines. Nevertheless, like other graphical means, they shall enable the (automatic) derivation of equations so that the dynamic behaviour of a system, already existing or still under design, can be analysed by solving these equations numerically or symbolically when possible and appropriate.

One possible way to derive a set of equations from a directed bond graph is to set up the constitutive equations of all nodes of the graph after the edges have been enumerated. The result will be a set of differential-algebraic equations (DAEs) with much redundant information due to the elements of the junction structure. For model developers, this approach is certainly convenient

- if all equations are set up automatically,
- if redundant variables and equations are removed symbolically,
- if the processed set of equations is passed on to a numerical solver for differential-algebraic equations and
- if solutions are available with reasonable computational effort.

However, it is not always appropriate to have a set of differential-algebraic equations generated with auxiliary variables neither with respect to the modelling aim nor with respect to its subsequent numerical computation. If, for instance, the aim is to develop a closed loop control for a system, it is more appropriate to use a state space model of the system, if possible. Furthermore, sometimes only the entries of

the matrices of linearised state space equations are needed for the use in a software program, e.g., MATLAB[®]/Simulink^{®1} [2]. In the case of mechanical multi-body systems, often the equations of motion are generated in the form of Lagrange equations. On the other hand, the most suitable form of a mathematical model to be derived from a graph also depends on the capabilities of the available software tools. In the past, many so-called general purpose simulation programs designed for the Continuous System Simulation Language (CSSL) standard [19] were not able or had limited capabilities to process differential-algebraic Equations (DAE) systems. However, even if a given software program can process DAEs, it may be convenient to get the program to generate an explicit state space model. A model developer primarily concerned with the design of a (control) system may prefer a state space model if numerical difficulties or unnecessary long simulation runs are to be expected. For instance, the program Dymola^{®2} [7] may be directed to partition a large DAE system into smaller coupled sets of equations to solve linear subsystems symbolically and to try to generate a set of explicit Ordinary Differential Equations (ODEs) without algebraic constraints.

Given an implicit set of equations, it is an open question which variable is defined by which equation, i.e., which variables are inputs to an equation and which variable is an output. When dealing with this problem at equation level, the decision can be made by means of a so-called *bipartite graph*³. In a bond graph framework, the same situation arises when the set of equations is generated by setting up the constitutive equations of each node of a bond graph. However, in this case, it is possible to already make the input-output decisions at the bond graph level by adding to one end of each bond a perpendicular stroke, the so-called *causal stroke*, as will be explained in the next section. This causal information assigned to a bond is completely separate from the reference direction of the energy flow indicated by the half arrow. An essential consequence of augmenting the bond graph with causal strokes is that the form of the mathematical model can be determined without formulating and manipulating any equation. After rule based manual or automatic assignment of causal strokes to all bonds, one can decide whether a set of explicit first order ODEs can be derived from the bond graph, or whether it will be a DAE system. If inspection of the causally completed bond graph reveals that the mathematical model to be derived is an ODE set, then its order (cf. Definition 3.12) and the independent state variables can be determined directly from the graph. If causal analysis of the bond graph reveals that the mathematical model has the form of a differential-algebraic system (descriptor form), then inspection of the causal pattern in the bond graph allows for statements about the differential index (see Definition 4.9) of the DAE system.

¹ MATLAB[®], Simulink[®] and Symbolic Math Toolbox[™] are trademarks of the MathWorks, Inc., 3 Apple Hill Drive, Natick, MA, 01760-2098 USA, <http://www.mathworks.com>

² Dymola[®] is a registered trademark of Dynasim AB, Ideon Science Park, SE-223 70 Lund, Sweden, <http://www.dymola.com>

³ In a bipartite graph, variables as well as equations are represented by nodes. If a variable appears in an equation, their corresponding nodes are connected by an undirected edge. See, for instance, [8].

Not only mathematical models in state space form or descriptor form may be derived from a bond graph augmented by causal strokes, but also Lagrange equations (Section 4.10), transfer functions (Section 6.2) or a so-called port-controlled Hamiltonian formulation of equations [6]. That is, bond graphs are a generic model representation from which different formulations of mathematical models can be deduced depending on the purpose of the model and the capabilities of the available software tools. Gawthrop calls bond graphs a *core model* from which other representation forms may be obtained by an order of transformation steps [10].

For small and medium scale bond graphs, equations may be manually derived in a systematic manner. By following bonds back and forth in the graph, as will be explained later, intermediate variables can be eliminated. In order to avoid mistakes, it is safer to refrain from elimination of intermediate variables in the case of large bond graphs and to have a program remove redundant information symbolically and sort the resulting set of equations. If a program is available that can directly process bond graphs, then there is no need to derive any equations. Moreover, in practice, modellers do not need to care about the orientations of bonds or about the assignment of causal strokes. Both information is added automatically by software programs, e.g., 20-sim^{®4} [4], or CAMP-G[®] [3] when power ports are connected. However, not all programs remove redundant information from the set of equations derived from a bond graph. In any case, augmentation of a bond graph by causal strokes provides valuable insight into the model.

- On the one hand, causal information can help the modeller to reveal inconsistencies which may lead to the reflection of (implicit) modelling assumptions. Moreover, the model can be checked for structural properties, e.g., structural controllability.
- On the other hand, causal information can be used by appropriate modelling programs for automatic generation of equations for transient analysis, transfer functions, parameter sensitivities, or equations for robustness study in symbolic form.

Bond graph related aspects of some modelling languages and software tools will be addressed later in Chapter 11.

Since it may be numerically more efficient to solve a set of explicit ODEs rather than a DAE system, and since numerical problems become more severe with higher index DAE systems (see Section 5.1) and Section 5.5), it would be useful to have information available about the form of the mathematical model before model equations are set up and symbolically manipulated. In order to derive such information from a bond graph, the graph must be causally completed by systematically assigning causal strokes to all bonds. For that purpose, the concept of *computational causalities* is introduced in the next section. A procedure for systematic assignment of causal strokes is given in Section 3.3. After a discussion of the question of which variables are to be chosen for state variables (Section 3.4) and after introduction of the key notion of a *causal path*, we will be able to consider the simplest class of

⁴ 20-sim[®] is a registered trademark of the University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands, <http://www.utwente.nl>

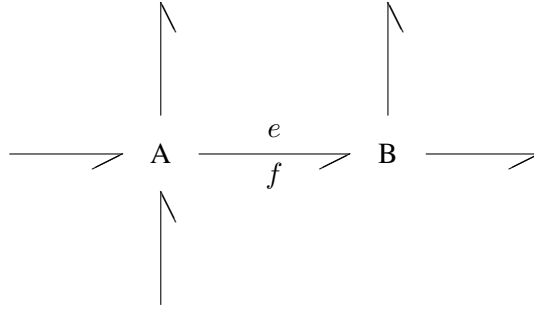


Fig. 3.1 Two multiport elements A and B connected by a bond

bond graphs from which state space models may be manually derived. This will be illustrated by a number of examples. In Chapter 4, classes of bond graphs will be considered that lead to DAE systems.

3.2 The Concept of Computational Causality

3.2.1 The Notion of Computational Causality

In bond graphs, a bond connects two power ports and there are always two power conjugated variables assigned to each bond. If we assume that one of the two power variables is computed in one of the two (multiport) elements connected by that bond while its conjugated variable is computed in the other element, then apparently, we must decide which variable is computed in which multiport element.

Consider the two multiport elements A and B depicted in Figure 3.1. Suppose that the flow f is an independent variable in the constitutive equations of multiport A. Then, there is an equation among the constitutive equations of A that enables the computation of the conjugate effort e . However, the flow f can only be an independent variable in the equations of multiport A if one of the constitutive equations of multiport B allows for computation of f , which in turn requires knowledge of the effort e .

$$e = f_{A_i}(\dots, f, \dots) \quad (3.1a)$$

$$f = f_{B_j}(\dots, e, \dots) \quad (3.1b)$$

The situation could be the other way round as well. Making a choice is called an assignment of *computational causality*. It is guided by some quite obvious rules that will be considered in the following. Causality assignment does *not* require linear

element characteristics. (In some cases causality assignment is determined by the characteristic of an element. For instance, if dry friction is modelled by a 1-port resistor with a slip friction force related to the sign of the velocity, then there is no choice. The velocity must be the independent variable in the equation of the 1-port resistor.) In bond graph literature, often, only the term *causality* has been used, which might be a bit confusing because it could be misunderstood as cause and effect in a physical sense. It is true that Newton's postulate *actio = re-actio* holds, though one cannot determine what is the cause and what must be the effect. A force acting on a body might be considered the cause for the body's motion. The opposite view is also possible. A body in an accelerated reference frame experiences a force. In bond graph modelling, assignment of causality only means the decision of which one of the two conjugated variables at a power port is the external one and, therefore, an independent variable in the constitutive equations of the multiport element under consideration. Consequently, the power conjugated variable is computed in that multiport element. That is, it is a dependent variable with respect to that element. Since in a bond graph context causality is related to the computation of variables, the more precise notion of *computational causality* should always be used. Assignment of computational causality superimposes what van Dixhoorn has called a *computational structure* on the physical structure represented by the non-causal bond graph. There is no unique computational structure. It depends on the modeller's decisions and the procedure how causalities are assigned. The computational structure is not determined by physics and must be distinguished from the physical structure.

The decision as to which one of the two power variables assigned to a bond is computed in which multiport element may be connected with the view that there are two opposite signals in the control theory sense assigned to each bond [17]. Thus, each bond graph element with n ports becomes a functional block with n inputs and n outputs. From this point of view, a causally completed bond graph may be considered a compact representation of a block diagram. In fact, each causal bond graph can be systematically transformed into a block diagram [16], although there is almost no need to do so due to today's software programs, except for a program that accepts graphical models only in the form of block diagrams. In Chapter 11, we will see how different types of software tools can support bond graph modelling. As mentioned in the introduction, not every block diagram corresponds to a bond graph. In block diagrams, pure functional relations may be expressed that are not necessarily consistent with conservation laws from physics.

3.2.2 Representation of Computational Causalities in Bond Graphs

If one of the two power variables at a port is determined to be the independent one, then the conjugate variable simultaneously is determined by the constitutive equations of the multiport element. That is, the choice of computational causality at a port can be expressed by a single bit of information added to the bond connected to the port. The founders of the bond graph methodology introduced a perpendicular

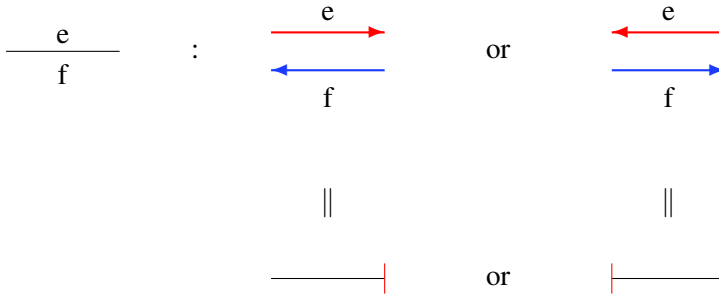


Fig. 3.2 Causal stroke at a bond



Fig. 3.3 Causal stroke and reference direction of the energy flow

stroke added to that end of the bond where the flow is computed in the adjacent element. Using the control theory view of two opposite signals, we can say that the causal stroke indicates the direction of the effort signal. At the same time, the other end of the bond indicates the direction of the flow signal (Figure 3.2).

It must be pointed out that the concept of computational causality is independent of the choice of reference directions for the energy flows. Consequently, there are four possibilities of adding a half arrow and a causal stroke to a bond, as depicted in Figure 3.3. The causal stroke is an additional feature of bond graphs. Hence, we can discriminate more precisely between causal and acausal bond graphs.

Definition 3.1 (Causal Bond graph). A directed bond graph is called a causal or causally completed bond graph if a decision with regard to computational causality has been made for each bond expressed by a perpendicular causal stroke added to one end of each bond.

Remark 3.1. In a causal bond graph, each bond has *two* orientations, one indicating the reference direction of the energy flow and the other determining the direction of the effort signal.

Definition 3.2 (Acausal bond graph). An acausal bond graph is a directed bond graph. That is, reference directions for the energy flows have been defined, though not computational causalities.

3.2.3 Activated Bonds

In measurement and in control processes, the power connected with a signal is so small compared to other energy flows in a system that it can be neglected. In bond

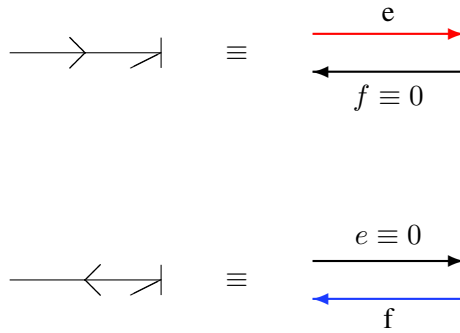


Fig. 3.4 Activated bond (Bell and Martens, 1974)

graph modelling, this means that only one of the two power conjugated variables assigned to a bond is used, while the other one is suppressed. That is, its value is set to zero for all time instances.

Definition 3.3 (*Activated bond*). A bond is called activated if one of its power conjugated variables is set zero for all time instances.

Remark 3.2. In the bond graph literature, some authors also talk about *active* bonds [17]. We will follow Gawthrop and Smith who use the term *activated* bond in their book [11].

In some older publications [1], the activation of a bond is expressed by a double arrowhead attached to the middle of the bond. Depending on which of the two opposite signals is to be activated, the additional full arrow either points to the causal stroke or to the other end of the bond. In the first case, the effort is activated and the flow is suppressed. In the second case, the flow is activated and the effort is set to zero (Figure 3.4).

Since the choice of the causality is independent of the reference direction of the energy flow, the activation arrow is an attribute independent of the half arrow. Usually, activated bonds are replaced simply by full signal arrows as they are used in block diagrams. These signal arrows start either from a 0- or a 1-junction, depending on whether they signify an effort or a flow (Figure 3.5). According to the constitutive Equations 2.15a and 2.15b of a 0-junction, the effort of one bond determines the effort of all other adjacent bonds. Conversely, for the determination of its conjugate flow, the flows of all other bonds are needed. This means that only one causal stroke may point towards a 0-junction. At all other bonds, the causal stroke must point away from the junction. It is the effort determining a 0-junction that can be conveyed by a signal arrow attached to the junction. In Figure 3.5, e_1 is the effort determining the 0-junction. According to Equations 2.16a and 2.16b, the role of a 1-junction is dual to that of a 0-junction. Consequently, the free end of a single bond is allowed to point towards a 1-junction. It is the bond that determines the flow at all other bonds. (In Figure 3.5, it is the flow f_1 .) At all other bonds, the free end must point away

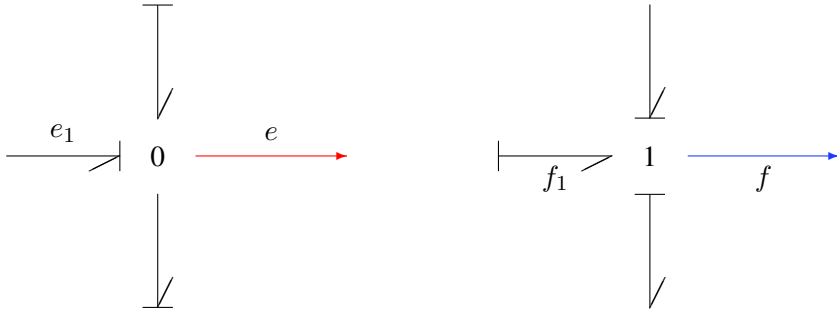


Fig. 3.5 Signal arrows conveying the effort of a 0-junction or the flow of 1-junction

from the junction. That is, the flow determining the 1-junction may be conveyed by a signal arrow attached to the junction (Figure 3.5). At the same time, the single causal stroke pointing away from a 1-junction means that the effort of that bond is equal to the sum of efforts at all remaining bonds.

A signal arrow conveying the information effort from a 0-junction or the information flow from a 1-junction does not affect power conservation at that junction because the power connected with the signal arrow is equal to zero.

3.2.4 Rules for Causality Assignment at the Ports of Bond Graph Elements

There are some obvious and intuitive rules for the choice of causalities at the ports of bond graph elements (cf. Figure 3.6).

Sources

Depending on the type of a source, its output is either an effort (Se) or a flow (Sf).

0- and 1-Junctions

At a 0-junction, only one effort is allowed to be an input. All others are outputs. At a 1-junction, conversely, all efforts except one must be inputs to the junction.

Transformers and Gytrators

Since a 2-port transformer couples the efforts of both ports and, separately, the conjugated flows, both efforts, apparently, cannot be inputs. That is, one effort may be an input while the other one must be an output. Note that a decision with regard to the role of the efforts immediately determines the inverse input-output relationship

for the flows. Conversely, because gyrators switch the role of effort and flow, both efforts of a 2-port gyrator must be either inputs or outputs.

Energy Stores

At a power port of an energy store, any of the two power variables may be the output, although there is a *preferred* causality. If the constitutive equation of a 1-port C energy store, $q = \Phi_C(e)$, (Equation 2.35) is solved for the effort, then the effort is an output and is obtained by integration of the input with respect to time. For that reason, this choice of causality is called *integral* causality.

$$\begin{aligned} e(t) &= \Phi_C^{-1}(q(t)) \\ &= \Phi_C^{-1}\left(\int_0^t f(\tau)d\tau\right) \end{aligned} \quad (3.2)$$

If the constitutive Equation 2.35 is differentiated with respect to time, then the flow is an output and results from differentiation of the input. In this case, the choice of causality is called *derivative* causality.

$$f(t) = \frac{d}{dt} \Phi_C(e(t)) = \frac{d\Phi_C}{de} \dot{e}(t) \quad (3.3)$$

If the input to an energy store with derivative causality immediately jumps to another value at time t , then there is a problem because the output exhibits a pulse of infinite height at that time. This problem does not appear with integral causality.

For the dual type of an energy store, the flow is obtained by integration of the effort with respect to time.

$$\begin{aligned} f(t) &= \Phi_I^{-1}(p(t)) \\ &= \Phi_I^{-1}\left(\int_0^t e(\tau)d\tau\right) \end{aligned} \quad (3.4)$$

That is, for an I energy store with integral causality, the effort is an input. If, however, the constitutive equation $p = \Phi_I(f)$ (Equation 2.44) is differentiated, then the effort is the output.

$$e(t) = \frac{d\Phi_I}{df} \dot{f}(t) \quad (3.5)$$

If integral causality can be assigned to all energy stores in a bond graph, explicit differential equations can be derived from the graph, possibly along with some algebraic constraints due to causally coupled dissipators. If there is a mixture of integral and derivative causalities, then the resulting mathematical model, in general, is a DAE system. In former times, simulation languages were designed for so-called *assignment statements*. Simulation programs based on these simulation languages had limited capabilities to solve differential equations with additional nonlinear al-

gebraic constraints before appropriate numerical solvers, e.g., Gear's DIFSUB or Petzold's DASSL were developed and used by simulation programs. For such reasons, explicit state space models played an important role and are even today still important in control theory. Thus, integral causality is the preferred causality at the power ports of energy stores. Nevertheless, e.g., in bond graphs of mechanical multibody systems, it is quite common that a number of ports of I energy stores have derivative causality because bodies are assumed to be rigidly coupled by joints. Moreover, it is also possible to choose derivative causality as the preferred causality. The connection between computational causalities in a bond graph and forms of mathematical models that can be derived from the bond graph will be discussed in detail in Chapter 4.

Dissipators

There are no general rules for the computational causalities at the ports of resistors. These are indifferent with regard to causality. If the characteristic of a 1-port resistor is one-to-one and has a unique inverse, then the effort or the flow may be the output. Following electrical engineering, sometimes the terms resistance causality or conductance causality are used. As mentioned, there are cases, e.g., Coulomb friction, with no choice of computational causality. Moreover, it can be appropriate to approximate a characteristic with a unique inverse by a piecewise linear characteristic, e.g., in the case of diodes. Hence, there is no choice in causality.

Causal Conflicts

Figure 3.6 summarises the admissible causal patterns at the ports of bond graph elements. Patterns that do not comply with these rules are called *causal conflicts*. For instance, clearly, two effort sources attached to a 0-junction must lead to a violation of the rules (Figure 3.7). If the type of the sources is observed, then two causal strokes point to the junction which violates the junction's permissible pattern. If there is no causal violation at the 0-junction, then one source obtains a causality that does not correspond to its type. Such conflicts indicate physically impossible situations. Two voltage sources in parallel cannot impose independent voltages. In this book, causal conflicts in bond graphs will be circled and highlighted by a flash as depicted in Figure 3.7.

Finally, Figures 3.8–3.10 show the block diagram fragments that correspond to bond graph elements if causalities have been chosen at their power ports. In Figure 3.10, a signal block of type INT denotes an integrator.

In the next section, a procedure will be given for the systematic assignment of computational causalities and their propagation from element ports into the bond graph. The result will be a causally completed bond graph. Such a bond graph could be transformed into a block diagram if bond graph elements were replaced by their block diagram fragments and bonds by two opposite oriented signal arrows.

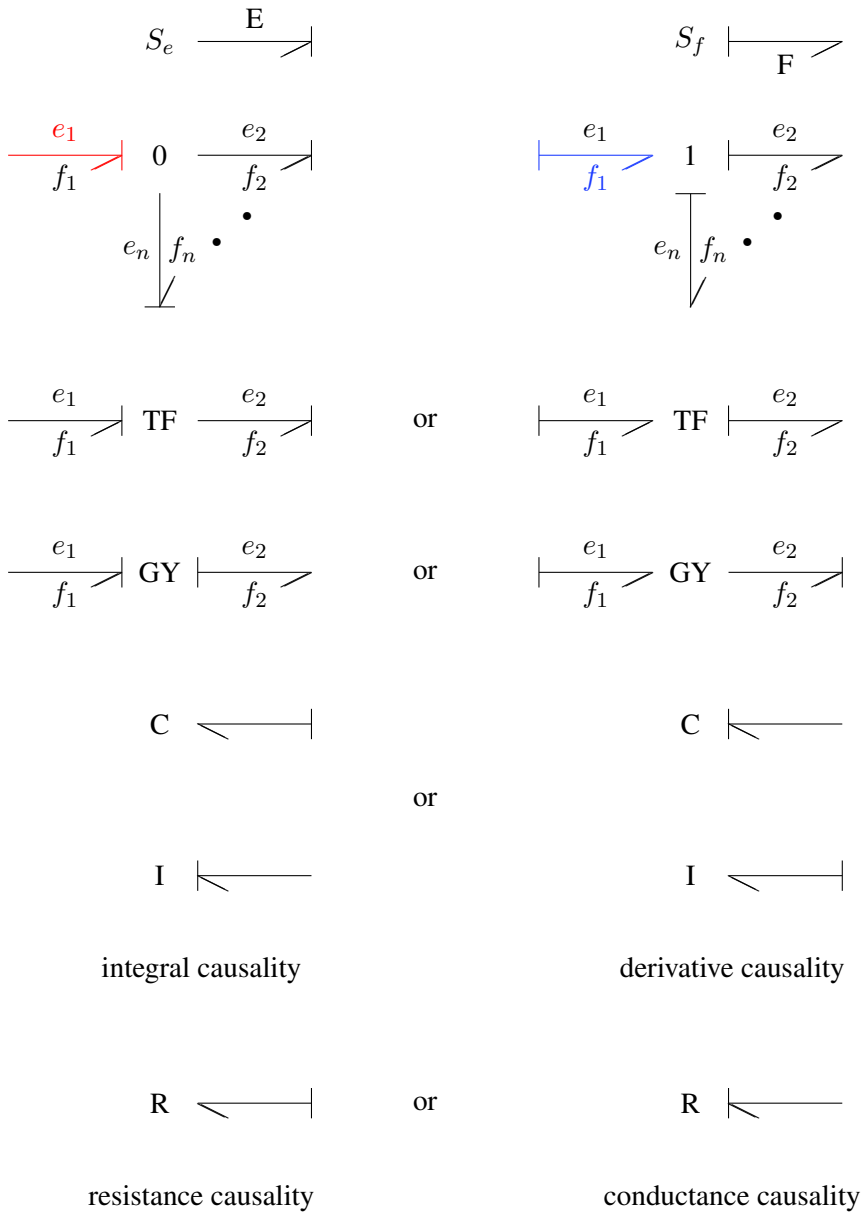


Fig. 3.6 Causalities at the ports of basic bond graph elements

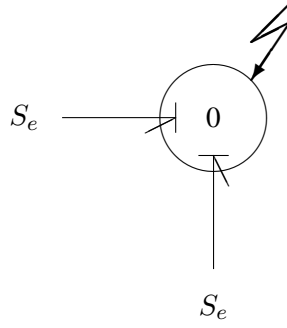


Fig. 3.7 Causal conflict at a 0-junction

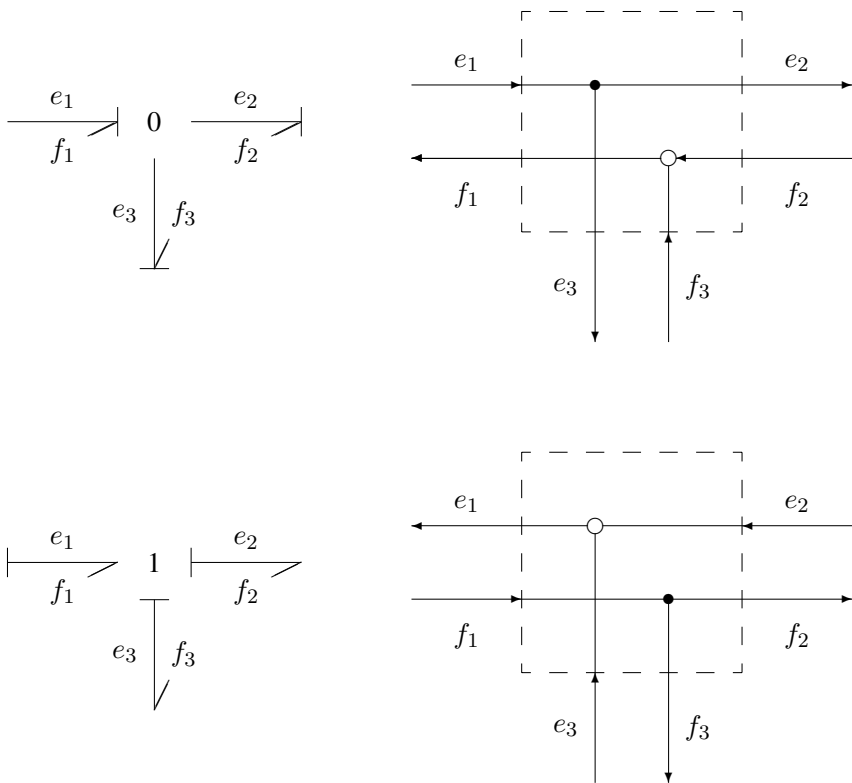


Fig. 3.8 Junctions with causalities at their ports and corresponding block diagram fragments

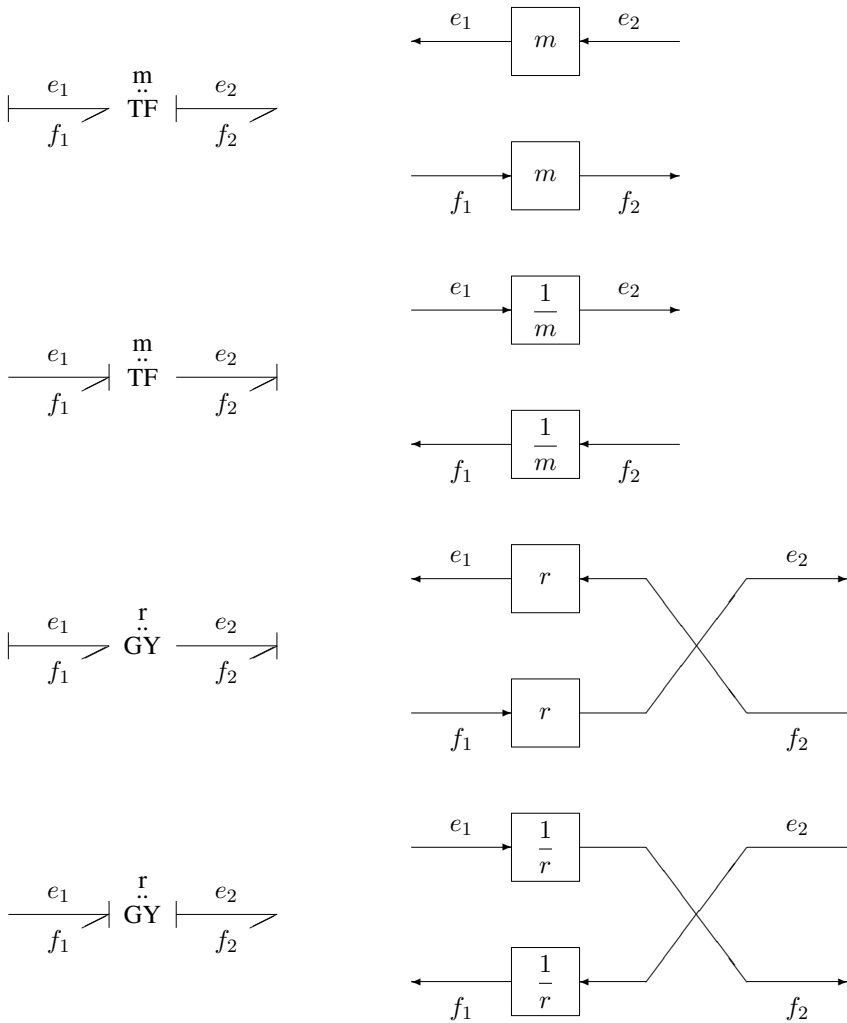


Fig. 3.9 Block diagram fragments corresponding to 2-port transformers and gyrators

3.3 Sequential Assignment of Computational Causalities

For a systematic, step by step assignment of computational causalities, Karnopp and Rosenberg introduced the so-called *Sequential Causality Assignment Procedure* (SCAP) [16, 17], which has become a standard in bond graph literature and will be recalled first. Later, when we consider the connection between causal patterns in bond graphs and forms of mathematical models that can be derived, some

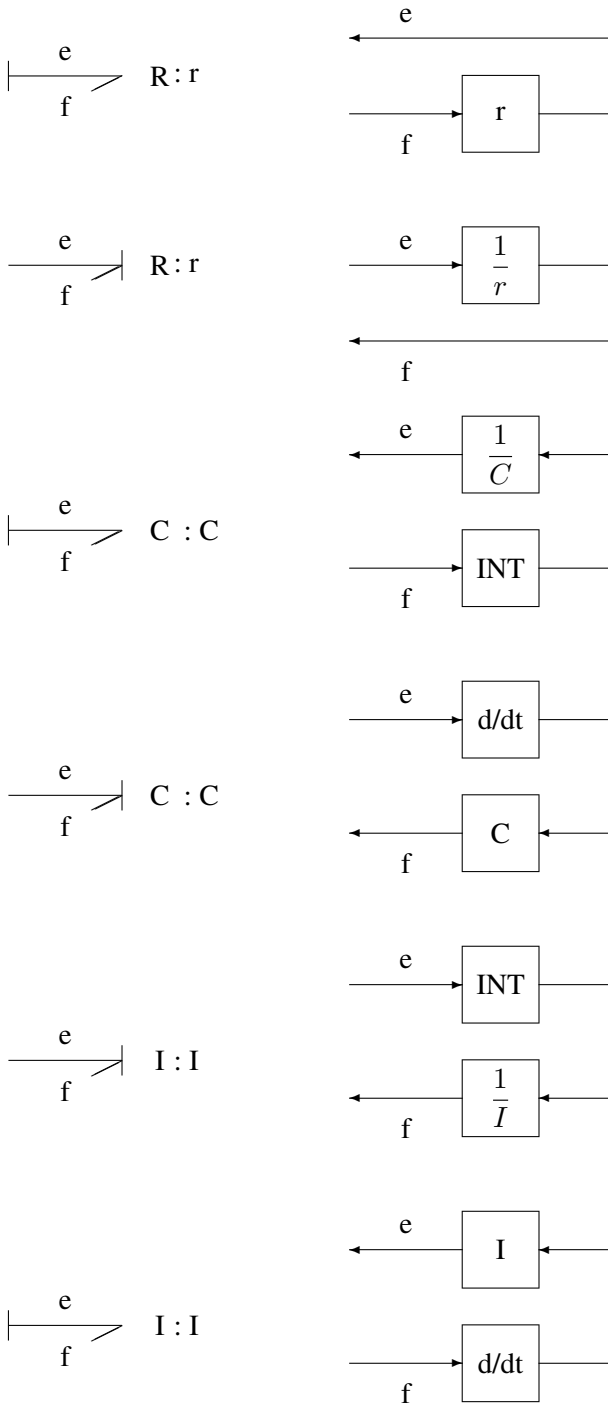


Fig. 3.10 Block diagram fragments corresponding to linear 1-port energy stores and dissipators

modifications of the standard procedure as well as alternative approaches will be discussed.

The Sequential Causality Assignment Procedure (SCAP) of Karnopp and Rosenberg

1. Assign causality to a power port of one of the sources according to its type and *propagate* this causal information into the bond graph through its junction structure as far as possible by observing causality rules at element ports. Repeat this step until all ports of all sources are assigned an appropriate causality.
If causal conflicts appear in this phase, model assumptions must be checked and the model must be changed appropriately.
2. If there is a resistor port with a characteristic that does not have a unique inverse, assign the required causality to ensure correct formulation. For example, the Coulomb friction between two sliding bodies is most commonly assumed to be proportional to the sign of the velocity difference between the two bodies. Propagate this causal information into the bond graph through its junction structure as far as possible by observing causality rules at element ports. Repeat this step until all such resistor ports have their correct causality.
3. Assign preferred integral causality to a port of an energy store and propagate this causal information into the bond graph as far as possible. Propagation of the causality at a storage port may lead to derivative causality at power ports of other energy stores and often entails an assignment of causality at resistor ports. For instance, if an I element and a 1-port resistor representing Coulomb friction are attached to a 1-junction, then the I element port must take derivative causality. Repeat this step until all storage ports are assigned a causality.
4. Finally, if there are still resistor ports or internal bonds without causality, one resistor port or an internal bond must be chosen. Causality is arbitrarily assigned and propagated through the junction structure. This step is repeated until no causally unassigned bonds are left. (If this last step is needed, this means that algebraic loops, namely, implicit algebraic equations, will be part of the mathematical model to be derived from the causal bond graph. However, many of today's software programs supporting bond graph modelling are able to cope with algebraic loops. They just issue a warning or process the model silently depending on the program's settings.)

If there are still unassigned bonds after the fixed causality of resistors and the preferred causality of energy stores has been propagated, then it appears not to be an optimal approach to assign an arbitrary causality on an arbitrary chosen bond still unassigned. In [12], Gawthrop and Smith stress that if bonds remain unassigned, each iteration of the last part of the procedure corresponds with an implicit algebraic equation, a so-called *algebraic loop*. In order to make algebraic variables involved in these algebraic loops explicit in the bond graph, they propose an alternative to the last part of the classic SCAP of Karnopp and Rosenberg. Such details will be addressed in Chapter 5 when combined symbolic, numerical approaches to the solution of DAE systems derived from a bond graph will be discussed. The aim of this

chapter, first of all, is to show how equations can be systematically derived from a causally completed bond graph.

The steps of the SCAP are demonstrated by means of the bond graph shown in Figure 2.52. It corresponds to the network of Figure 2.50. Figure 3.11 shows that the causal information of the sources is not propagated by the junction structure. Preferred integral causality at the element $C : C_1$ is propagated by the 0-junction to the left 1-junction, while preferred causality at the energy store $C : C_2$ is not propagated. Integral causality at the I energy store propagates from the left 1-junction through the transformer to the 1-junction on the right-hand side. Consequently, the flow at the right side 1-junction is determined. Hence, the resistor must have resistance causality. Thus, the bond graph has been causally completed. The remaining steps of the procedure are not necessary in this case.

Alternatively, step 3 of the SCAP could also start with assigning preferred integral causality to the I energy store. Propagation of this information and application of the rules, interestingly, results in a fully causally completed bond graph (cf. Figure 3.12 and Figure 3.13). In practice, often, sources and energy stores determine causalities at all other bonds of the graph. Examples of bond graphs that remain causally incomplete after step 3 are considered in Section 3.5.

3.4 On the Choice of State Variables

If a mathematical model is to be set up, the first question that must be answered is which variables are to be chosen for description of the dynamic behaviour. There is no unique choice of such variables. Since the bond graph methodology pursues a physical modelling approach, the choice is confined to those variables having a physical meaning. Moreover, if their initial values at time $t = 0$ are given and if the inputs to the model are known at all times $t \geq 0$, then they must enable a unique computation of all other system variables. Variables that meet these requirements are called *state variables* [9].

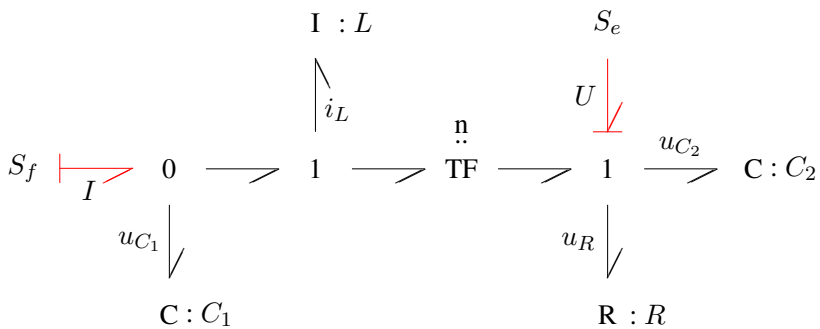
Definition 3.4 (*State variable, System state*). Suppose there are q inputs u_1, \dots, u_q to a dynamic system and n intermediate variables x_1, \dots, x_n . Moreover, physical laws may yield n differential equations

$$\dot{x}_i(t) = f_i(x_1(t), \dots, x_n(t); u_1(t), \dots, u_q(t)) \quad i = 1, \dots, n. \quad (3.6)$$

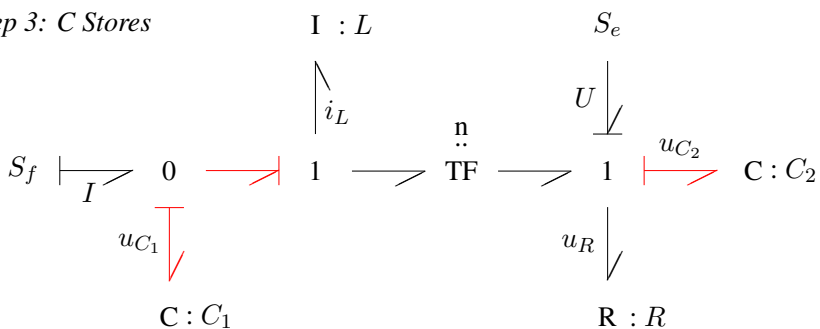
Let $t_0 \geq 0$ be an arbitrary time point. For all times $t \geq t_0$, values $u_1(t), \dots, u_q(t)$ of all q system inputs may be known. Then, n intermediate variables $x_1(t), \dots, x_n(t)$ are called *state variables* if they are uniquely determined for all times $t > t_0$ provided their initial values $x_1(t_0), \dots, x_n(t_0)$ are given. The set of all values $x_1(t), \dots, x_n(t)$ at time $t \geq 0$ is called the *state* of a system.

Apparently, energy variables, in particular, may be chosen as state variables. They quantify the content of the energy stores at all times $t \geq 0$. In this context, we will talk more precisely of the *energetic* state of a system. The energy variable of

Steps 1 and 2



Step 3: C Stores



Step 3: I Stores

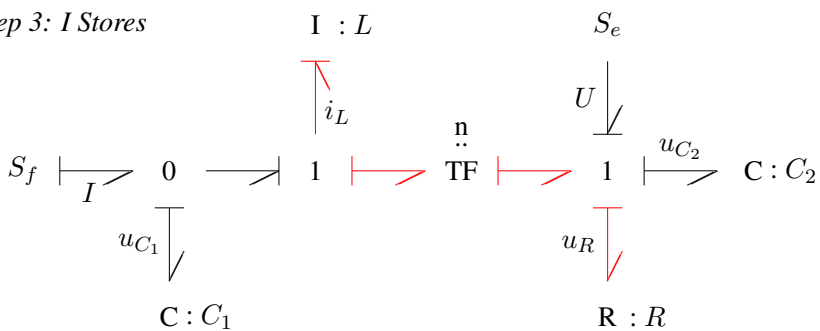
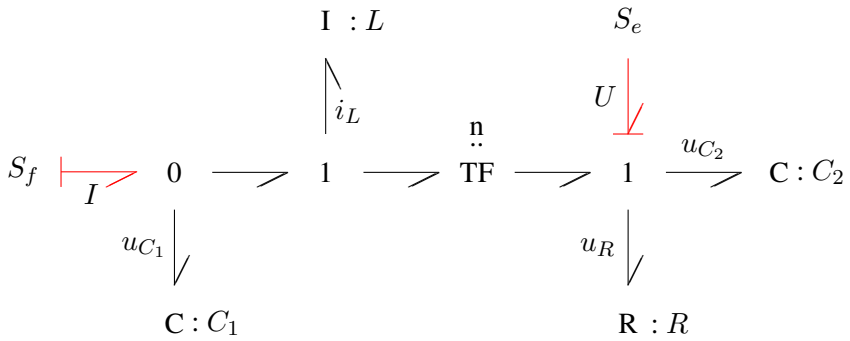
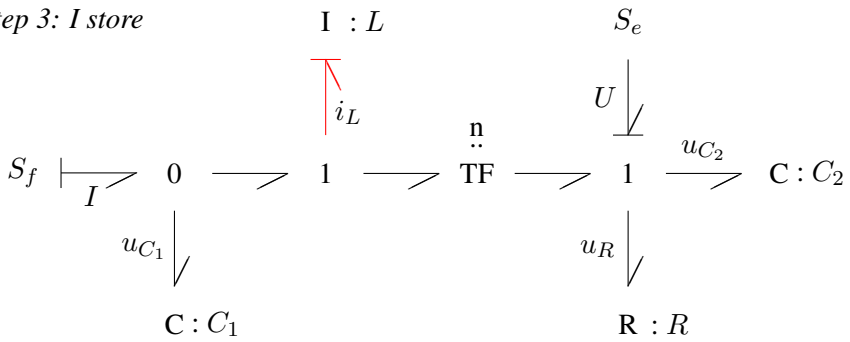


Fig. 3.11 Assignment of causalities to a bond graph

Steps 1 and 2



Step 3: I store



Propagation of flow i_L

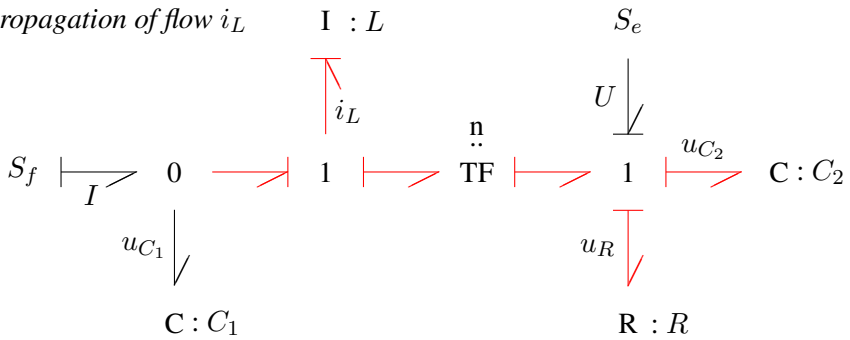


Fig. 3.12 Starting step 3 of the SCAP with assigning preferred integral causality to the I energy store

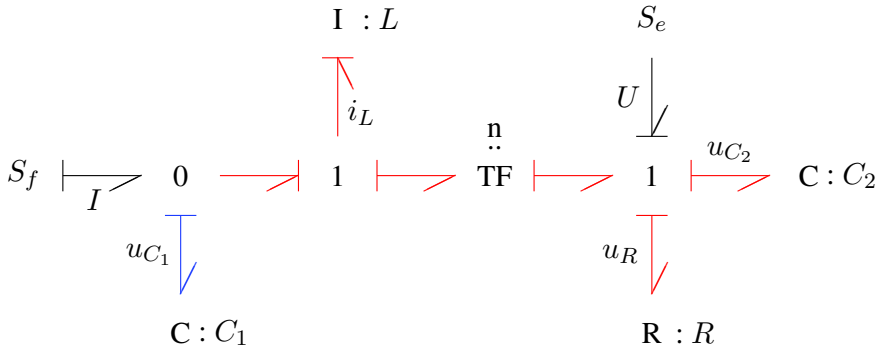


Fig. 3.13 The C store $C : C_1$ must take integral causality

a 1-port energy store is connected to the co-energy variable by the energy store’s constitutive equation. Since characteristics of energy stores are assumed to have a unique inverse, also the co-energy variable may be chosen as a state variable. It is a power variable and the output of an energy store with preferred integral causality. In bond graph literature, some authors chose the energy variables of energy stores in integral causality as state variables, while others use their co-energy variables. In this book, the latter choice is adopted⁵.

If a 1-port energy store must receive derivative causality in the process of causality propagation, then its energy variable is algebraically related to the energy variables of the energy stores with integral causality. In this case, its state variable is called a *dependent* state variable. Energy stores with derivative causality are classified accordingly as *dependent* stores. Stores with integral causality are called *independent*. These algebraic dependencies between the energy variables of energy stores of the same type may be especially found in models of rigid multibody systems due to geometric constraints (cf. Chapter 8).

Transformation of a Dependent Energy Store over a Transformer

Figure 3.14 depicts a simple but typical situation in which two inertias are connected by a transformer with *constant* modulus n .

Since the transformer couples the flows at its ports, only one of the two velocities can be independent and can be chosen as a state variable. Consequently, one of the two I energy stores must accept derivative causality. In this context, we want

⁵ Causally completed bond graphs may be converted into a block diagram if needed. While in block diagrams, usually integrator outputs are chosen as state variables, in bond graphs, the outputs of energy stores in integral causality are chosen as state variables (in this book). For a C energy store in integral causality, the output of the integrator is the generalised displacement q . The state variable e_C contributed by the C element is related to q by the energy store’s one-to-one function $\Phi_C : \mathbb{R} \rightarrow \mathbb{R}$. That is, there is a transformation relating state variables in the bond graph and state variables in the corresponding block diagram.

are energy variables. Alternatively, the voltage drop across capacitors and the current through inductors may also be chosen as state variables. The latter variables are co-energy variables. Each loop of capacitors and each cut-set of inductors reduces the number of independent state variables by one. According to Kirchhoff's voltage law and a more general form of his current law, capacitors in a loop and inductors in a cut-set are not independent. Another reason for a number of state variables smaller than the number of energy stores are controlled sources that establish relations between the variables of different energy stores.

3.5 Systematic Derivation of Equations from a Bond Graph Using Computational Causalities

First, the important notion of a *causal path* is introduced. It is used in the procedure for manual derivation of equations from bond graphs. Moreover, it is a frequently used term in the following.

Definition 3.5 (*Causal path*). A sequence of bonds from one power port of an element to a power port of another element is called a causal path if there is no 2-port gyrator in between and if all bonds have their causal stroke at the same end.

A cascade of bonds between two power ports with a gyrator in between is called a causal path if all bonds on one side of the gyrator have their causal stroke at the same end, while all bonds on the other side of the gyrator have their causal stroke on the opposite end. That is, the gyrator switches the direction of efforts on one of its sides.

Figure 3.16 shows two bond graph fragments with a causal path from port 1 to port 2 and from port 3 to port 4. It is obvious that there may be causal paths with joint edges, or closed causal paths leading to further definitions that are used in the next chapter when considering various causal pattern in bond graphs.

Definition 3.6 (*Disjoint causal paths*). Causal paths that do not share any bonds are called *disjoint*.

Definition 3.7 (*Bond loop or mesh*). A closed cascade of bonds is called a *bond loop*, a *mesh*, or a general mesh.

Definition 3.8 (*Simple mesh*). A bond loop is called a *simple mesh* if it includes no transformers, no gyrators, and no 2-port energy stores or 2-port resistors. In other words, a simple mesh is a loop of bonds that alternately interconnect 0- and 1-junctions.

In contrary, general meshes may contain any series of two-port elements and junctions.

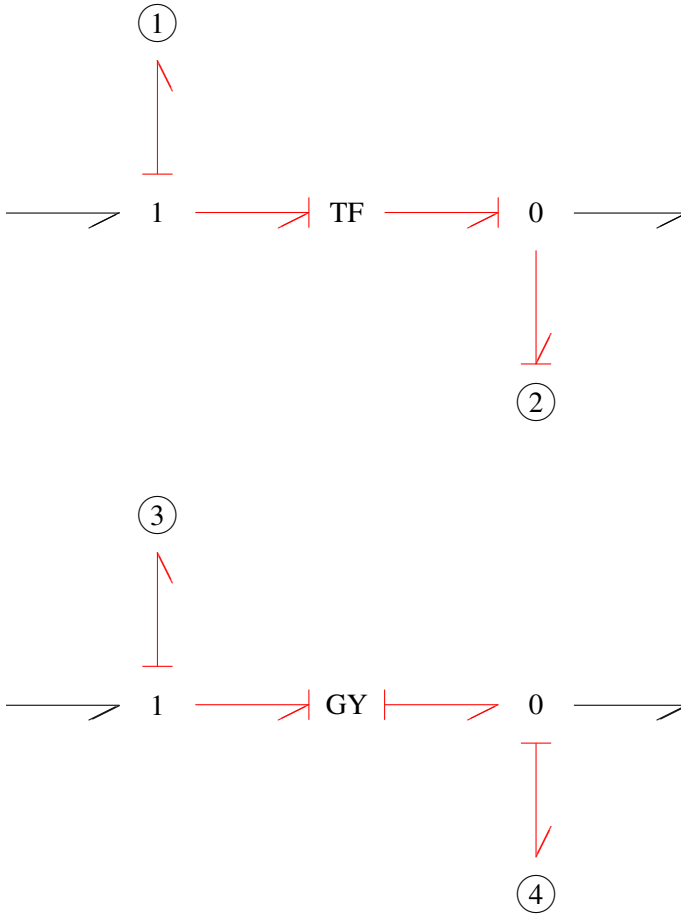


Fig. 3.16 Causal paths between two ports

Definition 3.9 (*Simple even (odd) mesh*). A simple mesh is called even (odd), if an even (odd) number of its bonds has the same energy flow reference direction in a clock-wise or counter-clockwise sense.

Definition 3.10 (*Causal loop*). If the bonds of a causal path only connect elements of the junction structure and if the causal path is closed, then it is called a *causal loop*.

Figure 3.17 shows an example of a causal loop with bonds 1-2-3-4.

Definition 3.11 (*Causal mesh*). A *causal mesh* is a closed causal path with an odd number of gyrators (Figure 3.18).

First, we will only consider simple examples of causal bond graphs with the following features:

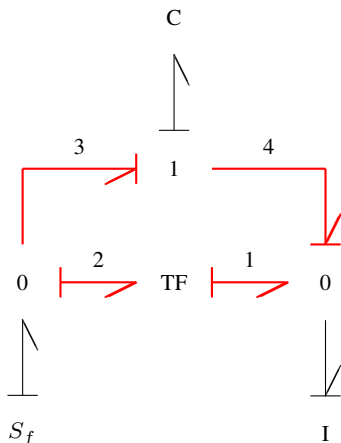


Fig. 3.17 Example of a causal loop with bonds 1-2-3-4

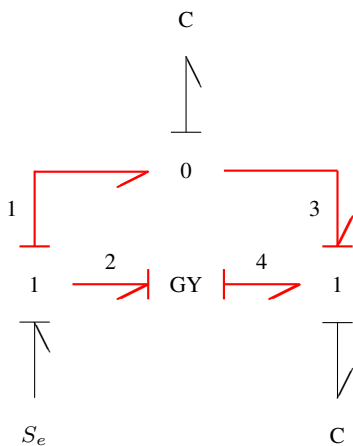


Fig. 3.18 Example of a causal mesh with bonds 1-2-3-4

- All energy stores have integral causality.
- There are no causal loops and no causal meshes.
- Causal paths between resistor ports do not share bonds.

More general cases are dealt with in Chapter 5. Once outputs of sources, energy stores and resistors are denoted in the bond graph, an ordered set of model equations can be manually derived from small and medium size bond graphs in a systematic

manner [21]. These equations may be coded in a simulation language [15] or in a modelling language directly from the bond graph.

3.5.1 Procedure for the Manual Derivation of Equations from a Causal Bond Graph

1. Write the constitutive equations for all independent sources. Their outputs are given functions of time.
2. In contrast, the output of a controlled source is algebraically related to its input. If the latter is not an output of an independent source or an energy store with integral causality, then it can be expressed by means of such outputs by back propagation of causal paths in the junction structure and by eliminating intermediate variables.
3. The outputs of resistors algebraically depend on their inputs. By back propagation along causal paths through the junction structure, their outputs can be expressed by sources either independent, or controlled ones and outputs of energy stores. The outputs of dependent sources do not need to be eliminated since they have already been determined in the previous step.

If there are causal paths between resistor ports, implicit and likely nonlinear algebraic equations will result.

4. For storage ports, the derivative with respect to time of an output is a function of the input(s). By working back causal paths, the inputs can be expressed by outputs of other energy stores, resistors, or sources.

3.5.2 Application of the Procedure to Some Examples

Bond Graph with Stores in Integral Causality, No Causal Paths between Resistive Ports, No Causal Loops and No Causal Meshes

For illustration of the procedure, equations will be derived from the causal bond graph of Figure 3.11 of the network in Figure 2.50. Application of the procedure yields the following ordered set of equations.

Independent Sources

$$U = f_1(t) \tag{3.7a}$$

$$I = f_2(t) \tag{3.7b}$$

Dissipators

$$u_R = R n i_L \quad (3.7c)$$

Energy Stores

$$\dot{u}_{C_1} = \frac{1}{C_1} [I - i_L] \quad (3.7d)$$

$$\dot{u}_{C_2} = \frac{1}{C_2} n i_L \quad (3.7e)$$

$$\frac{d}{dt} i_L = \frac{1}{L} [u_{C_1} - n(u_{C_2} + u_R - U)] \quad (3.7f)$$

The equations can be processed directly in this order. Moreover, after the elimination of the output of the resistor in the equations of the energy stores, the equations of the energy stores may be written in linear state space form

$$\underbrace{\frac{d}{dt} \begin{bmatrix} u_{C_1} \\ u_{C_2} \\ i_L \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 0 & -1/C_1 \\ 0 & 0 & n/C_2 \\ 1/L & -n/L & -n^2 R/L \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_{C_1} \\ u_{C_2} \\ i_L \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 & 1/C_1 \\ 0 & 0 \\ n/L & 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} U \\ I \end{bmatrix}}_{\mathbf{u}} \quad (3.8)$$

with the state vector $\mathbf{x} = [u_{C_1}, u_{C_2}, i_L]^T$ and the vector of system inputs $\mathbf{u} = [U, I]^T$. The superscript T denotes the transposition of vectors. In the case of non-linear resistor characteristics, the set of state equations takes the more general form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) . \quad (3.9)$$

Bond Graph with a Causal Path between Resistive Ports

In order to demonstrate the application of the procedure to a bond graph with a causal path between resistor ports, the I energy store in the bond graph of Figure 3.11 is replaced by a resistor (Figure 3.19).

From the modified bond graph portrayed in Figure 3.19, the following equations can be derived, eliminating intermediate variables by back tracking causal paths into the junction structure.

Sources

$$E = f_1(t) \quad (3.10a)$$

$$F = f_2(t) \quad (3.10b)$$

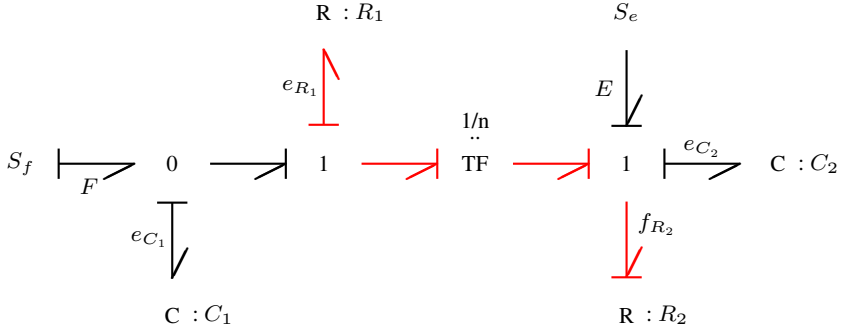


Fig. 3.19 Bond graph with a causal path between two resistor ports

Dissipators

$$e_{R_1} = R_1 n \frac{1}{R_2} [n(e_{C_1} - e_{R_1}) + E - e_{C_2}] \quad (3.10c)$$

$$f_{R_2} = \frac{1}{R_2} [n(e_{C_1} - R_1 n f_{R_2}) + E - e_{C_2}] \quad (3.10d)$$

Energy Stores

$$\dot{e}_{C_1} = \frac{1}{C_1} [F - n f_{R_2}] \quad (3.10e)$$

$$\dot{e}_{C_2} = \frac{1}{C_2} f_{R_2} \quad (3.10f)$$

The equations of the resistors preceding the equations of the energy stores are implicit, but not coupled. Thus, each of them may be solved independent from the other one and the set of equations can be solved exactly in that order. If the output of a resistor was expressed by means of the output of sources, energy stores and *the output of the other resistor*, the resulting equations for the resistors would be shorter because causal paths to the next output are shorter, but the equations would be coupled.

$$e_{R_1} = R_1 n f_{R_2} \quad (3.10g)$$

$$f_{R_2} = \frac{1}{R_2} [n(e_{C_1} - e_{R_1}) + E - e_{C_2}] \quad (3.10h)$$

If the equations of the resistors are linear, as in this example, they may be solved symbolically and the outputs of resistors may be eliminated in the equations of the energy stores. The result is a state space model (Equation 3.8 or Equation 3.9). Oth-

erwise, the set of equations remains in the form of a DAE system which results if certain types of causal paths appear in the bond graph. In the case of this simple example, the equations can be easily formulated in a simulation language or in a modelling language and processed by software supporting the language. More general cases of causal paths and the question of the differential index of the resulting DAE system are dealt with in the next chapter.

Bond Graph with a Controlled Source

Next, the procedure is applied to an example in which a source is controlled by a system input and a state variable. Consider the voltage follower depicted in Figure 3.20. Often, a voltage controlled voltage source is used as core functional model of an operational amplifier. Usually, in addition to a high amplification, $A_d < \infty$, the model accounts for an input resistance R_i of high value and an output resistance R_o of low value. The behavioural macro model (Figure 3.21) may be easily transformed into the bond graph depicted in Figure 3.22. The signals starting from the 0-junctions transmit the junction's effort as an *information* without affecting the power balance at these nodes. The power flows associated with the activated bonds are zero by definition. The summing node builds the voltage drop, u_d , across the input resistor that modulates the internal voltage source of the amplifier. First, we write the equation of the independent source.

Independent Sources

$$V_i = f_i(t), \quad (3.11a)$$

where f_i is a given function of time, t , that imposes a voltage V_i on the system.

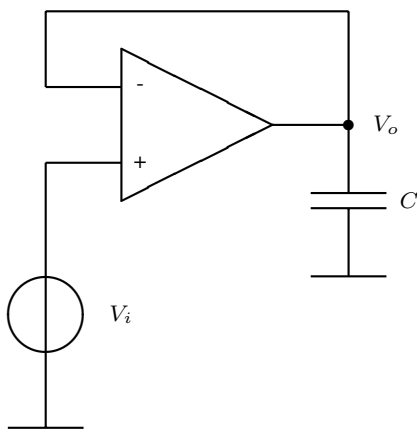


Fig. 3.20 Voltage follower

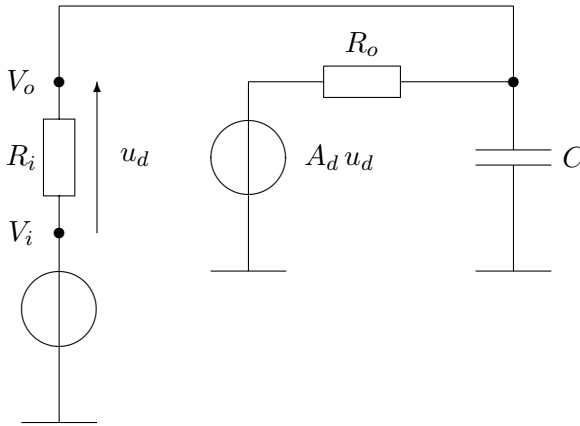


Fig. 3.21 Simple macro model of a voltage follower

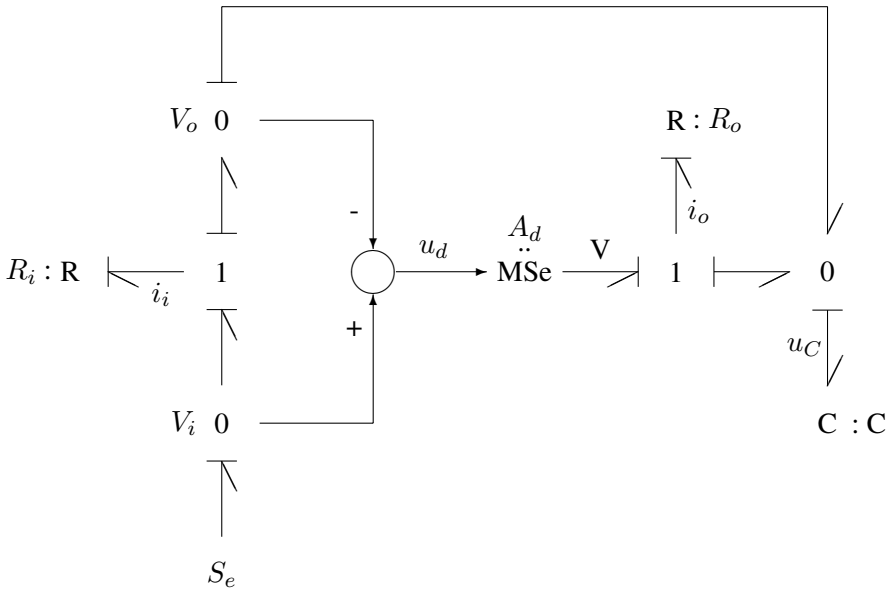


Fig. 3.22 Bond graph of the macro model of a voltage follower

From the bond graph of Figure 3.22, the following equations are derived.

Dependent Sources

$$V = A_d (V_i - u_c) \quad (3.11b)$$

Resistors

$$i_i = \frac{1}{R_i} (V_i - u_c) \quad (3.11c)$$

$$i_o = \frac{1}{R_o} [V - u_C] \quad (3.11d)$$

Energy Stores

$$\dot{u}_C = \frac{1}{C} (i_i + i_o) \quad (3.11e)$$

For high values of the amplification, $A_d \rightarrow \infty$, these equations reduce to $u_c = V_i$. That is, the voltage across the capacitors follows the input voltage.

In this example, the initial DAE system can be reduced to an explicit state space model. In general, if the signals modulating dependent sources, transformers, or gyrators can be expressed by system inputs or state variables, then the mathematical model can be written in state space form provided there are no causal paths between resistor ports, no causal conflicts at junctions, and all storage ports can be assigned integral causality [20].

Bond Graph of the Rolling Cylinder Example

Finally, we will derive the equations of motion for the rolling cylinder on an inclined plane (Figure 2.45) from the bond graph constructed step by step in Section 2.7. The simplified and causally completed bond graph is shown in Figure 3.23. In this simple example, the energy stores determine causalities at all other bonds.

From the bond graph, the following equations are derived.

Independent Sources

$$F = mg \sin \alpha \quad (3.12a)$$

Dissipators

$$F_R = b(v - r \omega) \quad (3.12b)$$

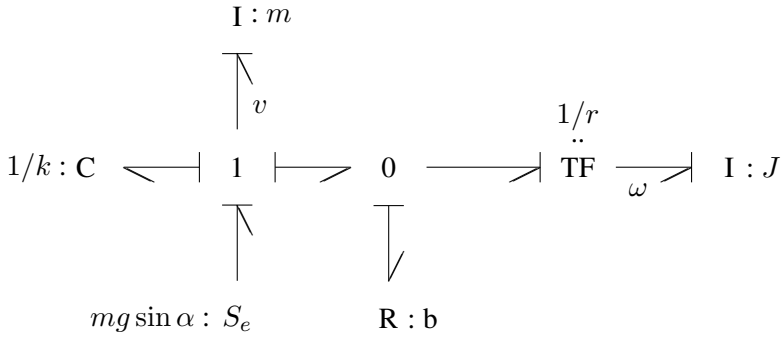


Fig. 3.23 Causally completed bond graph of the rolling cylinder on an inclined plane (Figure 2.45)

Energy Stores

$$\dot{v} = \frac{1}{m} [F - F_{sp} - F_R] \tag{3.12c}$$

$$\dot{F}_{sp} = k v \tag{3.12d}$$

$$\dot{\omega} = \frac{1}{J} r F_R \tag{3.12e}$$

These equations may be written in state space form.

$$\underbrace{\frac{d}{dt} \begin{bmatrix} v \\ F_{sp} \\ \omega \end{bmatrix}}_{\mathbf{\dot{x}}} = \underbrace{\begin{bmatrix} -b/m & -1/m & br/m \\ k & 0 & 0 \\ r b/J & 0 & -r^2 b/J \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} v \\ F_{sp} \\ \omega \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 1/m \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{[F]}_{\mathbf{u}} \tag{3.13}$$

The equations derived from the bond graph may also be rewritten into two Lagrange equations for the displacements, x, φ , of the inertias.

$$m\ddot{x} + b(\dot{x} - r\dot{\varphi}) + kx = mg \sin \alpha \tag{3.14a}$$

$$J\ddot{\varphi} + r m\dot{x} + r kx = r mg \sin \alpha \tag{3.14b}$$

The Lagrange equations can also be obtained in the traditional way by adding up the components in x-direction of all forces acting at the centre of gravity and by adding all moments at the contact point. Later in Section 4.10, we will see how Lagrange equations can be directly derived from a bond graph by using a procedure introduced by Karnopp.

3.6 Independent State Variables

In the previously considered examples, the differential equations derived from the bond graph can be transformed into a state space model (Equation 3.9) in which the number of independent state variables equals the number of energy stores with integral causality. By means of an example, we will show that the number of independent state variables may be smaller than the number of energy stores with integral causality.

Definition 3.12 (*Order of the model*). It is assumed that kinematic displacements are not needed to describe the dynamic behaviour of a system.

If integral causality has been assigned as preferred causality to the power ports of storage elements, then the order of the model is the number of power ports of energy stores with integral causality.

Remark 3.3. The order of the model apparently equals the number of independent initial conditions [20].

Energy stores with integral causality are important because their output or their energy variable may be chosen as a state variable. For energy stores with derivative causality, the energy variable algebraically depends on the energy variables of energy stores with integral causality and/or source outputs (system inputs). Therefore, they do not contribute to the system state. They are called *dependent state variables* and can be eliminated in the case of linear equations.

Each energy store in a bond graph contributes a differential equation determining a state variable. If kinematic displacements are needed, e.g., the bond graph contains transformers modulated by kinematic displacements, then in addition to the differential equations of all energy stores with integral causality, equations of the form $\dot{x} = v$ are needed for a complete description of the system state. (x denotes a kinematic displacement, while v is the flow of an I energy store.) Hence, in this case, the number of state variables is higher than the number of energy stores with integral causality. However, it may also be smaller.

Definition 3.13 (*Dimension of the state vector*). It is assumed that kinematic displacements are not needed to describe the dynamic behaviour of a system. Then, the dimension of the state vector equals the number of I and C ports.

Definition 3.14 (*Order of the set of differential equations*). The order of the set of differential equations is equal to the number of *independent* state variables ([20], p. 34).

Remark 3.4. A number of n independent state variables means that the set of first-order ODEs may be transformed into a single ODE of order n .

In the case of a set of linear first-order ODEs its order denotes the number of eigenvalues *distinct from zero*.

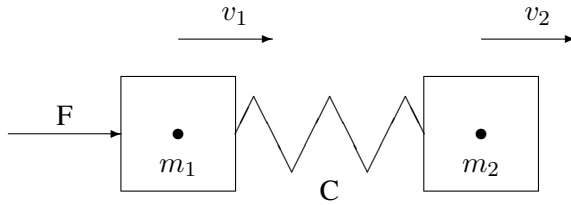


Fig. 3.24 Two oscillating masses interconnected by a spring

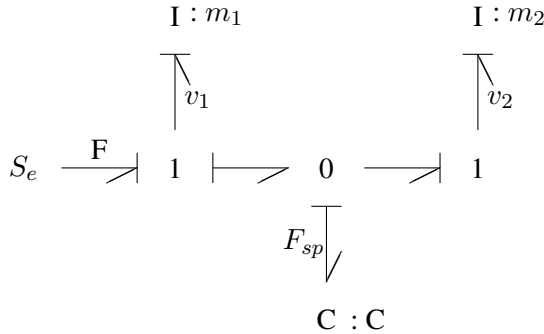


Fig. 3.25 Bond graph of two masses interconnected by a spring

Example: Two Oscillating Masses Connected by a Spring

Consider the simple example of two oscillating masses connected by a spring with an external force acting on mass m_1 as depicted in Figure 3.24 [13]. The corresponding bond graph in Figure 3.25 shows that the preferred integral causality can be assigned to all three energy stores. That is, their initial conditions can be chosen arbitrarily. Thus, the order of the model is three. Derivation of the equations shows that the *initial values* for the velocities $v_1(t)$, $v_2(t)$ can be independently chosen at times $t > 0$, however, the velocities are dependent. That is, the order of the system of equations is two.

The following equations can be derived from the bond graph.

$$\dot{v}_1 = \frac{1}{m_1} (F - F_{sp}) \tag{3.15a}$$

$$\dot{v}_2 = \frac{1}{m_2} F_{sp} \tag{3.15b}$$

$$\dot{F}_{sp} = \frac{1}{C} (v_1 - v_2) \tag{3.15c}$$

As there is no need for kinematic displacements as state variables, the dimension of the state vector is three. Equations 3.15a and 3.15b yield

$$m_1 v_1 + m_2 v_2 = \int_0^t F(\tau) d\tau + \text{const} . \quad (3.16)$$

The integration constant, *const*, is

$$p_0 := m_1 v_{10} + m_2 v_{20} = \text{const} . \quad (3.17)$$

That is, the initial values $v_{10} := v_1(t = 0)$, $v_{20} := v_2(t = 0)$ can be chosen arbitrarily. At times $t > 0$, the velocities are dependent.

If the Equations of motion 3.15a–3.15c are written as vector state equation, the dependency of state variables is also reflected by a *singular* system matrix \mathbf{A} of rank two with one vanishing eigenvalue and two complex eigenvalues (Equations 3.18 and 3.19).

$$\underbrace{\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{F}_{sp} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 0 & -\frac{1}{m_1} \\ 0 & 0 & \frac{1}{m_2} \\ \frac{1}{C} & -\frac{1}{C} & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ F_{sp} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 1 \\ m_1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} F \\ \mathbf{u} \end{bmatrix}}_{\mathbf{u}} \quad (3.18)$$

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \lambda \left[\lambda^2 + \frac{1}{C} \left(\frac{1}{m_1} + \frac{1}{m_2} \right) \right] = 0 \quad (3.19)$$

In the general case of a linear bond graph in which all energy stores have integral causality, the state space equations are of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \quad (3.20)$$

with the $n \times n$ system matrix \mathbf{A} , the input matrix \mathbf{B} , the vector of state variables \mathbf{x} and the vector \mathbf{u} of system inputs. The order of the model then equals the number, n , of rows (columns) of the state matrix \mathbf{A} , while the order of the system of differential equations is equal to the number, q , of eigenvalues of \mathbf{A} distinct from zero. In other words, the order of the model is equal to the degree of the denominator polynomial of all entries in the transfer matrix $\mathbf{H}(s) = [\mathcal{L}y_j / \mathcal{L}u_i]$, $s \in \mathbb{C}$, from any input u_i to any output y_j . The number of *independent* state variables is equal to this denominator degree minus the number, k , of integrators in a transfer function.

The number, q , of independent state variables is also equal to the rank of the system matrix. The characteristic polynomial, P , of the system matrix \mathbf{A} reads

$$P(s) := \det(s\mathbf{I} - \mathbf{A}) = s^n + \underbrace{(a_{11} + a_{22} + \dots + a_{nn})}_{\alpha_{n-1}} s^{n-1} + \dots + \underbrace{\det \mathbf{A}}_{\alpha_0} . \quad (3.21)$$

If a power of s , s^k , $k \geq 1$ can be factored out of this sum, that is, P takes the form

$$P(s) = s^k (s^{n-k} + \alpha_{n-1} s^{n-k-1} + \dots + \alpha_{k+1} s + \alpha_k) , \quad (3.22)$$

then k coefficients $\alpha_0 \dots \alpha_{k-1}$ must vanish. (The coefficients α_i , $i = 0, n - 1$ are functions of the coefficients of the system matrix.) For the system matrix \mathbf{A} , these constraints mean that k of its rows are linearly dependent of the $n - k$ other rows. Thus, $\text{rank} \mathbf{A} = n - k = q$.

In the example of two oscillating masses interconnected by a spring, the number of independent state variables is two. Thus, it is sufficient to choose the spring force F_{sp} and one of the two velocities, say v_1 , as state variables. If $p(t) := \int_0^t F(\tau) d\tau + p_0$ denotes the total momentum of both masses at time t , then

$$\dot{v}_1 = \frac{1}{m_1} (F - F_{sp}) \quad (3.23a)$$

$$\dot{F}_{sp} = \frac{1}{C} \left[\left(1 + \frac{m_1}{m_2} \right) v_1 - \frac{p(t)}{m_2} \right]. \quad (3.23b)$$

The other velocity, v_2 , is uniquely determined by Equation 3.16. Consequently, the two masses interconnected by a spring may be described by the following DAE system

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{F}_{sp} \\ \dot{v}_2 \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{m_1} & 0 \\ -\frac{1}{C} \left(1 + \frac{m_1}{m_2} \right) & 0 & 0 \\ m_1 & 0 & m_2 \end{bmatrix} \begin{bmatrix} v_1 \\ F_{sp} \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{m_1} F \\ -\frac{1}{C} \frac{p(t)}{m_2} \\ p(t) \end{bmatrix}. \quad (3.24)$$

In Equation 3.24, the matrix pre-multiplying the time derivative of the so-called *descriptor vector* $\mathbf{x} = [v_1, F_{sp}, v_2]^T$ includes a vanishing row for the dependent inertia m_2 . That is, the matrix is singular.

If Equation 3.15c for the spring force is differentiated with respect to time, then after elimination of the differentiated velocities, a second order ODE for the spring force is obtained

$$\ddot{F}_{sp} + \frac{1}{C} \frac{m_1 + m_2}{m_1 m_2} F_{sp} = \frac{1}{m_1} \ddot{F} \quad (3.25)$$

with the undamped natural frequency ω_0 given by Equation 3.27. That is, the order of the system of differential equations in this example is two.

Causalities in the bond graph of Figure 3.25 do not reflect that for times $t > 0$ the velocities are dependent. Thus, the number of storage ports with integral causality does not necessarily equal the number of *independent* state variables. Nevertheless, the number of independent state variables may be *directly* determined from a bond graph prior to the formulation of any equations. Again, we assume that no kinematic displacements are needed for a complete state description. In [14], Karnopp introduces a modification of the standard Sequential Causality Assignment Procedure (SCAP). In this modified procedure, all energy stores obtain *derivative* causality as the *preferred* causality. As a result of propagation of this preferred causality, it may happen that a storage port must accept integral causality. Again, this is an indication that the energy variables of some energy stores algebraically depend on those of other energy stores. If this so-called *all derivative* approach is applied to the exam-

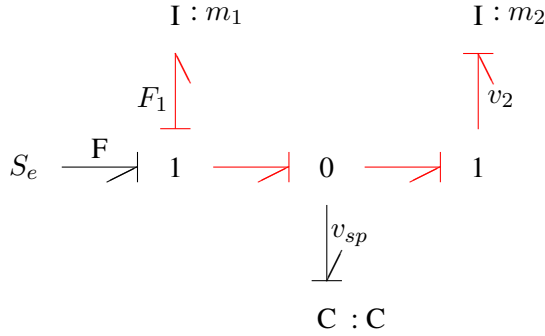


Fig. 3.26 Bond graph with derivative causalities as preferred causalities

ple of the two interconnected masses, it turns out that one I energy store must have integral causality (Figure 3.26). Hence, their co-energy variables, the velocities v_1 , v_2 , cannot be independent. The unmodified SCAP using integral causality as the preferred causality does not reveal this result.

If the outputs of the energy stores with preferred causality are chosen as state variable, the following equations are obtained from the bond graph in Figure 3.26.

$$\begin{bmatrix} F_1 \\ v_{sp} \end{bmatrix} = \begin{bmatrix} 0 & \frac{m_1 m_2}{m_1 + m_2} \\ -C & 0 \end{bmatrix} \begin{bmatrix} \dot{F}_1 \\ \dot{v}_{sp} \end{bmatrix} + \begin{bmatrix} \frac{m_1}{m_1 + m_2} \\ 0 \end{bmatrix} [F] + \begin{bmatrix} 0 \\ C \end{bmatrix} [\dot{F}] . \tag{3.26}$$

In Equation 3.26, the 2×2 matrix pre-multiplying the vector of rates, $[\dot{F}_1, \dot{v}_{sp}]^T$, is non singular. The eigenvalues, λ , of its inverse are given by the characteristic equation $\lambda^2 + \omega_0^2 = 0$ with

$$\omega_0^2 := \frac{1}{C} \frac{m_1 + m_2}{m_1 m_2} . \tag{3.27}$$

3.7 Determination of the Number of Independent State Variables Directly from the Bond Graph

The number of independent state variables may be determined from the bond graph in the following manner [20]. First, the bond graph is causally completed using integral causality as the preferred causality by means of the unmodified SCAP. In a second step, causality assignment is repeated by using derivative causality as the preferred causality (all derivative mode of the SCAP). As a result, the number of *independent* state variables equals the number of storage ports that have the *preferred causality in both modes*. In other words, if n denotes the order of the model and k

the number of energy stores that retain integral causality, when preferred derivative causality is assigned, then the number of independent state variables, q , equals $n - k$. This difference is also called the *bond graph rank* ($BG - \text{rank} \mathbf{A}$) of a $n \times n$ state matrix \mathbf{A} ([5], Chapter 8) and is equal to its rank if there are no numerical perturbations in the entries of the state matrix and no numerical errors in the computation of its rank. (It is also equal to the number of non-zero eigenvalues of the state matrix.) The number of energy stores that retain integral causality when preferred derivative causality is assigned, k , is called the number of (structural) *null modes*. If a transfer function is set up, k is equal to the number of integrators $1/s$ in the transfer function.

Assuming that kinematic displacements are not needed to describe the dynamic behaviour of a linear time-invariant system, then these results can be summarised in the following manner.

Proposition 3.1 (*Rank of the state matrix*). The rank of the state matrix equals the number of I and C ports in derivative causality if derivative causality is chosen as the preferred causality.

Proposition 3.2 (*Number of null eigenvalues of the state matrix*). The number of null eigenvalues of the state matrix equals the number of null poles in the transfer functions and is equal to the number of I and C ports that must accept integral causality when derivative causality is the preferred causality.

In the example under consideration, the number of storage ports that retain integral causality when derivative causality is the preferred one is equal to one. Notice that the all derivative mode of the SCAP is only used to compare the number of storage ports with preferred causality in both modes and to determine the number of independent state variables. The example shows that instead of the unmodified SCAP, also the SCAP in all derivative mode could be used. The equations, derived from the bond graph causally completed this way generally have the form of a DAE system. In the case of the example, it can be transformed into an ODE since the matrix pre-multiplying the vector of rates is non-singular.

Remark 3.5. The example of two oscillating masses interconnected by a spring is a well known example of a so-called *semi-definite* or *degenerated* vibratory system [18]. The system has a non-vibratory rigid body motion corresponding to the zero eigenvalue and a vibratory mode, viz. the relative motion between the two bodies corresponding to the pair of conjugate complex eigenvalues.

Example: RC Network

For illustration of the determination of the number of independent state variables from a bond graph, let us consider another example. Figure 3.28 displays the bond graph of the simple RC network of Figure 3.27.

Since all three C storage elements take the preferred integral causality, the order of the model is three. However, if derivative causality is the preferred causality, then only one of the three energy stores can accept derivative causality (cf. Figure 3.29).

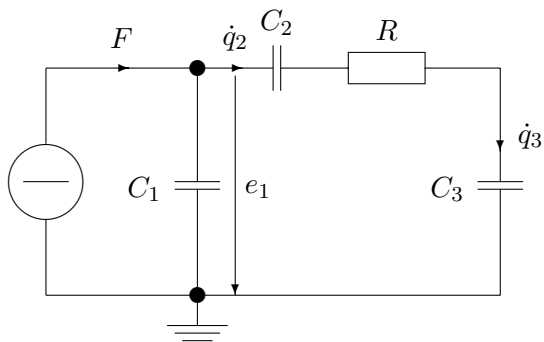


Fig. 3.27 Circuit schematic of a simple RC network

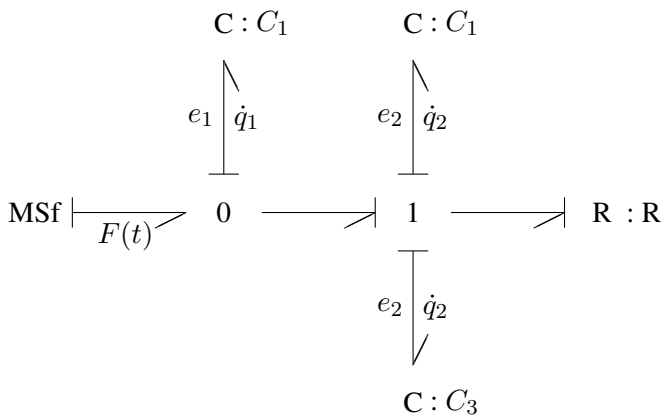


Fig. 3.28 Bond graph of the RC network in Figure 3.27

Consequently, the bond graph rank of the system matrix equals one, $(BG - \text{rank})\mathbf{A} = 1$. As a result, there should be one eigenvalue distinct from zero, and two null modes and the number of independent state variables is one. This will be verified. From the bond graph with preferred integral causality, the following state equations can be derived.

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{C_1 R} & \frac{1}{C_1 R} & \frac{1}{C_1 R} \\ \frac{1}{C_2 R} & -\frac{1}{C_2 R} & -\frac{1}{C_2 R} \\ \frac{1}{C_3 R} & -\frac{1}{C_3 R} & -\frac{1}{C_3 R} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} \\ 0 \\ 0 \end{bmatrix} [F] \quad (3.28)$$

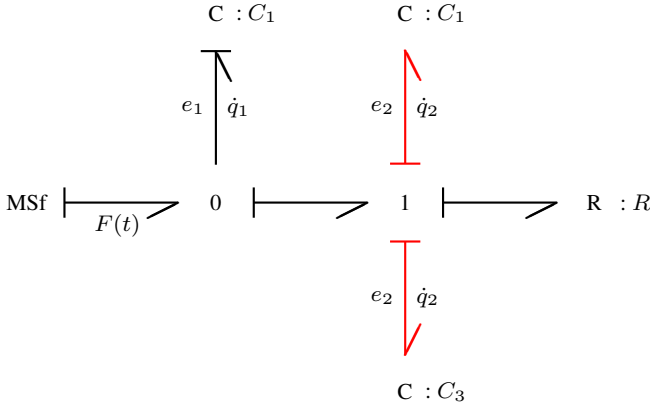


Fig. 3.29 Bond graph of the RC network with *preferred derivative* causality

Actually,

$$\det(s\mathbf{I} - \mathbf{A}) = s^2 \left[s^1 + \left(\frac{1}{C_1 R} + \frac{1}{C_2 R} + \frac{1}{C_3 R} \right) \right]. \quad (3.29)$$

That is, the number of null modes equals two ($k = 2$) and the number of independent state variables is one ($q = 1$). As it is easily checked on \mathbf{A} , $\text{rank}\mathbf{A} = 1$. It is not surprising that there is only one state variable. The following equations can be immediately derived from the circuit schematic of Figure 3.27.

$$\dot{q}_2 = F - C_1 \dot{e}_1 \quad (3.30a)$$

$$C_2 \dot{e}_2 = \dot{q}_2 = \dot{q}_3 = C_3 \dot{e}_3 \quad (3.30b)$$

$$e_R = R(F - C_1 \dot{e}_1) \quad (3.30c)$$

Hence, e_1 as a state variable and $F(t)$ as input are sufficient to determine all other variables.

3.8 Conclusion

In an initial phase of bond graph based physical modelling, an ideal physical model is developed by making assumptions and by deciding which physical effects are to be taken *qualitatively* into account. The next step towards a *quantitative* analysis is the systematic development of a mathematical model to be derived from a bond graph. For that purpose, the founders of bond graph methodology introduced the concept of *computational causality*. For each power port, one decides which of the two power variables is an independent variable in the constitutive equations of the

multiport under consideration. Simultaneously, the conjugate variable must be provided from that power port adjacent at the other end of the bond connecting both ports. This decision is expressed by a perpendicular stroke attached to one end of the bond. Thus, the physical structure displayed by the acausal bond graph is superimposed by a *computational structure*. Causality can be assigned to power ports without knowing actual constitutive equations. Only a certain type of constitutive equations is assumed. If the constitutive equations for all bond graph nodes are known, then the causally completed bond graph corresponds to a block diagram and could be transformed systematically into a block diagram. However, since results obtained from a block diagram can also be directly derived from the bond graph itself, such a transformation does not lead to further results from a methodological point view. Regarding the derivation of transfer functions or the simulation of the dynamic system behaviour, there are several programs available that accept a combination of bond graphs and block diagrams.

Concerning mathematical models to be derived from a bond graph, an essential feature of bond graphs is that after causalities have been assigned, statements can be made about the form of the mathematical model, the model order, the number of independent state variables without formulating and rearranging any equations. It is not even necessary to know actual functional relations. It is sufficient to know which types of variables are related, e.g., whether it is a relation between an effort and the integral of the flow or between the effort and the flow at a power port. In general, a nonlinear relation will be assumed. Relations between causal patterns in a bond graph and features of mathematical models will be considered in detail in the next chapter. In this chapter, rules for causality assignment are discussed that obviously result from the properties of the elements. Moreover, for the systematic assignment and the propagation of causalities, the Sequential Causality Assignment Procedure, SCAP, introduced by Karnopp and Rosenberg, has become a standard procedure in bond graph literature.

As state variables, either the energy or the co-energy variables of energy stores with integral causality may be chosen. In this book, co-energy variables are used. They are power variables and outputs of the energy stores. Both the energy and co-energy variable of a 1-port energy store are related by its characteristic assumed to be one-to-one and having a unique inverse. In addition to these state variables, kinematic displacements may be needed for a complete state description, especially for the planar or 3D motion of mechanical systems. Energy stores with derivative causality do not contribute to the state of a system. Their energy variable depends on the energy variables of other energy stores with integral causality. Having introduced the notion of a causal path, a procedure is given for the manual derivation of equations from bond graphs of small and medium size. The procedure assumes that there are no energy stores with derivative causality, no causal loops, no causal meshes and that causal paths between resistor ports do not share bonds. Application of the procedure has been illustrated by some simple examples.

Finally, it has been shown how the number of independent state variables, the order of the system of differential equations, can be determined from the bond graph. To that end, causality assignment is repeated by using derivative causality as the

preferred causality. The number of storage ports having the preferred causality in both modes of the SCAP equals the number of independent state variables. It is the smallest number of state variables needed for the description of the system state provided no kinematic displacements have to be added.

References

- [1] A.A. Bell and H.R. Martens. A comparison of linear graphs and bond graphs in the modeling process. *Proc. Joint Automation Control Conference*, pages 777–794, 1974.
- [2] A. Biran and M. Breiner. *MATLAB for Engineers*. Addison Wesley, 1999. URL <http://www.Mathworks.com>.
- [3] *CAMP-G – User’s Manual*. Cadsim Engineering, P. O. Box 4083, Davis, Ca 95617. URL <http://www.bondgraph.com>.
- [4] Controllab Products. 20-sim the power in modeling. URL <http://www.20sim.com>.
- [5] G. Dauphin-Tanguy. *Les bond graphs*. Hermes Science Europe Ltd., Paris, France, 2000. ISBN: 2-7462-0158-5.
- [6] A. Donaire and S. Junco. Derivation of input-state-output port-hamiltonian systems from bond graphs. *Simulation Modelling Practice and Theory*, 17(1):137–151, 2009.
- [7] Dynasim. URL <http://www.Dynasim.se>.
- [8] H. Elmqvist. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1978. Report CODEN: LUTFD2/(TFRT-1015)/1-226/(1978).
- [9] O. Föllinger. *Regelungstechnik – Einführung in die Methoden und ihre Anwendung*. Hüthig GmbH, Heidelberg, 1994. ISBN: 3-7785-2336-8.
- [10] P.J. Gawthrop. MTT: Model Transformation Tools. In F.E. Cellier and J.J. Granda, editors, *ICBGM’95, International Conference on Bond Graph Modeling and Simulation*, volume 27(1) of *Simulation Series*, pages 197–202. SCS Publishing, 1995.
- [11] P.J. Gawthrop and L. Smith. *Metamodeling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [12] P.J. Gawthrop and L. Smith. Causal Augmentation of Bond Graphs with Algebraic Loops. *Journal of the Franklin Institute*, 329:291–303, 1992.
- [13] D.C. Karnopp. On the order of a physical system model. *Journal of Dynamic Systems, Measurement, and Control*, 101:185–186, 1979.
- [14] D.C. Karnopp. Alternative Bond Graph Causal Patterns and Equation Formulations for Dynamic Systems. *Journal of Dynamic Systems, Measurement, and Control*, 105:58–63, 1983.
- [15] D.C. Karnopp. Direct Programming of Continuous System Simulation Languages Using Bond Graph Causality. *Transactions of the Society for Computer Simulation*, 1(1):49–60, 1984.
- [16] D.C. Karnopp and R.C. Rosenberg. *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics*. MIT Press, Cambridge, MA, 1968.
- [17] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1990.
- [18] S.S. Rao. *Mechanical Vibrations*. Addison-Wesley, 1995.
- [19] J.C. Strauss, D.C. Augustin, M.S. Fineberg, B.B. Johnson, R.N. Linebarger, and F.H. Sanson. The SCi Continuous System Simulation Language (CSSL). *SIMULATION*, pages 281–303, Dec. 1967.
- [20] J. van Dijk. *On the role of bond graph causality in modelling mechatronic systems*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1994.
- [21] P.E. Wellstead. *Introduction to Physical System Modelling*. Academic Press, London, 1979.

Chapter 4

Causal Bond Graphs and Forms of Mathematical Models

4.1 Causal Paths Between Resistive Ports

After the discussion of the systematic construction of bond graphs, their causal augmentation and the systematic derivation of equations from a causal bond graph, we are prepared to consider different causal patterns in bond graphs and their relation with different forms of mathematical models in detail.

As a general prerequisite, we assume that if controlled sources appear in a bond graph, their modulating signal can be expressed by system inputs or by state variables. Furthermore, if transformers and gyrators are modulated, they are allowed to be modulated only by state variables. The reason for this confinement is that otherwise, algebraic loops may result that are not easily detected by inspection of a causal bond graph as will be explained in more detail in Section 4.8.

In Chapter 5, approaches to the symbolic and numerical solution of mathematical models derived from bond graphs are considered in detail [4, 5].

It has already been pointed out in the previous chapter that a causally completed bond graph gives indication to the form of a mathematical model prior to any equations formulation. In that context, the notion of a *causal path* plays an important role (Definition 3.5). The simplest case of bond graphs has already been dealt with. That is, all storage ports have preferred integral causality. There are neither causal paths between resistive ports, nor causal loops, nor causal meshes in the junction structure. In this case, the equations derived from the bond graph can be written in state space form.

In the following, relaxations of these conditions will be considered. First, causal paths between resistive ports are allowed as they result in bond graphs, e.g., of electrical or hydraulic Wheatstone bridges. Figure 4.1 shows a hydraulic bridge circuit with variable area orifices and a symmetric double acting cylinder in the load diagonal. The piston has a cross sectional area A and a mass m . The transformation of the circuit schematic into a bond graph is straightforward. Choosing the return pressure of the reservoir, p_T , as a reference results in the bond graph depicted in Figure 4.2. The annotations $g_i(\cdot)$ of the resistor symbols denote a nonlinear function

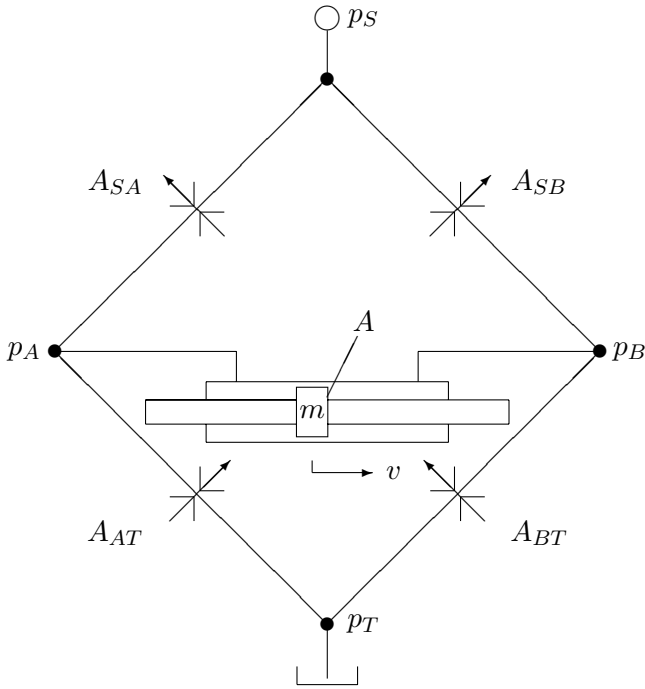


Fig. 4.1 Hydraulic Wheatstone bridge with a symmetric double acting cylinder in the load diagonal

relating the pressure drop across the orifice to the volume flow through the orifice according to Bernoulli’s law. Since the example is a hydraulic circuit, power variables are denoted by the symbol ‘p’ (pressure) and ‘Q’ (volume flow), as common in hydraulics. The bond graph of Figure 4.2 shows two disjoint causal paths between the resistive ports with bonds enumerated 1 – 2 – 3 and 4 – 5 – 6.

If the outputs of the resistors R_i are denoted p_i or Q_i depending on their causality, then the following set of equations can be derived from the bond graph of Figure 4.2.

$$p_S = f_S(t) \tag{4.1a}$$

$$p_1 = g_1^{-1} (Q_2 - A v) \tag{4.1b}$$

$$Q_2 = g_2 (p_S - p_1) \tag{4.1c}$$

$$Q_3 = g_3 (p_S - p_4) \tag{4.1d}$$

$$p_4 = g_4^{-1} (Q_3 + A v) \tag{4.1e}$$

$$\dot{v} = \frac{1}{m} [(p_S - p_4) - (p_S - p_1)] \tag{4.1f}$$

The system of algebraic equations for the outputs of the four resistors is divided into two separate subsystems with two coupled equations because the two causal paths

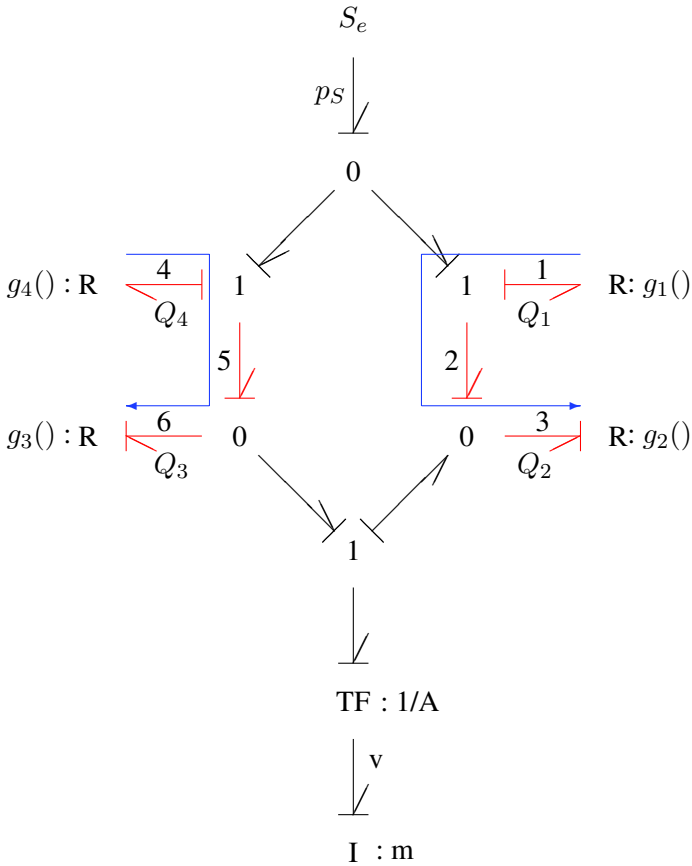


Fig. 4.2 Bond graph of the hydraulic bridge with two disjoint causal paths between resistive ports

are disjoint. Two of the four algebraic unknowns belong to one subsystem, while the other two belong to the other. Variables p_1 and Q_2 both belong to the causal path with bonds labelled 1 – 2 – 3. The corresponding Equations 4.1b and 4.1c may be graphically represented as a signal flow loop along the causal path 1 – 2 – 3. Consider the signal flow loop along the causal path 1 – 2 – 3. The volume flow Q_2 enters into resistor 1. The pressure p_1 leaving that resistor is an input into resistor 2, while the volume flow Q_2 is an output of resistor 2. That is, both power variables of each bond of the causal path are involved in the signal flow loop. This gives rise to the following definitions.

Definition 4.1 (Topological loop). A topological loop is a signal flow loop along a causal path or a causal loop. The causal path must not begin or end at an ideal source.

Remark 4.1. A causal path that begins or ends at an ideal source is not associated with a topological loop because the power port variables of an ideal source or sink are not related. In other words, an ideal source breaks a topological loop.

Definition 4.2 (*Topological path*). A topological path is a part of a topological loop. It is a signal flow graph fragment that represents bond variables and constitutive relations being part of a causal path.

Definition 4.3 (*Algebraic loop*). If the variables of a topological loop depend *algebraically* on themselves, that is no integration with respect to time is involved, then the topological loop is called an *algebraic loop*.

Definition 4.4 (*Order of a topological loop*). The order of a topological loop denotes the number of remaining integrators involved in the causal path. If there is no remaining integration in the causal path, then the topological loop is called a zero-order loop.

Definition 4.5 (*Flat loop*). A topological loop is called a *flat loop* if both opposite signals of each bond being part of the causal path or causal loop are involved in the signal flow loop.

Remark 4.2. A flat loop passes each bond of the causal path or causal loop twice.

Figure 4.3 depicts a flat loop related with the causal path $1 - 2 - 3$.

Definition 4.6 (*Open loop*). A topological loop that uses only one of the opposite signals of some or all bonds in a causal path is called an *open loop*.

Remark 4.3. Prerequisite for an open loop is a bond loop. (Definition 3.7).

The topological loop in Figure 4.4 only uses one of the two power variables of the bonds involved in the bond loop. Hence, according to the above definition, it is an open loop. There is a second open loop not depicted in the graph that runs in parallel to the first, but is oppositely oriented. The places where the signal path that leaves

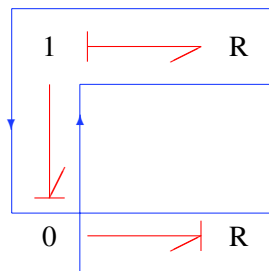


Fig. 4.3 Flat loop related with a causal path

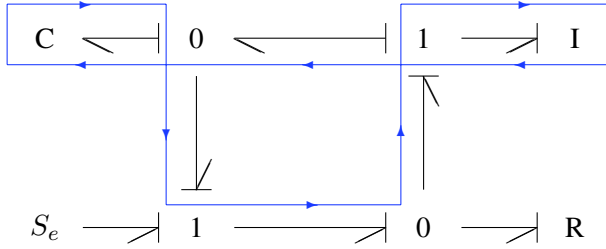


Fig. 4.4 Example of an open loop with mesh stubs

the bond loop reverses in a 1-port passive element and returns into the bond loop are called *mesh stubs* [10].

In [10], Brown observes:

Most loops in passive systems are flat, and in fact a tree-like bond graph (with no meshes) cannot have other than flat loops.

The topological loops in the bond graph of the example are parts of a signal flow graph that can be assigned to the bond graph. This link between bond graphs and signal flow graphs will be considered later in Chapter 6 in more detail. In this chapter, it sufficient to note that a causal path between two resistive ports indicates an algebraic loop.

In the example under consideration, the causal paths do not touch. Thus, for each of the two causal paths, we can substitute one of the algebraic equations into the other one. As a result, we obtain two implicit algebraic equations that can be separately solved. With the solutions of the two equations the other unknown can be directly computed. In Section 5.4, we will consider the case in which causal paths share bonds and introduce an approach leading to a small set of coupled equations. Once its solution is known, all other algebraic unknowns can be directly computed.

In [32], van Dijk introduces several classes of causal paths depending on the type of ports they connect or whether they are closed. He calls causal paths between resistive ports *class-2 zero-order causal paths*. The prefix *zero-order* emphasises that the variables of the causal path are algebraically coupled.

If the outputs of the resistors in the bond graph of Figure 4.2 are combined into an auxiliary vector $\mathbf{h} = (p_1, Q_2, p_4, Q_3)^T$, then the equations of the example may be written in the general form

$$\mathbf{h}(t) = \mathbf{f}_1(\mathbf{x}(t), \mathbf{h}(t), \mathbf{u}(t)) \tag{4.2a}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}_2(\mathbf{x}(t), \mathbf{h}(t), \mathbf{u}(t)) \tag{4.2b}$$

with $\mathbf{x} = [v]$ and $\mathbf{u} = [p_S]$.

The causal paths in the bond graph of Figure 4.2 indicate that the standard Sequential Causality Assignment Procedure (SCAP) requires an inversion of the characteristic of two of the four resistors, which is possible in this case. In the case of

non-invertible characteristics, the causalities at two resistive ports would not agree with the form of the constitutive equations. We could account for non-invertible resistor characteristics by writing Equations 4.1b and 4.1e in the form

$$g_1(p_1) = Q_2 - Av \quad (4.3a)$$

$$g_4(p_4) = Q_3 + Av, \quad (4.3b)$$

leading to a semi-explicit nonlinear DAE system of the general form

$$\mathbf{0} = \tilde{\mathbf{f}}_1(\mathbf{x}(t), \mathbf{h}(t), \mathbf{u}(t)) \quad (4.4a)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}_2(\mathbf{x}(t), \mathbf{h}(t), \mathbf{u}(t)). \quad (4.4b)$$

We will consider another causality assignment procedure in addition to the standard Sequential Causality Assignment Procedure (SCAP) used so far. It is the method of *relaxed* causalities as introduced by Joseph and Martens (cf. Section 4.9).

4.2 Some Fundamentals from the Theory of Differential-Algebraic Systems

Figure 4.2 of the previous example illustrates that the mathematical model derived from a bond graph takes the form of a DAE system if there are causal paths between resistive ports. An essential characteristic of DAE systems is their so-called *index*. To put it simply, it is an indication of how far away a DAE system is from an ODE system. An early 1982 article by Petzold [27] considering some of the difficulties that can occur with the numerical solution of DAE systems is titled:

Differential/Algebraic Equations are not ODEs

A general experience is that the higher the index is, the more difficulties are to be expected with the numerical solution of the DAE system (see Section 5.1 and Section 5.5).

Regarding the index of a DAE system, some definitions are required. By providing them, we will follow the presentation in the fundamental book of Brenan, Campbell and Petzold [8]. In the very first sentence of the preface, the authors point out that:

Differential-algebraic equations (DAE's) arise naturally in many applications, but present numerical and analytical difficulties which do not occur with ordinary differential equations.

Some books on DAE systems are, e.g. [16–18].

In the following, we assume a linear differential-algebraic set of equations of the form

$$\mathbf{A} \dot{\mathbf{x}}(t) + \mathbf{B} \mathbf{x}(t) = \mathbf{f}(t). \quad (4.5)$$

In Equation 4.5, \mathbf{A} , \mathbf{B} are $n \times n$ matrices with constant coefficients, $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$ is a vector function of the system inputs and $t \in [0, \infty)$.

Definition 4.7 (*Matrix pencil*). Let $\lambda \in \mathbb{C}$, then $\lambda\mathbf{A} + \mathbf{B}$ is called a *matrix pencil*.

If the determinant $\det(\lambda\mathbf{A} + \mathbf{B})$ is not identically zero as a function of λ , then the matrix pencil is called *regular*.

Remark 4.4. The definition of a regular matrix pencil is important because it is a necessary and sufficient condition for the solvability of a linear constant coefficient DAE (Equation 4.5). ([8], Theorem 2.3.1)

Definition 4.8 (*Index of a matrix*). A quadratic matrix \mathbf{M} is called *nilpotent* if there is positive integer k such that $\mathbf{M}^k = \mathbf{0}$.

If \mathbf{M} is a nilpotent matrix, then the smallest positive integer ν for which $\mathbf{M}^\nu = \mathbf{0}$ and $\mathbf{M}^{\nu-1} \neq \mathbf{0}$ is called the *index of nilpotency*.

The following theorem precedes the definition of the index of a linear constant coefficient DAE.

Theorem 4.1 (Kronecker). *Let $\lambda\mathbf{A} + \mathbf{B}$ be a regular matrix pencil. Then, there exist non-singular matrices \mathbf{P} and \mathbf{Q} such that*

$$\mathbf{PAQ} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \quad \mathbf{PBQ} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

where \mathbf{I} is an identity matrix and \mathbf{N} a nilpotent matrix of index k .

Definition 4.9 (*Index of a linear coefficient DAE*). Let $\lambda\mathbf{A} + \mathbf{B}$ be a regular matrix pencil, then the index of nilpotency, or index for short, of the linear constant coefficient DAE (4.5) is the index of nilpotency, k , of the matrix \mathbf{N} defined in Theorem 4.1.

If $\mathbf{N} = \mathbf{0}$, then define $k = 1$. In the case of a non-singular matrix \mathbf{A} , the index is defined as $k = 0$.

An important consequence of Kronecker's theorem is that the solution of the DAE system 4.5 can be given in analytical form. Suppose that the matrices \mathbf{P} and \mathbf{Q} in Theorem (4.1) are known. Then, by means of the transformation

$$\mathbf{x} = \mathbf{Q} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$$

and by scaling of Equation 4.5 using the matrix \mathbf{P} , the DAE system can be split into two uncoupled subsystems for which the analytical solution is known. With this transformation and the definition

$$\mathbf{Pf}(t) =: \begin{bmatrix} \mathbf{g}_1(t) \\ \mathbf{g}_2(t) \end{bmatrix},$$

an explicit first order ODE for \mathbf{y}_1 and a singular subsystem in canonical form for the unknown \mathbf{y}_2 results.

$$\dot{\mathbf{y}}_1 + \mathbf{C}\mathbf{y}_1 = \mathbf{g}_1 \quad (4.6a)$$

$$\mathbf{N}\dot{\mathbf{y}}_2 + \mathbf{y}_2 = \mathbf{g}_2 \quad (4.6b)$$

Let \mathcal{L}_- denote the Laplace operator

$$(\mathcal{L}_-x)(s) := \int_{0-}^{\infty} x(\tau)e^{-s\tau} d\tau, \quad (4.7)$$

where $s \in \mathbb{C}$ ([21], Section 1.2), then the Laplace transform of Equation 4.6b yields

$$\begin{aligned} (\mathcal{L}_-\mathbf{y}_2)(s) &= (\mathbf{N}s + \mathbf{I})^{-1}\mathbf{N}\mathbf{y}_2(0-) + (\mathbf{N}s + \mathbf{I})^{-1}(\mathcal{L}_-\mathbf{g}_2)(s) \\ &= \sum_{i=0}^{\infty} (-1)^i (\mathbf{N}s)^i \mathbf{N}\mathbf{y}_2(0-) + \\ &\quad \sum_{i=0}^{\infty} (-1)^i (\mathbf{N}s)^i (\mathcal{L}_-\mathbf{g}_2)(s). \end{aligned} \quad (4.8)$$

Since \mathbf{N} is a nilpotent matrix of index k , the infinite series reduces to a sum with a finite number of terms. Transformation back into the time domain gives $\mathbf{y}_2(t)$.

$$\mathbf{y}_2(t) = \sum_{i=0}^{k-1} (-1)^i \mathbf{N}^i \delta^{(i)} \mathbf{N}\mathbf{y}_2(0-) + \sum_{i=0}^{k-1} (-1)^i \mathbf{N}^i \mathbf{g}_2^{(i)}(t), \quad (4.9)$$

where $\delta^{(i)}$ denotes the i^{th} derivative of the Dirac pulse. Apparently, the Dirac pulse and its derivatives vanish and with it a source for big errors in the numerical solution due to a limited machine precision if values $\mathbf{y}_2(0-)$ vanish. If the index of the DAE system is > 1 , then the solution of the subsystem (4.6b) includes the derivatives of the function \mathbf{g}_2 up to the order $k - 1$. Their computation, if numerically performed, is ill conditioned. This problem does not appear if the matrix \mathbf{A} in Equation 4.5 is non-singular. Then, Equation 4.5 is not truly a DAE but an ODE. The solution of the explicit ODE, Equation 4.6a, is

$$\mathbf{y}_1 = \mathbf{e}^{-\mathbf{C}t} \mathbf{y}_1(0) + \int_0^t \mathbf{e}^{-\mathbf{C}(t-\tau)} \mathbf{g}_1(\tau) d\tau \quad (4.10)$$

for every initial value $\mathbf{y}_1(0)$.

Definition 4.10 (*Local index of a linear time-variant DAE*). If the coefficient matrices in Equation 4.5 are time dependent, then the *local* index is the index of the DAE system for some time t . It is denoted by $k(t)$.

For general nonlinear implicit differential-algebraic systems

$$\mathbf{F}(\dot{\mathbf{y}}, \mathbf{y}, t) = \mathbf{0}, \quad (4.11)$$

a so-called *differential* index has been introduced by Gear [14].

Definition 4.11 (*Differential index of a nonlinear DAE system*). If the matrix $\partial\mathbf{F}/\partial\dot{\mathbf{y}}$ is non-singular, the index k is set to zero, $k = 0$. (In this case Equation 4.11 is an implicit ODE that, in principle, can be transformed into an explicit ODE.)

Otherwise, the following set of equations is established by repeated differentiation of Equation 4.11 with respect to time

$$\begin{aligned} \mathbf{0} &= \mathbf{F}(\dot{\mathbf{y}}, \mathbf{y}, t) \\ \mathbf{0} &= \frac{d}{dt}\mathbf{F} = \frac{\partial\mathbf{F}}{\partial\dot{\mathbf{y}}}\ddot{\mathbf{y}} + \frac{\partial\mathbf{F}}{\partial\mathbf{y}}\dot{\mathbf{y}} + \frac{\partial\mathbf{F}}{\partial t} \\ \mathbf{0} &= \frac{d^2}{dt^2}\mathbf{F} = \frac{\partial\mathbf{F}}{\partial\dot{\mathbf{y}}}\mathbf{y}^{(3)} + \dots \\ &\vdots \\ \mathbf{0} &= \frac{d^j}{dt^j}\mathbf{F} = \frac{\partial\mathbf{F}}{\partial\dot{\mathbf{y}}}\mathbf{y}^{(j+1)} + \dots, \end{aligned}$$

in which $\dot{\mathbf{y}}, \dots, \mathbf{y}^{(j+1)}$ are considered separate independent algebraic variables being functions of the variables \mathbf{y} and t (considered independent). Since $\partial\mathbf{F}/\partial\dot{\mathbf{y}}$ is singular, it is not possible to solve for the highest derivative $\mathbf{y}^{(j+1)}$. However, if it is possible to solve for $\dot{\mathbf{y}}$ for some finite j , then the smallest j , for which this is possible, is defined the differential index k of the differential-algebraic system, Equation 4.11.

Remark 4.5. 1. Repeated differentiation of the initial DAE provides additional equations. If the resulting set of equations is solvable for $\dot{\mathbf{y}}$, the problem of solving the initial DAE can be transformed into the problem of solving an explicit ODE. In the above scheme, all equations of (4.11) are differentiated with respect to time, although it may be sufficient to differentiate only some of them in order to determine $\dot{\mathbf{y}}$ as a continuous function of \mathbf{y} and t . In [14], Gear gives an algorithm by which only equations are differentiated that do not include components of $\dot{\mathbf{y}}$.
2. For linear constant coefficient DAEs (Equation 4.5), the differential index equals the index of nilpotency introduced in Definition 4.9.

If the state vector \mathbf{x} and the vector of algebraic variables \mathbf{h} in the hydraulic bridge example are combined into a vector \mathbf{y} , then one step of differentiation yields that the DAE system is of index one if the partial derivative of the algebraic constraint with respect to \mathbf{y} is non-singular (cf. Equations 4.2a, 4.2b or Equations 4.4a, 4.4b).

For DAE systems of the form

$$\mathbf{0} = \mathbf{f}_1(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, t) \quad (4.12a)$$

$$\mathbf{0} = \mathbf{f}_2(\mathbf{x}, \mathbf{z}, t), \quad (4.12b)$$

also called *semi-state systems* or *systems in descriptor form*, the definition of the differential index can be reformulated.

Definition 4.12 (*Differential index of a semi-state system*). The index of a semi-state system is the minimum number of times that the algebraic part (4.12b) must be differentiated with respect to time in order to determine $\dot{\mathbf{x}}$ as a continuous function of the state vector \mathbf{x} , the vector of so-called *semi-state variables* \mathbf{z} and time t .

For linear time-variant differential-algebraic systems, Gear and Petzold have given an algorithm on matrices that can be used for the determination of the index [15]. Since in the following we want to determine the local index for some examples, we recall the algorithm for the determination of the index of linear constant coefficient DAEs [8].

Algorithm 4.1 (Index determination for linear constant coefficient DAEs).

1. The index k is initialised with zero.
2. If the matrix \mathbf{A} in Equation 4.5 is non-singular, we are done. No further iteration is necessary. The algorithm terminates.
3. Otherwise, Equation 4.5 is pre-multiplied by a non-singular matrix \mathbf{P} that transforms the DAE into the form

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{B}}_{11} & \tilde{\mathbf{B}}_{12} \\ \tilde{\mathbf{B}}_{21} & \tilde{\mathbf{B}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{f}}_1 \\ \tilde{\mathbf{f}}_2 \end{bmatrix},$$

- in which the row-rank of the matrix $[\mathbf{A}_{11} \ \mathbf{A}_{12}]$ equals the number of its rows.
4. After differentiation of the algebraic equation $\tilde{\mathbf{B}}_{21}\mathbf{x}_1 + \tilde{\mathbf{B}}_{22}\mathbf{x}_2 = \tilde{\mathbf{f}}_2$, we get the new system

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \tilde{\mathbf{B}}_{21} & \tilde{\mathbf{B}}_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{B}}_{11} & \tilde{\mathbf{B}}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{f}}_1 \\ \dot{\tilde{\mathbf{f}}}_2 \end{bmatrix}.$$

5. The index is increased by one. The old system is replaced by the new one. The algorithm continues with step 2.

Remark 4.6. If the algorithm terminates after k iterations, then the DAE system in the last iteration has the index zero. The DAE system in the next to last iteration is of index one. The original DAE system then has the index k . That is, the algorithm not only determines the index. Actually, it does an index reduction.

In general, the algebraic constraints of a DAE system are nonlinear as in the considered example of a hydraulic Wheatstone bridge. Hence, for each time t_n , often, they can be solved only numerically by iteration. If Equations 4.2a–4.2b derived from a bond graph with causal paths between resistive ports are linearised, then the local index of the linearised system equals one [32]. This means that they can be solved by means of an ODE based method. A widely used code for the numerical solution of DAE systems of index < 2 has become the solver called DASSL [8]. Public domain mathematical software such as Scilab [11, 29] or GNU Octave [2] provide a function that calls the DASSL code. In addition, Octave [2] includes a function that calls DASPCK [1, 30] a further development of the DASSL code.

4.3 Inserting Energy Stores into Causal Paths Between Resistive Ports

It is certainly an essential objective of object-oriented methodologies to separate the modelling phase from the formulation of equations and their numerical solution in order to achieve consistency in the development of large models. Following an object-oriented approach, the interfaces of submodels are connected according to the topology in a system schematic. Constitutive equations of the submodels are expected to be reformulated automatically, if required by the interconnection of the submodels. If library submodels are protected from modifications by the developer of a system model, then obviously less errors can occur. On the other hand, the model developer cannot affect the generation of system equations. Still, on a lower level, there are relations between decisions taken in the modelling phase and aspects with regard to the numerical solution of the generated equations. In former times, when no solvers for differential-algebraic equations systems were available, these relations were frequently used. Even with today's solvers, it makes sense to take into account aspects of an efficient numerical solution of the generated equations in the phase of the model development. Moreover, it may be useful to have the generated equations undergo a symbolic preprocessing before they are passed on to a numerical solver. For instance, it is well known that problems with the numerical solution of DAEs increase with an increase of their index. In Chapter 5, relations between bond graph modelling and the symbolic and numerical solution of system equations will be considered in detail. In the following, first, simple possibilities will be discussed to affect the form of the resulting mathematical model already in the modelling phase.

The algebraic constraints 4.2a can be avoided by inserting energy stores into causal paths between resistive ports. In order to ensure that the dynamic behaviour of the original model is not perceptibly affected, additional energy stores must be small. In complex nonlinear models, it is not easy to estimate how small parameter values of additional energy stores must be. On the other hand, including small energy stores may be justified from physics. There are always dynamic effects that are often neglected because their impact on the overall dynamic behaviour of the system is assumed to be negligible. In the bond graph of Figure 4.2, the two disjoint causal paths between resistors may be removed by adding two C energy stores to the lower 0-junctions accounting for the oil compliance in the volumes of the cylinder in the load diagonal of the bridge (Figure 4.5). From the modified bond graph in Figure 4.5, a state space model of order 3 may be derived so that a DAE solver is not necessary. As the added C elements are linear, the two additional ODEs can be written in the form

$$C_1 \dot{p}_{C_1} = g_1(p_1) + A v - Q_2 \quad (4.13a)$$

$$C_2 \dot{p}_{C_2} = g_4(p_4) - A v - Q_3 . \quad (4.13b)$$

For $C_1, C_2 \rightarrow 0$, the two ODEs turn into the algebraic Equations 4.1b and 4.1e.

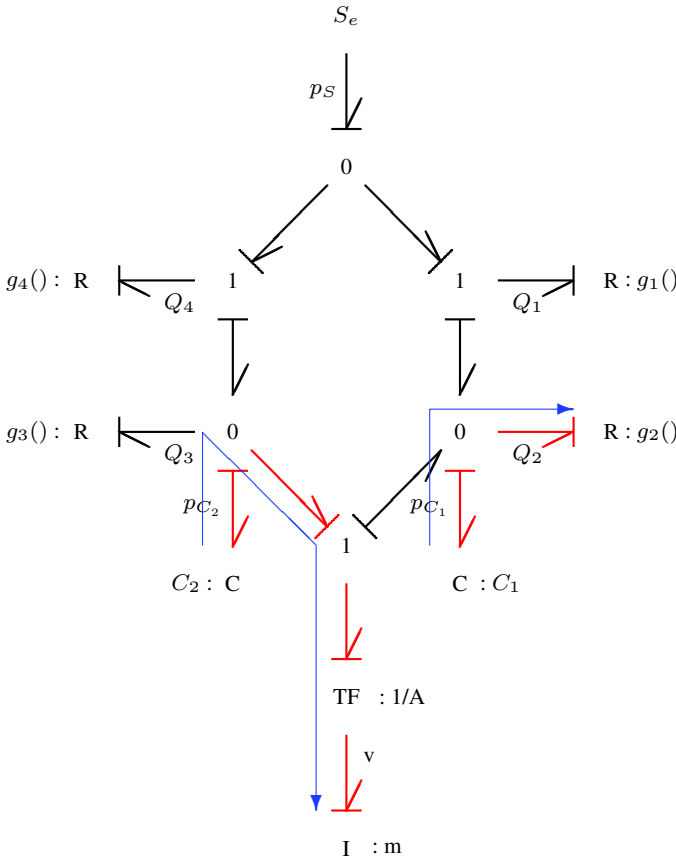


Fig. 4.5 Removal of causal paths between resistors by inserting C energy stores

Disadvantages of such an approach are that the order of the system is increased and that fast dynamic transients are introduced because the capacitances C_1 and C_2 of the added energy C stores are small. Hence, it is true that the result is a set of explicit ODEs, but its solution requires a small step size or an implicit numerical integration algorithm.

Estimation of the Time Constants of a Linear Model

It is well known that if a set of linearised ODEs is solved by means of an explicit numerical integration algorithm, then its step size must be chosen smaller than the smallest time constant. The determination of the time constants means the determination of eigenvalues which is quite costly. A rough estimate, however, can be obtained directly from the bond graph by following causal paths between storage ports and resistive ports. Suppose constitutive equations are linear or have been

linearised, then such causal paths identify first order transients considered isolated from other transients in the system and isolated from forcing inputs.

In Figure 4.5, their time constant is just the product of the parameters of the storage port at one end of the causal path and of the resistive port at its other end. To see this, the following equations are derived from the bond graph in Figure 4.5.

$$\dot{Q}_2 = g_2(p_{C_1}) \quad (4.14a)$$

$$\dot{p}_{C_1} = \frac{1}{C_1} [Q_1 + A v - Q_2] \quad (4.14b)$$

Consider the homogeneous ODE

$$\dot{p}_{C_1} + \frac{1}{C_1} g_2(p_{C_2}) = 0. \quad (4.15)$$

If the nonlinear characteristic of g_2 is replaced by a linear one of slope $1/R_2$ in the neighbourhood of an operating point, then the time constant, τ_{12} , of the free response given by Equation 4.15 is $\tau_{12} = R_2 C_1$. Apparently, by considering all causal paths between storage ports and resistive ports, the smallest time constant can be determined. However, it must be kept in mind that transients are considered decoupled this way. This, however, is not true. The evaluation of the exact value of the smallest time constant requires the solution of an eigenvalue problem. The corresponding considerable effort, however, is not worthwhile. Anyway, the usefulness of an estimation of the time constants is limited to systems of linear constant coefficient ODEs that are solved by means of an explicit integration algorithm. The inspection of the bond graph with regard to time constants of isolated transients may give an indication of how widely time constants are separated when small energy stores are included into the bond graph in order to avoid algebraic loops.

In addition to these first order transients, the C energy stores included in the bond graph of Figure 4.5 lead to another oscillations that are identified by causal paths between a C element and the I energy store. In addition to Equation 4.14b, the following state equation is immediately obtained from the bond graph.

$$\dot{v} = \frac{A}{m} (p_{C_2} - p_{C_1}) \quad (4.16)$$

By differentiation of Equation 4.14b and substituting it into the state equation of the I energy store, a second order ODE results for the output of the C energy store with capacitance parameter C_1 .

$$\ddot{p}_{C_1} + \frac{1}{C_1} \frac{A^2}{m} p_{C_1} = \frac{1}{C_1} (\dot{Q}_1 - \dot{Q}_2 + \frac{A^2}{m} p_{C_2}) \quad (4.17)$$

The natural frequency of the free undamped oscillation of pressure p_{C_1} reads

$$\omega_0^2 = \frac{A^2}{C_1 m}. \quad (4.18)$$

For small values of the fluid filled volume V_1 or the mass m , C energy stores for the fluid compliance in the volumes V_1 and V_2 can remove causal paths between resistive ports, but introduce high frequent oscillations superimposed on the dynamics of the pressures in the volumes. They may be damped by attaching resistors with a small parameter to the 0-junctions representing the two pressures p_{C_1} and p_{C_2} . However, in the case of an explicit integration algorithm, a small step size is still required.

These considerations show that in the case of causal paths between resistive ports, a DAE system can be avoided by including energy stores with a small parameter. Solution of the resulting ODE system, however, generally requires a stiffly stable integration algorithm. Of course, the ODE system derived from the modified bond graph can be converted into a DAE system. If capacitances tended to zero, the ODEs of the C energy stores reduce to algebraic constraints for the pressures in the volumes of the cylinder.

Finally, particularities of a model, if there are any, can be used to remove causal paths between resistive ports. If, for instance, the resistors of a bridge circuit are the control orifices of a spool valve with zero overlap in the central position, then two orifices in a diagonal are always open while the other two are closed. This can be used for a model reduction that removes the causal paths between resistive ports. Although such an approach leads to a model allowing for an efficient numerical computation, it is not generally applicable. On the other hand, it is always possible to include energy stores with small parameters. However, it appears that there is only a need for modifying a model this way if the available modelling and simulation software does not accept DAEs of index one and if a model reduction is not possible.

4.4 Causal Paths Between Storage Ports of the Same Type

In this section, we will exclude causal paths between resistive ports. Instead, we will consider causal paths between independent and dependent storage ports (van Dijk calls them *class-1 zero-order causal paths* [32]). Such causal paths often appear in bond graphs of multibody systems with algebraic constraints between the velocities of some rigid bodies caused by joints connecting them. In Section 3.4, we already considered a simple example of two inertias coupled by a transformer with constant modulus (Figure 3.14). In this case, the dependent energy store can be transformed over the transformer and can be combined with the independent one into a new one with integral causality. If energy stores have a linear characteristic, then C energy stores attached to a 0-junction or I energy stores connected to a 1-junction can be combined into one equivalent energy store.

If such reductions are not possible or if they are not performed, then the mathematical model to be derived from a bond graph with causal paths between storage ports of the same type is of the form of a DAE system

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}_1(\mathbf{x}_i(t), \dot{\mathbf{x}}_d(t), \mathbf{u}(t)) \quad (4.19a)$$

$$\mathbf{x}_d(t) = \mathbf{f}_2(\mathbf{x}_i(t), \mathbf{u}(t)) . \quad (4.19b)$$

In Equations 4.19a and 4.19b, $\tilde{\mathbf{x}}_i$ denotes the vector of independent state variables, whereas \mathbf{x}_d is the vector of dependent state variables. Again, \mathbf{u} is the vector of all system inputs.

Van Dijk has shown that the linearised DAE system obtained from a bond graph with causal paths between independent and dependent storage ports is of local index one [32].

Example: Slider Crank Mechanism

For illustration, consider the often used example of a simple slider crank mechanism depicted in Figure 4.6. A massless rod of length L links a flywheel of moment of inertia J to a piston of mass m . In [3], Allen considers a similar slider crank mechanism as an introductory example in the context of establishing Lagrange's equations for complex mechanical mechanism (cf. Section 4.10).

The rod establishes a geometric constraint between the angular position, $\varphi(t) := \int_0^t \omega(\tau) d\tau$, of the flywheel and the position of the piston. Differentiation with respect to time yields a constraint between the angular velocity, ω , and the translational velocity, v , of the piston

$$v = T(\varphi) \times \omega \quad (4.20)$$

and

$$T(\varphi) = \frac{r(r \cos \varphi + \sqrt{L^2 - r^2 \sin^2 \varphi}) \sin \varphi}{\sqrt{L^2 - r^2 \sin^2 \varphi}} . \quad (4.21)$$

Assuming that rotational power is transformed into translational power without any losses, yields for the moment \tilde{M} transformed into the force \tilde{F} acting on the piston

$$\tilde{M} = T(\varphi) \tilde{F} . \quad (4.22)$$

Thus, this transformation can be represented by a displacement modulated transformer of modulus $T(\varphi)$. Such transformers are often used for representing trans-

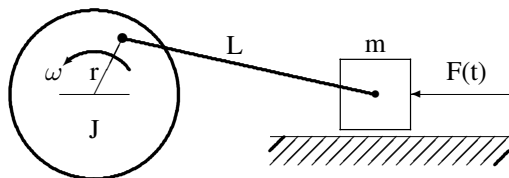


Fig. 4.6 Slider crank mechanism

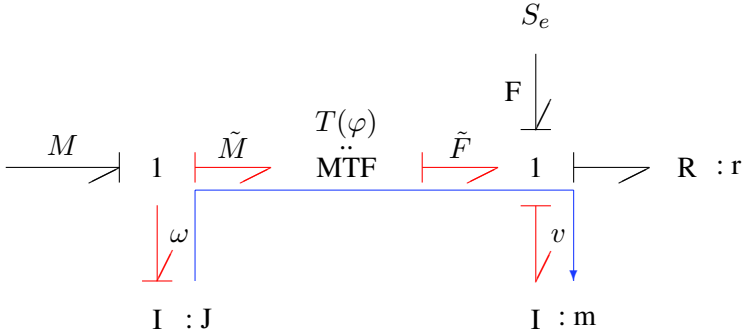


Fig. 4.7 Bond graph of the slider crank mechanism

formations between reference frames when the planar or 3D motion of mechanical systems is modelled. They have no counterpart in electrical engineering. Due to the constraint of the velocities, one I element in the bond graph of Figure 4.7 must have derivative causality. For the energy store with preferred integral causality, the state equation

$$\dot{\omega} = \frac{1}{J} [M - T(\varphi) \times (m\dot{v} + rv - F)] \quad (4.23)$$

is derived from the bond graph. In addition to the angular velocity ω in this example, the kinematic displacement of the φ is needed as another state variable which is typical for modelling planar or 3D motion of mechanical systems.

$$\dot{\varphi} = \omega \quad (4.24)$$

All three equations can be combined into a linear implicit DAE system also called a linearised descriptor form [13, 24].

$$\begin{bmatrix} J & T & m & 0 \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{v} \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 & Tr & 0 \\ -1 & 0 & 0 \\ T & -1 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ v \\ \varphi \end{bmatrix} = \begin{bmatrix} M + TF \\ 0 \\ 0 \end{bmatrix} \quad (4.25)$$

Due to the algebraic dependency between the velocities, the matrix pre-multiplying the time-rate of the descriptor vector is singular. Thus, the mathematical model derived from a bond graph with a causal path between an independent and a dependent storage port, in fact, takes the form of a true DAE system. If algorithm 4.1 is applied to this DAE, then the matrix pre-multiplying the time-rate of the descriptor vector becomes non-singular after one step. Hence, the local index of the DAE system equals one as to be expected.

The DAE system can be avoided, i.e., an explicit state space model can be derived, if the dependent energy store is transformed over the transformer like in the example of Figure 3.14. However in contrast to that example, here, the modulus of the transformer is not constant. Hence, the inertia of the resulting I element is not

constant. We come back to this issue in Section 4.10 when we consider the derivation of Lagrange's equations from bond graphs. Another option that can be justified from physics is to neutralise the kinematic constraint by assuming that the rod is not completely rigid, but has some compliance. With this assumption, a C energy store with a small parameter can be used to remove the causal path between the two inertias. Like in the hydraulic bridge example (Figure 4.2), this leads to a high frequency oscillation that can be damped by including a resistor in addition to the C energy store. The resistor may also be justified for physical reasons. Taking a closer look at the rod, it turns out that it is not a purely elastic, but quite stiff link. There are also energy losses in the rod. The simplest approximation to a continuum model is to use a pair of C and R elements along with parameters that are estimated on the basis of experience. For numerical stability, it is reasonable to use an implicit integration algorithm.

4.5 Closed Causal Paths

So far, causal paths between resistive ports or between independent and dependent storage ports have been considered. In addition, closed causal paths in the junction structure can occur. They are called *causal loops* (Definition 3.10).

For illustration, consider the simple example of a bond graph with a bond loop displayed in Figure 4.8. Figure 4.9 shows an electrical circuit that can be represented by the bond graph in Figure 4.8. As can be seen from the bond graph of Figure 4.8, the causality of the flow source and the preferred integral causality of the energy stores do not propagate into the junction structure. In this simple example, there is a need to perform step 5 of the sequential causality assignment procedure by choosing a bond and assigning causality to it. This gives rise to the introduction of the notion of *strong (weak) causal determination*.

Definition 4.13 (*Strong (weak) causal determination of a junction*). A bond imposes a *strong* causal determination on a junction J it is connected to if one of its power conjugate variables determines the variable common to all remaining adjacent bonds. Otherwise, the bond gives a *weak* causal determination to the junction [26].

Remark 4.7. If the causal stroke of a bond connected to a 0-junction is on its end attached to the 0-junction, then the effort common on all adjacent bonds is determined. That is, the effort imposed on a 0-junction is propagated to ports connected to it.

In the example of Figure 4.8, the two 0-junctions have a *weak* causal determination. In order to complete causality assignment, causality must be chosen at one internal bond (Definition 2.9) No matter which bond is chosen, the result is a causal loop (Definition 4.6) associated with two open signal flow loops of opposite orientation as depicted in Figure 4.10. One signal flow loop only relates the efforts of the

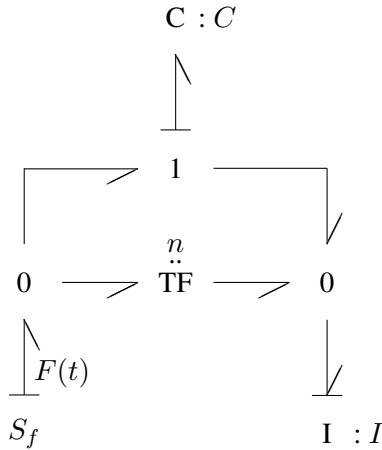


Fig. 4.8 Bond graph with a bond loop

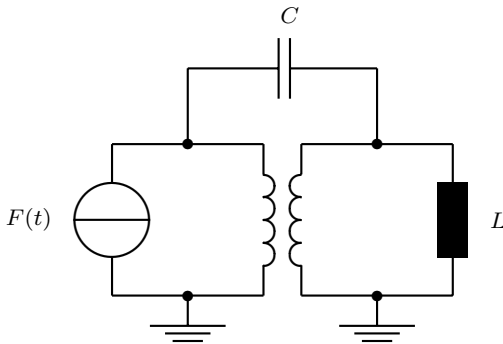


Fig. 4.9 Circuit realisation of the bond graph with a bond loop

internal bonds, while in the other, only the corresponding flows are involved. Thus, each algebraic loop contributes an auxiliary variable. If the outputs of the transformer e_4 and f_3 are chosen as auxiliary variables, then the following equations can be derived from the bond graph of Figure 4.10

$$0 = e_4 - \frac{1}{n} e_4 - e_C \tag{4.26a}$$

$$0 = F - \frac{1}{n} f_3 + f_3 - f_I \tag{4.26b}$$

$$\dot{e}_C = \frac{1}{C} (F - \frac{1}{n} f_3) \tag{4.26c}$$

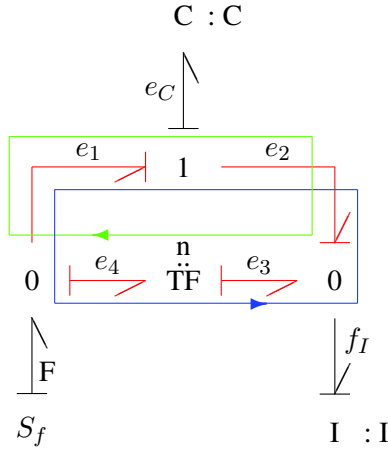


Fig. 4.10 Bond graph with causal loop and two associated topological loops

$$\dot{f}_I = \frac{1}{I} (e_4 - e_C) . \tag{4.26d}$$

Since in this example the system of equations determining the algebraic variables is linear, it can be symbolically solved and the algebraic variables can be removed from the state equations.

$$\dot{e}_C = \frac{1}{C(n-1)} [nF - f_I] \tag{4.27a}$$

$$\dot{f}_I = \frac{1}{I(n-1)} e_C \tag{4.27b}$$

However, as can be seen from the result, this elimination step is only possible for a transformer modulus $n \neq 1$. Ort and Martens [26] have given a mathematical criterion for the solvability of the algebraic equations of the junction structure. In [28], Rosenberg and Andry have given a criterion that can be directly checked on the bond graph. First, consider the mathematical criterion applied to our example. By looking at the bond graph in Figure 4.10, the following equations for efforts can be written.

$$\begin{bmatrix} e_S \\ e_I \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & n & 0 \end{bmatrix} \begin{bmatrix} e_C \\ e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} \tag{4.28}$$

Let us combine the efforts that are outputs of the junction structure into a vector $\mathbf{e}_{out} = (e_S, e_I)^T$ and all internal efforts into a vector $\mathbf{e}_{int} = (e_1, e_2, e_3, e_4)^T$.

The vector of efforts that are input to the junction structure has only one component $\mathbf{e}_{in} = (e_C)$. With these vectors, the system of equations 4.28 has the form

$$\begin{bmatrix} \mathbf{e}_{out} \\ \mathbf{e}_{int} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{S}_3 & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{e}_{in} \\ \mathbf{e}_{int} \end{bmatrix}. \tag{4.29}$$

In general, the system of equations 4.29 can be solved for \mathbf{e}_{int} if $\det(\mathbf{I} - \mathbf{S}_4) \neq 0$. The dimension of the identity matrix \mathbf{I} equals the number of internal bonds. In the example under consideration, the determinant is equal to $1 - n$. Solvability of the junction structure equations means that the mathematical model can be written in state space form.

Instead of establishing a linear system of junction structure equations and to check for its solvability at the level of equations, it appears to be more convenient to use a criterion that can be directly checked on the bond graph. Such a criterion has been given by Rosenberg and Andry [28]. In order to recall it here, some definitions are needed.

Definition 4.14 (Influence coefficient). The influence coefficient of a junction structure node is the ratio of the output variable to the input variable for a particular signal flow loop fragment associated with *two* adjacent bonds of opposite causal orientation [28].

For illustration of this definition, consider the bond graph fragment depicted in Figure 4.11. The causal path $1 - 2 - 3$ is associated with two signal flow loop fragments, one for the efforts of the bonds and one of opposite orientation for the flow variables. Consider the flow variables. They are equal at the 1-junction. Consequently, the influence coefficient $c_{12} = f_1/f_2$ is equal one. The influence coefficient $c_{23} = f_2/f_3$ of the transformer has the value n . Since 1-junctions add up effort variables, the sign of the influence coefficient $c_{21} = e_2/e_1$ depends on the power orientation of the bonds. For the transformer, we have $c_{32} = e_3/e_2 = c_{23}$ (Note that there is another causal path $4 - 2 - 3$).

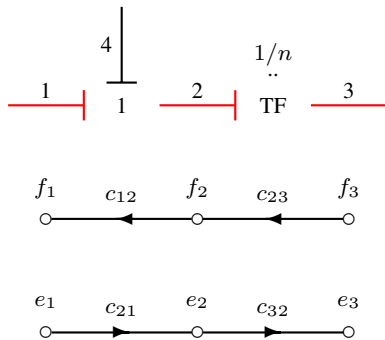


Fig. 4.11 Causal path $1 - 2 - 3$ and associated signal flow loop fragments

Definition 4.15 (*Loop gain of a topological loop*). The loop gain of a topological loop is the product of all influence coefficients [28].

There are two topological loops of opposite orientation associated with a causal loop in a bond graph (cf. Figure 4.10). It has been shown that the loop gain of both topological loops associated with a causal loop are equal. Hence, due to this unique value, the gain of a causal loop can be defined.

Definition 4.16 (*Causal loop gain*). The gain of a causal loop is the loop gain of the two topological loops of opposite orientation associated to the causal loop.

The bond graph based rule given by Rosenberg and Andry now states that the linear equations of a general junction structure (GJS) are solvable if and only if causal loops are pairwise disjoint and if the loop gain of every causal loop is different from +1. ([28], Theorem 3)

Van Dijk divides causal loops into two classes according to their loop gain. He calls causal loops of loop gain different from one *class-4 zero-order causal paths* and denotes loops of loop gain equal to one as *class-5 zero-order causal paths*. Concerning the local index of a DAE system derived from a bond graph with causal loops, he proves that it is equal to one if, as required in Rosenberg and Andry's theorem, causal loops in the graph are pairwise disjoint and if the loop gain of every causal loop is different from +1 ([32], Proposition 5.7).

Let us apply Rosenberg's and Andry's criterion to the example in Figure 4.10. If G_e denotes the loop gain of the topological loop of efforts variables and G_f the loop gain of the signal flow loop of flow variables, then their common value is equal to that of the transformer modulus n .

$$\begin{aligned} G_e &= c_{21} \times c_{32} \times c_{43} \times c_{14} \\ &= 1 \times 1 \times n \times 1 \end{aligned} \tag{4.30a}$$

$$\begin{aligned} G_f &= c_{12} \times c_{41} \times c_{34} \times c_{23} \\ &= 1 \times (-1) \times n \times (-1) \end{aligned} \tag{4.30b}$$

If $n \neq 1$, then the linear equations of the junction structure are solvable, as we know from the previous analysis at equations level.

If the modulus of transformers or the ratio of gyrators is not constant, e.g., there are displacement modulated transformers in the weighted junction structure, then the rule of Rosenberg and Andry is no longer applicable. In that case, the mathematical model takes the form of the Equations 4.2a–4.2b of a DAE system.

Remark 4.8. Since both topological loops associated with a causal loop either relate efforts or flow variables of all bonds of the causal loop, they are open loops according to Definition 4.6. In the bond graph of Figure 4.10, an even number of bonds of the causal loop has the same power orientation. Therefore, its loop gain is positive. If we changed the power orientation of the bond annotated with e_2 , then the loop gain would be $(-n)$. In the case of a causal mesh, viz., a closed causal path with an odd number of gyrators (Definition 3.11), there are also two open loops. The

absolute value of their loop gains is the same. However, they are different in sign. Brown calls a mesh *even* if the loop gain of both open loops of opposite orientation is positive, an *odd mesh* if the loop gain of both open loops is negative, and a *neutral mesh* if the loop gain of both open loops have opposite signs [10]. This way, the Definition 3.9 of a simple even (odd) mesh is extended to general meshes with transformers or gyrators.

4.6 Bond Graphs with Causal Paths from Different Classes

In the previous sections, bond graphs with causal paths from one class only have been considered. Now, bond graphs will be allowed to have causal paths from different classes. We already know that the underlying mathematical model has the form of a DAE system. The interesting question, however, is of what index they are. In the following, we will confine ourselves by considering some examples. A more comprehensive analysis has been performed by van Dijk in [32].

Figure 4.12 shows a modification of the circuit example of Figure 4.9. Consider the associated bond graph displayed in Figure 4.13. There is a causal path 5 – 2 – 6 – 7 – 8 between resistors R_1 – R_2 and another causal path 1 – 2 – 3 – 4 between the independent C energy store C_1 and the dependent C energy store C_2 . Both causal paths have bond 2 in common. These causal paths result in the following manner. First, causality of the flow source and preferred integral causality at the C energy stores do not propagate into the junction structure. Making a choice and assigning resistance causality to either R_1 or R_2 leads to a causal conflict at the upper 1-junction that can be removed by changing preferred integral causality at C_2 into derivative causality. The result is a causal path between the two energy stores. Assigning conductance causality to either one resistor or to both of them still leaves the bond graph causally incomplete such that causality at one of the bonds of the bond loop must be chosen. The result would be a causal loop. For both open loops associated with this causal loop, two algebraic variables would be needed.

By working along causal paths in the bond graph of Figure 4.13, the following equations can be derived.

$$e_5 = R_1 [n(F - f_8 - C_2 \dot{e}_4) + C_2 \dot{e}_4] \quad (4.31a)$$

$$f_8 = \frac{1}{R_2} n(e_1 + e_5) \quad (4.31b)$$

$$e_4 = (n - 1)(e_1 + e_5) \quad (4.31c)$$

$$\dot{e}_1 = \frac{1}{C_1} [n(F - f_8) - (n - 1)C_2 \dot{e}_4] \quad (4.31d)$$

They can be written in the linear implicit form

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \dot{\mathbf{x}} + \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_3 & \mathbf{B}_4 \end{bmatrix} \mathbf{x} = \mathbf{b}, \quad (4.32)$$

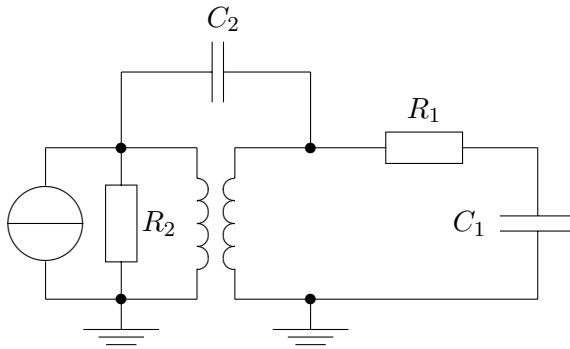


Fig. 4.12 Circuit example

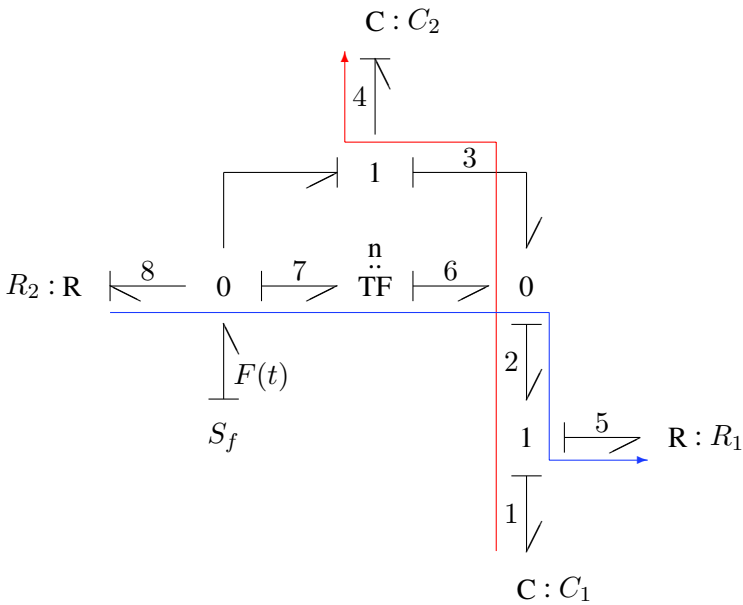


Fig. 4.13 Bond graph with class-2 and class-1 zero-order causal paths

in which the matrix pre-multiplying the derivative of the descriptor vector is singular. Application of algorithm 4.1 yields that the matrix

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{B}_3 & \mathbf{B}_4 \end{bmatrix}$$

is non-singular. That is, the (local) index of the DAE system is equal to one. This result generally holds for bond graphs with causal paths between resistive ports

as well as for bond graphs with causal paths between independent and dependent storage ports [32].

Now, let us modify the bond graph in Figure 4.13 such that in addition to the class 1 zero-order causal path between storage ports, a causal loop results (Figure 4.14). Again, assigning fixed causality to the flow source and integral causality to the energy stores leaves the bond graph causally incomplete. Choosing causality at bond 7 (as depicted in Figure 4.14) leads to a causal conflict at the lower 1-junction that can be removed by changing integral causality at the lower C energy store C_2 into derivative causality. Thus, there are two causal paths, 1 – 2 – 3 – 4 – 5 and 1 – 2 – 6 – 7 – 9 between both C energy stores. Only one of them has been highlighted for clarity.

Due to the causal loop 6 – 7 – 8 – 2, the output variables, e_7 and f_6 , of the transformer are chosen as auxiliary algebraic variables. Then, the following equations can be written.

$$\dot{e}_1 = \frac{1}{C_1} [-f_6 + n_2 C_2 \cdot \dot{e}_2 + f_{10}] \tag{4.33a}$$

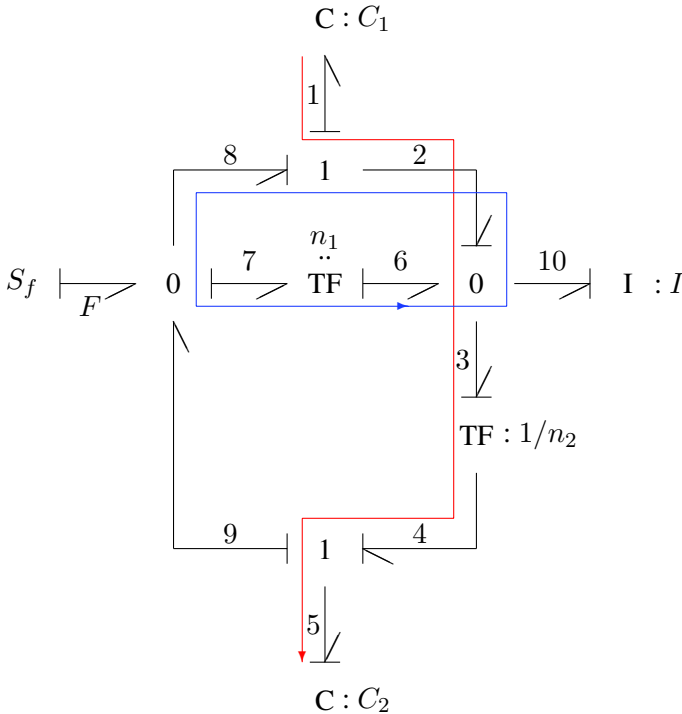


Fig. 4.14 Bond graph with causal loop and class 1 zero-order causal path

$$\dot{f}_{10} = \frac{1}{I} [e_7 - e_1] \quad (4.33b)$$

$$e_5 = n_2 (e_7 - e_1) - e_7 \quad (4.33c)$$

$$f_6 = n_1 [F + C_2 \dot{e}_5 - (f_{10} + n_2 C_2 \dot{e}_5 - f_6)] \quad (4.33d)$$

$$e_7 = n_1 (e_7 - e_1) \quad (4.33e)$$

If they are written in linear implicit form (cf. Equation 4.32), then the matrix pre-multiplying the descriptor vector is singular, as to be expected. After one step of algorithm 4.1 the resulting matrix

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{B}_3 & \mathbf{B}_4 \end{bmatrix} = \begin{bmatrix} C_1 & 0 & 0 & 0 & -n_2 C_2 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -n_1 (1 - n_2) C_2 \\ n_1 & 0 & 0 & (1 - n_1) & 0 \\ n_2 & 0 & 0 & (1 - n_2) & 1 \end{bmatrix} \quad (4.34)$$

is still singular. Hence, the (local) index is > 1 . This result does not hold only for the example for which the index is equal to 2. It can be shown that in general, the local index is > 1 , if the bond graph includes class 1 zero-order causal paths that join bonds with causal loops ([32], Proposition 5.10).

4.7 Causal Loops of Unity Loop Gain

Finally, we will address the case of causal loops of unity loop gain by considering the example of the electrical circuit depicted in Figure 4.15 [9, 32]. It will be shown that bond graphs do not directly display Kirchhoff's generalised non-local current law for cut-sets, but indirectly via a causal loop of unity loop gain or by means of a causal conflict at a 0-junction. It appears that the standard sequential causality assignment procedure (SCAP) yields a causal pattern that does not adequately reflect the global continuity of flow variables. This is why van Dijk has proposed a modification of the SCAP that avoids causal loops of unity loop gain. For details, see [32]. Moreover, by means of the delta circuit example (Figure 4.15), we will show that causal loops of unity loop gain can result in DAEs of index > 1 . Thus, with regard to a robust numerical solution of the DAE system, they should be avoided.

First, we assume that all passive elements of the delta circuit in Figure 4.15 have an invertible characteristic such that the computational causality at the ports of their corresponding bond graph elements is indifferent. As can be seen from the bond graph in Figure 4.16, fixed causality of the flow source and preferred integral causality at the I energy stores do not propagate into the junction structure.

If conductance causality is chosen at one of the resistor ports, then it imposes a strong causal determination at the 1-junction the resistor is connected to (cf. Definition 4.13). In the end, it results in a causal conflict at a 0-junction (Figure 4.17).

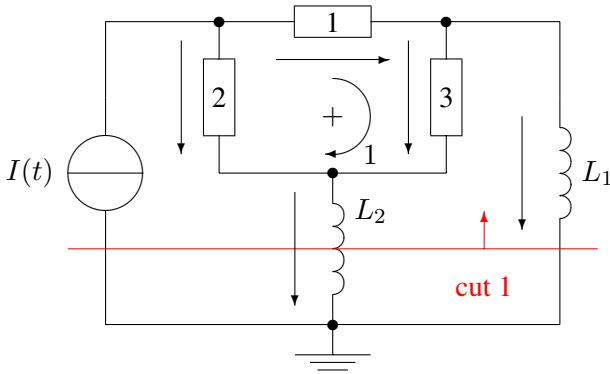


Fig. 4.15 Electrical network with a delta subnetwork of resistors

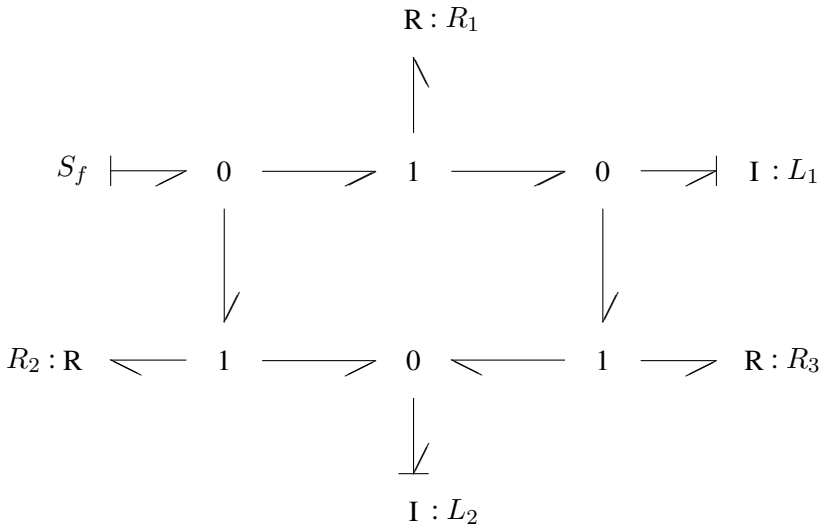


Fig. 4.16 Bond graph of the electrical network

If resistance causality is chosen at the resistive ports in order to avoid a causal conflict, then this causality is not propagated and the bond graph remains causally incomplete. Therefore, causality must be chosen at an internal bond leading to a causal loop. Since there are no transformers or gyrators in the loop, its loop gain is equal to one. There are two open loops of opposite orientation associated to the causal loop as depicted in Figure 4.18.

Consequently, in addition to the outputs of the energy stores, an effort variable must be chosen from one open loop and a flow variable from the other. These have been marked in the bond graph of Figure 4.18. Using these algebraic auxiliary variables, u and i , the following equations can be written

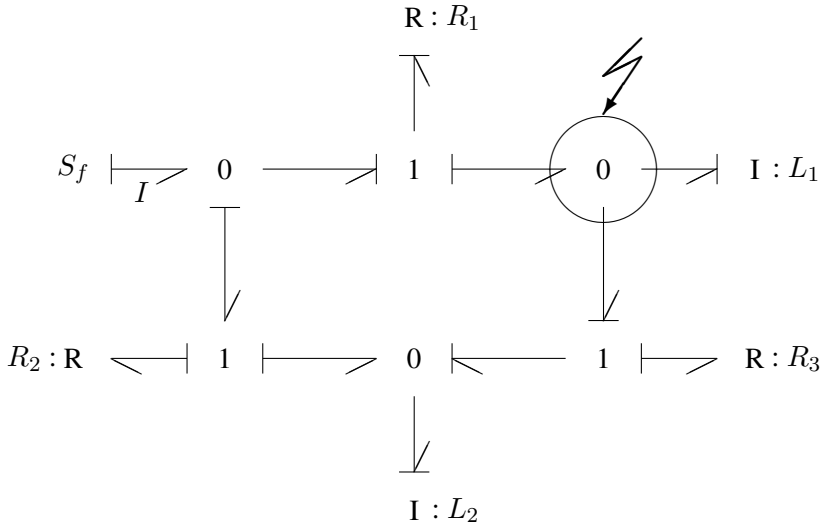


Fig. 4.17 Bond graph with a causal conflict at a 0-junction

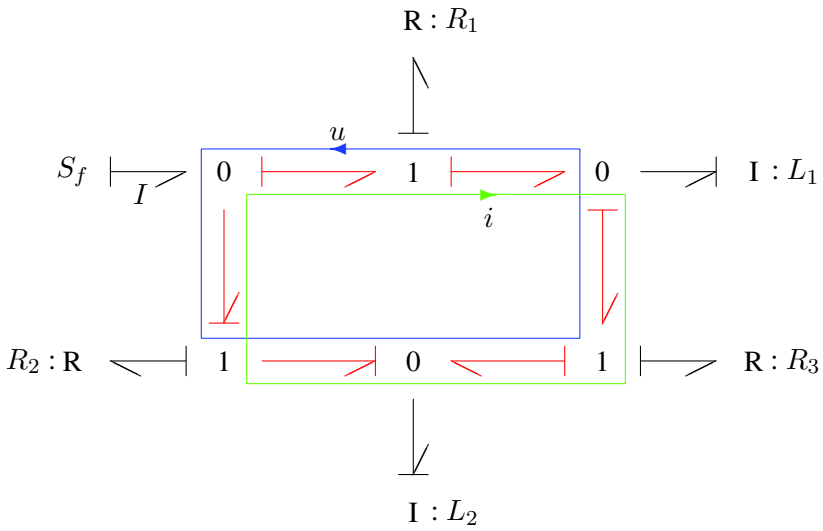


Fig. 4.18 Causal loop with unity loop gain

$$u_1 = R_1 i \quad (4.35a)$$

$$u_2 = R_2 (I(t) - i) \quad (4.35b)$$

$$u_3 = R_3 (i - i_{L_1}) \quad (4.35c)$$

$$\frac{d}{dt} i_{L_1} = \frac{1}{L_1} (u - u_2 + u_3) \quad (4.35d)$$

$$\frac{d}{dt} i_{L_2} = \frac{1}{L_2} (u - u_2) . \quad (4.35e)$$

Walking along the open loop for the efforts yields

$$u = u_1 + u_3 + u - u_2 , \quad (4.36)$$

which is the sum of all voltages around mesh 1 in the circuit (cf. Figure 4.15). From the second topological loop, we obtain Kirchhoff's generalised current law for cut-set 1.

$$i = I(t) - i_{L_1} - i_{L_2} + i \quad (4.37)$$

The algebraic relation between the outputs of the I energy stores,

$$0 = I(t) - i_{L_1} - i_{L_2} , \quad (4.38)$$

can also be obtained from the sum of all flows at the 0-junction with an undetermined common effort if the outputs of the resistors R_1 and R_2 are substituted.

From Equation 4.38, it can be concluded that the delta circuit of the three resistors behaves like a node for which Kirchhoff's current law must hold. In contrast to an actual node, however, it transforms electrical energy into heat. If, in addition to the principle of power conservation, a subsystem complies globally with Kirchhoff's node law, it is said to have the *nodicity* property [19, 33]. The term goes back to Paynter. Electrical networks of passive elements have this nodicity property. In particular for the delta sub-structure of the resistors, continuity of the currents into the structure holds. If the delta structure of resistors is represented by a 3-port, then, according to the causality pattern required for a 0-junction locally representing a balance of flows, an effort should be imposed at one of its ports (Figure 4.19). However, the nodicity property is not directly expressed in the bond graph of Figure 4.18. On the contrary, preferred integral causality at the I energy stores suggests that the currents into the delta subnetwork of resistors are independent. However, due to Kirchhoff's generalised current law for cut-sets (Equation 4.38), this is not true. If according to Equation 4.38, derivative causality is assigned to one of the two I energy stores, say L_1 , then the causal loop of unity loop gain disappears. Instead, a causal path 8 – 9 – 6 – 10 emerges between the energy stores and two other causal paths 1 – 2 – 3 – 4 and 5 – 6 – 7 – 4 between resistors $R_3 - R_1$, or $R_2 - R_1$ (Figure 4.20).

From the previous section, we know that the DAE system derived from the bond graph in Figure 4.20 is of index one. Hence, it can be numerically solved with a BDF based solver. However, if we leave the causal loop of unity loop gain in the bond graph of Figure 4.18, then the corresponding DAE system is of index > 1 . In fact, if

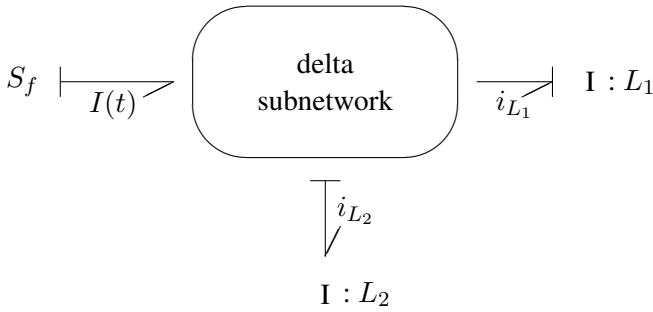


Fig. 4.19 Bond graph of the electrical network using a *nodic* 3-port

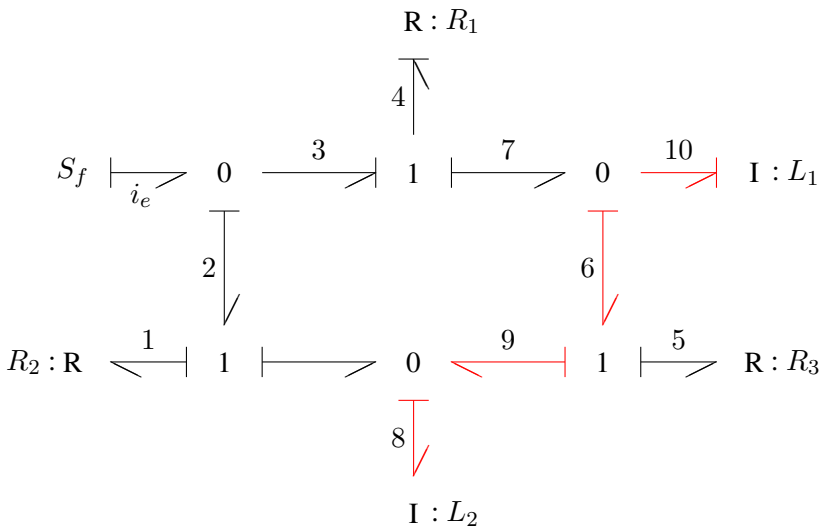


Fig. 4.20 Causalities according to the nodicity property

we write Equations 4.35a–4.35e and Equations 4.36 and 4.38 in linear implicit form

$$\begin{bmatrix} L_1 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{L_1} \\ i_{L_2} \\ u \\ i \end{bmatrix} + \begin{bmatrix} R_3 & 0 & -1 & -(R_2 + R_3) \\ 0 & 0 & -1 & -R_2 \\ -R_3 & 0 & 0 & R_1 + R_2 + R_3 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{L_1} \\ i_{L_2} \\ u \\ i \end{bmatrix} = \begin{bmatrix} -R_2 I(t) \\ -R_2 I(t) \\ R_2 I(t) \\ I(t) \end{bmatrix}, \tag{4.39}$$

then after one step of algorithm 4.1, the resulting matrix

$$\left[\begin{array}{cc|cc} L_1 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 \\ \hline -R_3 & 0 & 0 & R_1 + R_2 + R_3 \\ 1 & 1 & 0 & 0 \end{array} \right]$$

is still singular. Hence, the DAE system is of index > 1 .

The problem with the standard sequential causality assignment procedure is that it does not give advice on how to proceed such that the nodicity property is directly expressed in terms of causalities. This was one reason for van Dijk to propose a modification of the SCAP [32].

4.8 Algebraic Loops due to Internal Modulation

From the previous considerations, we know that causal paths between resistive ports, or between independent and dependent storage ports, or closed causal paths in the junction structure represent algebraic constraints such that the mathematical model derived from the bond graph has the form of a DAE system. However, even in bond graphs with no such causal paths, algebraic loops (Definition 4.3) can appear due to so-called *internal modulation*.

Definition 4.17 (Internal modulation). If a bond graph element is modulated by a power variable, then it is said to be *internally modulated*.

For illustration, consider the two examples in Figure 4.21. One constitutive equation of the transformer in Figure 4.21a) reads

$$f_2 = g(f_2) f_1, \tag{4.40}$$

which cannot be symbolically solved for f_2 in general. Thus, a state equation without algebraic constraint is not possible, although there is no causal path between resistive ports or between independent and dependent storage ports.

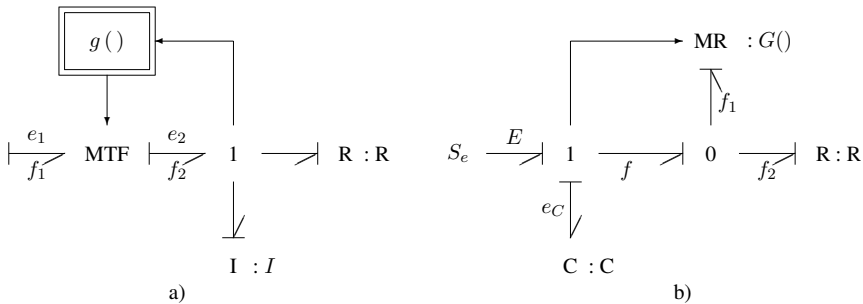


Fig. 4.21 Examples of internal modulation

For the example with the modulated resistor (Figure 4.21b, see also [12]), the following equations can be written

$$f_1 = G (E - e_C, f_1 + f_2) \tag{4.41a}$$

$$f_2 = R (E - e_C) \tag{4.41b}$$

$$\dot{e}_C = \frac{1}{C} (f_1 + f_2) . \tag{4.41c}$$

In both cases, modulation by a power variable originating from a junction results in an implicit nonlinear algebraic equation. We already used a modulated transformer for a model of the slider crank mechanism (see Figures 4.6 and 4.7). However, in that case, the transformer is modulated by a kinematic displacement, which is a state variable needed to describe the system state. On the contrary, in Figure 4.21a, the transformer is modulated by a *power* variable. This is not merely a hypothetical possibility. Consider the bond graph displayed in Figure 4.22. It represents a simplified model of a series motor in which the inductances of the field and the armature coils have been neglected [31]. Let r denote the ratio of the gyrator. Then, the following equations are obtained from the bond graph.

$$M = r \times i \tag{4.42a}$$

$$r = f (i) \tag{4.42b}$$

$$u = r \times \omega \tag{4.42c}$$

$$i = \frac{1}{R} (E(t) - u) \tag{4.42d}$$

Due to hysteresis and saturation, $f()$ is a nonlinear function such that the current i is determined by an implicit nonlinear relation. If the inductances of the coils are taken into account by adding an I element to the 1-junction, then the algebraic loop disappears.

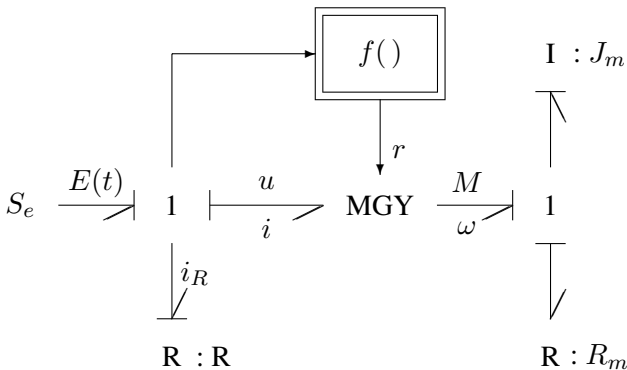


Fig. 4.22 Simplified bond graph of a series motor

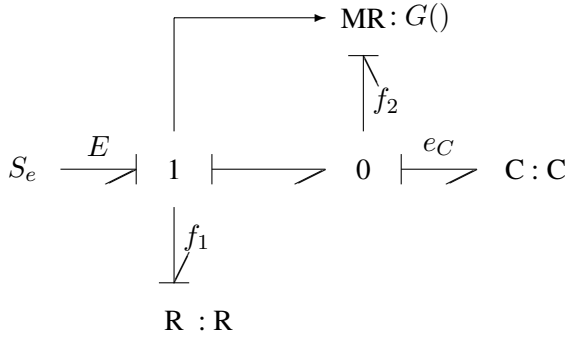


Fig. 4.23 Internal modulation without algebraic loop

Internal modulation does not necessarily result in algebraic loops. For illustration, we interchange the resistor and the C element in Figure 4.21b (cf. [12]). From the modified bond graph (Figure 4.23), we obtain the equations

$$f_1 = \frac{1}{R} (E - e_C) \tag{4.43a}$$

$$f_2 = G(e_C, f_1) \tag{4.43b}$$

$$\dot{e}_C = \frac{1}{C} (f_1 - f_2). \tag{4.43c}$$

Substitution of the output variables of the resistors into the equation of the energy store yields a state equation without algebraic constraint.

The examples of Figure 4.21 show that internal modulation may lead to algebraic loops that are not detected by inspecting causal paths alone. In addition, signals must also be taken into account that originate from junctions and modulate bond graph elements. On the other hand, internal modulation does not necessarily result in algebraic loops as the bond graph of Figure 4.23 illustrates. It is this possibility of algebraic loops not expressed by causal paths that explains why at the beginning of this chapter sources, transformers and gyrators have only been allowed to be modulated by state variables or system inputs. Finally, let us consider an example with an internally modulated source depicted in Figure 4.24. Since there is no causal path between resistors, the bond graph at a first glance suggests that the mathematical model can be reduced to state space form. From the bond graph of Figure 4.24, the following three algebraic equations for the controlled source and the two resistors and an ODE for the C energy store can be written.

$$u = k i_2 \tag{4.44a}$$

$$i_2 = \frac{1}{R_2} (u - u_C) \tag{4.44b}$$

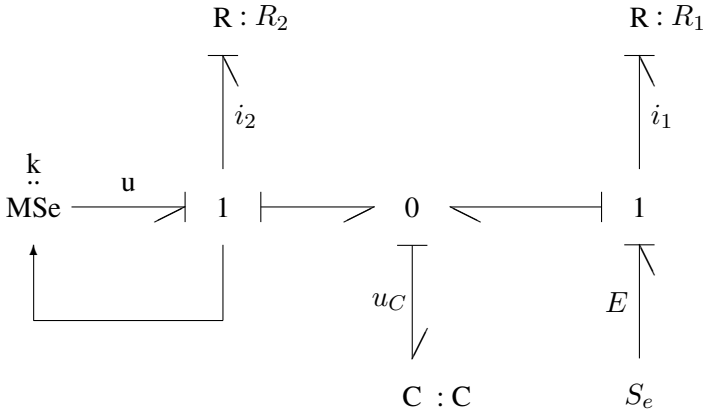


Fig. 4.24 Bond graph with an internally modulated source

$$i_1 = \frac{1}{R_1} (E - u_C) \tag{4.44c}$$

$$\dot{u}_C = \frac{1}{C} (i_1 + i_2) \tag{4.44d}$$

In the example of the voltage follower (Figure 3.22), the output of the controlled source is proportional to the difference of the input voltage. The voltage across the capacitor is a state variable. In this example, the output of the controlled source is proportional to the power variable i_2 . By taking into account the signal modulating the left-hand side source, we see that there is a flat loop between the controlled source and the resistor with resistance R_2 . Apparently, if the source was independent, then there would be no algebraic loop associated with the causal path between the source and the resistor R_2 . Substituting the constitutive equation of the controlled source into the equation of the resistor R_2 yields an equation that determines the controlling signal i_2 .

$$R_2 i_2 = k i_2 - u_C \tag{4.45}$$

Equation 4.45 is just the sum of all efforts at the left 1-junction. It is solvable for i_2 if $k \neq R_2$. Under this condition, the equations derived from the bond graph can be reduced into one state equation for u_C . In case $k = R_2$, the DAE system is of index one (The algebraic constraints need to be differentiated once in order to obtain $\dot{u}_C(t) = 0$. For $k = R_2$, the circuit degenerates into one with no dynamic element). The simple example shows that conclusions with regard to the form of the mathematical model cannot be drawn from considering causal paths alone if internal modulation is not excluded. The form of the mathematical model or the index of a DAE system may even depend on parameter values.

In [12], Cornet and Lorenz show how causality assignment and causality propagation can be used to establish a set of sorted equations in the sense of a compu-

tational order. Since algebraic loops prevent a complete sorting, their detection is essential in this context.

Breedveld argues that internal modulation should be relinquished from a conceptual point of view because basic features of elements can be changed and elements with arbitrary characteristics can be constructed by means of internal modulation [7]. Thus, there would be a potential risk that modelling violates physical conservation laws. Fundamental features of physical modelling would not be guaranteed any more. Van Dijk proposes to use a multiport with appropriate nonlinear constitutive equations instead of local internal modulation [32] where possible. He considers internal modulations as an exception.

4.9 The Method of Relaxed Causalities

So far, we have only used the standard procedure of Karnopp and Rosenberg for assigning and propagating computational causalities in bond graphs. Resulting causal patterns and their implication with regard to the form of the mathematical model have been analysed. The notion of the causal path has been essential in this context. In addition to the standard sequential causality assignment procedure SCAP, some modifications of this procedure have been proposed in the bond graph literature. In the following, first, the so-called *method of relaxed causalities* introduced by Joseph and Martens [20] is considered by means of the example of a hydraulic bridge (Figure 3.26).

In this example, the necessity appears to choose causality at two of the four resistors leading to two separate causal paths (cf. Figure 4.2) and the requirement that the resistors must have an invertible characteristic. Otherwise, causality at two resistors would not be in accordance with their characteristic. This can be taken into account by establishing implicit nonlinear equations that can only be numerically solved by iteration. In [20], Joseph and Martens propose a modification of the SCAP that enables one to assign nonlinear resistors the causality their characteristic requires and that allows violations at 0- and 1-junctions. For instance, if all resistors of the hydraulic bridge require conductance causality and if these causalities are assigned in the order R_1, R_2, R_3, R_4 , then no causal paths appear, but at the lower right side 0-junction and at the upper left side 1-junction, causality rules cannot be maintained any longer. There are two type of conflicts highlighted by a flash. At the 0-junction, there is no effort that determines the effort at the other adjacent bonds. At the 1-junction, there are two flows that want to determine the common flow at all bonds (cf. Figure 4.25). If a source requires a causality at a resistive port that is not in accordance with the resistors constitutive relation, then the method of relaxed causalities enables one to insert an extra 0- or 1-junction and to reassign causality at the resistive port resulting in a causal violation at the junction as depicted in Figure 4.26. If the causality at a resistive port is determined by the preferred integral causality of an energy store and if the resistor characteristic requires reverse

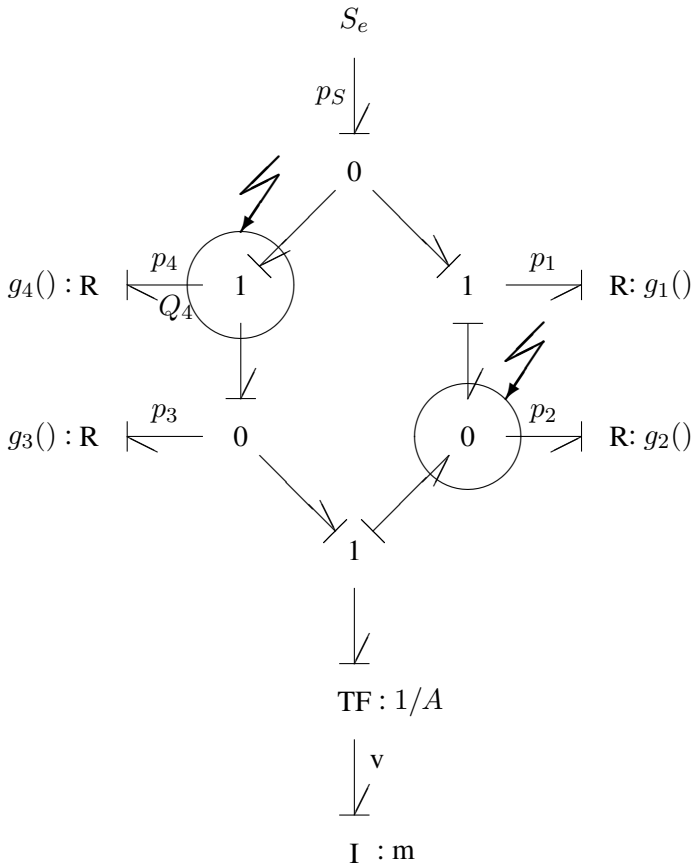


Fig. 4.25 Bond graph with two types of causal conflicts at a 0- and a 1-junction

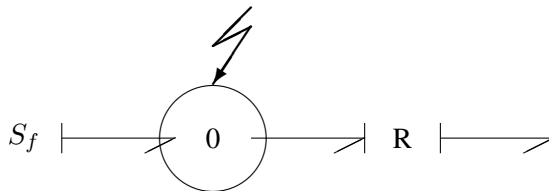


Fig. 4.26 Bond graph fragment with an extra 0-junction taking a causal violation

causality, then either the causality of the storage port can be changed into derivative causality or an extra junction can be included.

As the example of the hydraulic bridge shows, two types of causal violations at 0- and 1-junctions can be distinguished. Moreover, if several bonds want to determine

the common variable of a junction, Joseph and Martens have introduced a degree that indicates how many algebraic constraints must be established for each a junction in addition to the ODEs of the energy stores. That is, the resulting mathematical model is a DAE system.

Definition 4.18 (*Causal conflict of type 1*). In this case, there is no bond that determines the common variable of the junction. At a 1-junction, the causal stroke of all adjacent bonds point towards the 1-junction. At a 0-junction, the causal stroke at all adjacent bonds is pointing away from the junction (Figure 4.27).

Definition 4.19 (*Causal conflict of type 2 and degree k*). If there are $k + 1$ bonds that want to determine the common variable of a junction, then there is a causal conflict of type 2 and degree k . That is, at a 1-junction, not one but $k + 1$ flows are input to the junction, while at a 0-junction, $k + 1$ instead of only one single effort are input to the junction (Figure 4.28).

According to these definitions, the bond graph of Figure 4.25 shows a causal conflict of type 1 at the right side 0-junction and a causal conflict of type 2 and degree 1 at the left side 1-junction.

If there is a causal conflict of type 1 at a 0-junction, then the effort at one of its adjacent bonds must be chosen as an input variable. Preferably, a bond is chosen that connects the 0-junction to a nonlinear resistive port with conductance causality or a port of an I energy store with preferred integral causality. Joseph and Martens call such a variable an *algebraic state variable*. In the example of the hydraulic bridge (Figure 4.25), the effort p_2 is chosen and considered to be an input to the 0-junction. Adding up the flows at that junction yields an equation that determines the algebraic state variable p_2 .

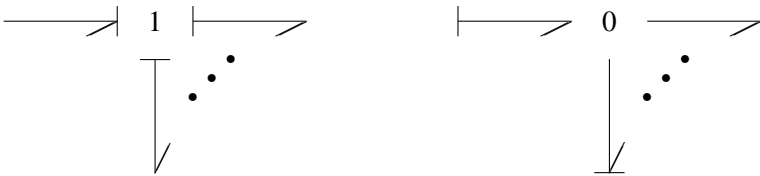


Fig. 4.27 Causal conflict of type 1



Fig. 4.28 Causal conflict of type 2 and degree k

$$0 = g_1(p_S - p_2) - g_2(p_2) + Av \quad (4.46)$$

If there is a causal violation of type 2 and degree k at a 0-junction with n adjacent bonds, and if none of the efforts of the adjacent bonds is a state variable, then an effort is arbitrarily chosen as an input to the 0-junction from those bonds that connect the junction to a nonlinear resistive port. Without loss of generality, e_1 is assumed to be that input. From the n flows at the 0-junction adding up to zero, flows f_{k+2}, \dots, f_n of those bonds causing no violation can be expressed by components of the state vector \mathbf{x} . That is, the flow f_1 into the resistive port can be represented in terms of the algebraic state variables f_2, f_3, \dots, f_{k+1} and the state vector \mathbf{x} . Hence, the constitutive equation of the resistive port takes the form

$$e_1 = \Psi_1(f_2, f_3, \dots, f_{k+1}, \mathbf{x}). \quad (4.47)$$

In addition, for each of the k bonds ($2, \dots, k+1$) that want to determine the common effort of the 0-junction like bond 1 and that are causing a violation in this way, an equation of the form

$$0 = e_i - \Psi_i(f_i), \quad i = 2, \dots, k+1 \quad (4.48)$$

holds. Since there is a common effort at all bonds of a 0-junction, Equation 4.47 can be substituted into Equation 4.48. The result is a set of k implicit algebraic equations determining the algebraic state variables f_2, \dots, f_{k+1} . Apparently, if e_1 is a state variable, then Equation 4.47 is not applicable.

Similar equations can be formulated for 1-junctions as their functionality is dual to that of 0-junctions.

In the case of a type 1 causal violation at a 1-junction, there is no flow that is input into the junction. Consequently, one flow must be chosen as an input. Effort continuity of the 1-junction provides an algebraic equation that determines the chosen input flow.

For causal violation of type 2 and degree k at a 1-junction, algebraic constraints corresponding to equations 4.47 and 4.48 are obtained for those flow variables that inflict causal violations. Consider the bond graph of Figure 4.25. From the two flows Q_4 and Q_5 that want to determine the common flow at the upper left 1-junction, Q_5 is chosen as an input. It can be represented in terms of the state variable, v , and the effort, p_4 , of that bond inflicting the causal violation.

$$Q_5 = g_3(p_S - p_4) + Av \quad (4.49)$$

For that bond causing the violation the algebraic relation

$$Q_4 = g_4(p_4) \quad (4.50)$$

holds. Since there is a common flow at a 1-junction, Equation 4.49 can be substituted into Equation 4.50. The result is an equation that determines the algebraic state variable p_4 . In this example, the method of relaxed causalities yields two coupled nonlinear implicit algebraic constraints for the algebraic state variables p_4 and p_2

and an ODE for the inertia of the piston.

$$0 = g_1 (p_S - p_2) + A v - g_2 (p_2) \quad (4.51a)$$

$$0 = g_4 (p_4) - g_3 (p_S - p_4) - A v \quad (4.51b)$$

$$\dot{v} = \frac{1}{m} A [(p_S - p_4) - p_2] \quad (4.51c)$$

Again, the resulting DAE has the form of Equations 4.4a–4.4b.

This result can also be obtained by inserting additional energy stores into the bond graph. If C energy stores are attached to the 0-junctions in the lower part of the bond graph shown in Figure 4.25, then the causal conflicts vanish. The mathematical model derived from the modified bond graph has the form of an explicit state space model. If the state equation of each additional C energy store is multiplied by its capacitance and if the latter tended to zero, then the above Equations 4.51a and 4.51b are the result.

4.10 Lagrange Causalities

In Section 4.4, the well known example of a slider crank mechanism was used to illustrate that the mathematical model takes the form of a DAE system if there is a causal path in the bond graph between an independent and a dependent storage port. Since the rod, assumed to be massless, links the piston to the flywheel, there is a geometric constraint between the piston's displacement and the angular position of the crank. Apparently, the number of unknowns in Equation 4.25 is not minimal. The mechanism has only one degree of freedom. Moreover, the balance of moments depends on the angular position of the crank. It is a peculiarity of mechanical systems in planar or 3D motion that generally equations determining their dynamic equilibrium cannot be formulated without knowing geometric positions. In contrast, power variables in electrical systems must comply with Kirchhoff's laws that do not depend on generalised displacements, viz. charge or flux linkage. Furthermore, the complexity of mathematical models of mechanical systems strongly depends on the reference frame(s) in use and on the choice of the state variables. Therefore, often appropriate coordinates, so-called *generalised coordinates*, used in mechanics and equations of motion are formulated as Lagrange equations of the second kind. This way, a compact and, due to geometric constraints, a strongly nonlinear model results with a minimal number of state variables. These are displacements and velocities of the bodies chosen with respect to appropriate reference frames. Lagrange equations, however, are not confined to mechanical systems. Generalised coordinates are only special generalised displacements, but in the context of Lagrange equations of motion, the notion of generalised coordinates is kept even when this method of establishing equations of motion is used in non-mechanical energy domains. If a Lagrange approach is applied, for instance, to describe nonlinear electrical systems, then generalised coordinates are the integral of currents with respect to time (These

quantities are not necessarily the charge of certain capacitors). Karnopp has given a procedure that, first of all, enables one to identify generalised coordinates and their associated so-called *generalised forces* in a bond graph [23]. In a second step, the Lagrangian L is determined from the bond graph as a function of the generalised coordinates and their time derivatives. Proper derivatives of the Lagrangian then, eventually, yield the Lagrange equations.

In the following, we will see that as opposed to a DAE system, Lagrange equations can be *directly* derived from a bond graph. In order to identify the needed generalised coordinates, we use the modification of the standard sequential causality assignment procedure (SCAP) given by Karnopp in [23]. Further procedures for the derivation of Lagrange equations with Lagrange multipliers directly from a bond graph have been proposed by Bos [6], van Dijk [32] and Marquis-Favre and Scavarda [25].

4.10.1 Identification of Generalised Coordinates in a Bond Graph

1. All independent sources are assigned a causality according to their type. This causal information is propagated into the junction structure as far as possible. If any causal conflicts appear, modelling assumptions must be checked and the model must be modified.
2. If the common flow variable of a 1-junction is still undetermined, then an *artificial flow source* is attached that imposes a flow. This flow information is propagated into the junction structure as far as possible. If there are no undetermined 1-junctions left and if the bond graph is still causally incomplete, then a 1-junction is inserted into an acausal bond.
3. Step 2 is repeated until all bonds have a causal stroke.

The 1-junctions to which an artificial flow source has been attached represent the time derivatives of the generalised coordinates we are looking for. There is no rule as to which undetermined 1-junction an artificial flow source has to be attached to first. Moreover, if there are no undetermined 1-junctions but still acausal bonds, then an acausal bond can be arbitrarily chosen for insertion of a 1-junction. That is, the set of generalised coordinates identified by this procedure is not unique.

4.10.2 Determination of Generalised Forces from a Bond Graph

The generalised forces are the efforts into the artificial flow sources. They are obtained by adding the efforts at the 1-junction to which an artificial flow source has been attached. However, in this sum, only efforts from sources and resistors are taken into account. In order to ensure that the generalised force is the sum of all these efforts including their sign, the half arrow points toward the artificial flow source [23]. More precisely, one should talk of artificial flow sinks.

4.10.3 Derivation of Lagrange Equations from a Bond Graph

Lagrange equations can be directly derived from a bond graph by adding up the efforts at all 1-junctions that represent the time derivative of a generalised coordinate. In contrast to the procedure applied for identification of generalised forces, in this balance, all efforts are taken into account. The sum of all efforts into an artificial flow source is equal to zero. In the following, both steps shall be applied to the example of the slider crank mechanism and to an example of an electrical network.

Example: Slider Crank Mechanism

If an artificial flow source is attached to the 1-junction representing the angular velocity $\omega = \dot{\varphi}$, and if this flow information is propagated into the junction structure, then the bond graph is already causally complete after this step (Figure 4.29). That is, the angle φ is the only generalised coordinate according to the one degree of freedom of the mechanism (If we had attached an artificial flow source to the right-hand 1-junction, the displacement of the piston would have become the generalised coordinate). From the balance of efforts at the left-hand 1-junction

$$0 = M - J\ddot{\varphi} - \tilde{M} \tag{4.52}$$

we obtain the same second order ODE for φ

$$(J + TmT)\ddot{\varphi} + (Tm\dot{T})\dot{\varphi} + (TrT)\varphi = M + TF \tag{4.53}$$

that would result after establishing and proper differentiation of the Lagrangian.

In general

$$T^* := \int_0^f p(\tilde{f}) d\tilde{f} \tag{4.54}$$

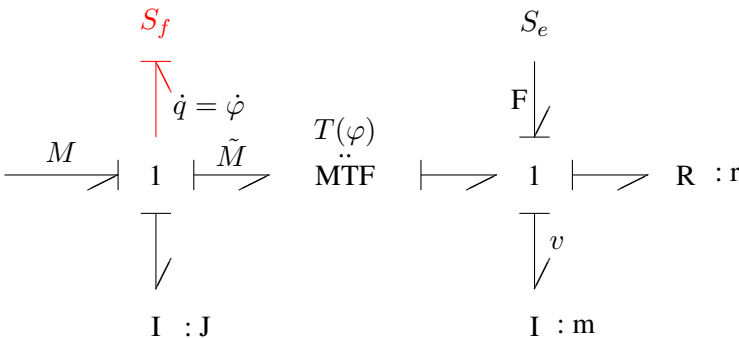


Fig. 4.29 Bond graph of the slider crank mechanism with an artificial flow source

denotes the *kinetic co-energy* and V the potential energy of a system. The Lagrangian is defined as the difference of both terms

$$L := T^* - V . \quad (4.55)$$

(The distinction between kinetic energy and kinetic co-energy is of relevance only for relativistic mechanics and for systems with non-mechanical, nonlinear inertias. Otherwise, $T^* = T$.) Let \mathbf{q} denote the vector of generalised coordinates and \mathbf{Q} the vector of generalised forces, then T^* is a function of \mathbf{q} and $\dot{\mathbf{q}}$, whereas V is a function of \mathbf{q} and

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}) - \frac{\partial L}{\partial \mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q} \quad (4.56)$$

is the Lagrange equation of motion. For the example of the slider crank mechanism, φ is the only generalised coordinate. The generalised force is $Q = M - T[r(T\dot{\varphi}) - F]$. The Lagrangian is the sum of the kinetic energy of the flywheel and of the piston.

$$L = \frac{1}{2} (J \dot{\varphi}^2 + m (T\dot{\varphi})^2) \quad (4.57)$$

If L is differentiated according to Equation 4.56, then Equation 4.53 results.

Virtual Inertia and its Companion Gyristor

Section 3.4 mentions that in the case of an independent I energy store and a dependent I energy store coupled by a transformer with a constant ratio (Figure 3.14), the energy store with derivative causality can be transformed over the transformer and can be combined with the independent energy store. Since in Equation 4.53 the factor pre-multiplying $\dot{\varphi}$ has the same form as the one in the bond graph of Figure 3.15, it is obvious to also represent Equation 4.53 by a bond graph. In the case of the slider crank mechanism, however, the modulus of the transformer is a function of the generalised coordinate φ . Consequently, the term $J + TmT$ viewed as moment of inertia, \tilde{J} , is not constant. In [3], Allan denotes \tilde{J} as an *instantaneous* or *virtual inertia*. Unlike the case of a transformer with constant modulus, the consequence of this view is that the additional term $(Tm\dot{T})\dot{\varphi}$ must be represented by a new artificial bond graph element (GR) called *gyristor* by Allan. It takes into account that the virtual inertia depends on the displacement φ and, therefore, is not constant ($Tm\dot{T} = (1/2)\dot{\tilde{J}}$). The new element ensures that the kinetic co-energy of the virtual inertia is conserved. That is, it is equal to the kinetic co-energy of the mechanism [3]. If the new artificial element is accepted, then Lagrange equation (4.53) can be represented by the bond graph of Figure 4.30.

The derivation of Lagrange equations from a bond graph is suitable especially for mechanical systems with complicated kinematics if moving reference frames are used and for nonlinear systems. The resulting second order equations of motion are strongly nonlinear. The model, however, is much more compact than a DAE system. This has been illustrated by Karnopp in an article on different approaches to the derivation of equations from a bond graph by comparing the equations for the

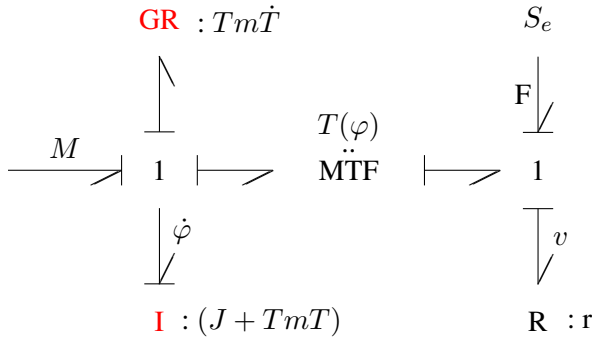


Fig. 4.30 Bond graph with virtual inertia and gyrator as companion

components of the momentum of a simple mathematical pendulum with respect to a fixed Cartesian reference frame to the well known second order equation for the angle [22].

Example: RLC Network

Finally, we will show by means of the simple RLC circuit of Figure 4.31 that the Lagrange method is not confined to mechanical systems. Application of Karnopp’s method to the bond graph of the circuit yields the two generalised coordinates \dot{q}_1 and \dot{q}_2 (Figure 4.32).

In the next step, all efforts at the 1-junctions are added up assuming that the efforts into the artificial flow sources are zero. After differentiation of the effort balance at the left-hand 1-junction, the two Lagrange equations are

$$\frac{1}{C_1} I_e = R \ddot{q}_1 + \left(\frac{1}{C_1} + \frac{1}{C_2} \right) \dot{q}_1 - \frac{1}{C_2} \dot{q}_2 \tag{4.58a}$$

$$0 = L_1 \ddot{q}_2 + \frac{1}{C_2} q_2 - \frac{1}{C_2} q_1 . \tag{4.58b}$$

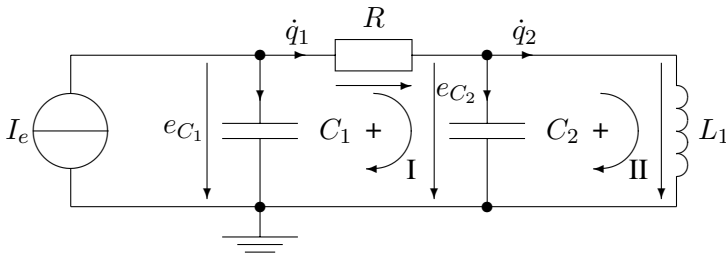


Fig. 4.31 RLC circuit

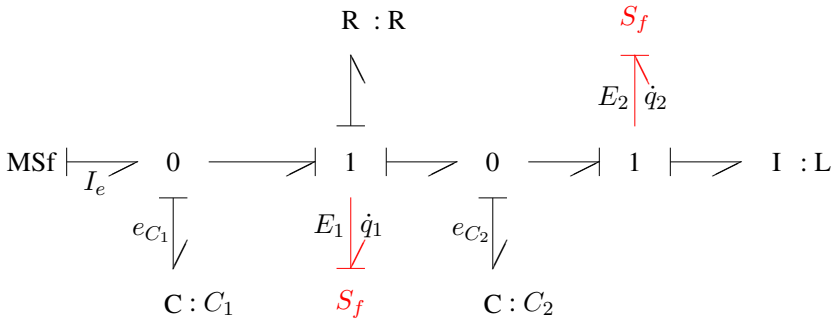


Fig. 4.32 Bond graph of the RLC circuit with artificial flow sources

Equation 4.58a can also be directly obtained from the network by adding up all voltages along mesh I, by differentiating this sum with respect to time and by using Kirchhoff’s current law to express the currents into the capacitors by the derivatives of the generalised coordinates. Likewise, the second Equation 4.58b results from adding up all voltages along mesh II.

The same equations result if the Lagrangian of the system

$$L = \frac{1}{2} L_1 \dot{q}_1^2 - \frac{1}{2C_1} \left(\int_0^t (I_e(\tau) - \dot{q}_1) d\tau \right)^2 - \frac{1}{2C_2} (q_1 - q_2)^2 \quad (4.59)$$

is differentiated according to Equation 4.56. The generalised forces

$$E_1 = -R \dot{q}_1 \quad (4.60a)$$

$$E_2 = 0 \quad (4.60b)$$

are obtained from the bond graph of Figure 4.32.

4.11 Conclusion

In the previous chapter, the important notion of a causal path was introduced. In the process of a systematic derivation of a mathematical model from a causal bond graph, first, only causal paths between resistive ports had been allowed. In this chapter, different types of causal paths and their influence on the form of the mathematical model have been considered. The following zero-order causal paths have been analysed.

1. Causal paths between resistive ports
2. Causal paths between storage ports of the same type
3. Causal loops
4. Bond graphs with different types of causal paths

In these cases, a mathematical model of the form of a DAE system can be derived from the bond graph. If this form is not supported by the available simulation program or if an explicit state space form is needed for other reasons, then causal paths can be removed by inserting additional energy stores or resistors.

In the case of *linear* algebraic constraints, one may try to remove them by symbolic manipulation supported by various algebra programs. The first approach is generally applicable and can be justified from a physics point of view by taking into account effects that are negligible with regard to the overall dynamics of a system. For instance, in multibody systems, joints linking bodies are not completely rigid, but have got some compliance and exhibit some dissipation. Consequently, inertias of bodies can be decoupled by a spring-damper pair. The disadvantage, however, is that fast transients or high frequency oscillations are introduced that are not significant for the overall system dynamics.

Transients or oscillations in a system considered to be isolated can be identified by looking at causal paths between a storage port and a resistive port and between storage ports of different type. For linear or linearised characteristics, time constants or natural frequencies can be obtained from element parameters. Since transients and oscillations in a system are not isolated but influence each other, this approach yields only a rough estimate of the actual time constants and natural frequencies of the linearised system which is not so bad because determination of eigenvalues is so costly that their evaluation prior to simulation is not worthwhile.

Inserting additional energy stores or resistors often results in a stiff ODE system. Numerical solution of such system requires an implicit stiffly stable integration method. Thus, the solution is costly with regard to computational time in comparison to an explicit method. Moreover, including additional energy stores increases the order of the ODE system. The disadvantage of a symbolic reduction approach is that it is not generally applicable.

The standard sequential causality assignment procedure, SCAP, may lead to the requirement of inverting the characteristic of some resistive ports. If that is not possible for a resistive port, causality in the bond graph is not in accordance with the form of constitutive relation. In this case, the equation of a port can be considered an implicit non-invertible algebraic relation for the determination of the output variable. A transformation of the DAE system into a state space model is not possible.

An alternative to the standard SCAP is the method of relaxed causalities introduced by Joseph and Martens. Following this procedure, nonlinear resistive ports always obtain the causality corresponding to their constitutive relation with the consequence of possible causal violations at 0- or 1-junctions. The type of causal violation indicates how many algebraic constraints must be formulated for each such junction. Hence, in general, the procedure of Joseph and Martens yields mathematical models of the form of a DAE system. Consequently, causal conflicts do not necessarily indicate contradictions in the model, meaning that the model equations cannot be solved. Whether a mathematical model can be reduced to an explicit state space model depends on whether the algebraic constraints resulting from causal violations at 0- or 1-junctions can be symbolically solved.

If there is a causal path between an independent and a dependent linear energy 1-port store, then both energy stores can be combined into one with integral causality. Transforming an energy I store with derivative causality over a transformer requires its parameter to be multiplied by the square of the transformer modulus. If the modulus is controlled by a generalised displacement, then the result is a so-called virtual inertia that has an artificial element, the gyristor, as a companion. The latter takes into account that the virtual inertia depends on the instantaneous value of the generalised displacement and ensures that the kinetic co-energy of the virtual inertia is conserved [3].

Another option is to apply a procedure introduced by Karnopp that enables one to identify generalised coordinates in a bond graph and to derive Lagrange equations for them. This can be achieved by adding all efforts at the 1-junctions representing the derivatives of generalised coordinates. The method of Lagrange equations is not confined to mechanical systems even if the terminology originating from mechanics is retained. For electrical systems, the charge of capacitors can be used as generalised coordinates. In any case, the result is a compact generally nonlinear model with a minimal number of state variables.

If the mathematical model derived from a bond graph takes the form of a DAE system, then its index is an important characteristic with regard to the numerical solution. Van Dijk has shown that conclusions with regard to the index can be drawn from inspection of causal patterns in the bond graph. For instance, if causal paths occur between resistive ports or between independent and dependent storage ports, then the DAE system for the outputs of the energy stores is of index one.

For bond graphs with causal loops (Definition 3.10), Andry and Rosenberg have shown that the algebraic equations of the general junction structure (Definition 2.14) are solvable if and only if causal loops are pairwise disjoint and if for each causal loop, its loop gain (Definition 4.16) is different from $+1$. If causal paths between independent and dependent storage ports share bonds with causal loops while causal loops do not touch, then the DAE system is of index > 1 , as has been shown by van Dijk. Consequently, solvers based on the BDF method, like the widely used DASSL code, cannot be used, in general, for such DAE systems.

Causal loops of unity loop gain in the junction structure are an indication of a weak point of the SCAP. In the bond graph of Figure 4.16, the fixed causality of the flow source and the preferred causalities of the I energy stores do not propagate into the junction structure. According to the SCAP, causality must be chosen at a resistor. If conductance causality is assigned, a causal conflict results at a 0-junction. On the contrary, assignment of resistance causality leaves the bond graph causally incomplete. Hence, causality must be chosen at an internal bond. No matter at which of the six internal bonds causality is chosen, the result is a causal loop of unity loop gain. Setting up equations reveals that the causal loop of unity loop gain as well as the causal conflict at the 0-junction indirectly express a nodicity property of the delta subnetwork of resistors. According to this nodicity property, the currents of the source and of the two I energy stores cannot be independent in contrast to the assignment of preferred causalities at the storage ports. The nodicity property can be expressed in the bond graph by assigning derivative causality to one of the

two I energy stores. As a result, the causal loop disappears. In the bond graph of Figure 4.18, the equation of cut-set 1 depicted in the network of Figure 4.15 is not expressed directly, but the bond graph is consistent with that constraint. In that light, the weakness of the SCAP could be accepted. With regard to the numerical solution of the DAE system, however, it turns out that it is of index > 1 . If one of the two I energy stores obtains derivative causality, then the DAE system is of index one, allowing for a safe numerical solution. Van Dijk has proposed a modification of the SCAP that avoids the emergence of causal loops of unity loop gain such that bond graphs causally completed this way lead to DAE systems of lower index.

Finally, internal modulation may cause algebraic loops. They are not expressed by causal paths alone. Their systematic detection is not so easy. Since internal modulation enables the representation of functional relations that do not comply with physical conservation laws, Breedveld suggests to refrain from using it.

References

- [1] DASPK. URL <http://www.cs.ucsb.edu/~cse/software.html>.
- [2] Octave. URL <http://www.gnu.org/software/octave/>.
- [3] R.R. Allen. Multiport Representation of Inertia Properties of Kinematic Mechanisms. *Journal of the Franklin Institute*, 308(3):235–253, 1979.
- [4] W. Borutzky. Kausalitäten in Bond-Graphen und Formen mathematischer Systemmodelle – Eine bewertende Untersuchung. In D. Tavangarian, editor, *Simulationstechnik, 7. Symposium in Hagen, September 1991*, pages 505–509. Vieweg Verlag, 1991.
- [5] W. Borutzky. On Interrelations Between Bond Graph Causal Patterns and the Numerical Solution of the Mathematical Model. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 93–100. SCS Publishing, January 17–20 1993. Simulation Series, volume 25, No. 2, ISBN: 1-56555-019-6.
- [6] A.M. Bos. *Modelling Multibody Systems in Terms of Multibond Graphs with Application to a Motorcycle*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1986.
- [7] P.C. Breedveld. *Physical Systems Theory in Terms of Bond Graphs*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1984.
- [8] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.
- [9] A.P.J. Breunese. *Automated support in mechatronic systems modeling*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1996.
- [10] F.T. Brown. Direct Application of the Loop Rule to Bond Graphs. *Journal of Dynamic Systems, Measurement and Control*, pages 253–261, September 1992.
- [11] S.L. Campbell, J.Ph. Chancelier, and R. Nikoukhah. *Modeling and Simulation in Scilab/Scicos*. Springer Science+Business Media, New York, NY, U.S.A., 2006.
- [12] A. Cornet and F. Lorenz. Equation Ordering Using Bond Graph Causality Analysis. In P.C. Breedveld et al., editor, *Modelling and Simulation of Systems*, pages 55–58. J.C. Baltzer AG, Scientific Publishing Co., 1989.
- [13] P.J. Gawthrop and L. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [14] C. W. Gear. Differential-algebraic equation index transformations. *SIAM J. Sci. Stat. Comput.*, 9(1):39–47, 1988.

- [15] C.W. Gear and L.R. Petzold. Ode methods for the solution of differential/algebraic systems. *SIAM J. Numer. Anal.*, 21:367–384, 1984.
- [16] E. Griepentrog and R. März. *Differential-Algebraic Equations and Their Numerical Treatment*, volume 88. Teubner, 1986.
- [17] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*. Springer-Verlag, 2nd edition, 1996.
- [18] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Number 1409 in Lecture Notes in Mathematics. Springer-Verlag, 1989.
- [19] N. Hogan and E.D. Fasse. Conservation principles and bond graph junction structures. In *Proc. ASME Winter Annual Meeting, Automated Modeling for Design*, volume DSC-8, pages 9–13. ASME, 1989. New York.
- [20] B.J. Joseph and H.R. Martens. The Method of Relaxed Causality in the Bond Graph Analysis of Nonlinear Systems. *Trans. ASME Journal of Dynamic Systems, Measurement, and Control*, 96:95–99, 1974.
- [21] Th. Kailath. *Linear Systems*. Englewood Cliffs, N. J., USA, 1980.
- [22] D.C. Karnopp. Alternative Bond Graph Causal Patterns and Equation Formulations for Dynamic Systems. *Journal of Dynamic Systems, Measurement, and Control*, 105:58–63, 1983.
- [23] D.C. Karnopp. Lagrange's Equations for Complex Bond Graph Systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 99(4):300–306, December 1977.
- [24] D.G. Luenberger. Dynamic equations in descriptor form. *IEEE Transactions Automatic Control*, 22:312–321, 1977.
- [25] W. Marquis-Favre and S. Scavarda. A procedure for generating the Lagrange equations from the bond graph representation using the λ -multiplier method. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 1999 International Conference on Bond Graph Modeling and Simulation*, volume 31 (1), pages 263–268, 1999.
- [26] J.R. Ort and H.R. Martens. The Properties of Bond Graph Junction Structure Matrices. *Journal of Dynamic Systems, Measurement, and Control*, 95:362–367, 1973.
- [27] L. Petzold. Differential/algebraic equations are not ode's. *SIAM J. SCI STAT COMPUT*, 3(3): 367–384, 1982.
- [28] R.C. Rosenberg and A.N. Andry. Solvability of Bond Graph Junction Structures with Loops. *IEEE Trans. on Circuits and Systems*, CAS-26(2):130–137, February 1979.
- [29] Scilab Consortium. Scilab. URL <http://www.scilab.org/>.
- [30] S.Li and L. Petzold. Design of New DASPCK for Sensitivity Analysis. Technical report, UCSB, 1999. URL <http://www.engineering.ucsb.edu/~cse/>.
- [31] J.U. Thoma. *Introduction to Bond Graphs and their Applications*. Pergamon Press, Oxford, 1975.
- [32] J. van Dijk. *On the role of bond graph causality in modelling mechatronic systems*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1994.
- [33] J. Won and N. Hogan. Nodic and Non-Nodic Structures in Physical Systems. In J.J. Granda and F.E. Cellier, editors, *ICBGM'99, 4th International Conference on Bond Graph Modeling and Simulation*, pages 90–95. SCS Publishing, 1999. Simulation Series, volume 31, No. 1, ISBN: 1-56555-155-9.

Chapter 5

Computing Mathematical Models Derived from Bond Graphs

In the previous chapter, we analysed causally completed bond graphs with regard to the form of the mathematical models that can be systematically derived. It has turned out that, in general, a DAE system results. An important indication for a robust numerical solution of such systems is their index. For that reason, the index has been determined for some bond graphs with causal paths from different classes and general results, proven by van Dijk [49], have been cited. DAE systems derived from bond graphs are often of index ≤ 2 , giving reason to the conjecture that this holds in general. However, there is no formal proof thus far.

For a comprehensive presentation of bond graph based physical systems modelling and simulation, in the following, the principle of some approaches to the symbolic and numerical solution of DAE systems and their features are dealt with. For details and mathematical proofs, readers are referred to publications in the area of numerical mathematics. First, the principle and features of the widely used numerical approach based on the so-called *Backward Differentiation Formula* (BDF) are considered. Subsequently, we will discuss how symbolic preprocessing of the DAE system can be supported at bond graph level. The idea is to reduce the index or to solve the system of algebraic constraints symbolically, if possible, before the DAE system is passed on to a numerical solver.

5.1 Numerical Solution of Differential-Algebraic Systems

5.1.1 The Backward Differentiation Formula

Linear stiffly stable multi-step methods have been used for the solution of systems of implicit stiff ODEs for a long time, e.g., for the transient analysis of integrated circuits. The idea of using them also for DAE systems goes back to Gear [24]. In a general implicit DAE,

$$\mathbf{F}(\mathbf{x}(t), \dot{\mathbf{x}}(t), t) = \mathbf{0}, \quad (5.1)$$

the derivative $\dot{\mathbf{x}}$ at the future time t_{n+1} is replaced by an approximation $\dot{\mathbf{x}}_{n+1}$ that is a weighted sum of the unknown approximate value \mathbf{x}_{n+1} at time t_{n+1} and k known approximate past values \mathbf{x}_{n+1-i} at times t_{n+1-i} ($i = 1 \dots k$)

$$\dot{\mathbf{x}}_{n+1} = -\frac{\alpha_0}{h_{n+1}}\mathbf{x}_{n+1} - \frac{1}{h_{n+1}} \sum_{i=1}^k \alpha_i \mathbf{x}_{n+1-i}, \quad (5.2)$$

where $h_{n+1} := t_{n+1} - t_n$ is the present time step and $\alpha_0, \alpha_1, \dots, \alpha_k$ are coefficients. Equation 5.2 is called the *Backward Differentiation Formula* (BDF) of order k . It transforms the initial DAE into a nonlinear set of algebraic equations for the unknown approximate vector \mathbf{x}_{n+1} at time t_{n+1} .

$$\mathbf{G}(\mathbf{x}_{n+1}, t_{n+1}) = \mathbf{0} \quad (5.3)$$

The BDF of order k gives the *exact* value $\mathbf{x}(t_{n+1})$ if the solution of the DAE (5.1) is a polynomial of order k . This requirement yields a linear algebraic system determining the coefficients α_i ($i = 0 \dots k$)

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & a_1^1 & \dots & a_k^1 \\ 0 & a_1^2 & \dots & a_k^2 \\ \vdots & \vdots & & \vdots \\ 0 & a_1^k & \dots & a_k^k \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (5.4)$$

where $a_i^j := \left[\frac{t_{n+1} - t_{n+1-i}}{h_{n+1}} \right]^j$, $i = 1, \dots, k$, $j = 1, \dots, k$. Note that these coefficients must be re-evaluated whenever the current time step h_{n+1} changes.

For order $k = 1$, for instance, the determination of the coefficients yields the well known implicit backward Euler method (BE) and Equation 5.3 takes the form

$$\mathbf{F}(\mathbf{x}_{n+1}, \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{h_{n+1}}, t_{n+1}) = \mathbf{0}. \quad (5.5)$$

The nonlinear set of algebraic equations (5.3) is solved by iteration. Usually, Newton-Raphson (NR) iteration is used. Let $\mathbf{J}_{\mathbf{G}}$ denote the Jacobian matrix of the function \mathbf{G} and $\mathbf{x}_{n+1}^{(m)}$ the m^{th} iteration of the unknown value \mathbf{x}_{n+1} . Then, the Newton-Raphson iteration is

$$\mathbf{J}_{\mathbf{G}}(\mathbf{x}_{n+1}^{(m)}) [\mathbf{x}_{n+1}^{(m+1)} - \mathbf{x}_{n+1}^{(m)}] = -\mathbf{G}(\mathbf{x}_{n+1}^{(m)}, t_{n+1}). \quad (5.6)$$

Let $\mathbf{x}^P(t)$ be the unique k^{th} degree interpolation polynomial that passes through the $k + 1$ past approximate values \mathbf{x}_{n+1-i} , ($i = 1, \dots, k + 1$). Then, $\mathbf{x}_{n+1}^P := \mathbf{x}^P(t_{n+1}) \neq \mathbf{x}_{n+1}$ is a predicted value of the unknown approximate value \mathbf{x}_{n+1} ,

$$\mathbf{x}_{n+1}^P = \sum_{i=1}^{k+1} \gamma_i \mathbf{x}_{n+1-i}, \quad (5.7)$$

that can be used as an initial guess for the Newton-Raphson iteration ($\mathbf{x}_{n+1}^{(0)} = \mathbf{x}_{n+1}^P$). The requirement that Equation 5.7 gives the exact value $\mathbf{x}_{n+1}^P = \mathbf{x}^P(t_{n+1})$ at t_{n+1} for all polynomials of degree $i = 1, \dots, k$ yields the set of linear equations (5.8) determining the coefficients γ_i ($i = 1, \dots, k+1$) [14].

$$\begin{bmatrix} 1 & \dots & 1 & 1 \\ a_1^1 & \dots & a_k^1 & a_{k+1}^1 \\ a_1^2 & \dots & a_k^2 & a_{k+1}^2 \\ \vdots & & \vdots & \vdots \\ a_1^k & \dots & a_k^k & a_{k+1}^k \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5.8)$$

The matrix in Equation 5.8 is called a *Vandermonde* matrix. Its first k columns are identical to columns 2, 3, \dots , $k+1$ of the matrix in Equation 5.4. That is, the coefficients $\alpha_0, \alpha_1, \dots, \alpha_k$ and $\gamma_1, \gamma_2, \dots, \gamma_{k+1}$ can be related. Now, if the step size changes, both sets of coefficients must be re-evaluated. However, as has been shown by Brayton, Gustavson and Hachtel in [7], the new coefficients can be efficiently derived from the ones for the old step sizes.

The computational effort of the BDF method can be considerably reduced if both the step size and the order of the method are controlled [7]. Software implementations of the BDF method control the step size as well as the order k in such a way that the absolute value of the local truncation error (LTE) remains below a given bound. For a method of order k , the local truncation error is approximately

$$E_{k,j} \approx \frac{h_{n+1}}{t_{n+1} - t_{n-k}} (x_{n+1,j} - x_{n+1,j}^P), \quad (5.9)$$

where the second subscript j denotes the component of the vectors ([7], Equation 5b, [14]). The possibility of varying the step size, in principle, makes the BDF methods suitable for models with discontinuities.

5.1.2 Problems with the Numerical Solution of DAEs by Means of the BDF

If the step size is varied, then the coefficients α_i in Equation 5.2 and, in particular, the first coefficient α_0 change. This would entail a repetitive LU decomposition of the Jacobian matrix. To avoid this effort, Brenan, Campbell and Petzold use a special form of the BDF method in their DASSL code in which the coefficient α_0 is independent of the step size [8]. This method is known as *fixed leading coefficient BDF*. Variation of the step size may lead to an instability of the method [7, 22]. Moreover, a reduction of the step size increases the condition number of the Jacobian matrix in

Equation 5.6. If the BDF, Equation 5.2, is rewritten in a simplified notation as

$$\dot{\mathbf{x}} = -\frac{\alpha_0}{h}\mathbf{x} + \frac{1}{h}\mathbf{c}, \quad (5.10)$$

where $\mathbf{c} := -\sum_{i=1}^k \alpha_i \mathbf{x}_{n+1-i}$ is a constant determined by the approximate values \mathbf{x}_{n+1-i} at the past $k+1$ time points, then the Jacobian matrix of \mathbf{G} reads

$$\mathbf{J}_{\mathbf{G}}(\mathbf{x}) = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \left(-\frac{\alpha_0}{h} \right) \right) (\mathbf{x}). \quad (5.11)$$

Substituting Equation 5.11 into Equation 5.6, multiplying the resulting equation by h and letting $h \rightarrow 0$ yields

$$-\alpha_0 \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}}(\mathbf{x}_{n+1}^{(m)}) [\mathbf{x}_{n+1}^{(m+1)} - \mathbf{x}_{n+1}^{(m)}] = \mathbf{0}. \quad (5.12)$$

For a DAE system of index one, the matrix $\partial \mathbf{F} / \partial \dot{\mathbf{x}}$ is singular. This problem does not appear if something is known about the structure of the set of equations such that the vector \mathbf{x} can be partitioned into a state vector and a vector of semi-state variables. From the previous chapter, we know that a DAE system derived from a bond graph with causal paths between resistive ports or causal loops of loop gain different from one is of semi-explicit form

$$\dot{\mathbf{x}} = \mathbf{f}_1(\mathbf{x}, \mathbf{z}, t) \quad (5.13a)$$

$$\mathbf{0} = \mathbf{f}_2(\mathbf{x}, \mathbf{z}, t), \quad (5.13b)$$

where \mathbf{x} denotes the vector of state variables and \mathbf{z} the vector of semi-state variables. Now, we substitute the derivative $\dot{\mathbf{x}}$ in Equation 5.13a by means of the BDF, Equation 5.10, apply Newton-Raphson formula, multiply the resulting equations by h and let $h \rightarrow 0$. Then, the result is the following equation to be solved for each iteration step

$$\begin{bmatrix} -\alpha_0 \mathbf{I} & \mathbf{0} \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{z}} \end{bmatrix} (\mathbf{x}_{n+1}^{(m)}, \mathbf{z}_{n+1}^{(m)}) \begin{bmatrix} \Delta \mathbf{x}_{n+1}^{(m+1)} \\ \Delta \mathbf{z}_{n+1}^{(m+1)} \end{bmatrix} = - \begin{bmatrix} -\alpha_0 \mathbf{x}_{n+1} \\ \mathbf{f}_2(\mathbf{x}_{n+1}^{(m)}, \mathbf{z}_{n+1}^{(m)}, t_{n+1}) \end{bmatrix}. \quad (5.14)$$

Clearly, Equation 5.14 is solvable if the sub-matrix $\partial \mathbf{f}_2 / \partial \mathbf{z}$ is non-singular. This condition is necessary and sufficient for the local index of the DAE system 5.13a)–(5.13b) to be equal to one. With regard to a practical check, this mathematical condition is less appropriate than a criterion based on the inspection of a causally completed bond graph. If the local index of an underlying semi-explicit DAE system is one, then this means that in each step of the Newton-Raphson iteration, the set of linear algebraic equations can be safely solved even for small step sizes h if the Newton-Raphson equation is scaled by h first.

If the sub-matrix $\partial \mathbf{f}_2 / \partial \mathbf{z}$ is singular, then the DAE is of local index > 1 . In that case, the matrix in the recurrence equation, Equation 5.14, is singular.

The condition number of the iteration matrix

$$\mathbf{J}_{\mathbf{G}} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} - \frac{\alpha_0}{h} \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \quad (5.15)$$

in Equation 5.6, in general, is of order $O(h^{-\nu})$ for a DAE system of local index ν ([8], Theorem 5.4.1). That is, for DAE systems of local index > 1 , small step sizes can result in large roundoff errors in the solution of the set of linear equations in each Newton-Raphson iteration step. As a consequence, the NR iteration may fail to converge. Small step sizes may occur especially at the beginning of the numerical integration and in the vicinity of discontinuities. Rounding errors, however, are not of the same order for all unknowns. They are larger for the algebraic unknowns than for the state variables. Lötstedt and Petzold [33] state that for the semi-explicit system (5.13a)–(5.13b), the roundoff error in \mathbf{x} is of the order $O(u)$ and for \mathbf{z} of the order $O(u/h)$ (u denotes the machine unit) if the row in the Jacobian matrix $h\mathbf{J}_{\mathbf{G}}$, according to the algebraic constraints, is scaled by $1/h$. Since the algebraic variables do not influence the dynamic behaviour beyond the actual time, convergence problems of the Newton-Raphson iteration in the case of semi-explicit DAE systems of local index > 1 can be reduced by tolerating much larger errors in the algebraic variables than in the state variables and by excluding them from error control decisions.

In addition to the problem of an increased condition number of the iteration matrix, Equation 5.11, a reduction of the step size h entails further problems in the numerical solution of DAE systems of higher index [8]. In his dissertation, Bujakiewicz [10] gives a formula for the error $e_n := \mathbf{x}_n - \mathbf{x}(t_n)$ in the solution of Equation 5.1 obtained by the BDF (Equation 5.2). In that formula, inclusion of the local truncation error of the BDF as well as the error in the solution of Equation 5.3 by iteration is of crucial importance. For a DAE system of index m , the inverse of the iteration matrix includes powers of $(1/h)$ up to the order $m - 1$. According to the formula of Bujakiewicz, this entails that the local truncation error of the BDF method is amplified by powers of $(1/h)$. The same holds for the errors that have emerged at k past times in the iteration of Equation 5.3. By means of an example of a linear DAE system of index 3, Bujakiewicz shows that for this reason, the local error in some variables cannot be diminished by reducing the step size [10].

In summary, the numerical solution of higher index DAE systems by means of the BDF method entails the following problems that complicate the step size control.

- The local truncation error is amplified by powers of $(1/h_n)$.
- The algebraic error, due to a limited number of Newton-Raphson iterations and due to roundoff errors in the solution of the Newton-Raphson equation, is amplified by powers of $(1/h_n)$.
- The condition number of the iteration matrix is increased by a reduction of the step size.

For nonlinear *semi-explicit* DAEs of index 2

$$\mathbf{F}_1(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{z}, t) = \mathbf{0} \quad (5.16a)$$

$$\mathbf{F}_2(\mathbf{x}, \mathbf{z}, t) = \mathbf{0}, \quad (5.16b)$$

where $\det \partial \mathbf{F}_1 / \partial \dot{\mathbf{x}} \neq 0$ and $\partial \mathbf{F}_2 / \partial \mathbf{z}$ has constant rank, Lötstedt and Petzold have shown in [34] that the k -step BDF ($k < 7$) is convergent and globally accurate to $O(h^k)$ after $k + 1$ steps if all initial values are accurate to $O(h^k)$ and if the result of the Newton-Raphson iteration at each time is accurate to $O(h^{k+1})$ [8].

Finally, it is recalled that for a fixed step size, the BDF method is identical with Gear's method. As it is well known, the BDF method is stiffly-stable. That is, in the vicinity of the imaginary axis of the complex ($h\lambda$)-plane, $\lambda \in \mathbb{C}$, stability is not ensured. Consequently, the BDF method is not well suited for the solution of oscillatory models if the linearised model includes weakly damped oscillatory components of eigenvalues $\lambda_i(t)$. Hence, for a given order and some step sizes h , the points $h\lambda_i(t)$ close to the imaginary axis may be outside the region of absolute stability (cf. [25], Figure 11.7).

5.2 Reduction of the Index of a Differential-Algebraic System

The previous considerations make it clear that, in general, a reliable numerical solution of DAEs by means of the BDF method is possible if they are of index < 2 . Index 2 DAE systems can be solved using the BDF method if they are of a special form, e.g., semi-explicit DAEs ([9], Theorem 3.2.2). However, as to the software implementation of the BDF method in the widely used DASSL code, error estimation has not been designed for the class of semi-explicit DAEs. The numerical solution of index 2 DAE systems has been subject of research, e.g., by März and Tischendorf [37, 38] and by Tischendorf and her group [28, 45]. Furthermore, Lötstedt and Petzold have shown that the BDF method also works for some special cases of index 3 DAEs [34].

A possible alternative to the BDF method are *implicit* Runge-Kutta methods. In contrast to linear multi-step methods, A-stable methods of high order can be constructed. As one-step methods, they are self-starting, they enable a simple step size control and are suited for models with frequent discontinuities. Whereas multi-step methods must restart at low order after a discontinuity has occurred, implicit Runge-Kutta methods can restart at higher order. The low order of the BDF at a restart is a problem in the solution of index 3 systems. However, the computational effort per step is substantially higher than for multi-step methods which has lead van der Houwen and Sommeijer [43] to the development of parallel implicit Runge-Kutta methods.

Hairer, Lubich and Roche [27] investigate the numerical solution of DAE systems by means of Runge-Kutta methods and describe a code called RADAU5 based on the three-stage RADAU IIA method that enables one to solve linear implicit systems

$$\mathbf{A}\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \quad (5.17)$$

of index 1,2 or 3 where \mathbf{A} is a square constant coefficient matrix.

The direct numerical solution of higher index DAE systems, in general, however, is difficult [8]. In [22], Gear points out:

We do not know how to solve problems of higher index numerically without resorting to symbolic manipulations of the equations. One such approach, suggested by Campbell [10], consists of performing the differentiation shown in (1.2), solving for y' ...

where differentiation means the differentiation to be carried out for determining the differential index of a nonlinear DAE (see Definition 4.11).

It is true that DAEs derived from bond graphs are often of index 1, but DAEs of index > 1 may also result for instance if a causal paths between an independent and a dependent storage port and a causal loop share a bond, as in the bond graph of Figure 4.14. In view of the problems that can occur in solving numerically higher index DAEs by means of the BDF method, it is obvious to try to reduce the index of a DAE system by symbolic differentiation and manipulation of the equations, as suggested by Gear, before passing it on to a numerical solver. This approach, however, entails another two problems, namely the determination of consistent initial values and the phenomenon of numerical drift.

Example: Mathematical Pendulum

For illustration of both problems, we will consider the classical example of a mathematical pendulum of point mass m and a massless bar of length L (Figure 5.1).

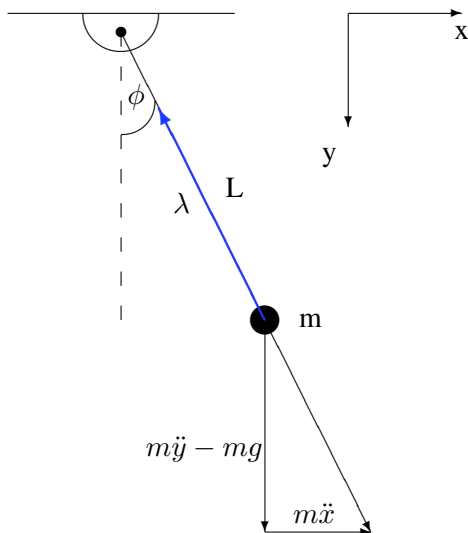


Fig. 5.1 Mathematical pendulum

Given a fixed Cartesian reference frame, let u denote the velocity of the point mass in x - and v its velocity in y -direction and λ the tension in the bar. Then, the motion of the point mass can be described by Equations 5.18a–5.18e.

$$\dot{x} = u \quad (5.18a)$$

$$\dot{y} = v \quad (5.18b)$$

$$\dot{u} = \frac{1}{mL} \lambda x \quad (5.18c)$$

$$\dot{v} = \frac{1}{mL} \lambda y + g \quad (5.18d)$$

$$L^2 = x^2 + y^2 \quad (5.18e)$$

The equations of motion can be derived either directly from the drawing of Figure 5.1, or by transforming the Euler-Lagrange equations of a pendulum into first order ODEs. Equation 5.18e is a constraint at the geometric level for the position of the point mass.

If we want to reduce the DAE system 5.18a)–(5.18e into an explicit ODE system, then we need to express the derivative of the tension, $\dot{\lambda}$, as a function of the variables x, y, u, v, λ . Differentiation of Equations 5.18c–5.18d yields two new equations which include $\dot{\lambda}$, but also two new variables \ddot{u} and \ddot{v} . After substitution of \dot{x} and \dot{y} we get

$$\ddot{u} = \frac{1}{mL} (\dot{\lambda}x + \lambda u) \quad (5.19a)$$

$$\ddot{v} = \frac{1}{mL} (\dot{\lambda}y + \lambda v) . \quad (5.19b)$$

By differentiation of the constraint at positional level and substitution of \dot{x} and \dot{y} , we get a constraint for the velocity vector $[u, v]^T$ of the mass-point

$$0 = xu + yv . \quad (5.20)$$

Equation 5.20 expresses that the velocity vector and the position vector of the point mass are perpendicular to each other. From the velocity constraint, we see that we need to differentiate it another two times and that we need to differentiate Equations 5.18a and 5.18b one more time to get rid of \ddot{x} and \ddot{y} in order to finally express $\dot{\lambda}$ by the variables x, y, u, v, λ .

$$0 = \ddot{x}u + \ddot{y}v + 2(\dot{x}\dot{u} + \dot{y}\dot{v}) + x\ddot{u} + y\ddot{v} \quad (5.21a)$$

$$\ddot{x} = \dot{u} \quad (5.21b)$$

$$\ddot{y} = \dot{v} \quad (5.21c)$$

Actually, by substitution of Equations 5.21b and 5.21c as well as Equations 5.19a and 5.19b into Equation 5.21a and by observing Equations 5.18e and 5.20, we get

$$\dot{\lambda} = -\frac{3mg}{L}v. \quad (5.22)$$

Thus, the DAE system is of index 3. In order to find this result, it was not necessary to differentiate all equations three times. This raises the questions regarding which equations need to be differentiated how many times in order to express the rate of the state vector as a continuous function of the state vector and of time.

Closely related with the reduction of a DAE system to an ODE system is the problem of consistent initial values for the DAE system. As can be seen from Equation 5.20, it is not sufficient to give initial values for the initial DAE system. The initial values rather must satisfy all additional equations that result from differentiation. Hence, repeated differentiation of a part of the DAE system is not only needed to determine the index, but also to determine a set of consistent initial values. In [41], Pantelides gives a graph-based algorithm that determines a minimal number of equations from which a set of consistent initial values can be computed. Since the differentiation of equations is used in the reduction of the index and since differentiation can be symbolically performed, the algorithm of Pantelides has been implemented in the program *Dymola* [19] for *symbolic* reduction of a higher index DAE systems (cf. Chapter 11).

Under the assumption that the occurrence of a variable in an equation is independent of the values of the parameters in the equation, the above steps for determination of $\dot{\lambda}$ can be systemised in the following manner. For the DAE system 5.1 in which $\dot{\mathbf{x}}$ denotes the vector of unknowns, a so-called *occurrence matrix* \mathbf{S} [35] can be set up. Their rows corresponds to the equations and their columns to the unknowns. If the j^{th} unknown occurs in the i^{th} equation, then the entry in position (i, j) of \mathbf{S} is set to 1. Otherwise, it is zero. Hence, it is a binary matrix that accounts for the *structural* relationship between variables and equations. It does not reflect the functional form of equations (The occurrence matrix \mathbf{S} reflects the structure of the Jacobian matrix $\partial\mathbf{F}/\partial\dot{\mathbf{x}}$ of Equation 5.1).

For Equations 5.18a–5.18e and the vector $\dot{\mathbf{x}}$ of unknowns, $\dot{\mathbf{x}} = [\dot{x} \ \dot{y} \ \dot{u} \ \dot{v} \ \dot{\lambda}]^T$, we get the occurrence matrix of Equation 5.23

$$\mathbf{S}_0 = \begin{array}{l} (5.18a) \\ (5.18b) \\ (5.18c) \\ (5.18d) \\ (5.18e) \end{array} \begin{array}{c} \dot{x} \ \dot{y} \ \dot{u} \ \dot{v} \ \dot{\lambda} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}. \quad (5.23)$$

There are no unknowns in Equation 5.18e. It rather establishes a relation between some components of the known vector \mathbf{x} . For that reason, the matrix \mathbf{S}_0 is singular. That is, the set of equations cannot be solved for $\dot{\mathbf{x}}$, the vector of unknowns. Therefore, the DAE system is of index > 1 . If Equation 5.18e is differentiated with respect to time and replaced by Equation 5.20, then the occurrence matrix \mathbf{S}_1 of the modified set of equations is still singular. Equation 5.20 also leads to a row in the

occurrence matrix with vanishing entries. If we differentiate Equations 5.18a–5.18d as well as Equation 5.20 and add the differentiated equations to the initial Equations 5.18a–5.18d, then we see that the algebraic constraint differentiated two times has lead to a vanishing entry on the diagonal of the occurrence matrix S_2 transformed to lower triangular form. Hence, also S_2 is singular. Let \tilde{S}_2 denote the occurrence matrix S_2 transformed to lower triangular form. Then,

$$\begin{aligned}
 & \text{(5.18a)} \\
 & \text{(5.18b)} \\
 & \text{(5.18c)} \\
 & \text{(5.18d)} \\
 & \text{(5.21b)} \\
 & \text{(5.21c)} \\
 & \text{(5.20)} \\
 & \text{(5.19b)} \\
 & \text{(5.19a)}
 \end{aligned}
 \tilde{S}_2 = \begin{pmatrix}
 \dot{x} & \dot{y} & \dot{u} & \dot{v} & \ddot{x} & \ddot{y} & \dot{\lambda} & \ddot{v} & \ddot{u} \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 0 & 0 & \boxed{0} & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1
 \end{pmatrix}. \tag{5.24}$$

If the algebraic constraint differentiated two times is differentiated a third time, then the occurrence matrix of the resulting set of equations can be transformed to lower block-triangular form with a structural non-singular block matrix. Thus, the set of equations is solvable and the reduction process stops after three steps. In each step, only those equations are differentiated that lead to a singular occurrence matrix. Apparently, occurrence matrices can be represented by bipartite graphs. The algorithm of Pantelides [41] for the identification of a minimal sub-set of equations to be differentiated for the determination of a set of consistent initial conditions works on bipartite graphs.

The second problem related with the reduction of the index of a DAE system mentioned at the beginning of this section is that of numerical drift. Let us consider the example of a mathematical pendulum once again. The formulation by means of the Equations 5.18a–5.18e includes a constraint for the position of the point mass. In the process of transforming the DAE system into an ODE system, the constraint on geometric level is differentiated. Thus, this information vanishes. Due to numerical imprecision in the numerical integration of the ODEs, the solution of the reduced system only approximates the geometric constraint. Consequently, if the pendulum with no friction in the pivot point starts swinging from a horizontal position, in the long run after many swings, it will not return to its horizontal starting position due to an increasing influence of numerical errors [20, 50]. Differentiation of the geometric constraint yields a constraint for the velocity (Equation 5.20. Its differentiation gives another constraint for the forces acting on the point mass (Equation 5.21a. If only the explicit ODE system resulting from the reduction process is used, then the constraints at geometric and at velocity level are not existent in the model and cannot be taken into account in the numerical solution. The problems with the direct numerical solution of higher index DAE systems and that of numerical drift suggest to reduce

a higher index system to one of index 1 and to add the initial constraints for stabilisation of the numerical solution. This yields an over-determined set of equations for the Newton-Raphson iteration. To overcome this problem, Führer has combined the Newton-Raphson iteration with Gauß' least square method. This approach has been implemented in the solver ODASSL [20]. Another option is to consider the derivatives of some variables as new *independent* variables such that the number of the unknowns corresponds with the number of the equations increased by the number of the differentiated equations that have been added. The derivatives considered to be new independent variables are called *dummy derivatives*. However, with this approach, the problem arises to decide which variables are state variables and which are not.

Finally, we will show that a bond graph model can be easily constructed for the mathematical pendulum that represents the DAE system of index 2. That is, the velocity constraint is taken into account, but not the geometric constraint. Since bond graphs represent relations between power variables, it is not possible to express geometric constraints.

In describing the planar or 3D motion of mechanical systems, it is typical to use several reference frames and transformations between them. In bond graphs, these transformations between reference frames can be represented by displacement modulated transformers. In particular, differentiation of the relations between polar coordinates (r, ϕ) and Cartesian coordinates x, y yields two relations between the corresponding velocities

$$\begin{bmatrix} \dot{\phi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \cos \phi & -\frac{\sin \phi}{r} \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}. \quad (5.25)$$

Moreover, let $\mathbf{e}_1, \mathbf{f}_1, \mathbf{e}_2, \mathbf{f}_2$ be vectors of efforts or flows and \mathbf{A} a non-singular matrix relating the vectors of flows

$$\mathbf{f}_2 = \mathbf{A} \mathbf{f}_1. \quad (5.26)$$

Then, power continuity

$$\mathbf{e}_1^T \mathbf{f}_1 = \mathbf{e}_2^T \mathbf{f}_2 \quad (5.27)$$

yields for the corresponding vectors of efforts

$$\mathbf{e}_1 = \mathbf{A}^T \mathbf{e}_2. \quad (5.28)$$

In the case of the mathematical pendulum, this relation takes the form

$$\begin{bmatrix} m \ddot{x} \\ m \ddot{y} - m g \end{bmatrix} = \begin{bmatrix} \frac{\cos \phi}{r} & \sin \phi \\ -\frac{\sin \phi}{r} & \cos \phi \end{bmatrix} \begin{bmatrix} M \\ \lambda \end{bmatrix}, \quad (5.29)$$

where M denotes the moment in the pivot point and λ , the tension in the bar. Since $\dot{r} \equiv 0$ and $M \equiv 0$, Equations 5.25 and 5.29 simplify to

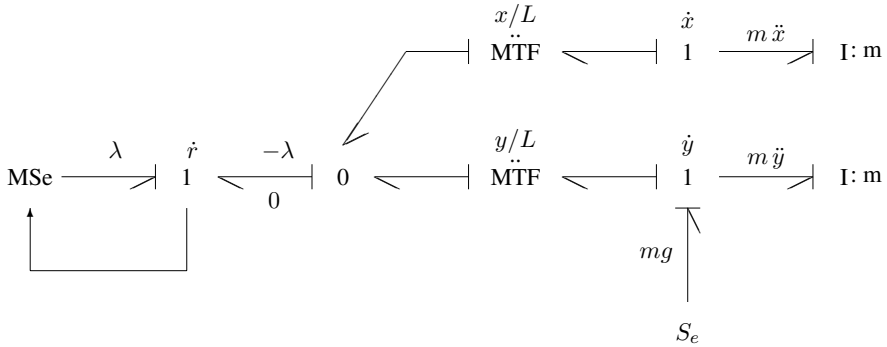


Fig. 5.2 Bond graph of the planar motion of a mathematical Pendulum using the constraint force λ

$$0 = \dot{x} \sin \phi + \dot{y} \cos \phi \tag{5.30a}$$

$$m \ddot{x} = \lambda \sin \phi \tag{5.30b}$$

$$m \ddot{y} = \lambda \cos \phi + m g . \tag{5.30c}$$

Substitution of $\sin \phi = x/L$ and $\cos \phi = y/L$ just yields the DAE system in which the geometric constraint has been replaced by that for the velocities. The corresponding bond graph is obtained by representing each of the velocities \dot{x} , \dot{y} and $\dot{r} (\equiv 0)$ by a 1-junction and the kinematic relation (Equation 5.30a) between them by means of modulated transformers as depicted in Figure 5.2. The components of the kinetic energy of the point mass and the gravitational force have been taken into account by two I energy stores and an effort source attached to the 1-junction representing \dot{y} . Due to power continuity at the 1-junctions, the equations of motion are automatically fulfilled. The controlled effort source on the left side of the graph in Figure 5.2 provides a force λ such that $\dot{r} \equiv 0$. The sum of flows at the 0-junction yields the velocity constraint (Equation 5.20). The sum of efforts at the right side 1-junctions give the equations of motion for the x- and y-direction. The geometric constraint (Equation 5.18e) is not displayed in the bond graph.

If we invert the kinematic relations (5.25) and take into account $\dot{r} = 0$ and $r = L$, then the bond graph representation of the kinematic equations

$$\dot{x} = (r \cos \phi) \dot{\phi} \tag{5.31a}$$

$$\dot{y} = (-r \sin \phi) \dot{\phi} \tag{5.31b}$$

can be extended into the bond graph of Figure 5.3 in a similar manner, as the bond graph in Figure 5.2 has been constructed [30].

In the bond graph of Figure 5.3, the controlled flow source provides the kinematic variable $\dot{\phi}$ such that the moment into the source vanishes. If we add the moments at the 1-junction of $\dot{\phi}$, eliminate \dot{x} , \dot{y} by differentiation of the flow relations of the transformers (Equations 5.31a–5.31b), then the result is the well known Lagrange

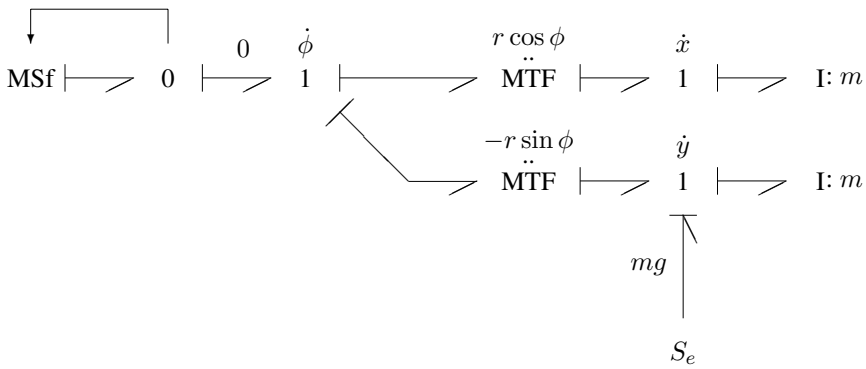


Fig. 5.3 Bond graph of the planar motion of a mathematical pendulum with Lagrange causalities

equation of the second kind

$$\ddot{\phi} + \frac{g}{L} \sin \phi = 0. \tag{5.32}$$

Alternatively, the two I energy stores could be transformed over the transformers and combined into one attached at the 1-junction representing $\dot{\phi}$. The gyrator elements that emerge if energy stores are transformed over modulated transformer cancel each other in this case.

Thus, if the kinematic displacement ϕ is chosen as an appropriate coordinate for the description of the motion of the point mass along a circle segment, then the geometric constraint is fulfilled for all values of ϕ and the constraint force λ can be eliminated from the Euler-Lagrange equations of motion. This leads us to a similar and quite elegant, approach that has been proposed by van der Shaft and Maschke in [47]. In the following, the basic idea shall be explained.

5.3 Reduction of Hamiltonian Equations of Motion with Constraints

In this section, the presentation is confined to lossless, so-called conservative mechanical systems. Hamiltonian equations, however, can also be formulated for physical systems in other energy domains [39]. Moreover, the formulation of Hamiltonian equations of motion for conservative systems originating from classical mechanics has been extended to so-called port-controlled Hamiltonian systems with dissipation [16, 46, 51]. In [16], Donaire and Junco show how a port-Hamiltonian formulation of equations can be obtained from a bond graph model including dissipation.

Hamiltonian Equations

As in Section 4.10 on Lagrange causalities, generalised coordinates of a system are denoted as $\mathbf{q} = (q_1, \dots, q_n)$ and the Lagrangian as $L(\mathbf{q}, \dot{\mathbf{q}})$. Let us define generalised momenta by $\mathbf{p} := \partial L / \partial \dot{\mathbf{q}}$. Then by applying the so-called Legendre transformation Γ to the Lagrangian, we obtain another function H of the variables \mathbf{q} and \mathbf{p} , known as the *Hamiltonian*.

$$\Gamma(L(\mathbf{q}, \dot{\mathbf{q}})) := \sum_{i=1}^n p_i \dot{q}_i - L(\mathbf{q}, \dot{\mathbf{q}}) \quad (5.33a)$$

$$H(\mathbf{q}, \mathbf{p}) := \Gamma(L(\mathbf{q}, \dot{\mathbf{q}})) \quad (5.33b)$$

Since L is the difference of the kinetic co-energy, T^* , and the potential energy, V , its Legendre transform just provides the sum of kinetic energy and potential energy.

$$\begin{aligned} H &= \sum_{i=1}^n p_i \dot{q}_i - (T^* - V) \\ &= T + V \end{aligned} \quad (5.34)$$

Let \mathbf{A}^T be an $k \times n$ matrix of row rank k (there are k independent kinematic constraints), $\boldsymbol{\lambda}$ the vector of Lagrange multipliers, \mathbf{B} an $n \times m$ matrix and \mathbf{u} the vector of system input flows. The equations of motion may be given in the form of Euler-Lagrange equations

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{A}(\mathbf{q})\boldsymbol{\lambda} + \mathbf{B}(\mathbf{q})\mathbf{u} \quad (5.35)$$

with the constraints

$$\mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (5.36)$$

for the generalised velocities $\dot{\mathbf{q}}$. The term $\mathbf{A}(\mathbf{q})\boldsymbol{\lambda}$ represents the constraint forces with $\boldsymbol{\lambda}$ uniquely determined by Equation 5.36, while $\mathbf{Q} := \mathbf{B}(\mathbf{q})\mathbf{u}$ are external forces affecting the system. By means of the Hamiltonian H , the n constrained second order Euler-Lagrange equations can be transformed into $2n$ constrained first order ODEs known as Hamiltonian equations.

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}(\mathbf{q}, \mathbf{p}) \quad (5.37a)$$

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}}(\mathbf{q}, \mathbf{p}) + \mathbf{A}(\mathbf{q})\boldsymbol{\lambda} + \mathbf{B}(\mathbf{q})\mathbf{u} \quad (5.37b)$$

$$\begin{aligned} \mathbf{0} &= \mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} \\ &= \mathbf{A}^T(\mathbf{q})\frac{\partial H}{\partial \mathbf{p}}(\mathbf{q}, \mathbf{p}) \end{aligned} \quad (5.37c)$$

Remark 5.1. If there are no constraints, then it follows immediately from the Hamiltonian equations that the Hamiltonian function, H , is in accordance with the energy conservation principle as to be expected.

$$\begin{aligned}
 \dot{H} &= \frac{\partial H}{\partial \mathbf{q}}(\mathbf{q}, \mathbf{p}) \dot{\mathbf{q}}^T + \frac{\partial H}{\partial \mathbf{p}}(\mathbf{q}, \mathbf{p}) \dot{\mathbf{p}}^T \\
 &= (\mathbf{Q} - \dot{\mathbf{p}}) \left(\frac{\partial H}{\partial \mathbf{p}} \right)^T + \frac{\partial H}{\partial \mathbf{p}} \dot{\mathbf{p}}^T \\
 &= \mathbf{Q} \left(\frac{\partial H}{\partial \mathbf{p}} \right)^T = \mathbf{Q} \dot{\mathbf{q}}^T
 \end{aligned} \tag{5.38}$$

Along with the constraints, the Hamiltonian equations constitute a DAE system of local index 2. Equations 5.37a and 5.37b can be written in the form

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial H}{\partial \mathbf{q}} \\ \frac{\partial H}{\partial \mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{A}(\mathbf{q}) \end{bmatrix} [\boldsymbol{\lambda}] + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}(\mathbf{q}) \end{bmatrix} [\mathbf{u}] , \tag{5.39}$$

where

$$\mathbf{J} := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \tag{5.40}$$

is called the Poisson structure matrix [47].

If we use the general mass matrix $\mathbf{M}(\mathbf{q})$, then the generalised momentum can be written as

$$\mathbf{p} = \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \tag{5.41}$$

and the Hamiltonian is of the form

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1}(\mathbf{q}) \mathbf{p} + V(\mathbf{q}) . \tag{5.42}$$

Regarding the constraints, Equation 5.36, we assume a so-called *holonomic* system. That is, new generalised coordinates $\tilde{q}_1, \dots, \tilde{q}_n$ can be found such that the derivatives with respect to time of $k < n$ of these coordinates vanish, $\dot{\tilde{q}}_{n-(k-j)} = 0$, ($j = 1, \dots, k$). That is, these coordinates $\tilde{q}_{n-(k-j)}$ are constant and only depend on the initial conditions. Hence, they can only be eliminated such that the equations of motion can be formulated by means of the remaining generalised coordinates. No constraints are needed [47]. In other words, it is the number of components of the vector $\dot{\mathbf{q}}$ that is necessary and sufficient to describe all motions at each time t [31]. In [31], the vector of generalised coordinates is also denoted as \mathbf{q}_K and is called the vector of *kinematic* displacements. For *non-holonomic* constraints, such elimination is not possible.

Example: Planar Motion of a Mathematical Pendulum

For the example of the planar motion of a mathematical pendulum, the constrained Hamiltonian equations are just the Equations 5.18a–5.18d and 5.20 if the Cartesian coordinates are denoted by q_x, q_y instead of x, y and if the velocities u, v are expressed by the momenta p_x, p_y .

$$\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix} = \begin{bmatrix} \frac{p_x}{m} \\ \frac{p_y}{m} \end{bmatrix} \quad (5.43a)$$

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{q_x}{L} \\ \frac{q_y}{L} \end{bmatrix} [\lambda] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ mg \end{bmatrix} \quad (5.43b)$$

$$0 = \begin{bmatrix} \frac{q_x}{L} & \frac{q_y}{L} \end{bmatrix} \begin{bmatrix} \frac{p_x}{m} \\ \frac{p_y}{m} \end{bmatrix} \quad (5.43c)$$

If we use the variables q_x, q_y, p_x, p_y also in the bond graph of Figure 5.2, then the constrained Hamiltonian equations of the pendulum can be directly derived from the bond graph. This does not only hold for the example of the pendulum. For general lossless mechanical systems, the Hamiltonian equations can be derived from the bond graph. If reference frames are chosen such that the general mass matrix becomes a diagonal matrix, then only 1-port I energy stores appear in the bond graph. Let \dot{p} denote the effort at such an I energy store port and \dot{q} the corresponding flow variable. For the i^{th} inertia, then clearly, the Hamiltonian equation $\dot{q}_i = (1/m_i)p_i$ holds. The force \dot{p}_i accelerating the i^{th} inertia results from spring forces and external forces. The same is true for moments. A spring force is related to the displacement q_C of the spring according to its characteristic. Furthermore, there is a geometric relation between the vector \mathbf{q}_C of displacements of the springs and the vector \mathbf{q} of the kinematic displacements of inertias $\mathbf{q}_C = \Phi(\mathbf{q})$. That is, the potential energy of all springs is a function of the vector \mathbf{q} of the kinematic displacements of inertias. The geometric relation differentiated with respect to time $\dot{\mathbf{q}}_C = (\partial\Phi/\partial\mathbf{q})\dot{\mathbf{q}}$ can be represented in a bond graph by means of transformers modulated by components of \mathbf{q} . Hence, equations can be derived from the bond graph that express the derivative of the vector of momenta, $\dot{\mathbf{p}}$, as a function of the vector \mathbf{q} of kinematic displacements and of external forces or moments. Thus, Hamiltonian equations can be derived directly from a bond graph of a lossless mechanical system.

Let us modify the simple example of a mathematical pendulum such that the point mass is connected to a spring and may slip along a massless rod. That is, the geometric constraint $q_x^2 + q_y^2 = L^2$ is dismissed. For this example, we obtain a bond graph that slightly differs from the one shown in Figure 5.2.

In the bond graph of Figure 5.4, we have the displacement of the spring q_C in addition to the kinematic displacements of the inertias. The Hamiltonian equations

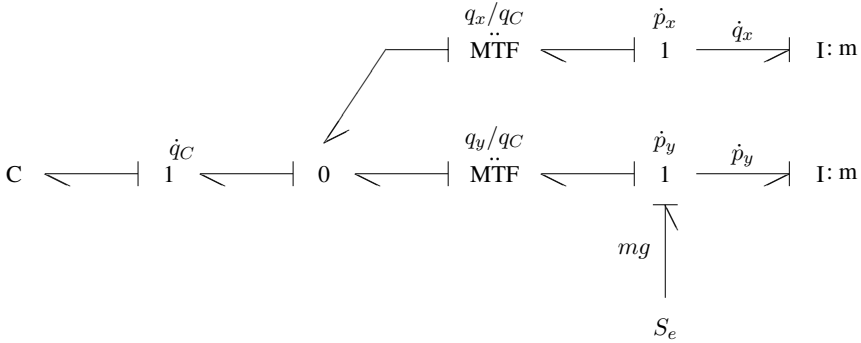


Fig. 5.4 Bond graph of a pendulum with a point mass connected to spring and sliding along a massless rod

$$\dot{q}_x = \frac{1}{m} p_x \tag{5.44a}$$

$$\dot{q}_y = \frac{1}{m} p_y \tag{5.44b}$$

$$\dot{p}_x = -\frac{1}{C} q_x \tag{5.44c}$$

$$\dot{p}_y = -\frac{1}{C} q_y + mg \tag{5.44d}$$

can be directly obtained from the bond graph in accordance with their analytical derivation from the Hamiltonian. The sum of flows at the 0-junctions yields the relation between the derivative of the spring displacement and the derivatives of the kinematic displacements of the inertias. The displacement modulated transformers establish the link between these velocities.

$$[\dot{q}_C] = \begin{bmatrix} \frac{q_x}{q_C} & \frac{q_y}{q_C} \end{bmatrix} \begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix} \left(= \frac{\partial \Phi}{\partial \mathbf{q}} \dot{\mathbf{q}} \right) \tag{5.45}$$

Reduction to Unconstrained Equations by Means of a Coordinate Transformation

Now, the basic idea of the reduction of constrained Hamiltonian equations presented by van der Schaft and Maschke [47, 48] is to perform a coordinate transformation and to use the constraints in order to reduce the number of equations of motion such that the constraint forces disappear.

To that end, first, a $n \times (n - k)$ matrix $\mathbf{S}(\mathbf{q})$ of row rank $(n - k)$ is determined such that the matrix $[\mathbf{S}|\mathbf{A}]$ is non-singular (\mathbf{A} is the matrix in the constraint equation (5.36)) and such that

$$\mathbf{A}^T(\mathbf{q})\mathbf{S}(\mathbf{q}) = \mathbf{0} \tag{5.46}$$

holds.

Now, let $\tilde{\mathbf{p}} = [\tilde{\mathbf{p}}^1, \tilde{\mathbf{p}}^2]$, then

$$\tilde{\mathbf{p}}^1 := \mathbf{S}^T(\mathbf{q}) \mathbf{p} \quad (5.47a)$$

$$\tilde{\mathbf{p}}^2 := \mathbf{A}^T(\mathbf{q}) \mathbf{p} \quad (5.47b)$$

defines a coordinate transformation. With this transformation, the Hamiltonian Equation 5.39 takes the form

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\tilde{\mathbf{p}}^1} \\ \dot{\tilde{\mathbf{p}}^2} \end{bmatrix} = \tilde{\mathbf{J}}(\mathbf{q}, \tilde{\mathbf{p}}) \begin{bmatrix} \frac{\partial \tilde{H}}{\partial \mathbf{q}} \\ \frac{\partial \tilde{H}}{\partial \tilde{\mathbf{p}}^1} \\ \frac{\partial \tilde{H}}{\partial \tilde{\mathbf{p}}^2} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{A}^T \mathbf{A} \end{bmatrix} [\boldsymbol{\lambda}] + \begin{bmatrix} \mathbf{0} \\ \mathbf{S}^T \mathbf{B} \\ \mathbf{A}^T \mathbf{B} \end{bmatrix} [\mathbf{u}] \quad (5.48)$$

and the constraint Equation 5.36 becomes

$$\frac{\partial \tilde{H}}{\partial \tilde{\mathbf{p}}^2} = \mathbf{0}. \quad (5.49)$$

In the transformed equation (5.48), $\tilde{H} = \tilde{H}(\mathbf{q}, \tilde{\mathbf{p}})$ is the Hamiltonian $H(\mathbf{q}, \mathbf{p})$ expressed in the new coordinates $\mathbf{q}, \tilde{\mathbf{p}}$. Due to the transformed constraints (5.49), the last k rows and columns in the matrix $\tilde{\mathbf{J}}$ can be deleted. The transformed constraints can be used to express $\tilde{\mathbf{p}}^2$ by \mathbf{q} and the remaining momenta $\tilde{\mathbf{p}}^1$. If the reduced Hamiltonian $H_r(\mathbf{q}, \tilde{\mathbf{p}}^1)$ is defined as $\tilde{H}(\mathbf{q}, \tilde{\mathbf{p}})$ with $\tilde{\mathbf{p}}$ satisfying the transformed constraint (5.49), then by disregarding the last row in the transformed Hamiltonian Equations 5.48, *reduced* Hamiltonian equations for the new coordinates $\mathbf{q}, \tilde{\mathbf{p}}^1$ result.

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\tilde{\mathbf{p}}^1} \end{bmatrix} = \mathbf{J}_r(\mathbf{q}, \tilde{\mathbf{p}}^1) \begin{bmatrix} \frac{\partial H_r}{\partial \mathbf{q}}(\mathbf{q}, \tilde{\mathbf{p}}^1) \\ \frac{\partial H_r}{\partial \tilde{\mathbf{p}}^1}(\mathbf{q}, \tilde{\mathbf{p}}^1) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{S}^T \mathbf{B} \end{bmatrix} [\mathbf{u}], \quad (5.50)$$

where $\mathbf{J}_r(\mathbf{q}, \tilde{\mathbf{p}}^1)$ is the skew symmetric $(2n - k) \times (2n - k)$ matrix

$$\mathbf{J}_r(\mathbf{q}, \tilde{\mathbf{p}}^1) = \left[\begin{array}{c|c} \mathbf{0}_{n \times n} & \mathbf{S}(\mathbf{q}) \\ \hline -\mathbf{S}^T(\mathbf{q}) & [-\mathbf{p}^T [S_i, S_j](\mathbf{q})]_{i,j=1,2,\dots,n-k} \end{array} \right]. \quad (5.51)$$

The expression $[S_i, S_j](\mathbf{q})$ denotes the Lie bracket of the i^{th} and the j^{th} column of the matrix $\mathbf{S}(\mathbf{q})$.¹ The vector \mathbf{p} is expressed in $\mathbf{q}, \tilde{\mathbf{p}}$ with $\tilde{\mathbf{p}}$ satisfying the constraint (5.49).

In his diploma thesis [42], supervised by the author and B. Maschke, R. Redin implemented this outlined reduction of constrained Hamiltonian equations in a procedure for the symbolic algebra program Maple™ [36]. Thus, Hamiltonian equations derived from a bond graph can be *automatically* reduced to unconstrained equations of motion before they are passed on to a numerical solver.

The approach of van der Schaft and Maschke for the reduction of constrained Hamiltonian equations for lossless mechanical systems to a reduced system of explicit ODEs without constraints is essentially based on a proper coordinate transformation. In contrast to the index reduction discussed in the previous section, the question as to which equations need to be differentiated how many times in order to obtain a set of consistent initial conditions and also the problem of numerical drift does not emerge. However, in practical symbolic computation, problems may appear, for instance, if the transformed constraint (5.49) does not allow one to express the vector of new momenta $\tilde{\mathbf{p}}^2$ as a function of \mathbf{q} and $\tilde{\mathbf{p}}^1$.

5.4 Tearing of Algebraic Constraints

In Chapter 4, bond graphs with causal paths between resistive ports as well as causal loops in the junction structure have been considered. In both cases, the mathematical model has the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{h}(t), \mathbf{u}(t)) \quad (5.52a)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}(t), \mathbf{h}(t), \mathbf{u}(t)) , \quad (5.52b)$$

where \mathbf{h} denotes the vector of auxiliary algebraic variables. For bond graphs with causal paths between resistive ports, the local index of the linearised model is equal to one. It is also equal to one for bond graphs with causal loops if they are pairwise disjoint and if their loop gain (Definition 4.16) is different from one [49]. Hence, the underlying mathematical model allows for a robust numerical solution by means of a solver based on the BDF method. However, the direct numerical solution is not always efficient. If the ODEs are not stiff and if the algebraic constraints can be symbolically solved, then there is no need for a stiffly stable implicit multi-step integration method. That is, the iterative solution of a set of algebraic equations at each time can be avoided and an integration method for explicit non-stiff ODEs

¹ Let \mathbf{F}, \mathbf{G} be two smooth functions of \mathbf{x} and let \mathbf{J} be the standard Poisson structure matrix. Then, the Lie bracket of both functions is defined as:

$$\{F, G\}(\mathbf{x}) := \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right)^T (\mathbf{x}) \mathbf{J} \left(\frac{\partial \mathbf{G}}{\partial \mathbf{x}} \right) (\mathbf{x}) .$$

could be used instead. On the other hand, the symbolic solution of the algebraic constraints can require a considerable amount of storage and computational time. Hence, an advantage can only be expected if small sets of algebraic equations are to be solved.

In [26], Granda considers the special case of a semi-explicit DAE system in which the ODEs and the algebraic equations are linear.

$$\dot{\mathbf{x}} = \mathbf{A}_0 \mathbf{x} + \mathbf{B}_0 \mathbf{u} + \mathbf{J} \mathbf{a} \quad (5.53a)$$

$$\mathbf{a} = \mathbf{K} \mathbf{x} + \mathbf{L} \mathbf{u} + \mathbf{M} \mathbf{a} \quad (5.53b)$$

where $\mathbf{A}_0, \mathbf{B}_0, \mathbf{J}, \mathbf{K}, \mathbf{L}, \mathbf{M}$ are matrices of appropriate dimensions. The components of the vector \mathbf{a} are auxiliary power variables that have been chosen in order to complete the causality assignment on a bond graph. The result, then, is a bond graph with algebraic loops. Clearly, if \mathbf{I} denotes the identity matrix and if the inverse of the matrix $(\mathbf{I} - \mathbf{M})$ exists for given numerical values of its entries, then, in this case, the semi-explicit DAE system can be transformed into a linear state space model. Its matrices can be generated in symbolic form by means of the program CAMP-G [11].

In this section, we will consider a symbolic and numerical approach that may be an alternative to a direct numerical solution of the initial DAE system [4]. It is based on the method of *tearing* algebraic equations introduced by G. Kron as early as 1962 [32]. The basic idea is to decompose a large system of equations into coupled smaller systems. Although tearing is not limited to linear systems, for the sake of a simpler presentation, we will assume that the system of algebraic constraints, Equation 5.52b, is linear in \mathbf{h} and can be solved for \mathbf{h} . That is, the system of algebraic constraints (5.52b) can be written in the form

$$\mathbf{A}(\mathbf{x}(t)) \mathbf{h} = \mathbf{b}, \quad (5.54)$$

where the $n \times n$ matrix $\mathbf{A}(\mathbf{x}(t))$ is non-singular for each time t . Moreover, in general, $\mathbf{A}(\mathbf{x})$ is a sparsely populated matrix since the components of a large physical systems usually are directly connected only to a small number of other components. If no information about the structure of the matrix is available, then a general sparse matrix solver, as it is common in circuit simulation, is used (cf., e.g., [40]). Such a solver aims to compromise between the requirement of preserving the matrix structure and the requirement of preserving its condition number (For reasons of accuracy, rows and columns will be interchanged, in order to make the element of largest absolute value in the current sub-matrix the pivot. On the other hand, additional non-vanishing elements, so-called *fill-ins*, emerge in the \mathbf{L} and \mathbf{U} factors of a permuted matrix \mathbf{A}). An alternative to a direct solution of the entire set of algebraic constraints is to partition the system with a sparse matrix into a set of small systems with dense matrices. If a sub-vector \mathbf{h}_2 of p so-called *tearing variables* is known and if the remaining components of \mathbf{h} are combined into a sub-vector \mathbf{h}_1 , then permutation matrices \mathbf{P} and \mathbf{Q} with

$$\mathbf{h} = \mathbf{Q} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{P} \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \quad (5.55)$$

can be found such that the set of algebraic equations (5.54) can be transformed into the form

$$\begin{bmatrix} \mathbf{L} & \tilde{\mathbf{A}}_{12} \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \quad (5.56)$$

In this form, \mathbf{L} is a lower block triangular matrix with blocks \mathbf{L}_{ii} ($i = 1, 2, \dots, (n - p)$) of minimal dimension on the main diagonal and diagonal elements different from zero (cf., e.g. [17]). Hence, the matrix in Equation 5.56 is of the form of a bordered block triangular matrix. This way, the task of solving a large set of algebraic equations with sparse coefficient matrix is reduced to the solution of a sequence of coupled small systems of equations with dense matrix. If, in particular, \mathbf{L} is a lower triangular matrix, then the initial task essentially reduces to the solution for the sub-vector \mathbf{h}_2 of *tearing variables*. Since \mathbf{L} is a lower triangular matrix, the equation

$$\mathbf{L} \mathbf{h}_1 = \mathbf{b}_1 - \tilde{\mathbf{A}}_{12} \mathbf{h}_2 \quad (5.57)$$

can be *symbolically* solved for \mathbf{h}_1 by forward substitution. For the sub-vector of *tearing variables*, \mathbf{h}_2 , we obtain the *smaller* system

$$(\tilde{\mathbf{A}}_{22} - \tilde{\mathbf{A}}_{21} \mathbf{L}^{-1} \tilde{\mathbf{A}}_{12}) \mathbf{h}_2 = \mathbf{b}_2 - \tilde{\mathbf{A}}_{21} \mathbf{L}^{-1} \mathbf{b}_1 \quad (5.58)$$

with a $p \times p$ coefficient matrix. Once, this set of equations is solved, which can also be done symbolically since it is usually small, then the components of \mathbf{h}_1 are obtained by forward substitution.²

Since the symbolic solution of large linear systems requires considerable memory, the reduction to a smaller system for the tearing variables \mathbf{h}_2 is essential for a *symbolic* solution of the entire system of algebraic constraints. If the smaller systems are solved numerically, then there is a reduction of computational time in comparison to a direct numerical solution of the entire original system. Let \mathbf{A} in Equation 5.54 be a dense $n \times n$ matrix and assume that the coefficient matrix in Equation 5.58 is a dense $p \times p$ matrix with $p < n$. Then, the computational cost of solving Equation 5.58 for the tearing vector, \mathbf{h}_2 , is of the order $O(p^3)$. Solving Equation 5.57 for \mathbf{h}_2 by forward substitution requires $O((n - p)^2)$ long operations. If the system of algebraic constraints is *not* torn and solved by Gaussian elimination, then the computational effort is $O(n^3)$. Thus, this way it is roughly $r := O(n^3)/(O((n - p)^2) + O(p^3))$ times more costly than solving the torn system. For illustration, consider some figures. For $n = 8$, $p = 3$, the ratio becomes $r = 9.8$.

² If \mathbf{L} is lower block triangular with blocks \mathbf{L}_{ii} on the main diagonal, then a number of small systems of equations with matrices \mathbf{L}_{ii} must be solved one after another in order to solve Equation 5.57 for \mathbf{h}_1 . If \mathbf{L} is a block diagonal matrix, the sub-vectors of \mathbf{h}_1 can be computed independently of each other. Thus, they can be computed in parallel.

The effort of solving the algebraic part of a semi-explicit DAE system 5.52a–5.52b becomes minimal if a sub-vector \mathbf{h}_2 with a minimal number of tearing variables can be identified. The equations determining the tearing variables are called *residual* equations. The problem is to find such variables. Unfortunately, the special task of finding a *minimal* number of tearing variables is a so-called NP-complete problem³ [35]. Consequently, some *heuristic* algorithms operating on the level of equations and determining a *small* number of tearing variables have been proposed in the literature (cf., e.g. [13, 44]).

In bond graphs, causal paths of a certain type express algebraic constraints. In the following, we will introduce special controlled sinks, so-called *residual sinks*, that can break up such causal paths or remove causal conflicts at junctions. By inspecting computational causalities in a bond graph and by adding these sinks in appropriate locations of a bond graph, a *small*, though not necessarily minimal, number of tearing variables can be identified and residual equations can be found in a heuristic manner. These sinks enable one to add tearing information to a bond graph that can be used in the automatic generation of equations for tearing the algebraic constraints and for their symbolic solution before the model is passed on to a numerical solver. In the following, first, this heuristic bond graph based approach to the tearing of algebraic constraints is illustrated for bond graphs with causal paths between resistive ports, but no causal paths between independent and dependent storage ports.

5.4.1 Causal Paths Between Resistive Ports

Consider the bond graph in Figure 5.5. Since the fixed causality of the effort source and the preferred integral causality of the C energy store do not propagate into the

³ The abbreviation NP stands for Nondeterministic Polynomial.

Definition 5.1 (*Polynomial problem*). A problem P is said to be *polynomial* if there exists a *deterministic* algorithm that solves the problem in time $T(n) = O(n^c)$ where n is the number of input data and c is a constant.

\mathbf{P} denotes the class of all polynomial problems.

Definition 5.2 (*Nondeterministic polynomial problem*). A problem Q is said to be NP if there exist a solution x and a *nondeterministic* algorithm that verifies the correctness of x with polynomial amount of time.

\mathbf{NP} denotes the class of all nondeterministic polynomial problems.

Remark: Each problem $Q \in \mathbf{NP}$ can be solved by means of a *deterministic* algorithm with *exponential* amount of time. Clearly, $\mathbf{P} \subseteq \mathbf{NP}$.

Definition 5.3 (*NP completeness*). A problem P is called *NP-complete* if $P \in \mathbf{NP}$ and if for every other problem $Q \in \mathbf{NP}$ there exists an algorithm T that transforms each input x of Q into an input y of P with polynomial amount of time such that x is a solution of Q if and only if y is a solution of P .

(cf. [15]). Remark: Given an NP-complete problem, it is unlikely that there exists a deterministic algorithm which solves it in polynomial time.

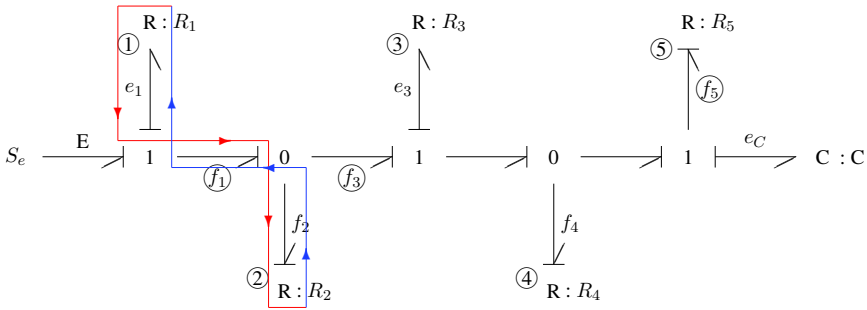


Fig. 5.5 Bond graph of a tree structure with touching causal paths between resistive ports

junction structure, the bond graph remains causally incomplete for the present. Thus, a bond must be chosen and causality must be assigned. If resistance causality is assigned to resistor R_1 , then this causal information is propagated to the 1-junction in the middle. As a consequence, again, a bond must be chosen and causality must be assigned to it. If resistance causality is chosen at resistor R_3 , then we obtain the causally completed bond graph of Figure 5.5. The resistive ports are denoted by encircled numbers. If we denote a causal path between two R elements by means of their resistance parameters, then there are the causal paths $R_1 - R_2$, $R_3 - R_4$, $R_1 - R_4$, $R_1 - R_5$ and $R_3 - R_5$. One of the flat loops (Definition 4.5) associated with these causal paths has been highlighted. The causal paths in this example have joint bonds. As we will see shortly, all algebraic loops can be broken up if the flow variables f_1, f_3, f_5 (marked in the bond graph of Figure 5.5) were known. From this bond graph, we can derive the equations

$$e_1 = R_1 f_1 \tag{5.59a}$$

$$e_3 = R_3 f_3 \tag{5.59b}$$

$$f_2 = f_1 - f_3 \tag{5.59c}$$

$$f_4 = f_3 - f_5 \tag{5.59d}$$

$$R_5 f_5 = E - e_1 - e_3 - e_C \tag{5.59e}$$

$$R_2 f_2 = E - e_1 \tag{5.59f}$$

$$R_4 f_4 = E - e_1 - e_3 \tag{5.59g}$$

$$\dot{e}_C = \frac{1}{C} f_5 . \tag{5.59h}$$

Writing the algebraic equations as a matrix equation

$$\left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & -R_1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -R_3 \\ 0 & 0 & 1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ \hline 1 & 1 & 0 & 0 & R_5 & 0 & 0 \\ 1 & 0 & R_2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & R_4 & 0 & 0 & 0 \end{array} \right] \begin{bmatrix} e_1 \\ e_3 \\ f_2 \\ f_4 \\ f_5 \\ f_1 \\ f_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ E - e_C \\ E \\ E \end{bmatrix}, \quad (5.60)$$

we see that, in fact, the latter is of the form (5.56). The left upper sub-matrix in this case is even a unity matrix. The variables e_1, e_3, f_2, f_4 can be determined by forward elimination, if the flow variables f_1, f_3, f_5 were known.

In order to obtain this information, we could add an I energy store to each of the 1-junctions. This way, the flow variables f_1, f_3, f_5 would become state variables and all causal paths between resistive ports would disappear. The outputs of the resistors would be determined by the outputs of the energy stores and the output of the effort source. The algebraic equations could be eliminated and a system of four explicit ODEs would result. In order to avoid a significant perturbation of the dynamic behaviour of the system, the parameters of the energy stores must be small leading to a stiff ODE system. If we multiply the constitutive equations of the I energy stores by their parameters to remove them from the denominator and if we let the parameters tend to zero, then the ODEs of the I energy stores reduce to coupled algebraic equations. The variables f_1, f_3, f_5 then are no longer state variables. However, together with the output of the C energy store they still determine the outputs of the resistors. Hence, they can be chosen as tearing variables.

Residual Sinks

In the bond graph, the I energy stores can be replaced by sinks that provide a flow such that the effort into the sink vanishes. For $I \rightarrow 0$, the constitutive equation of an I energy store $I \times \dot{f} = e$ becomes the trivial equation $e = 0$. The dual element would be a controlled sink that provides an effort such that the flow into the sink vanishes. We will call such sinks *residual sinks*.

In order to make the internal modulation of these sinks explicit and to avoid the introduction of a new bond graph element, we adopt a representation introduced by Bos [6] and also used by van Dijk [49].

The modulation of the flow sinks in Figure 5.6 means that these residual flow sinks adjust their output flow such that the effort into the sinks vanishes.

We used these residual sinks already in bond graph models of the mathematical pendulum (Figures 5.2 and 5.3). A residual effort sink can be used to represent an internal constraint force. Its value ensures that a velocity difference vanishes. Residual flow sinks also appear in Karnopp's approach to the derivation of Lagrange equations (Section 4.10). In that context, they are called artificial flow sources. They identify derivatives of generalised coordinates and the sum of all efforts into such

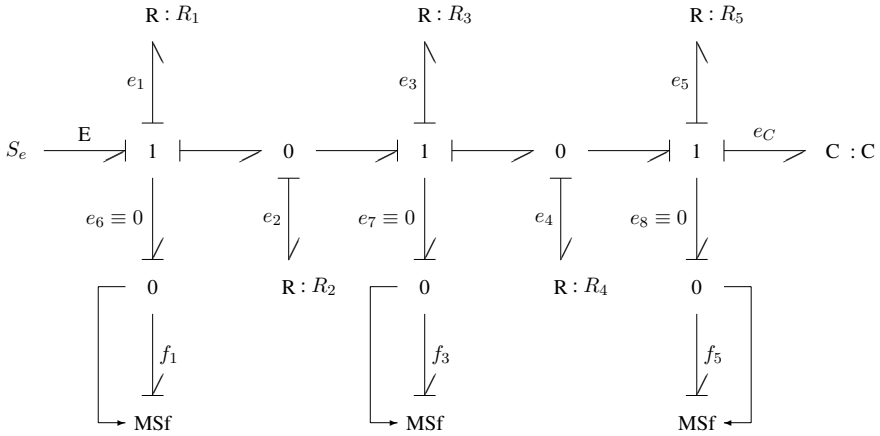


Fig. 5.6 Bond graph with three residual flow sinks

a source vanishes. In [21], Gawthrop and Smith propose a modification of the standard sequential causality assignment procedure (SCAP) by adding sources with the functionality of these residual sources to junctions that have been left causally incomplete after causality has been assigned to all storage ports and propagated into the junction structure. Furthermore, the use of residual sinks corresponds to the method of singular perturbation. In the state equations of a singular perturbed linear systems, the derivative of the state variable of the fast component is multiplied by a parameter ϵ with very small value. Letting $\epsilon \rightarrow 0$, the state equations turn into a DAE system (cf., e.g. [1]).

Observing that the efforts into the controlled sinks vanish, the following equations can be derived from the bond graph of Figure 5.6.

$$e_1 = R_1 f_1 \tag{5.61a}$$

$$e_2 = R_2 (f_1 - f_3) \tag{5.61b}$$

$$e_3 = R_3 f_3 \tag{5.61c}$$

$$e_4 = R_4 (f_3 - f_5) \tag{5.61d}$$

$$e_5 = R_5 f_5 \tag{5.61e}$$

$$0 = E - e_1 - e_2 \tag{5.61f}$$

$$0 = e_2 - e_3 - e_4 \tag{5.61g}$$

$$0 = e_4 - e_5 - e_C \tag{5.61h}$$

$$\dot{e}_C = \frac{1}{C} f_5 \tag{5.61i}$$

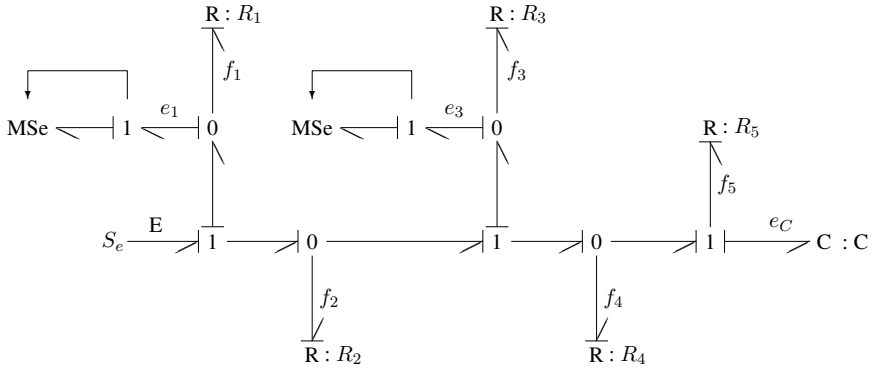


Fig. 5.7 Bond graph with two residual effort sinks

Again, these equations can be written in the form of Equation 5.56.

$$\left[\begin{array}{cccc|ccc}
 1 & 0 & 0 & 0 & 0 & -R_1 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & -R_2 & R_2 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & -R_3 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & -R_4 & R_4 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & -R_5 \\
 \hline
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0
 \end{array} \right] \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ f_1 \\ f_3 \\ f_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ E \\ 0 \\ e_C \end{bmatrix} \tag{5.62}$$

If residual flow sinks are added to the 1-junctions as depicted in Figure 5.6, then their flow variables, in fact, are possible tearing variable variables. Their number, however, is not necessarily minimal. If we take a closer look at the bond graph of Figure 5.5 and insert residual effort sinks as shown by the bond graph of Figure 5.7, then we see that these two residual effort sinks providing the efforts e_1 and e_2 are sufficient to make all causal paths between resistive ports disappear and to compute the outputs of all resistors.

From the bond graph of Figure 5.7, we obtain the matrix equation

$$\left[\begin{array}{ccc|cc}
 R_2 & 0 & 0 & 1 & 0 \\
 0 & R_4 & 0 & 1 & 1 \\
 0 & 0 & R_5 & 1 & 1 \\
 \hline
 R_1 & R_1 & R_1 & -1 & 0 \\
 0 & R_3 & R_3 & 0 & -1
 \end{array} \right] \begin{bmatrix} f_2 \\ f_4 \\ f_5 \\ e_1 \\ e_3 \end{bmatrix} = \begin{bmatrix} E \\ E \\ E - e_C \\ 0 \\ 0 \end{bmatrix} . \tag{5.63}$$

That is, the places where residual sinks can be inserted in a bond graph are not unique. Apparently, places are favourable where the residual sinks remove as many causal paths as possible. In [4], a simple heuristic algorithm has been proposed

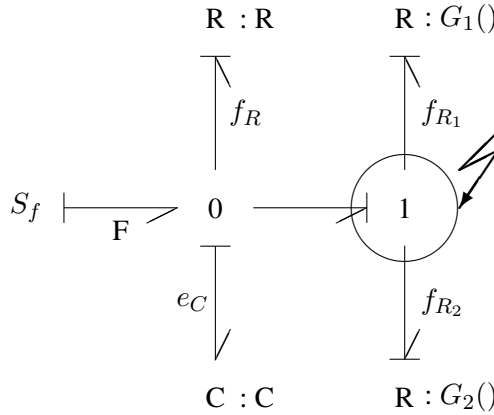


Fig. 5.8 Bond graph with causal violation at a 1-junction due to non-invertible resistor characteristics

that enables one to identify a small but not necessarily minimal number of tearing variables in bond graphs with causal paths between resistive ports and no causal paths between independent and dependent storage ports.

5.4.2 Causal Conflicts at Junctions

If there are resistive ports with non-invertible or preferable characteristic, then, instead of causal paths between resistive ports, causal conflicts at junctions can occur, especially if the method of relaxed causalities of Joseph and Martens [29] is used. As we know, causal violations at junctions introduce algebraic constraints. Their number depends on the type of the violation. In contrary to causal paths between resistive ports, violations of type 1 give an indication where a residual sink can be inserted. Consider the example of a simple circuit with two nonlinear resistors in series. Their characteristic is assumed to be non-invertible. The bond graph is depicted in Figure 5.8. The annotations $G_1()$, $G_2()$ to the resistors do not denote their conductances but the nonlinear non-invertible functions used in the constitutive equations of the resistors. Instead of adding a C energy store with a parameter of small value to the 0-junction, we insert a controlled sink that provides an effort such that the flow into the sink vanishes.

From the bond graph of Figure 5.9, we immediately get the equations

$$f_R = \frac{1}{R} e_C \tag{5.64a}$$

$$f_{R_1} = G_1(e_C - E) \tag{5.64b}$$

$$f_{R_2} = G_2(E) \tag{5.64c}$$

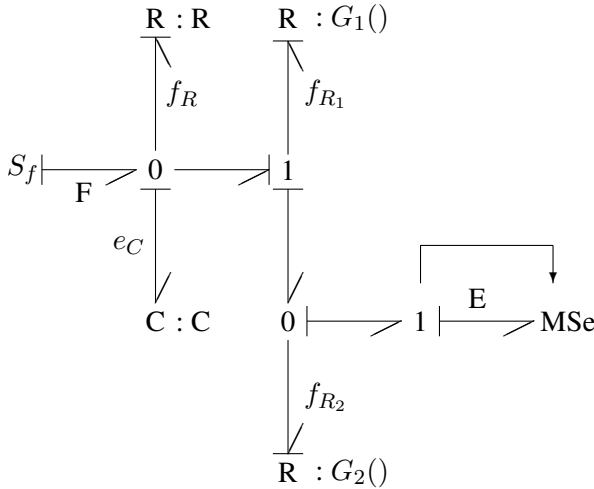


Fig. 5.9 Bond graph with residual effort sink added to the 0-junction

$$0 = f_{R_1} - f_{R_2} \tag{5.64d}$$

$$f_C = F - f_R - f_{R_1} \tag{5.64e}$$

$$\dot{e}_C = \frac{1}{C} f_C . \tag{5.64f}$$

If the variable E was known, then apparently, all other algebraic variables can be computed. This is also reflected in the so-called structural Jacobian matrix of the set of nonlinear equations. In this matrix, an entry of numerical value 1 in place (i, j) indicates that the j^{th} variable appears in the i^{th} equation (In Section 5.2, we used the term occurrence matrix). In the case of the example, the structural Jacobian matrix can be transformed into bordered lower triangular form (cf. Equation 5.56). The equation determining E is nonlinear in this case and, therefore, must be solved by iteration.

$$0 = G_1(e_C - E) - G_2(E) \tag{5.65}$$

As a result, the variable E is a tearing variable.

The fact that causal conflicts at junctions point to possible tearing variables is also evident in the bond graph of a hydraulic bridge in Figure 5.10 (cf. Figure 4.25). If we assume Bernoulli’s law in conductance causality for all four orifices, then causal conflicts at the 0-junction result that can be removed by inserting residual effort sinks. If their outputs were known, all other algebraic variables can be computed. The outputs of the residual sinks are semi-state variable that represent the pressure in the chambers of the cylinder [21].

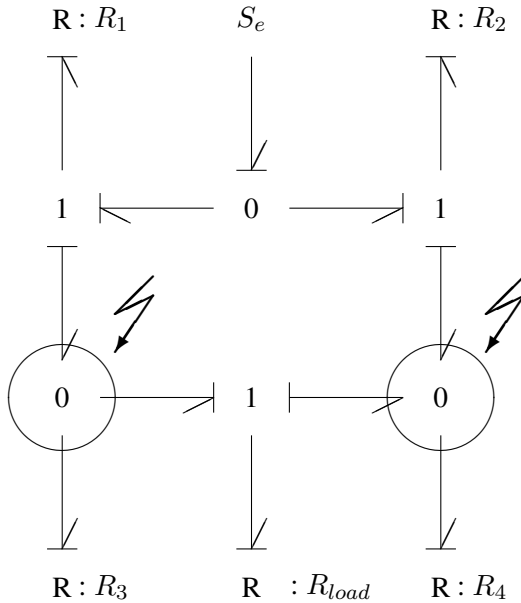


Fig. 5.10 Bond graph of a hydraulic bridge with causal conflicts at 0-junctions

5.4.3 Causal Paths Between Storage Ports of the Same Type

Thus far, only causal paths between resistive ports have been considered in this section. The approach, however, is also applicable to causal paths between independent and dependent storage ports if, as it is common in circuit analysis, energy stores are replaced by a their resistive companion model. The latter is obtained by application of the Backward Differentiation Formula, BDF, to the constitutive equation of an energy store. If energy stores are replaced this way, then causal paths between resistive ports result that can be removed by inserting residual sinks.

First, the construction of a resistive companion model for an energy store is briefly recalled. To that end, we consider a C energy store and apply the Backward Euler integration formula

$$x_n = x_{n-1} + h f_n \tag{5.66}$$

to its constitutive law.

$$\dot{u} = \frac{1}{C} i \tag{5.67}$$

In Equation 5.66, x_n denotes the approximation of the exact solution of the ODE $\dot{x} = f(x, t)$ at time t_n , $f_n := f(x_n, t_n)$ the derivative at (x_n, t_n) and h , the integration time step. As a result, we get

$$u_n = u_{n-1} + h \frac{1}{C} i_n . \tag{5.68}$$

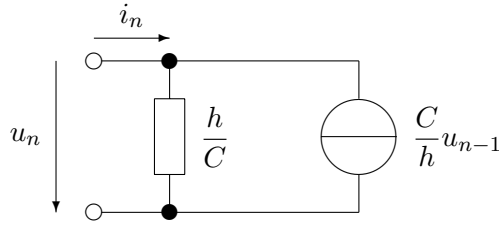


Fig. 5.11 Resistive companion model of a C energy store

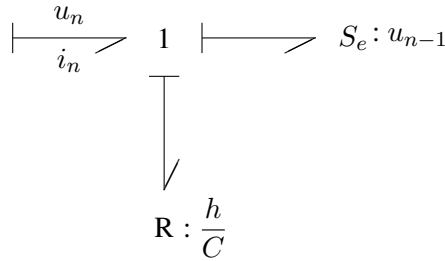


Fig. 5.12 Resistive bond graph model of a C energy store

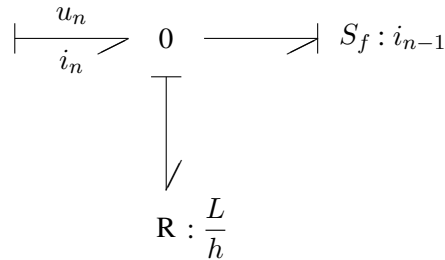


Fig. 5.13 Resistive bond graph model of an I energy store

In circuit analysis, Equation 5.68 is usually graphically represented by a resistor and a constant current source of known value as displayed in Figure 5.11. The network is called *companion model* [14]. It relates power variables u_n and i_n at present time t_n . The parameters of the elements must be updated after each time step. The corresponding bond graph model is shown in Figure 5.12.

For an I element, we get a similar resistive bond graph model (Figure 5.13). The structure of these resistive companion models remain invariant if, instead of the special implicit Euler method, the general BDF is used. Only the value of the constant source depends on the chosen BDF method.

In the following, the approach of replacing energy stores by their companion model is applied to the well known example of two capacitors connected in parallel

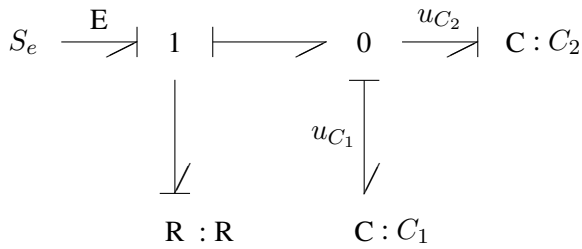


Fig. 5.14 Bond graph of two capacitors connected in parallel

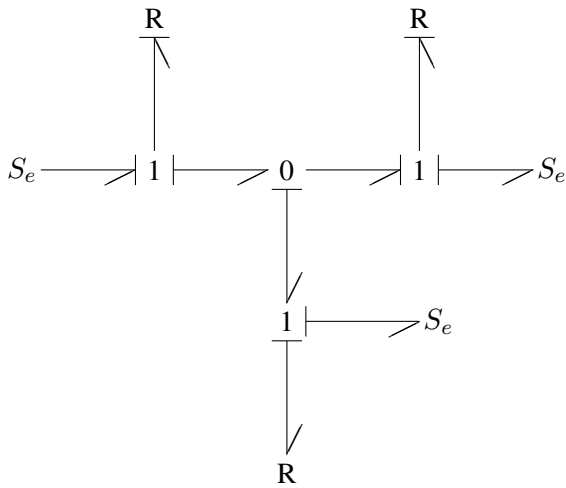


Fig. 5.15 Bond graph with C energy stores replaced by resistive companion models

(Figure 5.14). In the bond graph of this circuit, one C energy store must receive derivative causality. Replacing the C energy stores by their resistive companion model yields a bond graph with causal paths between the resistors (Figure 5.15). Again, the causal paths between resistors can be removed by inserting residual flow sinks. In the bond graph of Figure 5.16, the superscripts n or $(n - 1)$ denote times t_n or t_{n-1} .

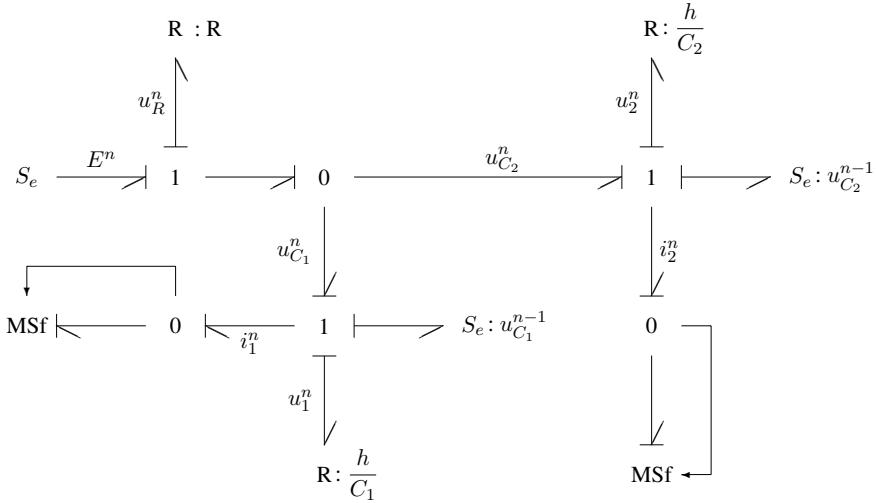


Fig. 5.16 Resistive bond graph with two residual flow sinks

The equations derived from the bond graph (Figure 5.16) can be written as a matrix equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -R & -R \\ 0 & 1 & 0 & 0 & 0 & -h/C_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -h/C_2 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ \hline 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_R^n \\ u_1^n \\ u_2^n \\ u_{C_1}^n \\ u_{C_2}^n \\ i_1^n \\ i_2^n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ E^n \\ 0 \\ u_{C_1}^{n-1} \\ u_{C_2}^{n-1} \end{bmatrix}. \quad (5.69)$$

As to be expected, Equation 5.69 has the form of Equation 5.56. Since the left upper sub-matrix is a lower triangular matrix, variables that are not tearing variables can be eliminated by forward substitution. The following residual equations result for the tearing variables:

$$\begin{bmatrix} \left(R + \frac{h}{C_1} \right) & R \\ R & \left(R + \frac{h}{C_2} \right) \end{bmatrix} \begin{bmatrix} i_1^n \\ i_2^n \end{bmatrix} = \begin{bmatrix} E^n - u_{C_1}^{n-1} \\ E^n - u_{C_2}^{n-1} \end{bmatrix}. \quad (5.70)$$

Solving Equation 5.70, we get for the voltage drop $u_{C_1}^n$ across the capacitor C_1 from Equation 5.69

$$u_{C_1}^n = \frac{R(C_1 + C_2)u_{C_1}^{n-1} + hE^n}{R(C_1 + C_2) + h}. \quad (5.71)$$

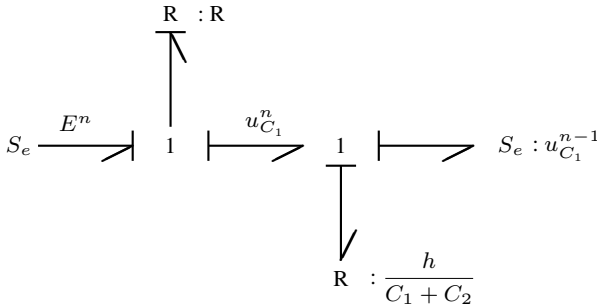


Fig. 5.17 Resistive bond graph model of the circuit with combined capacitors

Writing this equation in the form

$$u_{C_1}^n = \frac{h}{C_1 + C_2} \frac{E^n - u_{C_1}^n}{R} + u_{C_1}^{n-1}, \tag{5.72}$$

we see that we would get the same result if we combined the two parallel capacitors into one and replaced it by its resistive companion model. Equation 5.72 can be obtained directly from the bond graph of Figure 5.17. In [5], it has been shown that the case of causal paths between storage ports of the same type can be dealt with directly, without the use of resistive companion models, if the constitutive equation of the residual sources are slightly changed.

5.4.4 Causal Loops

The second case we want to consider in the context of tearing the algebraic part of a DAE system are causal loops in the junction structure [2, 3]. First, we want to show that it depends on the way causality is assigned whether a bond loop leads to a causal loop. To that end, consider the simple bond graph in Figure 5.18. The fixed causality of the flow source does not propagate beyond the 0-junction. Thus, the bond graph remains causally incomplete and a bond must be chosen for causality assignment. If *strong causality* is chosen at one of the resistive ports, i.e., a causality that is passed on by the junction the resistor is attached to, then we get a causally complete bond graph with two causal paths between the 1-port resistors but no causal loop (Figure 5.19).

If we assign *weak causality* to both resistive ports, which does not propagate, then we must chose an internal bond for causality assignment. The result is a causal loop in addition to a causal path between the resistors Figure 5.20. Finally, if we assigned weak causality to one of the resistive ports first, and strong causality to the

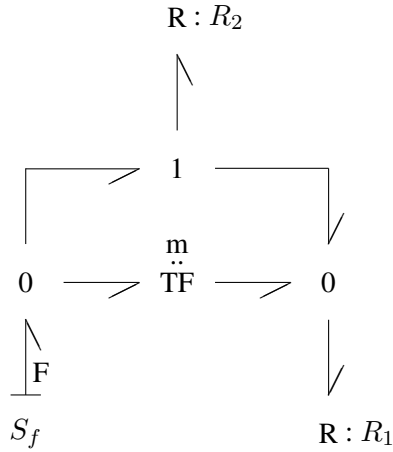


Fig. 5.18 Bond graph with bond loop

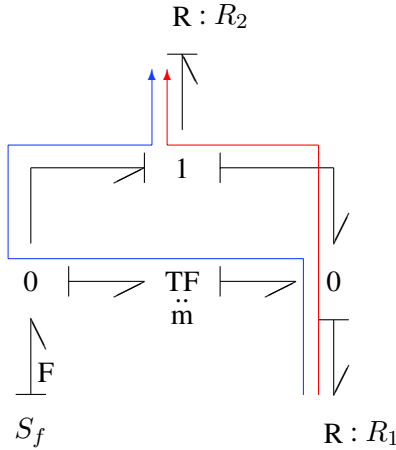


Fig. 5.19 Bond graph with causal paths between resistive ports

other resistor, then a causal conflict at the 1-junction would result because strong causality at one resistive port entails strong causality at the other one.

Now, consider the slightly modified bond graph depicted in Figure 5.21. Assigning weak causality to both resistive ports and causality to an internal bond leads to a causal loop. If strong causality is assigned to one of two resistors, say R_1 , then the result is a causal path between the two resistors and a causal conflict at one of the two 0-junctions. In this case, it is the 0-junction to which R_1 is attached. Consideration of this and other examples suggest a simple heuristic rule. Chose strong causality at an external bond (Definition 2.11) if causality must be chosen in step 5

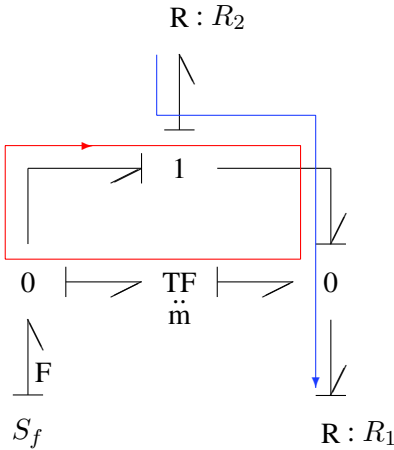


Fig. 5.20 Bond graph with causal loop and causal path between resistive ports

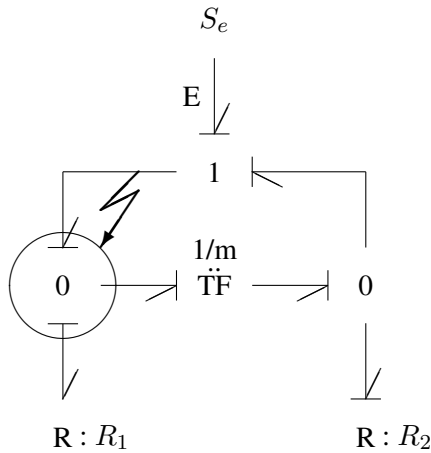


Fig. 5.21 Bond graph with causal conflict and causal path between resistive ports

of the sequential causality assignment procedure (SCAP) to enable causality propagation into the junction structure as far as possible. Emerging causal conflicts at junctions indicate a possible tearing variable. Let us move the causal conflict from one of the two 0-junctions to the 1-junction and remove the conflict as well as the causal path between both resistors by inserting residual flow sinks as displayed in Figure 5.22.

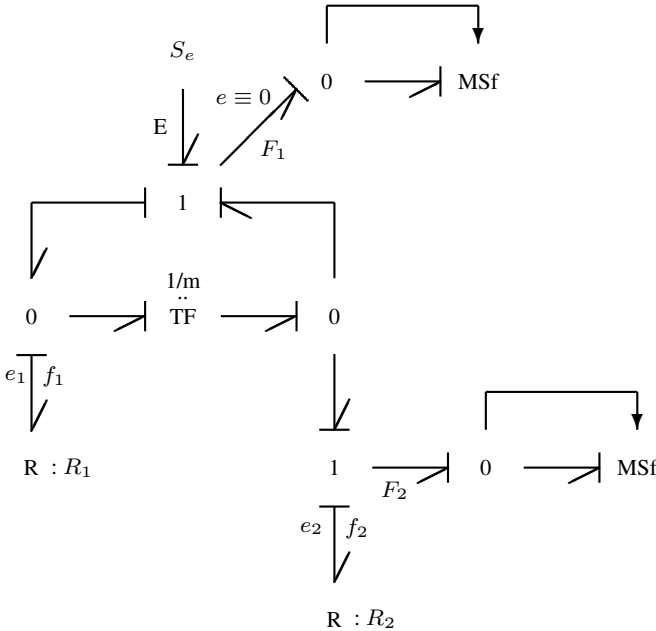


Fig. 5.22 Bond graph with residual flow sinks added

Writing equations derived from the bond graph in Figure 5.22 as a matrix equation

$$\left[\begin{array}{cc|cc} m & 0 & (1-m)R_1 & R_1 \\ 0 & 1 & 0 & -R_2 \\ \hline m & -1 & 0 & 0 \\ 1-m & 0 & 0 & 0 \end{array} \right] \begin{bmatrix} e_1 \\ e_2 \\ F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ E \end{bmatrix}, \quad (5.73)$$

we see that this equation has the form of Equation 5.56. That is, the outputs F_1 and F_2 of both residual flow sinks are indeed possible tearing variables. As can be seen from Equation 5.73, the transformer modulus must be different from unity. Otherwise, the matrix is singular.

If there are bond loops with joint bonds, then assignment of strong causality can help avoid touching causal loops. In [49], the bond graph of Figure 5.23 is causally completed such that two causal loops result that share the bond in the middle.

If strong causality is assigned to the lower left resistor, then instead of two touching causal loops, a causal conflict at a 0-junction and a causal path between the two resistors result. Both cases have been previously considered.

However, assignment of strong causalities is not always a remedy for causal loops. This can be seen from the bond graph of Figure 4.10 considered in Section 4.5. In this section, this bond graph is displayed again in Figure 5.24.

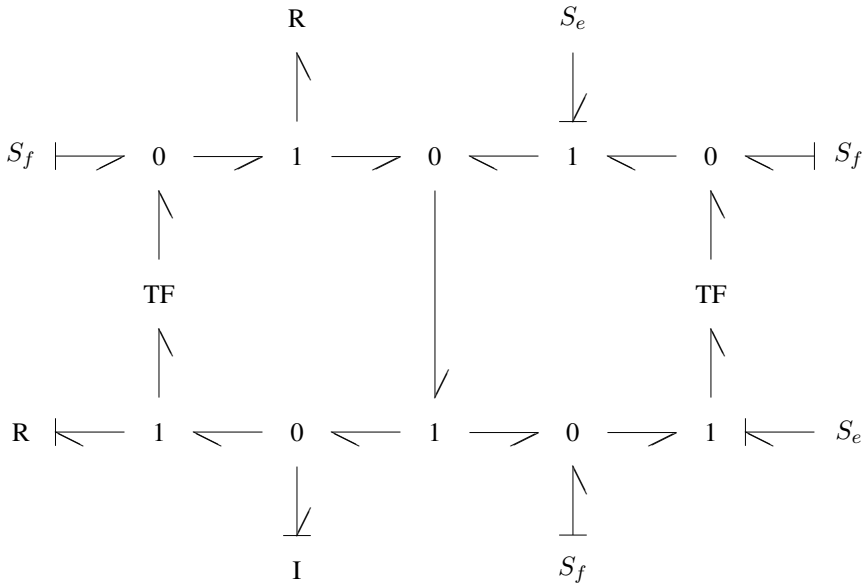


Fig. 5.23 Bond graph with two touching bond loops (van Dijk, 1994)

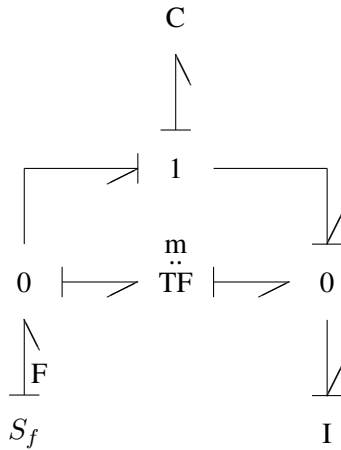


Fig. 5.24 Bond graph with causal loop that cannot be avoided

If preferred integral causality is assigned to the storage ports, then causality is determined at all external bonds. Since the fixed causality of the flow source and the preferred integral causality of the energy stores do not propagate, causality must be chosen at an internal bond leading to a causal loop. The latter is associated with algebraic loops of opposite orientation (cf. Figure 4.10). One of the two algebraic

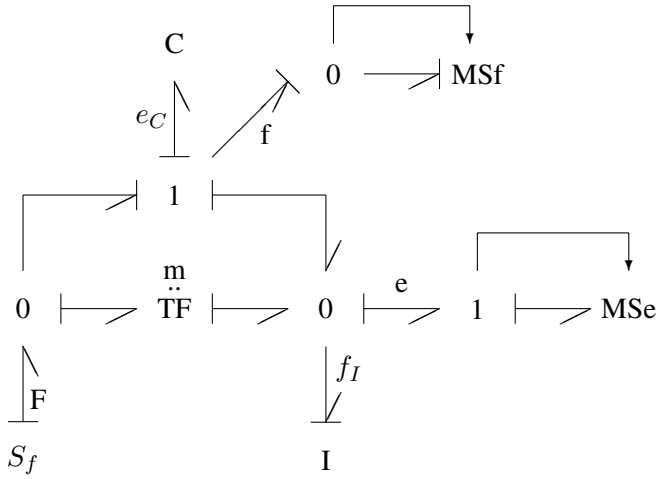


Fig. 5.25 Bond graph with residual sinks removing the causal loop

loops relates effort variables, while the other one establishes a relation between the conjugate flow variables. Hence, for each algebraic loop, a tearing variable is needed to break it. Accordingly, a residual effort and a residual flow sink have been inserted which make the causal loop disappear (Figure 5.25).

From the bond graph of Figure 5.25, two residual equations determining the tearing variables can be derived in addition to the equations for the energy stores.

$$\dot{e}_C = \frac{1}{C} f \tag{5.74a}$$

$$\dot{f}_I = e \tag{5.74b}$$

$$0 = m e - e_C - e \tag{5.74c}$$

$$0 = m (F - f) + f - f_I \tag{5.74d}$$

Since causal loops are closed causal paths in the junction structure, the algebraic loops associated with them are linear with respect to the power variables of the internal bonds if transformer moduli and gyrator ratios are either constant or depend on state variables only. Consequently, residual equations are linear with respect to the tearing variables, as in this example and, therefore, can be solved symbolically. For the existence of a unique solution, the matrix of the set of residual equations must be non-singular at each time (In the case of the example, this requirement reduces to the condition that the transformer modulus must be different from one).

If there are several causal loops that cannot be avoided by the assignment of strong causality, then the aim is to insert residual sinks that remove as many causal loops with joint bonds as possible. In the general case of large bond graphs with causal paths between resistive ports, causal conflicts at junctions and unavoidable causal loops, it may become difficult to insert the smallest possible number of resid-

ual sinks. For a symbolic solution of the algebraic part of a DAE system, however, it is essential to find a small set of tearing variables so that the solution of a large set of algebraic equations essentially reduces to the solution of a small set of residual equations for the tearing variables.

5.5 The Perturbation Index of Differential-Algebraic Equations

We close this chapter on the numerical or combined symbolic and numerical solution of differential algebraic systems by returning to the notion of the index. The latter indication of how severe difficulties in numerical solution of the DAE system can be. In Section 4.2, first, the notion of the *index of nilpotency* was introduced in the context of Kronecker’s theorem (Theorem 4.1) For general nonlinear fully implicit DAEs, the notion of the *differential index* going back to Gear was given (Definition 4.11). For linear constant coefficient DAEs, the differential index equals the index of nilpotency. In addition, Hairer, Lubich and Roche introduced the so-called *perturbation index* [27] . It is a measure of how sensitive the numerical solution of a DAE system is with regard to perturbations of the DAE. The latter can be caused by the local truncation error and the finite number of Newton-Raphson iterations at each time. Hence, the perturbation index is of fundamental importance.

Definition 5.4 (*Perturbation index of a DAE*). The general fully implicit DAE

$$\mathbf{F}(\dot{\mathbf{x}}, \mathbf{x}) = \mathbf{0} \tag{5.75}$$

has the perturbation index $p = m \geq 1$ if m is the smallest positive integer such that for solutions, $\mathbf{z}(t)$, of the DAE perturbed by the function $\mathbf{f}(t)$,

$$\mathbf{F}(\dot{\mathbf{x}}, \mathbf{x}) = \mathbf{f}(t) , \tag{5.76}$$

there exists a bound on the difference between $\mathbf{x}(t)$ and $\mathbf{z}(t)$ over the time interval $[0, t]$:

$$\|\mathbf{x}(t) - \mathbf{z}(t)\| \leq C (\|\mathbf{x}(0) - \mathbf{z}(0)\| + \max_{0 \leq \xi \leq t} \|\mathbf{f}(\xi)\| + \dots + \max_{0 \leq \xi \leq t} \|\mathbf{f}^{(m-1)}(\xi)\|) , \tag{5.77}$$

and if this bound is sufficiently small. The perturbation index is equal to zero if there exists an upper bound such that

$$\|\mathbf{x}(t) - \mathbf{z}(t)\| \leq C (\|\mathbf{x}(0) - \mathbf{z}(0)\| + \max_{0 \leq \xi \leq t} \|\int_0^\xi \mathbf{f}(\tau) d\tau\|) . \tag{5.78}$$

In both expressions, C is a constant that depends only on \mathbf{F} and on the length of the time interval.

The expression in Equation 5.77 clarifies why difficulties in the numerical solution of a DAE increase with an increasing perturbation index. In the case of a perturba-

tion index $m > 1$, not only the perturbation \mathbf{f} but also all its derivatives up to the $(m - 1)^{th}$ derivative affect the solution. In the numerical solution, the derivative leads to a term $O(\mathbf{f})/h^{m-1}$, where h denotes the step size. Hence, the higher the perturbation index and the smaller the step size, the more severe the influence of even small perturbations.

For the case of perturbed linear constant coefficient DAEs,

$$\mathbf{A} \dot{\mathbf{x}} + \mathbf{B} \mathbf{x} = \mathbf{f}. \quad (5.79)$$

Bujakiewicz [10] shows that the derivatives of the perturbation function \mathbf{f} result from the fact that the matrix $(\mathbf{A}s + \mathbf{B})^{-1}$, $s \in \mathbb{C}$, includes powers of s with exponents being non-negative integers if \mathbf{A} is singular.

In fact, by Laplace transform of the perturbed DAE, first, we get

$$(\mathcal{L}_- \mathbf{x})(s) = (\mathbf{A}s + \mathbf{B})^{-1} \mathbf{A} \mathbf{x}(0_-) + (\mathbf{A}s + \mathbf{B})^{-1} (\mathcal{L}_- \mathbf{f})(s). \quad (5.80)$$

If \mathbf{A} is non-singular, $(\mathbf{A}s + \mathbf{B})^{-1}$ can be developed into a series of powers of $1/s$.

$$\begin{aligned} (\mathbf{A}s + \mathbf{B})^{-1} &= (\mathbf{I} + \frac{1}{s} \mathbf{A}^{-1} \mathbf{B})^{-1} \frac{1}{s} \mathbf{A}^{-1} \\ &= \sum_{\nu=0}^{\infty} \left(\frac{1}{s} \mathbf{A}^{-1} \mathbf{B} \right)^{\nu} \frac{1}{s} \mathbf{A}^{-1} \end{aligned} \quad (5.81)$$

If \mathbf{A} is singular, then powers s^{ν} , $\nu \in \mathbb{N}$, lead to derivatives of the perturbation function \mathbf{f} in the back transformation of $s^k \mathcal{L}_- \mathbf{f}$.

$$(\mathcal{L}_-^{-1}(s^k \mathcal{L}_- \mathbf{f}))(t) = \mathbf{f}^{(k)}(t) + \sum_{\nu=0}^{k-1} \mathbf{f}^{(\nu)}(0_-) \delta^{(k-1-\nu)} \quad (5.82)$$

Furthermore, in [10], Bujakiewicz gives a graph-based algorithm that enables one to determine the powers of s with the highest non-negative exponents in the development of $(\mathbf{A}s + \mathbf{B})^{-1}$ by considering the indices of the non-vanishing elements of the matrices \mathbf{A} and \mathbf{B} . This result enables one to estimate how sensitive the numerical solution of each unknown is with regard to the perturbation in each equation. This information can be used to scale error estimates such that the step size control used for index 1 DAEs can also be used for higher index systems. In particular, information about the powers of s in the development of the matrix $(\mathbf{A}s + \mathbf{B})^{-1}$ can be used to scale the error estimate that decides on the termination of the Newton-Raphson iteration at each time. This way, perturbations can be diminished that result from too few iterations based on an inadequate error estimate.

The question is how this perturbation index can be related to the previously introduced differential index. In [23], Gear has shown that for ODEs, linear constant coefficient DAEs and semi-explicit systems (Equations. 4.4a–4.4b), the perturbation index is equal to the differential index. As mentioned in a remark to the definition of the differential index (Definition 4.11), the latter is equal to index of nilpotency

for linear constant coefficient DAEs. Hence, in this special case, all three indices are equal. Regarding the general case, it was believed for some time that the perturbation index p and the differential index d are related by the inequality

$$d \leq p \leq d + 1 \quad (5.83)$$

[23]. In [12], however, Campbell and Gear give an example which shows that this is not true. This example shall be recalled.

Let $m \in \mathbb{N}$, $\mathbf{y}(t) = [y_1(t), \dots, y_m(t)]^T$ and \mathbf{N} a $m \times m$ upper triangular nilpotent Jordan block. Consider the DAE

$$y_m \mathbf{N} \dot{\mathbf{y}} + \mathbf{y} = \mathbf{0}. \quad (5.84)$$

As the last row of \mathbf{N} vanishes, $y_m(t) \equiv 0$ and the differentiation index, d , becomes one. If the right-hand side is replaced by a perturbation, then $y_m(t)$ is no longer identical to zero and a differential index $p = m$ results.

5.6 Conclusion

In the previous chapter, it has been shown that the underlying mathematical model of bond graphs with certain types of causal paths has the form of a DAE system. In this chapter, approaches to the solution of DAE systems and emerging problems have been considered. First, it can be distinguished between approaches that pursue a direct numerical solution of DAE systems and those that aim to symbolically reduce the DAE system to an ODE system, or at least to simplify the original system before passing it on to a numerical solver. Regarding numerical approaches, we confined ourselves to the widely used BDF method. Discussion of the basic idea and of features of the BDF reveals that application of this method to higher order DAEs give rise to the following problems if the step size h_n is reduced:

- The local truncation error is amplified by powers of $(1/h_n)$.
- The algebraic error in the Newton-Raphson iteration is amplified by powers of $(1/h_n)$.
- The condition number of the iteration matrix becomes large for small step sizes.

Due to the above problems, it is obvious to reduce the index of a DAE system derived from a bond graph by symbolic differentiation before it is passed on to a numerical solver. This, however, entails two new problems, namely

- the problem of numerical drift and
- the problem of determining a set of consistent initial conditions for variables and their derivatives.

For illustration of both problems, the classical problem of planar motion of a mathematical pendulum has been considered.

For the problem of finding a set of consistent initial conditions, Pantelides has given a bipartite graph based algorithm that has been implemented in the modelling and simulation program *Dymola* [18]. In this software program, the algorithm is used to reduce the differential index of the DAE system.

The problem of numerical drift results from the fact that the solution of the differentiated equations only approximates the solution of the initial constraints not being differentiated. To minimise this problem, Führer proposed to differentiate some equations in order to achieve a lower index but to keep the initial algebraic constraints for numerical stabilisation such that an over-determined set of equations results. For its solution, he combines the Newton-Raphson iteration with Gauß' least square method. The approach has been implemented in the solver ODASSL.

In the case of the mathematical pendulum (Section 5.2), it has been shown that its planar motion can be described by an index 3 system taking into account the constraint force in the massless rod and the geometric constraint for the position of the point mass (cf. Equations 5.18a–5.18e). With regard to the numerical solution, an index 3 formulation is undesirable. Moreover, since the variables associated with the bonds in a bond graph are power variables, pure geometric constraints cannot be expressed in bond graphs even though displacements can modulate transformers. If the geometric constraint for the position of the point mass is replaced by a differentiated constraint, viz. a constraint for the velocity, then an index 2 DAE system results that can be represented by a bond graph (Figure 5.2). That is, if a bond graph is constructed for a mechanical system with geometric constraints, then the index of the DAE system derived from the bond graph is lower than the index of a direct formulation including constraints on geometric, on velocity and on acceleration level. The reason is that geometric constraints are not captured by a bond graph approach.

Alternatively to a (fixed) Cartesian reference frame, the angle to the vertical axis could be used as an appropriate coordinate for the description of the position of the point mass. With this coordinate, the geometric constraint is automatically fulfilled for all values of the angle. Moreover, the constraint force in the massless rod can be eliminated. The resulting model can be represented in a bond graph with Lagrange causalities (Figure 5.3).

The use of the angle as an appropriate generalised coordinate instead of Cartesian coordinates has led us to the reduction of constrained first order Hamiltonian equations of motion, as introduced by van der Schaft and Maschke. Their method uses a coordinate transformation and the constraints, and it results in a reduced set of Hamiltonian equations in which the constraint forces do not appear. The presentation of the method in this section was confined to lossless mechanical systems, although it can be applied in other energy domains, e.g., to LC networks as well. An appealing aspect of their approach is that the question as to which equations are to be differentiated how many times and the problem of determining a set of consistent initial values do not arise. As illustrated by means of the slightly modified example of the pendulum, Hamiltonian equations of motion can be directly derived from a bond graph of a mechanical system.

For bond graphs with causal paths between resistive ports or causal loops of loop gain different from one, we have seen that the underlying mathematical model is

of the form of a semi-explicit DAE system (Equations 5.52a–5.52b). Its reduction to a set of explicit ODEs by differentiation of the algebraic constraints entails the problem of numerical drift. The direct numerical solution based on the BDF method is possible since the local index is equal to one. However, at each time possibly a large set of algebraic equations must be solved in each step of the Newton-Raphson iteration. If the equations of the overall system have been automatically generated using submodels from different model libraries, information about the structure of the set of linearised equations may not be available. Consequently, only a general purpose sparse matrix solver can be used. This is not always efficient.

An alternative can be to solve the algebraic equations symbolically, or, if this is impossible due to nonlinear element characteristics, to decompose the algebraic part of a DAE system into a sequence of coupled smaller systems. If the algebraic Equations 5.52b are linear with respect to the auxiliary algebraic variables combined into a vector \mathbf{h} , then they can be symbolically solved. A large system, however, requires a considerable amount of memory and computational time. For that reason, we used the method of tearing introduced by G. Kron. With this method, the task of symbolically solving the entire system is essentially reduced to the solution of a small set of residual equations for the tearing variables. The problem is to identify a smallest possible number of tearing variables among the algebraic variables. In the literature, some heuristic algorithms operating on the equations level have been proposed.

In this chapter, a heuristic bond graph based approach has been presented. By inserting residual sinks into causal paths between resistive ports, into causal loops, or by adding them to junctions with causal conflicts, a small but not necessarily minimal number of tearing variables can be found. These residual sinks enable one to add information to a bond graph that can be used for automatic tearing of the algebraic constraints and for their possibly symbolic solution after model equations have been automatically generated. The effect of these residual sinks has been considered for bond graphs with causal paths between resistive ports and for bond graphs with causal paths between storage ports of the same type. The latter case has been reduced to the first one. Energy stores can be replaced by a resistive companion model based on the BDF, as it is common in circuit analysis.

In the case of bond graphs with bond loops, we have seen that it depends on the way causality is assigned whether causal loops will result. It has turned out that they often can be avoided if a strong causality is assigned to an external bond. Thus, instead of (touching) causal loops, causal conflicts at junctions emerge in addition to possible causal paths between resistive ports. Causal violations of type 1 can be viewed as an indication where to add a residual sink. A simple example has shown that causal loops cannot always be avoided. Since they are associated with two separate algebraic loops of opposite orientation, one for the flow variables and the other for the efforts, two residual sinks are needed to break them. In the case of bond graphs with several touching causal loops, it is obvious to insert residual sinks so that they remove as many causal loops as possible. The general case of bond graphs with causal paths between resistive ports, causal paths between storage

ports of the same type, causal loops and causal conflicts at junctions requires further investigation.

Finally, the notion of the perturbation index of a DAE system, as introduced by Hairer, Lubich and Roche, is given in order to clarify why problems in solving a DAE system numerically become more severe the higher the index is. In his dissertation, Bujakiewicz has given a graph-based algorithm that enables one to determine the perturbation index of perturbed linear constant coefficient DAEs. Moreover, the results of his research provide information on how sensitive the numerical solution of an unknown is to perturbations in some equations. This makes it possible to scale the error estimate for the step size control and the one for the termination of the Newton-Raphson iteration. Consequently, with these scalings, the DASSL code, widely used for index 1 DAEs, could also be used for direct numerical solution of higher index systems. There is no need to either modify the model or for a preceding index reduction by symbolic differentiation of equations.

References

- [1] P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella, and I. Zambettakis. *Modélisation et Identification des Processus*, volume tome 2. Éditions Technip, Paris, 1992.
- [2] W. Borutzky. Exploiting Tearing for Processing Models Containing Algebraic Constraints. In A. Javor, A. Lehmann, and I. Molnar, editors, *Modelling and Simulation 1996, Proc. of the 1996 European Simulation Multiconference*, pages 233–237. SCS Publishing, June 2–6 1996. Budapest, Hungary.
- [3] W. Borutzky. Tearing in Bond Graphs with Causal Cycles. In J. J. Granda and G. Dauphin-Tanguy, editors, *1997 International Conference on Bond Graph Modeling, and Simulation (ICBGM'97)*, pages 65–71. SCS Publishing, 1997. Simulation Series, volume 29, Number 1, ISBN: 1-56555-103-6.
- [4] W. Borutzky and F.E. Cellier. Tearing algebraic loops in bond graphs. *TRANSACTIONS of the SCS*, 13(2):102–115, June 1996.
- [5] W. Borutzky and F.E. Cellier. Tearing in Bond Graphs with Dependent Storage Elements. In *Proc. IMACS-IEEE Multiconference on Computational Engineering in Systems Applications (CESA'96)*, pages 1113–1119. IMACS, July 9–12 1996. Lille, France.
- [6] A.M. Bos. *Modelling Multibody Systems in Terms of Multibond Graphs with Application to a Motorcycle*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1986.
- [7] R.K. Brayton, F.G. Gustavson, and G.D. Hachtel. A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas. In *Proc. of the IEEE*, volume 60(1), pages 98–108, 1972.
- [8] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.
- [9] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, 1996.
- [10] P. Bujakiewicz. *Maximum weighted matching for high index differential algebraic equations*. PhD thesis, Delft University of Technology, 1994.
- [11] Cadsim Engineering. CAMP-G. URL <http://www.bondgraph.com>.
- [12] S.L. Campbell and C.W. Gear. The index of general nonlinear DAEs. *Numerische Mathematik*, 72:173–196, 1995.
- [13] F.E. Cellier and E. Kofman. *Continuous System Simulation*. Springer-Verlag, 2006. ISBN 978-0-387-26102-7.

- [14] L.O. Chua and P.M. Lin. *Computer-Aided Analysis of Electronic Circuits*. Prentice-Hall Inc., Englewood Cliffs, NJ., 1975.
- [15] S.A. Cook. The Complexity of Theorem Proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, May 1971.
- [16] A. Donaire and S. Junco. Derivation of input-state-output port-hamiltonian systems from bond graphs. *Simulation Modelling Practice and Theory*, 17(1):137–151, 2009.
- [17] L.S. Duff, A.M. Erisman, and J.K. Reid. *Direct Methods for Sparse Matrices*. Oxford Science Publications, 1986.
- [18] Dynasim. URL <http://www.Dynasim.se>.
- [19] H. Elmqvist, D. Brück, and M. Otter. *Dymola – User’s Manual*. Dynasim AB, Lund, Sweden, 1996. URL <http://www.Dynasim.se>.
- [20] C. Führer. *Differential-algebraische Gleichungssysteme in mechanischen Mehrkörpersystemen*. PhD thesis, Mathematisches Institut, Technische Universität München, 1988.
- [21] P.J. Gawthrop and L. Smith. Causal Augmentation of Bond Graphs with Algebraic Loops. *Journal of the Franklin Institute*, 329:291–303, 1992.
- [22] C. W. Gear. Differential-algebraic equation index transformations. *SIAM J. Sci. Stat. Comput.*, 9(1):39–47, 1988.
- [23] C. W. Gear. Differential-algebraic equations, indices and integral algebraic equations. *SIAM J. Numerical Analysis*, 27(6):1527–1534, 1990.
- [24] C.W. Gear. Simultaneous numerical solution of differential/algebraic equations. *IEEE Trans. Circuit Theory*, CT-18(1):89–95, 1971.
- [25] C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [26] J.J. Granda. The CAMP-G/MATLAB Solution of Algebraic Loops in Dynamic Systems represented by Bond Graph Models. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation (ICBGM’ 05)*, volume 37(1) of *Simulation Series*, pages 115–122. SCS, 2005.
- [27] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Number 1409 in *Lecture Notes in Mathematics*. Springer-Verlag, 1989.
- [28] I. Higuera, R. März, and C. Tischendorf. Stability preserving integration of index-2 DAEs. *Appl. Numer. Math.*, 45:201–229, 2003.
- [29] B.J. Joseph and H.R. Martens. The Method of Relaxed Causality in the Bond Graph Analysis of Nonlinear Systems. *Trans. ASME Journal of Dynamic Systems, Measurement, and Control*, 96:95–99, 1974.
- [30] D.C. Karnopp. Alternative Bond Graph Causal Patterns and Equation Formulations for Dynamic Systems. *Journal of Dynamic Systems, Measurement, and Control*, 105:58–63, 1983.
- [31] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1990.
- [32] G. Kron. *Diakoptics – The Piecewise Solution of Large-Scale Systems*. MacDonald & Co., London, UK, 1962.
- [33] P. Lötstedt and L.R. Petzold. Numerical Solution of Nonlinear Differential Equations with Algebraic Constraints. Technical Report SAND83-8877, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, November 1983.
- [34] P. Lötstedt and L.R. Petzold. Numerical solution of nonlinear differential equations with algebraic constraints I: Convergence results for backward differentiation formulas. *Math. Comp.*, 46:491–516, 1986.
- [35] R.S.H. Mah. *Chemical Process Structures and Information Flows*. Butterworths, 1990.
- [36] Maplesoft. URL <http://www.maplesoft.com/>.
- [37] R. März and C. Tischendorf. Solving more general index 2 differential algebraic equations. *Comp. and Math. with Appl.*, 28(10-12):77–105, 1994.
- [38] R. März and C. Tischendorf. Recent results in solving index 2 differential algebraic equations in circuit simulation. *SIAM J. Sci. Comput.*, 18 (1):139–159, 1997.

- [39] B.M. Maschke, A.J. van der Schaft, and P.C. Breedveld. An intrinsic Hamiltonian formulation of the dynamics of LC-circuits. *IEEE Trans. Circuits and Systems*, CAS-42:73–82, 1995.
- [40] L.W. Nagel. *SPICE2: A computer program to simulate semiconductor circuits*. PhD thesis, University of California, Berkeley, 1975. Memorandum No. ERL-MS20.
- [41] C.C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM, Journal of Scientific and Statistical Computation*, 9:213–231, 1988.
- [42] R. Reddin. Entwicklung einer computerunterstützten symbolischen Berechnung für die Reduzierung von zwangsgeführten mechanischen Systemen. Master's thesis, Fachhochschule Köln - École Supérieure des Arts et Métiers, Paris, Paris, 1997.
- [43] B.P. Sommeijer. Parallelism in the numerical integration of initial value problems. CWI tract 99, CWI, Amsterdam, 1993.
- [44] D.V. Steward. Partitioning and tearing systems of equations. *J. SIAM Numer. Anal., Ser. B.*, 2(2):345–365, 1965.
- [45] C. Tischendorf. *Solution of index-2-DAEs and its application in circuit simulation*. PhD thesis, Humboldt-Univ. zu Berlin, 1996.
- [46] A.J. van der Schaft. *L₂-Gain and Passivity Techniques in Nonlinear Control*. Springer-Verlag, London, 2000.
- [47] A.J. van der Schaft and B.M. Maschke. On the Hamiltonian formulation of nonholonomic mechanical systems. *Reports on Mathematical Physics*, 34(2):225–233, 1994.
- [48] A.J. van der Schaft and B.M. Maschke. The Hamiltonian Formulation of Energy Conserving Physical Systems with External Ports. *AEÜ*, 49(5/6):362–371, 1995.
- [49] J. van Dijk. *On the role of bond graph causality in modelling mechatronic systems*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1994.
- [50] J. van Dijk and P.C. Breedveld. Simulation of System Models Containing Zero-order Causal Paths — II. Numerical Implications of Class 1 Zero-order Causal Paths. *Journal of the Franklin Inst.*, 328(5/6):981–1004, 1991.
- [51] D. Vink. *Aspects of Bond Graph Modelling in Control*. PhD thesis, Dept. of Mechanical Engineering, University of Glasgow, 2005.

Chapter 6

Analysis of Causal Bond Graph Models

So far, we have seen that causalities can be assigned to a bond graph by different methods (SCAP, relaxed causalities, Lagrange causalities) and that different forms of mathematical time domain models can be derived from a bond graph (state space form, descriptor form, Lagrange equations). Since mathematical models derived from bond graphs frequently take the form of a DAE system, its index and approaches to a symbolic and numerical solution have been considered.

However, not only simulation of the dynamic behaviour of a multidisciplinary system is of concern, but other tasks also, e.g., the determination of a steady-state needed for the linearisation of the model equations, the establishment of transfer functions, the determination of pole-zero loci. Moreover, with regard to the design of a controller, properties, e.g., structural controllability and observability, are of interest. Of course, once a time domain model is available, the information needed can be derived from the linearised model equations.

In this chapter, we will see that not only time domain models but other information relevant for control system design can be derived directly from a causally completed bond graph. That is, bond graphs can be viewed as a *core* model representation from which information for different purposes can be derived in suitable form.

According to this view, Gawthrop and Smith developed a set of model transformation tools collected in a toolbox MTT (*Model Transformation Tools*) [35]. Depending on the actual task, these tools enable one to automatically transform one model representation into another where an acausal bond graph is the core representation. Aspects of this approach to automated modelling will be considered separately in Section 11.6.6.

In the next section, it is shown how a bond graph can be used to set up equations for the determination of the steady-state of a dynamic system.

6.1 Equations Determining the Steady-state of a Dynamic System

In bond graph terms, the steady-state of a dynamic system is characterised by the fact that flows into C energy stores and efforts into I elements are equal to zero. This can be expressed in a bond graph in two different ways. First, a C energy store can be replaced by a flow sink that imposes a flow equal to zero. The value of the effort into the sink is the steady-state to be determined. Alternatively, a C energy store can be replaced by an effort sink imposing an unknown constant effort such that the flow into the sink is equal to zero. Reverse statements hold for an I element dual to a C energy store. The first option corresponds to the approach adopted in circuit analysis. For the determination of the steady-state, capacitors in network are removed and inductances are replaced by short circuits. In bond graphs, the substitution of energy stores entails a reassignment of causalities. As a consequence, it can happen that a flow sink replacing a C energy store attached to a 0-junction does not determine the common effort at that junction anymore. In the same way, an effort sink replacing an I element attached to a 1-junction does not determine the common flow at that junction. Consequently, the causality at some resistive port or at an internal bond must be changed. Thus, a causal path between resistive ports or a causal loop may emerge. On the contrary, if C energy stores are replaced by constant effort sinks and I elements by constant flow sinks, then causalities are retained. In both cases, algebraic equations determining the steady-state can be derived from the modified bond graph. The resulting sets of equations are equivalent, but different in form due to different causalities in the bond graph.

Example: RC-Network

For illustration, consider the example of the simple RC-network depicted in Figure 6.1. If the capacitors are replaced by flow sinks, then the bond graph shown in Figure 6.2 results with three touching causal paths between the resistors. The causal paths could be removed by adding a residual effort sink (Section 5.4) to each 0-junction. Since the effort imposed by the residual sinks is equal to the effort into the flow sinks that replace the C energy stores, the flow sinks can be omitted. Consequently, we get the same bond graph (Figure 6.3) we would get if the C elements are replaced by effort sinks instead of flow sinks.

From the bond graph of Figure 6.3, the algebraic system

$$\left[\begin{array}{cccc|cc} R_1 & 0 & 0 & 0 & -1 & 0 \\ 0 & R_2 & 0 & 0 & -1 & 1 \\ 0 & 0 & R_3 & 0 & 0 & -1 \\ 0 & 0 & 0 & R_4 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 0 \end{array} \right] \begin{bmatrix} f_{R_1} \\ f_{R_2} \\ f_{R_3} \\ f_{R_4} \\ e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ E \\ F \\ 0 \end{bmatrix} \quad (6.1)$$

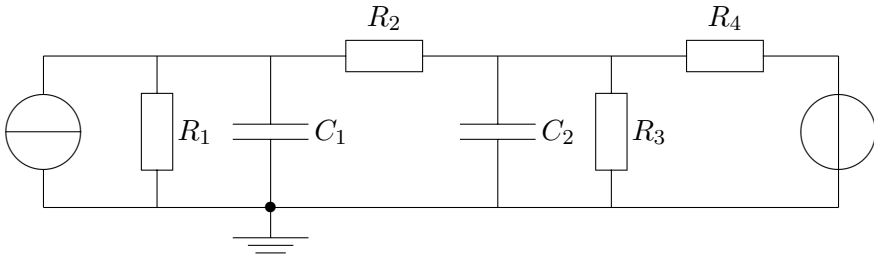


Fig. 6.1 RC-network

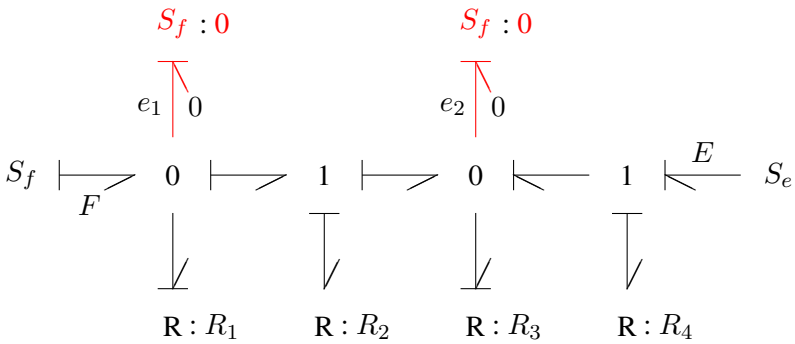


Fig. 6.2 Bond graph with flow sinks for determination of the steady-state

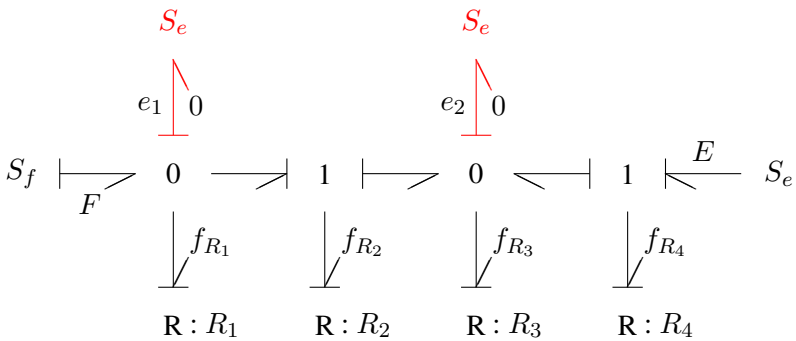


Fig. 6.3 Bond graph with effort sinks for determination of the steady-state

can be directly derived where e_1 and e_2 are the tearing variables to be determined. The residual equations

$$0 = F - f_{R_1} - f_{R_2} \tag{6.2a}$$

$$0 = f_{R_2} - f_{R_3} + f_{R_4} \tag{6.2b}$$

in the above set of Equations 6.1 can also be obtained by just setting the derivatives \dot{e}_1, \dot{e}_2 in the dynamic equations equal to zero (From the bond graph of Figure 6.2, a similar set of equations determining the efforts into the flow sinks can be derived).

Depending on the approach, either the output or the input of a sink replacing an energy store is equal to zero. Propagation of this information into the junction structure can lead to relations that are either inconsistent or are fulfilled for all values of a variable. That is, in such a case, the bond graph reveals that there is no unique solution to the steady-state equations or that they are fulfilled for an infinite number of values [16].

Example: RLC-Network

For illustration, consider the simple network in Figure 6.4. If in the corresponding bond graph (Figure 6.5) the C energy stores are replaced by an effort sink and the I element by a flow sink with inputs equal to zero, then flow continuity at the 0-junction requires that the flow of the independent flow source must be zero. If this is not the case, then no steady-state solution exists.

On the other hand, let us assume that the switch, S_2 , in the right side series connection of elements first is open and the capacitor C_1 is charged to a certain value. Then, the flow source is switched off and the switch S_2 is closed. In this case, the constitutive equation of the transformer relating the efforts, viz. $e_2 = n \times e_1$ holds for all values of e_1 . That is, an infinite number of steady-states exists.

In summary, we see that the equations determining the steady-state of a dynamic system can be derived directly from a bond graph after energy stores have been replaced by sinks for which either the input or the output is equal to zero depending on the chosen approach. If C energy stores are replaced by flow sinks and I elements by effort sinks, then causalities must be reassigned. If the output of a sink replacing an energy store vanishes, then propagation of the value zero into the junction structure may yield that a relation of a junction is not fulfilled or that it holds for an infinite number of values. In the first case, there is no steady-state solution. In the second case, there are many solutions.

For mechanical systems of which the steady-state is due to gravity effects, Dauphin-Tanguy and her co-authors use *structural* analysis of the bond graph in order to decide whether a solution exists and show how an existing steady-state can

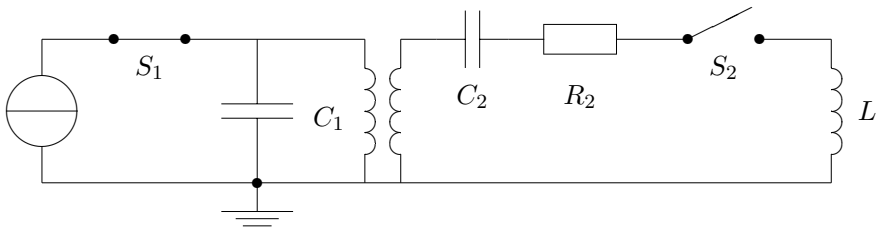


Fig. 6.4 Network for which no steady-state exists

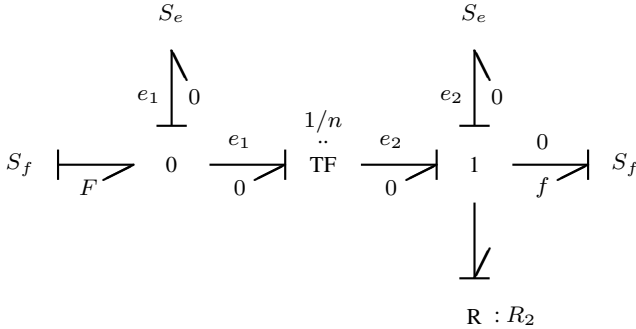


Fig. 6.5 Bond graph for the determination of the steady-state of the network in Figure 6.4

be determined *symbolically* directly from the bond graph [25]. Their approach also applies for the case that the $n \times n$ matrix \mathbf{A} of a linear state space model is singular. In that case, the steady-state values of r components of the state vector depend on the values of the remaining $n - r$ components, where $r = \text{rank} \mathbf{A}$.

If the equations determining the steady-state are nonlinear, then a problem well known, e.g., from circuit analysis, is that Newton-Raphson iteration only converges if an initial guess of the solution is already sufficiently close to the solution to be determined. A possible approach known from circuit analysis is to perform a so-called *pseudo* transient analysis in which the values of all independent sources are linearly increased from zero to their initial value during a ramping time. Then, they are held constant for some settling time. The final values of voltages and currents in this pseudo transient analysis are the steady-state solution that can be used for a subsequent real circuit simulation.

6.2 Transfer Functions

Once the steady-state of a system has been determined, the model equations can be linearised and a small signal analysis can be performed. Of course, the matrix of transfer functions can be obtained analytically by Laplace transform of the linearised DAE system.

On the other hand, transfer functions relating an input to an output can be derived directly from a bond graph of a linear time-invariant multiple input, multiple output system (MIMO system). This can be accomplished by means of the loop rule originally presented by Shannon in 1941 and rediscovered by Mason in 1955 [51]. The loop rule applies to signal flow graphs. However, as Brown [18] has shown, it can be applied directly to bond graphs so that there is no need to convert a bond graph into a signal flow graph.

6.2.1 Transfer Functions from the State Space Model

First, consider the DAE system

$$\mathbf{0} = \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}) \quad (6.3a)$$

$$\mathbf{y} = \mathbf{G}(\mathbf{x}, \mathbf{u}) \quad (6.3b)$$

and let $\mathbf{x}_s, \mathbf{u}_s, \mathbf{y}_s$ denote the steady-state solution. Then, the deviations from steady-state values $\Delta\dot{\mathbf{x}}, \Delta\mathbf{x}, \Delta\mathbf{u}$ are determined by the linear DAE system

$$\mathbf{E}\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u} \quad (6.4a)$$

$$\Delta\mathbf{y} = \mathbf{C}\Delta\mathbf{x} + \mathbf{D}\Delta\mathbf{u}, \quad (6.4b)$$

where

$$\mathbf{E} := \left. \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right|_{\mathbf{x}_s, \mathbf{u}_s} \quad (6.5a)$$

$$\mathbf{A} := - \left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_{\mathbf{x}_s, \mathbf{u}_s} \quad (6.5b)$$

$$\mathbf{B} := - \left. \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right|_{\mathbf{x}_s, \mathbf{u}_s} \quad (6.5c)$$

$$\mathbf{C} := \left. \frac{\partial \mathbf{G}}{\partial \mathbf{x}} \right|_{\mathbf{x}_s, \mathbf{u}_s} \quad (6.5d)$$

$$\mathbf{D} := \left. \frac{\partial \mathbf{G}}{\partial \mathbf{u}} \right|_{\mathbf{x}_s, \mathbf{u}_s}. \quad (6.5e)$$

Laplace transform of the linearised DAE system yields a direct relation between inputs and outputs

$$\mathcal{L}\{\Delta\mathbf{y}\}(s) = \mathbf{H}(s)\mathcal{L}\{\Delta\mathbf{u}\}(s), \quad (6.6)$$

where $s \in \mathbb{C}$ and

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (6.7)$$

is the matrix of transfer functions to be determined. Assuming that the linearised constant coefficient DAE system is solvable means that $(s\mathbf{E} - \mathbf{A})$ is a regular pencil. Hence, the inverse $(s\mathbf{E} - \mathbf{A})^{-1}$ exists.

An entry h_{ij} in the matrix \mathbf{H} represents the transfer function relating the j^{th} input to the i^{th} output under the assumption that all other inputs are not taken into account. As it is known, the inverse of a non-singular matrix \mathbf{M} can be computed by means of its adjoint according to $\mathbf{M}^{-1} = (1/\det \mathbf{M}) \cdot \text{Adj} \mathbf{M}$. Hence, each coefficient h_{ij} is a ratio of two polynomial in $s \in \mathbb{C}$ with the polynomial

$$\Delta(s) := \det(s\mathbf{E} - \mathbf{A}) \quad (6.8)$$

in the denominator. The poles of the transfer functions h_{ij} , i. e., the roots of the characteristic equation

$$\det(s\mathbf{E} - \mathbf{A}) = 0, \quad (6.9)$$

are equal to the eigenvalues of the matrix \mathbf{A} if factors common in the nominator and in the denominator polynomial are not cancelled (pole-zero cancellation). Their loci in the complex plane determine the stability of a linear time-invariant (LTI) system.

6.2.2 Transfer Functions from a Signal Flow Graph

Instead of deducing a transfer function between a given input and a given output from the linearised and Laplace transformed DAE system, alternatively, it can be obtained directly from a signal flow graph representation of the linearised model by means of Mason's loop rule. To that purpose, signal loops must be identified and their loop gain must be determined. Since every bond in a bond graph is associated with two signals of opposite direction, a bond graph of a linear time-invariant system could be converted systematically into a signal flow graph. The result would be less clear and concise than the original bond graph. As Brown has shown in [18], the information needed for application of the loop rule can be determined directly from the bond graph. For that reason, we will address the conversion of a bond graph into a signal flow graph only briefly with regard to signal loops and their loop gain. Moreover, in the following, the application of Mason's loop is only illustrated by means of some simple bond graph examples. For further details, see [17, 18].

In Section 4.1, two types of signal loops were introduced, namely *flat* loops (Definition 4.5) and *open* loops (Definition 4.6). As depicted in Figure 4.3, a causal path between two 1-port resistors corresponds to a flat loop. This fact is not restricted to resistors. After Laplace transform of the constitutive equations of linear 1-port energy stores, their input and output variables are algebraically related like the power variables of 1-port resistors. However, there is no signal loop associated with a causal path between a 1-port source and a 1-port resistor or a 1-port energy store because the input and output of an ideal independent source are not related. That is, the associated signal path is not closed.

In Section 4.3, we saw that causal paths between a storage port and a resistive port represent a transient. If this component of the response of the linearised system to input signals is considered isolated, then its time constant is equal to the product of the element parameters. Similarly, a causal path between a 1-port C energy store and a 1-port I energy store identifies an oscillation. If it is considered isolated, then its natural frequency is determined by the parameters of the 1-port energy stores (Equation 4.18). In Section 4.5 on causal loops in the junction structure, we introduced the notion of the *loop gain* (Definition 4.15) and calculated the loop gain of the causal loops in the bond graph of Figure 4.10. With regard to flat loops, here we add that the influence coefficient (Definition 4.14) of a linear 1-port resistor is either its parameter R or $1/R$ depending on its causality. Likewise, the influence

coefficient of a linear 1-port energy store of parameter p after Laplace transform of its constitutive equation is either $1/(sp)$ for integral causality or sp in the case of derivative causality. Thus, we can identify flat and open loops in bond graphs and can determine their loop gain. For illustration, consider the simple example in Figure. 6.6. The corresponding signal flow graph is shown in Figure 6.7.

Comparing both graphs, one can deduce how bond graphs with passive 1-port elements can be converted into a signal flow graph (multiport energy stores and resistors will have to be replaced by equivalent bond graph models having only 1-port energy stores and 1-port resistors). The product of all influence coefficients along the signal loop gives the loop gain $L = -n^2/(sRC)$. The result is not surprising since the causal path between the C energy store and the R element corresponds to a control loop with an open loop gain of the same value. Since the positive reference directions of the energy flows point to the ports of the passive elements, the power orientation must change for an odd number of bonds that are members of the causal path between the two passive elements (In the example, the power orientation changes from bond 1 to bond 3). Therefore, the loop gain of a flat loop between passive ports is always negative. For open loops associated with simple meshes (Definition 3.8), the loop gain is positive if the number of bonds in the bond loop is even (Definition 3.9). If the simple mesh is odd, then the loop gain of the open loop is negative ([18], Theorem 4, p. 257).

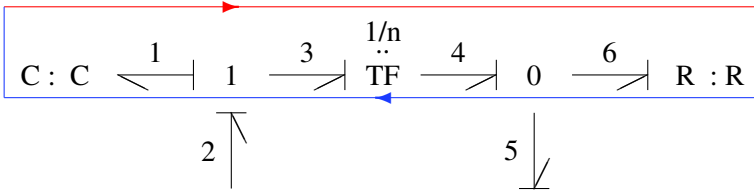


Fig. 6.6 Bond graph fragment with a flat loop

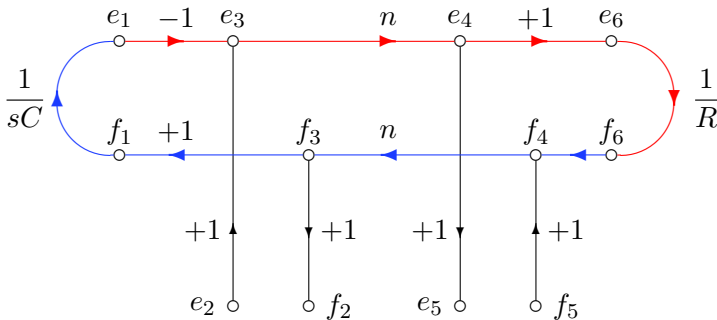


Fig. 6.7 Signal flow graph corresponding to the bond graph fragment in Figure 6.6

Mason's loop rule now states that the transfer function h_{ij} between the j^{th} input and the i^{th} output of a linear time-invariant system with multiple inputs and outputs can be determined by calculating loop gains. The loop rule can be expressed as

$$h_{ij} = \frac{1}{\Delta} \times \sum_k G_k \Delta_k . \quad (6.10)$$

In this equation, the denominator Δ is the same for all ratios of Laplace transformed outputs and inputs ($\Delta = \det(s\mathbf{E} - \mathbf{A})$). It is called the *graph determinant* [51] or the system determinant. The symbol G_k represents the path gain of the k^{th} path between two nodes in the signal flow graph representing an input and an output variable. The path gain is the product of all influence coefficients along the path. The symbol Δ_k denotes the determinant of the reduced signal flow graph that results if the k^{th} path is expunged with all its nodes. The sum extends over all paths from the node of the j^{th} input to the i^{th} . The graph determinant in the denominator is equal to the sum

$$\Delta = 1 - \sum_i G_i + \sum_{i,j} G_i G_j - \sum_{i,j,k} G_i G_j G_k + \dots . \quad (6.11)$$

In this sum, the term $\sum_i G_i$ is the sum of gains of all individual loops in the signal flow graph. The sum $\sum_{i,j} G_i G_j$ extends over all products of loop gains of two loops that do not touch, in other words, that do not share a node. Since loops in a signal flow graph usually often touch one another, there are only few pairs of non-touching loops among all loops. Similarly, the term $\sum_{i,j,k} G_i G_j G_k$ is the sum of all products of loop gains for sets of three non-touching loops (Different sets of three loops may share two loops but must have a distinct third loop). In practice, such sets are rather rare. For the k^{th} path between the nodes of an input and an output variable, the reduced signal graph is obtained by expunging all loops that touch the k^{th} path. For computation of the denominator, Δ , *all* loops must be considered. Among these, all loops not touching the k^{th} path between the nodes of the input and the output variable contribute to the determinant of the reduced signal flow graph, Δ_k , in the nominator.

6.2.3 Transfer Functions Directly from a Causal Bond Graph

Now, since we do not want to convert a bond graph into a signal flow graph in order to be able to apply Mason's loop rule, the signal loops must be identified in the bond graph itself. As explained, there are flat loops associated with causal paths between passive 1-port elements and open loops associated with causal loops. Concerning the touching of these loops, Brown gives the following rules in bond graph terms.

- Flat loops touch each other if and only if their corresponding causal paths have a common 0- or 1-junction. If a causal path between two passive 1-port elements shares a junction with another possibly closed path, then their corresponding signal loops touch one another.
- The two open loops of opposite orientation of a closed causal path touch each other if the mesh includes one or several mesh stubs (cf. Figure 4.4).
- Open loops of two different meshes touch one another if both have a common junction located in a mesh stub of one of both meshes (They may not touch even if both meshes have a joint bond).
- Finally, two signal loops touch each other, if their corresponding causal paths have a common passive 1-port element.

With this information, we are now able to derive transfer functions directly from a bond graph using Mason’s loop rule. This will be illustrated by means of two examples.

Example: Bond Graph with a Tree-Structure

As an example of a bond graph with a tree-like structure, consider the bond graph model of a DC motor depicted in Figure 6.8.

In this model, I_a denotes the inductance and R_a the resistance of the armature. The symbol k_m represents the torque constant. On the mechanical side, J_m is the moment of inertia and r_m the friction coefficient. The effort source accounts for a load moment. Since the bond graph is of tree-like structure, there are no bond loops and hence no open signal loops. However, there are three causal paths highlighted by additional oriented thin lines, one between the mechanical inertia and the inductance, another between the electrical resistor and the induction, and a third one between the mechanical resistor and the inertia where the second and the third path

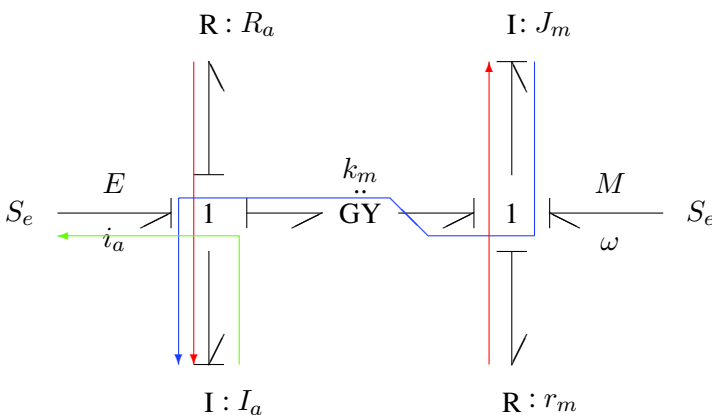


Fig. 6.8 Bond graph of a DC motor with tree-like structure

do not touch. The gain of the associated flat loops are $-k_m^2/(s^2 J_m I_a)$, $-R_a/(s I_a)$ and $-r_m/(s J_m)$. Hence, the graph determinant reads

$$\Delta = 1 + \frac{k_m^2}{s^2 J_m I_a} + \frac{R_a}{s I_a} + \frac{r_m}{s J_m} + \frac{R_a r_m}{s I_a s J_m} . \tag{6.12}$$

If we want to determine the transfer function between the input voltage u_a and the armature current i_a as the output, then the signal path between the nodes of these variables in the corresponding signal flow graph is associated with the causal path between the voltage source and the inductance. It is the only signal path between the two variables. Its path gain, G_1 , is equal to $-1/(s I_a)$. Since the flat loops between the electrical resistor and the inductance and the one across the gyrator do touch this signal path, they are left out in the computation of the reduced graph determinant Δ_1

$$\Delta_1 = 1 + \frac{r_m}{s J_m} . \tag{6.13}$$

According to Mason's loop rule, the transfer function in question therefore is

$$\begin{aligned} \frac{\mathcal{L}i_a}{\mathcal{L}E} &= \frac{G_1 \Delta_1}{\Delta} \\ &= \frac{r_m + s J_m}{R_a r_m + k_m^2 + (J_m R_a + I_a r_m) s + s^2 J_m I_a} . \end{aligned} \tag{6.14}$$

Example: Bond Graph with a Bond Loop

As a second example, consider the bond graph in Figure 4.4 depicted in this section in Figure 6.9. This bond graph involves two open loops along a bond loop with mesh two open stubs. The two signal loops of opposite orientation run along the bonds 1 – 2 – 3 – 4 – 5 – 6. Their loop gain is $+1/(s C s I)$. In addition, the bond graph has the flat loops listed in Table 6.1. All three loops touch each other. In the expression of the graph determinant, the gains of the first two flat loops cancel with the gains of the open loops. Hence, the graph determinant is

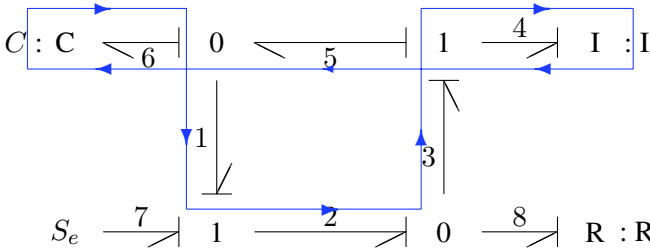


Fig. 6.9 Bond graph with two open signal loops

Table 6.1 Flat loops and their loop gains

Signal loop	Causal path	Loop gain
1	6-5-4	$-\frac{1}{sCsI}$
2	6-1-2-3-4	$-\frac{1}{sCsI}$
3	6-1-2-8	$-\frac{1}{sCR}$

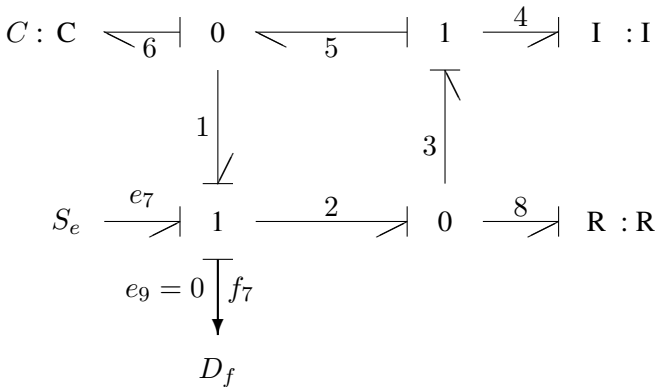


Fig. 6.10 Bond graph of Figure 6.9 with a flow detector D_f sensing the output f_7

$$\Delta = 1 + \frac{1}{sCR} . \tag{6.15}$$

Between the input e_7 and the output f_7 , there are two signal paths running along the causal paths $7 - 2 - 8$ and $7 - 2 - 3 - 4$. A flow detector D_f highlights the location of the output variable f_7 and helps to identify the signal paths between input and output (Figure 6.10).

Their path gain is $1/R$ or $1/(sI)$. The first path is not touched by signal loop number 1. The second path is touched by all signal loops. Hence, we get for the transfer function in question

$$\frac{\mathcal{L}f_7}{\mathcal{L}e_7} = \frac{1}{\Delta} \left[\frac{1}{R} \left(1 - \left(-\frac{1}{sCsI} \right) \right) + \frac{1}{sI}(1 - 0) \right] . \tag{6.16}$$

Of course, the results obtained directly from the bond graphs of the two examples by application of Mason’s loop rule can be checked by Laplace transform of the time

domain equations derived from the bond graphs and by solving them for the ratio of the Laplace transforms of output and input variables to be determined.

In case a model is not structurally controllable or not structurally observable (cf. Sections 6.4 and 6.4.2), Mason's loop rule leads to transfer functions with a denominator of reduced degree because of the cancellation of a null zero by a null pole.

For large bond graphs, the location of all signal loops, the determination as to which of them touch, and the computation of loop gains can be tedious and error-prone. Thus, a computer based symbolic computation of transfer functions is needed. The presented approach based on the direct application of the loop rule to bond graphs is implemented, for instance, in the program ARCHER [5].

6.3 Equations of the Inverse System

A frequent reason for the systematic development of a model is certainly to determine the response of a dynamic system to given external excitations. This kind of problem is usually called the *direct* problem [37]. If the dynamic behaviour of a system has been analysed and if a control system is to be designed, the reverse question of concern is how inputs must be chosen such that a system of known structure produces a required system response. This task is called the *control* problem. Regarding robots, for instance, forces and torques to be applied to joints by motors are wanted so that a robot's tip moves along a prescribed trajectory. A usual approach to answer such a question is to develop a graphical model of a dynamic system, to derive the model equations in a systematic manner and to transform them into the equations of the so-called *inverse* system.

Definition 6.1 (*Inverse system*). Let S denote a system; let \mathbf{u} be the vector of all inputs to S and \mathbf{y} the vector of outputs of S . Then, the *inverse* system S_I is the system that provides the signals \mathbf{u} as a response $\mathbf{y}_I = \mathbf{u}$ to the input signals $\mathbf{u}_I = \mathbf{y}$.

Of course, if a system S is described by the linear constant coefficient state space model

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (6.17a)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (6.17b)$$

then this is easily rewritten as a DAE system for the inverse problem of determining the inputs such that the system response is a given vector \mathbf{y} .

$$\begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{u}} \end{bmatrix} + \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix} \quad (6.18)$$

On the other hand, a bond graph representation offers the possibility to convert the bond graph of a system into the bond graph of the inverse system by just changing computational causalities. That is, the structure of the bond graph is retained

when going from a system to its inverse. If a block diagram representation is used, then the structure is generally *not* preserved because block diagrams represent the computational structure of a system. Apparently, from the bond graph of the inverse system, equations can be derived in the same manner as from the bond graph of the system.

Example: RC-Network and the Analogue Hydraulic Two Tank System

The determination of the equations of the inverse system from its bond graph is illustrated by means of the simple RC-network in Figure 6.1. The corresponding bond graph is displayed in Figure 6.11.

We get the same bond graph for the well known hydraulic system of two coupled tanks depicted in Figure 6.12 if the interconnection between the two tanks is modelled by the resistance R_2 of the valve. The capacitances of the tanks are represented by the C energy stores C_1 and C_2 and possible leakage of the tanks is taken into account by the resistances R_1 and R_3 . The resistor R_4 models the valve in the outlet of the second tank to the return system. The effort source represents the ambient pressure. The hydraulic system modelled this way is the hydraulic analogue of the RC-network in Figure 6.1.

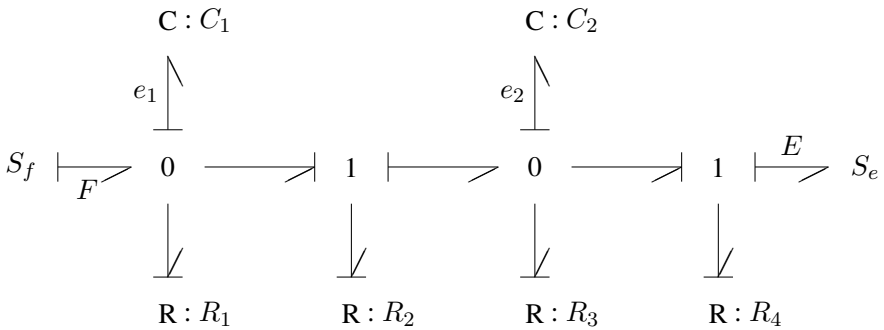


Fig. 6.11 Bond graph of the RC-network in Figure 6.1

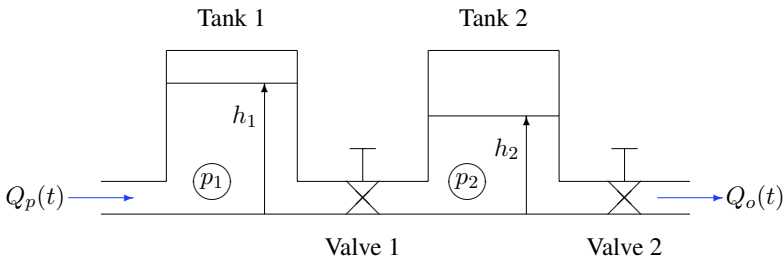


Fig. 6.12 Hydraulic analogue to the RC network in Figure 6.1: Two coupled tanks

Since there are only two energy stores in integral causality, the order of the model is two. Also, the order of the set of ODEs is two, as can be easily checked by applying the SCAP in all derivative mode. For simplicity, we assume linear R elements. Then, the equations derived from the bond graph can be written in state space form

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{C_1} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) & \frac{1}{C_1 R_2} \\ \frac{1}{C_2 R_2} & -\frac{1}{C_2} \frac{1}{R} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} & 0 \\ 0 & \frac{1}{C_2 R_4} \end{bmatrix} \begin{bmatrix} F \\ E \end{bmatrix}, \tag{6.19}$$

where $1/R := 1/R_2 + 1/R_3 + 1/R_4$.

In this model, the outputs of the C energy stores, e_1 and e_2 , are the independent state variables. Conversely, if we want to know the flow f that is necessary to produce a given voltage drop across the capacitor C_1 , or a given pressure, p_1 , at the bottom of the first tank, then we only need to replace the flow source in the bond graph of Figure 6.11 by an effort source that imposes the effort e_s . After adaption of causalities to this change (only the causality at the energy store C_1 must be reversed in this case), we obtain the bond graph of the *inverse* system with respect to the pair of variables $u_1 = F$ and $y_1 = e_1$ (Figure 6.13). Since in the case of a hydraulic system the pressure, p_1 , at the bottom of the first tank is proportional to the fluid level in that tank, the question is which input volume flow Q_i is necessary to maintain a given fluid level in the first tank.

As a consequence of replacing the flow source by an effort source, the C energy store with parameter C_1 has derivative causality. That is, its energy variable is not an independent state variable anymore. Thus, the order of the inverse system in this example is the order of the system reduced by one. In [33] (p. 98), Gawthrop states that, in general, the order of the inverse system (with respect to a pair of an input and an output variable) is equal to the number of energy stores that maintain their causality in the transition from the model of the system to the model of the inverse system. Due to the energy store with derivative causality, the equations of the inverse

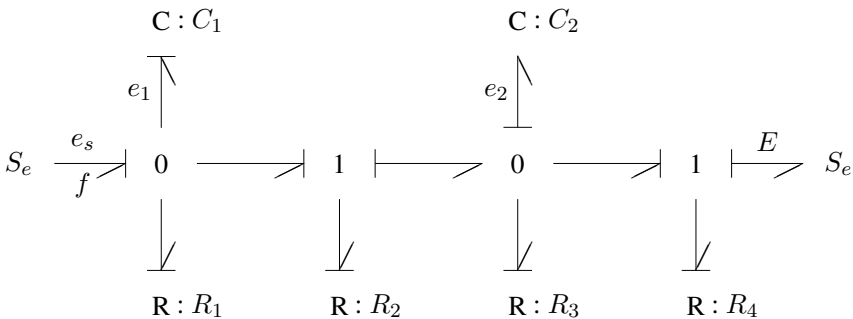


Fig. 6.13 Bond graph of the inverse system with respect to the pair $u_1 = F$ and $y_1 = e_1$

system with respect to the pair $u_1 = F$ and $y_1 = e_1$ can only be written in descriptor form and not in state space form as the model of the system.

$$\begin{bmatrix} C_2 & 0 & 0 \\ 0 & C_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{e}_2 \\ \dot{e}_1 \\ \dot{f} \end{bmatrix} + \begin{bmatrix} \frac{1}{R} & -\frac{1}{R_2} & 0 \\ -\frac{1}{R_2} & \frac{1}{R_2} & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} e_2 \\ e_1 \\ f \end{bmatrix} = \begin{bmatrix} \frac{1}{R_4} E \\ -\frac{1}{R_4} e_s \\ e_s \end{bmatrix} \quad (6.20)$$

Another inverse system exists with respect to the pair $u_1 = F$ and $y_2 = e_2$. In this case, the two variables are not a pair of conjugated power variables associated with one single bond (Figure 6.14). Gawthrop and Smith call such pairs of an input and an output variable *non-collocated* ([37], p. 162). This case of a non-collocated pair of an input and an output variable can be reduced to the case of pairs of collocated input and output variables. This can be achieved by taking into account the input into each source which leads to additional system outputs. Moreover, for each system output not linked with a source, e.g., $y_2 = e_2$, a source is added. In order to assure that these additional sources do not affect the system, they must be sources imposing an input on the system equal to zero. After this preparation, the step from the bond graph of the system to the one of the inverse system can be performed by reversing the causality at certain sources. The bond graph of the inverse system with respect to the non-collocated pair $u_1 = F$ and $y_2 = e_2$ obtained this way is shown in Figure 6.15. In this model, F is an output and e_{2s} is an input not linked to an output flow. The flow into the sink imposing the effort e_{2s} is equal to zero. Thus, both C energy stores have derivative causality. Hence, the mathematical model of this inverse system is also of the form of a DAE system.

In summary, the bond graph of an inverse system with respect to a given pair of an input and an output variable is obtained by adding sources in a first step such that each system output is input to a source. In a second step, the causality of those sources is reversed that correspond to the pair of input and output variables for which the inverse system is to be determined. Consequently, a source does not only impose an input on the system. Simultaneously, it can be a sensor of the conjugate power variable viewed as an output of the system. Gawthrop and Smith, therefore, introduced the so-called *source-sensor* element denoted by the symbol SS [37]. It can replace standard bond graph representations of sources and of ideal sensors. The latter can be represented by a usual effort or flow source with its conjugate power variables set to zero (cf. Section 6.7 on bicausal bond graphs).

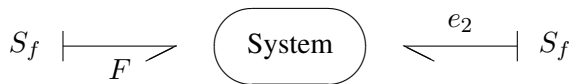


Fig. 6.14 Input and output not being power variables of a single bond

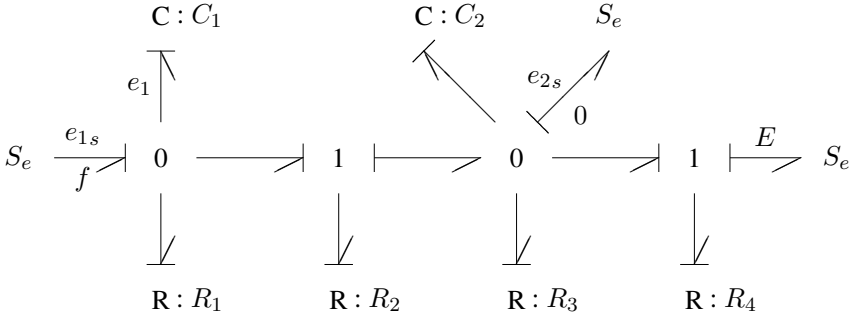


Fig. 6.15 Bond graph of the inverse system with respect to the non-collocated pair $u_1 = F$ and $y_2 = e_2$

In the case of linear time-invariant systems, a transfer function of the inverse system can be deduced either from its underlying DAE system or directly from its bond graph by application of the loop rule. The procedure is the same as for the system itself. Another option is to determine a transfer function of the system and then build its algebraic inverse. Since a transfer function of a system usually has more poles than zeros, the contrary holds for transfer functions of the inverse system. For illustration, let us consider the RC-network in Figure 6.1 once again. For the *direct* problem, the following transfer function can be deduced from the bond graph in Figure 6.11.

$$\frac{\mathcal{L}e_2}{\mathcal{L}f} = \frac{R_4}{a_2s^2 + a_1s + a_0}, \tag{6.21}$$

where

$$a_2 = C_1R_2C_2R_4 \tag{6.22a}$$

$$a_1 = (R_2 + R_4 + \frac{R_2R_4}{R_3})C_1 + \frac{R_2R_4}{R_1}C_2 + C_2R_4 \tag{6.22b}$$

$$a_0 = 1 + \frac{R_4}{R_3} + (R_2 + R_4 + \frac{R_2R_4}{R_3})\frac{1}{R_1}. \tag{6.22c}$$

For the inverse problem, in other words, the question as to which flow f is required to assure a given effort e_{2s} at the second C energy store (Figure 6.15), we just get the reciprocal of the transfer function in Equation 6.21, viz. a polynomial of degree 2.

6.4 Structural Controllability and Observability

A causally completed bond graph can provide many kinds of information before any equations are formulated. Its inspection can reveal possible inconsistencies in the modelling assumptions, or provide information about independent state variables and the form of the underlying mathematical model. Moreover, as mentioned in Section 3.1, properties, e.g., structural controllability and observability, playing an essential role in the context of control system design, can be directly analysed by inspection of the causally completed bond graph.

In the next two sections, two criteria for structural controllability and structural observability are briefly illustrated by means of examples. Both criteria can be directly applied to a causally completed bond graphs. The interested reader may find more information, e.g., in [22] (Chapter 8) and in [58, 63, 72]. It is assumed that the structure of a bond graph and the assignment of causalities remains invariant with regard to time. There are no components or phenomena in the system under consideration that give rise to the model abstraction of instantaneous changes from one state to another. Modelling of such systems is considered in Chapter 7. Rahmani and Dauphin-Tanguy have extended the analysis of structural controllability and structural observability to bond graphs of systems containing switching elements (cf., e.g. [59]).

6.4.1 Structural Controllability

With regard to the notion of structural controllability, first, it is necessary to provide some definitions.

Definition 6.2 (*Structural matrix*). A *structural matrix* $[\mathbf{A}]$ is determined by the number and the position of its non-zero entries. The latter are considered to be of indeterminate value and independent of each other. They are denoted by the symbol $*$.

Remark 6.1. The structural matrix is a class of structurally equivalent matrices. If non-zero entries are replaced by the value one, then all matrices of the class yield the same so-called Boolean matrix.

Definition 6.3 (*Admissible numerical realisation of a structural matrix*). Let $[\mathbf{A}]$ be a structural matrix. A matrix \mathbf{A} is called an *admissible numerical realisation* of $[\mathbf{A}]$ (for short $\mathbf{A} \in [\mathbf{A}]$) if all entries of indeterminate value are replaced by entries of particular numerical values [58].

Example

Let

$$[\mathbf{A}] := \begin{bmatrix} 0 & * \\ * & 0 \end{bmatrix},$$

then

$$\mathbf{A} := \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

is an admissible numerical realisation of $[\mathbf{A}]$.

In the following, we will confine ourselves to linear time-invariant MIMO systems given by the equations

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (6.23a)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (6.23b)$$

where \mathbf{x} denotes the vector of state variables, \mathbf{u} the vector of excitation functions and \mathbf{y} the vector of output variables. \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} are constant coefficient matrices, where $\mathbf{A} \in \mathbb{R}^n \times \mathbb{R}^n$, $\mathbf{B} \in \mathbb{R}^n \times \mathbb{R}^m$, $\mathbf{C} \in \mathbb{R}^p \times \mathbb{R}^n$, $\mathbf{D} \in \mathbb{R}^p \times \mathbb{R}^m$.

Definition 6.4 (*Complete state controllability*). The state $\mathbf{x}(t)$ is said to be controllable at time $t = t_0$ if there exists a piecewise continuous input function $\mathbf{u}(t)$ that causes the state vector to move to any final value $\mathbf{x}(t_f)$ in a finite time $t_f > t_0$. If each state \mathbf{x}_0 is controllable, then the system is said to be completely state controllable [42].

To check for the complete state controllability of a linear time-invariant system, Kalman has given the following sufficient and necessary condition.

Theorem 6.1 (Kalman). *A linear time-invariant MIMO system is completely state controllable if and only if the so-called $n \times (nm)$ controllability matrix*

$$[\mathbf{B} | \mathbf{A}\mathbf{B} | \mathbf{A}^2\mathbf{B} | \dots | \mathbf{A}^{n-1}\mathbf{B}]$$

has rank n .

Remark 6.2. Mathematical software such as Scilab [67], GNU Octave [3] or the (commercial) MATLAB[®] Control System Toolbox [2] provide functions to construct the controllability matrix.

For instance, the Octave function `is_controllable(A, B, tol)` returns the value 1 if the LTI system given by the matrices \mathbf{A} and \mathbf{B} is controllable, otherwise it returns the value 0. Note that the determination of the rank of the controllability matrix is sensitive to errors in the matrix entries and to roundoff errors.

Definition 6.5 (*Structurally complete state controllability*). A linear time invariant MIMO system with matrices \mathbf{A} and \mathbf{B} is said to be *structurally* completely state

controllable if there is at least one numerical admissible realisation \mathbf{A} of the structural matrices $[\mathbf{A}]$ and one admissible realisation \mathbf{B} of $[\mathbf{B}]$ for which the system is completely state controllable [68].

Having introduced the notion of structural controllability, a criterion is given that enables one to decide whether a linear time-invariant system is structurally controllable by checking its causally completed bond graph. It is assumed that

- energy sources, energy stores and dissipators in the bond graph are 1-port elements,
- there are only power bonds in the bond graph, no signals,
- all elements are linear.

That is, the mathematical model, corresponding to the causally completed bond graph is of the form of Equations 6.23a and 6.23b.

Theorem 6.2 (Sueur and Dauphin-Tanguy, 1991). *A linear time-invariant MIMO system with the $n \times n$ state matrix \mathbf{A} and the $n \times m$ matrix \mathbf{B} is structurally state controllable if and only if the following two conditions are satisfied.*

1. *Given that the preferred causality applied to the energy stores in the bond graph is integral causality, then for each energy store in integral causality there exists a causal path from a controlled source to the energy store. In other words, all states are input-reachable.*
2. *In a bond graph with preferred integral causality, all energy stores in integral causality take derivative causality if derivative causality is the preferred causality. If this condition is not met directly, then it is achieved by replacing some controlled sources in appropriate places by their dual.*

Remark 6.3. • If $BG - \text{rank} \mathbf{A} = n$, then one properly located input source is sufficient to control the system.

- If $BG - \text{rank} \mathbf{A} = n - k$, where $1 \leq k < n$ denotes the number of storage ports that must take integral causality when derivative causality is the preferred causality, then at least k input sources properly located are necessary to control the system.
- Proper location means that if the sources are replaced by their duals, then all storage ports are in derivative causality, which means that the system is state controllable by the added sources [24, 72].

Example

For illustration, consider an example. As both C stores take integral causality, the order of the model is two. In the bond graph depicted in Figure 6.16, there is a causal path 1 – 2 from the flow source to the C element with capacitance C_1 and a causal path 1 – 2 – 3 – 4 – 5 from the source to the other capacitor, C_2 , highlighted by additional lines. That is, the first condition in Theorem 6.2 is fulfilled.

Now, if derivative is the preferred causality, then one energy store must retain integral causality (Figure 6.17), which means that $BG - \text{rank} \mathbf{A} = 2 - 1 = 1$. That is,

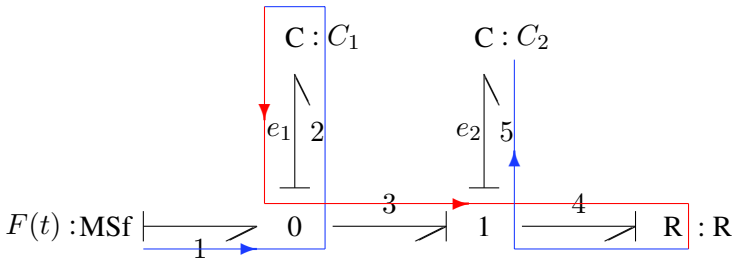


Fig. 6.16 Bond graph with preferred integral causality

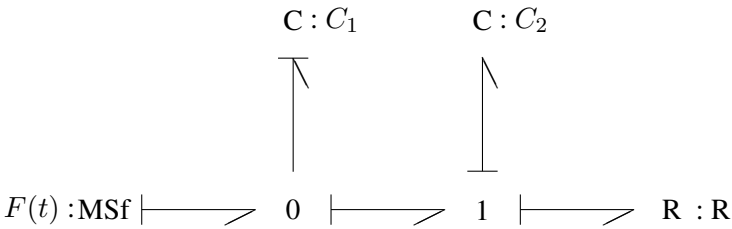


Fig. 6.17 Same bond graph with preferred derivative causality

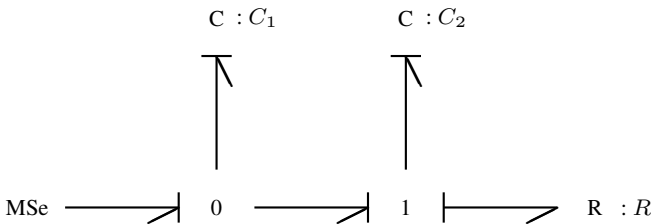


Fig. 6.18 Bond graph with preferred derivative causality after replacement of the source by its dual

at least one input source is necessary to control the system. However, the remaining integral causality can be removed by replacing the flow source by an effort source (Figure 6.18), which means that the existing source is properly located. As the second condition in Theorem 6.2 also holds, the system represented by the bond graph is structurally controllable. In fact, from the bond graph shown in Figure 6.16, the following state equations can be derived.

$$\frac{d}{dt} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{C_1 R} & \frac{1}{C_1 R} \\ \frac{1}{C_2 R} & -\frac{1}{C_2 R} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{C_1} \\ 0 \end{bmatrix}}_{\mathbf{B}} [F] \quad (6.24)$$

From the state matrix, it can be seen that $\text{rank} \mathbf{A} = 1$. With these matrices, we obtain the controllability matrix

$$[\mathbf{B} | \mathbf{A}\mathbf{B}] = \begin{bmatrix} \frac{1}{C_1} & -\frac{1}{RC_1^2} \\ 0 & \frac{1}{RC_1 C_2} \end{bmatrix}. \quad (6.25)$$

Its rank is two. Consequently, according to Kalman's criterion (Theorem 6.1), the system is completely state controllable. Thus, according to definition 6.5, it is structurally controllable.

A simple check to be performed on the bond graph whether a system is structurally *not* controllable has been given by Rosenberg and Andry in as early as 1979 [63]. The procedure is simple.

1. All sources are replaced by resistors.
2. The standard causality assignment procedure (SCAP) of Karnopp and Rosenberg is applied to the modified bond graph.

If an energy store in the modified bond graph receives derivative causality, then the system is *not* structurally controllable.

This criterion, however, is only sufficient. Consider the simple bond graph shown in Figure 6.19. After replacement of the effort source by a resistor, both energy stores can be assigned preferred integral causality. In this case, the test of Rosenberg and Andry fails. It does not enable one to decide whether the system is not structurally controllable.

In contrast, the criterion of Sueur and Dauphin-Tanguy (Proposition 6.2) yields that the system is *not* structurally controllable. First, it can be immediately seen that there are causal paths from the effort source to both capacitors. If derivative causality is the preferred causality, one energy store retains integral causality (Figure 6.20) which cannot be removed by replacing the effort source by its dual. On the contrary, neither energy stores can be assigned preferred derivative causality (Figure 6.21). According to Theorem 6.2, the system is *not* structurally controllable.

6.4.2 Structural Observability

Clearly, for the control of a system, its state $\mathbf{x}(t)$ at a current time instant, t , must be known. However, in general, not all state variables can be measured. Therefore, it is necessary to deduce a system state from the output variables measured for a limited

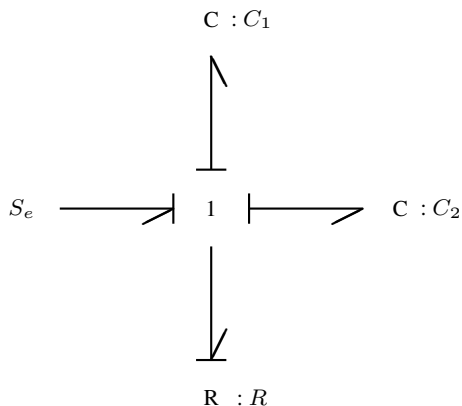


Fig. 6.19 Bond graph of a simple linear system

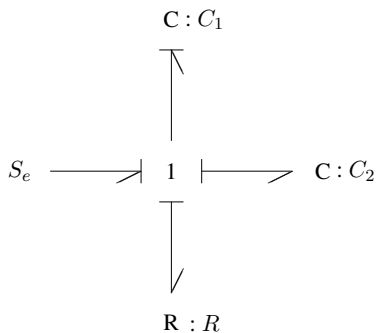


Fig. 6.20 Bond graph after assignment of preferred derivative causality

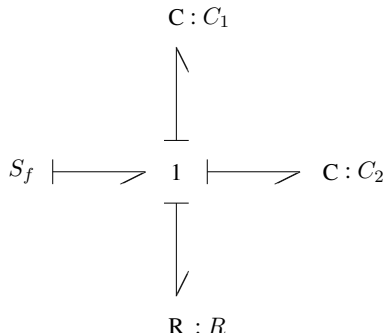


Fig. 6.21 Bond graph after replacement of the effort source by its dual

period of time and the knowledge of the input variables. This leads to the notion of *observability*.

Definition 6.6 (Complete observability). A system is said to be completely observable if for each initial state $\mathbf{x}(t_1)$ at time instant t_1 , there is a time $t_2 > t_1$ such that $\mathbf{x}(t_1)$ can be found from the known input vector $\mathbf{u}(t)$ and the output vector $\mathbf{y}(t)$ measured over the interval of time $[t_1, t_2]$.

Controllability, discussed in the previous section and observability are two dual concepts. For checking the observability of a linear time invariant MIMO system, Kalman has given a criterion that is similar in its structure to the one for controllability and is also necessary and sufficient.

Theorem 6.3 (Kalman). *A linear time-invariant MIMO system is completely observable if and only if the so-called $n \times (np)$ observability matrix*

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}$$

has rank n .

Clearly, the rank of the observability matrix depends on the numerical values of the model parameters. However, these can be subject to uncertainties. Furthermore, the model can be ill conditioned. Consequently, the application of a criterion that is based on numerical evaluations cannot be considered a robust indicator of complete observability. Thus, the question arises whether the elements of a model and the way they are connected permits observability independent of actual numerical values of the model parameters. Such a model property is captured by the notion of *structural* observability.

Definition 6.7 (*Structurally complete observability*). A linear time invariant MIMO system with matrices \mathbf{A} and \mathbf{C} is said to be *structurally* completely observable if there are numerical admissible realisations $\mathbf{A} \in [\mathbf{A}]$ and $\mathbf{C} \in [\mathbf{C}]$ for which the system is completely observable.

An alternative formulation is

Definition 6.8 (*Structurally complete observability*). A linear time invariant MIMO system with the structural matrices $[\mathbf{A}]$ and $[\mathbf{C}]$ is *structurally* completely observable if it is completely observable for almost all values of the model parameters.

As for structural controllability, structural observability can also be checked directly on a causally completed bond graph of a linear time-invariant system. Again, it is assumed that

- energy sources, energy stores and dissipators in the bond graph are 1-port elements,
- there are only power bonds in the bond graph, no signals and
- all elements are linear.

For checking structural observability directly on a causal bond graph, Sueur and Dauphin-Tanguy have given the following necessary and sufficient criterion.

Theorem 6.4 (Sueur and Dauphin-Tanguy, 1991). *A linear time invariant MIMO system with matrices \mathbf{A} and \mathbf{C} is structurally observable if and only if the following two conditions are satisfied.*

1. *Given that the preferred causality applied to the energy stores in the bond graph is integral causality, then every energy store in integral causality must have at least one causal path to a sensor.*

2. In a bond graph with preferred integral causality, all energy stores in integral causality must take derivative causality if derivative causality is the preferred causality. If this condition is not met directly, then it is achieved by replacing some sensors in appropriate places by their dual [24, 72].

Remark 6.4. If the two conditions of Theorem 6.4 are met, then $\text{rank}\mathbf{O} = n$. If $\text{rank}\mathbf{A} = n$, then one observer, suitably placed, is sufficient to assure complete observability. Otherwise, if $\text{rank}\mathbf{A} = q < n$, then $n - q$ observers are needed to guarantee observability. They are to be placed such that the first condition of Theorem 6.4 is satisfied.

Example: Hydraulic Two Tank System

For illustration, consider the example of a simple hydraulic two tanks system depicted in Figure 6.12. We assume that the valve at the outlet of the second tank is user controlled and that the pressure, p_2 , in the second tank is sensed. Furthermore, it is assumed that the model equations have been linearised in the vicinity of some operating point.

A bond graph with preferred integral causality is displayed in Figure 6.22. The sensor of the pressure, p_2 , in the second tank has been modelled by means of a zero flow sink. As both storage elements can take integral causality, the order of the model, n , is two.

If preferred derivative causality is assigned, then both C elements take derivative causality. That is, the bond graph rank of the state matrix equals the order of the model, $n = 2$. Consequently, one controlled source is sufficient to control the system.

As can be seen from the bond graph in Figure 6.22, there are causal paths from the controlled flow source representing the pump to the C elements. These paths have been pointed out by additionally indicating signal forward paths in thin lines.

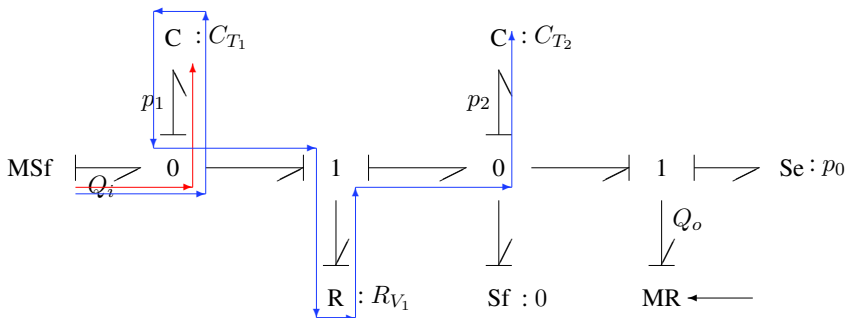


Fig. 6.22 Bond graph of the hydraulic two tanks system in Figure 6.12: causal paths from the controlled flow source to the energy stores

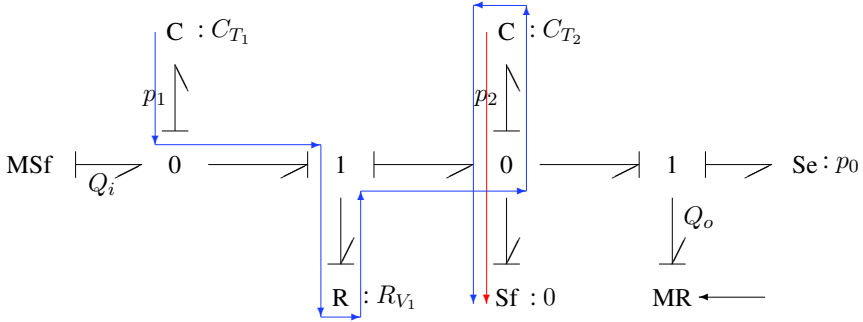


Fig. 6.23 Bond graph of the hydraulic two tanks system in Figure 6.12: causal paths from the storage elements to the sensor

thermore, as both storage elements take derivative causality when derivative causality is the preferred causality, there is no need to replace the controlled flow source by its dual. As a result, the corresponding linear time-invariant MIMO system is structurally controllable.

Now, for an observability check, consider the bond graph in Figure 6.23. As the bond graph rank of the state matrix equals two, we already know that one sensor is sufficient to observe the system.

According to Theorem 6.4, there is a causal path from each of the two C stores to the sensor. These paths are pointed out by additional thin lines indicating a signal forward path. Furthermore, both energy stores take derivative causality when derivative causality is the preferred causality. That is, both conditions of Theorem 6.4 are fulfilled. Hence, the two tank system with a pressure sensor at the second tank is observable.

As a remark to Theorem 6.4, it has been mentioned that if both conditions are satisfied, then the rank of the observability matrix equals the order of the model. In fact, from the bond graph of Figure 6.22, the following state space model can be derived.

$$\underbrace{\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -\frac{1}{C_1 R_{V_1}} & \frac{1}{C_1 R_{V_1}} \\ \frac{1}{C_2 R_{V_1}} & -\frac{1}{C_2 R} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \frac{1}{C_1} & 0 \\ 0 & \frac{1}{C_2 R} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} Q_i \\ p_0 \end{bmatrix}}_{\mathbf{u}} \quad (6.26a)$$

$$\underbrace{\begin{bmatrix} p_2 \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 & 0 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} Q_i \\ p_0 \end{bmatrix}}_{\mathbf{u}}, \quad (6.26b)$$

where $1/R := 1/R_{V_1} + 1/R_{V_2}$. It can be easily checked that $\text{rank}\mathbf{A} = 2$. Furthermore,

$$\text{rank}\mathbf{O} = \text{rank} \begin{bmatrix} 0 & 1 \\ \frac{1}{C_2 R_{V_1}} & -\frac{1}{C_2 R} \end{bmatrix} = 2. \quad (6.27)$$

6.5 Parameter Sensitivities

In this section, we will see that bond graphs can also be used to determine unnormalised frequency domain sensitivities in symbolic form. With regard to this aim, several approaches based on *pseudo bond graphs* have been reported in the literature [19, 34, 57]. In these pseudo bond graphs, the variables associated with the bonds are not power variables, but first order partial derivatives of the effort and the flow of a bond with respect to a parameter. That is, these pseudo bond graphs reflect relations between first order sensitivities. In contrast, in this section, a so-called *incremental* bond graph is considered. Incremental bond graphs for the determination of unnormalised frequency domain sensitivities was proposed by the author at the 2001 International Conference on Bond Graph Modelling (ICBGM 2001) [12]. An elaborated article was published in a journal in 2002 [13]. The presentation in this section follows [13].

Unlike *pseudo bond graph* approaches to the determination of sensitivities, incremental bond graphs represent relations between the increments $(\Delta e)(t)$, $(\Delta f)(t)$ of true power variables $e(t)$ and $f(t)$, $t \in \mathbb{R}$, $t \geq 0$ due to time-independent parameter perturbations. The latter are explicitly displayed in the incremental bond graph by means of sources modulated by a nominal power variable of the initial bond graph.

Incremental bond graphs can be constructed in a systematic manner from an initial bond graph by replacing the bond graph elements by their corresponding incremental models. For linear elements, their corresponding model in the incremental bond graph also has a linear characteristic. By deriving the system equations in symbolic state space form from the incremental bond graph in the same way they are derived from the initial bond graph, the sensitivity matrix of the system can be set up in symbolic form. Its entries are transfer functions depending on the nominal parameter values and on the nominal states and the inputs of the original model. The sensitivities can be determined automatically, e.g., by means the bond graph pre-processor CAMP-G[®] [20], or the widely used program MATLAB[®] [6] along with the Symbolic Math Toolbox for symbolic mathematical calculation. No particular software is needed for the approach proposed.

The sensitivity of the behaviour in the time or in the frequency domain with regard to parameter changes is expressed by partial derivatives. If y denotes an output of a lumped parameter system model and Θ a parameter, e.g., a resistance R , then

$$\frac{\partial y}{\partial \Theta} \times \frac{\Theta}{y}$$

is called the relative or normalized sensitivity of y with respect to the parameter Θ . The partial derivative $\partial y/\partial\Theta$ is called the *unnormalised* sensitivity. The objective in this section is the determination of unnormalised sensitivities. In the following, the adjective unnormalised is omitted. Sensitivities can be calculated in the time domain as well as in the frequency domain. In the second case, they are of complex value and depend on the frequency, ω , in general. In the following, we focus on frequency domain sensitivities.

A brute force method to the determination of sensitivities is to replace the differentials by differences. Performing numerical differentiation, however, has the disadvantage of poor accuracy because the difference of the computer representations of two nearly equal numbers results in a loss of information. Moreover, it is costly with regard to computation time because for each frequency and for each parameter that has changed a simulation run must be performed. To avoid numerical differentiation, a common approach in network analysis, is to construct a so-called *adjoint network* \hat{N} from an initial network N and to exploit Tellegen's theorem for an efficient calculation of sensitivities [27].

The incremental bond graph approach presented in the following starts from the observation that parameter changes cause perturbed power variables $e + \Delta e$ and $f + \Delta f$ at the power ports of an element [12, 13]. Both variables, the flow and the effort, at a power port are effected by a parameter variation due to the interaction with the rest of the system. In principle, a parameter change effects all variables in the system (The variables $e(t)$ and $f(t)$ denote nominal values, whereas $(\Delta e)(t)$ and $(\Delta f)(t)$ denote deviations from nominal values at time t). We will construct a *true* bond graph \tilde{G} for the incremental power variables Δe and Δf from the initial bond graph G . The derived bond graph \tilde{G} is called the associated *incremental* bond graph. Unnormalised sensitivities can be determined in symbolic form by deriving the system equations from the *incremental* bond graph. This approach is in accordance with the incremental network approach in network theory. The advantage over an approach corresponding to the adjoint network approach is that the incremental bond graph provides better insight into the effect of parameter changes because these are visualised in the incremental bond graph by sources.

6.5.1 Incremental Models of Linear Bond Graph Elements

For the sake of simplicity of the presentation, energy sources, energy stores and resistors are assumed to be 1-port elements with a linear constitutive law, whereas transformers and gyrators are assumed to have two ports. Power variables $e_n(t)$ and $f_n(t)$, $t \in \mathbb{R}$, $t \geq 0$, and parameters in the linear constitutive relation between the power variables of a port denote *nominal* values (indicated by a subscript n). Deviations from the nominal values of power variables, i. e., their increments, are denoted by $(\Delta e)(t)$ or $(\Delta f)(t)$, that is, $(\Delta e)(t) := e(t) - e_n(t)$, $(\Delta f)(t) := f(t) - f_n(t)$. Note that the increment Δv of a function v is different from its total differential dv , in general.

Now, consider a linear 1-port resistor with the nominal resistance R_n . The basic idea is that a time independent parameter variation ΔR causes perturbed power port variables $(e_{R_n} + \Delta e)(t)$ and $(f_{R_n} + \Delta f)(t)$ at each time t . Hence, the constitutive relation

$$e_{R_n}(t) = R_n \times f_{R_n}(t) \tag{6.28}$$

becomes

$$(e_{R_n} + \Delta e_R)(t) = (R_n + \Delta R) \times (f_{R_n} + \Delta f_R)(t) . \tag{6.29}$$

Substituting Equation 6.28 into Equation 6.29 and neglecting the higher-order term $(\Delta R) \times (\Delta f_R)$ results in a linear relation between the incremental power variables Δe_R and Δf_R

$$(\Delta e_R)(t) = R_n \times (\Delta f_R)(t) + f_{R_n}(t) \times \Delta R , \tag{6.30}$$

which could also be obtained by taking the total differential of the product $R \times f_R$. With regard to the determination of parameter sensitivities, the neglect of higher-order terms such as $(\Delta R) \times (\Delta f_R)$ is reasonable since small parameter perturbations, resulting in small deviations of power variables from nominal values are of interest. Hence, a *first order approximation* of the increment $(\Delta v)(t)$ of a power variable v at time instant t is justified. Equation 6.30 can be represented by the bond graph depicted in Figure 6.24.

As can be seen from Figure 6.24, the incremental model associated with a resistor differs from the R element only by an additional effort source modulated by the nominal flow $f_{R_n}(t)$. The latter originates from the initial bond graph.¹ The ad-

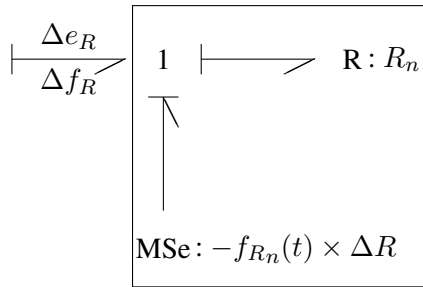


Fig. 6.24 First order incremental bond graph of a linear 1-port resistor

¹ Apparently, Equation 6.30 could be transformed into a relation between the partial derivatives of the power variables with respect to the parameter R .

$$\frac{\partial e_R}{\partial R}(t) = R_n \frac{\partial f_R}{\partial R}(t) + f_{R_n}(t)$$

If the partial derivatives of the power variables are taken as the effort and the flow of a bond, then this equation may be represented as a *pseudo bond graph* model that has the same structure as the incremental model in Figure 6.24 (cf. [19]).

ditional source does not affect initial computational causalities. Note that the product of the incremental power variables $(\Delta e_R)(\Delta f_R)$ has the physical dimension of power. Actually, it is only a part of the power change, $\Delta \mathcal{P}$, due to the parameter change ΔR .

$$\begin{aligned}\Delta \mathcal{P} &= (e_R + \Delta e_R) \times (f_R + \Delta f_R) - e_R \times f_R \\ &= (\Delta e_R) \times (\Delta f_R) + \text{further terms}\end{aligned}\quad (6.31)$$

Similar results can be obtained for linear 1-port energy stores. For instance, for a linear C element with the nominal capacitance C_n and the constitutive relation

$$q_n(t) = C_n \times e_{C_n}(t), \quad (6.32)$$

a time-independent perturbation ΔC leads to the equation

$$q_n(t) + (\Delta q)(t) = (C_n + \Delta C)(e_{C_n}(t) + \Delta e_C(t)). \quad (6.33)$$

If the higher-order term $(\Delta C)(\Delta e_C)$ is neglected, then we get the equation

$$\Delta e_C(t) = \frac{1}{C_n} \times \Delta q(t) - \frac{\Delta C}{C_n} \times e_{C_n}(t). \quad (6.34)$$

It can be represented by a bond graph as depicted in Figure 6.25.

As for the incremental model of an R element, the additional modulated source in the incremental model of an energy store does not affect computational causalities. However, contrary to an energy store in the initial bond graph, its associated incremental model is not energy conservative. Apparently, the modulated source can be split into an independent source representing the parameter change and a modulated transformer, as shown in Figure 6.26 for the case of an inertia. As for resistors and energy stores, incremental models can be easily developed for all other bond graph elements in the same manner. The only difference between a bond graph element and its associated incremental model is the modulated source attached to a junction that reflect for the parameter variation. The incremental models corresponding to

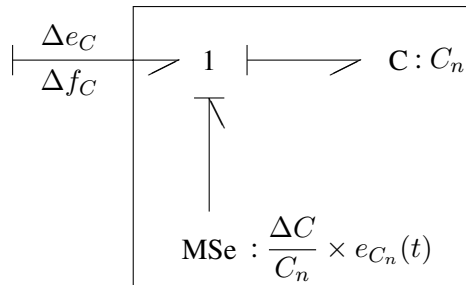


Fig. 6.25 First order incremental bond graph of a linear 1-port capacitor

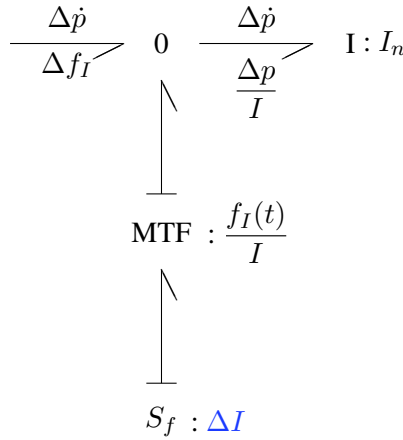


Fig. 6.26 First order incremental bond graph of a linear 1-port inertia

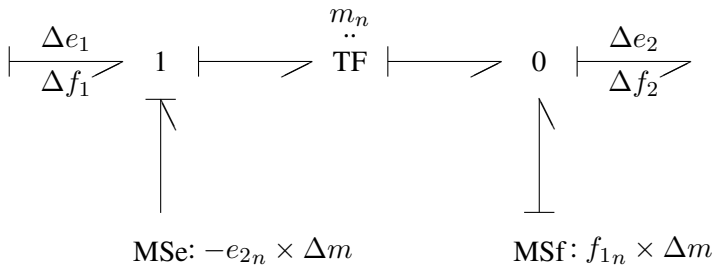


Fig. 6.27 First order incremental bond graph of a 2-port transformer

a 2-port TF or a GY element have two sources. That is, they are not power conservative. The incremental model of a 2-port transformer is shown in Figure 6.27. Accordingly, Figure 6.28 shows the incremental bond graph model of a 2-port gyrator.

Because 1- and 0-junctions are linear, their associated incremental model is identical to the junction. Consequently, a bond graph in which all elements have been replaced by their associated incremental model retains the structure of the initial bond graph with nominal parameters. Only the sources and their location are different. Apparently, a parameter independent source vanishes in the incremental bond graph. In other words, a parameter independent source in the initial bond graph transforms into a source of the same type with vanishing output. The incremental bond graph of a linear system accounts for all parameter changes. Due to the linearity of the junction structure, the result of changes in several parameters is the superposition of all effects due to a change in a single parameter. Thus, if the effect of a change in only one parameter is of interest, it is sufficient to replace only the affected element by its incremental model.

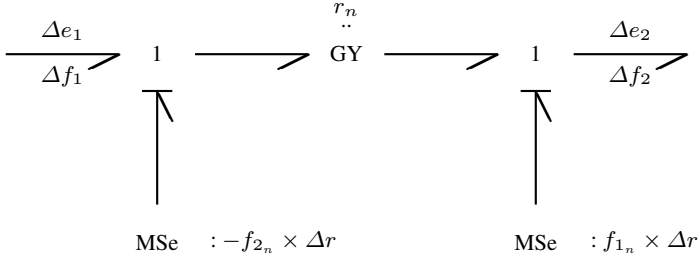


Fig. 6.28 First order incremental bond graph model of a 2-port gyrator

6.5.2 Derivation of Parameter Frequency Domain Sensitivities from an Incremental Bond Graph

Let Θ be the vector of all *uncertain* parameters, Θ_n the vector of nominal parameters, $\mathbf{x}(t, \Theta)$ the state vector of a system with uncertain parameters, and $\Delta \mathbf{x} := \mathbf{x}(t, \Theta) - \mathbf{x}(t, \Theta_n)$ the increment of \mathbf{x} due to parameter uncertainties $\Delta \Theta$. Thus, $\mathbf{x}_n(t) := \mathbf{x}(t, \Theta_n)$ denotes the state vector corresponding to the nominal bond graph, whereas $\Delta_1 \mathbf{x}$ means a first order approximation of the increment $\Delta \mathbf{x}$ that neglects higher order terms in the power series expansion. The nominal model is assumed to be linearised and described by the equations

$$\dot{\mathbf{x}}_n(t) = \mathbf{A}_n \mathbf{x}_n(t) + \mathbf{B}_n \mathbf{u}(t) \quad (6.35a)$$

$$\mathbf{y}_n(t) = \mathbf{C}_n \mathbf{x}_n(t) + \mathbf{D}_n \mathbf{u}(t) \quad (6.35b)$$

with constant coefficient matrices $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{D}_n$ depending on Θ_n and having appropriate dimensions. In general, the dependencies of the matrix entries from the components of the parameter vector Θ_n are nonlinear. The vectors \mathbf{u} and \mathbf{y} denote the vector of inputs and the vector of outputs of the nominal system. Then, the state equations derived from the incremental bond graph may be written in the form

$$\Delta_1 \dot{\mathbf{x}}(t) = \mathbf{A}_n \Delta_1 \mathbf{x} + \tilde{\mathbf{B}}(\mathbf{x}_n(t), \mathbf{u}(t), \Theta_n) \Delta \Theta \quad (6.36a)$$

$$\Delta_1 \mathbf{y}(t) = \mathbf{C}_n \Delta_1 \mathbf{x} + \tilde{\mathbf{D}}(\mathbf{x}_n(t), \mathbf{u}(t), \Theta_n) \Delta \Theta \quad (6.36b)$$

because the incremental bond graph retains the structure of the initial bond graph. It differs only by the sources. The matrices $\tilde{\mathbf{B}}(\mathbf{x}_n(t), \mathbf{u}(t), \Theta_n)$ and $\tilde{\mathbf{D}}(\mathbf{x}_n(t), \mathbf{u}(t), \Theta_n)$ are obtained by differentiation of the Equations 6.35a and 6.35b with respect to Θ . Since both matrices depend on vectors $\mathbf{x}_n(t)$ and $\mathbf{u}(t)$, they are time dependent.

$$\tilde{\mathbf{B}}(t) = \left. \frac{\partial}{\partial \Theta} (\mathbf{A}_n(\Theta) \mathbf{x}_n(t) + \mathbf{B}_n(\Theta) \mathbf{u}(t)) \right|_{\Theta = \Theta_n} \quad (6.37a)$$

$$\tilde{\mathbf{D}}(t) = \left. \frac{\partial}{\partial \boldsymbol{\Theta}} (\mathbf{C}_n(\boldsymbol{\Theta})\mathbf{x}_n(t) + \mathbf{D}_n(\boldsymbol{\Theta})\mathbf{u}(t)) \right|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}_n} \quad (6.37b)$$

Laplace transform of Equations 6.36a and 6.36b yields

$$\begin{aligned} (\mathcal{L}\Delta_1\mathbf{y})(s) &= (\Delta_1\mathcal{L}\mathbf{y})(s) \\ &= [\mathbf{C}_n(s\mathbf{I} - \mathbf{A}_n)^{-1}(\mathcal{L}\tilde{\mathbf{B}})(s) + (\mathcal{L}\tilde{\mathbf{D}})(s)]\Delta\boldsymbol{\Theta}, \end{aligned} \quad (6.38)$$

where $s \in \mathbb{C}$.

For infinitesimal small parameter variations, $\Delta\boldsymbol{\Theta}$, the matrix in Equation 6.38 is just the sensitivity matrix $\mathbf{S}(s)$

$$\mathbf{S}(s) := \partial\mathcal{L}\mathbf{y}/\partial\boldsymbol{\Theta} = \mathbf{C}_n(s\mathbf{I} - \mathbf{A}_n)^{-1}(\mathcal{L}\tilde{\mathbf{B}})(s) + (\mathcal{L}\tilde{\mathbf{D}})(s) \quad (6.39)$$

to be determined.

That is, once the matrices \mathbf{B}_n and \mathbf{D}_n in Equations 6.35a and 6.35b have been set up in symbolic form from the initial nominal bond graph by means of some suitable software, they could be symbolically differentiated with respect to the parameters yielding the matrices $\tilde{\mathbf{B}}$, $\tilde{\mathbf{D}}$ (Equations 6.37a and 6.37b). After their Laplace transform, the sensitivity matrix could be computed according to Equation 6.39.

In practice, the sensitivity matrix is not computed exactly that way. Instead, as has been shown by Borutzky and Granda in [13], software, e.g., the bond graph pre-processor CAMP-G[®][20] or MATLAB[®] [6] along with the Symbolic Math Toolbox can be used to automatically derive from an incremental bond graph the sensitivity matrix in symbolic form. In fact, if the outputs of the modulated sources in the incremental bond graph are combined into a vector \mathbf{w} , matrices \mathbf{A}_n , \mathbf{B}^* , \mathbf{C}_n , \mathbf{D}^* depending on $\boldsymbol{\Theta}_n$ can be automatically generated such that

$$\Delta_1\dot{\mathbf{x}}(t) = \mathbf{A}_n\Delta_1\mathbf{x}(t) + \mathbf{B}^*(\boldsymbol{\Theta}_n)\mathbf{w}(t) \quad (6.40a)$$

$$\Delta_1\mathbf{y}(t) = \mathbf{C}_n\Delta_1\mathbf{x}(t) + \mathbf{D}^*(\boldsymbol{\Theta}_n)\mathbf{w}(t). \quad (6.40b)$$

By looking at the modulated sources in the incremental models of the bond graph elements, a diagonal matrix $\mathbf{W}(t, \boldsymbol{\Theta}_n)$ can be easily found such that

$$\mathbf{w}(t) = \mathbf{W}(t, \boldsymbol{\Theta}_n)\Delta\boldsymbol{\Theta}. \quad (6.41)$$

Substituting Equation 6.41 into (6.40a) and comparing the result with Equation 6.36a yields

$$\tilde{\mathbf{B}}(t) = \mathbf{B}^*(\boldsymbol{\Theta}_n)\mathbf{W}(t, \boldsymbol{\Theta}_n). \quad (6.42)$$

A similar result is obtained for the matrix $\tilde{\mathbf{D}}$. With both results, Equation 6.39 for the sensitivity matrix reads

$$\mathbf{S}(s) = [\mathbf{C}_n(s\mathbf{I} - \mathbf{A}_n)^{-1}\mathbf{B}^* + \mathbf{D}^*](\mathcal{L}\mathbf{W})(s). \quad (6.43)$$

The matrices \mathbf{A}_n and \mathbf{C}_n are generated from the initial nominal bond graph, while matrices \mathbf{B}^* and \mathbf{D}^* are obtained in the same manner from the incremental bond

graph. The matrix operations in Equation 6.43 may be performed in a MATLAB[®] script.

Note that the ratio of an incremental variable $\mathcal{L}\Delta y$ and a parameter change $\Delta\theta_i$ is a transfer function between the output variable $\mathcal{L}\Delta y$ and the input variable $\Delta\theta_i$ if modulated sources representing parameter changes are split into an independent source and a modulated transformer (cf. Figure 6.27). As Brown has shown, transfer functions can be determined by identifying causal paths and determining loop gains as needed in Mason's loop rule (see Section 6.2). That is, instead of setting up matrices in symbolic form from the initial bond graph and in the same manner from its associated incremental bond graph as shown above, alternatively, sensitivities could be directly derived from the incremental bond graph by application of Mason's rule.

Another possible approach is to represent dependencies between sensitivities in a *pseudo bond graph* and to apply Mason's loop rule directly to the pseudo bond graph, as has been demonstrated by Kam in [45]. Mason's loop rule is implemented in the bond graph processing software ARCHER [5].

In the following, the procedure presented is illustrated by means of a small example for which calculations could be carried out manually.

Example: DC Motor with Constant Excitation

Consider the simple, well known example of a DC motor with constant excitation as portrayed in Figure 6.29. Figure 6.30 shows the corresponding bond graph. k_T is the torque constant of the motor, while M_{load} denotes an external disturbing load moment.

The state equations derived from the initial bond graph are

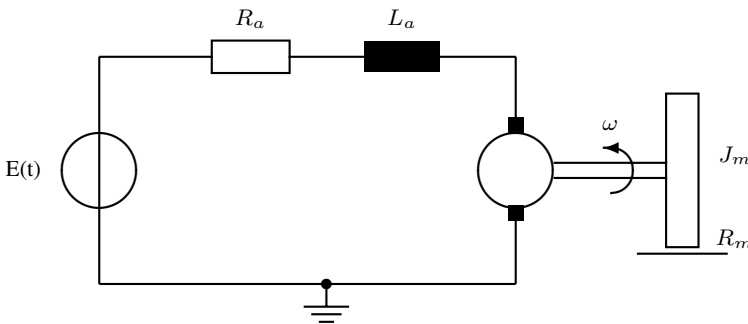


Fig. 6.29 DC motor with constant excitation positioning a mechanical load

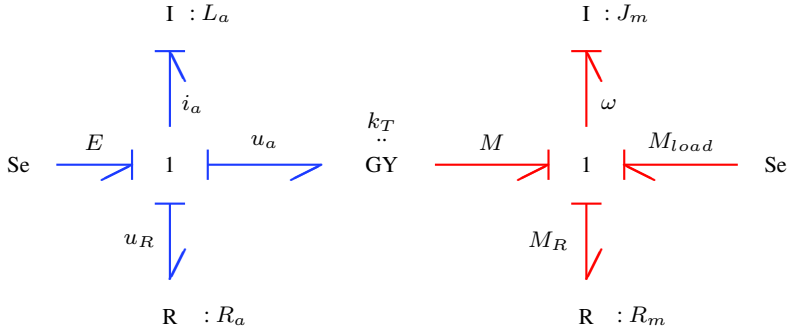


Fig. 6.30 Bond graph of the DC motor example

$$\underbrace{\frac{d}{dt} \begin{bmatrix} p_m \\ p_e \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -\frac{R_m}{J_m} & \frac{k_T}{L_a} \\ \frac{k_T}{J_m} & -\frac{R_a}{L_a} \end{bmatrix}}_{\mathbf{A}_n} \underbrace{\begin{bmatrix} p_m \\ p_e \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{\mathbf{B}_n} \underbrace{\begin{bmatrix} E \\ M_{load} \end{bmatrix}}_{\mathbf{u}}. \quad (6.44)$$

Let us choose the current through the armature coil, i_a , as an output variable y . Then, the matrices \mathbf{C}_n and \mathbf{D}_n read:

$$\underbrace{\begin{bmatrix} i_a \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 0 & \frac{1}{L_a} \end{bmatrix}}_{\mathbf{C}_n} \underbrace{\begin{bmatrix} p_m \\ p_e \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 & 0 \end{bmatrix}}_{\mathbf{D}_n} \underbrace{\begin{bmatrix} E \\ M_{load} \end{bmatrix}}_{\mathbf{u}}. \quad (6.45)$$

For the matrix $\tilde{\mathbf{B}}$ we get

$$\begin{aligned} \tilde{\mathbf{B}} &= \frac{\partial}{\partial \mathbf{p}} (\mathbf{A}_n \mathbf{x}_n + \mathbf{B}_n \mathbf{u}) = \frac{\partial}{\partial \mathbf{p}} (\mathbf{A}_n \mathbf{x}_n) \\ &= \frac{\partial}{\partial (R_a, L_a, k_T, R_m, J_m)} \begin{bmatrix} -\frac{R_m}{J_m} p_m + \frac{k_T}{L_a} p_e \\ \frac{k_T}{J_m} p_m - \frac{R_a}{L_a} p_e \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\frac{k_T}{L_a^2} p_e & \frac{1}{L_a} p_e & -\frac{1}{J_m} p_m & \frac{R_m}{J_m^2} p_m \\ -\frac{1}{L_a} p_e & \frac{R_a}{L_a^2} p_e & -\frac{1}{J_m} p_m & 0 & \frac{k_T}{J_m^2} p_m \end{bmatrix}. \end{aligned} \quad (6.46)$$

According to Equation 6.37b, the matrix $\tilde{\mathbf{D}}$ is

$$\begin{aligned}\tilde{\mathbf{D}} &= \frac{\partial}{\partial \mathbf{p}}(\mathbf{C}_n \mathbf{x}_n + \mathbf{D}_n \mathbf{u}) = \frac{\partial}{\partial \mathbf{p}}(\mathbf{C}_n \mathbf{x}_n) \\ &= \frac{\partial}{\partial (R_a, L_a, k_T, R_m, J_m)} \left[\frac{1}{L_a} p_e \right] \\ &= \begin{bmatrix} 0 & -\frac{1}{L_a^2} p_e & 0 & 0 & 0 \end{bmatrix}.\end{aligned}\quad (6.47)$$

The matrix $(s\mathbf{I} - \mathbf{A}_n)^{-1}$ then reads

$$(s\mathbf{I} - \mathbf{A}_n)^{-1} = \frac{1}{\Delta} \begin{bmatrix} s + \frac{R_a}{L_a} & \frac{k_T}{L_a} \\ -\frac{k_T}{J_m} & s + \frac{R_m}{J_m} \end{bmatrix}, \quad (6.48)$$

where

$$\Delta := s^2 + \left(\frac{R_a}{L_a} + \frac{R_m}{J_m} \right) s + \frac{R_m R_a}{J_m L_a} + \frac{k_T^2}{J_m L_a}. \quad (6.49)$$

With these results, a lengthy expression that is not given here is finally obtained for the sensitivity matrix

$$\mathbf{S}(s) = \mathbf{C}_n (s\mathbf{I} - \mathbf{A}_n)^{-1} (\mathcal{L}\tilde{\mathbf{B}})(s) + (\mathcal{L}\tilde{\mathbf{D}})(s). \quad (6.50)$$

Replacement of the bond graph elements in the bond graph of the DC motor example (Figure 6.29) by their associated incremental models discussed in Section 6.5.1 results in an incremental model of the system under consideration shown in Figure 6.31. The incremental bond graph differs from the initial bond graph by the fact that the incremental models of the inertances, the resistors and the gyrator introduce sinks that reflect parameter variations, while the time dependent, parameter independent source of the voltage supply and the load moment disappear. Besides these differences, the incremental bond graph is an ordinary bond graph that can be entered into a graphical bond graph editor. A combination of programs such as CAMPG[®] and MATLAB[®] can then generate the transfer matrix \mathbf{H} for this bond graph in symbolic form. As the inputs into the incremental bond graph are parameter variations, this matrix is just the sensitivity matrix (Equation 6.50).

Parameter Sensitivities of Transfer Functions

If there is a change only in one element parameter, then only that element needs to be replaced by its associated incremental bond graph and the sensitivity matrix reduces to a scalar. For instance, if there is only a variation in the mechanical friction coefficient ΔR_m , then the incremental bond graph takes the more simpler form displayed in Figure 6.32.

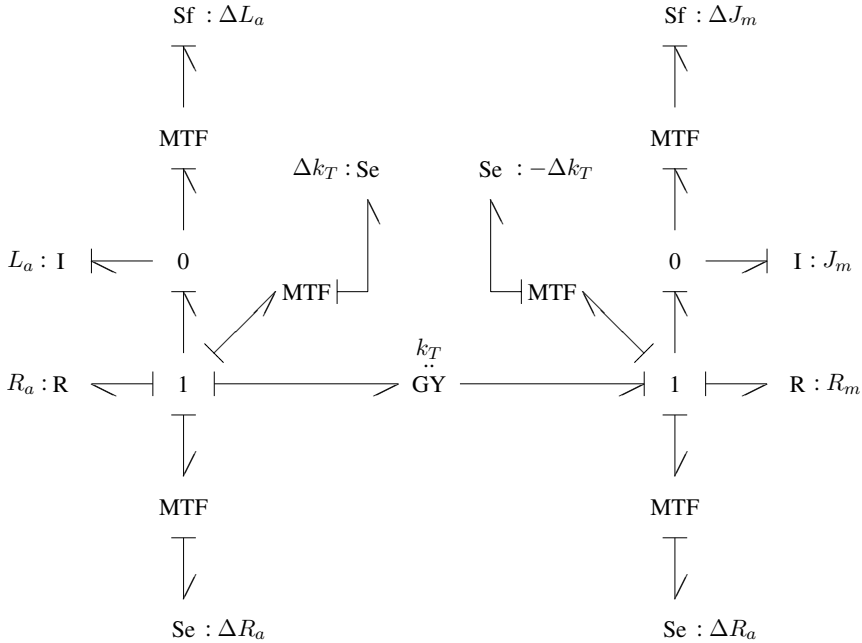


Fig. 6.31 Incremental bond graph of the DC motor example (Figure 6.29)

If the impact of the variation ΔR_m on the armature current is of interest then, first, from the incremental bond graph in Figure 6.32, the following two state equations can be immediately derived.

$$s\mathcal{L}\Delta i = \frac{1}{L}(-R_a\mathcal{L}\Delta i - k_T\mathcal{L}\Delta\omega) \tag{6.51a}$$

$$s\mathcal{L}\Delta\omega = \frac{1}{J_m}[k_T\mathcal{L}\Delta i - R_m\mathcal{L}\Delta\omega - (\mathcal{L}f_{R_m})\Delta R_m] \tag{6.51b}$$

Elimination of $\Delta\omega$ gives

$$\frac{\partial\mathcal{L}i}{\partial R_m} = \frac{[k_T/(L_a J_m)]\mathcal{L}f_{R_m}}{(s + R_a/L_a)(s + R_m/J_m) + k_T^2/(L_a J_m)} \tag{6.52}$$

The denominator in the transfer function of Equation 6.52 is the determinant in Equation 6.49. This is not surprising. Since the incremental bond graph has the same structure as the initial bond graph, both graphs share the same system matrix. The transfer function above can be obtained from the incremental bond graph (Figure 6.32) directly by observing causal paths and by applying Mason's rule (cf. Section 6.2). For an automatic derivation of the transfer function directly from a bond graph, the program ARCHER could be used.

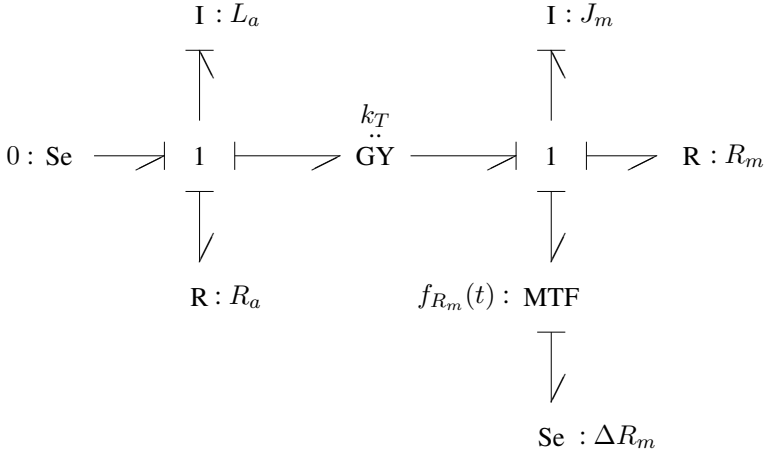


Fig. 6.32 Incremental bond graph of the DC motor example in case of a variation in mechanical friction only

The Laplace transform $\mathcal{L}f_{R_m} = \mathcal{L}\omega$ is obtained from the initial bond graph of the DC motor (Figure 6.30). Let $M_{load} = 0$. Then,

$$\mathcal{L}\omega = \frac{k_T}{k_T^2 + (sL_a + R_a)(sJ_m + R)} \mathcal{L}E. \tag{6.53}$$

Substitution into Equation 6.52 yields

$$\frac{\partial \mathcal{L}i}{\partial R_m} = \frac{k_T^2}{(L_a J_m \det(s\mathbf{I} - \mathbf{A}))^2} \mathcal{L}E. \tag{6.54}$$

Let $F_{11} := \mathcal{L}i/\mathcal{L}E$. Then, Equation 6.54 results in the sensitivity of the transfer function F_{11} with regard to the parameter R_m .

$$\frac{\partial F_{11}}{\partial R_m} = \frac{k_T^2}{(L_a J_m \det(s\mathbf{I} - \mathbf{A}))^2} \tag{6.55}$$

The result can be easily verified by derivation of F_{11} from the initial bond graph and by subsequent differentiation with respect to R_m .

Parameter sensitivities of the other transfer functions, e.g. $F_{21} := \mathcal{L}\omega/\mathcal{L}E$, can be determined in the same manner. In practice, the transfer functions such as the two given by Equation 6.52 and Equation 6.53 are available from the sensitivity matrices that can be set up automatically in symbolic form from the initial bond graph and from the incremental bond graph.

The incremental bond graph approach to the determination of parameter sensitivities presented so far in this section can be also applied to bond graphs of hybrid

models in which switches are modelled by means of a Boolean controlled modulated transformer and a resistor representing its ON state resistance (cf. Figure 7.6).

6.6 State Equations for Robustness Study

When developing models for dynamic systems, one must be aware of shortcomings due to uncertainties. These may be caused by external hazardous perturbations, or may be due to insufficient erroneous parameter identification, or due to tolerances in the manufacturing process of a system. In this section, we will address parameter uncertainties. The analysis of a system with respect to robustness aims at a robust closed loop control that ensures stability and a required system behaviour within acceptable tolerance boundaries in the presence of hazardous perturbations and parameter variations. For robustness study, two special forms of state equations are used. Especially for large systems, considerable computational effort may be necessary to construct these forms even if a software package for symbolic manipulation is used.

Kam and Dauphin-Tanguy [23, 43, 44] have shown how two special forms of state equations used in robustness study, viz. the *canonical form* of state equations with uncertain parameters and the *standard interconnection form* can be derived systematically from a so-called *uncertainty bond graph*. This is a bond graph with true perturbed power variables effort and flow in which elements have been split into a *nominal* and an *uncertain* part.

Both sensitivity of state variables and output variables with respect to parameter perturbations considered in the previous section as well as robustness in stability and robustness in performance are important issues in the design of a system and its control. In the following, it is shown that the *incremental* bond graph previously introduced for the determination of first order parameter sensitivities can also serve as an appropriate starting point for the systematic derivation of the canonical form of state equations used for robustness analysis *if* higher-order terms in the incremental models of bond graph elements are *not* neglected [10, 11]. The incremental bond graph approach to the derivation of state equations for robustness study is equivalent to the uncertainty bond graph approach of Kam and Dauphin-Tanguy and can be applied by using a bond graph program which can generate, in symbolic form, the entries in the matrices in the state equations of linear time-invariant systems [14]. Only a multiplication of matrices derived from the nominal bond graph and from the incremental bond graph needs to be carried out by a program for mathematical computation, e.g., MATLAB[®], in order to obtain the increments of matrices in the canonical form. In the following, the presentation follows the one in [11].

In the next subsection, first, incremental models of standard bond graph elements are briefly revisited. This time, the aim is not the derivation of first order parameter sensitivities, but the derivation of a particular form of state equations used for robustness study. Therefore, higher-order terms in relations between increments of power

variables are no longer neglected as they were in the previous section. In the following, we will see how the matrices in the *canonical form* of state equations can be set up systematically by means of the initial bond graph and its associated incremental bond graph. The approach, applicable to general linear time-invariant systems, is illustrated by means of an intentionally small example that can be checked by hand calculation to keep the presentation easy to survey. It should be kept in mind, however, that the major benefit of the method is the automatic derivation of state equations for robustness study in symbolic form for large systems.

6.6.1 Incremental Models of Linear Bond Graph Elements Revisited

Once again, consider the case of a linear 1-port resistor with nominal resistance R_n . Then, a time-independent parameter variation ΔR causes perturbed power port variables $(e_{R_n} + \Delta e_R)(t)$ and $(f_{R_n} + \Delta f_R)(t)$ at each time t and the linear constitutive equation

$$e_{R_n}(t) = R_n \times f_{R_n}(t) \quad (6.56)$$

becomes

$$(e_{R_n} + \Delta e_R)(t) = (R_n + \Delta R) \times (f_{R_n} + \Delta f_R)(t) \quad (6.57)$$

(cf. Section 6.5).

If the higher-order term $(\Delta R)(\Delta f_R)$ is *not* neglected, then the equation for the increment Δe_R reads

$$(\Delta e_R)(t) = R_n \times (\Delta f_R)(t) - (-f_R(t) \times \Delta R). \quad (6.58)$$

The incremental model representing Equation 6.58 is almost identical to the one depicted in Section 6.5, Figure 6.24. The difference is that the effort source is not modulated by the nominal flow $f_{R_n}(t)$ as indicated in Figure 6.24, but by the *uncertain* flow $f_R(t) = f_{R_n}(t) + (\Delta f_R)(t)$. This means that an internal modulation is introduced into the incremental bond graph model of the resistor as has been pointed out in Figure 6.33.

With $\delta_{R_n} := \Delta R/R_n$, Equation 6.58 can be rewritten in the form

$$\begin{aligned} (\Delta e_R)(t) &= R_n \times (\Delta f_R)(t) + \frac{\Delta R}{R} \times R \times f_R(t) \\ &= R_n \times (\Delta f_R)(t) - \left(-\frac{\delta_{R_n}}{1 + \delta_{R_n}} \right) \times e_R(t). \end{aligned} \quad (6.59)$$

Alternatively, an incremental model *without* internal modulation can be obtained by simply regrouping the terms in Equation 6.58 in a different manner

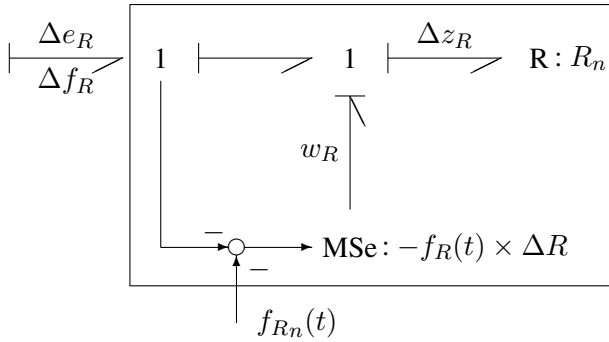


Fig. 6.33 Incremental bond graph model of a linear 1-port resistor with nominal parameter R_n and internal modulation

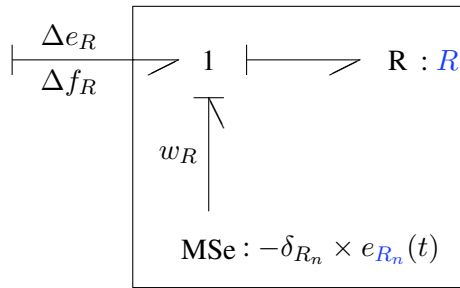


Fig. 6.34 Alternative incremental bond graph of a linear 1-port resistor with uncertain parameter R and no internal modulation

$$\begin{aligned}
 (\Delta e_R)(t) &= R_n \times (\Delta f_R)(t) + (\Delta R)(\Delta f_R)(t) \\
 &\quad - (\Delta R)(\Delta f_R)(t) + (\Delta R)f_R(t) \\
 &= R \times (\Delta f_R)(t) - (-\delta_{R_n} \times e_{R_n}(t)) , \tag{6.60}
 \end{aligned}$$

where $\delta_{R_n} := \Delta R / R_n$. Equation 6.60 can be represented by the incremental model shown in Figure 6.34. Notice that the nominal parameter value R_n is replaced by the *uncertain* value $R = R_n + \Delta R$.

Similar results can be obtained for 1-port energy stores. For instance, for a linear C element with nominal capacitance C_n , the relation for the increment Δe_C is

$$\Delta e_C(t) = \frac{1}{C_n} \times (\Delta q)(t) - \underbrace{\frac{\Delta C}{C_n}}_{=: \delta_{C_n}} \times e_C(t) \tag{6.61}$$

$$= \frac{1}{C} \times (\Delta q)(t) - \underbrace{\frac{\Delta C}{C}}_{=: \delta_C} \times e_{C_n}(t) . \tag{6.62}$$

Equation 6.61 can be represented by the incremental model of Figure 6.35, while Figure 6.36 shows the incremental model corresponding to Equation 6.62.

Like for a resistor and a C energy store, incremental bond graph models can be easily developed for all other bond graph elements *without* neglecting higher-order terms. For a transformer, for instance, the result is the same bond graph given in Figure 6.27. Only the modulus, m_n , is to be replaced by the *perturbed* parameter

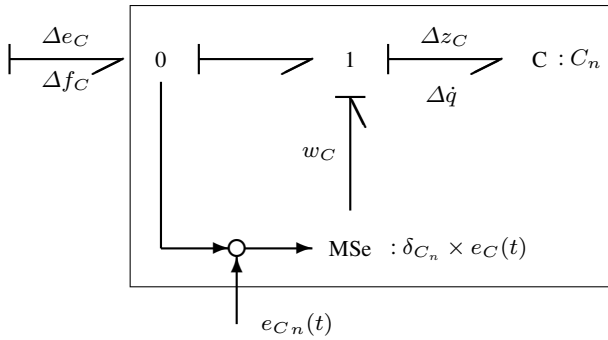


Fig. 6.35 Incremental bond graph of a linear 1-port capacitor with nominal capacitance C_n and internal modulation

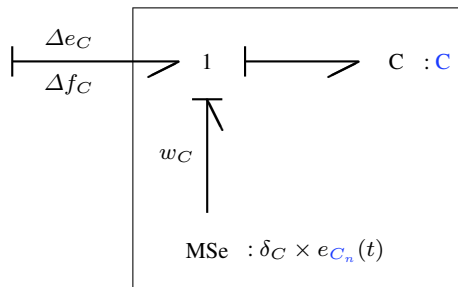


Fig. 6.36 Alternative incremental bond graph model of a linear 1-port energy store with uncertain parameter C and no internal modulation

$m = m_n + \Delta m$. The incremental models of 1- and 0-junctions are 1- and 0-junctions because their constitutive equations are linear and also hold for the incremental power variables. That is, the incremental bond graph retains the structure of the initial bond graph.

6.6.2 Derivation of the Canonical Form of State Equations from an Incremental Bond Graph

In this subsection, we consider linear time-invariant state space models of the special form

$$\dot{\mathbf{x}} = [\mathbf{A}_n + \Delta\mathbf{A}] \mathbf{x} + [\mathbf{B}_n + \Delta\mathbf{B}] \mathbf{u} \quad (6.63a)$$

$$\mathbf{y} = [\mathbf{C}_n + \Delta\mathbf{C}] \mathbf{x} + [\mathbf{D}_n + \Delta\mathbf{D}] \mathbf{u} \quad (6.63b)$$

referred to as the *canonical form* in the literature. The matrices \mathbf{A}_n , \mathbf{B}_n , \mathbf{C}_n , \mathbf{D}_n are of appropriate dimensions. Their entries depend on the components of the vector of nominal parameters $\boldsymbol{\Theta}_n$. The entries in the incremental matrices $\Delta\mathbf{A}$, $\Delta\mathbf{B}$, $\Delta\mathbf{C}$, $\Delta\mathbf{D}$, in general, are nonlinear functions of the parameter variations $\Delta\boldsymbol{\Theta}$ and the nominal parameters. $\Delta\mathbf{A} := \mathbf{A}(\boldsymbol{\Theta}) - \mathbf{A}(\boldsymbol{\Theta}_n)$, $\boldsymbol{\Theta} := \boldsymbol{\Theta}_n + \Delta\boldsymbol{\Theta}$. The matrices $\Delta\mathbf{B}$, $\Delta\mathbf{C}$, $\Delta\mathbf{D}$ are defined likewise. In principle, the required canonical form could be obtained by replacing nominal parameters Θ_{i_n} in the state equations of the nominal system by the perturbed value $\Theta_i := \Theta_{i_n} + \Delta\Theta_i$, ($i = 1, \dots, n_p$), where n_p is the number of parameters. However, the separation of each matrix into a sum of two term as indicated above may be difficult and may require considerable computational effort especially for large systems since the entries in the matrices are generally *nonlinear* functions of the parameters. The incremental bond graph approach as well as the uncertainty bond graph approach have the advantage that both of them can produce the required matrices in symbolic form. This means that the impact of each parameter perturbation on the robustness of a system can be analysed.

For the derivation of the canonical form of state equations from the associated incremental graph, both forms of incremental models of bond graph elements can be used [11]. In the following, incremental models with *perturbed* parameters and *no* internal modulation are used (Figures 6.34 and 6.36). In the subsequent formulae, this is indicated by a subscript 2. The state variables in the *incremental* bond graph are denoted by $\Delta\mathbf{x}$, while the output variables of the sources in the incremental models of standard bond graph elements are combined into a vector \mathbf{w}_2 . The latter is related to the vector \mathbf{z}_{2_n} of outputs of all elements with nominal parameters via a diagonal matrix Δ_2 (cf. Equations 6.60 and 6.62)

$$\mathbf{w}_2 = \Delta_2 \mathbf{z}_{2_n} . \quad (6.64)$$

The state equations for $\Delta\mathbf{x}$ can be derived in symbolic form from the incremental bond graph in the same manner in which state equations are derived from the nominal bond graph

$$\Delta\dot{\mathbf{x}} = \mathbf{A}(\Theta)\Delta\mathbf{x} + \mathbf{B}^*(\Theta)\mathbf{w}_2 \quad (6.65)$$

with matrices \mathbf{A} and \mathbf{B}^* depending on perturbed parameters. The initial bond graph with *nominal* parameters yields

$$\mathbf{z}_{2_n} = \mathbf{C}_{2_n}\mathbf{x}_n + \mathbf{D}_{2_n}\mathbf{u} . \quad (6.66)$$

The combination of the Equations 6.65, 6.64, and 6.66 gives for $\Delta\dot{\mathbf{x}}$

$$\begin{aligned} \Delta\dot{\mathbf{x}} &= \mathbf{A}\Delta\mathbf{x} + \mathbf{B}^*\Delta_2(\mathbf{C}_{2_n}\mathbf{x}_n + \mathbf{D}_{2_n}\mathbf{u}) \\ &= \mathbf{A}\Delta\mathbf{x} + (\mathbf{B}^*\Delta_2 \cdot \mathbf{C}_{2_n})\mathbf{x}_n + (\mathbf{B}^*\Delta\mathbf{D}_{2_n})\mathbf{u} . \end{aligned} \quad (6.67)$$

On the other hand, we can write $\Delta\dot{\mathbf{x}}$ in the form:

$$\begin{aligned} \Delta\dot{\mathbf{x}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}_n \\ &= [\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}] - [\mathbf{A}_n\mathbf{x}_n + \mathbf{B}_n\mathbf{u}] \\ &= \mathbf{A}(\mathbf{x}_n + \Delta\mathbf{x}) - \mathbf{A}_n\mathbf{x}_n + [\mathbf{B} - \mathbf{B}_n]\mathbf{u} \\ &= \mathbf{A}\Delta\mathbf{x} + (\Delta\mathbf{A})\mathbf{x}_n + (\Delta\mathbf{B})\mathbf{u} . \end{aligned} \quad (6.68)$$

Finally, comparison of the Equations 6.67 and 6.68 gives

$$\Delta\mathbf{A} = \mathbf{B}^*\Delta_2\mathbf{C}_{2_n} \quad (6.69a)$$

$$\Delta\mathbf{B} = \mathbf{B}^*\Delta_2\mathbf{D}_{2_n} . \quad (6.69b)$$

Similar results can be obtained for the matrices $\Delta\mathbf{C}$ and $\Delta\mathbf{D}$. Equation 6.67 will be illustrated by means of a small example. From a practical point of view, the matrices \mathbf{A}_n , \mathbf{B}_n , \mathbf{C}_{2_n} , \mathbf{D}_{2_n} (initial bond graph), and $\mathbf{B}^*(\Theta)$ (incremental bond graph) can be obtained by means of any software program that can derive state space equations in symbolic form from a bond graph, e.g., the bond graph preprocessor CAMP-G[®][20], or the computer aided modelling and analysis program ARCHER [5]. The multiplications of matrices needed can be performed by a program for mathematical computations, e.g., MATLAB[®] [6].

Example: Electrical Circuit with Two Energy Stores

The derivation of the canonical form of state equations shall be illustrated by means of the simple circuit with two energy stores shown in Figure 6.37. The initial bond graph corresponding to the circuit schematic is shown in Figure 6.38.

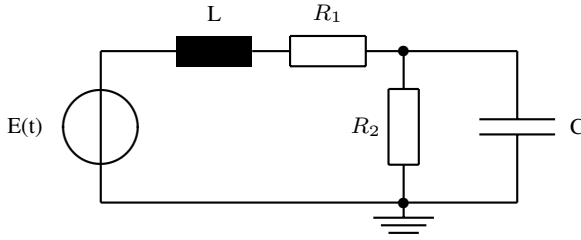


Fig. 6.37 Circuit with two energy stores

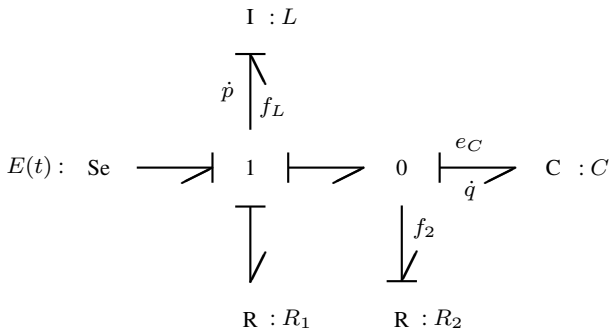


Fig. 6.38 Bond graph of the circuit

The state equations are

$$\underbrace{\begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -R_1/L & -1/C \\ 1/L & -1/(R_2C) \end{bmatrix}}_{\mathbf{A}(\boldsymbol{\Theta})} \underbrace{\begin{bmatrix} p \\ q \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{[E(t)]}_{\mathbf{u}} \quad (6.70)$$

The bond graph elements in the initial bond graph can be replaced equally by incremental models with or without internal modulation. In the following, incremental models without internal modulation, but with perturbed parameters are used. Figure 6.39 shows the resulting incremental bond graph (with uncertain parameters).

As for any other linear bond graph, the following equations can be derived in symbolic form.

$$\underbrace{\begin{bmatrix} \Delta \dot{p} \\ \Delta \dot{q} \end{bmatrix}}_{\Delta \dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -R_1/L & -1/C \\ 1/L & -1/(R_2C) \end{bmatrix}}_{\mathbf{A}(\boldsymbol{\Theta})} \underbrace{\begin{bmatrix} \Delta p \\ \Delta q \end{bmatrix}}_{\Delta \mathbf{x}} +$$

Thus, $\mathbf{D}_{2_n} = \mathbf{0}$ in this example. With these results, the product $\mathbf{B}^* \Delta_2 \mathbf{C}_{2_n}$ gives the incremental matrix $\Delta \mathbf{A}$ to be determined (Equation 6.69a).

$$\Delta \mathbf{A} = \begin{bmatrix} R_1 \frac{\Delta L}{LL_n} - \frac{\Delta R_1}{L_n} & \frac{\Delta C}{CC_n} \\ -\frac{\Delta L}{LL_n} & \frac{\Delta R_2}{R_2 R_{2_n} C_n} + \frac{\Delta C}{R_2 CC_n} \end{bmatrix} \quad (6.74)$$

Consider the special case in which only the two resistors are assumed to have uncertain parameters, i. e., $\Delta L = 0$, $\Delta C = 0$. Then, the matrix $\Delta \mathbf{A}$ reduces to

$$\Delta \mathbf{A} = \begin{bmatrix} -\frac{\Delta R_1}{L_n} & 0 \\ 0 & \frac{\Delta R_2 / R_{2_n}}{R_{2_n} C_n (1 + \Delta R_2 / R_{2_n})} \end{bmatrix} \quad (6.75)$$

in accordance with a result given earlier by Dauphin and Kam in [23]. Due to $\mathbf{D}_{2_n} = \mathbf{0}$, the increment $\Delta \mathbf{B}$ vanishes (cf. Equation 6.69b). This is true because the entries in \mathbf{B} are independent of any parameter values. If $\mathbf{y} = [e_c]$, then the increment $\Delta \mathbf{y}$ is obtained from the incremental bond graph.

$$\begin{aligned} \Delta e_c &= \frac{1}{C} \Delta q - w_c \\ &= \underbrace{[0 \ 1/C]}_{\mathbf{C}(\boldsymbol{\theta})} \Delta \mathbf{x} + \underbrace{[0 \ 0 \ 0 \ -1]}_{\mathbf{D}^*} \mathbf{w}_2 \end{aligned} \quad (6.76)$$

Substitution of the vector \mathbf{w}_2 by means of Equations 6.72 and 6.73 finally gives

$$\begin{aligned} \Delta \mathbf{C} &= \mathbf{D}^* \Delta_2 \mathbf{C}_{2_n} \\ &= [0 \ 0 \ 0 \ -1] \begin{bmatrix} \delta_L & & & \\ & -\delta_{R_{1_n}} & & \\ & & -\delta_{R_{2_n}} \times R_{2_n} & \\ & & & \delta_C \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} 1/L_n & 0 \\ R_{1_n}/L_n & 0 \\ 0 & 1/(R_{2_n} C_n) \\ 0 & 1/C_n \end{bmatrix} = [0 \ -\delta_C / C_n] . \end{aligned} \quad (6.77)$$

6.6.3 The Standard Interconnection Form

The standard interconnection form, or Linear Fractional Transformation (LFT) form, mentioned in the beginning of Section 6.6 is another form of state equations for robustness study. The underlying idea is to account for parameter uncertainties

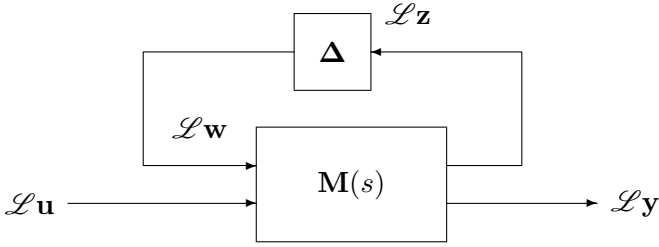


Fig. 6.40 Standard interconnection model

in linear time-invariant systems by means of an internal feedback as shown in Figure 6.40 [44]. The model in Figure 6.40 is also known as an *internal feedback loop model*. The matrix $\mathbf{M}(s)$, $s \in \mathbb{C}$, denotes the interconnection matrix depending on nominal parameter values only, while Δ is a diagonal matrix (δ_{ij}) with $|\delta_{ii}| < 1$ (\mathcal{L} denotes the Laplace operator). According to Figure 6.40, matrices of appropriate dimensions can be set up such that a linear system is given by the equations

$$\dot{\mathbf{x}} = \mathbf{A}_n \mathbf{x} + \underbrace{[\mathbf{B}_{1_n} \ \mathbf{B}_{2_n}]}_{\hat{\mathbf{B}}_n} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \quad (6.78a)$$

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C}_{1_n} \\ \mathbf{C}_{2_n} \end{bmatrix}}_{\hat{\mathbf{C}}_n} \mathbf{x} + \underbrace{\begin{bmatrix} \mathbf{D}_{11_n} & \mathbf{D}_{12_n} \\ \mathbf{D}_{21_n} & \mathbf{D}_{22_n} \end{bmatrix}}_{\hat{\mathbf{D}}_n} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \quad (6.78b)$$

$$\mathbf{w} = \Delta \cdot \mathbf{z} . \quad (6.78c)$$

With these matrices $\mathbf{M}(s)$ in Figure 6.40 reads

$$\mathbf{M}(s) = \hat{\mathbf{C}}_n (s\mathbf{I} - \mathbf{A}_n)^{-1} \hat{\mathbf{B}}_n + \hat{\mathbf{D}}_n . \quad (6.79)$$

The standard interconnection form is used by the μ -Analysis and Synthesis method for which a MATLAB[®] toolbox has been developed by Balas and his co-workers [31].

Now, first it is assumed that the incremental models corresponding to the standard bond graph elements incorporate internal modulation (cf. Figures 6.33, 6.35). In subsequent formulae, this case is indicated by a subscript 1. The state variables in the incremental bond graph are denoted by $\Delta \mathbf{x}$ and the output variables of the modulated sources in the incremental models of standard bond graph elements are combined into a vector \mathbf{w}_1 . The state equations for $\Delta \mathbf{x}$ can be derived automatically in symbolic form from the incremental bond graph as from any other linear bond graph.

$$\Delta \dot{\mathbf{x}} = \mathbf{A}_n \Delta \mathbf{x} + \mathbf{B}_n^* \mathbf{w}_1 \quad (6.80)$$

Since the elements in the incremental submodels have nominal parameters (cf. Figures 6.33, 6.35) the matrices in Equation 6.80 depend on nominal parameters as indicated by the subscript n .

If the initial bond graph is used with *uncertain* parameters and if the outputs e_R, e_C etc. of all elements with *uncertain* parameters are grouped into a vector \mathbf{z}_1 , then this vector is related to \mathbf{w}_1 in the incremental bond graph by means of a diagonal matrix $\mathbf{\Delta}_1 = (\delta_{1ii})$ with entries $|\delta_{1ii}| < 1$ (cf. Equations 6.59 and 6.61).

$$\mathbf{w}_1 = \mathbf{\Delta}_1 \mathbf{z}_1 \quad (6.81)$$

With these equations, the matrices in the standard interconnection form are easily obtained. Equations 6.80 and 6.35a result in

$$\begin{aligned} \dot{\mathbf{x}} &= \dot{\mathbf{x}}_n + \Delta \dot{\mathbf{x}} \\ &= [\mathbf{A}_n \mathbf{x}_n + \mathbf{B}_n \mathbf{u}] + [\mathbf{A}_n \Delta \mathbf{x} + \mathbf{B}_n^* \mathbf{w}] \\ &= \mathbf{A}_n \mathbf{x} + \mathbf{B}_n^* \mathbf{w} + \mathbf{B}_n \mathbf{u} . \end{aligned} \quad (6.82)$$

If the outputs of all elements in the *incremental* bond graph are combined into a vector $\Delta \mathbf{z}$ (cf. Figure 6.35), an equation

$$\Delta \mathbf{z} = \mathbf{C}_n \Delta \mathbf{x} + \mathbf{D}_{11_n} \mathbf{w} \quad (6.83)$$

may be automatically derived from the incremental bond graph while the initial bond graph with nominal parameters provides

$$\mathbf{z}_n = \mathbf{C}_n \mathbf{x}_n + \mathbf{D}_{12_n} \mathbf{u} . \quad (6.84)$$

Combining both equations leads to

$$\mathbf{z} = \mathbf{z}_n + \Delta \mathbf{z} = \mathbf{C}_n \mathbf{x} + \mathbf{D}_{11_n} \mathbf{w} + \mathbf{D}_{12_n} \mathbf{u} . \quad (6.85)$$

In the same manner, the matrices in the equation for the vector of output variables \mathbf{y} in the standard interconnection form (Equation 6.78b) are obtained. Finally, vectors \mathbf{w}_1 and \mathbf{z} are related by the diagonal matrix $\mathbf{\Delta}_1$ (Equation 6.81). That is, the matrices in the standard interconnection form can also be derived from the initial bond graph and its associated incremental bond graph.

6.6.4 Outline of the Uncertainty Bond Graph Approach

Section 6.6.2 has shown that the incremental bond graph introduced by Borutzky and Granda for supporting the deduction of frequency domain sensitivities in symbolic form can also be used to derive state equations for robustness study. Regarding the latter objective, Kam and Dauphin-Tanguy presented a different approach

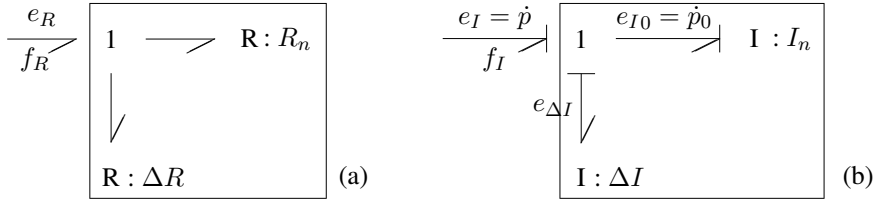


Fig. 6.41 Decomposition of passive 1-port elements with regard to the canonical form of state equations (Dauphin-Tanguy and Kam 1999). **a** Decomposition of a 1-port resistor. **b** Decomposition of a 1-port energy store.

based on a so-called *uncertainty* bond graph. Whereas bonds in the *incremental* bond graph carry increments of power variables, they are perturbed power variables $e := e_n + \Delta e$ and $f := f_n + \Delta f$ in the *uncertainty* bond graph as mentioned at the beginning of Section 6.6. The basic idea of their approach is to split each linear passive 1-port impedance into a nominal part and an uncertain part and to replace the linear 1-port element by two elements of the same type attached to a junction. The type of the additional element depends on whether the canonical form or the standard interconnection form is the objective. If the canonical form of state equations is required, then 1-port resistors and 1-port energy stores are decomposed as shown in Figure 6.41. To ensure that the order of the model is not increased by the companion store of an energy store with nominal parameter, the additional energy stores, representing the parameter variations must have derivative causality (Figure 6.41b). For the example circuit, the uncertainty bond graph is shown in Figure 6.42.

The standard interconnection form of state equations uses a diagonal matrix $\Delta = (\delta_{ii})$ with $|\delta_{ii}| < 1$. For that reason, passive 1-port elements are decomposed as shown in Figure 6.43 [44] for the case of a linear 1-port resistor in resistance causality.

The element De in Figure 6.43 denotes a *detector* that senses the effort of the 1-junction. Summation of efforts at the 1-junction gives

$$e_R = R_n \times f_R - \underbrace{(-\delta_{R_n})}_{w} \times z \quad (6.86)$$

$$= R_n \times f_R + \frac{\Delta R}{R_n} \times R_n \times f_R \quad (6.87)$$

$$= (R_n + \Delta R) \times f_R = R \times f_R . \quad (6.88)$$

Two-port transformers and gyrators may be split into two elements in a similar manner. The result for a TF is shown in Figure 6.44. If all elements are replaced in this way (0-junctions and 1-junctions remain invariant), then the resulting bond graph is called the *uncertainty bond graph*. Apparently, it reduces to the initial nominal bond graph if all parameter variations vanish.

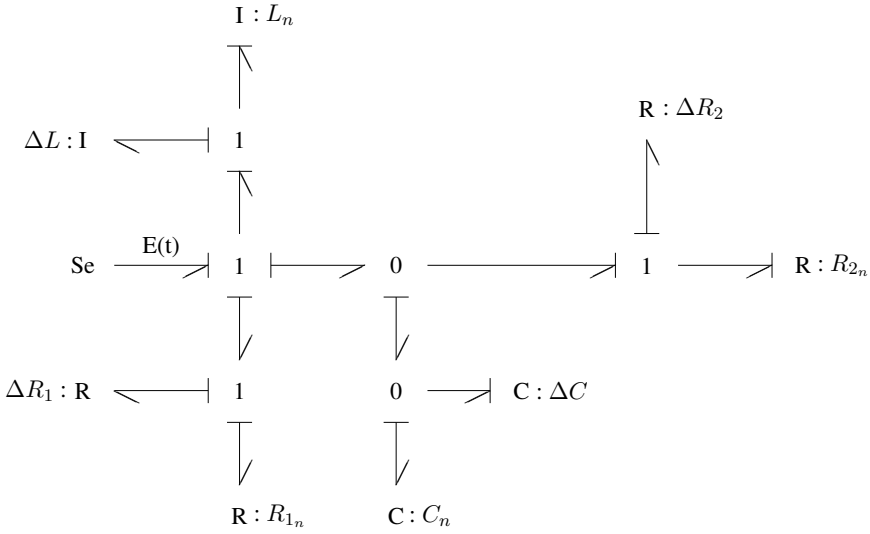


Fig. 6.42 Uncertainty bond graph of the example circuit in Figure 6.37

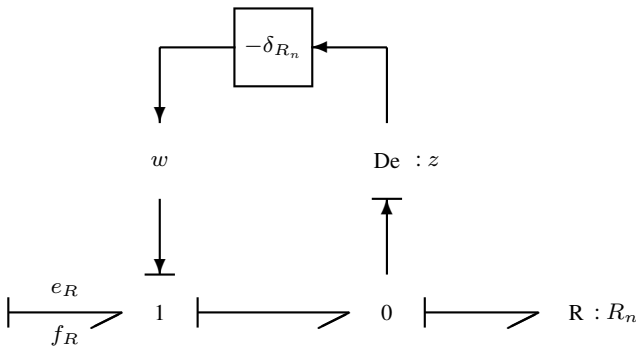


Fig. 6.43 Decomposition of a 1-port resistor with regard to the standard interconnection form (Kam and Dauphin-Tanguy, 2001)

Each uncertainty bond graph can be partitioned as shown in the block diagram of Figure 6.45. The latter is an extension of the general field representation of Karnopp and Rosenberg [46] (p. 272ff), or [62]. This block diagram is obtained by introducing the uncertain elements along with their variables and their constitutive laws.

With the vectors and matrices introduced in Figure 6.45, lengthy general expressions for the nominal matrices and the incremental matrices in the canonical form of state equations can be deduced. The procedure is straightforward and requires extensive symbolic computational effort. In fact, according to Fig-

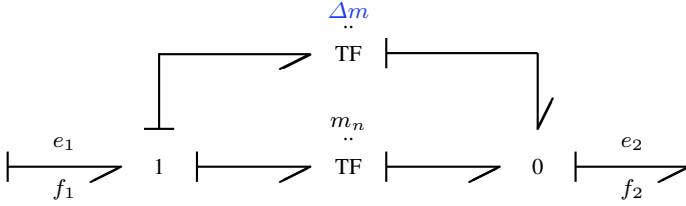


Fig. 6.44 Partitioning of a TF with an uncertain parameter into two elements

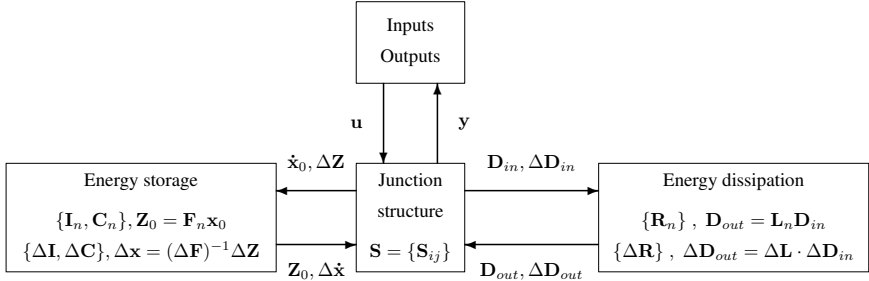


Fig. 6.45 Partitioning of an uncertainty bond graph (cf. Kam 2001)

ure 6.45, all output vectors of the junction structure can be combined into a vector $s_{out} := [\dot{x}_0, \Delta Z, D_{in}, \Delta D_{in}, y]^T$ and all input vectors of the junction structure into a vector $s_{in} := [Z_0, \Delta \dot{x}, D_{out}, \Delta D_{out}, u]^T$. Both vectors are related by the junction structure matrix $S := \{S_{ij}, i, j = 1, \dots, 5\}$

$$s_{out} = S \cdot s_{in} . \tag{6.89}$$

Furthermore, the state vector x of the *uncertainty* bond graph is related to the state vector x_0 of the *nominal* bond graph by means of a diagonal transformation matrix T . That is, $x = T x_0$. Since the junction structure equations as well as the constitutive equations of the elements are linear, the matrices in the canonical form of state equations (Equations 6.63a and 6.63b) can be deduced after basic but extensive symbolic manipulations. The result are lengthy expressions. Details may be found in Kam’s Ph.D. thesis [43]. The expressions for the nominal matrices in the canonical form of the state equations read

$$A_0 = [S_{11} + S_{13}L_nS_{31}]F_n \tag{6.90a}$$

$$B_0 = [S_{15} + S_{13}L_nS_{35}] \tag{6.90b}$$

$$C_0 = [S_{51} + S_{53}L_nS_{31}]F_n \tag{6.90c}$$

$$D_0 = [S_{55} + S_{53}L_nS_{35}] . \tag{6.90d}$$

The entries in these nominal matrices can be directly derived from the nominal bond graph by following causal paths and by determining loop gains according to Mason's loop rule.

For the incremental matrices in the canonical form of the state equations, the following expressions can be obtained

$$\Delta \mathbf{A} = \mathbf{A}_0(\mathbf{T}^{-1} - \mathbf{I}) + \mathbf{\Gamma T}^{-1} \quad (6.91a)$$

$$\Delta \mathbf{B} = [\mathbf{S}_{14} + \mathbf{S}_{13}\mathbf{L}_n\mathbf{S}_{34}]\Delta\mathbf{L}[\mathbf{I} - \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{34}\Delta\mathbf{L}]^{-1} \cdot [\mathbf{S}_{45} + \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{35}] \quad (6.91b)$$

$$\Delta \mathbf{C} = \mathbf{C}_0(\mathbf{T}^{-1} - \mathbf{I}) + \mathbf{\Lambda T}^{-1} \quad (6.91c)$$

$$\Delta \mathbf{D} = [\mathbf{S}_{54} + \mathbf{S}_{53}\mathbf{L}_n\mathbf{S}_{34}]\Delta\mathbf{L}[\mathbf{I} - \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{34}\Delta\mathbf{L}]^{-1} \cdot [\mathbf{S}_{45} + \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{35}] \quad (6.91d)$$

and

$$\mathbf{T} = [\mathbf{I} - \mathbf{S}_{12}\Delta\mathbf{F}\mathbf{S}_{21}\mathbf{F}_n] \quad (6.92a)$$

$$\mathbf{\Gamma} = [\mathbf{S}_{14} + \mathbf{S}_{13}\mathbf{L}_n\mathbf{S}_{34}]\Delta\mathbf{L}[\mathbf{I} - \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{34}\Delta\mathbf{L}]^{-1} \cdot [\mathbf{S}_{41} + \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{31}]\mathbf{F}_n \quad (6.92b)$$

$$\mathbf{\Lambda} = [\mathbf{S}_{54} + \mathbf{S}_{53}\mathbf{L}_n\mathbf{S}_{34}]\Delta\mathbf{L}[\mathbf{I} - \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{34}\Delta\mathbf{L}]^{-1} \cdot [\mathbf{S}_{41} + \mathbf{S}_{43}\mathbf{L}_n\mathbf{S}_{31}]\mathbf{F}_n \quad (6.92c)$$

The evaluation of the Equations 6.91a–6.91d for the incremental matrices to be determined requires a tool for symbolic manipulation, especially for models of large dimension.

The definition of the vectors $\Delta\mathbf{D}_{in}$, $\Delta\mathbf{D}_{out}$ and their affect on the causalities in the junction structure result in sub-matrices \mathbf{S}_{ij} of particular structure. For the reason of power conservation, the junction structure matrix \mathbf{S} is skew-symmetric. Consequently,

$$\mathbf{S}_{12} = -\mathbf{S}_{21} \quad (\text{diagonal matrices})$$

$$\mathbf{S}_{31} = -\mathbf{S}_{13}^T$$

$$\mathbf{S}_{41} = -\mathbf{S}_{14}^T.$$

Since resistors and energy stores are assumed to be linear 1-port elements, the matrices accounting for their constitutive equations are diagonal.

$$\mathbf{F}_n = \begin{bmatrix} 1/I_{in} & 0 \\ 0 & 1/C_{jn} \end{bmatrix} \quad (6.93a)$$

$$\Delta\mathbf{F} = \begin{bmatrix} \Delta I_i & 0 \\ 0 & \Delta C_j \end{bmatrix} \quad (6.93b)$$

$$\mathbf{L}_n = \begin{bmatrix} R_{kn} & 0 \\ 0 & 1/R_{ln} \end{bmatrix} \quad (6.93c)$$

$$\Delta \mathbf{L} = \begin{bmatrix} \Delta R_k & 0 \\ 0 & \Delta R_l \end{bmatrix} \quad (6.93d)$$

From the uncertainty bond graph in Figure 6.42, all matrices needed in the lengthy expressions of the incremental matrices can be derived. The result of the symbolic computation is the same obtained by means of the incremental bond graph approach presented in the previous subsection.

6.7 Bicausal Bond Graphs

For tasks such as

- system inversion,
- state estimation,
- parameter estimation, and
- fault detection

so-called *bicausal* bond graphs have proven useful. The concept of *bicausality* has been introduced by Gawthrop [36]. The idea of bicausal bonds basically is to decouple the orientation of the effort from that of the flow at a bond.

Until now, bicausal bond graphs are not unanimously appreciated in the community of bond graph modellers and have not become a standard part of the bond graph methodology. Aiming at the ordering of equations derived from bond graphs with internal modulation, Cornet and Lorenz in as early as 1989 proposed a representation that is able to independently follow the two power variables associated with bonds [21]. However, in recent years, bicausal bond graphs have received increased attention by researchers, especially in France [50].

Figure 6.46 explains the principle of bicausal bonds. As can be seen from Figure 6.46, standard unicausal bonds are a special case of bicausal bonds. If both causal half strokes coincide at a bicausal bond, then it turns into a standard unicausal bond. In other words, bicausal bonds result if the standard causal stroke is split into two causal half strokes that are attached at both ends of a bond.

In bicausal bond graphs, a bond connected to the port of the 1-port source sensor element, SS, introduced by Gawthrop may be assigned six different causal pattern. Accordingly, the SS element assumes different roles as listed in Table 6.2 [54].

Tables 6.3 and 6.4 depict assignments of bicausalities to the bonds attached to the standard bond graph elements and to two additional elements, AE and AF, introduced by Gawthrop [55].

For a 0 (1)-junction, the effort (flow) is the input at one port, while flows (efforts) are inputs at all ports except one. According to this formulation of the rule, for a bicausal junction this means, that two and only two adjacent bonds can be bicausal

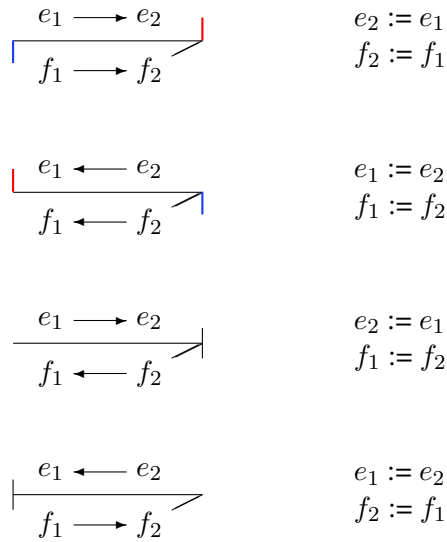


Fig. 6.46 The principle of bicausal bonds

Table 6.2 Causal patterns for the source-sensor element SS (cf. Ngwompo and Gawthrop, 1999)

Causal pattern	Nature of the SS element	
SS	Effort source, flow sensor	Se element
SS	Flow source, effort sensor	Sf element
SS	Zero flow source, effort sensor	De element: Effort detector
SS	Zero effort source, flow sensor	Df element: Flow detector
SS	Flow source, effort source	
SS	Flow sensor, effort sensor	

bonds, i.e., if at one port both effort and flow are inputs then there must be another port, where both power variables are outputs as indicated in Table 6.3. The two additional elements AE and AF are an effort and a flow amplifier respectively. Their

Table 6.3 Causal patterns for the standard bond graph elements and two additional elements

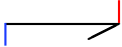
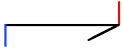
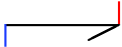
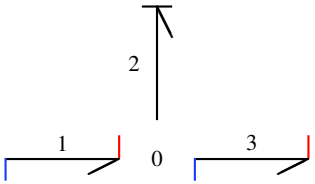
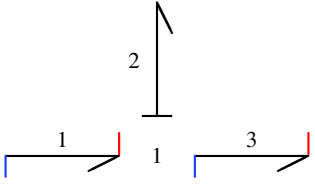
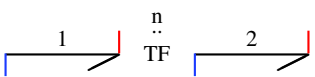
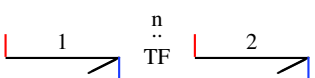
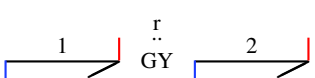
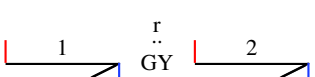
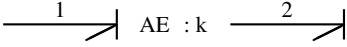
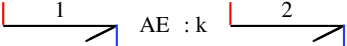
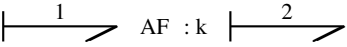
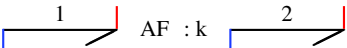
Bicausal pattern	Assignment statements
 $C : C$	$C := q/e$
 $I : I$	$I := p/f$
 $R : R$	$R := e/f$
	$e_2 := e_1$ $e_3 := e_1$ $f_3 := f_1 - f_2$
	$f_2 := f_1$ $f_3 := f_1$ $e_3 := e_1 - e_2$
	$e_2 := e_1/n$ $f_2 := n \times f_1$
	$e_1 := n \times e_2$ $f_1 := f_2/n$
	$e_2 := r \times f_1$ $f_2 := e_1/r$
	$e_1 := r \times f_2$ $f_1 := e_2/r$

Table 6.4 Causal patterns for two additional elements

Bicausal pattern	Assignment statements
	$e_2 := k \times e_1$ $f_1 := 0$
	$e_1 := e_2/k$ $f_1 := 0$
	$f_2 := k \times f_1$ $e_1 := 0$
	$f_2 := k \times f_1$ $e_2 := 0$

function can be represented alternatively in a more traditional way by signal arrows and a GAIN block for amplification of the input signal.

6.7.1 Bicausal Bond Graphs for Parameter Estimation

An illustration of a bicausal bond graph is given in Figure 6.47. With this assignment of bicausality, the SS element imposes both the system input $u = E$ and the system output $y = i$. It is assumed that both variables have been measured and that the values for the capacitance parameter C and the capacitor's initial state q_0 are known. This simple bicausal bond graph now shows that the resistance of the R element can be estimated from the measured input and output. Since the capacitance, C , and the initial state, q_0 , are known and since the current, i , is an input to the capacitor, its output is determined. Thus, the bond connect to the C element is uncausal. Consequently, as indicated by the bicausal bond connect to the R element, both effort and flow are known. That is, its resistance can be deduced. Accordingly, the following assignment statements can be derived from the bicausal bond graph.

$$\dot{u}_c := \frac{1}{C}i \tag{6.94a}$$

$$R := \frac{E - u_C}{i} \tag{6.94b}$$

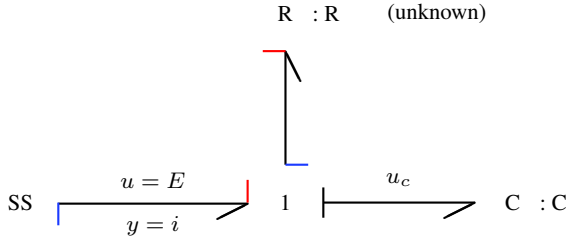


Fig. 6.47 Bicausal bond graph of a RC series interconnection

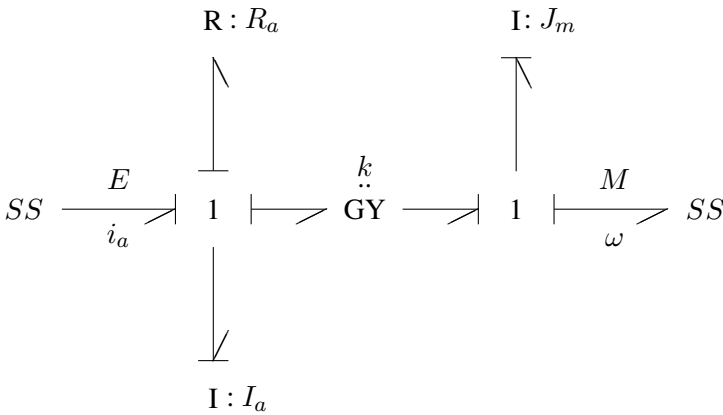


Fig. 6.48 Simple bond graph model of a DC motor

6.7.2 Bicausal Bond Graphs for System Inversion

The subject of system inversion has already been dealt with in Section 6.3. In this section, a bicausal bond graph is used for the derivation of the equations of the inverse system. For illustration, consider the simple bond graph model of a DC motor depicted in Figure 6.48. In this example, the motor is fed by an electrical voltage E and operates against an external mechanical torque M . As a result, the mechanical load rotates with an angular velocity ω and the voltage supply is subject to a feedback current i . That is, there are two inputs, E and M into the system and two outputs, ω and i_a .

Now, let us consider a partial inversion of the system with respect to $u = E$ and $y = \omega$ by questing for the voltage that is necessary to maintain a certain angular velocity under the given load torque. The bicausal bond graph corresponding to this question is given in Figure 6.49. Note that both the voltage E and the current i_a are outputs at the electrical port of the motor model, while both the load torque M

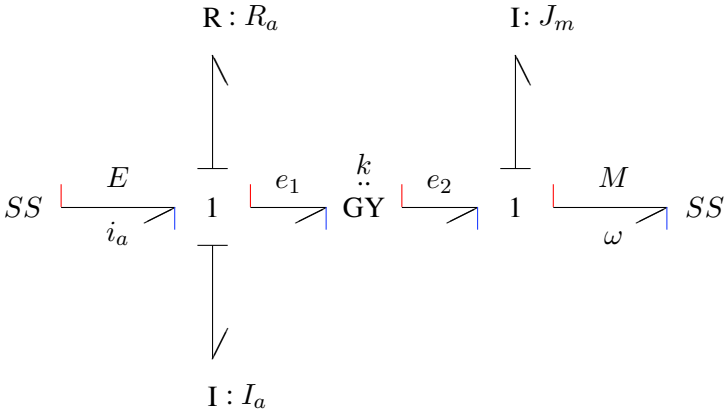


Fig. 6.49 Bicausal bond graph of the inverse system of the DC motor with respect to $u = E$ and $y = \omega$

and the angular velocity ω are inputs to the mechanical port. Furthermore, in contrast to the system model, Figure 6.48, both energy stores have derivative causality. Consequently, the inverse system model has no states.

From the bicausal bond graph of the inverse system model, the following equations can be directly derived.

$$e_2 = J_m \frac{d\omega}{dt} + M \quad (6.95a)$$

$$e_1 = k \times \omega \quad (6.95b)$$

$$i_a = \frac{1}{k} e_2 \quad (6.95c)$$

$$E = I_a \frac{di_a}{dt} + R_a \times i_a + e_1 \quad (6.95d)$$

Elimination of the efforts of the Gyrator and Laplace transform gives the following transfer matrix

$$\begin{bmatrix} \mathcal{L}i \\ \mathcal{L}E \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \mathcal{L}\omega \\ \mathcal{L}M \end{bmatrix}, \quad (6.96)$$

where

$$h_{11} = \frac{J_m}{k} \quad (6.97a)$$

$$h_{12} = \frac{1}{k} \quad (6.97b)$$

$$h_{21} = \frac{I_a}{k} J_m s^2 + \frac{R_a}{k} J_m s + k \quad (6.97c)$$

$$h_{22} = \frac{R_a}{k}. \quad (6.97d)$$

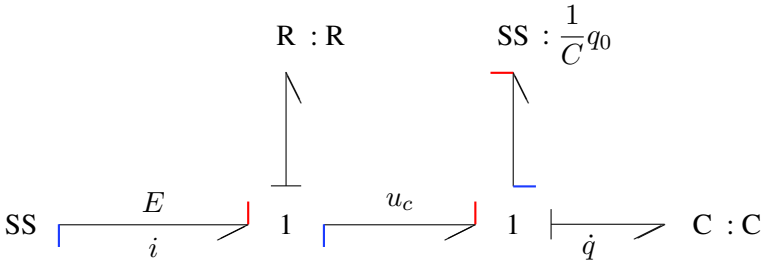


Fig. 6.50 Bicausal bond graph of a RC series interconnection for initial state estimation

This section on bicausal bond graphs concludes by briefly considering their use for state estimation. The basic idea also goes back to Gawthrop [36].

6.7.3 Bicausal Bond Graphs for State Estimation

For illustration, the simple example of a RC series interconnection is used again. According to the constitutive equation of a capacitor, $u_c(t) = q(t)/C + q_0/C$, the initial state, q_0 , can be represented explicitly in a bond graph by an effort source. Given fixed component parameters, the aim is to determine the initial state q_0 from the measured input v and output i . That purpose serves the bicausal bond graph depicted in Figure 6.50.

From the bicausal bond graph of Figure 6.50, the following two equations are immediately derived.

$$u_C = E - R \times i \quad (6.98a)$$

$$\frac{1}{C} q_0 = u_c - \frac{1}{C} q \quad (6.98b)$$

6.8 Bond Graph Model-based Fault Detection and Isolation

The previous sections of this chapter clearly demonstrate that the bond graph methodology can well serve various tasks beyond the development of a model to be used for simulation. This section presents a bond graph approach to model-based fault detection and isolation.

Automatic fault detection and isolation (FDI) is a prerequisite for system reconfiguration, fault tolerant closed loop control and supervision of process engineering systems. Model-based fault detection and isolation means that a system model is available that provides information about the desired accurate dynamic behaviour to

be used for comparison with the measured behaviour of the real process engineering system being subject to faults. Significant deviations can serve as indicators to possible faults in some system components.

Due to its importance, model-based FDI has been the subject of many publications. A survey has been given by Isermann in [41]. Besides research articles, presentations of model-based approaches to FDI not using bond graphs can also be found in textbooks, e.g. [7, 26, 38]. As to bond graph modelling, its use for *quantitative* as well as to *qualitative* model-based FDI has been addressed in various research articles, e.g. [8, 15, 32, 39, 40, 52, 66, 69]. The textbook of Mukherjee, Kamarkar and Samantaray [53] devotes an entire chapter to model-based FDI using bond graphs. Furthermore, Ould Bouamama and Samantaray recently published a textbook on model-based process supervision [65].

This section shows that residual bond graph sinks previously used in bond graph models of the mathematical pendulum (Section 5.2, Figure 5.2 and Figure 5.3) and for tearing algebraic constraints (Section 5.4) can also serve as the base of a bond graph approach to the *numerical* computation of fault indicators in quantitative model-based fault detection. In the modelling, the role of residual sinks is intuitive and (automatic) derivation of equations from the bond graph leads to a DAE system of which the descriptor vector includes fault indicators as components.

6.8.1 Analytical Redundancy Relations

In model-based FDI, so-called Analytical Redundancy Relations (ARRs) [71, 74] play a key role. ARR establish constraints between known variables (input variables and measured output variables) and, in general, also include known model parameters. Under normal mode conditions, the numerical evaluation of ARR should produce values equal to zero. In practice, the result of an evaluation of an ARR, the output variable of an ARR, also called *residuals*, will be within certain small error bounds due to numerical inaccuracies, sensor noise or process parameter uncertainties. If the measured system is subject to faults in some system components, then the values of some residuals may be outside given thresholds and can serve as fault indicators. Structural analysis of analytical constraint relations reveals whether faults can be isolated or not.

In bond graphs, such algebraic constraints arise from junctions. Each junction contributes a continuity equation for flows or efforts. By using the constitutive equations of bond graph elements and by elimination of unknown variables, ARR may be obtained in symbolic form *if* nonlinearities permit necessary eliminations. The form of the set of ARR is not unique and depends on the choice of computational causalities in a bond graph and the procedure that is applied. Moreover, algebraic dependencies indicated by causal paths in the bond graph and nonlinear constitutive relations may *prevent* the elimination of unknown variables.

Given that unknown variables can be eliminated, then structural analysis of each resulting equation leads to what is called a *signature* in terms of known variables

and system component parameters for each residual. Summing up power variables at two junctions may result in the same signature. That is, derivation of ARR from all junctions of a bond graph would produce redundant information. ARRs are derived only from those bond graph junctions with a detector connected to it. The detector models the measuring of a process variable (presented by the common junction variable).

Clearly, for FDI, a model of a system under study should be structurally observable. Moreover, it must be structurally controllable to enable fault tolerant control (FTC). Sensors should be placed such that these conditions are met. Sueur and Dauphin-Tanguy have given sufficient and necessary criterions that enable one to check structural observability and structural controllability directly on a causal bond graph of a linear time-invariant MIMO system (cf. Theorem 6.2, Section 6.4 and Theorem 6.4, Section 6.4.2).

Example: Hydraulic Two Tank System

For illustration, the simple hydraulic two tank system of Figure 6.12 will be considered again. Figure 6.51 shows a bond graph of the system with two effort detectors (De) representing sensors that read the pressures in the tanks.

Evidently, there is a causal path from each C element to a sensor. Moreover, both C storage elements could take preferred differential causality if the causality of the sensors is inverted. That is, the system with the two pressure sensors is *structurally* state observable according to Theorem 6.4. The tank pressures are states that can be observed. If the sensor of pressure p_1 is removed and a sensor of the flow through the valve between both tanks is attached to the left 1-junction, then this system is also structurally observable as it meets the criterion of Theorem 6.4. Other sensor placements also lead to a structurally observable system. For instance, in addition to the two pressure sensors, a flow sensor could be attached to the right-hand side 1-junction measuring the volume flow through the outlet valve (Figure 6.52).

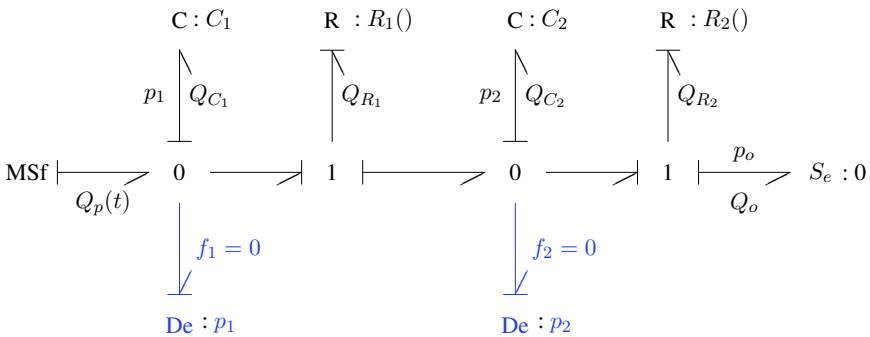


Fig. 6.51 Bond graph of the hydraulic two tank system with two pressure sensors

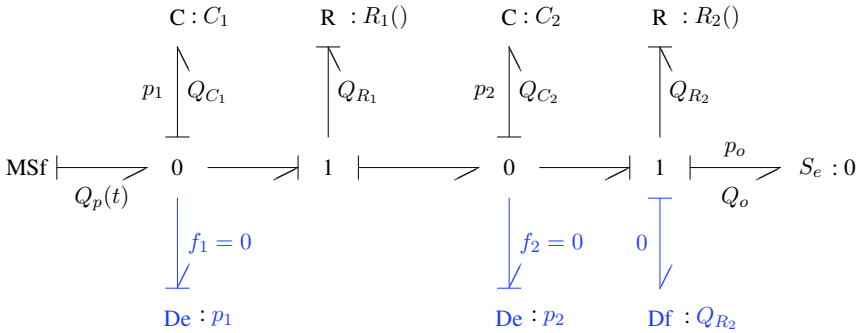


Fig. 6.52 Bond graph of the hydraulic two tank system with tank pressure sensors and a flow sensor in the outlet valve

In the bond graph of Figure 6.51, the left-hand side 0-junction representing the tank pressure p_1 provides the continuity equation for the volume flows:

$$0 = Q_p - Q_{C_1} - Q_{R_1} . \tag{6.99}$$

If in this equation the left-hand side zero is replaced by a variable called residual, res_1 , and if the constitutive equations of the (linear) energy C-store, C_1 and of the nonlinear hydraulic resistor, R_1 , are inserted, then equation Equation 6.99 reads:

$$res_1 = Q_p - C_1 \dot{p}_1 - k_1 \text{sign}(p_1 - p_2) \sqrt{|p_1 - p_2|} . \tag{6.100}$$

The right-hand side of this equation only includes known variables and known component parameters because the pressures in the tanks are measured output variables as indicated by the two effort detectors in Figure 6.51. That is, Equation 6.100 is an ARR. A substitution of the unknown flow variable Q_{C_1} by means of the measured pressure p_1 is also possible in case of an invertible nonlinear characteristic.

Likewise, the sum of all volume flows at the right-hand side 0-junction of tank pressure p_2 leads to an ARR for another residual res_2 .

$$res_2 = k_1 \text{sign}(p_1 - p_2) \sqrt{|p_1 - p_2|} - C_2 \dot{p}_2 - k_2 \sqrt{p_2} \tag{6.101}$$

In the same way, an ARR, res_3 , could be established by summing up all efforts at the 1-junction representing the flow Q_{R_1} through the valve between the two tanks.

$$0 = p_1 - p_2 - \frac{1}{k_1^2} (Q_p - C_1 \dot{p}_1)^2 \text{sign}(Q_p - C_1 \dot{p}_1) = res_3 \tag{6.102}$$

However, inspection of this equation reveals that it does not provide other information with respect to known variables and component parameters than does the ARR

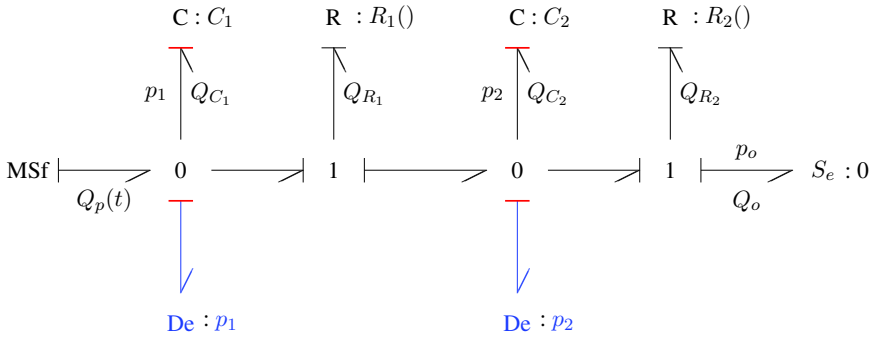


Fig. 6.53 Bond graph of the hydraulic two tank system in preferred derivative causality and sensors with inverted causality

of residual res_1 . That is, both ARR_s have the same signature. Their residuals are not structurally independent.

Note that the same ARR_s, Equations 6.100 and 6.101 for residuals res_1 and res_2 , are obtained from a bond graph of the two tank system in which preferred derivative causality has been assigned to the C stores and causality of the pressure detectors has been inverted (Figure 6.53). The assignment of preferred derivative causality for the determination of residuals is favoured by Samantaray and his co-workers [66].

6.8.2 Structural Fault Signature Matrices

Inspection of ARR_s with respect to the known variables and component parameters they link leads to a so-called *fault signature matrix*. For instance, in the ARR of residual res_1 , Equation 6.100, the left side tank, T_1 , contributes a parameter C_1 and the valve between the two tanks contributes a parameter k_1 . Looking at ARR_s this way, an occurrence matrix can be set up with one row for each known variable or component parameter and one column for each residual. A known variable or component parameter present in an ARR is indicated by ‘1’, its absence by ‘0’. That is, the resulting matrix shows which components contribute to which residuals. The columns in a structural fault signature matrix indicate the signatures of the ARR_s of the residuals. The rows display the fault signatures of components. Such a matrix is usually extended by two additional columns. The first one with the heading ‘Det’ indicates whether a fault can be detected. The second additional column with the heading ‘Iso’ indicates whether a fault can be isolated [15], that is, can be identified unequivocal as the cause of a failure among other possible faults in one or several system components. If this is feasible, then it is expressed by ‘1’ and otherwise by ‘0’. A fault can be detected if there is at least one non-zero entry in that row. This is indicated by ‘1’ in the first additional column and otherwise by ‘0’.

Table 6.5 Structural fault signature matrix of the two tank system with two pressure sensors

Component	Variable/Parameter	res ₁	res ₂	Det	Iso
Pump	Q_p	1	0	1	0
Tank 1	$C_1 := A_{T_1}/(\rho g)$	1	0	1	0
Valve 1	$k_1 := c_d A_{V_1} \sqrt{2/\rho}$	1	1	1	0
Tank 2	$C_2 := A_{T_2}/(\rho g)$	0	1	1	0
Valve 2	$k_2 := c_d A_{V_2} \sqrt{2/\rho}$	0	1	1	0
Pressure sensor 1	p_1	1	1	1	0
Pressure sensor 2	p_2	1	1	1	0

For illustration, Table 6.5 shows the structural fault signature matrix of the hydraulic two tank system with two pressure sensors. Clearly, the continuity equation of each junction in the bond graph of Figure 6.51 leads to an ARR. However, as only two of them differ with respect to their signature, the ARRs obtained from junctions with a sensor attached to it are included in the fault signature matrix in Table 6.5.

6.8.3 Fault Isolation

As can be seen from the fault signature matrix in Table 6.5, all faults can be detected, but none of them can be isolated. If for the system with two pressure sensors the value of residual 2 is outside the given thresholds, then it cannot be decided whether this is due to a leakage from tank 2 or due to a fault in the outlet valve, or due to simultaneous failures in both components. Clearly, a single fault can be located (isolated) if it can be detected and if the pattern of non-zero entries in the corresponding row of the matrix is unique. The mentioned problem can be solved by adding a flow sensor (Df) that measures the volume flow through outlet valve. Then, the constitutive equation of the outlet valve provides the ARR

$$\text{res}_4 = Q_o - k_2 \sqrt{p_2} = 0. \quad (6.103)$$

Adding the signature of this ARR and a row for the flow sensor results in the structural fault signature matrix shown in Table 6.6. In the fault signature matrix of Table 6.6, the fault signatures of the second tank and the outlet valve are unique. Consequently, faults in these two components can be isolated. If it is assumed that sensors are not faulty, then their rows can be eliminated from the fault signature matrix.

For the two tank system with two pressure sensors, the number of structurally independent residuals is two. In the second case with two pressure sensors and a flow sensor, their number is three. In general, for an observable system, their number is equal to the number of sensors present in the system [53].

Table 6.6 Structural fault signature matrix of the two tank system with two pressure sensors and a flow sensor

Component	Variable/Parameter	res ₁	res ₂	res ₄	Det	Iso
Pump	Q_p	1	0	0	1	0
Tank 1	$C_1 := A_{T_1}/(\rho g)$	1	0	0	1	0
Valve 1	$k_1 := c_d A_{V_1} \sqrt{2/\rho}$	1	1	0	1	0
Tank 2	$C_2 := A_{T_2}/(\rho g)$	0	1	0	1	1
Valve 2	$k_2 := c_d A_{V_2} \sqrt{2/\rho}$	0	1	1	1	1
Pressure sensor 1	p_1	1	1	0	1	0
Pressure sensor 2	p_2	1	1	1	1	1
Flow sensor	Q_0	0	0	1	1	1

For the considered simple example of a hydraulic two tank system, fault signature matrices have been obtained by setting up continuity equations for junctions and by eliminating unknown variables. This intuitive approach has only been used for an easy introduction into the topic. Clearly, as mentioned, such a procedure fails if nonlinearities do not one permit to eliminate unknown variables. Moreover, it is clearly inefficient to set up continuity equations for each junction, eliminate unknown variables and then to single out the structurally independent ARRs. For the numerical computation of residuals it is not necessary to set up ARRs in symbolic form and a structural fault signature matrix can be set up *directly* from a causal bond graph by inspection of causal paths [65, 66], regardless of the special form of nonlinear constitutive element equations. It is sufficient to know which of the two conjugate variables at a power port has been assigned the role of an input variable. Ghoshal [39] and Samantaray and Ghoshal [64] have made use of bicausalities properly assigned to a bond graph in order to construct a fault signature matrix directly from the bond graph.

Having introduced and illustrated the notions of ARRs and of residuals as fault indicators, the next section introduces a bond graph approach to the *numerical* computation of fault indicators in quantitative model-based fault detection based on the use of residual sinks. Unknown variables do not need to be eliminated from model equations in order to obtain ARRs in *symbolic* form. Fault indicators being components of the descriptor vector of a DAE system are numerically computed.

6.8.4 Residual Sinks in Bond Graph Model-based Fault Detection

As the deliberate introduction of faults into a real engineering process for test purposes may lead to hazardous situations, to periods of process instability if the equipment allows for introduction of faults at all, it is obvious to replace the real process by a behavioural model which enables one to introduce all kinds of faults without

risk and to analyse various fault scenarios through off-line simulation. That is, a model of the faulty real process and a faultless process model are computed simultaneously and ‘measured’ signals being outputs of the simulated faulty real process are to be compared with the behaviour produced by the faultless process model.

In the bond graph model-based approach to FDI presented by Samantaray and his coworkers [52, 65, 66], a behavioural bond graph model of the real process subject to faults is coupled to a so-called *diagnostic* bond graph. The latter is a bond graph in which energy stores are assigned preferred *derivative* causality and in which sensors attached to junctions are replaced by modulated sources. The power variable into such a modulated source, equated to zero, is measured by means of a virtual residual sensor (The adjective virtual means that these sensors do not represent a real sensor, but only have computational meaning. In bond graphs, a star (*) is added as a superscript to the symbol of a virtual detector in order to distinguish it from the one of a detector representing a real sensor. For instance, Df^* denotes a virtual flow detector). The coupling of the behavioural and the diagnostic bond graph is achieved by feeding signals from the behavioural model, or in an online simulation, measurements from the real process into the modulated sources (Figure 6.54).

Differentiation with respect to time according to the preferred derivative causality in the diagnostic bond graph is numerically performed by using a functional bond graph model of an electronic analogue differentiator device.

In that approach, derivative causality is given preference over integral causality in the diagnostic bond graph because differentiation makes the computation of residuals independent of initial values of states. Furthermore, computation of a residual based on integration must be re-initialised when the fault disappears after repair.

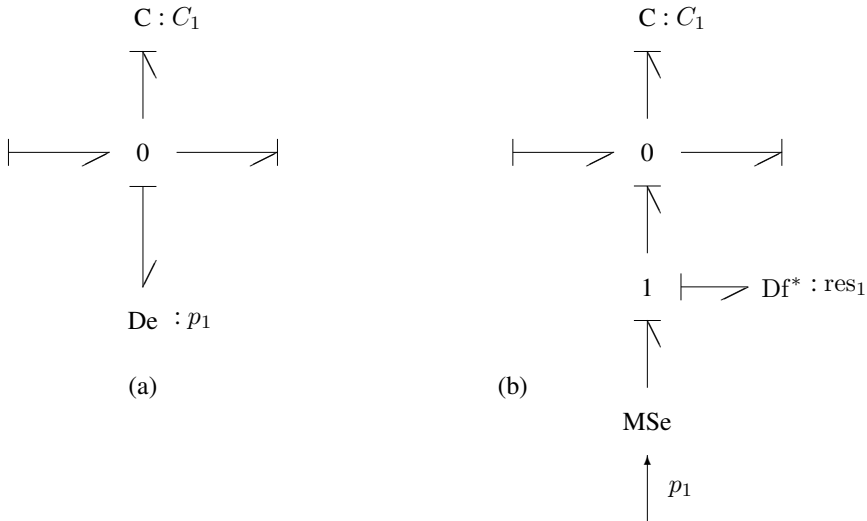


Fig. 6.54 Effort sensor De with inverted causality (a) and its replacement (b)

However, the model of the analog differentiator is only an approximation of the differentiation operation that depends on the chosen values of the model parameters. Moreover, as is well known, numerical differentiation of signals carrying noise degrades the accuracy.

In contrast, in the following, the real engineering process model is coupled to a model of the faultless process by means of residual bond graph sinks and *integral* causality as the *preferred* computational causality is used in both bond graph models. Instead of performing numerical differentiation, the model is formulated as a DAE system. Signals from the behavioural real process model sensed by detectors D_e or D_f , control modulated sources. Their output values are compared with corresponding values from the model of the faultless process.

The real engineering process model accounts for faults by means of modulated sources that can be switched on and off and by modulated resistors with a time-dependent parameter. These elements allow for the analysis of various fault scenarios by deliberately introducing one single fault at a time.

If no faults are introduced into the real process model, then the difference between ‘measured’ signals and their corresponding signals from the faultless process model, theoretically, is equal to zero. In any case, such differences are input into residual sinks. If a difference vanishes, the output of a residual sink is equal to zero. However, if a process variable differs from the corresponding variable in the faultless process model due to a fault introduced into the real process model, the residual sink provides a flow or an effort in order to adapt the faultless process model’s behaviour to the perturbed process behaviour and to force the difference to zero (Figure 6.55).

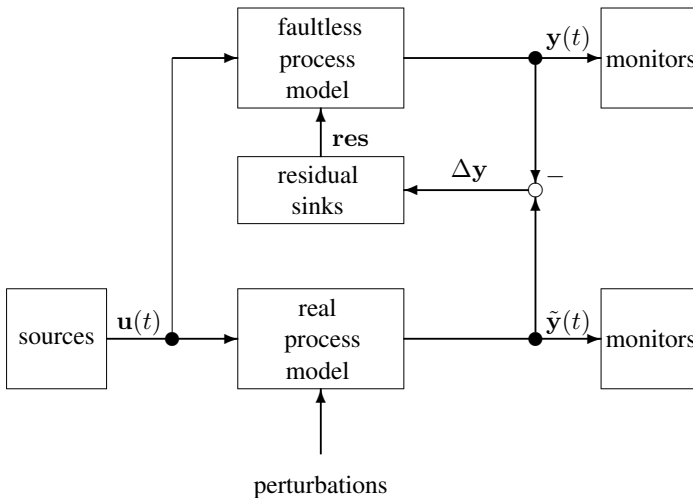


Fig. 6.55 Coupling of the models of the real and the faultless process by residual sinks

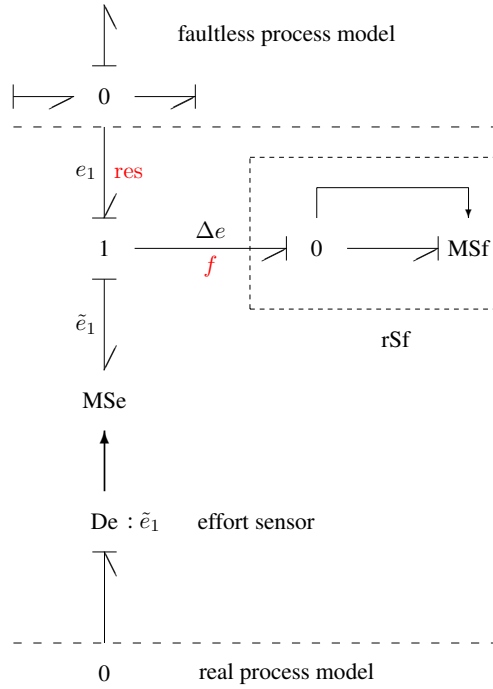


Fig. 6.56 A residual flow sink, rSf, giving rise to a residual, res, at the upper 0-junction

The non-zero output of a residual sink is equal to a residual of an ARR and is a numerical indicator to a fault as shown by Figure 6.56. In Figure 6.56, \tilde{e}_1 is a ‘measured’ effort from the real faulty process model. For brevity, in the following, residual sinks are denoted by rSf or rSe .

As there are no time derivatives of residuals, derivation of equations from the coupled bond graphs of the faultless and the faulty process model result in a DAE system. Energy stores may have a nonlinear characteristic. If all of them can take preferred integral causality, then the system is of the form of a semi-explicit DAE system

$$\dot{\mathbf{x}}(t) = \mathbf{f}_1(t, \mathbf{x}(t), \mathbf{w}(t)) \tag{6.104a}$$

$$\mathbf{0} = \mathbf{f}_2(t, \mathbf{x}(t)) , \tag{6.104b}$$

or

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1(t, \mathbf{x}(t), \mathbf{w}) \\ \mathbf{f}_2(t, \mathbf{x}(t)) \end{bmatrix} , \tag{6.105}$$

where \mathbf{x} denotes the vector of state variables and \mathbf{w} the vector of residuals. That is, the residuals are *numerically* computed as components of the descriptor vector $[\mathbf{x} \ \mathbf{w}]^T$ so that there is no need for establishing continuity equations and for elimi-

nation of unknown variables in order to obtain ARRr in *symbolic* form. The DAE system is of index 2 if

$$\det \begin{pmatrix} \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{w}} \end{pmatrix} \neq 0. \quad (6.106)$$

Remark 6.5. The form of Equations 6.104a–6.104b is also known as the Hessenberg index-2 form. Codes suitable for numerical solution of such DAE systems are, e.g. DASPK 3.1 [1, 70] (based on the BDF method) and Radau5 [4] (implicit Runge-Kutta method of order 5).

6.9 Reduction of Model Complexity

For large systems, modelling commonly follows a bottom-up approach. That is, a model is built by assembling component models taken from libraries as has been shown in Section 8.3 for multibody systems. This modular approach clearly has a number of advantages, including the design of a hierarchical model structure, independent, concurrent design of reusable submodels and a facilitation of model verification. However, resulting overall models may not be necessarily computationally efficient. A reason is that component models kept in libraries have been designed to serve various applications and purposes. As a result, with regard to a given design task, they may contain unnecessary complexity. That is, a mathematical model may be too comprehensive even if redundant equations due to the coupling of component models have been eliminated symbolically prior to numerical computation of the model. Moreover, conceptual efficiency suffers from too many details. Phenomena not relevant with respect to a specific engineering task should be identified and their representation removed from an overall model composed of library submodels. Hence, for handling large systems, reduction of model complexity is essential. This has been a challenge. Accordingly, different approaches and software supporting them are in use to cope with model complexity and to increase computational efficiency.

6.9.1 Model Partitioning

One approach aims at identifying “weak” two-way signal couplings in order to replace them by one-way connections. This is usually known as partitioning. In the bond graph realm, Rideout and Stein [60] and Rideout, Stein and Louca [61] have applied this approach to the junction structure of a bond graph aiming at partitioning a model into driving and driven submodels. Partitioning can be guided by different criteria, e.g., separation of model parts with fast dynamics from those with slow dynamics (multi-time scale systems) [73], or partitioning of a model into parts with

weakly damped dynamics and those with strongly damped dynamics [56]. Partitioning, in general, does *not* follow submodel boundaries. For parallel computation, it is essential that the communication overhead for exchanging information between processors is well below the effort of computing the submodels. In addition, in regard to computational load balance, model partitions should be of similar scale. Clearly, a partitioning into fast and slow parts has the advantage that different appropriate integration algorithms can be simultaneously used.

6.9.2 Model Reduction

Another approach to a reduction of model complexity is model reduction [47, 48]. In order to increase computational efficiency, a lower-order model is developed that approximates the dynamic behaviour while capturing essential dynamics. Louca et. al. introduced and implemented a model reduction algorithm (MORA) at bond graph level based on a so-called activity metric A that measures the energy flow across a power bond [49]. It is defined as the accumulated instantaneous power P over a time interval $[t_1, t_2]$ along a power bond.

$$A := \int_{t_1}^{t_2} |P(\tau)| d\tau = \int_{t_1}^{t_2} |e(\tau) \times f(\tau)| d\tau \quad (6.107)$$

Assuming that an element with low activity at its power ports contributes less to the dynamics of a system, this metric is used for a systematic reduction of a bond graph model. In MORA, this metric is only applied to external power bonds connected to 1-port storage elements and 1-port dissipator, while in their partitioning approach, Rideout, Stein and Louca also apply this metric to the internal bonds of junctions. Obviously, if the activity of a bond connected to a 0-junction is low compared to the activity of all other bonds connected to it, then the flow of that bond can be neglected in the sum of flows at the 0-junction. Conversely, if one of the bonds connected to a 1-junction has a low relative activity, its effort can be neglected in the sum of efforts at the 1-junction. In both cases, the bond with low activity representing a two-way signal connection can be replaced by a one-way signal connection displayed by means of a modulated source [61].

Figure 6.57 shows some cases in which a bond has been replaced by a one-way signal connection. If, in the bond graph of Figure 6.57, part a), the activity of bond 1 is low compared to the one of the other bonds attached to the 0-junction, then flow coming from the 1-junction can be neglected in the sum of flows at the 0-junction. Hence, the connection between the two junctions can be replaced as shown in Figure 6.57, part b). Note that case a) also covers the case of just a bond connecting both junctions. If in case c) the activity of bond 1 is low compared to the one of the other bonds connect to the left 1-junction, then the effort coming from the right-hand side 1-junction can be neglected in the sum of efforts at the left 1-junction. Accordingly, the flow of the left 1-junction is transformed by the

modulated gyrator into an effort that affects the right-hand side 1-junction, but not vice versa. In the bond graph of Figure 6.57, part d), the modulated flow source and the modulated gyrator can be combined into a modulated effort source. Figure 6.58 shows a case in which the activity of an external bond is low compared to the one of the other bonds attached to a junction. In Figure 6.58, the letter Z represents either a storage or an R element. With regard to the search of a lower-order model this means that the Z element can be removed from the model.

6.9.3 Structural Model Simplification

Finally, the initial complexity of a model can be reduced by simplification without compromising accuracy. For instance, a bond graph model of a mechanism that moves in a plane can be built by using instances of the general library model of a rigid body moving freely in space as a building block for multibody system models (cf. Figure 8.16) and models of the joints (cf. Section 8.3) connecting the limbs of the mechanism. However, a considerably simplified model may be achieved without compromising accuracy if it is recognised that motion of the mechanism takes place in a plane. Well known examples of simplification at the equation level are the elimination of Lagrange multipliers or the pole-zero cancellation in transfer functions. Clearly, if model simplification is performed on the equations derived from a bond graph, then features of the simplified model are not visible at the graphical level. In [29], Ersal, Fathy and Stein use the activity metric for identification and elimination of “inactive” junctions with the aim of a systematic model simplification. They consider a junction inactive if all the bonds connected to a junction have an activity below a user-defined inactivity threshold and consider its removal as a model simplification that does not compromise accuracy. Clearly, the idea can be viewed as a generalisation of the fact a 1-junction with a flow equal to zero and a 0-junction with a vanishing effort can be removed from the bond graph. Note that a single bond with vanishing activity attached to a junction does not necessarily make the junction a candidate for removal.

In Section 7.1.4, the concept of switching off degrees of freedom by sinks of invariant causality has been introduced. As an example of application, modelling of the stick-slip effect between rigid bodies has been considered. If two bodies stick together and move with a common velocity, then a constraint force (Lagrange multiplier) can be introduced as depicted in Figure 6.59, which enforces that the relative velocity between both bodies remains zero during the sticking phase. Obviously, in the case of two bodies sticking together, the 0-junction representing the vanishing relative velocity cannot be removed. If an inactive junction is removed from the bond graph, element and submodel ports become detached. They are also removed.

Note that according to the definition (Equation 6.107), the activity on a power bond depends on the proper choice of a time window $[t_1, t_2]$. Moreover, energy flows in the model, and hence activities, depend on excitations, parameters and initial conditions. Once the time window has been chosen, detection of inactive junctions

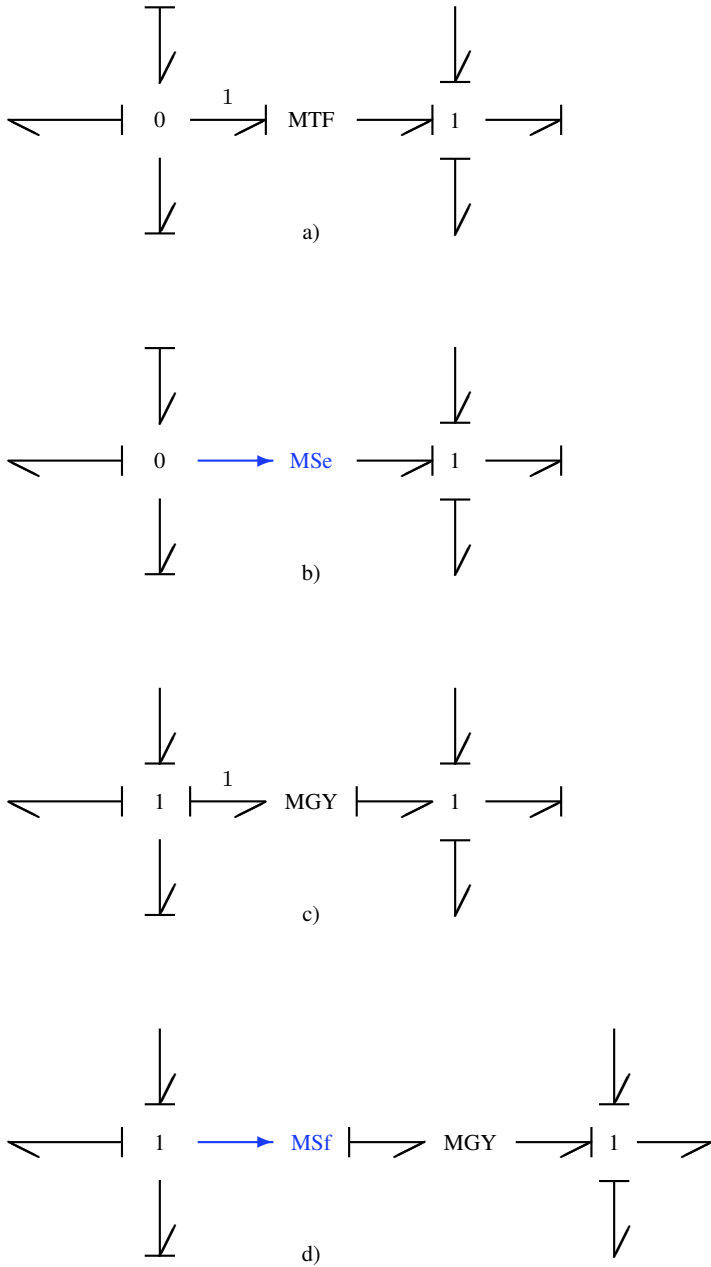


Fig. 6.57 Replacement of an internal bond by means of a one-way signal connection. **a** Bond graph fragment with a MTF. **b** Activation of bond 1. **c** Bond graph fragment with a GY. **d** Activation of bond 1

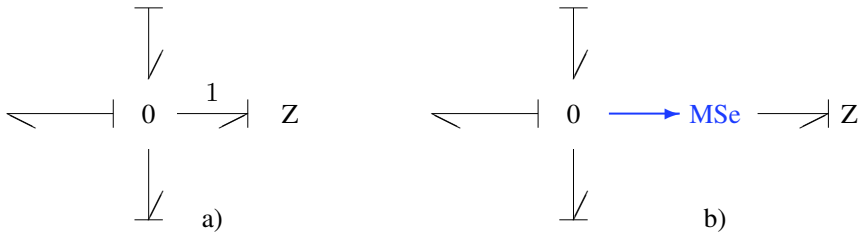


Fig. 6.58 Replacement of an external bond by means of a one-way signal connection. **a** Bond graph fragment with a 1-port impedance Z . **b** Activation of bond 1

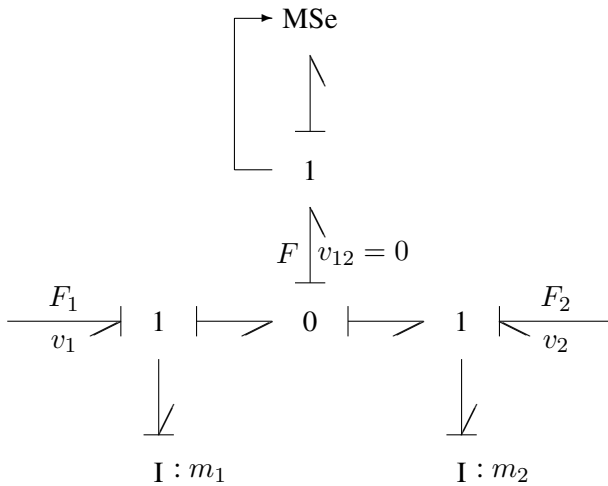


Fig. 6.59 Constraint force between two rigid bodies sticking together

requires computation of the initial model still not simplified and the recording of activities. After having selected an inactivity threshold, inactive junctions can be identified and eliminated (preserving those inactive junctions that are needed for generating modulated signals).

Benefits of a simplification of a bond graph model are an improved conceptual efficiency as well as computational efficiency. The latter is of importance if a model is to be computed repeatedly, e.g., in an optimisation, or if the model is used in real-time simulation. In modelling multibody systems, a proper choice of reference frames is crucial for the complexity of the mathematical model. Commonly, the rotational dynamics of a rigid body are described with respect to a body fixed coordinate system sitting either in the body's centre of gravity or in a hinge point. If coordinate systems are not appropriately oriented and if a modular model has been built by assembling library models for bodies and joints, the resulting overall model may allow only for little simplification, while proper orientation of reference

frames, e.g., in alignment with motions or constraints, may be the key to significant simplifications of an initial complex model. In [30], Ersal, Fathy and Stein present a procedure that checks for each rigid body of multibody system if there is an orientation of the body fixed coordinate frame preferred for simplification. If so, the coordinate transformation in need is found. After this step of coordinate frame reorientation, simplification based on the search for inactive junctions is performed. As reorientations of coordinate frames are locally, the overall model structure still corresponds to the structure of the physical system.

6.10 Conclusion

It has been the objective of this chapter to demonstrate that bond graphs are a generic graphical model description that can serve a variety of tasks. The primary purpose of the bond graph methodology is certainly to guide and to formalise the development of a model from initial ideas and engineering experience to a mathematical model that can be used for experiments on a computer. Not only the dynamic behaviour of a real system can be simulated by solving generated equations in time domain. Moreover, other questions can be addressed as well by analysing a causally completed bond graph.

For an analysis in the frequency domain, first, it is necessary to determine the steady state. The equations describing the steady state can be directly derived from a bond graph after energy stores have been replaced by sinks for which either the input or the output is equal to zero depending on the approach chosen. Manual or automatic inspection of the modified and causally completed bond graph can indicate that there is no steady state or that the equations determining the steady state have an infinite number of solutions. They can be derived from the bond graph in the same way as the equations describing the dynamic behaviour. If there is a unique solution, the bond graph methodology does not give any further help in its determination, viz. the numerical iteration of the nonlinear equations. The problem remains to find a start value for the iteration which must be sufficiently close to the unknown solution if the Newton-Raphson method is used. The problem is well known in the field of circuit analysis.

Once the steady state is known, the dynamic system equations may be linearised and transfer functions as well as pole-zero placements can be determined. These results can be automatically obtained at the level of equations by means of a computer algebra program, e.g. Maple™.

Alternatively, as Brown has shown in [18] in as early as 1972, loop gains needed for the symbolic computation of transfer functions according to Mason's loop rule can be directly determined from a causal bond graph. For small bond graphs, transfer functions can be manually determined in this way. For larger bond graphs, a software implementation of this approach is needed as it is available with the computer aided modelling and analysis program ARCHER [5]. In this chapter, the pro-

cedure has been illustrated by means of two fairly simple bond graphs, one having a tree-like structure (Figure 6.8) and one including a bond loop (Figure 6.9).

In control engineering, the *inverse system* with respect to a pair of a given input and an output variable is of interest. A possible approach is to derive the equations determining the dynamics of the system and to solve them for the inputs needed to have the system generate a prescribed response. As shown in this chapter, this process on equation level can be avoided. Instead, the causal bond graph of a system can be converted into the bond graph of the inverse system from which equations can be derived in the same manner as they are derived from the bond graph of the system. In this transformation, the bond graph retains its structure and only causalities change. If a pair of an input and an output variable is bound to one and the same bond attached to a source, in other words, if both variables are collocated, then the source is simply replaced by its dual and causalities in the graph are readjusted. A source not only imposes an input onto a system, but at the same time, it is also a sensor of the power conjugate variable. Because of this dual role, Gawthrop uses a so-called *source-sensor* element instead of the usual effort and flow sources if a bond graph is used in the solution of a control problem. In bond graphs, the source-sensor element is denoted by the symbol SS [37]. If an input and an output variable are not collocated, then this case can be reduced to the case of pairs of collocated variables by taking into account the input into each source and by adding a source for each output variable that is not connected with a source. This way, the number of system outputs and sources is increased. If a source is added to the graph, it must not have an influence on the model. Thus, additional sources must impose an input equal to zero. After this preparation, causalities at source-sensor elements are reversed as needed. This reversal of causalities can lead to a change from preferred integral to derivative causality for some energy stores (cf. Figure 6.13). Consequently, even if the model of the system is in state space form, generally, this does not hold for the inverse system. Its equations can only be formulated in the more general descriptor form.

Other control related questions may also be addressed by inspection of a causal bond graph. Sueur and Dauphin-Tanguy have formulated conditions for *structural controllability* and *structural observability* that can be easily directly applied to a causal bond graph [24]. Andry and Rosenberg have given a simple test that allows one to decide whether a system is *not* structurally controllable.

In Section 6.5, incremental bond graphs are introduced that can be used to determine unnormalised parameter sensitivities. For linear time-invariant systems, parameter sensitivities can be derived in *symbolic* form. For nonlinear systems, the incremental bond graph is linear time variant so that the equations for the sensitivities must be solved simultaneously together with the state equations of the system. Furthermore, it is shown that incremental bond graphs can also be used to deduce the canonical form of state equations used for robustness analysis. Regarding the latter objective, Kam and Dauphin-Tanguy presented a different approach based on *uncertainty* bond graphs. This approach has been briefly outlined. Although both approaches serve different purposes and were independently developed, they are equivalent.

Finally, bond graph modelling can also serve model-based fault detection. In Section 6.8, an approach has been presented that uses residual sinks for coupling a bond graph model of the faultless system to a bond graph model of the real engineering system subject to faults. An essential feature of this approach is that there is no need to set up Analytical Redundancy Relations in symbolic form. Fault indicators are computed *numerically* as components of a descriptor vector of a DAE system. Due to tolerances in manufacturing processes or faulty parameter identification, parameter values may vary around nominal values. A natural question is how sensitive numerically computed fault indicators are to parameter uncertainties and which of them affect a residual of an ARR most significantly. In [9], a sensitivity pseudo bond graph is coupled to the model of the faultless process and residuals and sensitivities of residuals with respect to a parameter are simultaneously computed. In [28], Djeziri et. al. derive ARRs from an *uncertainty* bond graph of a system in which elements are decomposed with regard to the standard interconnection form. As a result, ARRs can be split into a nominal part and into an uncertain part that is the subject of a sensitivity analysis.

References

- [1] DASPK. URL <http://www.cs.ucsb.edu/~cse/software.html>.
- [2] Matlab control system toolbox. URL <http://www.Mathworks.com/products/control/>.
- [3] Octave. URL <http://www.gnu.org/software/octave/>.
- [4] Radau5. URL <http://www.unige.ch/~hairer/software.html>.
- [5] A. Azmani and G. Dauphin-Tanguy. Archer: a program for computer aided modelling and analysis. In P. C. Breedveld and G. Dauphin-Tanguy, editors, *Bond Graphs for Engineers*, pages 263–278. Elsevier, North-Holland, 1992.
- [6] A. Biran and M. Breiner. *MATLAB for Engineers*. Addison Wesley, 1999. URL <http://www.Mathworks.com>.
- [7] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, Berlin, 2003.
- [8] W. Borutzky. Residual Bond Graph Sinks for Numerical Evaluation of Analytical Redundancy Relations in Model Based Single Fault Detection and Isolation. In *Proc. of the 20th European Conference on Modelling and Simulation*, pages 166–172, Sankt Augustin, Germany, 2006. European Council for Modelling and Simulation.
- [9] W. Borutzky. Bond Graph Model-Based Fault Detection Using Residual Sinks. *Proc. of the Institution of Mechanical Engineers Part I Journal of Systems and Control Engineering*, 223(3):337–352, 2009.
- [10] W. Borutzky and G. Dauphin-Tanguy. Incremental bond graph: starting point for sensitivity analysis as well as for robustness study. In F.E. Cellier and J.J. Granda, editors, *2003 International Conference on Bond Graph Modeling, and Simulation (ICBGM 2003)*, pages 15–20. SCS Publishing, 2003. Simulation Series, volume 35, no 2, ISBN: 1-56555-257-1.
- [11] W. Borutzky and G. Dauphin-Tanguy. Incremental bond graph approach to the derivation of state equations for robustness study. *Simulation Modelling Practice and Theory*, 12(1): 41–60, 2004.
- [12] W. Borutzky and J. J. Granda. Determining sensitivities from an incremental true bond graph. In J.J. Granda and G. Dauphin-Tanguy, editors, *2001 International Conference on Bond Graph Modeling, and Simulation (ICBGM 2001)*, pages 3–8. SCS Publishing, 2001. Simulation Series, Vol. 33, Number 1, ISBN: 1-56555-221-0.

- [13] W. Borutzky and J. J. Granda. Bond Graph Based Frequency Domain Sensitivity Analysis of Multidisciplinary Systems. *Proc. Instn Mech. Engrs, Part I, Journal of Systems and Control Engineering*, 216(1):85–99, 2002.
- [14] W. Borutzky, G. Dauphin-Tanguy, and C. Kam. Relations between two bond graph approaches to sensitivity analysis and study of robustness. *Mathematical & Computer Modelling of Dynamic Systems, Special Issue: Bond Graph Modelling*, 12(2–3):141–157, April–June 2006. doi: 10.1080/13873950500068955.
- [15] B. Ould Bouamama, A.K. Samantaray, M. Staroswiecki, and G. Dauphin-Tanguy. Derivation of Constraint Relations from Bond Graph Models for Fault Detection and Isolation. In J.J. Granda and F.E. Cellier, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM'03*, pages 104–109, Orlando, Florida, USA, January 19–23 2003. SCS Publishing. Simulation Series, volume 35, no 2, ISBN: 1-56555-257-1.
- [16] P.C. Breedveld. A Bond Graph Algorithm to Determine the Equilibrium State of a System. *Journal of the Franklin Institute*, 318(2):71–75, 1984.
- [17] F.T. Brown. *Engineering System Dynamics*. Marcel Dekker, New York, Basel, 2001. ISBN: 0-8247-0616-1.
- [18] F.T. Brown. Direct Application of the Loop Rule to Bond Graphs. *Journal of Dynamic Systems, Measurement and Control*, pages 253–261, September 1992.
- [19] J.M. Cabanellas, J. Féllez, and C. Vera. A formulation of the sensitivity analysis for dynamic systems optimization based on pseudo bond graphs. In F.E. Cellier and J.J. Granda, editors, *ICBGM'95, International Conference on Bond Graph Modeling and Simulation*, pages 135–144. SCS Publishing, 1995. Simulation Series, volume 27, no. 1.
- [20] *CAMP-G – User's Manual*. Cadsim Engineering, P. O. Box 4083, Davis, Ca 95617. URL <http://www.bondgraph.com>.
- [21] A. Cornet and F. Lorenz. Equation Ordering Using Bond Graph Causality Analysis. In P.C. Breedveld et al., editor, *Modelling and Simulation of Systems*, pages 55–58. J.C. Baltzer AG, Scientific Publishing Co., 1989.
- [22] G. Dauphin-Tanguy. *Les bond graphs*. Hermes Science Europe Ltd., Paris, France, 2000. ISBN: 2-7462-0158-5.
- [23] G. Dauphin-Tanguy and C.S. Kam. How to model parameter uncertainties in a bond graph framework. In G. Horton, D. Möller, and U. Rüde, editors, *Simulation in Industry, 11th European Simulation Symposium, ESS'99*, pages 121–125, 1999. Erlangen, Germany.
- [24] G. Dauphin-Tanguy, A. Rahmani, and C. Sueur. Bond graph aided design of controlled systems. *Simulation Practice and Theory*, 7(5-6):493–513, 1999.
- [25] G. Dauphin-Tanguy, C. Niesner, F. Guillemard, and M. Pengov. Symbolic Determination of the Steady State due to Gravity Effects on Mechanical Systems Modelled by Bond Graphs. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation, ICBGM'05*, volume 37(1) of *Simulation Series*, pages 101–106. SCS, 2005.
- [26] S. Ding. *Model-based Fault Diagnosis Techniques*. Springer, Berlin, 2008.
- [27] S.W. Director and R.A. Rohrer. The Generalized Adjoint Network and Network Sensitivity. *IEEE Trans. Circuit Theory*, CT-16:318–323, Aug. 1969.
- [28] M.A. Djeziri, R. Merzouki, B. Ould Bouamama, and G. Dauphin-Tanguy. Robust Fault Diagnosis by Using Bond Graph Approach. *IEEE/ASME Transactions on Mechatronics*, 12(6):599–611, December 2007.
- [29] T. Ersal, H.K. Fathy, and J.L. Stein. Structural Simplification of Modular Bond Graph Models Based On Junction Inactivity. *Simulation Modelling Practice and Theory*, 17(1):175–196, 2009.
- [30] T. Ersal, H.K. Fathy, and J.L. Stein. Orienting Body Coordinate Frames Using Karhunen-Loeve Expansion For More Effective Structural Simplification. *Simulation Modelling Practice and Theory*, 17(1):197–210, 2009.
- [31] G.J. Balas et al. *μ -Analysis and Synthesis Toolbox, version 3, June 1998*. URL <http://www.Mathworks.com>.

- [32] P.J. Feenstra, P.J. Mosterman, G. Biswas, and P.C. Breedveld. Bond Graph Modeling Procedures for Fault Detection and Isolation of Complex Flow Processes. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM'01*, pages 77–82, Phoenix, Arizona, USA, January 7–11 2001. SCS Publishing. Simulation Series, volume 33, no 1, ISBN: 1-56555-221-0.
- [33] P. J. Gawthrop. Control System Configuration: Inversion and Bicausal Bond Graphs. In J. J. Granda and G. Dauphin-Tanguy, editors, *1997 International Conference on Bond Graph Modeling, and Simulation (ICBGM'97)*, pages 97–102. SCS Publishing, 1997. Simulation Series, volume 29, no 1, ISBN: 1-56555-103-6.
- [34] P.J. Gawthrop. Sensitivity Bond Graphs. *Journal of the Franklin Institute*, 337:907–922, 2000.
- [35] P.J. Gawthrop. MTT: Model Transformation Tools. In F.E. Cellier and J.J. Granda, editors, *ICBGM'95, International Conference on Bond Graph Modeling and Simulation*, volume 27(1) of *Simulation Series*, pages 197–202. SCS Publishing, 1995.
- [36] P.J. Gawthrop. Bicausal Bond Graphs. In F.E. Cellier and J.J. Granda, editors, *ICBGM'95, International Conference on Bond Graph Modeling and Simulation*, volume 27(1) of *Simulation Series*, pages 83–88. SCS Publishing, 1995.
- [37] P.J. Gawthrop and L. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [38] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker Inc., New York, USA, 1998.
- [39] S. K. Ghoshal. *Model-based Fault Diagnosis and Accommodation using Analytical Redundancy: A Bond Graph Approach*. PhD thesis, Dept. of Mechanical Engineering, Indian Institute of Technology, Kharagpur, India, 2006.
- [40] S.K. Ghoshal, A.K. Samantary, and A. Mukherjee. Improvements to Single Fault Isolation Using Estimated Parameters. In J.J. Granda and F.E. Cellier, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM'05*, pages 301–306, New Orleans, Louisiana, USA, January 23–27 2005. SCS Publishing. Simulation Series, volume 37, no 1, ISBN: 1-56555-287-3.
- [41] R. Isermann. Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control*, 29:71–85, 2005.
- [42] Th. Kailath. *Linear Systems*. Englewood Cliffs, N. J., USA, 1980.
- [43] C.S. Kam. *Les bond graphs pour la modélisation des systèmes linéaires incertains*. PhD thesis, L'Ecole Centrale de Lille et L'Université des Sciences et Technologies de Lille, Lille, France, 2001.
- [44] C.S. Kam and G. Dauphin-Tanguy. Bond graph tools for standard interconnection structure determination. In J. J. Granda and G. Dauphin-Tanguy, editors, *2001 International Conference on Bond Graph Modeling, and Simulation (ICBGM 2001)*. SCS Publishing, 2001. Simulation Series, volume 33, no 1, ISBN: 1-56555-221-0.
- [45] C.S. Kam and G. Dauphin-Tanguy. Sensitivity function determination on a bond graph model. In *Simulation in Industry, 13th European Simulation Symposium 2001, ESS'01*, pages 735–739, Marseille, France, 18–20 Oct 2001. SCS.
- [46] D.C. Karnopp and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1975.
- [47] L.S. Louca and J.L. Stein. Energy-based model reduction of linear systems. In I. Troch and F. Breitenecker, editors, *Proc. MATHMOD 09 Vienna - Full Papers CD Volume*, number 35 in ARGESIM-Report, pages 629–639. ARGESIM, 2009. ISBN 978-3-901608-35-3.
- [48] L.S. Louca and B.U. Yildir. Modelling and reduction techniques for studies of integrated hybrid vehicle systems. *Mathematical & Computer Modelling of Dynamical Systems*, 12 (2-3):203–218, 2006.
- [49] L.S. Louca, J.L. Stein, G.M. Hulbert, and J. Sprague. Proper model generation: An energy-based methodology. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proceedings of the International Conference on Bond Graph Modeling*, volume 29, pages 44–49. SCS Publishing, 1997.

- [50] W. Marquis-Favre, X. Xia, and S. Scavarda. Bicausality-based Procedures for Transfer and Transmission Matrix Determination of Single Source Single Load Linear Systems. In F.E. Cellier and J.J. Granda, editors, *2003 International Conference on Bond Graph Modeling, and Simulation (ICBGM 2003)*, pages 128–133. SCS Publishing, 2003. Simulation Series, volume 35, no 2, ISBN: 1-56555-257-1.
- [51] S. J. Mason. Feedback theory – further properties of signal flow graphs. In *Proc. IRE 44*, pages 920–926, 1956.
- [52] K. Medjaher, A.K. Samantaray, and B. Ould Bouamama. Diagnostic Bond Graphs for Direct Residual Evaluation. In J.J. Granda and F.E. Cellier, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM'05*, pages 307–312, New Orleans, Louisiana, USA, January 23–27 2005. SCS Publishing. Simulation Series, volume. 37, no 1, ISBN: 1-56555-287-3.
- [53] A. Mukherjee, R. Karmakar, and A.K. Samantaray. *Bond Graph in Modeling, Simulation and Fault Identification*. I.K. International Publishing House, New Delhi, India, 2006. ISBN: 81-88237-96-5.
- [54] R.F. Ngwompo and P.J. Gawthrop. Bond graph-based simulation of nonlinear inverse systems using physical performance specifications. *Journal of the Franklin Institute*, 336(8):1225–1247, 1999.
- [55] R.F. Ngwompo, S. Scavarda, and D. Thomasset. Inversion of linear time-invariant SISO systems modelled by bond graphs. *Journal of the Franklin Institute*, 333(2):157–174, 1996.
- [56] A.Y. Orbak, O.S. Turkay, E. Eskinat, and K. Youcef-Tourmi. Model reduction in the physical domain. *Proc. Institute of Mechanical Engineers Part I: Journal of Systems and Control Engineering*, 217:481–496, 2003.
- [57] P.H.Roe and J.U. Thoma. A new bond graph approach to sensitivity analysis. In I. Troch and F. Breitenecker, editors, *Proc. 3rd MATHMOD Vienna, IMACS Symposium on Mathematical Modelling*, pages 743–746. ARGESIM, 2000. ISBN: 3-901608-15-X.
- [58] A. Rahmani. *Etude structurelle des systèmes linéaires par l'approche bond graph*. PhD thesis, L'Université des Sciences et Technologies de Lille, Lille, France, 1993.
- [59] A. Rahmani and G. Dauphin-Tanguy. Structural analysis of switching systems modelled by bond graph. *Mathematical and Computer Modelling of Dynamical Systems*, 12(2-3):235–247, April-June 2006.
- [60] D.G. Rideout and J.L. Stein. Breaking Subgraph Loops for Bond Graph Model Partitioning. In J. J. Granda and F. Cellier, editors, *Proc. of the 2007 International Conference on Bond Graph Modeling and Simulation*, volume 39 (1), pages 241–249, 2007.
- [61] D.G. Rideout, J.L. Stein, and L.S. Louca. Power-Based Dynamic System Model Decoupling and Reduction Using Bond Graphs. *Simulation Modelling Practice and Theory*, 17(1):271–292, 2009.
- [62] R.C. Rosenberg. State-Space Formulation for Bond Graph Models of Multiport Systems. *Journal of Dynamic Systems, Measurement, and Control*, pages 35–40, March 1971.
- [63] R.C. Rosenberg and A.N. Andry. A controllability test for linear systems using a graphical technique. In *Proc. of the IFAC Symposium*, pages 149–147, August 1979. Zürich, Switzerland.
- [64] A. K. Samantaray and S. K. Ghoshal. Bicausal bond graphs for supervision: From fault detection and isolation to fault accommodation. *Journal of the Franklin Institute*, 345:1–28, 2008.
- [65] A.K. Samantaray and B. Ould Bouamama. *Model-based Process Supervision – A Bond Graph Approach*. Advances in Industrial Control. Springer, London, 2008. ISBN 978-1-84800-158-9.
- [66] A.K. Samantaray, K. Medjaher, B. Ould Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy. Diagnostic bond graphs for online fault detection and isolation. *Simulation Modelling Practice and Theory*, 14(3):237–262, 2006.
- [67] Scilab Consortium. Scilab. URL <http://www.scilab.org/>.
- [68] R.N. Shields and J.B. Pearson. Structural Controllability of Multiinput Linear Systems. *IEEE Trans. Automatic Control*, AC-21(2):203–212, 1976.

- [69] K. Sia and A. Naamane. Bond Graph: a suitable tool for component faults diagnostic. In J.J. Granda and F.E. Cellier, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM'03*, pages 89–103, Orlando, Florida, USA, January 19–23 2003. SCS Publishing. Simulation Series, volume 35, no 2, ISBN: 1-56555-257-1.
- [70] S.Li and L. Petzold. Design of New DASPCK for Sensitivity Analysis. Technical report, UCSB, 1999. URL <http://www.engineering.ucsb.edu/~cse/>.
- [71] M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37:687–699, 2001.
- [72] C. Sueur and G. Dauphin-Tanguy. Bond-graph Approach for Structural Analysis of MIMO Linear Systems. *Journal of the Franklin Institute*, 328(1):55–70, 1991.
- [73] C. Sueur and G. Dauphin-Tanguy. Bond graph approach to multi scale systems analysis. *Journal of the Franklin Institute*, 328:1005–1026, 1991.
- [74] M. Tagina, J.P. Cassar, G. Dauphin-Tanguy, and M. Staroswiecki. Monitoring of Systems Modelled by Bond-Graphs. In F.E. Cellier and J.J. Granda, editors, *ICBGM'95, International Conference on Bond Graph Modeling and Simulation*, volume 27(1) of *Simulation Series*, pages 275–280, Las Vegas, Nevada, USA, 15–18 Jan 1995. SCS Publishing.

Chapter 7

Models of Variable Structure

As explained in the beginning of Chapter 2, bond graph modelling starts from the intuitive idea that the dynamic behaviour of a system is determined by the energy exchange between system components. If radiation through empty space is assumed to be zero, then the energy exchange between system components is bound to real technical devices, which in most cases connect components permanently. Such connections may be, for example, mechanical shafts and joints, electrical wires, or hydraulic pipes or hoses. Since the energy exchange between components is represented by the edges of a bond graph, first of all, bond graphs are suited to represent systems of time-invariant structure. Moreover, the fact that an exchange of energy is bound to real physical links entails that it happens in conjunction with an exchange of physical quantities, e.g. momentum, mass, or electrical charge. The rate at which such physical quantities change is a continuous function of the location where the change happens and of time. In other words, transitions between energetic modes always need a finite amount of time and do not happen instantaneously. Nevertheless, given a macroscopic time scale, such an abstraction often is convenient and appropriate. For instance, when the piston in a hydraulic valve or in a cylinder hits a stop, then an energy exchange takes place for a short period of time which, in general, is not significant for the dynamic behaviour of the valve or the cylinder. In addition, in this case, the energy exchange does not take place continuously, but repeatedly during a very short time. Since the transformation of the kinetic energy of the piston into heat during a negligible oscillation with the stop does not effect the dynamics of the valve, it is acceptable to assume an abrupt stop of movement in which the kinetic energy is instantaneously transformed into heat. That is, reaching a limiting position, the piston's velocity versus time curve discontinuously drops to zero. Other similar well known examples are mechanical as well as electrical switches, hydraulic check valves, electrical diodes and thyristors, mechanical clutches and stick-slip effects between bodies. Such elements or phenomena entail changes of the model structure. From a macroscopic time scale view, it is appropriate to consider them as discrete time events and to describe them discontinuously. When two bodies stick together for some time, they may be considered as a single body for that time. As a result, the structure of the model and its order change. Similarly, open

switches, reversed biased electrical diodes and hydraulic check valves partition an electrical or a hydraulic network into subnetworks if it is assumed that in reverse mode there is no current or volume flow. Since such elements mostly switch very fast in comparison to the dynamics of the other system components, the abstraction of an instantaneous discontinuous transition from one mode to another is justified. In contrast, a physical system modelling approach strictly based on conservation laws of physics would require a microscopic time scale view. In practice, this, however, entails a number of disadvantages. Taking into account the dynamics of negligible and parasitic effects may considerably and unnecessarily increase model complexity and computational time without significant improvements in accuracy. Moreover, a more detailed model usually introduces new parameters for which realistic values may be difficult to obtain. Finally, numerical problems may emerge during the simulation since a microscopic view accounts for transients that are orders of magnitude faster compared to the overall system behaviour. Such models will result in a stiff set of equations which are usually more difficult and problematic to solve. Clearly, the general principle should also hold for bond graph based physical systems modelling that a model should be as simple as possible and, with respect to its purpose, as detailed as necessary. The question posed is how dynamic systems models including some components described discontinuously can be represented in a bond graph framework. This question has been the topic of numerous publications during the last decade (see, for instance, [1, 4–6, 8, 11, 12, 18, 25, 29, 32]) and has resulted in a number of dissertations [19, 28, 31] and the habilitation thesis of Buisson [9]. In the following sections, some approaches will be considered.

7.1 Bond Graph Models with Fixed Causalities

7.1.1 *Extending Element Characteristics*

If the displacement of a body is limited by a stop, as this is the case in some hydraulic valves, then a straightforward modelling approach is to model the stop by means of a spring damper pair. Thus, the contact between the body and the stop can be represented by the simple bond graph depicted in Figure 7.1.

This bond graph can only be used as long as the body is in contact with the stop. When the body is freely moving (think of the well known bouncing ball problem) then there is no C and R element affecting the motion of the body. In other words, there are two different bond graphs describing the two modes of such a system. If we wish to have a single bond graph valid for both modes (free motion, or body in contact with the stop), then this can be achieved by artificially extending the characteristic of the C energy store such that the spring stiffness is equal to zero as long as the body is not in contact with the stop. Similarly, the resistance, R, for the contact has a non-zero value only if the body is in contact with the stop. For

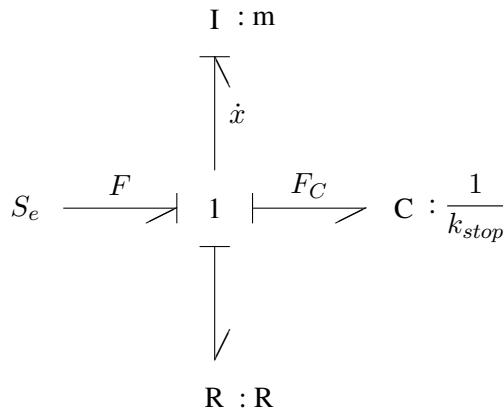


Fig. 7.1 Bond graph representing the contact between a rigid body and a stop

example, if the body is a spool valve that moves against a biased spring, then the spring characteristic can be superimposed with the spring characteristic of the stop.

This way, a stop can be approximately modelled without the need for an extension of the bond graph methodology. If the original model describing the free body is in state space form, then it remains in state space form if the stop is modelled by extending the spring characteristics as shown in Figure 7.2. The advantage of a single set of equations for both modes, however, comes with some considerable disadvantages. Firstly, the bond graph does not reflect the change in model structure. The change between modes is hidden in the characteristics of some elements. Moreover, this modelling approach is not really satisfying. If, in a spool valve, for

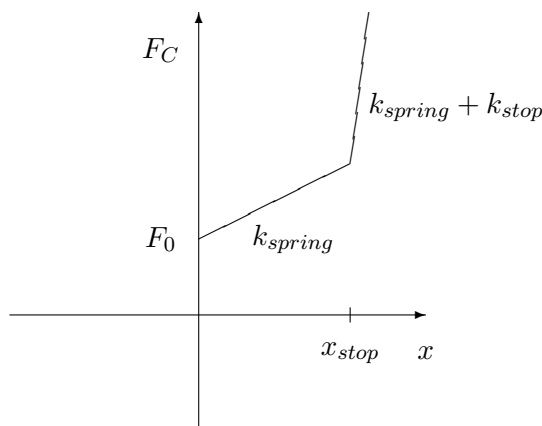


Fig. 7.2 Superposition of the characteristics of a biased spring and a stop

instance, the spool reaches a stop and is kept in that position for a while by the net force acting on it, then it is appropriate to represent the valve for this time period by a model that accounts only for the valves' orifices. If the stop is taken into account by an extended spring characteristic, then the stiffness of the C energy store is significantly increased when the stop is reached. By consequence, a high frequency oscillation is introduced that is not relevant for the main dynamic behaviour of the system. In order to assure that these oscillations die out quickly and in order to avoid unrealistic bouncing of the body at the stop, the parameter value of the resistor must be appropriately chosen. Modelling a stop by means of a spring damper pair follows the above mentioned microscopic view and may lead to numerical stiff model equations. During the high frequency oscillation at the stop, the kinetic energy of the body is converted into heat. Since these oscillations die out after a very short time, it is reasonable to assume that it approaches zero, which means that the kinetic energy of the body is instantaneously converted into heat. This modelling view, however, cannot be expressed by means of conventional bond graphs. A possible approach is presented in Section 7.3.

7.1.2 Switching Between System Modes by means of Modulation

If we want to keep the structure of the bond graph unchanged and if we want to express both system modes (free motion and body in contact with stop) explicitly in the bond graph, instead of accounting for them in the characteristics of the elements, it is possible by means of a transformer with a modulus controlled by a Boolean variable [5, 32]. In this approach one part of the bond graph represents the moving body and the other one accounts for the stop. Both parts are structurally permanently connected by the MTF (Figure 7.3). When the body is not in contact with the stop, then this is merely expressed by a modulus value equal to zero.

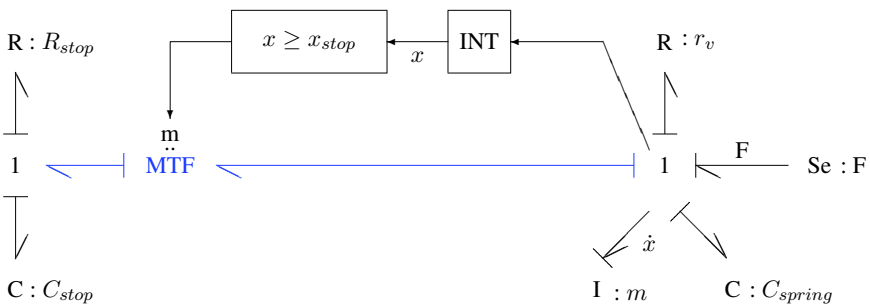


Fig. 7.3 Switching a mechanical stop on or off by means of a MTF

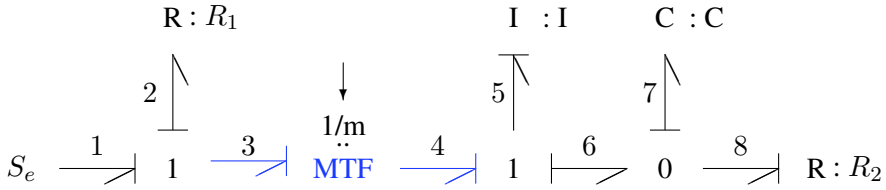


Fig. 7.4 Bond graph of an electrical circuit with an MTF modelling an ideal switch (Strömberg, Top and Söderman 1993)

The transformer modulated by a Boolean variable has been used by Ducreux, Dauphin-Tanguy and Rombaut [18] for modelling resistive elements, e.g. electrical diodes as non-ideal switches. The exponential characteristic is approximated by a piecewise linear function and the Boolean transformer modulus enables one to switch between a forward and a reverse biased mode. The MTF with Boolean modulus in Figure 7.3 can be viewed as an ideal switch that connects or disconnects the two submodels. However, as Strömberg et al. have shown by means of the bond graph of an electrical circuit reproduced in Figure 7.4 [32], MTFs require special consideration if used for modelling ideal switches. If the switch is open ($m = 0$), the current, f_5 , through the I energy store and thus the current, f_6 , into the parallel connection of the capacitor and the resistor should be zero. From the bond graph, however, only the condition $f_5 = f_6$ can be obtained. Strömberg et al. point out that this problem is due to the fact that causalities of the MTF do not change with the switch state.

In devices like electrical diodes and hydraulic check valves, the transition from one mode into the other takes place very fast in comparison to the dynamic behaviour of the whole system. Hence, the assumption of an instantaneous discontinuous transition is appropriate. In the following section, a closer look will be taken at the modelling of hydraulic check valves. Electrical diodes may be modelled in a similar way. In hydraulic check valves, a biased spring of low stiffness forces a ball of small mass to close a bore. If the hydraulic pressure at the valve’s entrance exceeds a set point, p_s , determined by the biased spring, then the valve opens very fast and the fluid flow through the open valve experiences a low resistance R_{on} . Since the mass of the valve poppet and the stiffness of the spring are low, the dynamics can be neglected. Hence, the valve can be modelled as a resistor with a nonlinear flow characteristic. As for electrical diodes, the characteristic can be approximated by a piecewise linear relation. Its slope changes at the threshold value p_s . If the flow characteristic is approximated by a piecewise linear curve as depicted in Figure 7.5, then the resistor representing the valve can only accept conductance causality.

The piecewise linear characteristic implies a switching between switch states ‘closed’ and ‘open’. In the first mode, the valve completely blocks the volume flow. In the mode ‘open’, there is a small ON-resistance R_{on} . A change of the switch state happens if the pressure drop across the valve, $|\Delta p|$, exceeds or drops below the set point p_s . If the primary purpose of the bond graph model is not simulation but

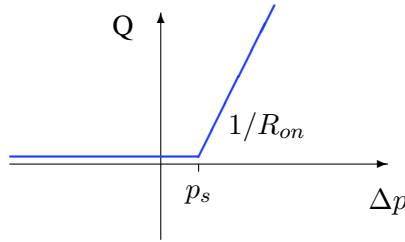


Fig. 7.5 Piecewise linear approximation of the static characteristic of a check valve

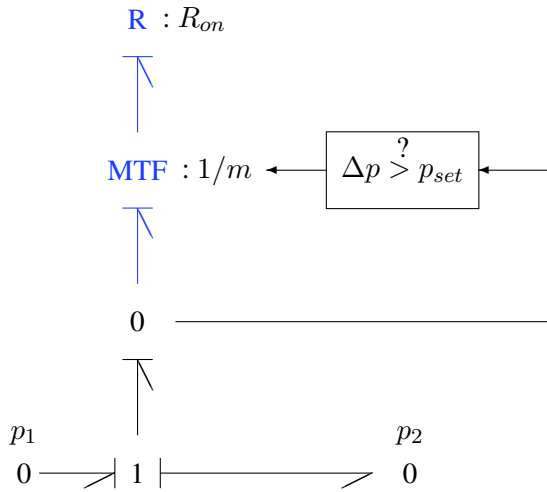


Fig. 7.6 Static bond graph model of a fast switching element (Ducreux, Duaphin-Tanguy and Rombaut , 1993)

the controller design, Ducreux et al. propose to represent fast switching elements, e.g. check valves not by a resistor with a piecewise linear characteristic, but to combine a linear resistor with a transformer of Boolean modulus m controlled by the effort difference across the element (Figure 7.6). For $m = 0$, the part of the linear characteristic in the third quadrant is switched off.

The transformer with Boolean modulus explicitly expresses the switching. If the absolute value of the pressure difference exceeds the set point, then its modulus is set to one. The ON-resistance then determines the volume flow. If the absolute value of the pressure difference is below the threshold, then the transformer modulus and the flow between the two pressures p_1 and p_2 is equal to zero. In this model, the pressure difference is the input. Since the pressure difference controls the transformer modulus, we have a model with internal modulation (Definition 4.17). Independent of the switch state, the output of the model is the flow through the element. Consequently, if it is connected in series with an I energy store, the latter must have

derivative causality. The linear R element takes into account that *real* fast switching elements dissipate free energy. The abstraction of *ideal* switches is discussed in Sections 7.1.3, 7.1.4 and 7.2. Since the value of the ON-resistance, R_{on} , is small in general, it may lead to small time constants in the linearised model equations. An advantage of this switch model is that a bond graph can be constructed in the usual way and that the structure of the bond graph and its causalities remain unchanged for all switch states. This means that a unique set of model equations can be derived from the bond graph that holds for all configurations of switch states. If, apart from the fast switching elements, all other elements are linear and if transformers and gyrators have a constant, time invariant modulus or ratio, then transfer functions can be determined either from the time domain state equations or directly from the bond graph. The coefficients of these transfer functions depend on the moduli of the switches. This approach has been used by Ducreux et al. [18] and by J. Garcia [20] for the modelling of commutation phenomena in power electronic circuits.

Of course, in Figure 7.6, the model of a switch can also take resistance causality. If all switches in a circuit are represented according to Figure 7.6 and if conductance causality is assigned to all switch models, then causal conflicts at 0-junctions may result as has been shown by Dauphin-Tanguy and Rombaut for the case of an elementary commutation cell (Figure 7.7) being a basic building block of power electronic converters [17]. Figure 7.8 displays a bond graph of the elementary commutation cell in which conductance causality has been assigned to both switch models.

In order to avoid the causal conflict resulting at the 0-junction, Dauphin-Tanguy and Rombaut suggest to add a C element (or R element) that has no physical meaning. This is justified because if in the equations derived from the causally corrected bond graph the currents into additional C elements are set to zero, then the modified equations can be interpreted as those of an equivalent circuit [17]. Motivated by this observation which was confirmed by a number of examples, Dauphin-Tanguy and Rombaut proposed to use the model in Figure 7.6 for all switches, to assign unique static conductance causality to all switch models for all switch states and to

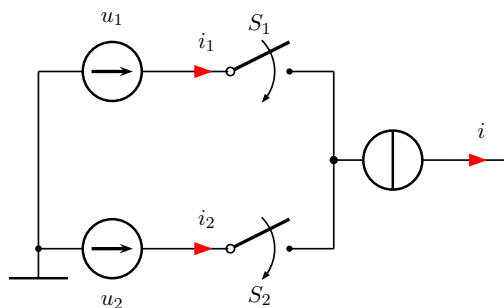


Fig. 7.7 Schematic of an elementary commutation cell (Dauphin-Tanguy and Rombaut, 1993)

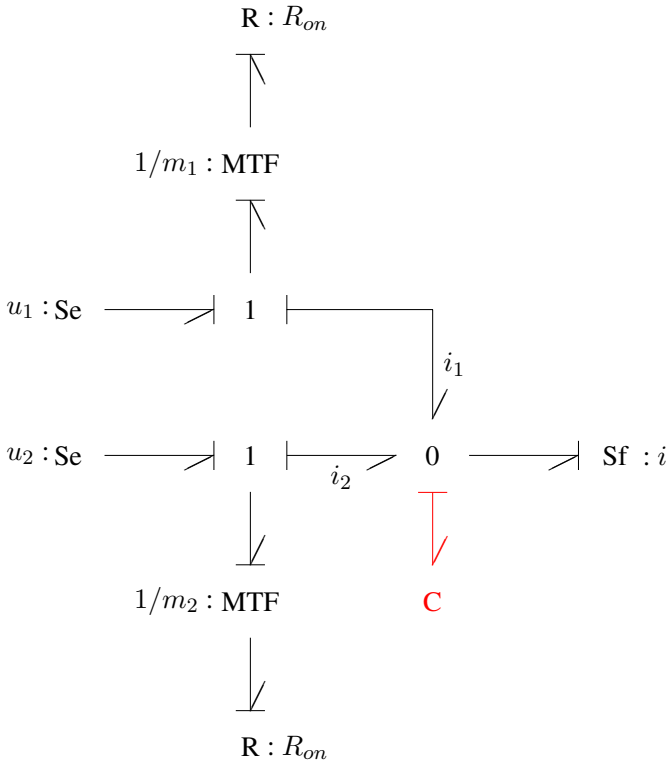


Fig. 7.8 Bond of the elementary commutation cell (Dauphin-Tanguy and Rombaut, 1993)

solve resulting causal conflict at 0-junctions by adding a resistor or a capacitor with a parameter of small value that is suppressed in the mathematical model. That is, the physical model is slightly modified and the derived mathematical model is to be manipulated. The result is a unique valid mathematical model covering all switch stages.

In her dissertation [20], Garcia also takes into account thermal effects in switching elements. R elements are replaced by RS elements. In an unconventional but pragmatic way, the RS elements are the interface to pseudo bond graphs that model heat conduction in semiconductors and in their housing or account for the radiation of cooling devices. This way, the temperature effect on the electrical operating point can be taken into account and the design of cooling devices can be improved.

In Figure 7.9, the pseudo bond graph accounting for the thermal behaviour is portrayed by means of thin lines. The right upper 0-junction represents the temperature of the diode relative to the ambient temperature, while Q denotes the heat flow that corresponds to the losses of electrical energy. The C element takes into account the heat storage in the semiconductor. Heat conduction to the carrier on which the de-

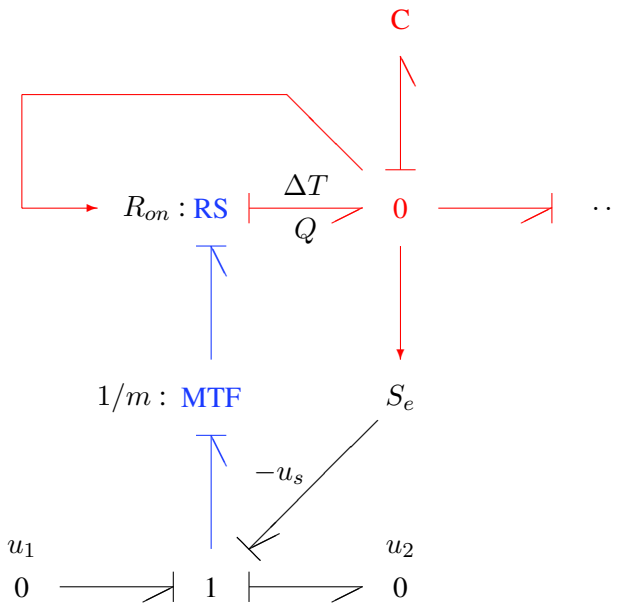


Fig. 7.9 Electro-thermal model of a diode (Garcia Gomez, 1997)

vice is mounted is not shown in Figure 7.9. In comparison to the model in Figure 7.6, it can be seen that there is an additional effort source. Moreover, the ON-resistance R_{on} is modulated. Both additional elements take into account that the threshold of the forward voltage as well as the ON-resistance change with temperature.

7.1.3 Switched Power Junctions

To account for ideal, no power consuming switching in a unique bond graph with fixed causalities, Umarikar and Umanand introduced so-called *switched power junctions* (SPJs) [37]. The idea is to allow for more than one bond imposing an effort on a 0-junction and more than one bond imposing a flow on a 1-junction. However, at each time instant, only one of these bonds determines the causality passed on by the junction so that there is no causal conflict. If there are n bonds that want to impose efforts e_1, \dots, e_n on a 0-junction then, at each time instant, one of these bonds can be activated while deactivating the remaining $n - 1$ bonds by expressing the effort imposed on that 0-junction, e_{0_s} , as a weighted sum of all n efforts e_1, \dots, e_n .

$$e_{0_s} = b_1 \times e_1 + \dots + b_n \times e_n \quad (7.1)$$

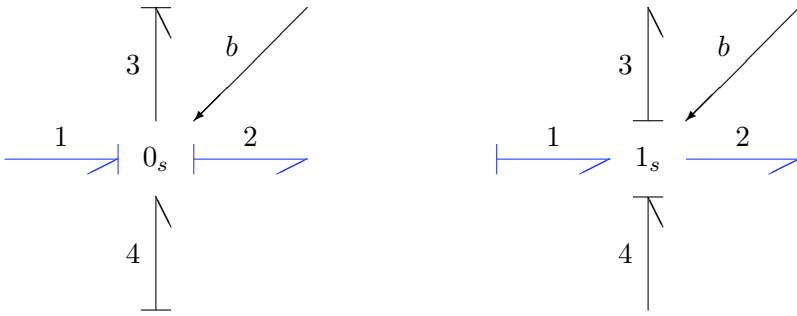


Fig. 7.10 Switched power junctions with two bonds that want to determine their causality

In Equation 7.1, the variables b_1, \dots, b_n can take values either 0 or 1 and only one of them can be equal one at a time. In this way, bonds impinging at a junction can be switched on and off connecting model parts and disconnecting from others. Figure 7.10 shows a 0- and 1-junction for the case that two bonds want to determine their causality.

The switching of the junctions is indicated by an activated bond delivering the signal that switches on one of bonds 1 and 2 while switching of the other. Let f_{0_s} denote the flow at the bond determining the causality at a junction then the equations of the switched 0-junction in Figure 7.10 are

$$e_{0_s} = b \times e_1 + (1 - b) \times e_2 \quad (7.2a)$$

$$f_{0_s} = f_4 - f_3. \quad (7.2b)$$

A closer look at switched power junctions immediately reveals that they can be reduced to an interconnection of standard bond graph junctions and boolean modulated transformers as they have been used in the previous section. Apparently, the switched 0-junction in Figure 7.10 can be equivalently represented by the bond graph in Figure 7.11. The equivalent representation of a switched 1-junction has the same structure as in Figure 7.11. Only the 0- and the 1-junctions are interchanged [22].

Example: The Bouncing Ball Problem

For illustration, the switched power junction approach is applied to the well known bouncing ball problem. It is assumed that the ball is elastic and touches on a hard surface that does not move. Clearly, there are two modes of operation. Either the ball is moving in the air experiencing gravity force only, or it is in contact with the ground experiencing elastic deformations. Figure 7.12 shows a bond graph model with fixed causalities for both modes that makes use of two switched power junctions.

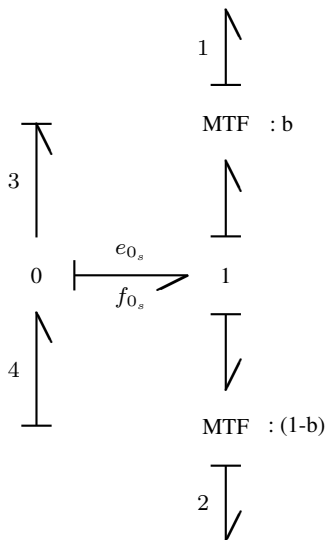


Fig. 7.11 Equivalent representation of the switched 0-junction in Figure 7.10 by standard bond graph elements

In the bond graph of Figure 7.12, the two modes are taken into account by the signal b . For $b = 1$, the bond labelled b is activated, while the bonds labelled \bar{b} are switched off. In this mode, the ball is in contact with the ground. The other mode (ball in the air) is captured by the value $\bar{b} = 0$. In this case, the bond labelled b is switched off and the two bonds labelled \bar{b} are activated. That is, although there are two bonds for each of the two switched power junctions that want to determine its causality, actually only one does at each point in time.

If the two switched junctions are replaced by standard junctions and modulated transformers according to the equivalent representation of a switched 0-junction in Figure 7.11, then the result is the bond graph displayed in Figure 7.13. As can be clearly seen from the bond graph of Figure 7.13, due to boolean modulated transformers, the force f_b is either zero (ball in the air) or it is equal to the sum of spring force and friction force (ball in contact with the ground). Also, due to the boolean transformers, either the velocity $v_g - v$ or the velocity $-F_s/r$ is input into the right-hand side 1-junction. The latter velocity means that after a reflection from the ground, the centre of gravity and the former contact point of the elastic ball still move with different velocities.

For $b = 1$ (ball in contact with the ground), the bond graph in Figure 7.13 could be reduced to the one shown in Figure 7.14. The other mode of operation may be presented by the simplified bond graph shown in Figure 7.15.

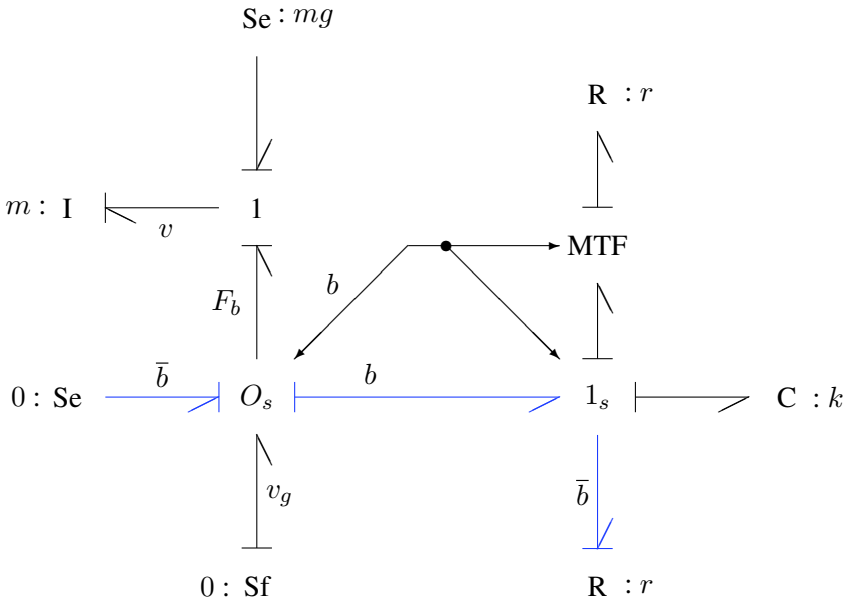


Fig. 7.12 Bond graph model of the bouncing ball problem with fixed causalities for both modes using two switched power junctions

When the ball hits the ground ,it is deformed and energy is stored in the spring representing its elasticity. When the downward motion of the ball’s centre of gravity comes to a halt ($v = 0$), momentum reverses, the compressed spring starts to expand and some of its stored energy is lost in the damper. The ball starts to release from the ground when the normal force F_b (Figure 7.12) becomes zero. When the ball is in the air again, only the gravity force acts on it. Immediately after the ball has released from the ground, the spring fully expands and discharges its remaining energy into the damper. That is, the ball regains its shape.

In conclusion, the generalisation of standard 0- and 1-junctions into switched power junctions allows for including ideal no power consuming switching into a compact bond graph representation with fixed causalities for all modes of operation. Bonds that are relevant for a mode under consideration can be easily identified by the boolean variable attached to them. Hence, a bond graph with switched junctions can be read by disregarding parts of it. As has been indicated and illustrated by means of a bond graph for the bouncing ball problem, switched power junctions can be replaced by standard 0- and 1-junctions and boolean modulated transformers. In some cases, the result may be a somewhat less compact bond graph.

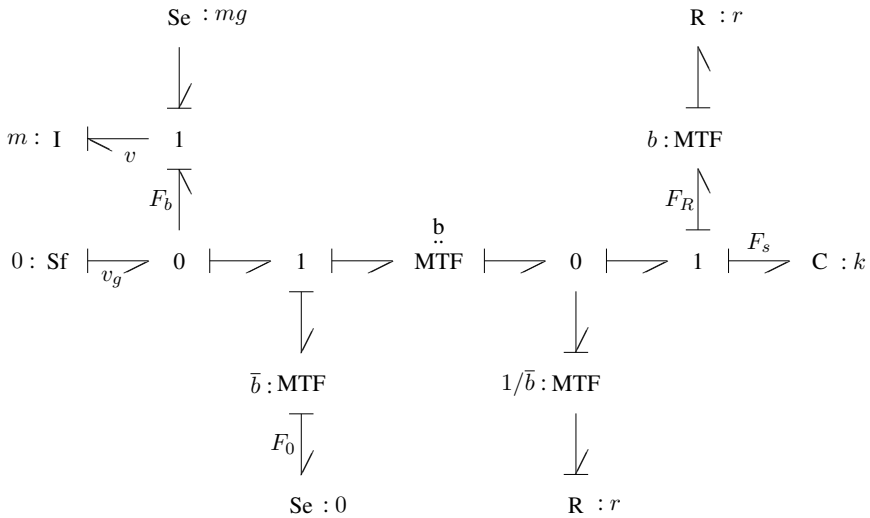


Fig. 7.13 Bond graph model of the bouncing ball problem with fixed causalities for both modes using standard elements

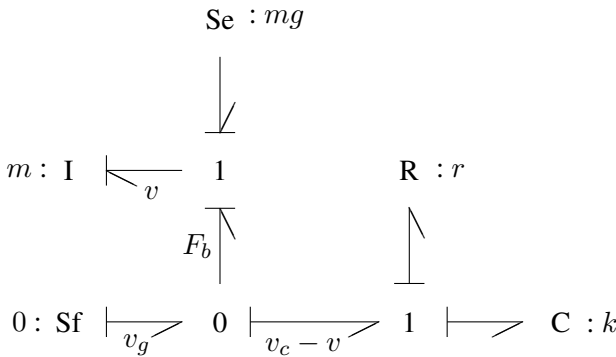


Fig. 7.14 Bond graph model of the bouncing ball in contact with the ground

7.1.4 Switching Off Degrees of Freedom by Sinks of Invariant Causality

In [4], the author proposes another possibility of representing models using the abstraction of instantaneous mode transitions by means of bond graphs of time-invariant structure and time-invariant causality. As a result, a single set of equations can be derived from the bond graph for all configurations of switch states. In this approach, sinks of fixed causality are used in order to impose an effort or a flow at the advent of a discrete event such that there is an instantaneous state transition and

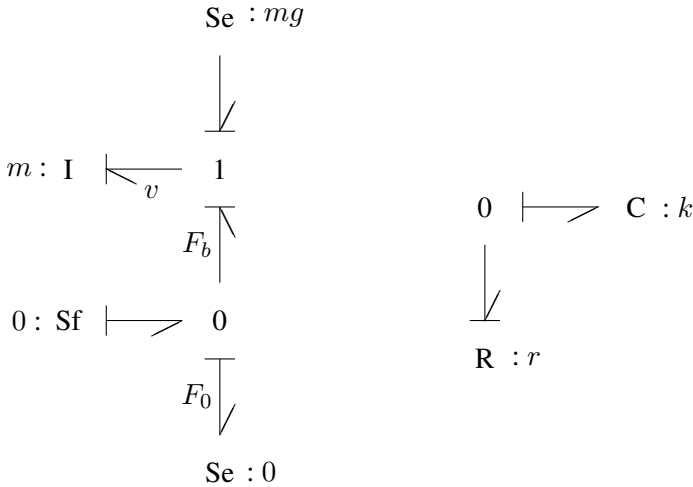


Fig. 7.15 Bond graph model of the bouncing ball moving in the air

the conditions of the new state are met. Using this approach, *ideal* switches can also be modelled.

Switches in Electronic Circuits

For illustration of the approach, consider the well known example of two electrical nodes in an integrated MOS circuit. Each of them has a capacitance to ground. Both vertices belong to different sub-circuits and are connected by a pass-transistor (Figure 7.16). The latter can be modelled as an ideal switch. Figure 7.17 shows the bond graph for Figure 7.16.

The ideal switch has been modelled by a transformer with the Boolean modulus b and a flow sink. If the switch is open, then the modulus and the current between two sub-circuits are equal to zero. On the contrary, if the switch is closed ($b = 1$), then the sink imposes a current, i_S , such that the voltages u_1 and u_2 across the

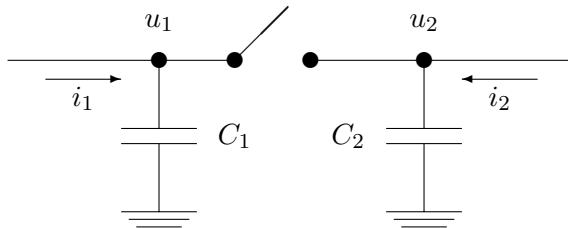


Fig. 7.16 Ideal switch between two capacitors

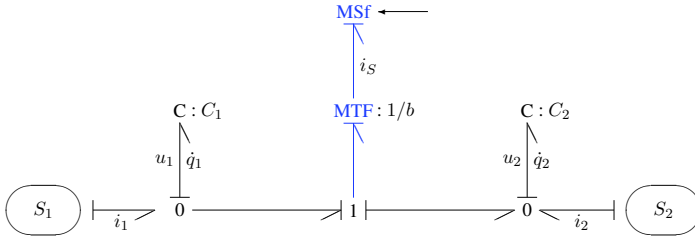


Fig. 7.17 Bond graph for Figure 7.16

capacitors are equal for all currents, i_1 and i_2 , charging the capacitors. Thus, the current, i_S , is not constant. Since an ideal switch is assumed, the voltages across the capacitors instantaneously jump from their present values to the new common one when the switch is closed. As a result, the current of the sink exhibits a Dirac pulse. The electrical energy that is lost at switching time is absorbed by the sink. For simplicity, consider the special case that the capacitances are equal and that the capacitors, first, are disconnected. Now, after charging one capacitor while the other is empty, the switch is closed. Clearly, the charge must be conserved and just distributes over two capacitors of equal capacitance. As a result, half of the electrical energy is lost. A simple computation yields the current, i_S , that must be imposed to ensure equality of the voltages u_1 and u_2 . If the currents charging the capacitors,

$$\dot{q}_1 = i_1 - i_S \tag{7.3a}$$

$$\dot{q}_2 = i_2 + i_S, \tag{7.3b}$$

are integrated with respect to time, then denoting the height of the pulse at time t_0 by I_0 , the charges are

$$q_1^+ - q_1^- = -I_0 \times 1s \tag{7.4a}$$

$$q_2^+ - q_2^- = I_0 \times 1s, \tag{7.4b}$$

where

$$q_i^+ = \lim_{t \rightarrow t_0^+} q(t) \tag{7.5a}$$

$$q_i^- = \lim_{t \rightarrow t_0^-} q(t) \quad (i = 1, 2). \tag{7.5b}$$

The condition

$$u_1 = \frac{1}{C_1} q_1^+ = u_2 = \frac{1}{C_2} q_2^+ \tag{7.6}$$

then yields the height, I_0 , of the switching pulse

$$I_0 \times 1s = \frac{1}{C_1 + C_2} [C_2 q_1^- - C_1 q_2^-]. \tag{7.7}$$

In self-switching elements, e.g. electrical diodes, such a pulse does not occur since these devices switch when the effort or the flow vanishes.

With $\dot{u}_1 = \dot{u}_2$ and $u_i = q_i/C_i$ ($i = 1, 2$), we obtain from Equations 7.3a and 7.3b for $t > 0$

$$i_S = \frac{1}{C_1 + C_2} [C_2 i_1 - C_1 i_2]. \quad (7.8)$$

With this current of the sink, the derivatives of the voltages across the capacitors read

$$\begin{aligned} \dot{u}_1 &= \dot{u}_2 \\ &= \frac{1}{C_1 + C_2} [i_1 + i_2]. \end{aligned} \quad (7.9)$$

The result corresponds to the fact that in the case of an ideal closed switch, both capacitors can be combined into one. Since in this modelling approach assigned causalities remain fixed independent of the switch states, this possible simplification is not expressed in the bond graph. The latter rather suggests the independence of both energy stores that does not exist any longer after the switch has been closed. Their interdependence becomes evident only indirectly due to the fact that the sink is modulated and imposes a current such that the voltages across the capacitors are equal.

If there are several ideal switches in the model, then the conditions that determine the currents to be imposed by the sinks may be dependent. As a result, a set of equations must be solved. For illustration, let us extend the example by adding a second switch in parallel to the capacitor, C_1 (Figure 7.18). The bond graph of the modified example circuit is shown in Figure 7.19. The switches can be controlled independently by external signals originating from other parts of the overall circuit.

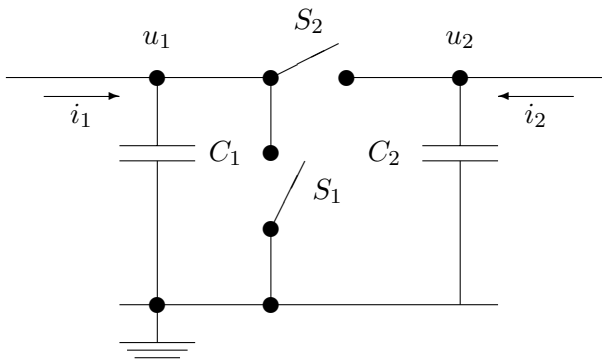


Fig. 7.18 Circuit with two ideal switches

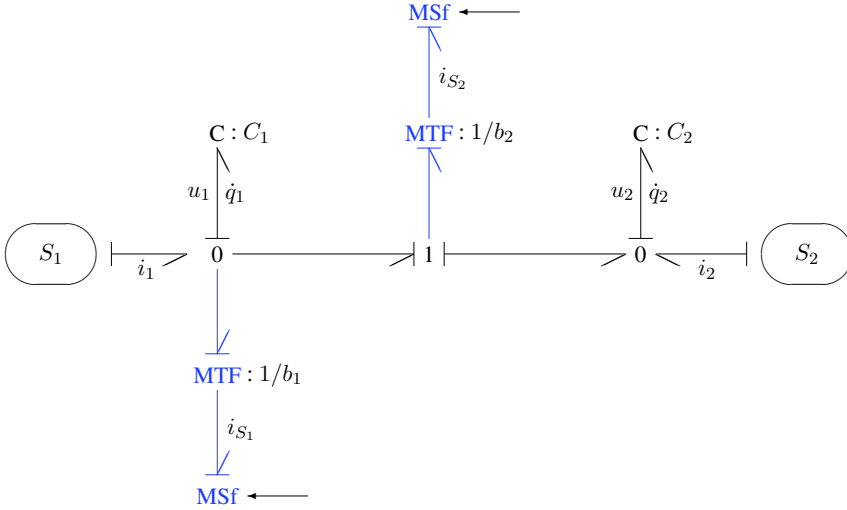


Fig. 7.19 Bond graph of the circuit with two ideal switches

Hence, in this case, two constraints can be formulated

$$b_1 \times u_1 = 0 \tag{7.10a}$$

$$b_2 \times (u_1 - u_2) = 0, \tag{7.10b}$$

where $b_i = 0$ ($i = 1, 2$) denotes that the i^{th} switch is open, while $b_i = 1$ indicates that it is closed. For the two capacitors, the following two state equations are derived from the bond graph

$$C_1 \dot{u}_1 = i_1 - b_1 \times i_{S1} - b_2 \times i_{S2} \tag{7.11a}$$

$$C_2 \dot{u}_2 = i_2 + b_2 \times i_{S2}. \tag{7.11b}$$

Differentiation of the constraints, Equation 7.10a and Equation 7.10b, and substitution into the state equations results in an algebraic system that for $t > t_0$ determines the currents i_{S1}, i_{S2} to be imposed by the sinks

$$\begin{bmatrix} \frac{b_1^2}{C_1} & \frac{b_1 \times b_2}{C_1} \\ \frac{b_1 \times b_2}{C_1} & \frac{b_2^2}{C_1} + \frac{b_2^2}{C_2} \end{bmatrix} \begin{bmatrix} I_{S1} \\ I_{S2} \end{bmatrix} = \begin{bmatrix} \frac{b_1}{C_1} i_1 \\ \frac{b_2}{C_1} i_1 - \frac{b_2}{C_2} i_2 \end{bmatrix}. \tag{7.12}$$

In the case $b_1 = 0, b_2 = 1$, i. e., switch S_1 is open, while switch S_2 is closed, the above equations give the previously obtained result (Equation 7.8). If both switches are closed, then both capacitors are short-circuited and the currents through the switches are

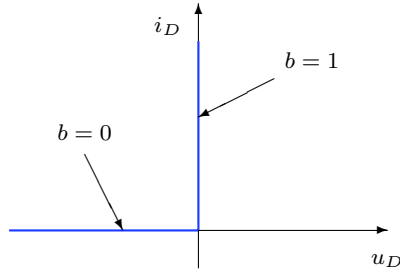


Fig. 7.20 Characteristic of an ideal diode

$$i_{S_1} = i_1 + i_2 \quad (7.13a)$$

$$i_{S_2} = -i_2 . \quad (7.13b)$$

The set of equations determining the currents to be imposed by the sinks must be solved whenever the state of one of the switches and hence the value of its Boolean variable changes. Let us assume that the switches represent ideal diodes with the characteristic given in Figure 7.20. Then, the value of the Boolean variable is equal to one as long as the voltage drop across the diode vanishes and the current is still positive. When the current becomes zero, the diode enters into the reverse mode and the value of the Boolean variable becomes zero. This means that a coefficient in the equations determining the currents of the sinks, has changed. Hence, the currents of the sinks must be re-evaluated. Thus, there is a control mechanism associated with each ideal switch that can be hardly expressed in a bond graph. The purpose of bond graphs is to represent energy flows, not the logic of control algorithms. In a later section, we will come back to this aspect. For the stepwise numerical integration of the dynamic model equations, it is important to check after each integration step whether an event has occurred. If this is the case, the time of the event must be located and the algebraic output equations of the sinks must be solved again.

This example with ideal switches shows another aspect. The two C energy stores in the bond graph both have integral causality. Even if there are no derivative causalities or algebraic loops in the models of the subsystems, the resulting mathematical model is in DAE form. This is because the currents of the sinks are multiplied by a Boolean variable in the state equations and because for each sink there is an algebraic constraint of the form $b \times u_S = 0$ where b is a Boolean variable and u_S the voltage drop across the switch. By closing the switches, energy stores may become dependent. Consequently, the differential index of the DAE system may change with the structure of the model. If, for instance, each of the subsystems S_1 and S_2 is a series connection of a voltage source and a resistance, then the bond graph of Figure 7.19 has the form shown in Figure 7.21. If the switch is open, then the two RC low pass filters are connected only by ground and independently operate. That is,

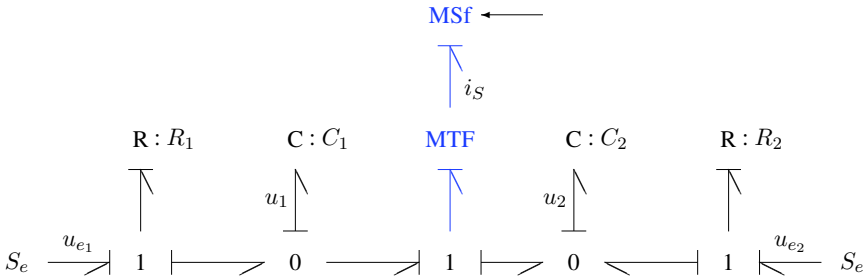


Fig. 7.21 Bond graph of two RC low pass filters connected by an ideal switch

there are two decoupled state equations for the capacitors and the differentiation index of the set of equations is zero. If the switch is closed, however, then the capacitors become dependent and the set of equations for the voltage drops across the capacitors and the current through the switch is of differential index 2.

$$\begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \frac{d}{dt}i_S \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1} & 0 & b \\ 0 & \frac{1}{R_2} & -b \\ b & -b & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ i_S \end{bmatrix} = \begin{bmatrix} u_{e1} \\ \frac{u_{e1}}{R_1} \\ \frac{u_{e2}}{R_2} \\ 0 \end{bmatrix} \quad (7.14)$$

Switching Off Degrees of Freedom in Mechanical Systems

The approach of modelling instantaneous state transitions by means of modulated transformers combined with a controlled sink imposing an effort or flow is not limited to ideal switches in models of electrical circuits. In mechanical systems, this approach corresponds to the accounting of constraint forces by means of Lagrange multipliers [24]. The sinks ensure that some degrees of freedom are switched off as long as certain conditions hold. The sinks are switched off at the moment when the system returns into the previous unconstrained state.

Example: The Bouncing Ball Problem

For illustration, let us come back to the well known bouncing ball problem considered in Section 7.1.1. Figure 7.22 shows the schematic and the corresponding bond graph.

In the bond graph of Figure 7.22, the upper part models the freely moving rigid ball, while the lower part represents the elastic upper layer of the floor. The variable λ denotes the constraint force that ensures that the velocity of the ball is equal to the one of the floor as long as the ball is in contact with the floor. In that case, the value of the Boolean transformer modulus is equal to one and the following equations can

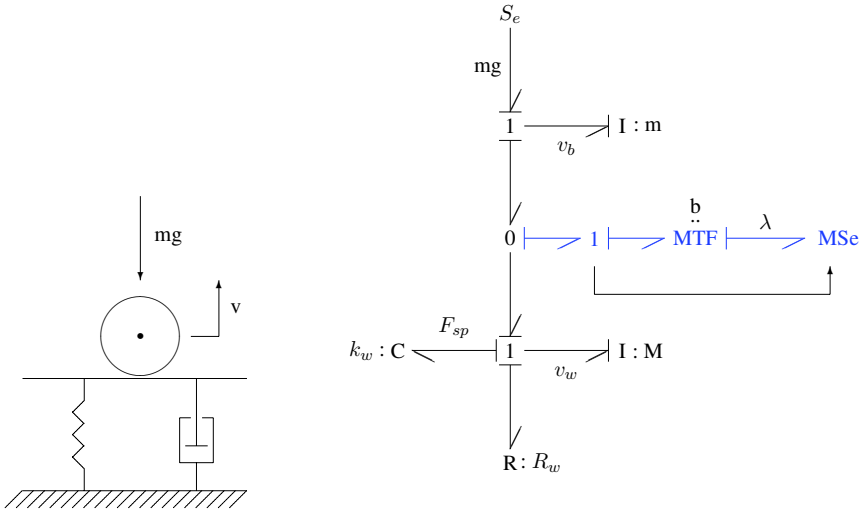


Fig. 7.22 Schematic and bond graph of the bouncing ball problem

be derived from the bond graph.

$$\dot{v}_b = \frac{1}{m}[mg - \lambda] \tag{7.15a}$$

$$\dot{F}_{sp} = k_w v_w \tag{7.15b}$$

$$\dot{v}_w = \frac{1}{M}[\lambda - F_{sp} - R_w v_w] \tag{7.15c}$$

$$0 = v_b - v_w \tag{7.15d}$$

Differentiation of the velocity constraint yields for the constraint force

$$\lambda = \frac{M}{m + M} mg + \frac{m}{m + M} [F_{sp} + R_w v_b]. \tag{7.16}$$

If the mass of the upper layer of the floor tends to zero ($M \rightarrow 0$), then the expression for the constraint force reduces to the equivalent of the sum of spring force and damping force. On the contrary, for $M \rightarrow \infty$, the constraint force λ equals the gravitational force mg .

Example: Stick-Slip Friction

The approach of using sinks of fixed causalities can also be used to model the well known stick-slip effect in dry friction. To demonstrate this, let us consider two rigid bodies. One of them moves against dry friction on the floor, while the second one is moving against dry friction on top of the first one as depicted in Figure 7.23.

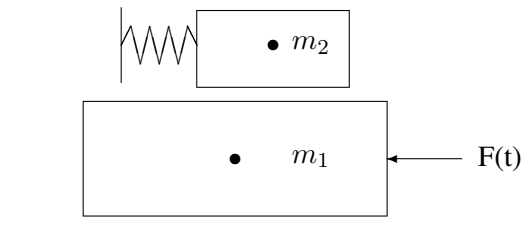


Fig. 7.23 Stick-slip friction between rigid bodies

The relation between the friction force F_f and the relative velocity between the two bodies is often approximated by the sign function, which is not defined for $v = 0$. The value of the friction force at $v = 0$ is determined by other elements of the model and can exceed the slip friction value F_s at $v \neq 0$ (Figure 7.24).

Sometimes the *sign* characteristic is *approximated* by the continuous arctan function having a unique single valued inverse. The disadvantage of both functions is that they do not properly take into account the sticking behaviour. Moreover, the steep gradient of the arctan function at $v = 0$ approximating the signum characteristic may cause numerical problems in the simulation.

The sticking behaviour can be captured by considering two bodies sticking together not as one single rigid body, but by assuming a constraint force λ between them. The latter ensures that the relative velocity between them is equal to zero. This constraint force is active as long as the absolute value of the net force is below the breakaway value, F_H , of the friction force (cf. Figure 7.24). This view is reflected in the bond graph of Figure 7.25. In this bond graph, effort sinks account for stick friction while the (nonlinear) resistors model slip friction. Both the sinks and the resistors are controlled by the constraint forces λ_i ($i = 1, 2$). A pair of modulated transformers with the Boolean moduli b_i and $\bar{b}_i = 1 - b_i$ ($i = 1, 2$) enables one

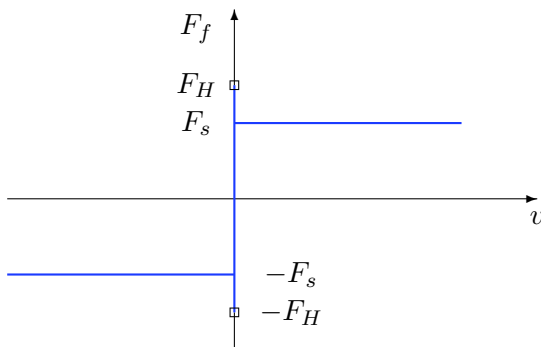


Fig. 7.24 Usual dry friction law

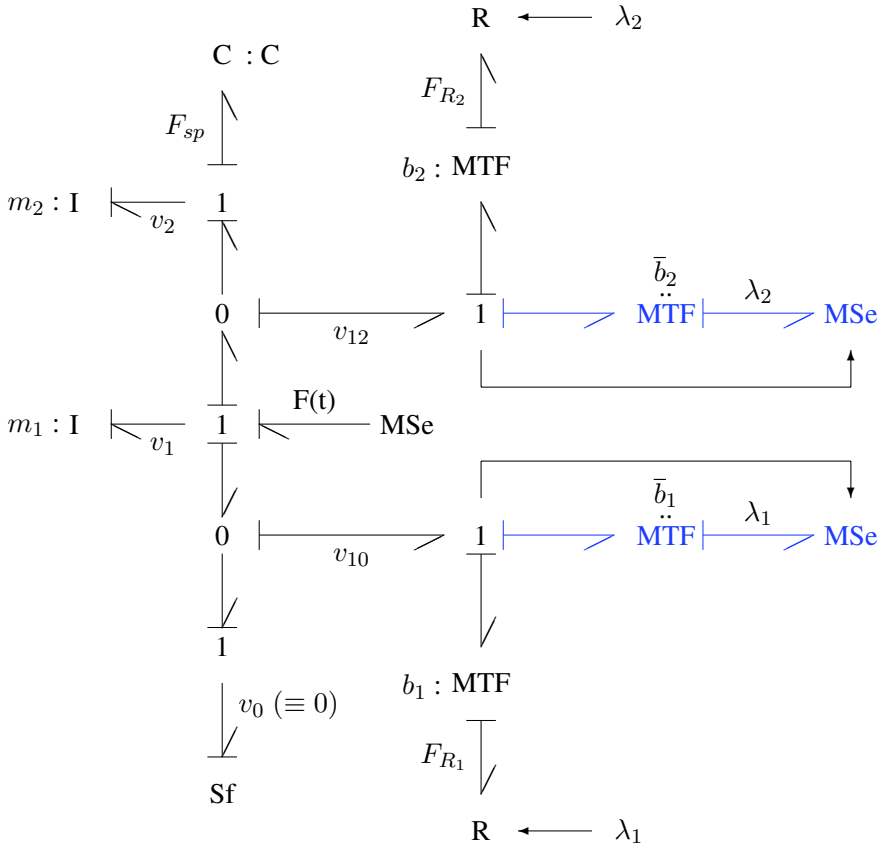


Fig. 7.25 Single bond graph representation of the stick-slip friction problem

to switch between slipping and sticking mode. As in the previous example (Figure 7.21), the two energy stores have fixed integral causality, although there is only one degree of freedom during stick condition.

From the bond graph in Figure 7.25, the following equations can be derived.

$$v_{10} = v_1 - v_0 \tag{7.17a}$$

$$v_{12} = v_1 - v_2 \tag{7.17b}$$

$$F_{R_1} = R_1(v_{10}) \tag{7.17c}$$

$$F_{R_2} = R_2(v_{12}) \tag{7.17d}$$

$$\dot{v}_1 = \frac{1}{m_1} [F(t) - b_1 \times F_{R_1} - \bar{b}_1 \times \lambda_1 - b_2 \times F_{R_2} - \bar{b}_2 \times \lambda_2] \tag{7.17e}$$

$$\dot{v}_2 = [b_2 \times F_{R_2} + \bar{b}_2 \times \lambda_2 - F_{sp}] \quad (7.17f)$$

$$\dot{F}_{sp} = \frac{1}{C} v_2 \quad (7.17g)$$

In these equations, the nonlinear function F_{R_2} is given by the equation

$$F_{R_2} = \begin{cases} F_{s_2} \operatorname{sign}(v_{12}) & \text{for } |v_{12}| > \varepsilon \\ F_{s_2} \operatorname{sign}(\lambda_2) & \text{for } |v_{12}| < \varepsilon \end{cases}, \quad (7.18)$$

where F_{s_2} denotes the value of the friction force for $v_{12} \neq 0$ (cf. Equation 7.24) (Approximations of real numbers in a computer are considered to be equal if the absolute value of their difference drops below a tolerance ε). A similar equation holds for the resistor R_1 . The values of the Boolean moduli b_i ($i = 1, 2$) are determined by a separate control logic associated with the bond graph. If the lower body is sticking, then $\bar{b}_1 = 1$ and thus $b_1 = 0$. Equally, $b_2 = 0$ if the upper body is sticking. The four possible combinations can be combined into the following velocity constraints

$$\bar{b}_1 \times (v_1 - v_0) = 0 \quad (7.19a)$$

$$\bar{b}_2 \times (v_1 - v_2) = 0. \quad (7.19b)$$

These two algebraic equations must be added to the equations of motions of both bodies. They determine the unknown constraint forces λ_1 and λ_2 . The resulting DAE system describing the motion of both bodies against dry friction is of differential index 2 (The approach corresponds with the modelling of the two sub-circuits connected by an ideal switch, cf. Figure 7.19).

For example, consider the special case that the lower body is sliding on the floor, ($\bar{b}_1 = 0, b_1 = 1$), while the upper body is sticking on the lower one, ($\bar{b}_2 = 1, b_2 = 0$), then the simplified DAE system

$$\dot{v}_1 = \frac{1}{m_1} [F(t) - F_{R_1} - \lambda_2] \quad (7.20a)$$

$$\dot{v}_2 = \frac{1}{m_2} [\lambda_2 - F_{sp}] \quad (7.20b)$$

$$\dot{F}_{sp} = \frac{1}{C} v_2 \quad (7.20c)$$

$$0 = v_1 - v_2 \quad (7.20d)$$

results. By differentiation of the velocity constraint, we obtain for the constraint force λ_2

$$\lambda_2 = \frac{m_2}{m_1 + m_2} [F(t) - F_{R_1}] + \frac{m_1}{m_1 + m_2} F_{sp}. \quad (7.21)$$

If both bodies are sliding, ($\bar{b}_1 = \bar{b}_2 = 0$), then the mathematical model reduces to two explicit ODEs.

Bond graph modelling of mechanical systems with inherent stick-slip friction has been subject of several publications [16, 23, 26].

Mechanical Systems with Multiple Rigid Bodies

The approach illustrated by means of the example of two bodies (Figure 7.23) can be applied to a system of N rigid bodies, of which some are connected by springs, while others stick together temporarily. Let \mathbf{v}_I be the vector of velocities of all bodies, \mathbf{F}_C the vector of all spring forces, \mathbf{u} the vector of all system inputs and $\boldsymbol{\lambda}$ the vector of all Lagrange multipliers. Then, the system is described by the equations of motions

$$\mathbf{M} \dot{\mathbf{v}}_I = \mathbf{f}_1(\mathbf{v}_I, \mathbf{F}_C, \mathbf{u}) + \mathbf{B}_1 \boldsymbol{\lambda} \quad (7.22a)$$

$$\dot{\mathbf{F}}_C = \mathbf{C}^{-1} \mathbf{f}_2(\mathbf{v}_I, \mathbf{F}_C, \mathbf{u}) + \mathbf{B}_2 \boldsymbol{\lambda} \quad (7.22b)$$

and some additional velocity constraints

$$\tilde{\mathbf{B}} \mathbf{v}_I = \mathbf{0} . \quad (7.23)$$

The velocity constraints are obtained from the bond graph by observing that the inputs into the sinks representing the constraint forces are zero. Obviously, differentiation of the velocity constraints yields a set of algebraic equations for the Lagrange multipliers.

$$\tilde{\mathbf{B}} \mathbf{M}^{-1} \mathbf{B}_1 \boldsymbol{\lambda} + \tilde{\mathbf{B}} \mathbf{M}^{-1} \mathbf{f}_1(\mathbf{v}_I, \mathbf{F}_C, \mathbf{u}) + \dot{\tilde{\mathbf{B}}} \mathbf{v}_I = \mathbf{0} \quad (7.24)$$

Let \mathbf{x} denote the vector of the displacements of all bodies and

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \quad (7.25)$$

the holonomic constraints. Then, the constraint forces \mathbf{Q} can be written in the form

$$\mathbf{Q} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \boldsymbol{\lambda} . \quad (7.26)$$

That is,

$$\mathbf{B}_1 = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \quad (7.27)$$

and

$$\underbrace{\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T}_{\tilde{\mathbf{B}}} \dot{\mathbf{x}} = \mathbf{0} . \quad (7.28)$$

Algorithm Controlling the Change Between Stick and Slip Mode

When the relative velocity between two bodies is zero, an algorithm associated with the bond graph checks whether a transition between the slip and the stick mode has occurred. Since the relative velocity is not exactly zero in stepwise numerical integration, the check must be performed when the values v_n and v_{n+1} at two suc-

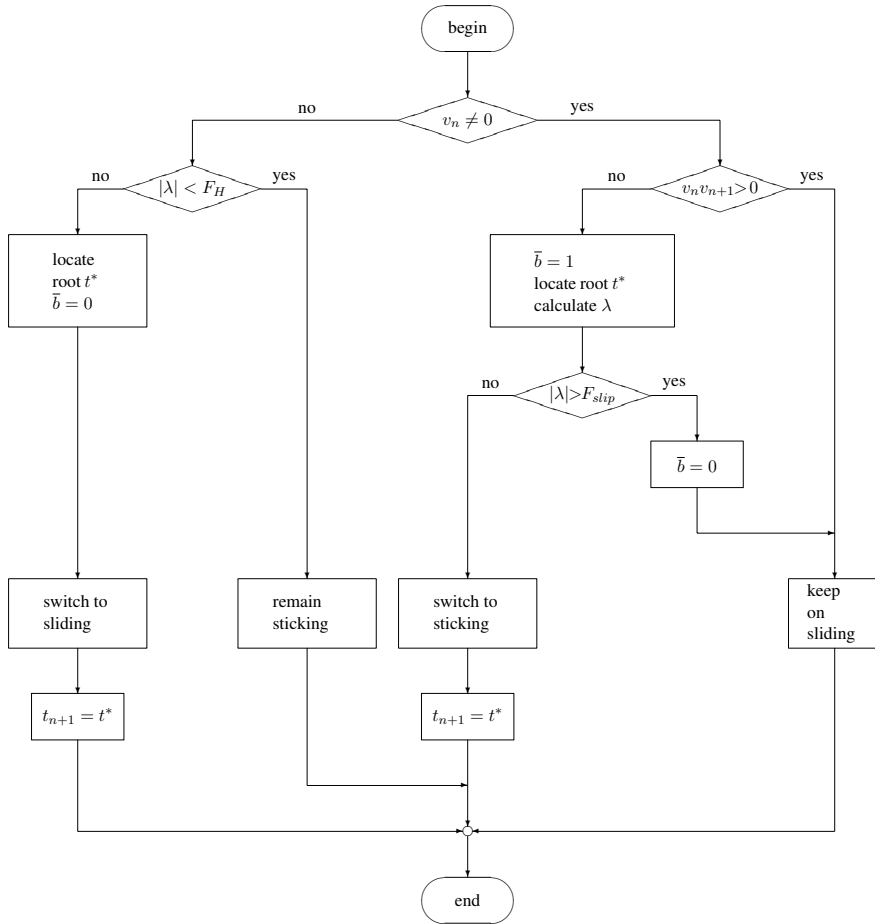


Fig. 7.26 Algorithm controlling a Boolean variable in the bond graph model of dry friction between a moving body and a non-moving surface

ceeding discrete times t_n and t_{n+1} differ with respect to their sign. The algorithm is given in Figure 7.26 for the simple case of a rigid body moving against dry friction on a surface that does not move.

The algorithm in Figure 7.26 can be easily formulated in a simulation language, e.g. ACSL [33]. The latter provides a so-called SCHEDULE statement that directs the simulator to interrupt the computation of the dynamic behaviour when the constraint formulated in the SCHEDULE statement is met to locate the time t^* of the event and to execute the statements in the DISCRETE section associated with the event. After this interruption, the simulator continues with the computation of the dynamic behaviour. Figure 7.27 shows a fragment of the ACSL formulation of the algorithm. In the DISCRETE sections, the transformer moduli are switched.

```

...
DERIVATIVE
...
  lambda = ...
...
  v12    =  v1 - v2
...
  SCHEDULE slip2stick .XZ. v12
  SCHEDULE stick2slip .XP. (ABS(lambda) - FH)
END ! of DERIVATIVE
!
DISCRETE slip2stick
  IF ( ABS(lambda) .LE. Fslip2) b2 = 0
END ! of DISCRETE slip2stick
!
DISCRETE stick2slip
  b2 = 1
END ! of DISCRETE stick2slip
...

```

Fig. 7.27 Fragment of the ACSL formulation of the algorithm controlling the switching between slip and stick mode

Alternatively, the program DASSLRT, a version of the well known DAE solver DASSL, can be used. It includes a root finder that can determine the discrete state event. After modification of the equations, numerical integration of the DAE system is restarted ([7], p. 135f).

For each pair of rigid bodies that can stick together, a separate algorithm controlling the corresponding Boolean transformer moduli must be formulated. Since more than two bodies may be linked by dry friction, the algorithms may depend on each other. If velocity constraints change, then the entries in the matrix $\bar{\mathbf{B}}$ (Equation 7.23) change and the constraint forces must be re-evaluated.

Summary of Features

Finally, the features of the modelling approach discussed in this section may be summarised as follows:

- A single bond graph of invariant structure and fixed causalities is developed that holds for all configurations of switch states.
- Similar to the approach of Ducreux et al. [18], parts of the bond graph are activated or switched off by changing the values of Boolean transformer moduli according to the system mode. The bond graph does not directly show which parts are latent, i. e., which parts do not experience any dynamic changes.
- Controlled sinks impose an effort or a flow. The value of their output meets the conditions of the state after the transition, e.g., an electrical switch is closed or

two bodies stick together. Constraints are reflected by the fact that the input into the controlled sinks is zero. At switching time, the sinks take free energy out of the system. However, they cannot be replaced by resistors since there is no relation between the input and the output.

- Since the inputs into the controlled sinks vanish, the mathematical model derived from the bond graph is a DAE system even if all energy stores show integral causality. The Boolean transformer moduli accounting for state transitions and the outputs of the sinks appear in the system equations. If all moduli are equal to zero, the mathematical model is a set of ODEs. If some moduli are different from zero, then the differentiation index of the DAE system is two (cf. Equation 7.12). In this regard, fixed integral causalities at energy stores may be somewhat misleading.
- The approach can be used to model ideal switches in electrical circuits, the transition between slip and stick mode, or the instantaneous conversion of kinetic energy into heat at stops in mechanical systems. The controlled sinks correspond to constraint forces in mechanical systems introduced by means of Lagrange multipliers.
- The value of a Boolean transformer modulus and thus the instantaneous discontinuous state transition is determined by an algorithm that is separately formulated from the bond graph.
- If the conditions for a discontinuous state transition are met, the time of the event must be determined, the output of the controlled sinks must be re-evaluated and the integration must be re-initialised.

7.2 Variable Causality Bond Graphs

7.2.1 *Ideal Switches as Another Basic Bond Graph Element*

Strömberg et al. prefer a different approach to bond graph representation of models that use the abstraction of instantaneous state transitions. Following usual electrical circuit schematics, they extend the set of basic bond graph elements by an ideal switch. They argue that switches like sources represent boundary conditions of a system and propose the mnemonic symbol S_w ([32], p. 119). In contrast to the standard bond graph elements, a switch cannot be assigned a time-invariant causality. If the switch is open, then it corresponds to a sink that imposes a vanishing flow. If the switch is closed, the flow sink becomes a sink that provides a vanishing effort (Figure 7.28). Before and after switching time, the energy flow is zero. At switching time, however, an ideal switch takes free energy out of the system as explained in Section 7.1.4). This aspect also suggests to represent an ideal switch by a sink in a bond graph.

The change of causality at switch elements due to the change of its state entails a change of causalities in parts of the bond graph. That is, a single assignment of fixed

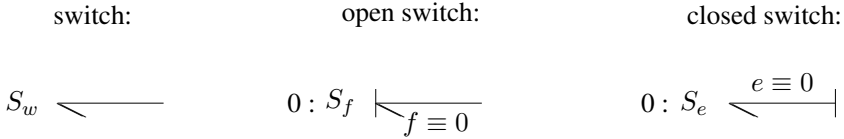


Fig. 7.28 Bond graph representation of ideal switches

causalities before the simulation is not sufficient. Adaption of causalities must rather be performed throughout computation of the dynamic system behaviour whenever a switch changes its state. As a result, if switches close, it may happen that some storage ports must change from preferred integral causality to derivative causality. That is, the dimension of the state vector is not constant, but can rather change, which may lead to Dirac pulses if energy stores instantaneously become dependent [10]. If the ideal switch is accepted as another basic bond graph element, then computational causality is no longer a time invariant attribute of power ports. Consequently, the previously considered bond graphs with time invariant causalities are to be replaced by variable causality bond graphs that can represent models of variable structure. Following this approach, the concept of computational causality loses some of its virtue. In [15], p. 55, Cellier takes the following view.

It has been shown that the concept of causality, as propagated throughout the bond graph literature, is an oversold concept that had its justification at a time when bond graphs were drawn by hand onto sheets of paper to be translated manually into state-space models before feeding them to an ODE solver. However, bond graph causality does not represent a physical property, and its questionable use is limited to analysing fixed structure models. The concept breaks down entirely when faced with variable structure models.

In order to keep the effect of a causality change at a switch port locally, Asher [1] proposes the use of a switch element in conjunction with a so-called ‘causality resistor’ which accounts for the causality change at the switch port such that the rest of the bond graph is not affected (Figure 7.29).

In [1], Asher denotes the ideal switch by the symbol T and calls the accompanying resistor, R, a ‘causality resistor’. Such a pair of an ideal switch and a resistor is no longer a model of an ideal switch. In the case of an electrical diode, the causal-

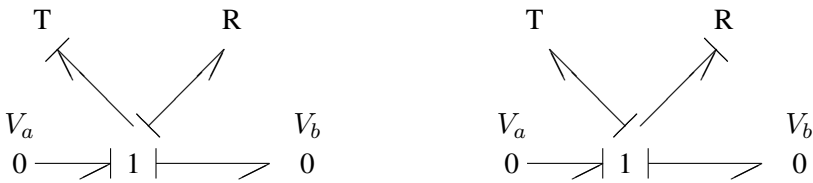


Fig. 7.29 Ideal switch with accompanying ‘causality resistor’ (Asher 1993)

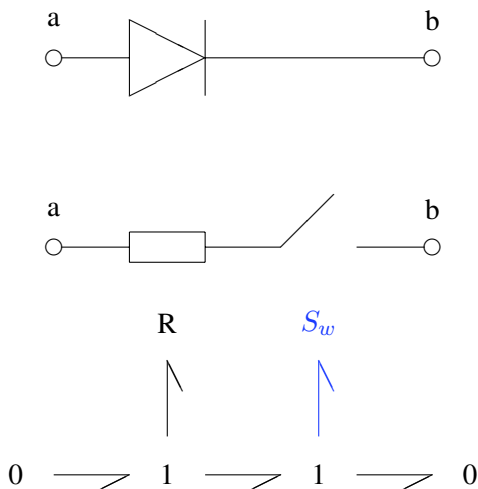


Fig. 7.30 Bond graph representation of an electrical diode by means of a bond graph switch element

ity resistor in Figure 7.29 can represent the ON-resistor (of small resistance). If the ideal switch and the causality resistor are used in a parallel connection, the causality resistor could represent the diode’s very high resistance in reverse mode. The parameter value of the causality resistor, in general, can be chosen freely within some reasonable limits as long as it has no significant effect on the dynamic behaviour. However, causality resistors can lead to fast transients, i.e., to stiff model equations. As indicated in Figure 7.29, causalities in the rest of the bond graph, especially integral causalities at storage ports, remain unaffected.

Diodes in electrical circuits can be modelled by a resistor representing the ON-resistance and an ideal switch connected in series. If this series connection is represented in a bond graph by means of the newly introduced switch element (Figure 7.30), then the latter corresponds to a real device.

In bond graphs of mechanical systems, this correspondence between the switch element and a real device is not always convincing. If, for instance, a body of mass m is *not* in contact with a stop, then there is no link between the body and the stop. Strömberg et al. talk about an ‘invisible component’ ([32], p. 118). An electrical diode is always a real connection between two nodes even if it is in reverse mode.

In Figure 7.31, the symbol Ψ besides the switch element denotes the switching algorithm associated with this element. The latter can be represented by a simple Petri net with two states and two state transitions as shown in Figure 7.32. In the Petri net, x_{12} denotes the distance between the body and the stop ($\dot{x}_{12} := v_1 - v_2$).

Since the causality at the switch port depends on the switch state, causalities cannot be completed in the bond graph of Figure 7.31. Depending on the switch state, the switch can be represented either by a flow or an effort source. Thus, for each

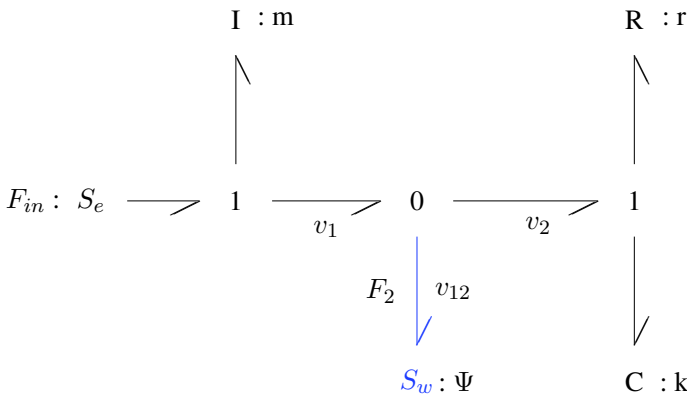
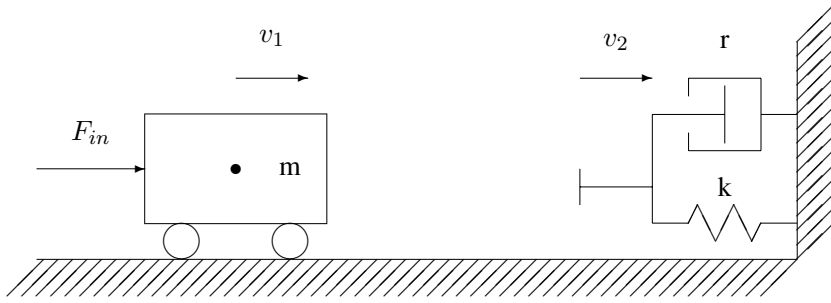


Fig. 7.31 Bond graph modelling of a mechanical stop using a switch element (Strömberg, Top, Söderman 1993)

switch state, a causally complete bond graph can be given from which equations can be derived that hold for this state (see Figure 7.33).

For n switches, there are 2^n possible combinations of switch states and as many system modes. In practice, however, not all are physically feasible. If during the simulation we do not want to switch between different models of the same system depending on switch state combinations, then a possible alternative is to characterise the state of a switch by a variable that takes values either zero or one depending on the switch state in order to assign causalities in a bond graph as far as possible and to derive a DAE system in which switch variables b_i , ($i = 1, \dots, n$) occur. Let $b = 1$ characterise an open switch and let $\bar{b} := 1 - b$, then the following equations can be derived from the bond graph in Figure 7.34.

$$v_{12} = v_1 - v_2 \tag{7.29a}$$

$$0 = b \times F_2 + \bar{b} \times v_{12} \tag{7.29b}$$

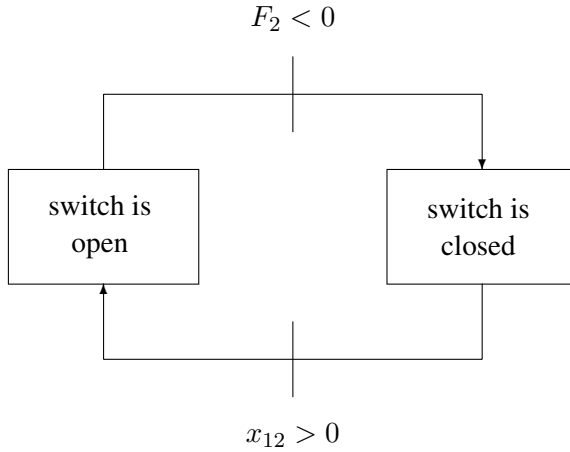


Fig. 7.32 Control algorithm of the switch

$$0 = F_2 - F_R - F_{sp} \quad (7.29c)$$

$$0 = F_R - r v_2 \quad (7.29d)$$

$$\dot{v}_1 = \frac{1}{m} [F_{in} - F_2] \quad (7.29e)$$

$$\dot{F}_{sp} = k v_2 . \quad (7.29f)$$

If we reduce the algebraic equations to a set of equations for the unknowns F_2 and v_2 , the determinant of the latter is always different from zero independent of the switch state. If the reduced set of algebraic systems is symbolically solved, then there is the following unique solution for both switch states.

$$F_2 = \frac{\bar{b}}{-br + \bar{b}} (r \times v_1 + F_{sp}) \quad (7.30a)$$

$$v_2 = \frac{1}{-br + \bar{b}} (b \times F_{sp} + \bar{b} \times v_1) \quad (7.30b)$$

This, however, is not always possible. In order to show this, consider the simple bond graph in Figure 7.35. Firstly, we see that the I energy store cannot have preferred integral causality since this would entail an invariant effort causality at the switch element. Integral causality at the storage port is only possible when the switch is closed. In that case, the switch can be represented by an effort source.

From the bond graph of Figure 7.35, the following equations are derived.

$$0 = \bar{b} \times u_s + b \times i \quad (7.31a)$$

$$u_e = R i + u_L + u_S \quad (7.31b)$$

$$u_L = L \frac{di}{dt} \quad (7.31c)$$

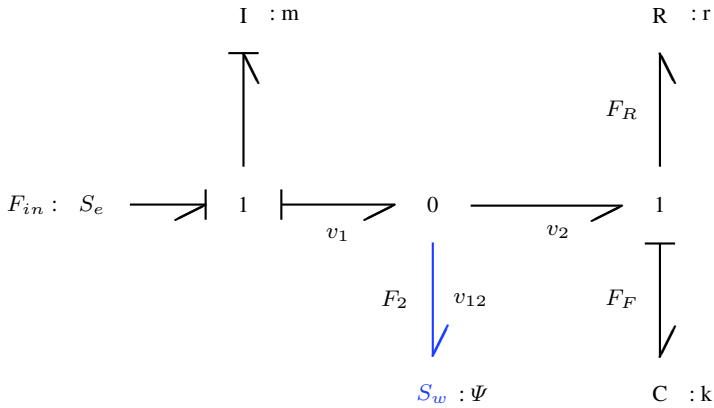


Fig. 7.34 Causally incomplete bond graph of the example in Figure 7.31

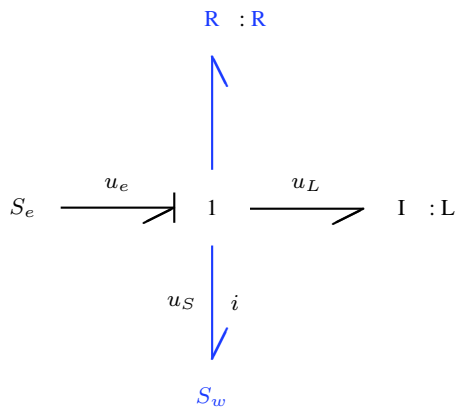


Fig. 7.35 Bond graph of a series connection of a diode and an inductance

7.2.2 Controlled Junctions – Hybrid Bond Graphs

If variable causalities are accepted, then another approach to a bond graph representation of models of variable structure is not to use an ideal switch element, but to link 0- and 1-junctions with a control algorithm. The latter defines under which conditions a junction with all its adjacent bonds is switched off or is re-activated [28]. Mosterman and also other authors such as Biswas [3, 30] call such bond graphs *Hybrid Bond Graphs* (HBGs).

In a similar way, Thoma [35] used the concept of *time dependent junctions* allowing for switching connections between power ports off and on in 1974. In [35] (p. 107), he denotes such junctions by the symbol *tdj*.

Mosterman shares the author's view that switches are not energy processing elements. Therefore, it is not obvious to introduce a switch as another bond graph element. Switches represent control rather than physical aspects ([28], p. 53). The control algorithm linked with a controlled junction is represented *separately* from the bond graph as a finite state machine (FSM). If a 1-junction is switched off, then the flow vanishes at all adjacent bonds. Consequently, the energy flow disappears at all bonds and a disconnection of energetic links between model parts occurs. Since a 1-junction in a Kirchhoff junction structure is connected to a 0-junction, in general, and since the latter does have more than two adjacent bonds, a 1-junction that has been switched off can be viewed as removed from the bond graph. Moreover, the flow causality of a 1-junction that has been switched off is not propagated beyond adjacent 0-junctions. Similar statements hold for 0-junctions. However, if a controlled junction changes its switch state, then this can affect the causalities at element ports via a causal path from the controlled junction. As a result, this modelling approach also relies on variable causalities.

Hybrid Bond Graph for the Bouncing Ball Problem

Using a controlled 0-junction, the well known bouncing ball problem can be represented by a hybrid bond graph and an associated simple finite state graph as depicted in Figure 7.36. In Figure 7.36, the R element accounts for friction in the air; the subscript of the 0-junction identifies its associated finite state automaton. The latter controls the junction's behaviour. As the finite state automaton part of the models shows, the controlled junctions turns on and connects the lower zero value flow source when the ball touches the floor. That is, the ball velocity becomes zero and the I element must take derivative causality. When the force F_b becomes negative, the 0-junction turns off, disconnecting the source from the 1-junction. As a result, the I element switches back to integral causality. The ball's momentum reverses and the ball goes up into the air.

According to the two states, OFF (ball in the air) and ON (ball in contact with the floor), the two bond graphs shown in Figure 7.37 can be obtained from Figure 7.36.

When the ball is in the air, then the force $F_b = 0$ and the 0-junction is not active. In this case, it can be replaced by an effort source (with two ports) that expresses this boundary condition. The subscript besides the controlled 0-junction points to the associated finite state machine. When the ball is in contact with the floor, then the 0-junction is active. It forwards the zero velocity and the causality of the flow source. As a result, the I energy store takes derivative causality. This switch state remains activated until the force the ball imposes on the floor, changes its sign.

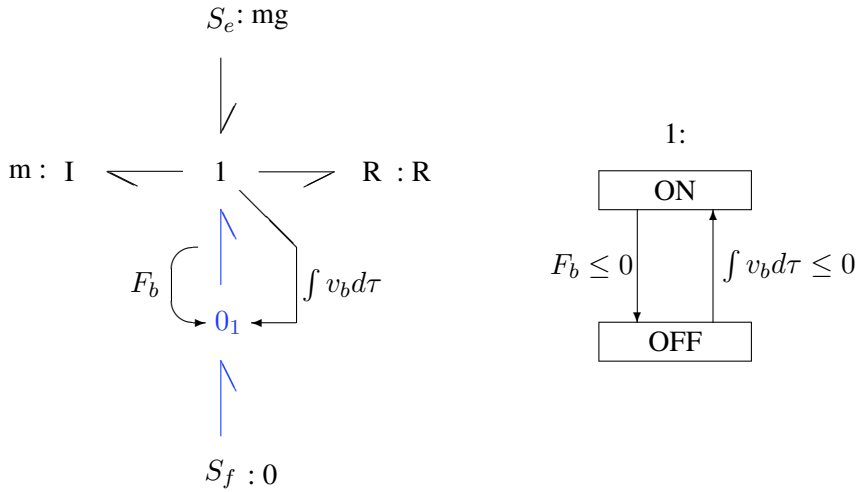


Fig. 7.36 Hybrid bond graph with a controlled 0-junction representing the bouncing ball problem (Mosterman, 1997)

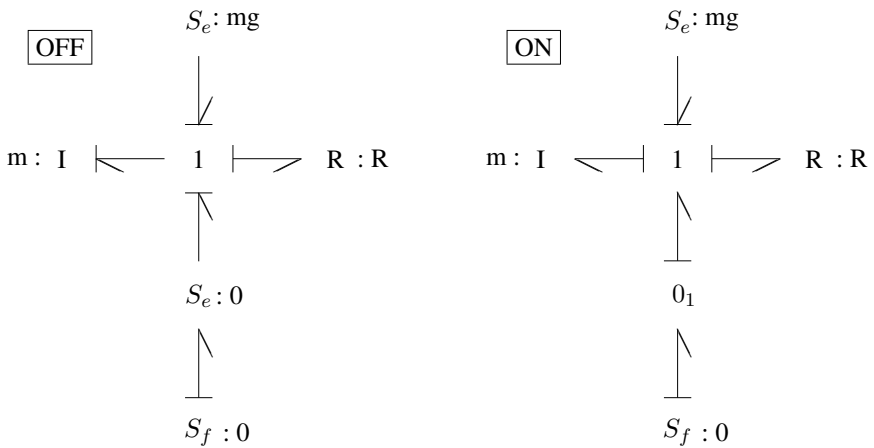


Fig. 7.37 Bond graphs for the two states OFF (no contact with the floor), ON (ball in contact with the floor)

Software for Hybrid Bond Graphs

The concept of junctions controlled by finite state machines and consequently the approach of changing causalities and of re-initialisations of the model after switching *at run-time* has been implemented in the modelling and simulation software HYBrSIM [27]. This software has been especially developed for the support of hybrid systems and a continuous-discrete event bond graph modelling approach. The graphical representation of bond graphs in HYBrSIM has been inspired by the *Java applet for bond graph modelling* developed by R. Bajzát at the University of Miskolc, Hungary [2].

In [3, 30], the authors present an implementation of the hybrid bond graph paradigm that builds on block diagrams. In this approach, a hybrid bond graph model is converted into a structurally adaptable block diagram constituting the computational model. Initially and whenever a mode change happens due to the switching of controlled junctions, causality is assigned and re-assigned to the hybrid bond graph such that each junction has got a strong causal determination. When a controlled junction changes its state, the block diagram is modified and data flow paths through the computational model are updated accordingly at run-time. All energy storage elements of the hybrid bond graph are assumed to have integral causality. They are implemented by means of standard Simulink[®] blocks as well as those elements with fixed causality while the switching of junction states and their implications on the computational structure are handled by S-functions [34] implementing elements with variable causality (junctions, resistors, transformers and gyrators). After a reassignment of causality to the hybrid bond graph, S-functions use the new determining bonds at junctions to deliver their outputs accordingly, allowing signal flows to be rerouted dynamically through a structurally static Simulink[®] model. The advantage of this approach is that it seamlessly integrates with widely used (commercial) simulation software.

7.3 A Combined Petri Net – Bond Graph Representation of Variable Structure Models

The approaches to a bond graph representation of models of variable structure considered in the previous section are similar in the sense that instantaneous discontinuous state transitions are modelled in the bond graph itself either by switch elements, controlled junctions or by transformers modulated by a Boolean variable. The algorithms controlling instantaneous state transitions are given separately from the bond graph for each switch or controlled junction. Switching conditions are formulated only for local variables. The advantage is that local control algorithms can be kept easy and clear. Moreover, modular models can be developed for large systems.

One can argue that the approach of modelling instantaneous state transitions by bond graph elements linked with a control algorithm contradicts the concept of bond graph based physical systems modelling. If controlled junctions, for instance, are

used, then the junction structure loses its fundamental characteristic of conserving power. Moreover, by using local control algorithms, it is not clear which states and which state transitions exist in a system. If in a system with several switches one of them changes its state, it is not apparent from the associated control algorithm which other switches are affected and what system state will result. Such considerations lead to the following approach [6].

- Discrete system states and transitions between them are represented in a global Petri net for the overall system.
- For each system mode captured in the Petri net, the dynamic system behaviour is described by one or, if necessary, several disjunct bond graphs using only standard bond graph elements.

This view immediately gives rise to a number of remarks.

- A standard Petri net description of a system does not take into account whether a state transition takes place continuously with time or instantaneously.
- The switching between states is not expressed in the bond graph at all. The virtue of bond graphs is the description of continuous processes.
- There is not a single bond graph of a system, but a given number of them. Each bond graph model only holds for the time between two discrete events, that is, as long as the system is in a certain mode. In principle, for each mode, the structure of the model can be different. Thus, variable structure systems can be modelled.
- The mathematical model is not one single DAE system for all combinations of switch states in which Boolean moduli appear in some coefficients of the equations, but a set of initial value problems (IVPs).
- Finally, it is necessary to determine the time of an instantaneous state transition before switching to another initial value problem.

Example: Two Bodies Moving Against Stick-Slip Friction

Before adding further considerations of this approach, first, it shall be illustrated by applying it to the problem of two bodies moving against Coulomb friction (Figure 7.25) already considered in Section 7.1.4. In the following, however, the spring attached to the upper body is removed so that the latter can fall down from the lower body (Figure 7.38).

The first task in this modelling approach is to set up a global Petri net. To that end, the system states listed in Table 7.1 can be identified. The lower body can either slip or stick on the non-moving surface. Similarly, the upper body can slip or stick on the lower one. Furthermore, it can fall down from the lower body. In that case, we will assume that it starts falling down having a horizontal velocity component. Finally, after falling down from the lower body, the upper body rests permanently on the floor, while the lower body still can slide or stick. By considering what is physically feasible under these conditions, we get the Petri net shown in Figure 7.39.

In this Petri net, an attribute c_{ij} annotated to a transition symbol ($\rightarrow \blacksquare \rightarrow$) denotes the condition under which the transition fires. Some state transitions can only take place in one direction, while others are reversible. For instance, if the body of

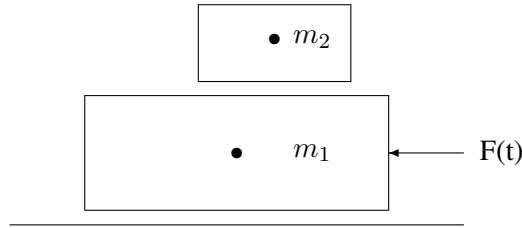


Fig. 7.38 Two rigid bodies moving against stick-slip friction

Table 7.1 Discrete system states

No.	System state
1	: m_1 sticking, m_2 sticking
2	: m_1 sliding, m_2 sticking on m_1
3	: m_1 sliding, m_2 sliding
4	: m_1 sticking, m_2 sliding
5	: m_1 sticking, m_2 falling
6	: m_1 sliding, m_2 falling
7	: m_1 sliding, m_2 resting on the floor
8	: m_1 sticking, m_2 resting on the floor

mass m_2 is no longer in contact with the lower one, then clearly, only the lower one can switch from sliding to sticking and vice versa. Hence, transitions, e.g., t_{36} from state 3 to state 6 are not reversible. On the other hand, if, for instance, both bodies are sliding (state 3) and if the value of the external force $F(t)$ drops below the breakaway level F_{H10} of the Coulomb force between the lower body and the floor, then the lower body will stick if its velocity is zero at that moment, whereas the upper body can still continue sliding (state 4). Conversely, if the value of the external force, $F(t)$, again exceeds the value F_{H10} , while the upper body is still sliding, the system returns to its previous state (both bodies are sliding). Thus, two opposite transitions t_{34} and t_{43} between system states 3 and 4 can happen. Conditions c_{ij} can be formulated straightforward as Boolean expressions. Transition t_{34} , e.g., is governed by the condition c_{34}

$$(v_1 = 0) \wedge (|F_1| < F_{H10}),$$

in which F_1 is the net force acting on m_1 . Condition c_{43} for the reverse transition t_{43} reads

$$|F_1| > F_{H10}.$$

After having identified all discrete system states and having captured physically feasible state transitions in a global Petri net, bond graph models have to be developed for each system mode in the second phase of this modelling approach. The Figures 7.40, 7.41, and 7.42 show bond graphs for the system modes 2,3,4,5 and 6.

In contrast to the approach that uses controlled sinks with invariant causality switching off and on degrees of freedom (Section 7.1.4), bond graphs obtained by

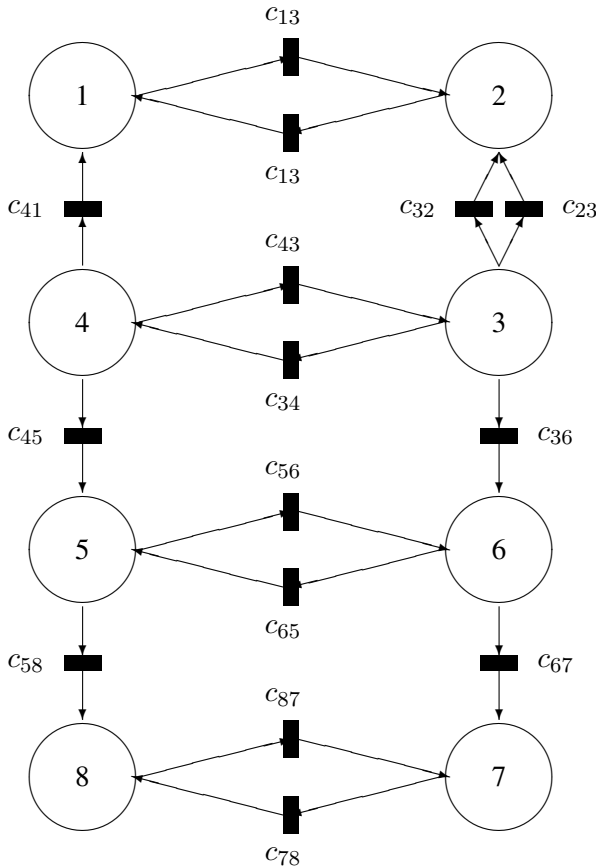


Fig. 7.39 Petri net indicating system states and possible transitions between them

this approach can be simplified. This has been done for the bond graph corresponding to system mode 2.

Since the bond graphs corresponding to the system modes represented in the Petri net only use standard bond graph elements, they can be entered into a bond graph processing program, e.g. CAMP-G [14], in order to automatically generate the model equations in a simulation language, e.g. ACSL. For small models, they may be derived even manually. For each system mode depicted in the Petri net, the dynamic equations derived from the corresponding bond graph go into the DERIVATIVE section of an ACSL model description. Computation of these models depend on conditions. The switching between models is controlled by the conditions for firing of the state transitions in the Petri net. These conditions are formulated in the DERIVATIVE section and are used in SCHEDULE statements, allowing for the determination of the time of a switching event. The proper initialisation of the model

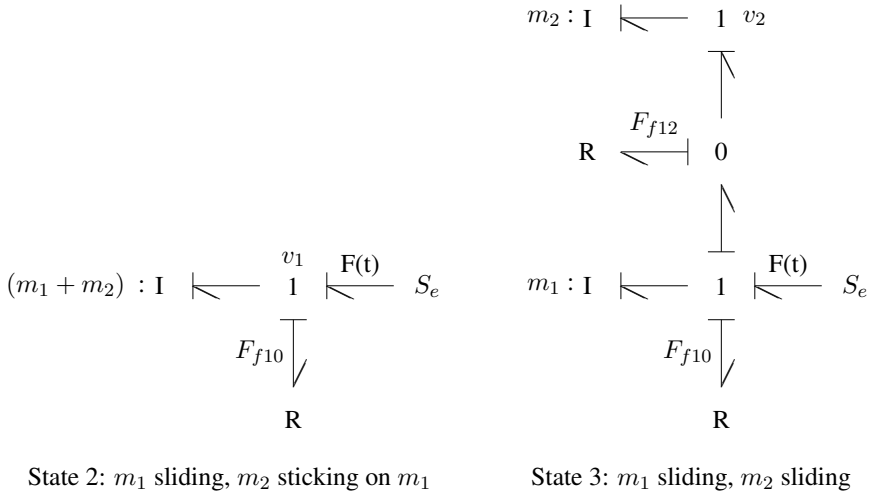


Fig. 7.40 Bond graphs holding for system modes 2 and 3

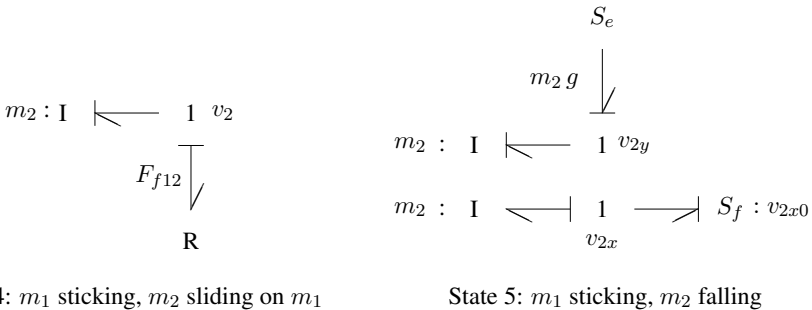


Fig. 7.41 Bond graphs holding for system modes 4 and 5

to be used after the switching event is performed in a DISCRETE section linked with a SCHEDULE statement via a name.

Clearly, the description of a variable structure model by means of a global Petri net in combination with a certain number of bond graphs is not bound to the use of a particular software program. A formulation in ACSL is just one option. Since the mathematical model is not a single DAE system in which Boolean switching variables appear, but a set of initial value problems, DAE solvers with an incorporated root finder, e.g. LSODAR [21] or DASSLRT ([7], p. 135) can be used as well. These solvers determine the point in time at which switching happens and reset numerical integration. For models with frequent discontinuities, the efficient implementation of an implicit Runge-Kutta method, e.g. in the STRIDE code [13] may be an alter-

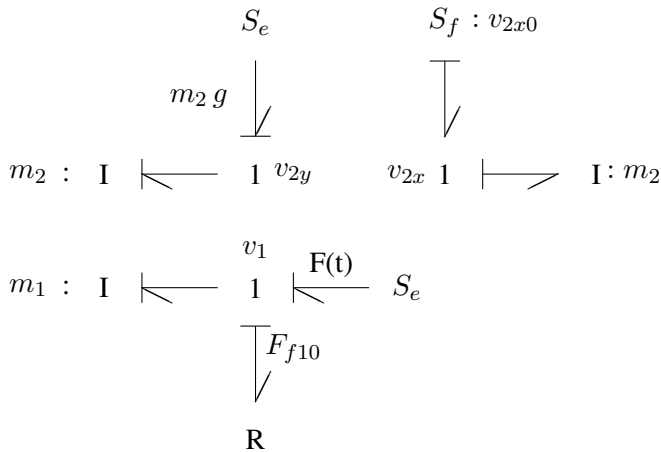


Fig. 7.42 Bond graph holding for system mode 6: m_1 sliding, m_2 falling

native to the often used BDF method implemented, e.g. in DASSLRT. In contrast to linear multi-step methods, Runge-Kutta methods can restart with high order after a discontinuity has been located.

Pros and Cons of the Combined Petri Net - Bond Graph Approach

The graphical representation of models including discontinuous state changes presented in this section has the following advantages.

- Consideration of discrete system modes and transitions between them is clearly separated from modelling the dynamic behaviour in a certain system mode. Switching between system modes is not represented at bond graph level. There is no need to extend the set of basic bond graph elements by an ideal switch element or by controlled junctions.
- Systematic model development takes place in two steps. In each modelling phase, a suitable standard graphical model representation formalism is used without any modification. Each graphical formalism is supported by software.
- The Petri net to be developed does not represent local switch states, but physically feasible global system modes and transitions between them.
- Since bond graph models representing different system modes may differ with regard to their structure, changes in the model structure can be taken into account in this way.
- In contrast to variable causality bond graphs, there is no need for re-assigning causalities after each discontinuous state transition. Since there is a finite number of system modes, causalities can be assigned to each bond graph model corresponding to a system mode once prior to simulation.

These advantages are opposed by the disadvantage that the number of system modes with the number of switches or discontinuities rapidly increases, making the development of a global Petri net difficult and error-prone. This aspect is of importance because in the first modelling phase, the identification of physically feasible system modes and of transitions between them is hardly supported by any software. In general, the latter comes into play when an existing Petri net is to be entered by means of a graphical editor. It is true that all 2^n possible combinations of switch states can be automatically determined, but those physically impossible cannot be automatically disregarded. This is a general problem in the development of large discrete event models. If system modes and discontinuous transitions between them are implicitly taken into account by Boolean switching variables appearing in the dynamic system equations and by separate local control algorithms, then the modeller does not have the problem of identifying all global system modes and all transitions between them.

7.4 Conclusion

The topic of this chapter has been the question as to how the abstraction of instantaneous state transitions (useful from the point of view of a macroscopic time scale) can be included in the framework of bond graph-based physical systems modelling. Transitions from one energetic system mode to another are caused by the exchange of energy between system components which takes place during a nonzero time period. Therefore, bond graph modelling based on conservation laws from physics, would require a microscopic time scale. The energy exchange is linked with the exchange of physical quantities, e.g., mechanical momentum or electrical charge. Their values as well as the value of power cannot change discontinuously. In reality, switching pulses always have a nonzero width. Nevertheless, in practice, the abstraction of discontinuous state transitions is indispensable.

Throughout the past decades, a number of solutions have been proposed in the literature to include this abstraction in a bond graph framework. Some of them have been discussed in this chapter. All of them have their advantages and disadvantages. Some approaches appear to be more suitable than others with regard to specific applications. As a result, in the author's view, so far there still is no unique answer to the question.

Three different approaches have been considered in this chapter.

- One option is to use a single bond graph with causalities invariant for all system modes.
- Another possibility is to relax the concept of computational causalities and to accept that causalities must be dynamically re-assigned after each discontinuous state transition.
- The third approach uses Petri nets for representing system modes and transitions between them and bond graphs for the description of the dynamic system behaviour in each system mode.

Regarding the approaches using bond graphs with invariant causalities, one can distinguish between those that explicitly model discontinuous state transitions by means of transformers controlled by Boolean variables (Section 7.1.2) and those that implicitly account for them by extending element characteristics (Section 7.1.1). The latter approach is an option if only simple simulation programs are available. It neither requires one to extend the set of basic bond graph elements nor to restart the integration algorithm. This approach is of rather historic relevance.

Transformers controlled by Boolean variables have the advantage that discontinuous state transitions can be accounted for in bond graphs without introducing any new bond graph elements. Bond graphs have static, time invariant causalities. Thus, equations can be derived in the usual manner. Boolean switching variables appear in the coefficients of the model equations. The model developer does not need to find out which physical feasible system states and which transitions exist between them. Ducreux et al. [18] combine the transformer controlled by a Boolean modulus with a resistor to account for the ON-resistance of real switching elements, e.g., electrical diodes, or hydraulic check valves. If the resistors accompanying the transformers are linear (as all other elements in the bond graph), then transfer functions can be derived from the bond graph for controller design purposes. If transformers controlled by a Boolean modulus are combined with a controlled sink (Section 7.1.4), then stick-slip effects or the impact of a body on a stop can be described as discontinuous events. The sink activated by the modulus of the transformer imposes an effort or a flow that meets the conditions of the mode immediately after the switching. In mechanical systems, these sinks can represent constraint forces introduced by means of Lagrange multipliers. If they are active, they switch off degrees of freedom as long as certain constraints are fulfilled.

The representation of ideal switches by transformers controlled by a Boolean switch variable has the disadvantage that it is not evident from the bond graph which of its parts are latent. To see this effect, the values of the switch variables must be known. Their value is determined in control algorithms that are represented separately from the bond graph, either in the form of a flow diagram [18], as a Petri net, or a finite state machine. If controlled sinks are used for switching off degrees of freedom, then energy stores keep integral causality even when they have become dependent due to the switching (Causalities once assigned remain invariant). For models with ideal switches, the differential index of the corresponding DAE system is not constant because the Boolean switch variables appear in the coefficients of the equations. In the example of Figure 7.21, the differential index of the corresponding DAE system is zero if the switch is open and two when it is closed.

Extending the set of basic bond graph elements by an ideal switch implies that causalities can no longer be invariant attributes of power ports. Whenever a switch changes its state during simulation, causalities must be re-assigned. Moreover, state variables must be re-initialised and the integration algorithm must be restarted. That is, the use of an ideal switch poses significant demands on the simulation program to be used. In electrical or hydraulic circuits, a switch as a bond graph element directly corresponds with fast switching devices, e.g. diodes, or hydraulic check valves. Even if these devices are open, they are a real link between components. In

mechanical systems, it is not always appropriate to model structural changes by an ideal switch. If a body is not in contact with a stop, then there is no link between the body and the stop.

Acausal component models with switches can be composed to a hybrid model of the entire system according to the system's structure. In the course of the simulation, parts of the bond graph decompose, while others are reconnected depending on which switches open or close.

In [36], Top takes the view that reassignment of causalities and the derivation of equations from the bond graph after a change of switch states can be performed automatically during simulation. This would mean that the model developer does not need to identify all physically feasible combinations of switch states and to develop a bond graph model for each combination of switch states. On the other hand, the interruption of the simulation, the reformulation, the compilation and the initialisation of new model equations cause a significant increase in the simulation time. Reformulation of equations and re-compilation can be avoided if the software implementation of an element or a submodel accounts for all possible causality patterns at its ports. As switching may entail a change from integral to derivative causality at an energy storage port and because numerical differentiation is unwanted, this means that the mathematical model must be formulated as a DAE system in any case.

Clearly, the more switches exist in a model, the more significant the increase in simulation time due to dynamic adaption of the model will be.

Instead of the ideal switch element introduced by Strömberg, Top and Söderman [32], Mosterman [28] uses controlled junctions. At the advent of an event specified by the control algorithm of a junction, all of its adjacent bonds are switched off. As a result, submodels are disconnected. That is, there is no energy flow and no propagation of causality information. In contrast, transformers modulated by a Boolean switch variable even propagate a causality when the modulus is zero. The use of controlled junctions has the disadvantage that the junction structure of a bond graph no longer complies with the principle of power conservation.

Furthermore, Umarikar and Umanand introduced switched power junctions that have been considered in Section 7.1.3.

Finally, Borutzky, Broenink and Wijbrans proposed in [6] to separate the consideration of discrete system states and transitions between them on the one hand and the modelling of the dynamic behaviour in a system mode on the other hand, and to use different standard graphical model representation formalisms without modifications for both tasks. Consequently, for each formalism, available software can be used. The consideration of a system performing in one system mode can lead to a simpler bond graph because system parts may be latent. For instance, two bodies sticking together temporarily may be combined into one body. Since bond graphs for different system modes may be different, variable structure models can be described this way. On the other hand, since the number of system modes and transitions between them rapidly increases with the number of switches, this approach is more suited for models with a small number of switches, in practice.

References

- [1] G.M. Asher. The Robust Modelling of Variable Topology Circuits Using Bond Graphs. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 126–131. SCS Publishing, January 17–20 1993. Simulation Series, volume 25, no. 2, ISBN: 1-56555-019-6.
- [2] R. Bajzát. BOND GRAPH – Java Applet and Application for Bond Graphs. URL <http://gold.uni-miskolc.hu/~iitbajzi/bond/>.
- [3] C.D. Beers, E.J. Manders, G. Biswas, and P.J. Mosterman. Building efficient simulation from hybrid bond graph models. In *Proc. of the 2nd IFAC Conf. on Analysis and Design of Hybrid Systems*, pages 71–76, Alghero, Italy, 2006.
- [4] W. Borutzky. Representing discontinuities by means of sinks of fixed causality. In F. E. Cellier and J. J. Granda, editors, *1995 International Conference on Bond Graph Modeling, ICBGM'95, Proc. of the 1995 Western Simulation Multiconference*, pages 65–72. SCS Publishing, January 15–18 1995. Simulation Series, Vol. 27, Number 1, ISBN: 1-56555-037-4.
- [5] W. Borutzky. Discontinuities in a Bond Graph Framework. *Journal of the Franklin Institute*, 332B(2):141–154, 1995.
- [6] W. Borutzky, J.F. Broenink, and K.C.J. Wijbrans. Graphical description of physical systems containing discontinuities. In A. Pavé, editor, *Modelling and Simulation 1993, Proc. of the 1993 European Simulation Multiconference*, pages 208–214. SCS Publishing, June 7–9 1993. Lyon, France.
- [7] K.E. Brennan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.
- [8] J.F. Broenink and K.C. Wijbrans. Describing Discontinuities in Bond Graphs. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 120–125. SCS Publishing, January 17–20 1993. Simulation Series, Vol. 25, No. 2, ISBN: 1-56555-019-6.
- [9] J. Buisson. Modélisation, Analyse et Commande des Systèmes en Commutation. Université de Rennes 1, Rennes, France, 2002. l'habilitation à diriger des recherches (HDR).
- [10] J. Buisson. Analysis and characterisation of hybrid systems with bond-graphs. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 264–269. SCS Publishing, January 17–20 1993. Simulation Series, volume 25, no. 2, ISBN: 1-56555-019-6.
- [11] J. Buisson, H. Cormerais, and P-Y. Richard. Analysis of the bond graph model of hybrid physical systems with ideal switches. *Proc. Instn Mech. Engrs, Part I, Journal of Systems and Control Engineering*, 216(1):47–63, 2002.
- [12] J. Buisson, H. Cormerais, and P-Y. Richard. Analysis of the bond graph model of hybrid physical systems with ideal switches. *Proc. of the Institution of Mechanical Engineers Part I: Systems and Control Engineering*, 216(1):47–63, 2002.
- [13] K. Burrage, J.C. Butcher, and F.H. Chipman. An implementation of singly-implicit Runge-Kutta methods. *BIT*, 20:326–340, 1980.
- [14] *CAMP-G – User's Manual*. Cadsim Engineering, P. O. Box 4083, Davis, Ca 95617. URL <http://www.bondgraph.com>.
- [15] F.E. Cellier, M. Otter, and H. Elmqvist. Bond graph modeling of variable structure systems. In F.E. Cellier and J.J. Granda, editors, *ICBGM'95, International Conference on Bond Graph Modeling and Simulation*, pages 49–55. SCS Publishing, 1995. Simulation Series, volume 27, no. 1.
- [16] G. Chan and D.C. Karnopp. Bond Graph Modeling of Stick-Slip Behavior in Friction Belt Drives. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation*, volume 37 (1), pages 161–166, 2005.
- [17] G. Dauphin-Tanguy and C. Rombaut. Why a unique causality in the elementary commutation cell bond graph model of a power electronics converter. In *1993 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 257–263, 1993.

- [18] J.P. Ducreux, G. Dauphin-Tanguy, and C. Rombaut. Bond Graph Modelling of Commutation Phenomena in Power Electronic Circuits. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 132–136. SCS Publishing, January 17-20 1993. Simulation Series, volume 25, no. 2, ISBN: 1-56555-019-6.
- [19] K. Edström. *Switched Bond Graphs: Simulation and Analysis*. PhD thesis, Linköping University, Linköping, Sweden, 1999.
- [20] J. Garcia-Gomez. *Approche bond graph pour la modélisation des effets thermiques dans les composants de commutation en lectronique de puissance*. PhD thesis, Université des Sciences et Technologies de Lille, Lille, France, 1997.
- [21] A.C. Hindmarsh. ODEPACK A Systemized Collection of ODE Solvers. Preprint UCRL-88007, Lawrence Livermore National Laboratory, August 1982.
- [22] S. Junco, G. Diguez, and F. Ramirez. On Commutation Modeling in Bond Graphs. In J. Granda and F. Cellier, editors, *Proc. of the 2007 International Conference on Bond Graph Modeling ICBM 2007*, volume 39(1) of *Simulation Series*, pages 12–19. SCS, 2007.
- [23] D.C. Karnopp. Computer Simulation of Stick-Slip Friction in Mechanical Sytems. *Trans. ASME Journal of Dynamic Systems, Measurement and Control*, 107 (1):100–103, 1985.
- [24] D. Kölsch and G.P. Ostermeyer. Coulombsche Reibung in der Fahrzeugsimulation. *Automobil-Industrie*, 5:385–388, 1992.
- [25] F. Lorenz. Discontinuities in bond graphs: What is required? In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 137–142. SCS Publishing, January 17-20 1993. Simulation Series, volume 25, no. 2, ISBN: 1-56555-019-6.
- [26] D.L. Margolis. Bond Graph Model of a Multi-Plate Clutch in a Vehicle System Using Fixed Causality Slip-Stick Friction. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation*, volume 37 (1), pages 167–172, 2005.
- [27] P.J. Mosterman. HYBrSIM – a modelling and simulation environment for hybrid bond graphs. *Proc. Instn Mech. Engrs, Part I, Journal of Systems and Control Engineering*, 216 (1):35–46, 2002.
- [28] P.J. Mosterman. *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis*. PhD thesis, Vanderbilt University, Nashville, TN, USA, 1997.
- [29] P.J. Mosterman and G. Biswas. Behavior Generation Using Model Switching A Hybrid Bond Graph Modelling Technique. In *Proc. of the International Conference on Bond Graph Modeling, ICBGM'95*, pages 177–182, San Diego, California, Jan. 15–18 1995. Simulation Series, volume 27, no. 1.
- [30] I. Roychoudhury, M. Daigle, G. Biswas, X. Koutsoukos, and P.J. Mosterman. A Method for Efficient Simulation of Hybrid Bond Graphs. In J. J. Granda and F. E. Cellier, editors, *Proc. of the 2007 International Conference on Bond Graph Modeling and Simulation*, volume 39 (1), pages 177–184, 2007.
- [31] J.E. Strömberg. *A Mode Switching Modelling Philosophy*. PhD thesis, Linköping University, Linköping, Sweden, 1994.
- [32] J.E. Strömberg, J. Top, and U. Södermann. Variable Causality in Bond Graphs Caused by Discrete Effects. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 115–119. SCS Publishing, January 17–20 1993. Simulation Series, volume 25, no. 2, ISBN: 1-56555-019-6.
- [33] The AEGIS Technologies Group Inc. acslX Language Reference Guide Version 2.5 March 2009. URL <http://www.acslx.com>.
- [34] The Mathworks. MATLAB®/Simulink®. URL <http://www.mathworks.com/products/simulink/>.
- [35] J.U. Thoma. *Introduction to Bond Graphs and their Applications*. Pergamon Press, Oxford, 1975.

- [36] J. Top. *Conceptual Modelling of Physical Systems*. PhD thesis, Twente University of Technology, Enschede, Niederlande, 1993.
- [37] A.C. Umarikar and L. Umanand. Modelling of switched systems in bond graphs using the concept of switched power junctions. *Journal of the Franklin Institute*, 342:131–147, 2005.

Chapter 8

Multibody Systems

In Chapter 2, the fundamentals of bond graph based physical modelling have been provided. In the subsequent chapters, several aspects of bond graph modelling have been discussed in detail. In this chapter, we will consider an extension of the bond graph methodology that naturally follows from a formal introduction of the bond graph concept. Since this extension known as multibond graphs, is especially suited for modelling the three-dimensional (3D) motion of multibody systems (MBS) in mechanics, its presentation has been postponed to this chapter.

8.1 Brief Survey of Bond Graph Modelling of Multibody Systems

Clearly, modelling of the 3D motion of multibody systems is essential in robotics as well as for the analysis of vehicle dynamics in the automobile industry. There are numerous bond graph related publications in each of the two fields. As a comprehensive survey cannot be given in this section, some early pioneering research as well as some more recent work is referenced. As to bond graph modelling of robots with bodies assumed to be rigid, early research work has been carried out by Tiernego and Bos [54]. In 1992, Zeid and Chung reported the development of a library of bond graph models for three-dimensional joints [60]. Favre and Scavarda considered bond graph modelling of multibody systems with kinematic loops [16, 17]. Furthermore, several authors allowed parts of a robot to be flexible [31, 39, 58].

In bond graph modelling of vehicle dynamics, early publications are due to Bos [3] and Falco and Riviezzo [11] (modelling of a motorcycle), Pacejka and his co-authors [12, 47, 48] (tyre models, dynamics of heavy trucks), Hrovat [20] (automotive power trains) and Karnopp [25]. More recent work can be found, e.g., in [33–35, 40, 42, 52]. Moreover, as the increasing number of cars with petrol-operated engines cause increasing problems with regard to petrol consumption and pollution of the environment, there is an ongoing research for alternatives. Accordingly, electric vehicles have been the subject of bond graph modelling [15, 26].

In addition to industrial robots and vehicles, the human body [38], mobile robots such as walking robots [29, 41], or autonomous underwater vehicles [51] and space robots [49, 50] have also been modelled by bond graphs.

In the following, multibond graphs are introduced. On this basis, the multibond graph approach to bond graph modelling of multibody systems as introduced by Bos [3] is presented. The advantage of this clear and systematic approach is that models of a freely moving rigid body and various types of joints can be assembled according to the structure of the multibody system under consideration. Alternatively, it is shown how bond graph modelling can be used for the joint coordinate method, popular in modelling multibody systems in order to come up with a reduced set of equations of motions.

8.2 Multibond Graphs

In the introduction to bond graph modelling, bonds have been combined into so-called *multibonds* in order to simplify the general structure of all bond graphs (Figure 2.8). Moreover, it has already been observed that the vertices of a bond graph may have several power ports. In that case, they are called *multiports* (Definition 2.4). Examples of multiports already mentioned are 1- and 0-junctions. The fact that the other basic bond graph elements may also have several power ports and that these ports may be connected by multibonds goes back to an idea of Paynter. To the author's knowledge, such a bond graph representation was first used by Bonderson [2] for one-dimensional models of distributed parameter systems. He called it a *vector bond graph*. Vector bond graphs were taken up by Breedveld and integrated into a formal concept [6]. In order to avoid the association with oriented physical quantities, he replaced the term vector bonds by the notion of *multibonds*. Bos [3] and Tiernego [54] have used the multibond graph concept developed by Breedveld to describe the 3D motion of complex systems with rigid bodies in a clear and compact form. A more recent presentation of multibond graph modelling of multibody systems can be found in [10], Chapter 4.

Multibond graphs have also been used in other application areas (e.g. for thermodynamic systems) [4, 21, 53]. However, the representation of multibond graphs is not completely uniform in the literature and this compact bond graph representation is not equally appreciated by authors. Some critical remarks may be found in [14]. We will follow the definitions of the multibond graph language as given by Breedveld and will consider some of its features used in practice. In [6], Breedveld also introduced some less commonly used notions, e.g. fields of multibonds.

8.2.1 Multibonds and Arrays of Bond Graph Elements

If we consider the bond graphs of Figures. 5.2 and 5.3 in Section 5.2 that were developed to represent the motion of a mathematical pendulum in a plane, we see that the velocity of the point mass has been described with respect to a fixed Cartesian reference frame and that both components of the velocity in the x- and y-directions have been represented by a 1-junction. In modelling the 3D motion of a body, we have to account for a third velocity component.

A more compact representation is achieved by combining the 1-junctions into an array of 1-junctions (Figure 8.2) and adjacent bonds into multibonds (Figure 8.1).

To express that the symbol ‘1’ represents an array of 1-junctions, it is given an underscore. If required, the dimension of the array can be indicated by an index attached to the right side of the underscore. The number of bonds combined into a multibond is shown between the two parallel lines of the multibond. Thus, another attribute of multibonds in addition to the reference direction of the energy flow is its dimension. Single bonds can therefore be treated as a one-dimensional multibond. The dimension of multibonds is usually omitted because it is implicitly known from the context. For instance, edges in multibond graphs that model the 3D motion of multibody systems in mechanics usually have dimension three.



Fig. 8.1 Combination of n single bonds into a multibond of dimension n

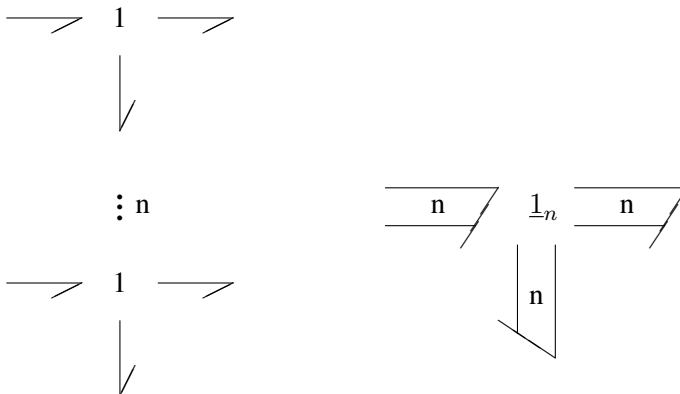


Fig. 8.2 Array of 1-junctions

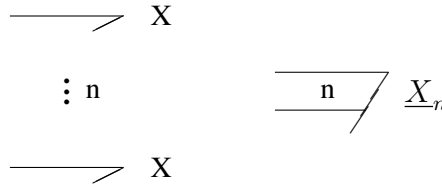


Fig. 8.3 Array of passive 1-port elements of the same type

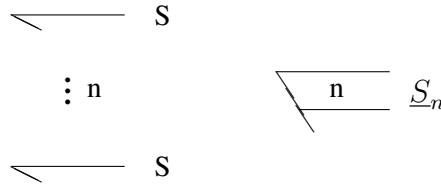


Fig. 8.4 Array of 1-port sources of the same type

Similarly, 0-junctions can be combined into arrays and any set of 1-port elements of the same type (dissipators, energy stores, or sources) can be grouped into a multiport element as depicted in Figure 8.3 where the symbol ‘X’ stands for a passive 1-port element (R, C, I). Figure 8.4 shows an array of 1-port sources of the same type, where the general symbol ‘S’ indicates either an S_e or S_f source.

Finally, we note that signal arrows (activated bonds (Definition 3.3) usually used for modulating elements can also be combined (Figure 8.5).

8.2.2 Multiport Energy Storage Elements

The combination of elements of the same type, as described in the previous section, does not link variables at different bonds. That is, only the j^{th} effort and the j^{th} flow out of the power variables of a multibond are related by a constitutive equation of a passive power port. Clearly, if independent sources are combined into an



Fig. 8.5 Combination of signal arrows



Fig. 8.6 C energy store with two power ports and its representation in a multibond graph

array of sources, then no relation exists between them. On the other hand, constitutive equations for devices from different areas, e.g., a lump of a beam assumed to be massless, a piezo crystal also assumed to be massless, mutually interacting coils, or a gas filled cylinder, indicate the need to introduce energy stores with several power ports where the port variables are related by the stored energy. In bond graphs with one-dimensional bonds, as many bonds as needed are simply attached to the storage element. In multibond graphs, all adjacent bonds are combined into a multibond. If a multibond of dimension n is connected to a power port, we say that the port has the dimension n . Elements with multibond ports will be depicted in this chapter by a bold symbol in order to distinguish them from those with ports for one-dimensional bonds (Figure 8.6). In the bond graph literature, the historical term *field* is sometimes used for multiport elements.

If all efforts of a multibond attached to a C energy store are grouped into a vector \mathbf{e} and correspondingly all flows into a vector \mathbf{f} , then the amount of power P into the multiport energy store can be expressed as

$$P = \mathbf{e}^T \cdot \mathbf{f}, \quad (8.1)$$

where the superscript T denotes the transposition of the vector. Clearly, from its equivalence to a 1-port C energy store, a C element with n ports for one-dimensional bonds is characterised by n functions that establish relations between a single effort and all generalised displacements. Any combination of integral and derivative causalities may occur at its power ports. This case cannot be expressed if a single multibond is used. Instead, the multibond must be split into two multibonds of different dimension and with different causality assigned. In the case when all ports have derivative causality, the constitutive equation of the C energy store takes the form

$$\mathbf{q} = \boldsymbol{\Phi}_C(\mathbf{e}), \quad (8.2)$$

where \mathbf{q} denotes the vector of generalised displacements (cf. Equation 2.35). If E denotes the energy stored in the multiport C element, then it is a function of the generalised displacement, as in the case of a 1-port C store.

$$E = E_0 + f_E(\mathbf{q}), \quad (8.3)$$

where E_0 denotes the energy stored at initial time t_0 . The effort at each power port i is obtained by differentiation of the state function (Equation 8.3) with respect to time.

$$e_i = \frac{\partial E}{\partial q_i} \quad (8.4)$$

($i = 1, \dots, n$). Since an energy store is energy conservative, it can be shown that the constitutive equations of a C energy store with n one-dimensional power ports must comply with *Maxwell's reciprocity condition*

$$\frac{\partial e_i}{\partial q_j} = \frac{\partial e_j}{\partial q_i}, \quad (8.5)$$

($i \neq j \quad i, j = 1, \dots, n$). For a linear constitutive relation, Maxwell's reciprocity condition means that the function Φ_C in Equation 8.2 must be specified by a *symmetric* matrix.

$$\mathbf{e} = \mathbf{C}^{-1} \mathbf{q} \quad (8.6)$$

In mechanical engineering, the matrix \mathbf{C}^{-1} is known as a stiffness matrix.

Similarly, an I energy store with n power ports (the dual counterpart to the multiport C store) is characterised by n constitutive equations that relate the flow at port i ($i = 1, \dots, n$) to all generalised momenta. Clearly, some ports of a multiport I energy store can have integral causality while the others have derivative causality. If all power ports are assigned derivative causality, the constitutive equation is of the form

$$\mathbf{p} = \Phi_I(\mathbf{f}), \quad (8.7)$$

where \mathbf{p} denotes the vector of generalised momenta. Again, from the principle of energy conservation, Maxwell's reciprocity condition results in the form

$$\frac{\partial f_i}{\partial p_j} = \frac{\partial f_j}{\partial p_i}, \quad (8.8)$$

($i \neq j \quad i, j = 1, \dots, n$). In mechanics, for example, the linear case of Equation 8.7 gives the relation between angular momentum \mathbf{p} and angular velocity $\boldsymbol{\Omega}$

$$\mathbf{p} = \mathbf{I} \cdot \boldsymbol{\Omega}, \quad (8.9)$$

where the inertia matrix \mathbf{I} is only a diagonal matrix if the axes of the reference frame are parallel to the principle axes of inertia of a body. In that case, the components of \mathbf{p} are decoupled.

8.2.3 Multiport Transformers and Gyrotors

The 2-port transformers and gyrotors introduced in Section 2.5.2 can also be extended into multiport elements. If the adjacent bonds of a transformer with $2n$ ports are grouped into two multibonds with a reference direction corresponding to the energy flow through the element, then the scalar modulus of a 2-port transformer becomes a $n \times n$ matrix \mathbf{T}

$$\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{f}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{T}^T \\ \mathbf{T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{e}_2 \end{bmatrix}. \quad (8.10)$$

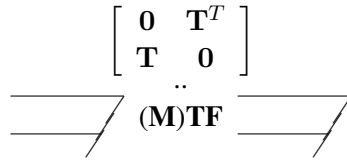


Fig. 8.7 Representation of a multiport transformer



Fig. 8.8 Representation of a multiport gyrator

Figure 8.7 shows the representation of a multiport transformer with two adjacent multibonds. Multiport transformers can be used to represent transformations between reference frames that frequently occur in the modelling of the planar or 3D motion of mechanical systems [23]. Such multiport transformers can be decomposed in a bond graph with one-dimensional bonds that uses 2-port transformers as well as 1- and 0-junctions as has been shown in Section 5.2 for the transformation between Cartesian and polar coordinate systems.

Similarly to Equation 8.10, a multiport gyrator is characterised by the equation

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \tag{8.11}$$

where the matrix \mathbf{G} denotes the ratio of the multiport gyrator. If *all* the one-dimensional bonds of a multiport transformer (or a multiport gyrator) are combined into one single multibond with an inward orientation for the reference direction of the energy flow (Figure 8.8), then the matrix in the constitutive equation of both elements must be skew-symmetric because both elements must be power conservative.

8.2.4 Rotation of a Rigid Body in Space Described by a Multiport Gyrator

A well known application of a modulated gyrator described by Karnopp [24] is the rotation of a rigid body in space according to Euler’s equation. In mechanics, the rotation of a rigid body with respect to an axis through its centre is usually described by a body fixed reference frame (x, y, z) , while the translational motion

of the centre of gravity is described with respect to a reference frame (X, Y, Z) fixed in space. Let

$\boldsymbol{\Omega}'$ be the absolute angular velocity of the body, \mathbf{p}' the angular momentum and \mathbf{M}' be the moment acting on the body be described in coordinates of the body fixed moving reference frame. Then, the rotation of the body is described by Euler's equation

$$\mathbf{M}' = \frac{d\mathbf{p}'}{dt'} + \boldsymbol{\Omega}' \times \mathbf{p}', \quad (8.12)$$

where d/dt' denotes the time derivative in the body fixed frame. The cross product $\boldsymbol{\Omega}' \times \mathbf{p}'$ in Equation 8.12 accounts for the fact that the reference frame is moving. If the axes of the body fixed coordinate system are parallel to the body's principle axes of inertia, then the equation for the angular momentum is of the form

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix} = \underbrace{\begin{bmatrix} J'_x & 0 & 0 \\ 0 & J'_y & 0 \\ 0 & 0 & J'_z \end{bmatrix}}_{=: \mathbf{J}'} \begin{bmatrix} \Omega'_x \\ \Omega'_y \\ \Omega'_z \end{bmatrix} \quad (8.13)$$

where the principle momenta of inertia are constant. If the Eulerian equations are written for each component of the angular momentum, then, as has been shown by Karnopp and Rosenberg [27], they can be elegantly represented as a ring of modulated gyrators (Figure 8.9), which has been called the *Eulerian Junction Structure* (EJS).

If $\tilde{\boldsymbol{\Omega}}'$ denotes the skew symmetric matrix

$$\tilde{\boldsymbol{\Omega}}' := - \underbrace{\begin{bmatrix} 0 & \Omega'_z & -\Omega'_y \\ -\Omega'_z & 0 & \Omega'_x \\ \Omega'_y & -\Omega'_x & 0 \end{bmatrix}}_{=: X(\boldsymbol{\Omega})}, \quad (8.14)$$

generated by the vector $\boldsymbol{\Omega}' = [\Omega'_x \ \Omega'_y \ \Omega'_z]^T$, then the cross product $\boldsymbol{\Omega}' \times \mathbf{p}'$ of angular velocity $\boldsymbol{\Omega}'$ and angular momentum \mathbf{p}' can be written in the form

$$\boldsymbol{\Omega}' \times \mathbf{p}' = (\tilde{\boldsymbol{\Omega}}' \mathbf{J}') \cdot \boldsymbol{\Omega}' = -(\widetilde{\mathbf{J}\boldsymbol{\Omega}}) \cdot \boldsymbol{\Omega} = X(\mathbf{J}\boldsymbol{\Omega}) \cdot \boldsymbol{\Omega}, \quad (8.15)$$

where $\tilde{\boldsymbol{\Omega}}'$ denotes the matrix in Equation 8.14 and \mathbf{J}' the inertia matrix in Equation 8.13. That is, in extended form, the external product $\boldsymbol{\Omega}' \times \mathbf{p}'$ reads

$$\begin{aligned} \boldsymbol{\Omega}' \times \mathbf{p}' &= X(\mathbf{J}\boldsymbol{\Omega}) \cdot \boldsymbol{\Omega} \\ &= \begin{bmatrix} 0 & J'_z \Omega'_z & -J'_y \Omega'_y \\ -J'_z \Omega'_z & 0 & J'_x \Omega'_x \\ J'_y \Omega'_y & -J'_x \Omega'_x & 0 \end{bmatrix} \cdot \begin{bmatrix} \Omega'_x \\ \Omega'_y \\ \Omega'_z \end{bmatrix} \end{aligned}$$

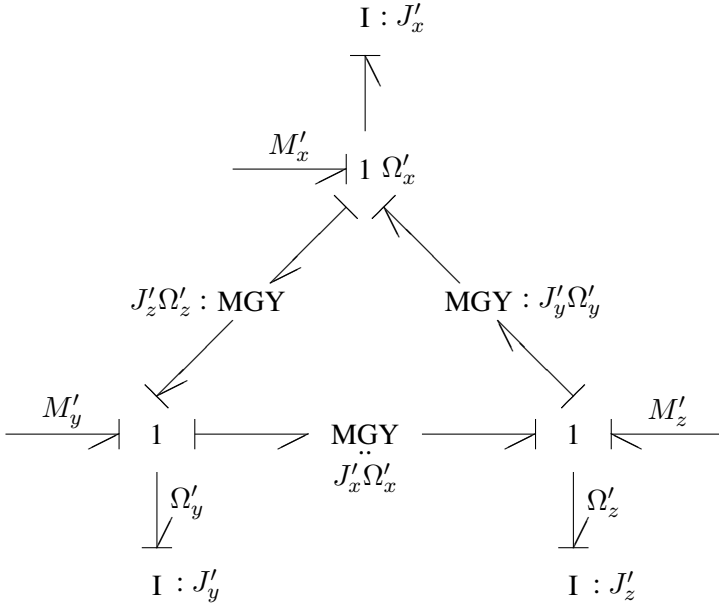


Fig. 8.9 Eulerian Junction Structure, EJS, of a rotating body (Karnopp and Rosenberg 1968)

$$= \begin{bmatrix} -(J'_y - J'_z)\Omega'_y\Omega'_z \\ (J'_x - J'_y)\Omega'_x\Omega'_z + (J'_y - J'_z)\Omega'_x\Omega'_z \\ -(J'_x - J'_y)\Omega'_x\Omega'_y \end{bmatrix}. \quad (8.16)$$

Equation 8.16 gives rise to another bond graph representation of Euler’s equation displayed in Figure 8.10. This representation without bond loops has been found by Breedveld. In [7], he shows that it has some advantages over the well known and frequently used Eulerian junction structure. Clearly, if there are any body symmetries, they immediately affect the bond graph as the moduli of the modulated gyrators include a difference of moments of inertia. Moreover, a closer look at the signal loops indicate that rotation around the smallest and the largest axis is stable, while it is not around the middle axis. This is not reflected by the common Eulerian junction structure because of its symmetry.

Furthermore, Equation 8.15 means that we can represent Euler’s equation in a compact way by a multibond graph in which an I field and a modulated multiport gyrator with the matrix $\mathbf{EJS} := \tilde{\Omega}'\mathbf{J}'$ are attached to an array of 1-junctions (Figure 8.11).

The modulated multiport gyrator arises because the rotation of the body is described with respect to the moving reference frame (x, y, z) and not to a fixed inertial frame (X, Y, Z) . Thus, it can be viewed as the result of a change of the reference frame. Let Ω denote the angular velocity with respect to an inertial frame and Ω' the angular velocity expressed in coordinates of the moving reference frame, then

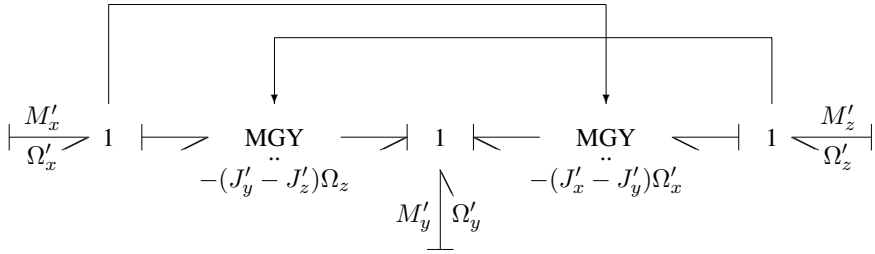


Fig. 8.10 Bond graph representation of Euler’s equation according to Breedveld (1999)

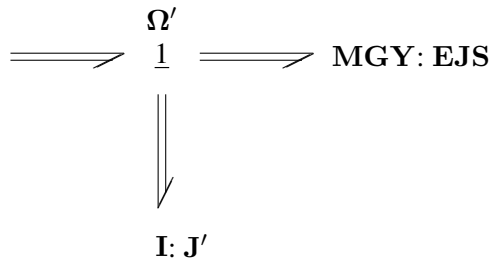


Fig. 8.11 Multibond graph representation of Euler’s equation of motion

the transformation

$$\mathbf{\Omega} = \mathbf{A}^T \mathbf{\Omega}' \tag{8.17}$$

can be represented by a modulated multiport transformer with the matrix \mathbf{A} . Clearly, since the body fixed coordinate system moves, the coefficients of the matrix \mathbf{A} depend on the momentary position of the body in space. Since a transformation between velocities is power conservative, the torque on the body with respect to the moving frame and the torque with respect to the inertial system are related by the equation

$$\mathbf{M}' = \mathbf{A}^T \mathbf{M} . \tag{8.18}$$

This transformation and the torque equation

$$\mathbf{M} = \frac{d}{dt} (\mathbf{J}\mathbf{\Omega}) \tag{8.19}$$

are expressed in the multibond graph of Figure 8.12, where quantities without prime refer to the inertial system.

If the rotation is described with respect to the body fixed moving reference frame, then the inertia must be ‘transformed’ over the modulated transformer. From the simple bond graph of a slider crank mechanism (Figure 4.30), we know that the transformed inertia is accompanied by a so-called gyristor. Allen [1] chose this notation because the element can be decomposed into a gyrator and into a resistor. If

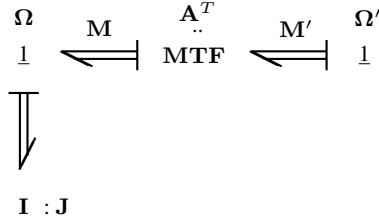


Fig. 8.12 Transformation of the angular velocity

the matrix of the multiport transformer represents a transformation between coordinate systems, then the resistor vanishes. In order to see this, we rewrite the relation between the moments, Equation 8.18, in several steps.

$$\mathbf{M}' = \mathbf{A}^T (\dot{\mathbf{J}}\boldsymbol{\Omega} + \mathbf{J}\dot{\boldsymbol{\Omega}}) \tag{8.20}$$

$$= \mathbf{A}^T \dot{\mathbf{J}}\mathbf{A}\boldsymbol{\Omega}' + \mathbf{A}^T \mathbf{J} \frac{d}{dt} (\mathbf{A}\boldsymbol{\Omega}') \tag{8.21}$$

$$= (\mathbf{A}^T \frac{d}{dt} (\mathbf{J}\mathbf{A})) \boldsymbol{\Omega}' + (\mathbf{A}^T \mathbf{J}\mathbf{A}) \dot{\boldsymbol{\Omega}}' \tag{8.22}$$

The first term in this sum establishes a relation between an effort and a flow and describes the gyristor element. The second term represents the inertia with respect to the moving coordinate system. Now, if \mathbf{A} is the matrix of a transformation between reference frames, then

$$\mathbf{A}^T = \mathbf{A}^{-1} \tag{8.23}$$

and

$$\dot{\mathbf{A}} = \mathbf{A} \tilde{\boldsymbol{\Omega}}' . \tag{8.24}$$

By using both properties, reformulation of the first term in Equation 8.22 yields, after a number of steps, the expression $(\boldsymbol{\Omega}'\mathbf{J}')\boldsymbol{\Omega}$ because $\tilde{\boldsymbol{\Omega}}'\boldsymbol{\Omega}' = \mathbf{0}$. In this expression, \mathbf{J}' is the inertia matrix of the body with respect to the moving reference frame ($\mathbf{J}' = \mathbf{A}^T \mathbf{J}\mathbf{A}$). Thus, the gyristor reduces to a modulated gyristor.

$$\mathbf{M}' = \mathbf{J}'\dot{\boldsymbol{\Omega}}' + (\tilde{\boldsymbol{\Omega}}'\mathbf{J}') \boldsymbol{\Omega}' \tag{8.25}$$

Again, this is Euler’s equation represented by the multibond graph in Figure 8.11. By looking at the multibond graph of Figure 8.12, we see that the result is indeed a transformation of the inertia over the modulated transformer for which the gyristor accompanying the inertia reduces to a modulated gyristor (Figure 8.13).

$$\begin{array}{c}
 \begin{array}{c} \overline{\overline{M}} \\ \overline{\overline{1}} \end{array} \begin{array}{c} \Omega \\ \underline{1} \end{array} \begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \end{array} \begin{array}{c} \overline{\overline{A^T}} \\ \overline{\overline{MTF}} \end{array} \begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \end{array} \begin{array}{c} \overline{\overline{-M'}} \\ \overline{\overline{1}} \end{array} \begin{array}{c} \Omega' \\ \underline{1} \end{array} \begin{array}{c} \rightleftarrows \\ \rightleftarrows \\ \rightleftarrows \end{array} \text{MGY} : \tilde{\Omega}' \text{J}' \\
 \\
 \begin{array}{c} \Downarrow \\ \Downarrow \\ \Downarrow \end{array} \\
 \text{I} : \text{J}'
 \end{array}$$

Fig. 8.13 Transformation of an inertia over the MTF of a coordinate transformation

8.2.5 Multiport Resistors

Clearly, resistors can also have several power ports. Nonlinear devices, e.g. bipolar transistors, are often described as resistors with several ports if the modelling context permits a static model. On the other hand, 1-port resistors connected by a part of the junction structure may be replaced by a multiport resistor if, for instance, algebraic loops occur in a part of the bond graph. Karnopp and Rosenberg call such multiport resistors with an internal structure *implicit fields* [28].

A multiport resistor is characterised by algebraic relations between its effort and flow variables. Depending on the elements, an R field connected to all of its ports may have resistance or conductance causality, or some ports may have resistance causality while the rest have conductance causality. If one of the constitutive relationships is not invertible, then the determination of the causality at that port is independent of the power port it is connected to.

There is no condition similar to Maxwell’s reciprocity for stores that the algebraic equations of a multiport resistor must comply with. What can be said is that if a multiport resistor is an implicit field in which linear 1-port resistors are connected via 0- and 1-junctions and transformers, then the matrix in the constitutive vector equation is symmetric *in general* if the equations of *all* 1-port resistors are written in resistance or in conductance causality. In the literature, this is sometimes called *Onsager’s reciprocity*.

Furthermore, because the element is dissipative, the matrix of a linear multiport resistor must be positive definite. Let us assume that all equations can be written in resistance causality. If all efforts are grouped into a vector \mathbf{e} and all flows into a vector \mathbf{f} , then the amount of power, P , into the element can be written as $P = \mathbf{e}^T \mathbf{f} = \mathbf{f}^T \mathbf{e}$. The constitutive equation $\mathbf{e} = \mathbf{R} \mathbf{f}$ yields

$$P = \mathbf{f}^T \mathbf{R} \mathbf{f} > 0 \tag{8.26}$$

for all $\mathbf{f} \neq \mathbf{0}$. That is, the matrix \mathbf{R} is positive definite. Note that it is not always possible to write the constitutive equations of a linear multiport resistor in Onsager’s form. Karnopp and Rosenberg ([28] p. 262) give a counter example. However, each matrix can be split into a symmetric and an antisymmetric part where the latter does not contribute to the matrix being positive definite. For a multiport resistor in



Fig. 8.14 Generalisation of the RS element into a RS field

resistance causality, the matrix \mathbf{R} can be written in the form

$$\mathbf{R} = \mathbf{R}^s + \mathbf{R}^a , \tag{8.27}$$

where $\mathbf{R}^s = (\mathbf{R} + \mathbf{R}^T)/2$, $\mathbf{R}^a = (\mathbf{R} - \mathbf{R}^T)/2$ and

$$\mathbf{f}^T \mathbf{R}^a \mathbf{f} = 0 . \tag{8.28}$$

The matrix \mathbf{R}^s describes the irreversible dissipation of free energy. In the case of a 2×2 matrix, its antisymmetric part \mathbf{R}^a is a gyrator. Accordingly, Breedveld generalises the definition of a multiport gyrator (Equation 8.11) by considering each multiport element characterised by an antisymmetric matrix as multiport gyrator ([5], p. 23).

Finally, if in the non-isothermal case, irreversible transformation of non-thermal energy into heat is represented in a multibond graph, then the multiport resistor must be replaced by the multiport version of the RS element (Section 2.5.4) introduced by Thoma. On the non-thermal side, the one-dimensional bond is replaced by a multibond (Figure 8.14). According to Equation 2.67 the RS field is power conservative.

$$\mathbf{e}^T \mathbf{f} = T \times \dot{S} \tag{8.29}$$

Furthermore, $\mathbf{e}^T \mathbf{f} > 0$ as $\dot{S} > 0$.

8.2.6 Splitting a Multibond

We conclude this section on multibond graphs by considering the splitting of a multibond into one-dimensional bonds or multibonds of lower dimension as introduced by Breedveld in [6] and called *direct sum*. Figure 8.15 shows a multibond of dimension n split into n one-dimensional bonds. Such a splitting of a multibond allows some of the bonds of a multibond to be connected with element ports and to express constraints that do not apply for all components of the multibond. We will make use of this feature in bond graph modelling of the 3D motion of multibody systems.

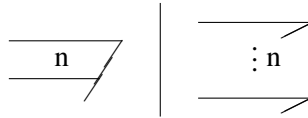


Fig. 8.15 Splitting of an n -dimensional multibond into n one-dimensional bonds

8.3 Bond Graph Modelling of the 3D Motion of Multibody Systems

The multibond graph fragments introduced in the previous section allow the 3D motion of multibody systems with rigid bodies to be modelled in a clear and concise form. The approach of Bos [3] and Tiernego [54] can be applied to develop reusable models for a freely moving rigid body as well as for several types of joints. Instantiations of the rigid body model and of joint models are then simply assembled according to the structure of the multibody system.

8.3.1 Multibond Graph of a Freely Moving Rigid Body

The multibond graph of a rigid body represents the Newton-Euler equations of motion. Translational motion of the centre of gravity is described with respect to a fixed inertial frame while the body's rotation is related to a body fixed reference frame with the origin at the body's centre of gravity. This coincidence of the reference point on the body with its centre of gravity results in the translational velocity of the centre of gravity being decoupled from the angular velocity and the forces being decoupled from torques acting on the body. As a result, the multibond graph of a rigid body takes a simple symmetric form. If, for simplicity, it is assumed that two hinge points connect the body to the joints, then the multibond graph of a freely moving rigid body has the form shown in Figure 8.16 [3].

It serves as a building block in bond graph modelling of multibody systems with rigid bodies. In the case of robots with a tree-like structure, a hinge point on the joints linking the robot's limbs is generally chosen as the origin of the body fixed reference frame rather than the centre of mass of a body. The resulting multibond graph of a freely moving body is given in [3]. In this chapter on multibond graph modelling of multibody systems, it is assumed that a body fixed reference frame sits in the body's centre of gravity and that multibody systems have no kinematic loops. A simple example of a multibody system with a kinematic loop is the well known four bar mechanism. It consists of four bars that are connected by revolute joints and can move in a plane. For bond graph modelling of multibody systems with kinematic loops, refer to [17].

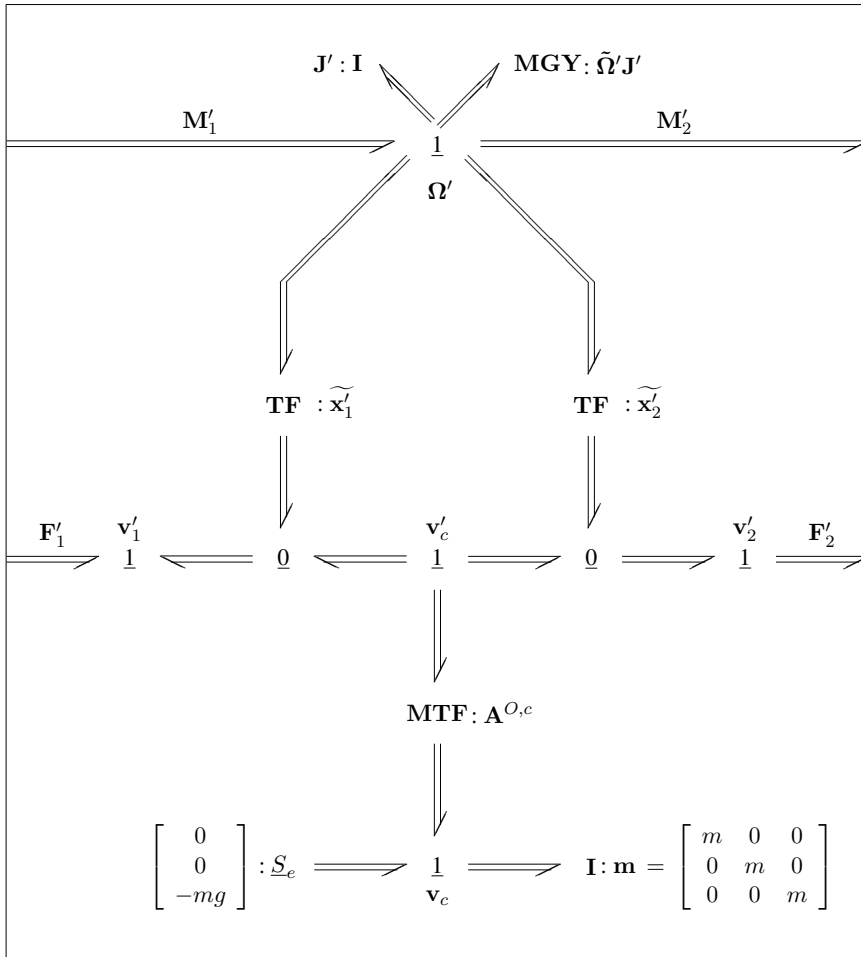


Fig. 8.16 Multibond graph of a freely moving rigid body with two hinge points (Bos 1986)

A velocity in space has three components with respect to a reference frame. In a bond graph with one-dimensional bonds, each component must be represented by a 1-junction. A clearer and more compact model representation is achieved by combining all three velocity components into a multibond that represents a velocity in space by a three-dimensional array of 1-junctions (Figure 8.17).

Thus, all 1-junctions with an underscore in the multibond graph of Figure 8.16 represent a velocity in space. In particular, the array of 1-junctions with the annotation v_c stands for the translational velocity of the body's centre of mass with respect to a fixed inertial frame (X, Y, Z) with the origin O , while the array of 1-junctions annotated by v'_c represents the absolute velocity of the centre of mass expressed



Fig. 8.17 Bond graph and multibond graph representation of a velocity in space

in the basis of the body fixed reference frame (x, y, z) . The modulated transformer with the matrix $\mathbf{A}^{O,c}$ establishes the transformation from the local body fixed moving reference frame with its origin in the body’s centre of mass to the inertial frame with its origin in the fixed point O .

$$\mathbf{v}_c = \mathbf{A}^{O,c} \mathbf{v}'_c \tag{8.30}$$

Since the body’s centre of mass is moving in space, the coefficients of the transformation matrix depend on the momentary position of the body with respect to the inertial frame. That is, the coefficients are not constant. Therefore, a modulated transformer is used in the multibond graph of Figure 8.16. The coefficients can be expressed by means of Cardan angles or Euler parameters. The latter have the advantage over Cardan angles that the coefficients of the transformation matrix cannot become singular [56].

In Figure 8.16, the upper array of 1-junctions with the name Ω' represents the body’s absolute angular velocity expressed in coordinates of the local reference frame with its origin in the body’s centre of mass. The left side and the right side arrays of 1-junctions with names \mathbf{v}'_1 and \mathbf{v}'_2 represent the absolute velocities of the two hinge points expressed in coordinates of the body fixed reference frame. Generally speaking, all quantities with a prime are expressed in coordinates of the body fixed reference frame. The sum of efforts at the lower array of 1-junctions accounts for Newton’s law applied to the body’s centre of mass. From the multibond graph in Figure 8.16, we derive

$$(\mathbf{A}^{O,c})^T (\mathbf{m}\dot{\mathbf{v}}_c - \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}) = \mathbf{F}'_1 - \mathbf{F}'_2, \tag{8.31}$$

where \mathbf{F}'_1 and \mathbf{F}'_2 are external forces acting in the body’s hinge points. The sum of efforts at the upper array of 1-junctions gives Euler’s equation.

$$\mathbf{J}' \Omega' + \widetilde{\Omega}' \mathbf{J}' \Omega' = \mathbf{M}'_1 - \mathbf{M}'_2 + (\widetilde{\mathbf{x}'_1})^T \mathbf{F}'_1 - (\widetilde{\mathbf{x}'_2})^T \mathbf{F}'_2, \tag{8.32}$$

where \mathbf{x}'_1 and \mathbf{x}'_2 are the distances of the hinge points from the origin of the body fixed reference frame in coordinates of that frame. $(\widetilde{\mathbf{x}'_i})$ ($i = 1, 2$) denotes the skew

symmetric matrix generated by \mathbf{x}'_i (cf. Equation 8.14). \mathbf{M}'_1 and \mathbf{M}'_2 are external torques acting on the body.

The left side and the right array of 0-junctions express that the velocity of a hinge point is the velocity of the centre of mass superimposed by a term $\tilde{\mathbf{x}}'_i \boldsymbol{\Omega}' = \boldsymbol{\Omega}' \times \mathbf{x}'_i$ ($i = 1, 2$) that is due to the rotation of the body.

$$\mathbf{v}'_i = \mathbf{v}'_c + \tilde{\mathbf{x}}'_i \boldsymbol{\Omega}' \tag{8.33}$$

($i = 1, 2$)

8.3.2 Connecting Instances of the Rigid Body Model

The external forces and moments in the dynamic equations are expressed in coordinates of the local body fixed reference frame, as are the angular velocity and the velocities of the hinge points. This means that if the submodels of two bodies are to be connected, the outputs of one submodel must be transformed to the local reference frame of the other. The coefficients of the transformation matrix are not constant. Therefore, the models of the rigid bodies of a multibody system must be connected by modulated transformers.

In Figure 8.18, the lower index of a velocity denotes the body and the upper index the reference frame to which the vectors coordinates are related. If $\mathbf{A}^{0,i}$ describes the transformation from the i^{th} body to the inertial frame, then the transformation from the reference frame of the i^{th} body to that of the $(i + 1)^{th}$ body is given by the matrix

$$\mathbf{A}^{i+1,i} = (\mathbf{A}^{0,i+1})^T \mathbf{A}^{0,i} . \tag{8.34}$$

Now, the rigid bodies of a multibody system are not rigidly coupled as shown in Figure 8.18, but are connected by joints that are generally of different type. Thus, in addition to the modulated transformers, a submodel of the joint needs to be inserted between the submodels of the rigid bodies (Figure 8.19). Joints allow for a number

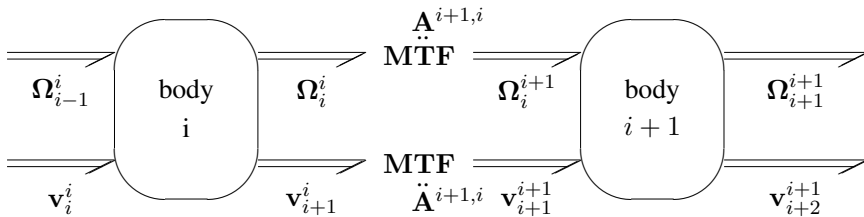


Fig. 8.18 Connection of submodels of two rigid bodies via modulated transformers (Tienego and Bos 1985)

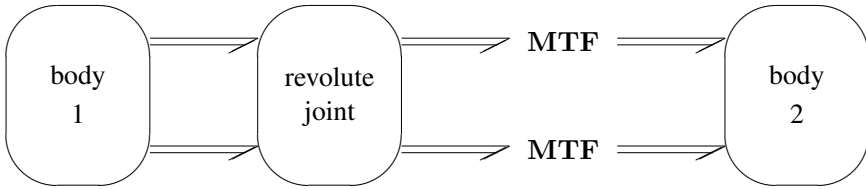


Fig. 8.19 Word-bond graph of two rigid bodies connected by a revolute joint

of degrees of freedom. The nature of these degrees of freedom depend on the type of the joint.

Since the purpose of this chapter is to show the potential of multibond graphs for modelling multibody system in principle, we will confine ourselves to consider a revolute joint and a prismatic joint and use their models in the example of a simple robot with three degrees of freedom (cf. Figure 8.22).

8.3.3 Multibond Graph Model of a Revolute Joint

If body 2 can rotate with respect to the x-axis of body 1 against friction, then the angular velocities of both bodies differ only in this component, while the translational velocities of their centres of mass are the same. The difference in the x-component of their angular velocity can be expressed by splitting the corresponding multibond into one-dimensional bonds and by inserting a 0-junction in the one-dimensional bond representing the angular velocity's x-component. Thus, two rigid bodies connected by a revolute joint can be represented by a word-bond graph as shown in Figure 8.19. The submodel of the revolute joint has the structure shown in Figure 8.20.

The resistor accounts for friction in the joint. In Figure 8.20, \mathbf{v}_{21}^1 is the velocity of the hinge point number 1 of body 2 expressed in coordinates of the reference frame of body 1. Accordingly, \mathbf{v}_{12}^1 is the velocity of hinge point number 2 of body 1 in coordinates of reference frame 1. Similarly, for the velocities of the hinge points, Ω_2^1 means the angular velocity of body 2 expressed in the coordinate frame of body 1.

8.3.4 Multibond Graph Model of a Prismatic Joint

The bond graph model of a prismatic joint has a structure analogous to that of the revolute joint. Let us assume that body 3 is connected to body 2 and can move along the y-axis of the reference frame of body 2 as depicted in Figure 8.22. In this case, the translational velocities of the hinge points of the two bodies differ, while both

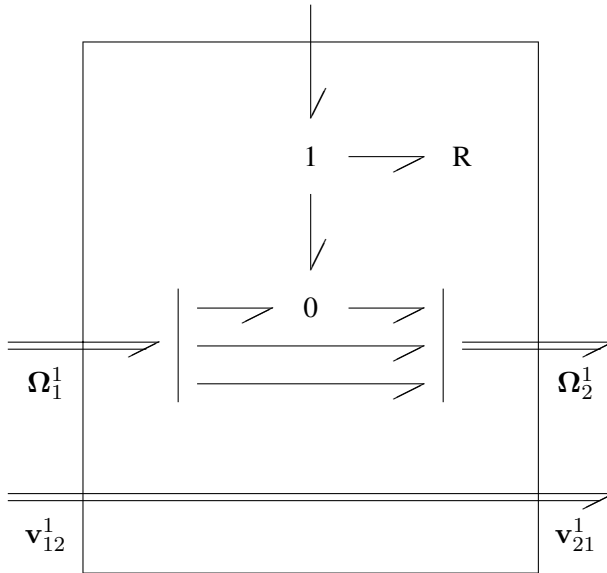


Fig. 8.20 Bond graph model of a revolute joint

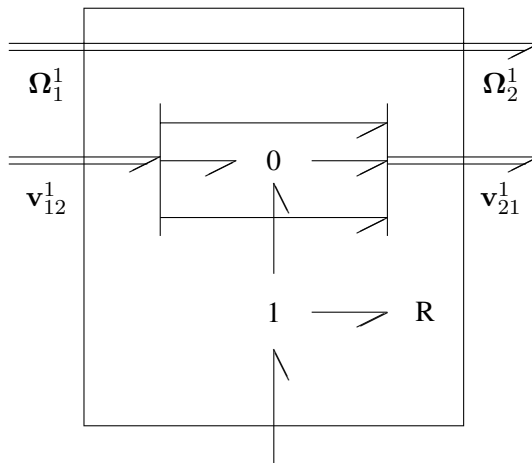


Fig. 8.21 Bond graph model of a prismatic joint

share the same angular velocity. Thus, the bond graph model of the prismatic joint has the structure shown in Figure 8.21. Again, the resistor represents friction in the joint.

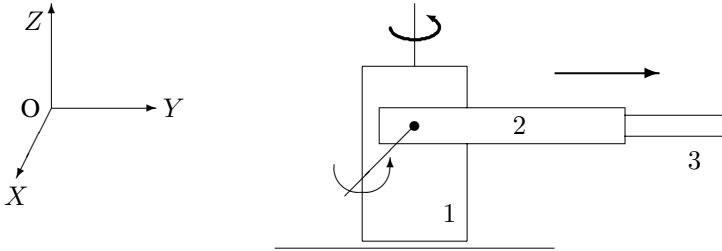


Fig. 8.22 Schematic of a robot with three degrees of freedom

8.3.5 Multibond Graph of a Three Degrees of Freedom Robot

Now, by means of the model of a freely moving rigid body (Figure 8.16) and the models of a revolute joint and a prismatic joint, a multibond graph model can be composed in a systematic manner for the robot sketched in Figure 8.22 with two revolute degrees of freedom and one prismatic degree of freedom. Each joint is driven by a motor.

The resulting multibond graph of the robot is shown in Figure 8.23. In this bond graph model a load of mass m has been added at the tip of body 3. In Figure 8.23, velocities equal to $\mathbf{0}$ are taken into account by sources imposing these boundary conditions. For instance, body 1 is connected to the floor and can only rotate with respect to its Z -axis. Thus, the translational velocity of its hinge point number 1 is equal to zero. Two components of its angular velocity are fixed at zero, while the body is rotated by a motor with respect to the Z -axis. Therefore, the multibond of the angular velocity has been split into three one-dimensional bonds.

A simpler version of the robot in Figure 8.22 with the prismatic joint being locked is considered in Section 12.9. In that case study, the standard form of robot equations [9] is systematically derived from the multibond graph of the robot.

8.3.6 Causalities in Multibond Graphs

When assigning causalities to a multibond graph of a multibody system, some multibonds may need to be split because not all of their components can carry the same causality. Moreover, the joint models will generally result in many dependent I stores. Several of the six degrees of freedom of a freely moving rigid body may disappear due to a joint of certain type that links one body to another. The joint models considered represent *ideal* rigid joints solely described by kinematic constraints. If they connect instantiations of the body model, then the body velocities become dependent. The resulting derivative causalities at the power ports of I stores

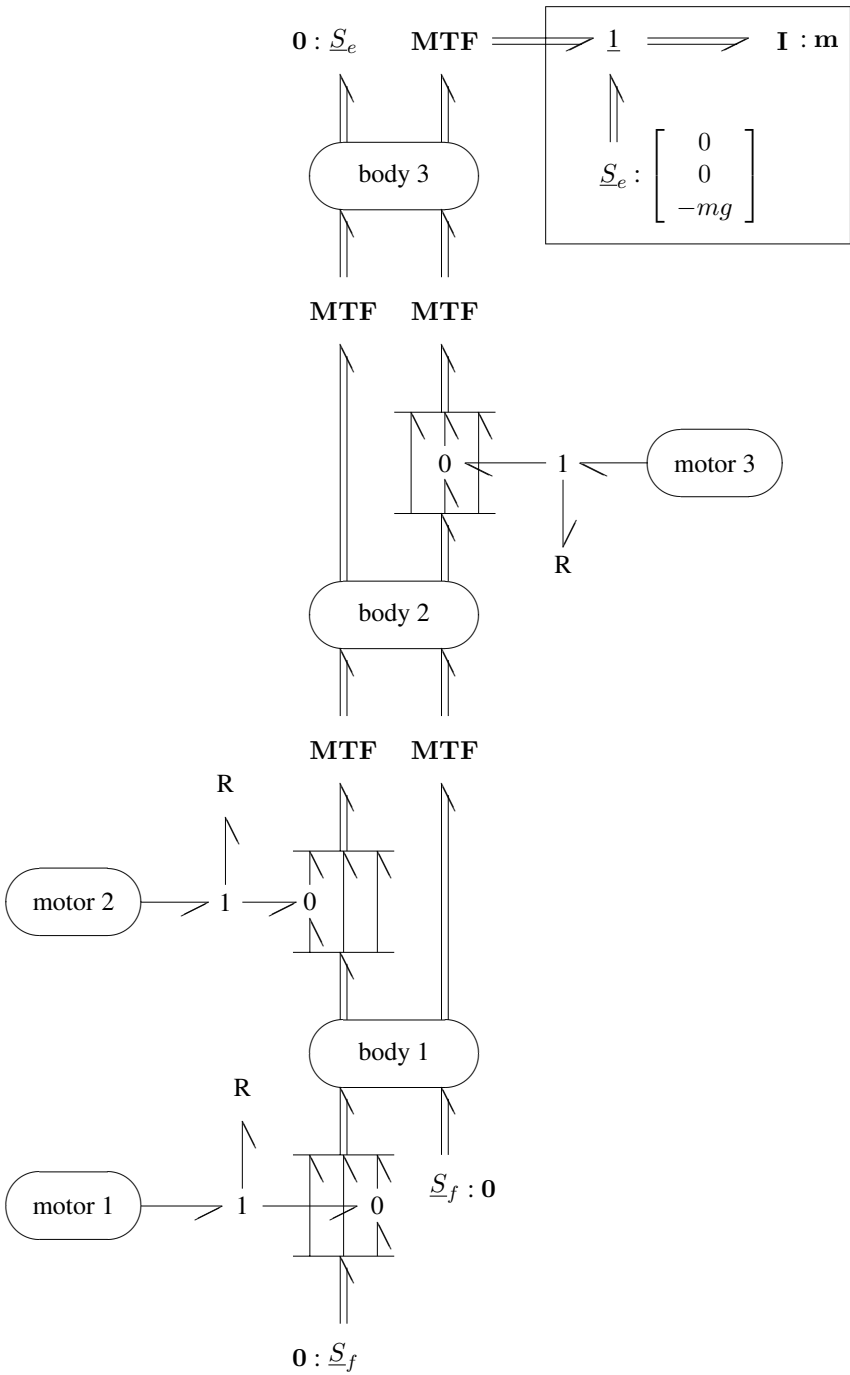


Fig. 8.23 Multibond graph of the robot in Figure 8.22

can be avoided by inserting special controlled sinks into the joint models. Their effort represents a constraint force or a constraint torque that ensures that the relative velocity (angular velocity) is zero. This kind of controlled sinks has already been considered in Sections 5.2 and 5.4. These constraint efforts are introduced by means of a Lagrange multiplier. Figure 8.24 shows a model of a revolute joint with controlled sinks for constraint forces and constraint torques. Due to the use of constraint forces or constraint torques in the joint models, the ports of all I stores receive integral causality. Consequently, their ODEs are of explicit form with a right-hand side that depends on the constraint forces or moments. The kinematic constraints of the joints, however, establish algebraic dependencies between the state variables of the inertia that seem to be independent. This semi-explicit state space model can be solved by means of a code based on the BDF method, although it is a DAE system of index 2 due to the fact that the constraint forces or moments do not appear in the algebraic constraints. A program that can process bond graph models of multibody systems including Lagrange multipliers is BONDYN [18]. It uses the DASSL code [8].

8.4 The Joint Coordinate Method

The bond graph approach to modelling multibody systems, going back to Bos [3], enables a multibond graph for a multibody system to be composed in a clear and systematic manner by using a model of a freely moving rigid body with models for the different types of joints. In the rigid body model, the translational motion of the centre of mass is expressed in coordinates of an inertial frame while the body's rotation is related to a body fixed reference frame with its origin in a body's point. In the previous section, the centre of mass was chosen as the origin of the body fixed reference frame because it results in a bond graph model with a simple and symmetric structure. However, another point, e.g. a hinge point, can also be chosen as a reference point [3].

The advantages of the method are offset by the disadvantage that the set of equations used is very large compared with the number of degrees of freedom in a multibody system, which results in a corresponding high computational effort. If f_j denotes the number of relative degrees of freedom of a joint, j , then there are six equations of motion for each body and $6 - f_j$ kinematic constraints on the velocities at each joint to be derived from the bond graph. The resulting DAE system is of index 1. If derivative causalities at many inertia ports are removed by accounting for constraint forces or torques in the joints, then the index of the system is increased to two. It is true that such a DAE system can be solved by a code based on the BDF method, but results are less accurate and the computational effort is higher in comparison to the case where causal paths between inertia ports are accepted and a DAE system of index 1 is solved [55].

Still, it is quite common to formulate equations of motions by starting with six coordinates for each body, adding kinematic constraints and breaking algebraic de-

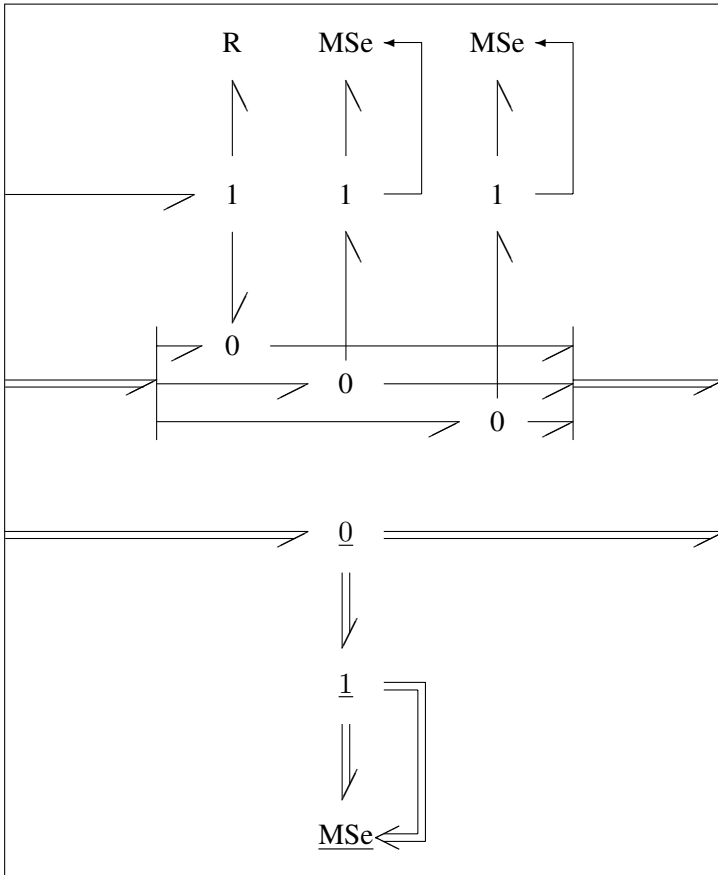


Fig. 8.24 Model of a revolute joint with controlled sinks representing constraint forces and constraint torques

dependencies by using constraint forces or moments [46]. However, in mechanics, this approach has not been traditionally used in combination with bond graphs.

In order to reduce the number of equations, some authors use other generalised coordinates. For example, the *joint coordinate method* is well known in mechanics [43, 45] and will be briefly recalled in its classical form. Subsequently, relations to bond graph modelling are considered. For simplicity, the presentation is confined to multibody systems with a tree-like structure. That is, no kinematic loops will be taken into account.

8.4.1 Formulation of a Reduced Set of Equations of Motion

The joint coordinate method uses joint coordinates as generalised coordinates and generalised constraint forces introduced by means of Lagrange multipliers and has the attractive feature that it leads to a minimal set of equations of motion in which generalised constraint forces no longer appear. For systems with a tree-like structure, the number of equations of motions is equal to the number of degrees of freedom.

If generalised constraint forces in the joints are introduced, then Newton-Euler's equations of motion can be written in the form

$$\mathbf{M} \dot{\mathbf{v}} - \mathbf{D}^T \boldsymbol{\lambda} = \mathbf{g}, \quad (8.35)$$

where the absolute translational velocities and angular velocities of a rigid body are combined into a vector \mathbf{v} . The mass matrix \mathbf{M} includes the masses and moments of inertia of all bodies. The term $\mathbf{D}^T \boldsymbol{\lambda}$ represents the generalised constraint forces and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Finally, the vector \mathbf{g} includes all gyroscopic accelerations, the forces and torques due to springs and resistors as well as external forces and torques. The joints impose additional kinematic constraints of the form

$$\mathbf{D} \mathbf{v} = \mathbf{0}. \quad (8.36)$$

The essential step in the reduction of the number of equations of motion is to express absolute Cartesian body coordinates by means of a lower number of independent relative joint coordinates grouped into a vector \mathbf{w} .

$$\mathbf{v} = \mathbf{B} \mathbf{w} \quad (8.37)$$

If the velocity transformation, Equation 8.37, is differentiated with respect to time and used in the equation of motion, Equation 8.35, we then get

$$\mathbf{M} \mathbf{B} \dot{\mathbf{w}} - \mathbf{D}^T \boldsymbol{\lambda} = \mathbf{g} - \mathbf{M} \dot{\mathbf{B}} \mathbf{w}. \quad (8.38)$$

Multiplication of Equation 8.38 by the matrix \mathbf{B}^T now has the interesting effect that the Lagrange multipliers disappear since the product $\mathbf{D} \mathbf{B}$ vanishes due to the kinematic constraints, Equation 8.36, in which the relative joint coordinates \mathbf{w} are independent.

$$\mathbf{B}^T \mathbf{M} \mathbf{B} \dot{\mathbf{w}} = \mathbf{B}^T \mathbf{g} - \mathbf{B}^T \mathbf{M} \dot{\mathbf{B}} \mathbf{w} \quad (8.39)$$

Moreover, a symmetric mass matrix, $\mathcal{M} := \mathbf{B}^T \mathbf{M} \mathbf{B}$, results, which is generally densely populated. Their coefficients are strongly nonlinear with respect to the joint coordinates \mathbf{w} . The right-hand side of the equation of motion, Equation 8.39,

$$\mathbf{f} := \mathbf{B}^T \mathbf{g} - \mathbf{B}^T \mathbf{M} \dot{\mathbf{B}} \mathbf{w} \quad (8.40)$$

is strongly nonlinear with respect to the joint coordinates \mathbf{w} and their time derivatives $\dot{\mathbf{w}}$.

For systems without kinematic loops (open loop systems), the number of equations in Equation 8.41

$$\mathcal{M} \dot{\mathbf{w}} = \mathbf{f} \quad (8.41)$$

is equal to the number of degrees of freedom. For the numerical integration, it is of relevance that instead of the initial set of ODEs (Equation 8.35) and the accompanying set of algebraic constraints (Equation 8.36), the mathematical model now is a set of first order ODEs for \mathbf{w} and the vector \mathbf{q} of positions, where $\dot{\mathbf{q}} = \mathbf{w}$.

In [45], Nikravesh and Gim have shown that a modified form of the method is also applicable to multibody systems with closed kinematic loops. In this generalisation of the method, it is best to choose a *cut joint* in each kinematic loop so that subsystems with a tree-like structure result. Clearly, this method can be applied to each subsystem. If the kinematic loops are closed again, then the joint coordinates are no longer independent. Another transformation step, the so-called *closed loop velocity transformation*, can make the Lagrange multipliers disappear.

Eventually, the matrix \mathbf{B} in the velocity transformation, Equation 8.36, can be composed from sub-matrices of the joints in a systematic manner according to the topology of the entire system. The sub-matrix each joint contributes has a fixed structure that depends on the joint's type [44]. As is well known, each element in an electrical circuit contributes a stamp to the nodal admittance matrix. In a similar way, the matrix \mathbf{B} can be systematically constructed directly from the topology of a multibody system.

8.4.2 Reduction of the Equations of Motion: Transformation of I Stores in the Bond Graph

In the following, we will first show that the above reduction of equations of motion to a minimal number can be interpreted in bond graph terms as the transformation of inertias over the multiport MTF of the velocity transformation. For this purpose, the vector of velocities, \mathbf{v} , in Equation 8.35 is partitioned into a sub-vector \mathbf{v}_i of independent velocities and a sub-vector \mathbf{v}_d of dependent velocities. The mass matrix \mathbf{M} is partitioned accordingly.

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_d \end{bmatrix} \quad (8.42)$$

Let \mathbf{T} be a matrix relating both sub-vectors. That is, $\mathbf{v}_d = \mathbf{T} \mathbf{v}_i$ and let \mathbf{E} be the $\dim(\lambda) \times \dim(\lambda)$ unity matrix and $\mathbf{g} := [\mathbf{f} \ \mathbf{0}]^T$. Then, the equation of motion, Equation 8.35, and the kinematic constraints, Equation 8.36, can be written in the form

$$\begin{bmatrix} \mathbf{I}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_d \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_i \\ \dot{\mathbf{v}}_d \end{bmatrix} - \begin{bmatrix} -\mathbf{T}^T \\ \mathbf{E} \end{bmatrix} \lambda = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \quad (8.43)$$

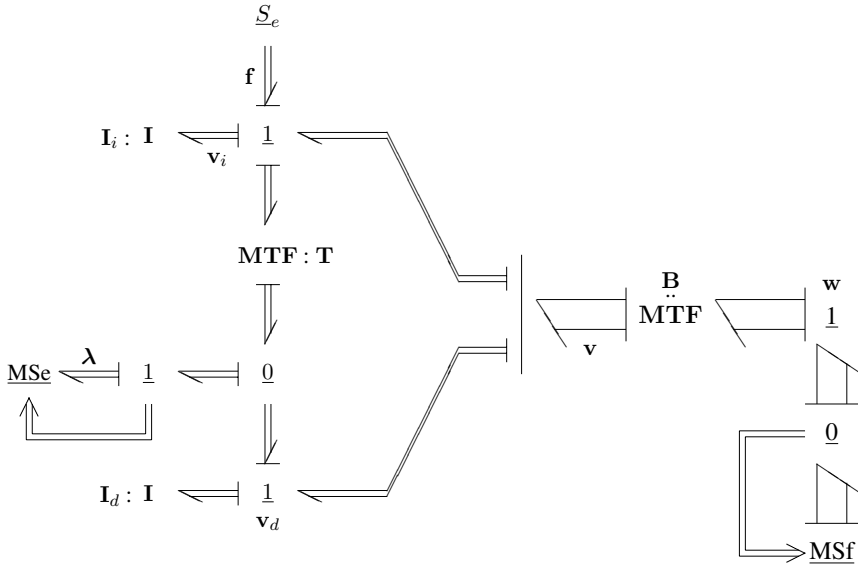


Fig. 8.25 General multibond graph of a multibody system with velocity transformations

$$[-\mathbf{T} \ \mathbf{E}] \begin{bmatrix} \mathbf{v}_i \\ \mathbf{v}_d \end{bmatrix} = \mathbf{0} . \tag{8.44}$$

The rewritten equations and the velocity transformation, Equation 8.37, can be represented by the multibond graph in Figure 8.25.

If we write the velocity transformation, Equation 8.37, in the form

$$\begin{bmatrix} \mathbf{v}_i \\ \mathbf{v}_d \end{bmatrix} = \begin{bmatrix} \mathbf{B}_i \\ \mathbf{B}_d \end{bmatrix} \mathbf{w} , \tag{8.45}$$

then the direct sum of multibonds and the transformer with the matrix \mathbf{B} can be replaced by two transformers with the matrices \mathbf{B}_i and \mathbf{B}_d . Now, if the inertia \mathbf{I}_i is transformed over the transformer with the matrix \mathbf{B}_i and if, likewise, \mathbf{I}_d is transformed over the MTF with matrix \mathbf{B}_d , then we get the multibond graph shown in Figure 8.26.

By summing the efforts at the lower array of 1-junctions representing the velocity \mathbf{v}_d (Figure 8.26), we get the equality

$$\mathbf{T} (\mathbf{B}_i \mathbf{w}) = \mathbf{B}_d \mathbf{w} . \tag{8.46}$$

If we write this equation in the form

$$- [-\mathbf{T} \ \mathbf{E}] \begin{bmatrix} \mathbf{B}_i \\ \mathbf{B}_d \end{bmatrix} \mathbf{w} = \mathbf{0} , \tag{8.47}$$

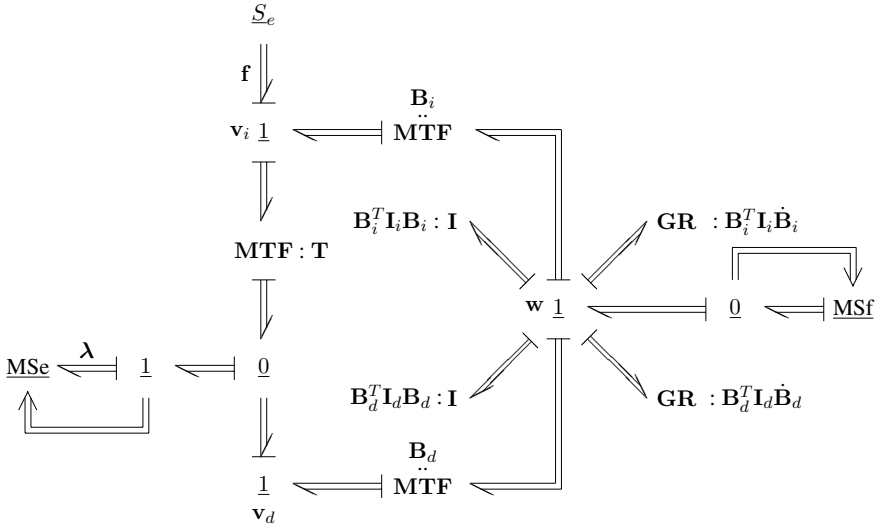


Fig. 8.26 Multibond graph after transformation of inertias over MTFs

then it is the constraint

$$(DB) \mathbf{w} = \mathbf{0} . \tag{8.48}$$

Since it has been assumed that the multibody system has no kinematic loops, the relative velocities, \mathbf{w} , are independent of each other. Thus, the product DB must vanish.

Summing the efforts at the right side array of 1-junctions representing the independent velocities \mathbf{w} yields

$$\mathbf{0} = (B_i^T I_i B_i) \dot{\mathbf{w}} + (B_i^T I_i \dot{B}_i) \mathbf{w} + (B_d^T I_d B_d) \dot{\mathbf{w}} + (B_d^T I_d \dot{B}_d) \mathbf{w} + B_i^T (T^T \lambda - \mathbf{f}) + B_d^T (-\lambda) . \tag{8.49}$$

By taking into account the partitioning of the mass matrix M , Equation 8.42, of the transformation matrix B , Equation 8.45, and of the vector $\mathbf{g} (= [\mathbf{f} \ \mathbf{0}]^T)$, and by observing that the product DB vanishes, we see that Equation 8.49 is just the equation of motion (8.39).

8.4.3 Deriving the Reduced Form of Equations of Motion from the Bond Graph

Once a multibond graph for a multibody system has been constructed, the equations of motion in reduced form can be derived from the bond graph in a procedure that is similar to the one proposed by F  lez, Vera and Cacho in [19].

- First, a controlled flow source is attached to each array of 1-junctions representing a relative velocity of a joint. To ensure that the model is not modified, the effort into the sources must be zero. Propagation of the causality of these sources results in derivative causalities. As a result, all velocities become dependent of the chosen relative velocities of the joints. In this way we get the velocity transformation matrix \mathbf{B} .
- Next, all added controlled flow sources are removed and controlled effort sources are inserted in their place into the joint models to account for constraint forces or torques. Assigning causalities leads to integral causalities at all inertia ports. With the multibond graph modified in this way, equations of motion can be derived in the un-reduced form (Equation 8.35). Thus, in particular, the term \mathbf{g} in this equation is known.
- Finally, if the velocity transformation is symbolically differentiated, then all matrices and vectors needed in the reduced form of equations of motion are available.

As Bos points out in [3], multiplication of matrices should not be performed symbolically because large expressions for the coefficients may result that will need a considerable amount of memory to be handled in formula manipulation programs. Moreover, numerical evaluation of these expressions is inefficient. It is true that some terms in these expressions will often cancel, but formula manipulation programs have limited capabilities to identify possible simplifications. In general, significant simplification can only be expected for symbolic expressions that are not too large. Requirements for memory can be reduced and the efficiency of expression evaluation can be improved by introducing intermediate variables. In doing this, however, the virtue of symbolic expressions gets somewhat lost because the meaning of an expression is obscured the more intermediate variables appear in it.

8.4.4 Application of the Procedure to a Planar Pendulum

In the following, the procedure will be illustrated by means of an example. To enable an easy review and to keep the computational effort reasonably small, we confine ourselves to the planar pendulum depicted in Figure 8.27. It can be viewed as a body b' that is composed of a massless rod of length L and a rigid body b , with mass m and with moment of inertia J with respect to the z -axis through its centre of mass. Body b' has one hinge point that is connected by a revolute joint to the inertial frame. The distance from the hinge point to the centre of mass is of length L . Again, the origin of the body fixed reference frame is assumed to be at the centre of mass.

In this example, body b' has only one hinge point. Its axis of rotation through the hinge point is perpendicular to the plane in which the body moves, which results in a simplified multibond graph model of a freely moving body (cf. Figure 8.16). The modulated gyrator disappears since the cross product $\boldsymbol{\Omega}' \times (\mathbf{J}'\boldsymbol{\Omega}') = (\boldsymbol{\Omega}'\mathbf{J}')\boldsymbol{\Omega}'$ vanishes. The model of the revolute joint between the inertial frame and the upper end of the rod reduces to a resistor that accounts for friction. Thus, the multibond

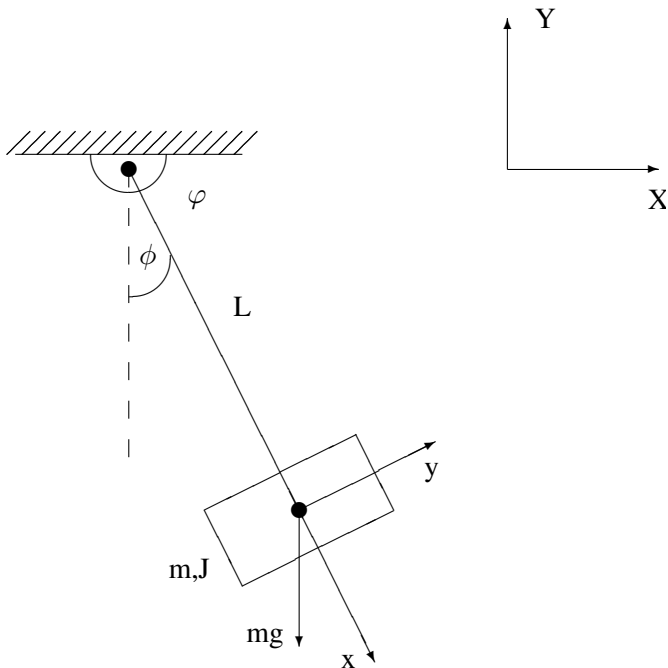


Fig. 8.27 Planar pendulum

graph of a freely moving body is easily adapted to the model of the planar pendulum depicted in Figure 8.28.

The revolute joint allows for only one degree of freedom. Consequently, according to the above procedure, we first insert a controlled source in the model of the joint that prescribes the relative angular velocity $\dot{\varphi}$. As a result of causality assignment, both inertias get derivative causality (Figure 8.29).

Observing the matrix **A**

$$\mathbf{A} = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} \tag{8.50}$$

of the MTF, the following velocity relations are easily obtained from the multibond graph of Figure 8.29.

$$\begin{bmatrix} \Omega \\ v_X \\ v_Y \end{bmatrix} = \begin{bmatrix} 1 \\ L \sin \varphi \\ -L \cos \varphi \end{bmatrix} [\dot{\varphi}] \tag{8.51}$$

Thus, the matrix **B** in the velocity transformation, Equation 8.37, is

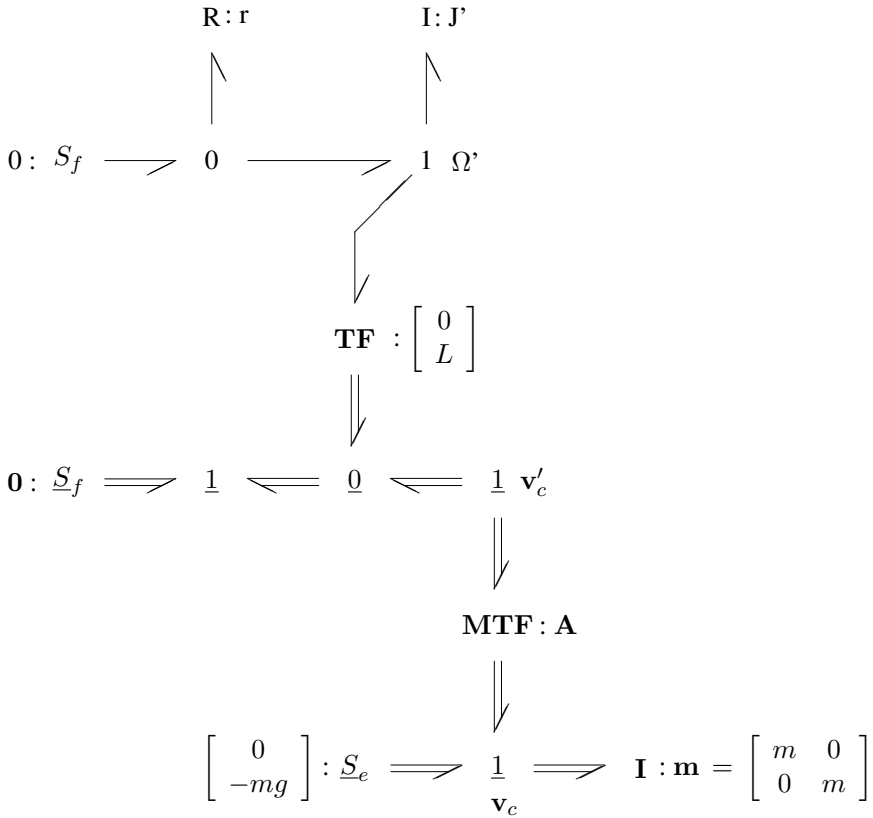


Fig. 8.28 Multibond graph of a planar pendulum

$$\mathbf{B} = \begin{bmatrix} 1 \\ L \sin \varphi \\ -L \cos \varphi \end{bmatrix}. \tag{8.52}$$

The next step is to determine the matrices and vectors needed in the equation of motion in the reduced form. To this end, we follow the above procedure and derive a second bond graph from the initial one in Figure 8.28 by adding controlled effort sources that account for the constraint forces in the revolute joint (The upper end of the rod has no translational velocity). Assigning causalities to this bond graph results in integral causality at the inertia ports (Figure 8.30).

Adding up the efforts at the upper 1-junction in the multibond graph of Figure 8.30, we get the torque balance

$$-r\Omega = J\dot{\Omega} + [0 \ L](-\lambda)$$

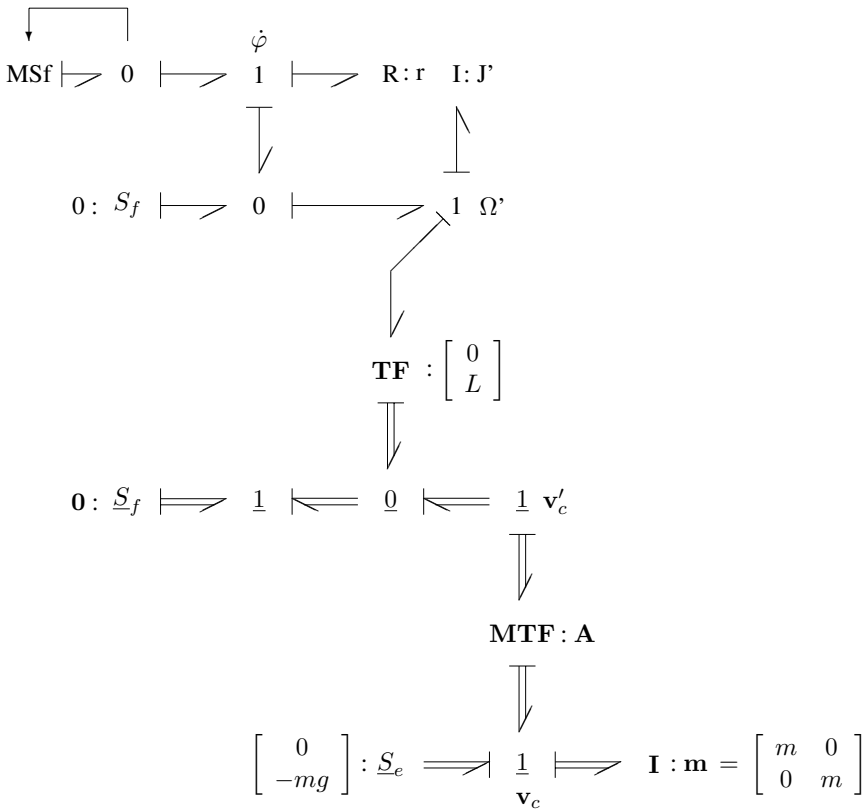


Fig. 8.29 Multibond graph for the derivation of the velocity transformation

$$\begin{aligned}
 &= J\dot{\Omega} + [0 \ L] \begin{bmatrix} -\lambda_x \\ -\lambda_y \end{bmatrix} \\
 &= J\dot{\Omega} - L\lambda_y.
 \end{aligned}
 \tag{8.53}$$

From the lower array of 1-junctions representing the velocity of the centre of mass, \mathbf{v}_c , we derive the balance of forces

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{m}\dot{\mathbf{v}}_c - \begin{bmatrix} 0 \\ -mg \end{bmatrix}.
 \tag{8.54}$$

Both balances can be combined into the form of Equation 8.35, with the matrices

$$\mathbf{M} = \begin{bmatrix} J & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}.
 \tag{8.55}$$

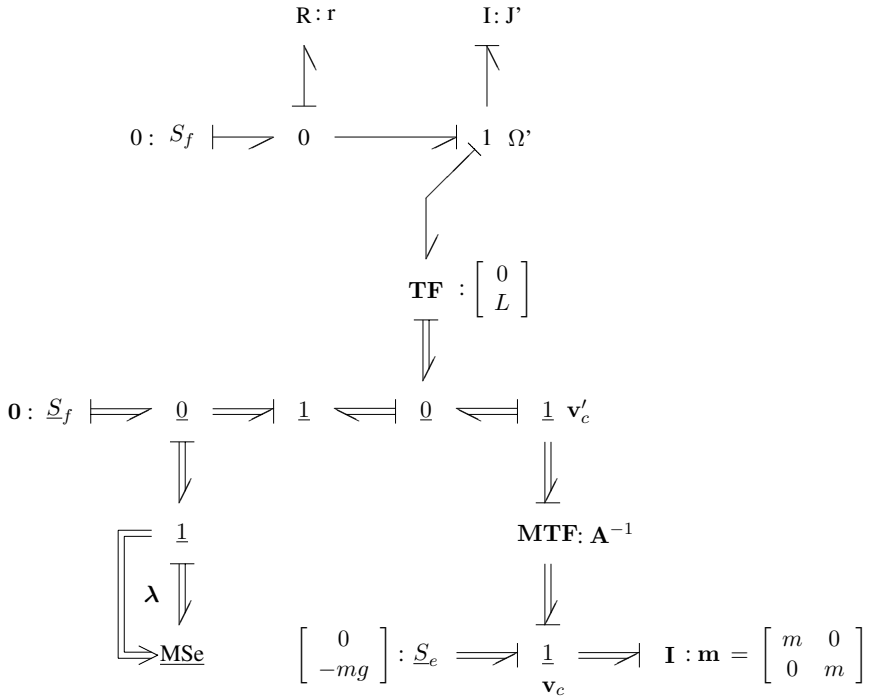


Fig. 8.30 Multibond graph of the planar pendulum accounting for constraint forces

and

$$\mathbf{D}^T = \begin{bmatrix} 0 & L \\ \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}, \tag{8.56}$$

and the vector $\mathbf{g} = [-r\Omega \quad 0 \quad -mg]^T$. By computing the matrix products in Equation 8.39, we get as expected,

$$\mathbf{B}^T \mathbf{D}^T = \mathbf{0} \tag{8.57}$$

$$\mathbf{B}^T \mathbf{M} \mathbf{B} = J + mL^2 \tag{8.58}$$

$$\mathbf{B}^T \mathbf{M} \dot{\mathbf{B}} = 0. \tag{8.59}$$

Thus, in the case of a planar pendulum, Equation 8.39 takes the well known form

$$(J + mL^2) \ddot{\varphi} + r\dot{\varphi} = mgL \cos \varphi \tag{8.60}$$

or

$$(J + mL^2) \ddot{\phi} + r\dot{\phi} = -mgL \sin \phi. \tag{8.61}$$

As a result, instead of three ODEs and two kinematic constraints according to the number of degrees of freedom, we get one second order ODE in the unknown φ (Lagrange equation of the second kind). By observing $\Omega' = \dot{\varphi}$, the second order ODE is easily transformed into a set of two first order ODEs.

8.5 Software for Modelling and Simulation of Multibody Systems

As in other disciplines, e.g., circuit simulation, there exist quite a number of special purpose programs for the simulation of multibody systems. For example, Adams^{®1} [46], NEWEUL [30] and MESA VERDE [57]. Also, there are some general purpose modelling and simulation programs, e.g., Dymola[®] [13], that can generate equations of motion by using model libraries for multibody systems. However, traditional special purpose programs for multibody systems and general purpose programs such as Dymola[®] have not been designed to support bond graph modelling. The latter methodology certainly has the advantage that models of subsystems from other domains, e.g., models for the motors driving the joints of robots, can be easily included in the modelling of the mechanical multibody system in a uniform manner. Software programs that support the development of multibond graph models for multibody system and that are able to process these models are BONDYN [18], already mentioned in Section 8.3, and the commercial general modelling and simulation software environment 20-sim[®]. For the planar motion of multibody systems, Marquis-Favre, Bideaux and Scavarda developed a bond graph library [36, 37] that is used by the commercial multi-domain modelling and simulation software AMESim or LMS Imagine.Lab AMESim^{®2} [32].

8.6 Conclusion

This chapter demonstrates how the bond graph methodology can support the systematic modelling of complex systems that are composed of rigid bodies and different types of joints. The extension of bond graphs representations with one-dimensional bonds into so-called multibond graphs has been presented.

Among the multiport extensions of the basic bond graph elements, the displacement modulated transformer is of particular importance for modelling the 3D motion of mechanical systems. It enables a compact description of the transformations between reference frames commonly used in the modelling of multibody systems. This is certainly an advantage since relations between generalised velocities simply result from differentiation of geometric constraints that are obtained by direct

¹ Adams is a trademark or registered trademark of MSC Software Corporation, 2 MacArthur Place, Santa Ana, CA 92707, USA, <http://www.mscsoftware.com>

² AMESim[®] is a registered trademark of LMS International, Research Park Z1, Interleuvenlaan 86, B-3001 Leuven, <http://lmsintl.com>

inspection of a system schematic. The importance of power conserving bond graph transformers has been pointed out in as early as 1969 by Karnopp [23]. It is interesting to note that in the modelling of non-mechanical systems, the modulated transformer does not play any similar prominent role, especially not in the modelling of electrical systems. This indicates some limitations in the analogy between mechanical and non-mechanical systems.

Another multiport bond graph element relevant for the modelling of multibody systems is the modulated gyration. It can be used to represent gyroscopic forces or torques that must be taken into account if the motion of a rigid body is described with respect to a moving reference frame. Since it is common in the modelling of multibody systems to use body fixed (moving) reference frames, modulated gyrators inevitably occur in corresponding multibond graphs. In bond graphs with one-dimensional bonds, the modulated multiport gyration used for representing Euler's equation of motion has the well known symmetric ring structure found by Karnopp and Rosenberg [27] and depicted in Figure 8.9.

Once a multibond graph model of a freely moving rigid body [3] and models for the different types of joints are available, multibond graph models of multibody systems can be built in a straightforward systematic manner. The corresponding mathematical model is a system of six ODEs for each body (Newton-Euler's equation of motion) and $6 - f_{J_i}$ additional kinematic constraints for the i^{th} joint, where f_{J_i} denotes its number of relative degrees of freedom. Due to the kinematic constraints, causal paths occur between inertia ports. Thus, the mathematical model is a DAE system of index 1.

It is common in mechanics to use Lagrange multipliers that take into account constraint forces introduced in the joints. This removes the dependencies between inertias. The resulting DAE system is of index 2. Although it can be solved by a code based on the BDF method, the disadvantage is that the number of equations with respect to the number of degrees of freedom is too high.

Another option is to relax constraints by assuming that joints are not a perfect rigid connection between rigid bodies, but show some elasticity and backlash in addition to friction. Clearly, by inserting C energy stores into joint models [22, 59], the inertias of the rigid bodies can be decoupled. That is, the kinematic constraints are replaced by ODEs. However, this increases the order of the model and the result is a system of stiff ODEs.

As an example of the approaches that aim at a formulation of equations of motion for a minimal number of generalised coordinates, we considered the well known joint coordinate method applied to systems with a tree-like structure. Nikravesh and Gim [45] show that the method is also applicable to systems with closed kinematic loops. In this chapter, it has been shown that the reduction in the number of equations of motion can be interpreted in bond graph terms as a transformation of inertias over MTFs. A three step procedure similar to the one proposed by Félez, Vera and Cacho [19] has been given for deriving equations of motion in reduced form from a multibond graph. It has been illustrated by the simple application of a planar pendulum.

References

- [1] R.R. Allen. Multiport Representation of Inertia Properties of Kinematic Mechanisms. *Journal of the Franklin Institute*, 308(3):235–253, 1979.
- [2] L.S. Bonderson. Vector bond graphs applied to one-dimensional distributed systems. *Journal of Dynamic Systems, Measurement and Control*, pages 75–82, 1975.
- [3] A.M. Bos. *Modelling Multibody Systems in Terms of Multibond Graphs with Application to a Motorcycle*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1986.
- [4] P.C. Breedveld. *Physical Systems Theory in Terms of Bond Graphs*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1984.
- [5] P.C. Breedveld. Multibond Graph Elements in Physical Systems Theory. *Journal of the Franklin Institute*, 319(1/2):1–36, 1985.
- [6] P.C. Breedveld. A Definition of the Multibond Graph Language. In S.G. Tzafestas and P. Borne, editors, *Complex and Distributed Systems: Analysis, Simulation and Control*, pages 69–72. Elsevier Science Publishers, 1986.
- [7] P.C. Breedveld. Insight in rigid body motion stability via an alternative for the eulerian junction structure. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 1999 International Conference on Bond Graph Modeling and Simulation (ICBGM'99)*, volume 31(1) of *Simulation Series*, pages 269–274. SCS, 1999.
- [8] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.
- [9] J.J. Craig. *Introduction to Robotics Mechanics & Control*. Pearson Prentice Hall, Upper Saddle, New Jersey, USA, 2005.
- [10] G. Dauphin-Tanguy, editor. *Les bond graphs*, chapter 4. Systèmes mécaniques multicorps. Systèmes automatisés. Hermès Science, Paris, 2000.
- [11] D. de Falco and E. Riviezzo. Bond graph modeling the longitudinal dynamics of motorcycles. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 1997 International Conference on Bond Graph Modeling and Simulation*, volume 29 (1), pages 274–279, 1997.
- [12] W. Drozd and H.B. Pacejka. Development and Validation of a Bond Graph Handling Model of an Automobile. *Journal of the Franklin Institute*, 328 (5/6):941–957, 1991.
- [13] H. Elmqvist, D. Brück, and M. Otter. *Dymola – User's Manual*. Dynasim AB, Lund, Sweden, 1996. URL <http://www.Dynasim.se>.
- [14] E.P. Fahrenthold and D.J. Wargo. Vector Bond Graph Analysis of Mechanical Systems. *Trans. ASME, Journal of Dynamic Systems, Measurement and Control*, 113:344–353, 1991.
- [15] A. Fakri and F. Rocaries. Bond graph components for modelling an electrically powered vehicle. In B. Zupančič, R. Karba, and S. Blažič, editors, *Proc. EUROSIM 2007*, 2007. ISBN 978-3-901608-32-2.
- [16] W. Favre. *Contribution à la représentation bond graph des systèmes mécaniques multicorps TOME 1*. PhD thesis, L'Institut National des Sciences Appliquées de Lyon, 1997.
- [17] W. Favre and S. Scavarda. Bond Graph Representation of Multibody Systems with Kinematic Loops. *Journal of the Franklin Institute*, 335B, No. 4:643–660, 1998.
- [18] J. Félez, C. Vera, I. San José, and R. Cacho. BONDYN: A Bond Graph Based Simulation Program for Multibody Systems. *Trans. ASME, Journal of Dynamic Systems, Measurement and Control*, 112:717–727, 1990.
- [19] J. Félez, C. Vera, and R. Cacho. Bond Graph Formulation in Terms of Relative Coordinates. In F.E. Cellier and J.J. Granda, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM'95*, pages 217–224. SCS Publishing, January 15-18 1995. Simulation Series, volume 27, no 1, ISBN: 1-56555-037-4.
- [20] D. Hrovat and W.E. Tobler. Bond graph modeling of automotive power trains. *Journal of the Franklin Institute*, 328 (5/6):623–662, 1991.
- [21] J.L. Baliño. Galerkin finite element method for incompressible thermofluid flows framed within the bond graph theory. *Simulation Modelling Practice and Theory*, 17 (1):35–49, 2009.

- [22] D. C. Karnopp and D. L. Margolis. Analysis and Simulation of Planar Mechanism Systems Using Bond Graphs. *Journal of Dynamic Systems, Measurement and Control*, 101:187–191, 1979.
- [23] D.C. Karnopp. Power-conserving Transformations: Physical Interpretations and Applications using Bond Graphs. *Journal of the Franklin Institute*, 288(3):175–201, 1969.
- [24] D.C. Karnopp. The Energetic Structure of Multibody Dynamic Systems. *Journal of the Franklin Institute*, 306(2):165–181, 1978.
- [25] D.C. Karnopp. Understanding Multibody Dynamics Using Bond Graphs. *Journal of the Franklin Institute*, 334B (4):631–642, 1997.
- [26] D.C. Karnopp. Dynamic modeling of electric vehicle performance using bond graphs. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 1997 International Conference on Bond Graph Modeling and Simulation (ICBGM '97)*, volume 29 (1), pages 261–266, 1997.
- [27] D.C. Karnopp and R.C. Rosenberg. *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics*. MIT Press, Cambridge, MA, 1968.
- [28] D.C. Karnopp and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1975.
- [29] S.A. Kayani and M.A. Malik. Modeling and simulation of biped kinematics using bond-graphs. In *Proc. International Conference on Emerging Technologies (ICET '06)*, pages 677 – 682, Peshawar, 2006. ISBN 1-4244-0502-5.
- [30] E. Kreuzer and W. Schielen. NEWEUL: Software for the Generation of Symbolic Equations of Motion. In W. Schielen, editor, *Multibody Systems Handbook*. Springer-Verlag, 1990.
- [31] J. Landaluze, C.F. Nicols, G. Urzelai, M. Calzada, and R. Reyro. Bond-graph aided modelling and simulation of a 2d-o-f experimental arm. In Guasch and Huber, editors, *Proc. of the SCS European Simulation Multiconference ESM '94*, pages 731–739, Barcelona, 1994.
- [32] LMS. URL <http://www.lmsintl.com/solutions>.
- [33] L.S. Louca, J.L. Stein, and D.G. Rideout. Generating Proper Integrated Dynamic Models for Vehicle Mobility Using a Bond Graph Formulation. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 2001 International Conference on Bond Graph Modeling and Simulation (ICBGM '01)*, volume 33 (1), pages 339–345, 2001.
- [34] D. L. Margolis and T. Shim. Instability due to interacting hydraulic and mechanical dynamics in backhoes. *Journal of Dynamic Systems, Measurement and Control*, 125 (3):497–504, 2003.
- [35] W. Marquis-Favre, E. Bideaux, O. Mechin, S. Scavarda, F. Guillemard, and M. Ebalard. Mechatronic bond graph modelling of an automotive vehicle. *Mathematical & Computer Modelling of Dynamical Systems*, 12 (2/3):189–202, 2006.
- [36] W. Marquis-Favre, E. Bideaux, and S. Scavarda. A planar mechanical library in the AMESim simulation software. Part I: Formulation of dynamics equations. *Simulation Modelling Practice and Theory*, 14(1):25–46, 2006.
- [37] W. Marquis-Favre, E. Bideaux, and S. Scavarda. A planar mechanical library in the AMESim simulation software. Part II: Library composition and illustrative example. *Simulation Modelling Practice and Theory*, 14(2):95–111, 2006.
- [38] L. Martinez, C. Vera, and J. Félez. Bond graph model for the analysis of the dynamic behavior of the human body. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 1997 International Conference on Bond Graph Modeling and Simulation (ICBGM '97)*, volume 29 (1), pages 295–300, 1997.
- [39] B. Maschke. *Contribution à une approche par bond-graph de l'étude et la conception de lois de commande de robots contenant des segments flexible*. PhD thesis, Université de Paris-Sud Centre d'Orsay, 1990.
- [40] J.M. Mera, C. Vera, J. Félez, and J.J. Esperilla. Influence of the Roll Axis Consideration in Vehicle Dynamics. Bond Graph Models. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2003 International Conference on Bond Graph Modelling and Simulation*, volume 35 (2), pages 207–213, 2003.

- [41] A. Mukherjee, P.M. Pathak, and A. Dasgupta. Self Balancing Two Legged Walking Robot. In F.E. Cellier and J.J. Granda, editors, *Proc. of the 2003 International Conference on Bond Graph Modeling and Simulation*, volume 35 (2), pages 182–187, 2003.
- [42] C. Niesner, G. Dauphin-Tanguy, D.L. Margolis, F. Guillemard, and M. Pengov. A 4 Wheel Vehicle Bond Graph Model Including Uncertainties on the Car Mass and the Centre of Mass Position. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation (ICBGM '05)*, volume 37 (1), pages 179–184, 2005.
- [43] P.E. Nikravesh. *Computer-Aided Analysis of Mechanical Systems*. Prentice-Hall, Englewood Cliffs, N.J., USA, 1988.
- [44] P.E. Nikravesh. Computational Methods in Multibody Systems. In *COMETT*, DTH Lyngby, May 27–31 1991.
- [45] P.E. Nikravesh and Gwanghun Gim. Systematic Construction of the Equation of Motion for Multibody Systems Containing Closed Kinematic Loops. *Transactions of the ASME, Journal of Mechanical Design*, 115:143–149, 1993.
- [46] N. Orlandea, M.A. Chace, and D.A. Calahan. A Sparsity-Oriented Approach to Dynamic Analysis and Design of Mechanical Systems Part I and II. *ASME Journal of Engineering for Industry Ser. B*, 99:773–784, 1977.
- [47] H.B. Pacejka. Tyre factors and vehicle handling. *International Journal of Vehicle Design*, 1: 1–23, 1979.
- [48] H.B. Pacejka and C.G.M. Tol. A bond-graph computer model to simulate the 3-d dynamic behaviour of a heavy truck. In *Proc. 10th IMACS World Congress on System Simulation and Scientific Computation*, volume 3, pages 398–401, Montreal, 1982.
- [49] P. M. Pathak, A. Mukherjee, and A. Dasgupta. Impedance control of space robots using passive degrees of freedom in controller domain., *Journal of Dynamic Systems, Measurement and Control*, 127:564–578, 2005.
- [50] P.M. Pathak, S. Kumar, and A. Mukherjee. Study of Inter-axis Coupling in Space Robot with Three Reaction Wheels as Attitude Controllers. In J. J. Granda and F. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation*, volume 37 (1), pages 199–205, 2005.
- [51] T. Periasamy, T. Asokan, and M. Singaperumal. Study on Coupled Dynamics of Underwater-Manipulator System: A Bond Graph Approach. In *Proc. of COPEN 2007*, pages 93–98, 2007.
- [52] G. Romero, J. Flez, J. Maroto, and M.L. Martinez. Simplified bond graph models for simulation of earth moving machines. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2007 International Conference on Bond Graph Modeling*, volume 39 (1), pages 139–147, 2007.
- [53] J.U. Thoma. Thermofluid Systems by Multi-bondgraphs. *Journal of the Franklin Institute*, 329(6):999–1009, 1992.
- [54] M.J.L. Tiernego and A.M. Bos. Modelling the dynamics and kinematics of mechanical systems with multibond graphs. *Journal of the Franklin Institute*, 319(1/2):pp. 37–50, 1985.
- [55] J. van Dijk and P.C. Breedveld. Simulation of System Models Containing Zero-order Causal Paths — II. Numerical Implications of Class 1 Zero-order Causal Paths. *Journal of the Franklin Inst.*, 328(5/6):981–1004, 1991.
- [56] J. Wittenburg. *Dynamics of Systems of Rigid Bodies*. Teubner-Verlag, Stuttgart, 1977.
- [57] J. Wittenburg and U. Wolz. Meza verde: A symbolic program for nonlinear articulated-rigid-body dynamics. In *Proc. 10th ASME Design Engineering Div. Conf. on Mechanical Vibration and Noise*, 1985. Cincinnati, Ohio, USA.
- [58] A. Yazman. *Modélisation des robots flexibles par les bond-graphs – application à l’analyse de leurs performance dynamics*. PhD thesis, Université de Paris-Sud, Centre d’Orsay, 1988.
- [59] A. Zeid. Bond Graph Modeling of Planar Mechanisms With Realistic Joint Effects. *Journal of Dynamic Systems, Measurement and Control*, 111:15–23, 1989.
- [60] A. Zeid and C. H. Chung. Bond Graph Modeling of Multibody Systems: a Library of Three-dimensional Joints. *Journal of the Franklin Institute*, 329 (4):605–636, 1992.

Chapter 9

Bond Graph Approximation of Distributed Parameter Models

In the previous chapter, bodies have been assumed rigid in the modelling of multi-body systems. However, there are flexible systems one encounters in every day life. For example, if we walk or drive across a long bridge, we might feel the vibrations corresponding to the deflections. Many mechanisms are not really rigid either. Other obvious examples are the chassis and the body of a vehicle or the robotic arm that moves a load from one place to another. Some parts of multibody systems experience elastic deformations leading to vibrations that cannot be neglected. Such vibrations can be noticed, for instance, when looking at the bending of the wings of a commercial airplane in flight. Clearly, the elasticity of a robot's arm affects its dynamics and the accuracy of positioning a tool or a load. Mechanical bodies not only have an inertia, but are also elastic to some degree. Both physical properties are continuously spatially distributed. This holds, for instance, for long hydraulic or acoustic lines.

So far, the spatial distribution of inertia and elasticity has been approximated in models by assuming that each property can be lumped in certain points in a body. This has given rise to the introduction of two distinct types of energy stores, viz. an I element and a C element. All bond graphs considered so far in this text represent so-called *lumped parameter models*. While this assumption can be justified for modelling many mechanical systems or hydraulic circuits without long lines, it is not sufficient for typical continua like the ones already mentioned. The dynamics in each point of a continuum are determined by the propagation of waves, their reflection at the boundaries of the continuum being part of an overall system and their superposition. For instance, if an electromagnetic valve connected to a hydraulic line closes in a very short period of time, then the local pressure change caused by this closing travels with the speed of sound through the fluid and is reflected at the ends of the transmission line. In contrast to previous bond graph modelling leading to ODEs (and algebraic constraints), continua are described by partial differential equations (PDEs). Since modelling takes into account that physical properties, e.g. inertia, compliance and friction are spatially distributed, corresponding models are called *distributed parameter models*. If we recall that the solution of the wave equation can be represented in the form of a series, we see that system components

accounted for by a distributed parameter model have an infinite number of degrees of freedom in contrast to systems represented by lumped parameter models.

For instance, a robot, apart from its arm, can be modelled as a system composed of rigid bodies while the arm is conceived as a continuum. In that case, a boundary value problem must be solved. However, in contrast to classical boundary value problems, the boundary conditions in this case are determined by the solution of the initial value problem for the rest of the robot. It is true that there exist special solvers for boundary value problems in certain domains, e.g., for fluid-mechanical problems or electromagnetic fields. Usually, these rely on finite element methods. On the other hand, there are robust general purpose solvers for DAE systems. However, a combination of a set of DAEs and a partial differential equation is not possible with many modelling and simulation programs. Thus, the question arises as to how a distributed parameter model of a system component, such as a beam, can be appropriately approximated by a lumped parameter model.

9.1 Approximation of a One-dimensional Distributed Parameter Model by an Oscillator Chain

A well known and obvious approach to the development of spatially one-dimensional models is to approximate a continuum by a chain of segments in which one lump is assumed to be an ideal rigid body, while its neighbour has no inertia but is ideally compliant. Using this clear approach, the ordinary differential equations for a rigid segment and its completely compliant neighbour turn into the initial partial differential equations for the continuum if the thickness of the segments tend to zero. However, there is a serious disadvantage. If the number of segments is increased, accuracy does not increase likewise. Within given accuracy boundaries, unfortunately, it cannot easily be determined how many lumps will be needed to reproduce the dynamic behaviour determined by low natural frequencies. If we disregard cases with low accuracy requirements, then generally, a high number of segments will be necessary, leading to an ODE system of high order. Although this will mean considerable computational effort, higher natural frequency modes are captured rather inaccurately (Of course, if the thickness of the segments tended to zero, then in the limit, there would be an infinite number of degrees of freedom).

This chapter demonstrates that bond graph modelling can also be used to develop lumped parameter models that appropriately approximate distributed parameter models of continua.

Components, e.g., rods or bars with longitudinal vibrations or hydraulic lines, clearly can be modelled approximately by a chain of lumps in which rigid and fully compliant segments of thickness Δx alternate. A generic bond graph representation of such spatial one-dimensional chains of oscillators is shown in Figure 9.1.

For low accuracy requirements, typically, only short oscillator chains are used. Clearly, causal boundary conditions are imposed by the submodels that connect to

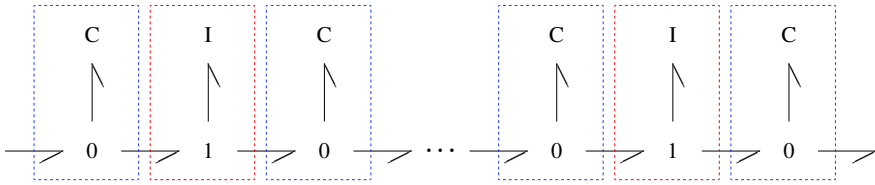


Fig. 9.1 Generic bond graph representation of a one-dimensional chain of oscillators

the lumped parameter approximation of the continuum. They decide which storage elements are admissible at the ports of the coarse continuum model (Figure 9.2).

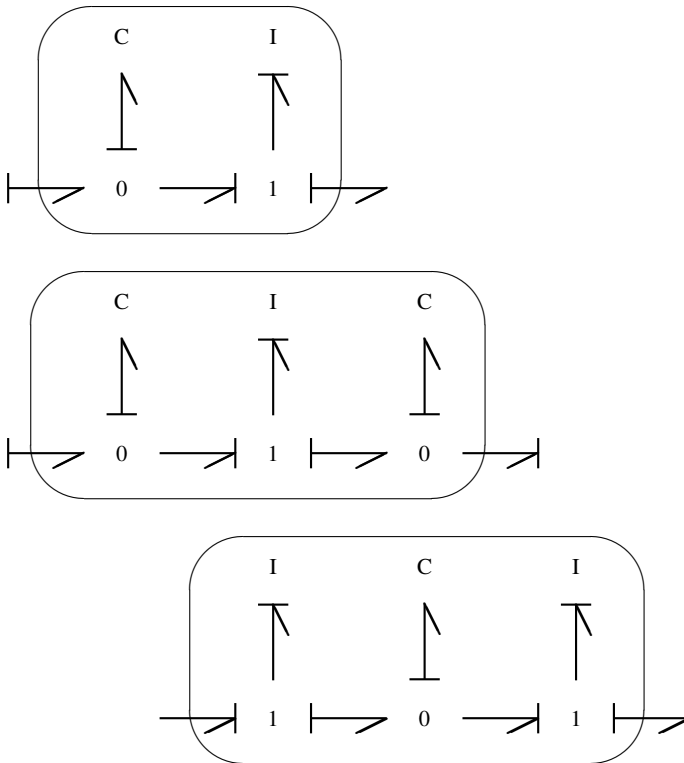


Fig. 9.2 Low order continuum models with different causalities at their ports

9.2 Brief Survey of Bond Graph Approximations of Distributed Parameter Models

Further approaches to lumped parameter bond graph approximations of distributed parameter models have been reported in the literature. Whereas Bonderson proposed an approach based on finite differences [3] in 1975, modal analysis has been used by Karnopp [18], Margolis [22] and Lebrun [20]. Elramady and Granda used modal analysis to develop a bond graph model for the flexible solar panels of a module of the International Space Station (ISS) [11] by considering the photovoltaic array panels as Bernoulli-Euler beams (cf. Section 9.3.1). The ISS is clearly a complex multibody assembly of bodies that can be assumed to be rigid while others must be considered flexible.

Granda and Kong start from finite element methods in order to model heat conduction in one and two dimensions by means of bond graphs [12, 13]. In [29], Pelegay, Doblaré and Buil demonstrate how bond graph based modelling can be combined with the finite element method. Further work combining bond graph modelling and the finite element method can be found in [9, 10, 25, 30]. Baliño considers computational fluid dynamics problems in a bond graph framework [15–17]

Damić [6, 7] and Damić and Čohodar [8] have presented an approach to bond graph modelling of flexible multi-bodies that is based on co-rotational formulation [5]. Finally, in [31, 32], Čohodar, Borutzky and Damić compare the finite element co-rotational formulation to the finite element absolute nodal coordinate formulation [2] in a bond graph framework. Applications to the well known planar flexible pendulum and the slider crank mechanism indicate that in these cases, the co-rotational formulation results in computationally less costly simulation runs.

The following two sections illustrate how modal analysis as well as the finite element method can be used to develop lumped parameter approximations of distributed parameter models of system components to be used in a bond graph of an overall multidisciplinary system.

9.3 Modal Analysis

9.3.1 *The Bernoulli-Euler Beam*

A generic bond graph approximation of one-dimensional distributed parameter models can be developed using modal analysis. Focussing on fundamental aspects, let us consider the classical example of a beam assuming the Bernoulli hypothesis that rotary inertia and shear deformation can be neglected (Bernoulli-Euler beam). Let us also assume that only transversal forces act on the beam. A more comprehensive presentation that also covers the so-called Timoshenko beam may be found, e.g., in [19, 23, 27].

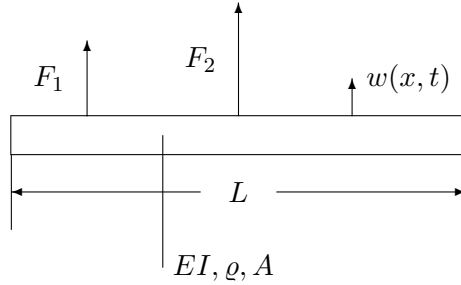


Fig. 9.3 Uniform Bernoulli-Euler beam

If there are two forces acting on the beam, as shown in Figure 9.3, then the transverse displacement, $w(x, t)$, at position x and time t is given by the partial differential equation

$$EI \frac{\partial^4 w}{\partial x^4} + \rho A \frac{\partial^2 w}{\partial t^2} = F_1 \delta(x - x_1) + F_2 \delta(x - x_2) \tag{9.1}$$

and boundary conditions at both ends. In Equation 9.1, the product of Young’s modulus, E , and the area moment of inertia, I , represent the bending stiffness of the beam. It is assumed to be constant. Furthermore, we assume that the cross sectional area A and the mass density ρ are constant along the beam. The expression $F_i \delta(x - x_i)$ using Dirac’s generalised delta function represents the force acting at position x_i ($i = 1, 2$).

The classical modal analysis assumes that two functions, Y and η , exist such that the solution of Equation 9.1 can be written in the form

$$w(x, t) = Y(x) \times \eta(t) . \tag{9.2}$$

This approach is called the *separation of variables* and has been well known for a long time. It leads to an ordinary differential equation (ODE) independent of the time t for the function Y .

$$EI \frac{d^4 Y}{dx^4} - \rho A \omega^2 Y = 0 \tag{9.3}$$

The assumption that the solution of Equation 9.3 is an exponential function together with the boundary conditions result in equations for the mode frequencies ω_n and for the corresponding so-called *mode shapes* or *normal modes*. That is, the natural frequencies as well as the mode shape functions depend on the actual boundary conditions. Strictly speaking, whenever the latter change, the mode frequencies and the mode shape functions must be re-evaluated. However, as Karnopp has shown in [18], the solutions in the case of force-free boundary conditions, i.e., there are no shear forces and no moments at the boundaries

$$w_{xx}(0, t) = 0 \quad (9.4a)$$

$$w_{xx}(L, t) = 0 \quad (9.4b)$$

$$w_{xxx}(0, t) = 0 \quad (9.4c)$$

$$w_{xxx}(L, t) = 0 \quad (9.4d)$$

(*free-free modes*) can be used to approximate the solution with sufficient accuracy in the case of other boundary conditions. Therefore, in the following, ω_n and Y_n always mean the mode frequencies and the mode shapes for the case of force-free boundary conditions. In any case, the mode shapes are orthogonal in the sense that

$$\int_0^L Y_n(x) Y_m(x) dx = 0 \quad (9.5)$$

for $n, m \in \mathbb{N}$ and $n \neq m$. Once the mode shape functions have been determined, the forced solution of Equation 9.1 is assumed to have the form

$$w(x, t) = \sum_{\nu=0}^{\infty} Y_{\nu}(x) \eta_{\nu}(t) . \quad (9.6)$$

If this series is inserted into Equation 9.1, then multiplication by Y_{μ} and subsequent integration along the beam (observing the orthogonality of mode shape functions) yields the non-homogenous second order ODE

$$m_{\mu} \ddot{\eta}_{\mu} + m_{\mu} \omega_{\mu}^2 \eta_{\mu} = Y_{\mu}(x_1) \times F_1 + Y_{\mu}(x_2) \times F_2 \quad (9.7)$$

for the function η_{μ} ($\mu = 0, 1, 2, \dots$), where m_{μ} denotes the modal mass,

$$m_{\mu} := \rho A \int_0^L Y_{\mu}^2(x) dx , \quad (9.8)$$

and $m_{\mu} \omega_{\mu}^2$ the modal stiffness. Its reciprocal may be interpreted as a modal compliance C_{μ} .

$$\frac{1}{C_{\mu}} := m_{\mu} \omega_{\mu}^2 \quad (9.9)$$

With the mode shape functions Y_{ν} and the time dependent functions η_{ν} , differentiation of Equation 9.6 with respect to time yields for the bending velocities at positions x_i ($i = 1, 2$)

$$\dot{w}(x_i, t) = \sum_{\nu=0}^{\infty} Y_{\nu}(x_i) \dot{\eta}_{\nu}(t) . \quad (9.10)$$

Hence, the instantaneous power at positions x_i ($i = 1, 2$) and time t is

$$F(x_i) \dot{w}(x_i, t) = \sum_{\nu=0}^{\infty} [Y_{\nu}(x_i) F(x_i)] \dot{\eta}_{\nu}(t) . \quad (9.11)$$

mation of forces at the 1-junctions yields the decoupled second order ODEs (9.7) for the modal oscillators. Conversely, the transformers supply modal velocities $\dot{\eta}_\mu$ weighted by the mode shapes $Y_\mu(x_i)$ ($i = 1, 2$). To keep the bond graph clear, transformer moduli are not depicted. The summation of weighed modal velocities at the 0-junctions yields the bending velocities $\dot{w}(x_i, t)$ at positions x_i ($i = 1, 2$) given by Equation 9.10. Since the mode shapes $Y_\mu(x_i)$ ($i = 1, 2$) only depend on fixed positions x_i , the transformer moduli are constant.

9.3.3 State Space Approximation

It is true that the modal analysis provides an infinite number of mode frequencies ω_μ ($\mu = 2, \dots$). In practice however, only a sufficiently small number is used so that a *low* order state space model results. If n modal oscillations are taken into account, then all modal masses may be combined into a diagonal matrix \mathcal{M} , all modal stiffnesses into a diagonal matrix \mathcal{K} , and all mode shapes $Y_{ij} := Y_i(x_j)$ into a matrix $\mathbf{Y} = (Y_{ij})$. Using these matrices and the vectors $\dot{\boldsymbol{\eta}} = [\dot{\eta}_1 \dot{\eta}_2 \dots \dot{\eta}_n]^T$, $\mathbf{F} = [F_1 F_2]^T$ and $\dot{\mathbf{w}} = [\dot{w}_1 \dot{w}_2]^T$, the equations of the bond graph in Figure 9.4 may be approximated and the result formulated in compact form.

$$\mathcal{M}\dot{\boldsymbol{\eta}} + \mathcal{K}\boldsymbol{\eta} = \mathbf{Y}\mathbf{F} \quad (9.14a)$$

$$\dot{\mathbf{w}} = \mathbf{Y}^T \dot{\boldsymbol{\eta}} \quad (9.14b)$$

The second order ODE for the modal oscillators, Equation 9.14a, and the velocity transformation, Equation 9.14b, may be represented by the multibond graph in Figure 9.5.

By introducing a vector of all momenta, $\mathbf{p} := \mathcal{M}\dot{\boldsymbol{\eta}}$, and renaming the modal displacements, $\mathbf{q} := \boldsymbol{\eta}$, we get a state space model

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{0} & -\mathcal{K} \\ \hline \mathcal{M}^{-1} & \mathbf{0} \end{array} \right] \cdot \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} \mathbf{Y} \\ \mathbf{0} \end{bmatrix} [\mathbf{F}] \quad (9.15)$$

that approximates a distributed parameter model. If the modal displacement $\boldsymbol{\eta}$ is eliminated in Equations 9.14a–9.14b, then we obtain a *macro-model* of the distributed parameter subsystem relating the external forces acting on the beam and the corresponding bending velocities $\dot{w}(x_i, t)$ at these positions.

$$\mathbf{M}\ddot{\mathbf{w}} + \mathbf{K}\mathbf{w} = \mathbf{F} \quad (9.16)$$

The matrices \mathbf{M} and \mathbf{K} in this macro-model are given by the transformation

$$\mathbf{M} = \mathbf{Y}^{-1} \mathcal{M} (\mathbf{Y}^{-1})^T \quad (9.17a)$$

$$\mathbf{K} = \mathbf{Y}^{-1} \mathcal{K} (\mathbf{Y}^{-1})^T \quad (9.17b)$$

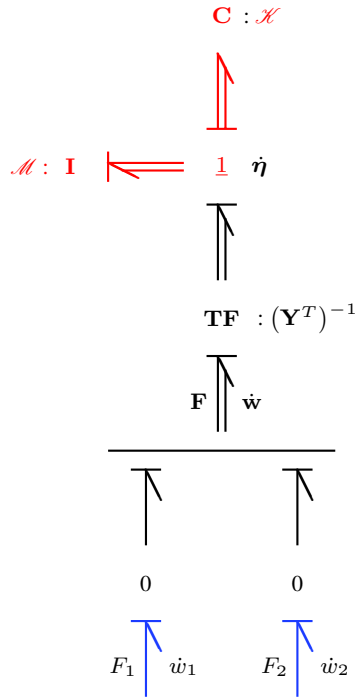


Fig. 9.5 Modal bond graph of a distributed parameter model in multibond graph representation

9.3.4 Features of the Generic Modal Beam Bond Graph Model

The modal analysis approach presented above is characterised by the following features.

- It is true that the modal bond graph model in Figure 9.4 has been developed under some simplifying assumption for the example of a uniform beam. The structure of the bond graph, however, remains the same for other linear distributed parameter models, even for planar and 3D-models, although the determination of mode shapes, mode frequencies and modal masses may become quite difficult as Karnopp, Margolis and Rosenberg have observed [19].
- The mode shapes and the modal masses may be numerically approximated. However, numerical approximations of the mode shapes, $Y_n, n \in \mathbb{N}$, should retain the fact that mode shapes are orthogonal. (cf. Equation 9.5).
- If there are more than two forces acting in different positions on the distributed parameter system, then additional corresponding 0-junctions can be added to the bond graph in Figure 9.4. They must be connected via transformers to all 1-junctions of the modal velocities. Moreover, the model can be extended to ac-

- count for external moments acting on a distributed parameter subsystem. In the case of an external moment, the modulus of the transformer converting it into a modal force is not a mode shape $Y(x)$, but the derivative $dY(x)/dx$ [21].
- External forces and moments are determined by the subsystems that the distributed parameter system is connected to. In other words, this means that a lumped parameter model can be established for an overall system including subsystems that must be considered distributed parameter systems. For the latter, the modal bond graph model of Figure 9.4 is used as a generic building block. The 0-junctions in this model represent the connecting points of the distributed parameter subsystem to the other subsystems. In particular, this means that by repeated use of the Bernoulli-Euler beam model, it is possible to set up models for flexible mechanical structures.
 - Finally, it must be pointed out that in contrast to a simple concatenation of spring-mass pairs considered in the beginning of this chapter, the modal analysis approach enables one to develop accurate low order models. For instance, in [22], Margolis reports that the finite difference method applied to the problem of the interaction between a high speed vehicle and an elevated roadway has resulted in ten times more equations compared to an approximation based on modal analysis, while the numerical results practically cannot be distinguished.

9.3.5 Further Aspects of the Generic Modal Beam Bond Graph Model

The modal analysis approach to a representation of distributed parameter models by bond graphs considered thus far still leaves some questions to be addressed.

The Number of Modal Oscillators

As previously mentioned, in practice, a certain number of the theoretically infinite number of modal frequencies is used. The question is how many modes must be taken into account in order to capture the characteristics of the dynamic behaviour and to obtain sufficiently accurate simulation results. As a rule of thumb, Margolis suggests to take into account all modal frequencies up to a frequency that is twice the highest frequency of interest in the model of the overall system ([19], p. 398ff). However, it is not possible to compute the natural frequencies that determine the dynamic behaviour of a system prior to a simulation run. Consequently, engineering experience is required to limit the frequency range of interest and to decide up to which frequency modal frequencies are to be taken into account. For instance, in a study of the interaction between a high speed vehicle and an elevated guideway presented by Margolis in [21], the guideway is considered to be composed of connected uniform Bernoulli-Euler beams. Simulation results of that study showed that it was sufficient to retain five modes in the generic modal beam model.

Accounting for Damping

In general, distributed parameter systems are lightly damped. Of course, the loss of free energy is also spatially distributed. However, it is generally not easy to add proper damping terms to the partial differential equation of the subsystem. Moreover, the well known separation-of-variables approach, in general, is only applicable if there is no damping term in the differential equation. In practice, useful results can be obtained by adding a linear resistor $R_\mu = 2D_\mu\omega_\mu m_\mu$ to the two energy storage elements of a modal oscillator connected to a 1-junction where the value of the mode damping ratio, D_μ , may be guessed based on experimental data.

Velocity Inputs into the Generic Modal Beam Bond Graph Model

Regarding causalities, one can see from the bond graph of Figure 9.4 that forces or moments imposed on the modal beam model lead to integral causality at all storage ports. However, if at one point a subsystem imposes a velocity on the distributed parameter model, then conversely, one of the modal inertias must accept derivative causality. Consequently, a causal path arises between this modal inertia and each one of the other modal inertia in integral causality. All these causal paths have in common the 0-junction at which the external velocity is imposed. That is, the velocity of the modal inertia in derivative causality depends on all other modal velocities and the external velocity imposed at the 0-junction (Figure 9.6). In this case, the mathematical model is of the form of a DAE system of differential index one.

Derivative causalities at modal inertias due to velocity inputs can be removed by the use of modal compliances C_μ ($\mu > n$) without their accompanying modal inertia in addition to the n modal oscillators with natural frequencies ω_ν , ($\nu = 1, 2, \dots, n$), [22]. An example with one velocity input is shown in Figure 9.7. In this way, it is

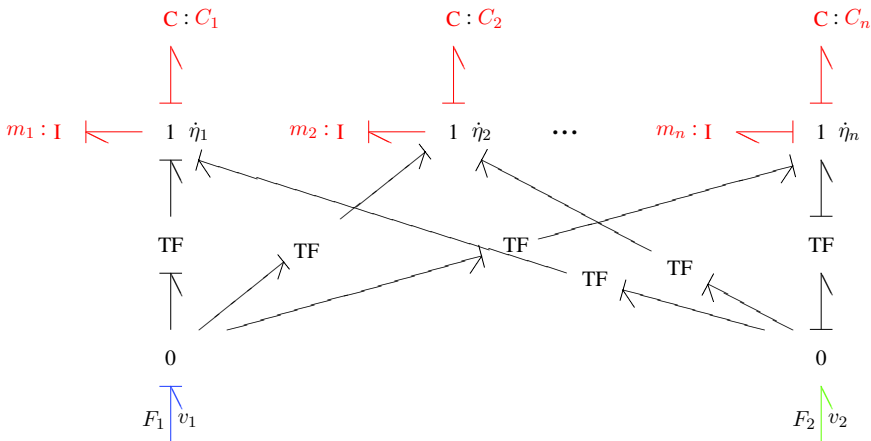


Fig. 9.6 Modal bond graph with n modal oscillators and a velocity input at location x_2

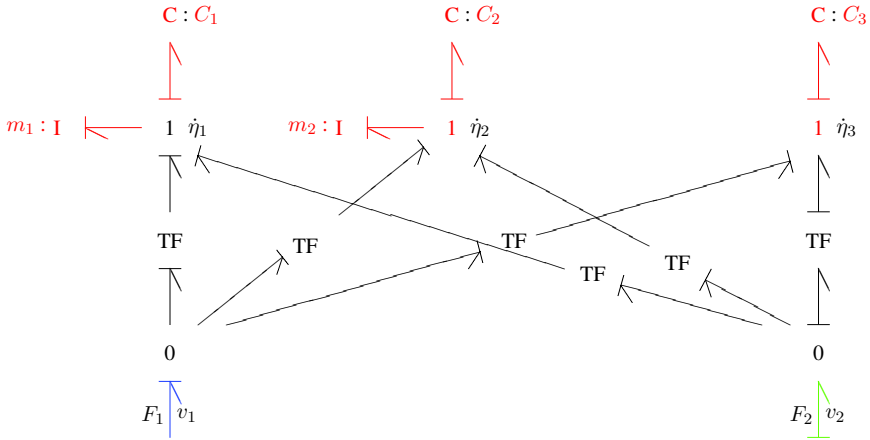


Fig. 9.7 Modal bond graph with two modal oscillators, one velocity input v_2 and an additional modal compliance C_3

possible to assign integral causality to all energy stores. Hence, the mathematical model approximating the distributed parameter model is of the form of an explicit state space model. Since the number, n , of modal frequencies has been chosen such that the highest natural frequency is at least twice the highest frequency that is of interest in the model of the overall system, further modal oscillations are stimulated far below their natural frequency. Consequently, accounting for further modal oscillators does not result in a significant improvement of simulation results. That is, they can be omitted. If the C energy stores of further oscillators are included in the model, then the dynamic behaviour determined by low frequencies is improved.

Significance of Forced High Frequency Modal Oscillations

For illustration of the above considerations, we assume that the μ^{th} modal oscillator is forced by a harmonic input and performs weakly damped oscillations described by an equation

$$\ddot{\eta}_\mu + 2D_\mu\omega_\mu\dot{\eta}_\mu + \omega_\mu^2\eta_\mu = \frac{1}{m_\mu}Y_\mu F, \tag{9.18}$$

where D_μ is the damping ratio and F the harmonic input of excitation frequency Ω ($F = F_0 \sin \Omega t$). If Ω is in the frequency range of interest and if the natural frequency of the undamped μ^{th} modal oscillator, ω_μ , is far above this range, i.e., $\Omega \ll \omega_\mu$, then the amplitude $\eta_{\mu 0}$ of the forced oscillation (of frequency $\omega_\mu \times \sqrt{1 - D_\mu^2}$) is

$$\eta_{\mu 0} = \frac{1}{m_{\mu}} Y_{\mu} F_0 \times \frac{1}{\sqrt{(\omega_{\mu}^2 - \Omega^2)^2 + (2D_{\mu} \omega_{\mu} \Omega)^2}} \tag{9.19a}$$

$$\approx \frac{1}{m_{\mu}} Y_{\mu} F_0 \times \frac{1}{\omega_{\mu}^2}, \tag{9.19b}$$

while the maximum value at resonance ($\Omega = \omega_{\mu} \times \sqrt{1 - 2D_{\mu}^2}$) is

$$\eta_{\mu \max} = \frac{1}{m_{\mu}} Y_{\mu} F_0 \times \frac{1}{\omega_{\mu}^2} \times \frac{1}{2D_{\mu} \sqrt{1 - D_{\mu}^2}}. \tag{9.20}$$

For $D_{\mu} \ll 1$, i.e., for weak damping, the amplitude of the forced modal oscillation is only a fraction of the maximum value at resonance.

$$\eta_{\mu 0} \approx 2D_{\mu} \eta_{\mu \max} \tag{9.21}$$

Combining Additional Modal Compliances into a C-Field

If from all modal oscillators with a natural frequency above a certain value only the C energy stores are retained, then the latter can be combined into a two-port C element between the two 0-junctions. Since this C field must accept derivative causality at its ports, it contributes to the bending velocities at the 0-junctions of the two point with force inputs.

The equations of the C field in Figure 9.8 can be obtained in the following manner. For each additional C energy store without accompanying inertia, Equation 9.7 reduces to

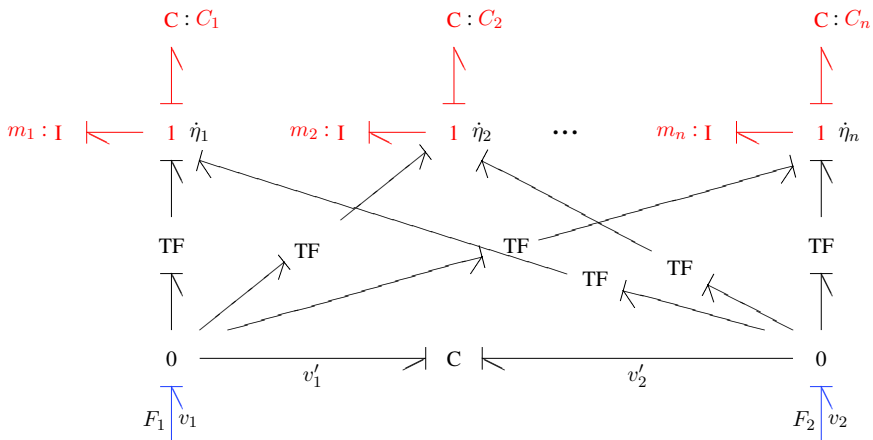


Fig. 9.8 Modal bond graph with n modal oscillators and a C field

$$\frac{1}{C_\mu} \eta_\mu = Y_{\mu 1} \times F_1 + Y_{\mu 2} \times F_2, \quad (9.22)$$

where $Y_{\mu 1} := Y_\mu(x_1)$ and $Y_{\mu 2} := Y_\mu(x_2)$. If we differentiate this sum of forces with respect to time and insert the result into Equation 9.10 for the velocities at locations x_i , $v_i = \dot{w}(x_i, t)$ ($i = 1, 2$), then we get

$$v_i = \sum_{\nu=0}^n Y_{\nu i} \dot{\eta}_\nu + \sum_{\nu=n+1}^{\infty} Y_{\nu i} C_\nu [Y_{\nu 1} \dot{F}_1 + Y_{\nu 2} \dot{F}_2]. \quad (9.23)$$

Hence, the constitutive equation of the C field reads

$$\begin{bmatrix} v'_1 \\ v'_2 \end{bmatrix} = \underbrace{\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \dot{F}_1 \\ \dot{F}_2 \end{bmatrix}, \quad (9.24)$$

where

$$c_{11} = \sum_{\nu=n+1}^{\infty} C_\nu Y_{\nu 1}^2 \quad (9.25a)$$

$$c_{22} = \sum_{\nu=n+1}^{\infty} C_\nu Y_{\nu 2}^2 \quad (9.25b)$$

$$c_{12} = c_{21} = \sum_{\nu=n+1}^{\infty} C_\nu Y_{\nu 1} Y_{\nu 2} \quad (9.25c)$$

(The symmetry of the matrix \mathbf{C} indicates that the C field is energy conservative). Thus, taking into account only the C energy store of modal oscillators with a natural frequency far above the frequency range of interest, derivative causality at the inertia of a modal oscillator can be removed without affecting the dynamic behaviour.

Algebraic Loops in the Generic Bond Graph Beam Model

If there are two or more velocity inputs, then this method yields algebraic loops (Definition 4.3) in the junction structure of the modal model (cf. Figure 9.9).

In the bond graph of Figure 9.9, bonds 1 – 8 build a causal loop (Definition 3.10) associated with two opposite oriented algebraic loops. Let Y_{ij} be the i^{th} mode shape, ($i = 1, \dots, 4$), at location x_j , ($j = 1, 2$). Then by summing up the flows at the two 0-junctions, we get a set of two linear algebraic equations for the additional modal velocities $\dot{\eta}_3, \dot{\eta}_4$

$$\dot{\eta}_3 = \frac{1}{Y_{31}} (v_1 - Y_{11} \dot{\eta}_1 - Y_{21} \dot{\eta}_2 - Y_{41} \dot{\eta}_4) \quad (9.26a)$$

$$\dot{\eta}_4 = \frac{1}{Y_{42}} (v_2 - Y_{32} \dot{\eta}_3 - Y_{22} \dot{\eta}_2 - Y_{12} \dot{\eta}_1). \quad (9.26b)$$

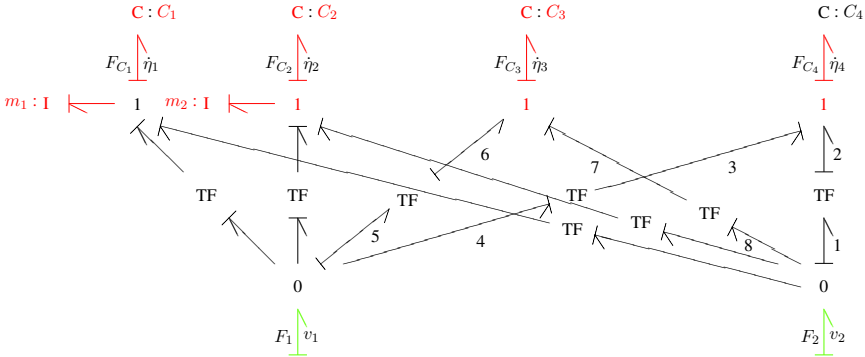


Fig. 9.9 Bond graph with two modal oscillators, two velocity inputs and an algebraic loop

The determinant of the set of equations

$$\Delta = 1 - \frac{Y_{32}}{Y_{42}} \times \frac{Y_{41}}{Y_{31}} \tag{9.27}$$

is different from zero since $Y_{i1} \neq Y_{i2}$, ($i = 3, 4$). Thus, the additional modal velocities $\dot{\eta}_3, \dot{\eta}_4$ may be expressed in terms of the external velocities v_1, v_2 and the modal velocities $\dot{\eta}_1, \dot{\eta}_2$. As a result, a state space model can be set up. The components of the state vector are outputs of the energy storage elements in integral causality, i.e. the modal velocities $\dot{\eta}_1, \dot{\eta}_2$ and the modal spring forces F_{C_1}, F_{C_2} .

9.3.6 Flexible Mechanical Structures

We conclude these considerations of the modal analysis from a bond graph modelling point of view with some remarks on how to proceed in the case of more complex flexible structures. The above given list of features characterising the modal analysis approach mentions that the modal bond graph of the Bernoulli-Euler beam can be considered a generic model. It can be used repeatedly for substructures in order to build a model of a more complex flexible structure. In fact, if we partition an overall mechanical structure into substructures and use the generic model for each substructure, then the modal models can be connected by C energy stores. This approach has been used, e.g., by Margolis [21] as well as by Buil and Pelegay [4]. In order to ensure that there is no relative motion between the substructures, the springs between them must be sufficiently stiff. An advantage of coupling the substructures by springs is that the modal models are excited by external efforts. Hence, all energy stores in the bond graph model of the entire flexible structure have integral causality if there is no velocity input otherwise. That is, after linearisation, if necessary, a state space model of the form

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{A}_{pq} \\ \mathbf{A}_{qp} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{e} \quad (9.28)$$

can be set up for the entire flexible structure or for the entire system if no friction has to be taken into account and if the system is externally excited by efforts only. In this equation, the vector \mathbf{p} combines the momenta of all inertias while the displacements of all C energy stores are grouped into the vector \mathbf{q} . Input efforts are combined into the vector \mathbf{e} . If the parameters of all inertias are grouped into the diagonal matrix \mathbf{M} , then the state space model can be reduced to the form

$$\mathbf{M}\ddot{\boldsymbol{\eta}} + \mathbf{K}\boldsymbol{\eta} = \mathbf{B}\mathbf{e} \quad (9.29)$$

by means of the transformation $\mathbf{p} = \mathbf{M}\dot{\boldsymbol{\eta}}$ with the symmetric stiffness matrix $\mathbf{K} = -\mathbf{A}_{pq}\mathbf{A}_{qp}\mathbf{M}$.

In [4], Buil and Pelegay inspect the structure of the matrices \mathbf{A}_{pq} , \mathbf{A}_{qp} and point out that these matrices can be systematically established for the entire system without the need for constructing a bond graph if the flexible mechanical structure is partitioned into several elements and if a generic modal submodel is used for each of these distributed parameter subsystems. Once these matrices are available, the Lagrange equation for the displacements of the inertias can be set up. With its solutions, the bending $w(x, t)$ can be computed for each location x of the entire flexible structure.

After an eigenvalue analysis of the homogeneous part of Equation 9.29, i.e., after determination of the natural frequencies and the eigenvectors of the entire system, Margolis performs another transformation leading to a decoupling of the equations [24]. If the eigenvectors \mathbf{r}_i ($i = 1, 2, \dots, k$) are combined into a the matrix $\mathbf{R} = [\mathbf{r}_1 \mathbf{r}_2 \dots \mathbf{r}_k]$, then the transformation

$$\mathbf{z} := \mathbf{R}^{-1}\boldsymbol{\eta} \quad (9.30)$$

converts Equation 9.29 into a set of *decoupled* equations of the form

$$\mathcal{M}\ddot{\mathbf{z}} + \mathcal{K}\mathbf{z} = \mathbf{R}^T\mathbf{B}\mathbf{e}, \quad (9.31)$$

where $\mathcal{M} := \mathbf{R}^T\mathbf{M}\mathbf{R}$ and $\mathcal{K} := \mathbf{R}^T\mathbf{K}\mathbf{R}$ are diagonal matrices. Thus, after this second transformation step, the *entire* flexible structure can be represented in the same manner as a distributed parameter subsystem by a set of decoupled oscillators excited by external efforts via transformers. The number of the oscillators in the overall model is equal to the sum of all modal oscillators in the submodels of the flexible elements and other oscillators in lumped parameter models that excite the overall flexible structure.

9.4 Finite Element Method

Besides modal analysis, the finite element method is another powerful and often used approach to an approximation of distributed parameter models. This method is supported by software programs used in industry, e.g. ANSYS^{®1} [1] or NASTRAN^{®2} [28]. Most often, the finite element method (FEM) is applied to flexible mechanical components, entire structures or to fluids. However, the finite element method is also used to compute heat transfer problems or magnetic fields. The applications mentioned indicate that the finite element method is mainly used to model subsystems or problems within one discipline, whereas bond graphs are specially suited for modelling multidisciplinary systems or problems. This section shows that both methods may be suitably combined. The presentation is confined to mechanical systems.

As it is well known, the starting point of the finite element method is to partition the geometric shape of a body into parts or regions of appropriate shape and size called finite elements or simply elements. The latter are coupled with adjacent elements by means of nodes on their boundaries. As a result, the overall geometric shape is composed of numerous elements of different shape and size connected together by nodes on their boundaries. The finite element method enables to set up systematically a set of equations for the physical unknowns in the nodes. Depending on the problem under consideration, the latter can be mechanical displacements, forces, stress, pressures or temperatures.

One reason for the popularity of the finite element method is that real technical components with complex geometric shapes can be built by composing elements of simpler geometric shapes according to a unit construction system. For that purpose, FEM software provides libraries with numerous element types (rods, beams, plates, etc.). The mass and the stiffness matrices of these elements can be used to systematically set up the corresponding matrices in the state space model of the overall structure. By this way, a (large) set of linear equations results that approximately describes the overall system. Using (local) refinement of the finite element grid, a required accuracy can be met at the price that the number of unknowns and consequently the computation time for solving the set of equations significantly increases.

This kind of computational problem, however, has the advantage that its computation can be distributed over several processors and can be performed in parallel. In contrast, analytical calculations based on the theory of elasticity are essentially confined to simple elements. There are quite a number of textbooks on the finite element method. Although the method is applied to problems in many fields, application to multidisciplinary problems still is not easy and not fully supported by standard FEM software.

¹ ANSYS, ANSYS Workbench, AUTODYN, CFX, FLUENT and any all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries.

² NASTRAN is a registered trademark of NASA, Suite 5K39 Washington, DC 20546-0001, USA, <http://www.nasa.gov/>

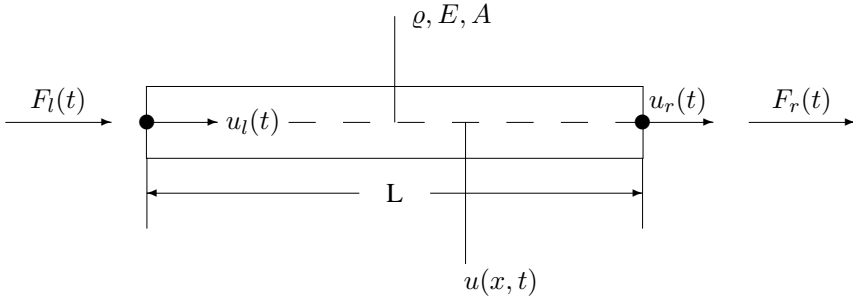


Fig. 9.10 Longitudinal vibrating bar element with two nodes

9.4.1 Classical Finite Element Method Revisited

Before we address a combination of bond graph modelling and the classical finite element method, some basics of the latter are briefly recalled.

Longitudinal Vibrations in a Bar

For that purpose, consider as basic element a bar of length L , of constant cross sectional area A , uniform density ρ and Young's modulus E as depicted in Figure 9.10. In the bar, longitudinal vibrations may take place and $u(x, t)$ may denote the elongation at location x at time t . Let $u_l(t)$, $u_r(t)$ be the elongations at the left end and at the right end side of the bar.

From the balance of forces acting on an infinitesimal small lump of length dx , the partial differential equation

$$\rho A \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x} \left(EA \frac{\partial u}{\partial x} \right) \quad (9.32)$$

results. For a displacement $u(x, t)$ inside the element, that is, $0 < x < L$, the finite element method assumes that it can be approximated by a function

$$\tilde{u}(x, t) := g_1(x)u_l(t) + g_2(x)u_r(t), \quad (9.33)$$

where g_1 , g_2 are linear independent, still unknown so-called *shape functions*. Let $\mathbf{d}(t) := [u_l(t) \ u_r(t)]^T$ and $\mathbf{G}(x) := [g_1(x) \ g_2(x)]$. Then, Equation 9.33 can be written as

$$\tilde{u}(x, t) = \mathbf{G}(x) \cdot \mathbf{d}(t). \quad (9.34)$$

For the kinetic energy, T , of the bar element

$$T(t) = \frac{1}{2} \int_0^L \rho A \left(\frac{\partial \tilde{u}}{\partial t} \right)^2 dx \quad (9.35)$$

we get

$$T(t) = \frac{1}{2} \dot{\mathbf{d}}^T \mathbf{M} \dot{\mathbf{d}}, \quad (9.36)$$

where

$$\mathbf{M} := \int_0^L \rho A \mathbf{G}^T(x) \mathbf{G}(x) dx \quad (9.37)$$

denotes the mass matrix. The potential energy, V , of the bar element is equal to the elastic deformation energy and is given by

$$V(t) = \frac{1}{2} \int_0^L EA \left(\frac{\partial \tilde{u}}{\partial x} \right)^2 dx. \quad (9.38)$$

By means of Equation 9.34, this expression can be written in the form

$$V(t) = \frac{1}{2} \mathbf{d}^T \mathbf{K} \mathbf{d}, \quad (9.39)$$

where

$$\mathbf{K} := \int_0^L EA (\mathbf{G}')^T(x) \mathbf{G}'(x) dx \quad (9.40)$$

is the stiffness matrix. By combining the forces acting on the end nodes into a vector of node forces, $\mathbf{F}(t) = [F_l(t), F_r(t)]^T$, and substituting it into the Lagrange equation

$$\frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\mathbf{u}}} \right) - \frac{\partial L}{\partial \mathbf{u}} = \mathbf{F}, \quad (9.41)$$

$L := T - V$, along with the expressions for the kinetic and the potential energy, we get for the vector \mathbf{u} of displacements in the end nodes of the bar the equation of motion

$$\mathbf{M} \ddot{\mathbf{d}}(t) + \mathbf{K} \mathbf{d}(t) = \mathbf{F}(t). \quad (9.42)$$

This equation can be represented by the multibond graph in Figure 9.11.

The mass matrix as well as the stiffness matrix are symmetric as their above given expressions show. Their coefficients are determined by the shape functions. The latter result from the assumption that the axial stiffness is time-independent throughout the system (cf. [14], Section 13.4). That is, integration of the requirement

$$EA \frac{\partial^2 \tilde{u}}{\partial x^2} = 0 \quad \forall x \in (0, L) \quad (9.43)$$

with respect to the location x yields

$$\tilde{u}(x, t) = c_1(t)x + c_2(t). \quad (9.44)$$

From the boundary conditions $\tilde{u}(0, t) = u_l(t)$ and $\tilde{u}(L, t) = u_r(t)$ we get the ‘constants of integration’.

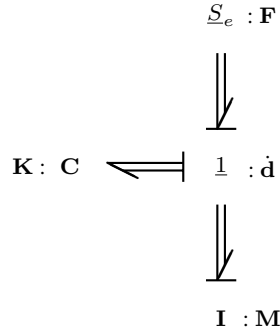


Fig. 9.11 Multibond graph representation of a bar element

$$c_1(t) = \frac{u_r(t) - u_l(t)}{L} \tag{9.45a}$$

$$c_2(t) = u_l(t) \tag{9.45b}$$

Thus, the shape functions are

$$g_1(x) = 1 - \frac{x}{L} \tag{9.46a}$$

$$g_2(x) = \frac{x}{L} . \tag{9.46b}$$

With these shape functions, the mass matrix and the stiffness matrix take the form

$$\mathbf{M} = \frac{\rho A L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{9.47a}$$

$$\mathbf{K} = \frac{E A}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} . \tag{9.47b}$$

Reference Frames

The above computation of the mass matrix and the stiffness matrix of a bar element assumes a *local* reference frame sitting in the left side node of the element. However, if the bar is part of a flexible structure, e.g. a planar triangle built by three rods, then the bar’s position must be taken into account with reference to global frame. For that purpose, we assign displacements in a global reference system (\bar{x}, \bar{y}) a horizontal bar over the symbol: $\bar{\mathbf{u}} = [\bar{u}_x, \bar{u}_y]^T$. If, for instance, a local reference frame (x, y) is rotated by an angle α with respect to the global frame, then the transformation of a displacement $\bar{\mathbf{u}}$ into the corresponding displacement \mathbf{u} in the local frame is

$$\mathbf{u} = \mathbf{T} \bar{\mathbf{u}} , \tag{9.48}$$

where \mathbf{T} denotes the matrix

$$\mathbf{T} := \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \\ 0 & 0 & -\sin \alpha & \cos \alpha \end{bmatrix}. \quad (9.49)$$

Thus, in the global frame, the equation of motion Equation 9.42 reads

$$\mathcal{M} \ddot{\bar{\mathbf{u}}} + \mathcal{K} \bar{\mathbf{u}} = \bar{\mathbf{F}}, \quad (9.50)$$

where $\mathcal{M} = \mathbf{T}^T \mathbf{M} \mathbf{T}$, $\mathcal{K} = \mathbf{T}^T \mathbf{K} \mathbf{T}$ and $\bar{\mathbf{F}} = \mathbf{T}^T \mathbf{F}$. Note that the transformation matrix \mathbf{T} is orthogonal.

Assemblies of Bars

If an equation of motion has been set up for the displacements of each finite element of an overall flexible structure, then the mass matrix and the stiffness matrix in the dynamic equation of the entire flexible structure can be constructed in a systematic manner from the corresponding matrices of the elements. For that purpose, one makes use of the simple fact that the displacements of nodes joined together must be equal and that forces in these nodes must be in balance. In this brief outline of the finite element method, we will refrain from an illustration how to build the mass and the stiffness matrix of a flexible structure from the corresponding matrices of its elements. For details, see e.g. [26].

If a flexible structure can be partitioned into a small number of simple elements, then an alternative option is to set up the Lagrange function for each element analytically and to add them up into the Lagrange function of the entire flexible structure. Its proper differentiation yields the equations of motion for the entire structure. For instance, assume a cantilever bar that can be divided into three elements with different cross sectional areas and different length. Each bar element can be considered to be a bar with two end nodes for which the Lagrange function is analytically determined. From the sum of these functions, the Lagrange equations for the displacements of the four nodes can be derived (cf. [14], Example 13.4.3, pp. 297)

Often, it is assumed that the stress inside an element is constant because linear interpolation is used and the gradients of stress or displacements are constant within each finite element. This consideration was used above when we determined the shape functions for a bar element. Since the stress in an element differs from the one in the neighbouring element, there is a discontinuity in the node connecting both elements. These discontinuities can be decreased by a refinement of the finite element grid. The disadvantage, however, is that the computational amount significantly increases with the number of elements. To avoid this disadvantage and to improve accuracy, the number of degrees of freedom of each element is increased. Besides the nodes on the boundary, an element has one or more internal nodes. We may think that, for example, a rod being part of a flexible structure is itself composed of rod elements. As an example, Figure 9.12 shows a planar triangle composed of

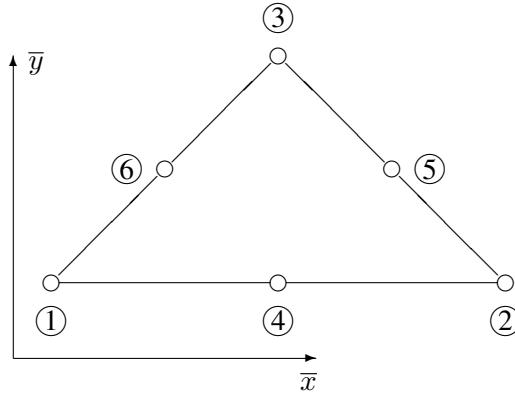


Fig. 9.12 Planar triangle composed of three bars and six nodes

three bars. For each of these an internal node in its centre is taken into account. That is, the triangle has six nodes and 12 degrees of freedom for a two-dimensional model.

As the simple example of a longitudinal vibrating bar shows, the finite element method converts the problem of solving the partial differential equation, Equation 9.32, into the task of solving a linear second order ODE, Equation 9.42, for the displacements of the nodes on the boundary of an element. This is achieved by assuming that the displacement, $u(x, t)$, at any location x , $0 < x < L$, can be expressed as a sum of the displacements at the boundary weighted by shape functions, Equation 9.33, that depend on x . If damping forces due to friction inside the material are neglected, then the matrices in the resulting ODE are the mass and the stiffness matrix of the element.

9.4.2 Bond Graph Representation of Finite Element Models

The equations of motion of a flexible structure modelled by finite elements can be represented by a bond graph if the matrices are considered as the matrices of an I field or a C field. One field stores kinetic energy while the other one stores potential energy. Since both matrices are symmetric, their constitutive equations comply with Maxwell's reciprocity condition (Equation 8.8 or Equation 8.5). That is, both fields are energy conserving. The sum of inertia and spring forces according to Equation 9.42 can be represented by 1-junctions for the node velocities [29]. Hence, a finite element with one internal node can be represented by the multibond graph in Figure 9.13. The dimension of the multibonds in that graph is equal to the number of degrees of freedom of the nodes.

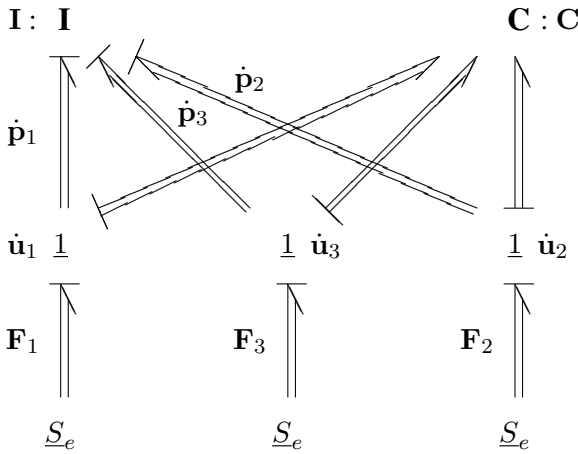


Fig. 9.13 Multibond graph representation of a finite element with three nodes

The structure of the bond graph and its elements are retained, if there is a change from a local frame to a global one. Only the matrices are to be replaced by the corresponding transformed ones (cf. Equation 9.50). If the description of each finite element of a flexible structure is related to a global reference frame, then the finite element models can be composed to a bond graph model of the overall flexible structure. However, since end nodes of different finite elements have the same velocity when merged into a node connecting two adjacent finite elements, many of the ports of the I field of a finite element must accept derivative causality. As can be seen from Figure 9.13, the 1-junction representing the velocity of a joint node between two finite elements is connected with the I field of both finite element models. The problem of many storage ports with derivative causality can be overcome by lumping the inertia of a flexible structure in the nodes connecting finite elements.

Essential Degrees of Freedom

Furthermore, since a finite element grid with a high number of nodes leads to a correspondingly high number of degrees of freedom, it has become common to distinguish between essential and non-essential displacements before an eigenvalue analysis is performed. Assuming that no external forces act at points with non-essential displacements, the corresponding degrees of freedom can be removed and the order of the model can be reduced. If the mass of a flexible structure is condensed into the nodes connecting adjacent finite elements, then it is obvious to consider internal nodes of finite elements as candidates of nodes with non-essential displacements and to eliminate them. Thus, the vector \mathbf{u} of nodal displacements can be partitioned into a sub-vector \mathbf{u}_c of displacements of coupling nodes and a sub-vector \mathbf{u}_e of displacements of internal nodes to be eliminated. Consequently, the stiffness matrix \mathbf{K} can be partitioned and in steady-state, Equation 9.42 reads

$$\begin{bmatrix} \mathbf{K}_{cc} & \mathbf{K}_{ce} \\ \mathbf{K}_{ec} & \mathbf{K}_{ee} \end{bmatrix} \begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_e \end{bmatrix} = \begin{bmatrix} \mathbf{F}_c \\ \mathbf{0} \end{bmatrix}, \tag{9.51}$$

where \mathbf{F}_c denotes the vector of external forces acting on the coupling nodes. For the displacements \mathbf{u}_e to be eliminated, it follows

$$\mathbf{u}_e = \mathbf{K}_{ee}^{-1} \cdot \mathbf{K}_{ec} \mathbf{u}_c. \tag{9.52}$$

Thus,

$$\mathbf{u} = \mathbf{T} \mathbf{u}_c, \tag{9.53}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{ee}^{-1} \mathbf{K}_{ec} \end{bmatrix}. \tag{9.54}$$

With this relation, Equation 9.42 reduces to an equation of motion for the essential degrees of freedom.

$$\mathbf{T}^T \mathbf{M} \mathbf{T} \ddot{\mathbf{u}}_c + \mathbf{T}^T \mathbf{K} \mathbf{T} \mathbf{u}_c = \mathbf{T}^T \mathbf{F} \tag{9.55}$$

Thus, if the spatially distributed mass of a flexible structure is condensed into the nodes that join adjacent finite elements and if displacements of internal finite element nodes considered non-essential are eliminated, then a finite element can be represented by the simplified bond graph model depicted in Figure 9.14.

If the displacements of nodes connecting finite elements are expressed in a global reference frame (cf. Equation 9.50), then, by repeated use of the model in Figure 9.14, a bond graph model for a flexible structure composed of bar elements can be assembled. The resulting bond graph model has a *reduced* number of degrees of freedom and shows no causal conflicts.

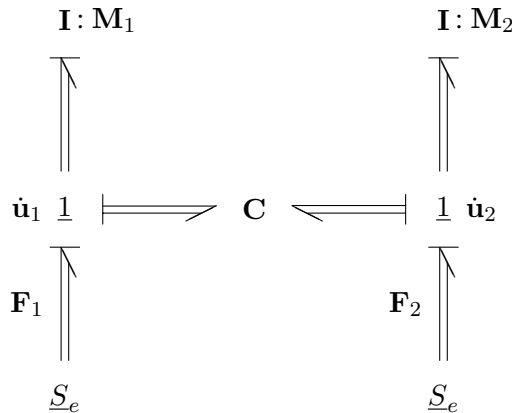


Fig. 9.14 Finite element bond graph model after mass condensation and reduction of internal degrees of freedom

Bond Graph Representation of a Finite Element Model of a Beam with Transverse Motion

The bond graph model of Figure 9.14 is not confined to finite elements of longitudinal vibrating bars. For transverse motion of beam elements, the model has the same structure. To see this, consider the two-dimensional case. Let $w(x, t)$ be the transverse displacement at location x of a beam of length L and $q(x, t)$ a force load distributed over the length of the beam element. Under the Bernoulli hypothesis that there is no shear deformation and that lumps of thickness Δx have no rotary inertia (Bernoulli-Euler beam), the transverse displacement $w(x, t)$ is determined by the partial differential equation

$$\rho A \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^4} - q = 0 . \quad (9.56)$$

Like for a longitudinal vibrating bar, this partial differential equation can be converted into a second order ODE for the displacements at both ends of a beam element. Let the angle ψ denote the rotation of the neutral axis with respect to a horizontal reference, $\psi(x, t) := \partial w / \partial x$, and let the transverse displacement $w(x, t)$ and the angle ψ have an index l or an index r at the left side or the right side of the beam. These four displacements are combined into a vector $\mathbf{d}(t) := [w_l(t) \psi_l(t) w_r(t) \psi_r(t)]^T$.

Similar to the case of a longitudinal vibrating bar, it is assumed that the exact solution $w(x, t)$ of Equation 9.56 can be approximated by a function

$$\tilde{w}(x, t) := \mathbf{G}(x) \cdot \mathbf{d}(t) , \quad (9.57)$$

where the coefficients of the matrix $\mathbf{G}(x) = [g_1(x) \ g_2(x) \ g_3(x) \ g_4(x)]$ are shape functions to be determined³.

Furthermore, let $D_1 := \partial^2 / \partial t^2$ and $D_2 := \partial^4 / \partial x^4$ be differential operators and $D := (\rho A \times D_1 + EI \times D_2)$ such that Equation 9.56 can be written in the form

$$D(w) = q , \quad (9.58)$$

for short.

As $\tilde{w}(x, t)$ is an approximation of $w(x, t)$,

$$R(x, t) := D(\tilde{w}(x, t)) - q(x, t) \neq 0 . \quad (9.59)$$

Now, according to Galerkin's weighted residual method, the error or residual, $R(x, t)$, is multiplied by the functions $g_j(x)$ and it is required that

³ Note that the exact solution of Equation 9.56 is *approximated* by a finite linear combination of linear independent functions from a set of basis functions $\{g_i(x)\}$ in contrast to the classical modal analysis where it is assumed that the exact solution can be represented as a series of time dependent functions $\eta_\nu(t)$ weighted by functions $Y_\nu(x)$ being solutions of Equation 9.3.

$$\int_0^L g_j(x) R(x, t) dx = 0 \quad j = 1, \dots, 4. \quad (9.60)$$

Substituting $\tilde{w}(x, t)$ into the functional of Equation 9.60 results in

$$\sum_{i=1}^4 \left(\int_0^L \varrho A g_i(x) g_j(x) dx \right) D_1(d_i(t)) + \sum_{i=1}^4 \left(\int_0^L EI \times D_2(g_i(x)) g_j(x) dx \right) d_i(t) = \int_0^L q(x, t) g_j(x) dx. \quad (9.61)$$

This leads to an equation of motion for the nodal displacements at the ends of the beam

$$\mathbf{M} \ddot{\mathbf{d}}(t) + \mathbf{K} \mathbf{d}(t) = \mathbf{F}(t). \quad (9.62)$$

That is, a beam element can also be represented by the multibond graph in Figure 9.11.

The coefficients in these matrices are

$$m_{ij} := \varrho A \int_0^L g_i(x) g_j(x) dx \quad (9.63a)$$

$$k_{ij} := EI \int_0^L g_i''(x) g_j''(x) dx \quad (9.63b)$$

where $i, j = 1, \dots, 4$.

The components of the vector \mathbf{F} are given by

$$F_j(t) := \int_0^L g_j(x) q(x, t) dx, \quad (9.64)$$

where $j = 1, \dots, 4$.

Now, let us determine the still unknown shape functions $g_i(x)$. Assume that there is no load distributed over the length of the beam. Then, repeated integration of the ODE

$$EI \frac{\partial^4 \tilde{w}}{\partial x^4} = 0 \quad (9.65)$$

eventually yields for the transverse displacement $\tilde{w}(x, t)$

$$\begin{aligned} \tilde{w}(x, t) = & \left(1 - \frac{3x^2}{L^2} + \frac{2x^3}{L^3}\right) \times w_l(t) + \left(\frac{x}{L} - \frac{2x^2}{L^2} + \frac{x^3}{L^3}\right) L \times \psi_l(t) \\ & + \left(\frac{3x^2}{L^2} - \frac{2x^3}{L^3}\right) \times w_r(t) + \left(-\frac{x^2}{L^2} + \frac{x^3}{L^3}\right) L \times \psi_r(t). \end{aligned} \quad (9.66)$$

Hence, the shape functions to be determined are

$$g_1(x) = 1 - \frac{3x^2}{L^2} + \frac{2x^3}{L^3} \quad (9.67a)$$

$$g_2(x) = \left(\frac{x}{L} - \frac{2x^2}{L^2} + \frac{x^3}{L^3} \right) L \quad (9.67b)$$

$$g_3(x) = \frac{3x^2}{L^2} - \frac{2x^3}{L^3} \quad (9.67c)$$

$$g_4(x) = \left(-\frac{x^2}{L^2} + \frac{x^3}{L^3} \right) L. \quad (9.67d)$$

Again, assume that the mass of the beam element can be condensed in the left and the right side points. Then, a beam element with transverse motion can also be represented by the bond graph shown in Figure 9.14.

Finally, if longitudinal and transverse motion are superimposed, then the 1-junctions represent the derivatives of the nodal displacements $\mathbf{d}_i = [u_i, w_i, \psi_i]$ ($i = 1, 2$), where u_i denotes the *longitudinal* displacement of the i^{th} node at the left or right end of an element. Forces and moments at that node are combined into a vector \mathbf{F}_i .

The stiffness matrix of a beam element results from the evaluation of the integral in Equation 9.63b using the expressions for the shape functions (Equations 9.67a–9.67d) and by taking into account the forces and bending moments acting at the left and the right end of a beam element. The resulting stiffness matrix reads

$$\mathbf{K} = \begin{bmatrix} EA/L & 0 & 0 & -EA/L & 0 & 0 \\ 0 & 12EI/L^3 & 6EI/L^2 & 0 & -12EI/L^3 & 6EI/L^2 \\ 0 & 6EI/L^2 & 4EI/L & 0 & -6EI/L^2 & 2EI/L \\ -EA/L & 0 & 0 & EA/L & 0 & 0 \\ 0 & -12EI/L^3 & -6EI/L^2 & 0 & 12EI/L^3 & -6EI/L^2 \\ 0 & 6EI/L^2 & 2EI/L & 0 & -6EI/L^2 & 4EI/L \end{bmatrix}. \quad (9.68)$$

Bond Graph Model of a Cantilever Beam

Having developed a bond graph model for longitudinal vibrations of bar elements and transverse motion of beam elements, let us consider the well known example of a cantilever beam fixed at its left end as depicted in Figure 9.15.

The external point forces acting at three different locations may be due to other subsystems of an overall systems. Again, we assume a Bernoulli-Euler beam described by Equation 9.56. The beam is divided into three finite elements. If the mass of the beam is condensed in the nodes coupling the finite elements and if the displacements of internal element nodes are considered non-essential that can be eliminated, then a 2D problem with 3×2 degrees of freedom results. These degrees of freedom are the longitudinal displacement u_i , the transverse displacement w_i and the rotation angle ψ_i of the neutral axis with respect to a horizontal reference at locations x_i , $i = 3, 5, 7$ (cf. Figure 9.15). Under these assumptions, the generic bond graph model of Figure 9.14 can be used for each beam element. By concatenating the element models and condensing their inertias at the nodes between finite

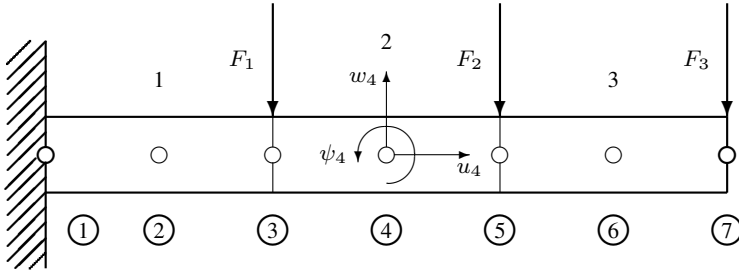


Fig. 9.15 Cantilever beam with point forces acting at three locations

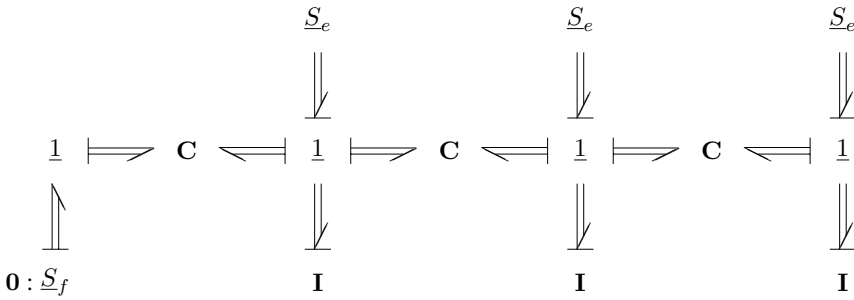


Fig. 9.16 Bond graph model of a cantilever beam with three finite elements and condensation of the mass into the coupling nodes

elements, we get a bond graph model of the cantilever beam that accounts for its 2D motion (Figure 9.16).

9.5 Conclusion

The previous chapter has shown that in mechanics, bond graph modelling is *not* confined to one-dimensional translational or rotational motion of rigid bodies, but rather enables a compact description of more general 3D motion of systems of linked rigid bodies. In this context, the question has come up as to how mechanical subsystems, to be considered as a continuum, can be incorporated in a bond graph modelling of an overall system. In other words, the question has been how a distributed parameter model can be approximated by a lumped parameter bond graph model. The question is of practical relevance because a closer look at some mechanical components shows that inertia and compliance effects are to be considered spatially distributed

and lumping them may not be sufficient, depending on the requirements modelling a particular system has to meet.

The assumption of a distributed parameter model is not confined to mechanical subsystems. A consequence, however, is that the mathematical model of the overall system would consist of a boundary value problem for the continuum and an initial value problem with a set of ordinary differential equations for the rest of the system where the latter provides the boundary conditions for the partial differential equations describing the continuum. That is, the boundary conditions cannot be specified as functions of time, but rather result from the solution of the initial value problem which, again, needs input from the boundary value problem. Many modelling and simulation programs do not accept such a combined description, at least not directly. Thus, the question is how a distributed parameter model of a subsystem can be sufficiently approximated by a lumped parameter model to be represented by a bond graph.

If the aim is the development of a lumped parameter model, then an obvious approach is to divide a mechanical continuum along one axis into ideal rigid segments followed by ideal flexible segments (cf. Figure 9.1). If the thickness of the segments tended to zero, then their corresponding ordinary differential equations become the partial differential equations describing the continuous spatial distribution of mass and compliance. The disadvantage of such a ladder structure model, however, is that the accuracy of simulation results does not increase accordingly with the number of lumps. Thus, this approach to a lumped parameter approximation of a distributed parameter model results in a high number of state equations.

Beyond this simple approach, it has been shown that the results of two classical methods, namely the normal mode analysis and the finite element method, can be represented in terms of bond graphs. Consequently, the advantage of the two methods can be used in bond graph modelling. The relation between modal analysis and the bond graph methodology basically relies on the fact that the equations resulting from the classical separation of variables approach can be represented by a power conserving bond graph junction structure connecting external excitations with modal oscillators (Figure 9.4). This important observation goes back to Karnopp [18].

For practical computation, only a finite number of the infinite degrees of freedom can be retained. As has been shown, after this step, the model can be clearly represented by a multibond graph. The number of normal frequencies to be retained depends on the range of frequencies that is of interest with regard to the overall system. This, however, is not known a priori and must be estimated based on engineering experience which may be considered a weak point of the approach. However, if significant normal frequencies are captured, highly accurate low order models can be used as Margolis has found in several different projects. As a rule of thumb, he suggests to account for all modal frequencies up to a frequency that is above twice the highest frequency of interest in an overall system (cf. [22], or [19], p. 398ff). Motion time histories resulting from models with different numbers of modal frequencies are given by Margolis and Tabrizi [24].

The basic model given by Karnopp (Figure 9.4) is *generic* in the sense that the structure of the bond graph remains invariant for linear distributed parameter models of different continua. Only modal masses, stiffnesses and mode shapes appearing in transformer moduli must be computed according to the geometric shape of the continuum under consideration. If necessary, this can be done numerically. Nevertheless, this can become difficult at least for 3D models of bodies with complicated geometric shape. In such cases, an obvious approach is to partition a flexible structure into substructures in such a way that the overall system model can be built by repeated use of the generic model (Figure 9.4) for the substructures. Moreover, as pointed out by Karnopp and Margolis, the force free model can be used to approximate other boundary value problems.

As long as forces or moments are inputs of the generic model of flexible substructures, no causal problems occur. Consequently, the distributed parameter model of an overall flexible structure can be approximated by a state space model. This result can be achieved by assembling the modal models of substructures in such a way that models are coupled by C energy stores [4, 21].

To ensure that there is no relative motion between substructures, connecting springs must be sufficiently stiff. If a velocity is imposed on the modal model of a flexible substructure by another submodel or by the rest of the overall system model, then a modal inertia becomes dependent from all other modal inertias (Figure 9.6). This can be avoided by taking into account only the C energy store of all modal oscillators with a natural frequency that is beyond a certain frequency range (Figure 9.7). Since these oscillators do not contribute significantly to the dynamic response, their inertia can be neglected and their compliance can be combined into a two-port C element. If the generic modal model of substructure is excited by more than one velocity input, then algebraic loops arise in its junction structure that are to be symbolically solved (Figure 9.9).

An essential feature of the finite element method is that a model of a system component of complex geometric shape can be composed by building blocks for elements of simple geometric shape. Finite element programs come with libraries that make models for various element types available. With the information about the geometry and the positions of nodes (relative to a local body fixed reference frame) about material properties and boundary conditions, stiffness and mass matrices of finite elements can be set up and composed into the corresponding matrices of the overall flexible mechanical structure (related to a global reference frame). The result is a large set of second order ODEs for the displacements of the nodes of the finite element grid.

The finite element approach can be linked to bond graph modelling by considering the stiffness and the mass matrix of finite elements as a C-field or an I-field and by representing the equations of motion for the nodes of an element by a multi-bond graph (Figure 9.13). The latter can be considered *generic* like the one resulting from normal mode analysis because it does not only hold for bars. For other element types it has the same structure. If this model is related to a global reference frame (Equation 9.50), then models of finite elements can be composed to a model of a flexible system component. Since end nodes of adjacent finite elements have the

same joint velocity, many parts of the I-fields must accept derivative causality. This problem can be removed by lumping the mass of connected finite elements in their coupling nodes. Moreover, if displacements of internal nodes of a finite element are eliminated, then its generic bond graph model simplifies to the one depicted in Figure 9.14. For illustration, this model has been used to build a bond graph model of a clamped beam.

Consideration of the modal analysis and of the finite element method from a bond graph modelling point of view leads to a generic model. By computing the model parameters for each element and by connecting these submodels, a bond graph of an overall flexible structure can be constructed that can be used as a submodel in a bond graph of an overall system. For dividing a flexible system component into finite elements and for setting up the mass and the stiffness matrix, a standard finite element program can be used independently of the modelling of the rest of the system.

The considerations in this chapter show that bond graph methodology can be well combined with classical methods for distributed parameter models in order to construct low order lumped parameter bond graph models that approximate distributed parameter models sufficiently accurate. That is, overall system models incorporating distributed parameter submodels can be uniformly represented by bond graphs and processed by contemporary bond graph modelling and simulation software.

References

- [1] ANSYS Inc. URL <http://www.ansys.com/products/>.
- [2] M. Berzeri and A.A. Shabana. Development of simple models for the elastic forces in the absolute nodal coordinate formulation. *J. Sound Vib.*, 235:539–565, 2000.
- [3] L.S. Bonderson. Vector bond graphs applied to one-dimensional distributed systems. *Journal of Dynamic Systems, Measurement and Control*, pages 75–82, 1975.
- [4] F. Buil and J.L. Pelegay. Bond-graph modelling of continuous dynamic systems and the systematic computation of their corresponding mass and stiffness matrices. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 59–64. SCS Publishing, January 17–20 1993. Simulation Series, volume 25, no. 2. ISBN: 1-56555-019-6.
- [5] M.A. Crisfield and G.F. Moita. A unified co-rotational framework for solids, shells and beams. *Int. J. Solids Struct*, 33:2969–2992, 1996.
- [6] V. Damić. A Bond Graph Approach to Modelling of Spatial Flexible Multibody Systems Based on Co-rotational Formulation. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation*, volume 37 (1), pages 213–218, 2005.
- [7] V. Damić. Modelling flexible body systems: a bond graph component model approach. *Mathematical & Computer Modelling of Dynamic Systems*, 12 (2-3):175–187, 2006.
- [8] V. Damić and M. Čohodar. A Bond Graph Approach to Modelling of Spatial Flexible Multibody Systems Based on Co-rotational Formulation. In J.J. Granda and F.E. Cellier, editors, *Proc. 2005 International Conference on Bond Graph Modeling and Simulation (ICBGM '05)*, volume 37(1) of *Simulation Series*, pages 213–218. SCS, 2005.
- [9] A. Derkaoui, E. Bideaux, and S. Scavarda. A first approach of distributed parameters systems sizing using bond graphs. In F. E. Cellier and J. J. Granda, editors, *Proc. of the 2003 Inter-*

- national Conference on Bond Graph Modeling and Simulation*, volume 35 (2), pages 79–86, 2003.
- [10] A. Derkaoui, E. Bideaux, and S. Scavarda. Finite Element local structure assembly and shape functions influence on bond graph modelling. In J. J. Granda and F. E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation*, volume 37 (1), pages 78–85, 2005.
- [11] A. Elramady and J.J. Granda. Modeling and Simulation of Flexible Multi-Bodies with Bond Graphs Application to the Zvezda Mission of the International Space Station. In J.J. Granda and F.E. Cellier, editors, *Proc. of the International Conference on Bond Graph Modeling and Simulation*, volume 37 (1), pages 2006–212, 2005.
- [12] J.J. Granda and N. Kong. Time Dependent Computational Relations Between Finite Elements and Bond Graph Modeling. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, volume 25 (2), pages 29–34. SCS Publishing, January 17-20 1993. ISBN 1-56555-019-6.
- [13] J.J. Granda and N. Kong. Pseudo Bond Graph Models and Finite Element Models of Transient Heat Transfer Problems. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 1993 International Conference on Bond Graph Modeling and Simulation*, volume 25 (2), pages 293–300, 1993.
- [14] D.J. Inman. *Vibration – With Control, Measurement, and Stability*. Prentice Hall, Englewood Cliffs, New Jersey, 1989.
- [15] J.L. Baliño. Bond-graph formulation of CFD problems with constant piecewise shape functions. *International Journal of Heat and Technology*, 21 (1):59–66, 2003.
- [16] J.L. Baliño. Galerkin finite element method for incompressible thermofluid flows framed within the bond graph theory. *Simulation Modelling Practice and Theory*, 17 (1):35–49, 2009.
- [17] J.L. Baliño and A.E. Larreteguy and E.F.G. Gandolfo Raso. A general bond graph approach for computational fluid dynamics. *Simulation Modelling Practice and Theory*, 14:884–908, 2006.
- [18] D.C. Karnopp. Computer representations of continuous vibratory systems using normal modes and bond graph techniques. *Simulation*, 10(3), 1968.
- [19] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1990.
- [20] M. Lebrun. The use of modal analysis concepts in the simulation of pipeline transients. *Journal of the Franklin Institute*, 319(1/2):137–156, 1985.
- [21] D.L. Margolis. Finite Mode Bond Graph Representation of Vehicle-Guideway Interaction Problems. *Journal of the Franklin Institute*, 302(1):1–17, 1976.
- [22] D.L. Margolis. A Survey of Bond Graph Modeling for Interacting Lumped and Distributed Systems. *Journal of the Franklin Institute*, 319(1/2):125–135, 1985.
- [23] D.L. Margolis. Application of Bond Graphs to Distributed Parameter Systems. In P.C. Breedveld and G. Dauphin-Tanguy, editors, *Bond Graphs for Engineers*, pages 153–175. Elsevier, North-Holland, 1992.
- [24] D.L. Margolis and M. Tabrizi. Reduction of Models of Interacting Lumped and Distributed Systems Using Bond Graphs and Repeated Modal Decomposition. *Journal of the Franklin Institute*, 309(5):309–322, 1984.
- [25] H.A. Mergen and J.J. Granda. Bond Graph Models and Finite Element Models for Engine Valve Spring Transient Response. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 1997 International Conference on Bond Graph Modeling and Simulation*, volume 29 (1), pages 129–134, 1997.
- [26] S. Moaveni. *Finite Element Analysis – Theory and Application with ANSYS*. Prentice Hall, 1999.
- [27] A. Mukherjee, R. Karmakar, and A.K. Samantaray. *Bond Graph in Modeling, Simulation and Fault Identification*. I.K. International Publishing House, New Delhi, India, 2006. ISBN: 81-88237-96-5.

- [28] NEi Software. URL www.nenastran.com/.
- [29] J.L. Pelegay, M. Doblare, and F.G. Buil. Towards an Efficient Integration of Bond-Graph and Finite Element Systematizations in the Simulation of Continuous Dynamic Systems. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 35–40. SCS Publishing, January 17-20 1993. Simulation Series, volume 25, no. 2, ISBN: 1-56555-019-6.
- [30] P.-Y. Richard and J. Buisson. Using finite element principles for bond graph modelling of continuous systems. In J. J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 1997 International Conference on Bond Graph Modeling and Simulation*, volume 29 (1), pages 123–128, 1997.
- [31] M. Čohodar, W. Borutzky, and V. Damić. The application of a co-rotational approach in bond graph setting to the modeling of general spatial mechanisms undergoing large motions. In J. J. Granda and F. E. Cellier, editors, *2007 International Conference on Bond Graph Modeling (ICBGM 2007)*, pages 148–155. SCS Publishing, 2007. Simulation Series, volume 39, no 1, ISBN: 1-56555-310-1.
- [32] M. Čohodar, W. Borutzky, and V. Damić. Comparison of different formulations of 2D beam elements based on Bond Graph technique. *Simulation Modelling Practice and Theory*, 17 (1):107–124, 2009.

Chapter 10

Bond Graph Modelling of Open Thermodynamic Systems

The basic starting point for the bond graph modelling approach is the exchange of energy between subsystems or system components. Apart from phenomena like electromagnetic radiation, energy exchange, in most cases, is bound to real connections having a mass, e.g. mechanical links or electrical wires. These connections remain invariant with regard to space and time if there are no switches.

In hydraulic and pneumatic systems, the transport of energy is bound to a mass flow. That is, in addition to the transport of potential and kinetic energy, convection of mass, momentum and internal energy must be taken into account. Hence, a bond graph model should not only comply with the principle of energy conservation, but also with conservation laws for mass and momentum. As early as 1977, A. Schöne observed in an introductory article on abstract models of technical systems [33] that

in general, the building blocks of models of thermodynamic systems in process engineering are connected not only by energy flows but also by mass flows.¹

Taking a Lagrangian point of view [21], that is, the reference point is bound to the motion of a group of particles and the requirement for mass conservation is met in a trivial way. However, for computing practical problems, such an approach is not suitable.

Common practice is to introduce a Eulerian control volume of appropriate boundary that can change its size and, if necessary, can move. This consideration is bound to a reference frame fixed in space. Inside the control volume, inertia and compliance of a fluid are spatially distributed. If the boundary of the control volume is pervious to mass such that mass flows can enter and leave, then we have a so-called *open* system. These convection flows make bond graph modelling more difficult because the choice of two variables for which the product equals the total energy at an element port is not straightforward. In hydraulic systems, the product of hydrostatic pressure and volume flow only approximates the amount of instantaneous power.

In view of these difficulties, it is not surprising that several approaches to bond graph modelling of open thermodynamic systems have been proposed in the literature. In the following, two approaches to basic problems shall be considered. This

¹ translated from the German source [33]

chapter focusses on some typical phenomena, viz. the storage of a compressible fluid in a control volume (of variable size), its flow through a restriction, and flow forces on spools in hydraulic valves.

Further reading on bond graph modelling in thermal and chemical engineering may be found, e.g., in [37, 38] exclusively devoted to this topic. Other books include chapters on bond graph modelling in thermodynamics, see, for instance, [6, 9, 11, 25, 28] (This list of references is not meant to be exhaustive). A number of these chapters also present some case studies. With regard to hydraulic systems, Dransfield used bond graphs in his textbook on hydraulic control systems in as early as 1981 [13]. A more recent introduction to bond graph modelling of hydraulic and pneumatic systems by Scavarda can be found in [32]. For the latest journal articles using bond graphs in thermodynamics, see for instance [5, 12, 18, 19, 26, 30, 31].

10.1 Modelling Thermodynamic Systems by Pseudo Bond Graphs

The approaches to bond graph modelling of open thermodynamic systems proposed in the literature can be divided into two categories. One option is to try to develop *true* bond graphs. That is, the product of effort and flow equals the instantaneous total power transmitted between two power ports. This approach has been promoted especially by the research of Breedveld [8]. By considering a number of idealised case studies and by specifying an Eulerian control volume, Beaman and Breedveld [2] showed that for open thermodynamic systems, in principle, *true* bond graphs can be developed without introducing ad hoc elements. This has been carried out for some real-world systems under some simplifying assumptions. Willson and Traver [39], for instance, performed a control volume analysis of components of a 2-stroke internal combustion liquid piston pump resulting in a true bond graph model in which heat transfer has not been taken into account.

Another option is to dismiss this fundamental requirement in favour of more flexibility. Resulting bond graphs are known as *pseudo* bond graphs (cf. Section 2.6). The pseudo bond graph approach to modelling thermodynamic systems has been used, e.g., by Karnopp [20, 22], by Gawthrop and Smith [15], and by Delgado and her co-workers for modelling process engineering systems [16].

In both cases, mass flows and the convection of thermal energy are represented by *separate* bonds, although mass and energy convection cannot be separated. Approaches differ in their choice of effort and flow variables. In the following section, the modelling of thermodynamic systems by pseudo bond graphs is considered.

Apart from the restriction that the product of the variables associated with each bond does not equal the instantaneous power, pseudo bond graphs share all other features with true bond graphs.

- Like for true bond graphs, basic elements, e.g. sources, resistors as well as 0- and 1-junctions can be introduced.

- Pseudo bond graphs can be systematically constructed like true bond graphs.
- The concept of computational causalities is applicable.
- Equations can be automatically derived from causally completed pseudo bond graphs in the same manner as for causal true bond graphs.

Opposed to a greater flexibility in modelling thermodynamic systems with mass flows and thermal convection, there are also some disadvantages.

- C elements are not energy conservative.
- Pseudo bond graphs accounting for thermodynamic processes in an engineering system and true bond graphs for other energy domains cannot be simply coupled via TF elements.

Remark 10.1. Although C elements in pseudo bond graphs are not energy conservative, there are still some advantages.

- Variables common in thermodynamics, e.g. the heat flow, \dot{Q} , can be used.
- The accumulation of mass as well as of heat can be expressed explicitly. Therefore, the C element in pseudo bond graphs is also called an *accumulator* [24].

10.1.1 Pseudo Bond Graph of a Heated Stirred Tank

As a simple introductory example of an open thermodynamic system, consider an open topped tank with a mass flow \dot{m}_i of temperature T_i entering the tank and a mass flow \dot{m}_o of temperature T_o leaving it. It is assumed that the direction of the mass flows does not reverse. The fluid stored in the tank is heated. We assume that stirring ensures a spatially uniformly distributed temperature T in the tank (Figure 10.1, see also [15], p. 44). Moreover, we will assume that

- heat losses to the ambient can be neglected,
- the fluid is incompressible and that
- inertia effects can be neglected as well.

For a description of the hydraulic aspects, we choose as effort the hydrostatic pressure p and as flow the mass flow \dot{m} . Apparently, the product of these two variables does not equal power. Hence, the bond graph in Figure 10.2 accounting for the transport and the accumulation of mass is a *pseudo* bond graph.

Summation of flow variables at the 0-junction just gives the mass balance

$$\dot{m} = \dot{m}_i - \dot{m}_o . \quad (10.1)$$

Note that if one takes p/ρ as effort, then the bond graph in Figure 10.2 becomes a true bond graph accounting for hydrostatic power.

Since convection of thermal energy is bound to mass flows, we follow Karnopp [20] or Gawthrop and Smith [15] and choose the enthalpy flow \dot{H} as flow and the

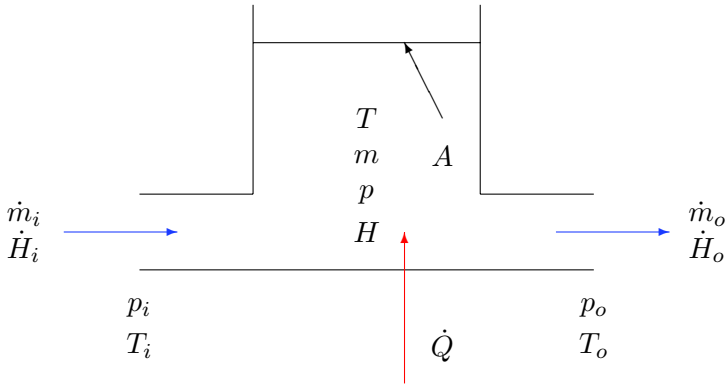


Fig. 10.1 Heated stirred tank

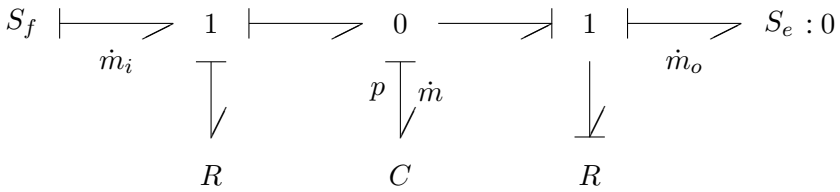


Fig. 10.2 Pseudo bond graph accounting for hydraulic aspects

temperature T as effort for modelling the thermodynamic aspects. Again, the product of both variables does not equal power. The enthalpy flow rather has the physical dimension of power.

Enthalpy Flow Through a Short Pipe

For a short, well insulated conduit, the enthalpy flow is

$$\dot{H} = h \dot{m} \tag{10.2}$$

$$= c_p T \dot{m} , \tag{10.3}$$

where h denotes the enthalpy of a unit mass, c_p the specific heat at constant pressure, and T the upstream temperature. Equation 10.3 represents an algebraic relation between the effort T and the flow \dot{H} . Consequently, Gawthrop and Smith consider it to be the constitutive equation of a thermal resistor with the special features that

- the flow does not depend on an effort difference,
- the resistor is modulated by the mass flow from the hydraulic domain.

Both properties are expressed in Figure 10.3. Since the right side bond connected to the 1-junction is activated, it is not the temperature *difference* that is used in the

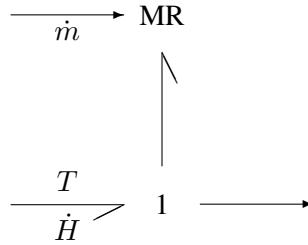


Fig. 10.3 Pseudo bond graph fragment representing an enthalpy flow through a short, insulated pipe segment (Gawthrop and Smith, 1996)

constitutive equation of the R element, but the upstream temperature. The activated bond propagates the enthalpy flow \dot{H} .

Accumulation of Enthalpy

The enthalpy flows entering and leaving the tank, \dot{H}_i or \dot{H}_o , cause a temperature change inside the tank. Let T denote the temperature inside the tank, then

$$H = c_p m T , \tag{10.4}$$

and

$$\dot{H} = \dot{H}_i + \dot{Q} - \dot{H}_o . \tag{10.5}$$

Equation 10.4 relates the effort T with the generalised displacement H and can therefore be considered the constitutive equation of a thermal capacitor controlled by the hydraulic quantity m (Figure 10.4).

In Section 2.5.3, it has been pointed out that modulated energy stores cannot exist because they violate the principle of energy conservation. However, since in pseudo bond graphs the product of the two variables associated with a bond does not necessarily equal the amount of power flowing along the bond and because the C element is not energy conservative, modulated C elements may be admitted in *pseudo* bond graphs. The C element introduced by Equation 10.4 is only an *accumulator* of enthalpy.

A Pseudo Bond Graph Model Accounting for Mass and Enthalpy Accumulation

Both effects, namely the accumulation of mass and the accumulation of heat, can be expressed by the pseudo bond graph in Figure 10.5.

From the pseudo bond graph of Figure 10.5, the following sorted set equations can be derived in the same manner as equations are derived from a true bond graph

$$\dot{m}_o = f_R (p) \tag{10.6a}$$

$$\dot{m} = \dot{m}_i - \dot{m}_o \tag{10.6b}$$

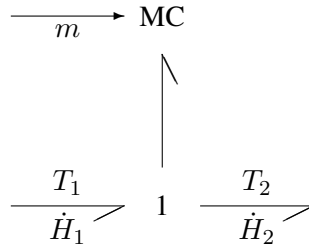


Fig. 10.4 Pseudo bond graph representation of a thermal accumulator (Gawthrop and Smith, 1996)

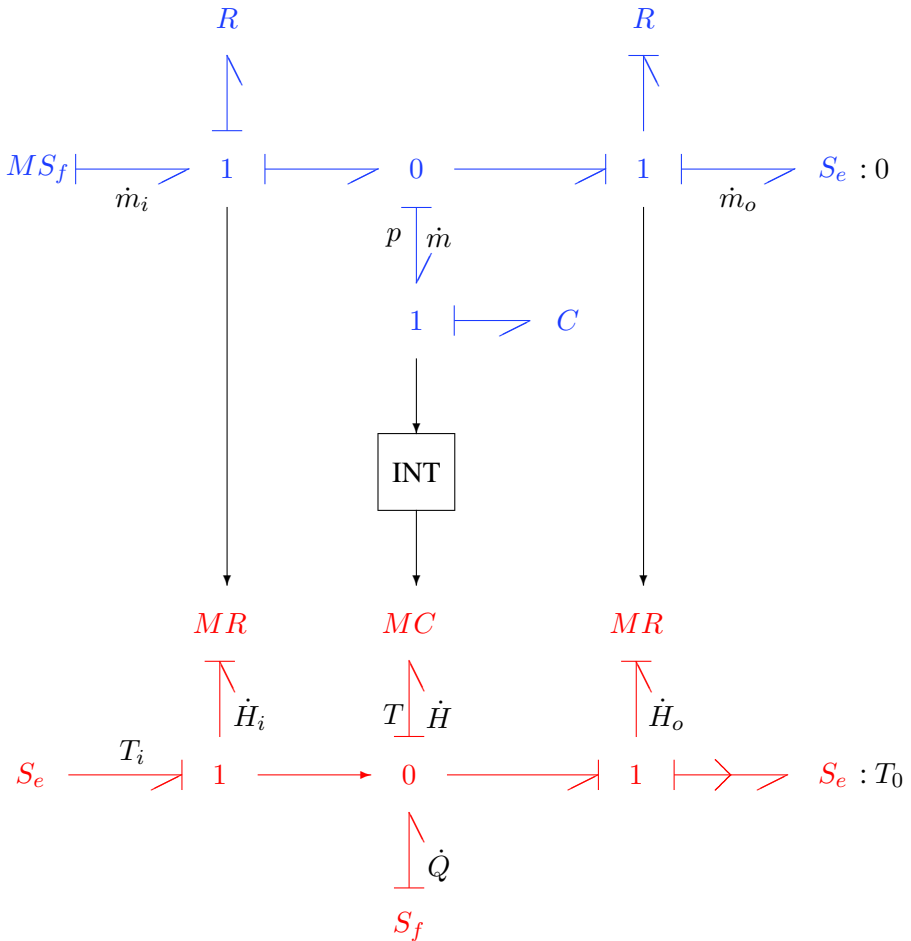


Fig. 10.5 Pseudo bond graph of the stirred tank in Figure 10.1 (Gawthrop and Smith, 1996)

$$\dot{p} = \frac{1}{\left(\frac{A}{g}\right)} \dot{m} \quad (10.6c)$$

$$\dot{H}_i = (c_p \dot{m}_i) T_i \quad (10.6d)$$

$$\dot{H}_o = (c_p \dot{m}_o) T \quad (10.6e)$$

$$\dot{H} = \dot{H}_i - \dot{H}_o + \dot{Q} \quad (10.6f)$$

$$\dot{T} = \frac{1}{c_p m} \dot{H}, \quad (10.6g)$$

where the characteristic of the hydraulic inlet and outlet resistors are given by the nonlinear function f_R . By elimination of algebraic quantities, two state equations can be established. If the pressure, p , and the temperature, T , are chosen as state variables, then the state equations are

$$\dot{p} = \frac{g}{A} [-f_R(p) + \dot{m}_i] \quad (10.7a)$$

$$\dot{T} = \frac{g}{c_p A p} [c_p \dot{m}_i T_i - c_p f_R(p) T + \dot{Q}]. \quad (10.7b)$$

Alternatively, the mass, m , and the enthalpy, H , inside the tank could be chosen as state variables as well.

A feature of this approach is that hydraulic and thermal aspects can be represented by two *separate* pseudo bond graphs that are linked by activated bonds from the hydraulic to the thermal part. In contrast, in the next section we will use multiport elements having hydraulic and thermal ports.

10.1.2 Pseudo Bond Graph of a Variable Pneumatic Control Volume

Let us modify the example of a stirred tank considered in the previous section a bit and develop a pseudo bond graph for a tank that stores an amount of gas and that is topped by a moving piston as shown in Figure 10.6. In practice, it could be a chamber of a pneumatic cylinder.

In this example, a mass flow \dot{m}_i of density ϱ_i enters into the control volume carrying internal and kinetic energy. There is also the convection of momentum bound to the mass flow. The two mass flows entering and leaving the chamber cause a change of momentum inside the control volume. Again, we will assume that there is an instantaneous mixing inside the tank yielding the same temperature T , the same pressure p , the same density ϱ everywhere in the tank and a uniformly distributed velocity v . The balances for energy, mass and momentum then read

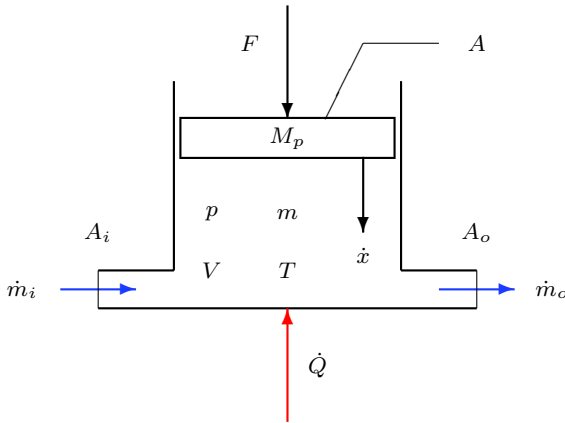


Fig. 10.6 Variable pneumatic control volume

$$\dot{U} = (h_i + \frac{1}{2}v_i^2)\dot{m}_i - (h_o + \frac{1}{2}v_o^2)\dot{m}_o - p\dot{V} + \dot{Q} \quad (10.8a)$$

$$\dot{m} = \dot{m}_i - \dot{m}_o \quad (10.8b)$$

$$\frac{d}{dt}(mv) = A_i p_i - A_o p_o + F + \dot{m}_i v_i - \dot{m}_o v_o . \quad (10.8c)$$

In Equation 10.8a, U denotes the internal energy in the control volume

$$U = mc_v T , \quad (10.9)$$

where c_v is the specific heat at constant volume. The velocity of the gas entering the control volume is v_i . The gas leaves the tank at velocity v_o . Moreover, in this example,

$$\dot{V} = A\dot{x} . \quad (10.10)$$

It is assumed that inertia effects can be neglected as well as the convected kinetic energy in comparison to the convected enthalpy. Then, only the reduced energy equation

$$\dot{U} = h_i \dot{m}_i - h_o \dot{m}_o - p\dot{V} + \dot{Q} \quad (10.11)$$

and the mass balance, Equation 10.8b, are to be taken into account.

In addition, there is the well known ideal gas law

$$pV = mRT , \quad (10.12)$$

where R is the *special* gas constant depending on the gas in use (For instance, for air, this constant has the numerical value $287Nm/(kgK)$).

Karnopp suggests to consider the Equations 10.11, 10.8b, 10.9 and the ideal gas law 10.12 as the constitutive equations of a three-port C element that accumulates

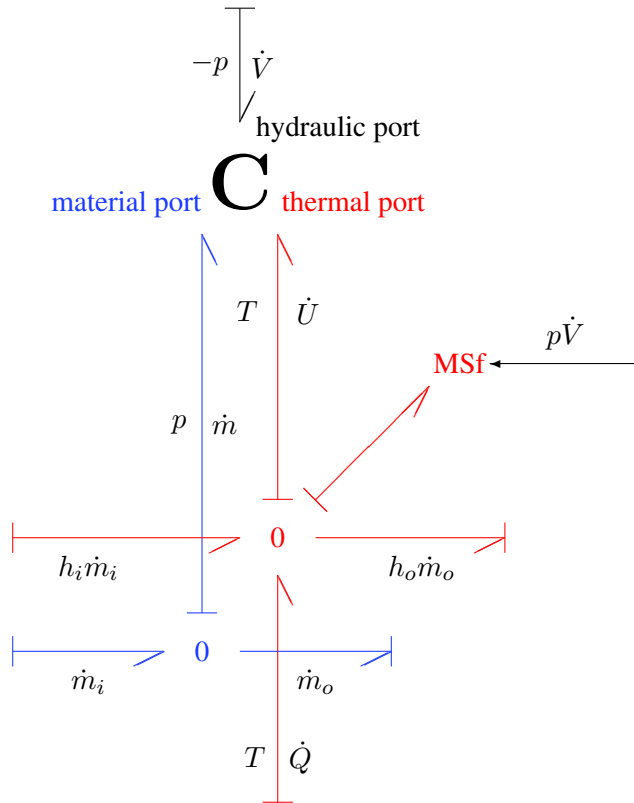


Fig. 10.7 Pseudo bond graph representation of a variable control volume (Karnopp, 1979)

energy, mass and volume (Figure 10.7). Since it is *not* energy conservative, he calls it the *thermodynamic accumulator* [22]. For this C element, the energy is not a function of the state variables from which the constitutive equations can be derived by partial differentiation as in the case of energy conservative C fields. In this pseudo bond graph representation, the energy itself is a state variable. Since at the hydraulic $(-p, \dot{V})$ -port the product of effort and flow equals the instantaneous power, the thermodynamic accumulator can be linked with a bond graph of a mechanical subsystem via a transformer.

In the pseudo bond graph of Figure 10.7, thermal bonds are highlighted by thin lines. Summation of flows at the 0-junction connected to the thermal (T, \dot{U}) -port of the accumulator yields the energy balance of Equation 10.11. The second 0-junction connected to the material (p, \dot{m}) -port represents the mass balance of Equation 10.8b. The controlled flow source attached to thermal 0-junctions accounts for the mechanical work performed by the compressible gas on the control volume surface.

Even though the thermal accumulator is not an energy conservative C field (the energy balance is expressed separately by the thermal 0-junction), it operates like

a C energy store. Given integral causality at all three ports, it takes in the rates \dot{U} , \dot{m} and \dot{V} , produces the respective states and, by using the constitutive equations, outputs the efforts, viz. the pressure, p , and the temperature T . Both quantities are given by reformulation of Equations 10.9 and 10.12.

$$p = \frac{R}{c_v} \frac{U}{V} \quad (10.13a)$$

$$T = \frac{1}{c_v} \frac{U}{m} \quad (10.13b)$$

The initial values of the state variables are related via the ideal gas law, Equation 10.12 and Equation 10.9, for the internal energy U .

$$p_0 V_0 = m_0 R T_0 \quad (10.14a)$$

$$U_0 = m_0 c_v T_0 \quad (10.14b)$$

This model also holds for control volumes containing a two-phase mixture of a fluid, e.g. a steam boiler [22]. Assume that the fluid is water and let m_w be the mass of the water, m_s the mass of the steam and m the total mass of water and steam. If the total mass and the total internal energy in the steam boiler of constant volume V_0 are known, then

$$m = m_w + m_s \quad (10.15a)$$

$$V_0 = v_w m_w + v_s m_s \quad (10.15b)$$

$$U = u_w m_w + u_s m_s, \quad (10.15c)$$

where v_w, v_s are the specific volumes of water and steam for a unit mass and u_w, u_s are the corresponding internal energies per unit mass. There are tables for saturated steam that provide values for volumes and internal energies for given values of temperature or pressure. With tabulated values for v_w, v_s, u_w, u_s , the masses m_w, m_s can be computed by solving Equations 10.15b and 10.15c. The procedure is to be repeated until both masses satisfy Equation 10.15a. This way, the entry into a table, viz. the temperature, or the pressure is known. The other variable can be determined in the same manner. Of course, during simulation of a thermodynamic system, this iteration process must be carried out after each integration step.

10.1.3 Pseudo Bond Graph of a Compressible Fluid Flow Through an Orifice

In pneumatic systems, a compressible fluid flow passes orifices as depicted in Figure 10.8 on its way between chambers. In a pseudo bond graph, the flow of a compressible fluid through an orifice can be represented by the 4-port resistor shown in

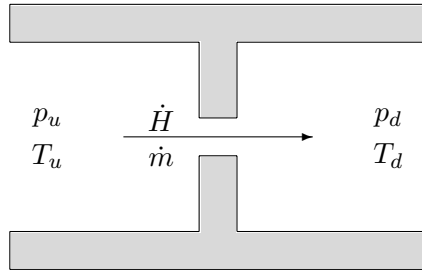


Fig. 10.8 Compressible fluid flow through an orifice

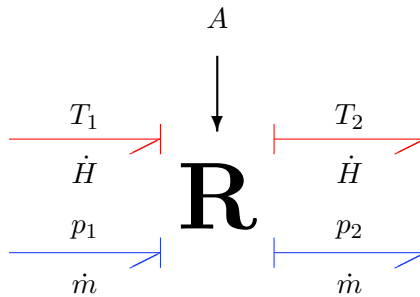


Fig. 10.9 Pseudo bond graph for a compressible fluid flow through an orifice (Karnopp, 1979)

Figure 10.9 with fixed causalities independent of the direction of the fluid flow [22] (A denotes the cross sectional area of the orifice).

If this R element is used to connect the accumulators of two chambers, then pressures and temperatures are determined in the accumulator models while the R element provides the mass flows and enthalpy flows. In Figure 10.8, the variables p_u, T_u denote upstream pressure and upstream temperature. Corresponding downstream quantities are marked by a subscript d . In order to compute the mass and the enthalpy flows, first, it must be determined which of the pressures at both sides of the orifice is the upstream one.

$$\text{If } p_1 > p_2, \text{ then } p_u = p_1, T_u = T_1, p_d = p_2. \quad (10.16a)$$

$$\text{If } p_1 < p_2, \text{ then } p_u = p_2, T_u = T_2, p_d = p_1. \quad (10.16b)$$

The mass flow \dot{m} depends on the ratio

$$r := \frac{p_d}{p_u}. \quad (10.17)$$

This ratio is to be compared with the critical value

$$r_{crit} := \left(\frac{2}{\kappa + 1} \right)^{\frac{\kappa}{\kappa - 1}}, \quad (10.18)$$

where $\kappa := c_p/c_v$ ².

$$\text{If } r < r_{crit}, \quad \text{then } r = r_{crit}^3. \quad (10.19)$$

After these steps, the absolute value of the mass flow, $|\dot{m}|$, through the orifice is

$$|\dot{m}| = C_q \times C_m \times A \times \frac{p_u}{T_u^{1/2}}, \quad (10.20)$$

where

$$C_m = \left(\frac{2\kappa}{R(\kappa - 1)} \right)^{1/2} \times \left(r^{2/\kappa} - r^{(\kappa+1)/\kappa} \right)^{1/2} \quad (10.21)$$

(cf., e.g. [32]). The discharge coefficient, C_q , takes into account that the downstream pressure, p_d , is used and not the pressure in the vena contracta. It also depends on the geometry of the orifice. Finally, the mass flow through an orifice is thus

$$\dot{m} = |\dot{m}| \times \text{sign}(p_1 - p_2). \quad (10.22)$$

Once the mass flow is known, the enthalpy flow is

$$\dot{H} = c_p T_u \dot{m}. \quad (10.23)$$

Thus, the R element in Figure 10.9 is completely described. Like the previously introduced thermodynamic accumulator, it is a basic building block. Thoma uses the name RECO (Resistor for Convection) for this enthalpy flow conserving four port resistor describing the basic phenomenon of compressible fluid flow through a restricted passage [37, 38].

10.1.4 Pseudo Bond Graph of a Pneumatic Bridge Circuit

With both elements, the accumulator and the 4-port resistor, pseudo bond graphs can be conveniently constructed for a number of practical thermodynamic problems. For illustration, we come back to the typical example of a hydraulic bridge circuit with four displacement modulated orifices and a double acting cylinder in the load diagonal considered in Section 4.1. Such a bridge circuit results in modelling four-way

² For air, the value is $r_{crit} = 0,53$

³ In this case, the flow is said to be *choked*. A reduction of the downstream pressure p_d has no effect on the mass flow.

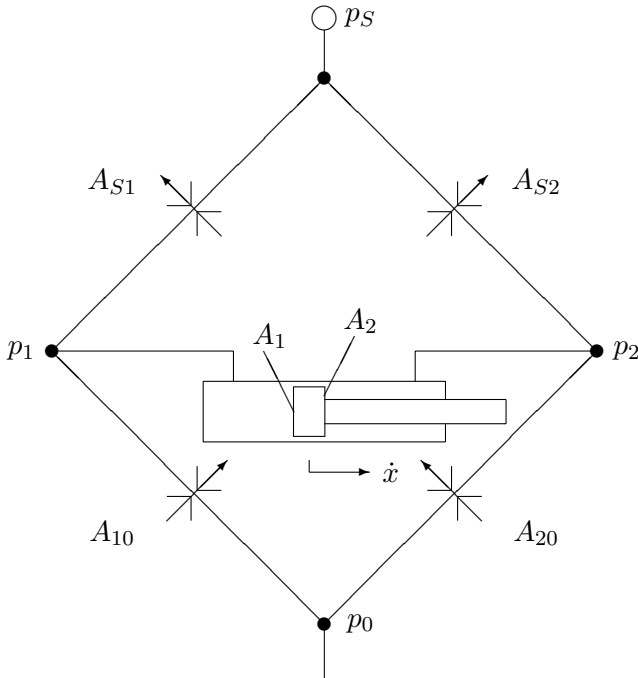


Fig. 10.10 Bridge circuit with a pneumatic single rod cylinder in the load diagonal

valves that control the amount and the direction of a fluid flow into and out of a load cylinder. In this section, we will not assume hydraulic but pneumatic components. The circuit schematic with a single rod cylinder is shown again in Figure 10.10. If each chamber of the cylinder is represented by an accumulator and each orifice by the 4-port R element (Figure 10.9), then the connection of these elements according to the structure of the bridge circuit yields a pseudo bond graph of a pneumatic circuit representing a four-way valve connected to a double acting cylinder.

Figure 10.11 clearly shows that the pseudo bond graph of the pneumatic subsystem can be connected to a true bond graph of a mechanical subsystem via the hydraulic $(-p, \dot{V})$ port of the accumulators.

The approach introduced by Karnopp has the advantage that mechanical properties of a thermodynamic system can be modelled by a true bond graph in a conventional manner, while thermodynamic aspects are intuitively and clearly represented by a pseudo bond graph making use of variables common in thermodynamics.

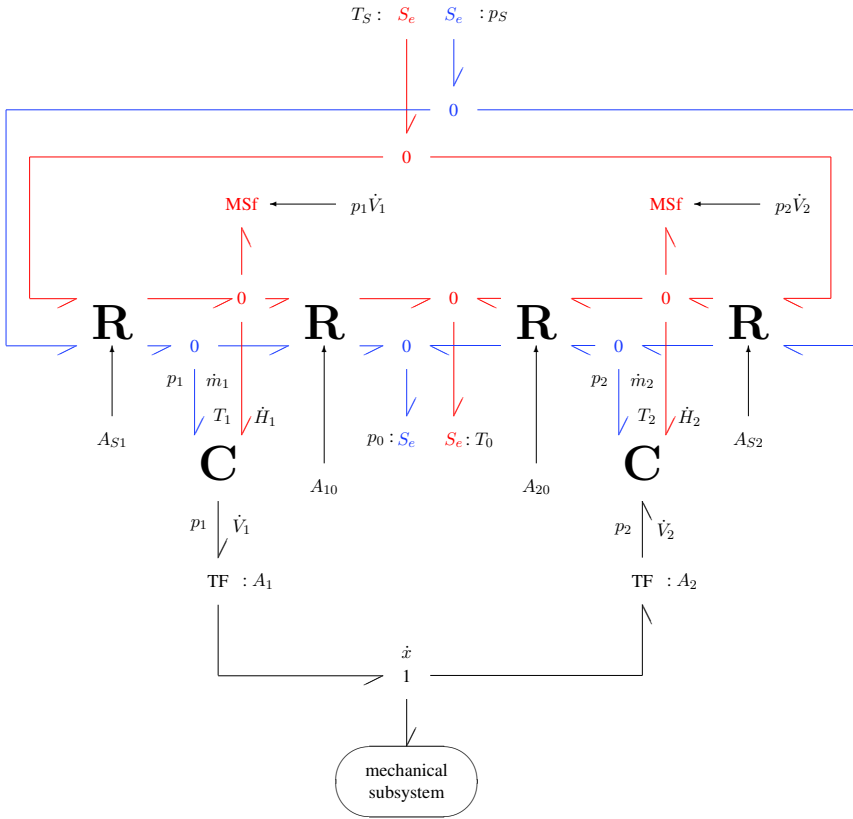


Fig. 10.11 Pseudo bond graph of the pneumatic bridge circuit in Figure 10.10

10.2 True Bond Graph Models of Thermodynamic Systems

As has been shown, pseudo bond graphs are a convenient approach to the modelling of thermodynamic systems and have some clear advantages. Nevertheless, another option is to keep the fundamental requirement that the product of variables associated with a bond must equal the instantaneous total power transmitted between ports. In the following, the true bond graph approach shall be considered.

10.2.1 True Bond Graph of a Variable Pneumatic Control Volume

Again, we consider the variable pneumatic control volume depicted in Figure 10.6. For simplicity, we assume that the inlet of cross sectional area A_i is closed. That is,

the fluid can only leave the tank through the cross sectional area A_o . Otherwise, we had to account for the entropy production that results when two amounts of fluid of different temperature mix. Furthermore, let us assume that the kinetic energy in the tank can be neglected. Then, the stored energy equals the internal energy U . The latter is a function, f_U , of the quantities entropy S , volume V and the number of moles, N , of a substance.

$$U = f_U(S, V, N) \quad (10.24)$$

The quantities entropy, volume and the amount of substance are also called *extensive* variables because they are proportional to the extent of a thermodynamic system (cf., e.g. [8], p. 96, or [11], p. 360)). These *extensive* quantities are the independent variables of the thermodynamic energy function f_U .

The total differential of the internal energy, U , is

$$dU = \frac{\partial U}{\partial S} dS + \frac{\partial U}{\partial V} dV + \frac{\partial U}{\partial N} dN . \quad (10.25)$$

In thermodynamics, the following definitions are used.

$$T := \left. \frac{\partial U}{\partial S} \right|_{V, N = \text{const.}} \quad (10.26a)$$

$$-p := \left. \frac{\partial U}{\partial V} \right|_{S, \mu = \text{const.}} \quad (10.26b)$$

$$\mu := \left. \frac{\partial U}{\partial N} \right|_{S, V = \text{const.}} , \quad (10.26c)$$

where μ denotes the so-called *chemical potential* (cf. Table 2.4). These physical quantities defined as a partial derivative of the internal energy with respect to an extensive variable are called *intensive* variables in thermodynamics since they do not depend on the extent of the thermodynamic system. With the above definitions, the change of internal energy, dU , reads

$$dU = TdS - pdV + \mu dN . \quad (10.27)$$

Equation 10.27 is Gibbs' well known fundamental equation. If the state of the gas in the control volume *slowly* changes, then the power balance reads

$$\frac{dU}{dt} = T\dot{S} - p\dot{V} + \mu\dot{N} . \quad (10.28)$$

The first term in this power balance can be considered as the power of a heat flow into the control volume. The second term accounts for the work performed by the gas on the control volume surface when the control volume changes. The third term is the power associated with a mass flow into the control volume⁴. The heat flow is

⁴ If a compressible fluid is composed of k chemical substances of N_i ($i = 1, \dots, k$) number of moles and if μ_i denotes the power conjugate chemical potential associated with N_i , then

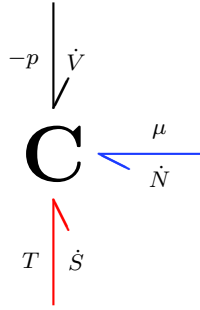


Fig. 10.12 True bond graph of a variable pneumatic control volume (Breedveld, 1985)

due to the conduction across the control surface and the transport of entropy along with the mass flow.

In his Ph.D. thesis ([8], pp. 106–107), Breedveld proposes to represent Equation 10.28 by a 3-port C field with a thermal, a hydraulic and a material port (Figure 10.12).

Due to Equations 10.26a–10.26c, the introduced 3-port C field satisfies Maxwell's reciprocity conditions. That is, in contrast to the thermodynamic accumulator in pseudo bond graphs, it is energy conservative.

$$\frac{\partial T}{\partial V} = \frac{\partial}{\partial V} \left(\frac{\partial U}{\partial S} \right) = \frac{\partial}{\partial S} \left(\frac{\partial U}{\partial V} \right) = \frac{\partial(-p)}{\partial S} \quad (10.29a)$$

$$\frac{\partial T}{\partial N} = \frac{\partial}{\partial N} \left(\frac{\partial U}{\partial S} \right) = \frac{\partial}{\partial S} \left(\frac{\partial U}{\partial N} \right) = \frac{\partial \mu}{\partial S} \quad (10.29b)$$

$$\frac{\partial \mu}{\partial V} = \frac{\partial}{\partial V} \left(\frac{\partial U}{\partial N} \right) = \frac{\partial}{\partial N} \left(\frac{\partial U}{\partial V} \right) = \frac{\partial(-p)}{\partial N} \quad (10.29c)$$

Equation 10.28 can also be considered as the scalar product of the two vectors $\mathbf{e} := (T, -p, \mu)^t$ and $\mathbf{f} := (\dot{S}, \dot{V}, \dot{N})^t$ (Since T already denotes the temperature, the transpose of a vector is indicated by the lower case letter t). Thus, we have two vector quantities, effort and flow, of which the product is the total power into the control volume.

$$\dot{U} = \mathbf{e}^t \cdot \mathbf{f} \quad (10.30)$$

Figure 10.13 shows the corresponding multibond graph representation. Thus, the total energy and mass transport into a control volume can be expressed by the product of two vector quantities. However, not all of their components can be directly measured.

the product $\mu \times N$ in Equation 10.28 must be replaced by the scalar product of vectors $\boldsymbol{\mu}^t \dot{\mathbf{N}}$ where $\boldsymbol{\mu}^t = (\mu_1, \dots, \mu_k)$ and $\dot{\mathbf{N}}^t = (\dot{N}_1, \dots, \dot{N}_k)$. The superscript t of a vector denotes its transpose.

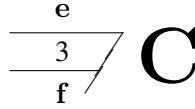


Fig. 10.13 Multibond graph representation of variable pneumatic control volume

Constitutive Equations of the 3-Port C Field

In the following, the energy function, f_U , and the constitutive equations of the 3-port C field, representing a variable pneumatic control volume are derived assuming an *ideal gas*. The ideal gas law is

$$pV = NRT, \quad (10.31)$$

where R is the *universal* gas constant ($R = 8.3145 \text{ Jmole}^{-1}\text{K}^{-1}$) and

$$R = c_p - c_v. \quad (10.32)$$

The internal energy is

$$U = Nc_v(T - T_0). \quad (10.33)$$

If molar quantities (quantities divided by the number, N , of moles) are denoted by a small letter and a superscript m , then Gibbs fundamental equation reads

$$u^m = Ts^m - pv^m + \mu. \quad (10.34)$$

With Equations 10.31–10.33, we obtain for the chemical potential μ

$$\mu = (c_p - s^m)T - c_vT_0. \quad (10.35)$$

Substituting Equations 10.33, 10.31, 10.35 into Equation 10.27 yields

$$c_vT dN + c_vN dT = T dS - \frac{NRT}{V} dV + (c_p - s^m)T dN, \quad (10.36)$$

which can be written in the form

$$c_v \frac{dT}{T} = \frac{dS}{N} - S \frac{dN}{N^2} + R \frac{dN}{N} - R \frac{dV}{V}. \quad (10.37)$$

Integration yields

$$c_v \ln \frac{T}{T_0} = \left(\frac{S}{N} \right) - \frac{S_0}{N_0} + R \ln \frac{N}{N_0} - R \ln \frac{V}{V_0}, \quad (10.38)$$

where initial values at time t_0 are indicated by a subscript 0. Thus, for the temperature T , we obtain

$$T = T_0 \left(\frac{V_0 N}{N_0 V} \right)^{R/c_v} \exp \left(\frac{S}{N c_v} - \frac{S_0}{N_0 c_v} \right). \quad (10.39)$$

Substituting this expression into the ideal gas law, Equation 10.31, yields for the pressure p

$$\begin{aligned} p &= p_0 \left(\frac{V_0 N}{N_0 V} \right) \frac{T}{T_0} \\ &= p_0 \left(\frac{V_0 N}{N_0 V} \right)^\kappa \exp \left(\frac{S}{N c_v} - \frac{S_0}{N_0 c_v} \right), \end{aligned} \quad (10.40)$$

where $\kappa = c_p/c_v$. Finally, substituting the expression for the temperature T into Equation 10.35 gives for the chemical potential μ

$$\begin{aligned} \mu &= (c_p - s^m) T_0 \left(\frac{V N_0}{N V_0} \right)^{-R/c_v} \exp \left(\frac{S}{N c_v} - \frac{S_0}{N_0 c_v} \right) \\ &\quad - c_v T_0 \end{aligned} \quad (10.41)$$

and $\mu_0 = (R - s_0^m) T_0$.

With Equations 10.33 and 10.39, the energy function f_U is

$$U = N c_v T_0 \left[\left(\frac{V_0 N}{N_0 V} \right)^{R/c_v} \exp \left(\frac{S}{N c_v} - \frac{S_0}{N_0 c_v} \right) - 1 \right]. \quad (10.42)$$

If molar quantities are used in the constitutive Equations 10.39, 10.40, 10.42 and if $S_0 = 0$ is assumed, then these are identical to those equations given in ([24], p. 453).

Example: Gas Tank of Variable Volume with an Outlet and Heat Influx

If the influx of heat into the control volume due to conduction (\dot{Q} in Figure 10.6) is modelled by an RS element, then a gas filled tank of variable volume with an outlet can be represented by the true bond graph depicted in Figure 10.14.

In this bond graph, the product, $-h^m \dot{N}$, denotes the power, \dot{E}_{conv} , leaving the control volume by convection through the outlet. As can be seen from the bond graph, this power is composed of the part, $\mu \dot{N}$, due to the mass flow and the part, $T s^m \dot{N}$, due to the convection of entropy.

$$\dot{E}_{conv} = -h^m \dot{N} = -(\mu \dot{N} + T s^m \dot{N}) \quad (10.43)$$

If there is no influx of heat due to conduction, then the flow balance at the lower left 0-junction in the bond graph of Figure 10.14 reduces to

$$\dot{S} = s^m \dot{N}. \quad (10.44)$$

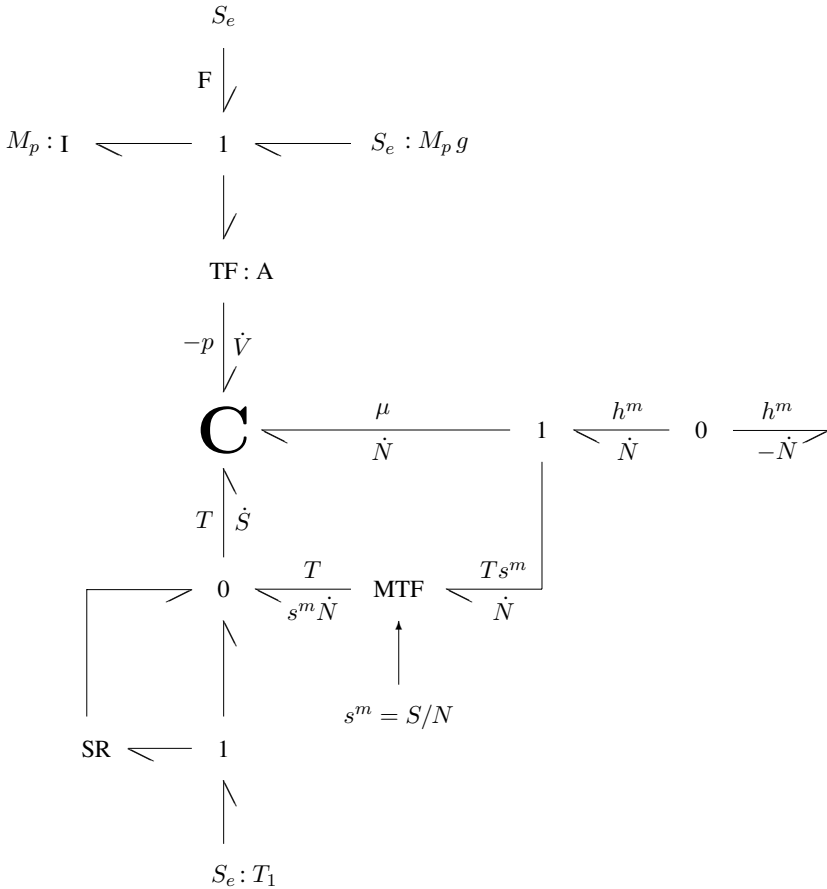


Fig. 10.14 True bond graph of a variable pneumatic control volume with an outlet and heat influx (cf. Figure 10.6)

Hence,

$$\mu \dot{N} + T \dot{S} = \mu \dot{N} + T s^m \dot{N} = (\mu + T s^m) \dot{N} = h^m \dot{N}. \quad (10.45)$$

In this case, the 3-port C field and the modulated transformer can be combined into a 2-port C element as depicted in Figure 10.15.

If the mole flow, \dot{N} , is converted into a mass flow, \dot{m} , then we get the bond graph given by Karnopp and Rosenberg ([23], p. 377) for the isentropic case (Figure 10.16) in which M denotes the molar mass of the gas. With Equation 10.34, $h^m = \mu + T s^m$ takes the form $h^m = u^m + p v^m$. Consequently, $h = u + p/\rho$.

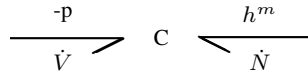


Fig. 10.15 Two-port C energy store representing a pneumatic control volume without heat conduction

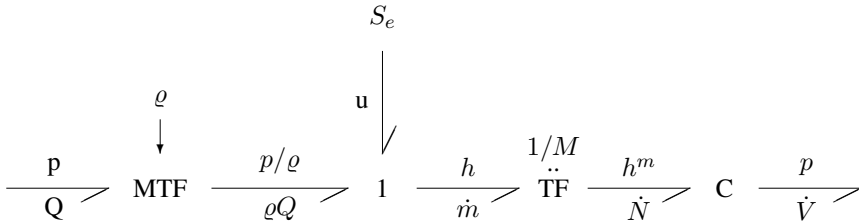


Fig. 10.16 Isentropic compression of a gas in a variable control volume (Karnopp and Rosenberg, 1975)

Special Cases

The 3-port C field introduced by Breedveld for representation of a compressible fluid in a variable control volume, in many cases, reduces to a 2-port energy store. For instance, for *closed* systems with no mass transport into and of out of the control volume, i. e. $\dot{N} = 0$, the material port disappears. If the volume remains constant, then the hydraulic $(-p, \dot{V})$ port is not needed.

The Isothermal Case

In the isothermal case ($\dot{T} = 0$), the power $T \times \dot{S}$ at the thermal port does not vanish. Nevertheless, it is common to drop this port. Thus, the two-port C element no longer represents the storage of internal energy, but the storage of the *Helmholtz free energy* $F := U - T \times S$. The remaining bonds adjacent to the C element represent the flow of free energy.

$$\begin{aligned}
 -p \dot{V} + \mu \dot{N} &= \dot{U} - T \dot{S} \\
 &= \dot{U} - T \dot{S} - \dot{T} S \\
 &= \dot{U} - \frac{d}{dt}(T S) \\
 &= \dot{F}
 \end{aligned}
 \tag{10.46}$$

The Isobaric Case

Similarly, in the isobaric case ($\dot{p} = 0$), the two-port C element represents the storage of enthalpy and the remaining adjacent bonds express an enthalpy flux into the control volume.

$$\begin{aligned}
 T \dot{S} + \mu \dot{N} &= \dot{U} + p \dot{V} \\
 &= \dot{U} + p \dot{V} + \dot{p} V \\
 &= \dot{U} + \frac{d}{dt}(pV) \\
 &= \dot{H}
 \end{aligned} \tag{10.47}$$

The Helmholtz free energy is the result of a Legendre transformation (cf. Section 5.3) of the internal energy U with respect to the entropy S . The enthalpy H is obtained by a Legendre transformation U with respect to the volume V . The chemical potential μ is the molar Gibbs free energy, $g^m := G/N$, and G is defined by a double Legendre transformation of U . That is, $G := U + pV - TS$.

10.2.2 True Bond Graph of a Pneumatic Outlet Orifice

As we know that the storage of internal energy in a pneumatic control volume can be represented by an energy conservative 3-port C energy store, we want to address the question of how an orifice in the outlet can be represented in a true bond graph. Such an outlet orifice neither stores mass nor energy. Thus, the entering mole flow, \dot{N} , equals the one leaving the orifice. Likewise, the power into the orifice is equal to the one leaving it. Let us assume a steady lossless flow out of the chamber and that the kinetic energy inside the chamber can be neglected. Let h_{CV}^m denote the specific molar enthalpy in the control volume, CV, h_c^m the molar enthalpy in the vena contracta and M the molecular mass of the gas. Then, the power balance reads

$$h_{CV}^m \dot{N} = (h_c^m + \frac{1}{2} M v_c^2) \dot{N}, \tag{10.48}$$

where v_c denotes the velocity of the gas in the vena contracta. With the assumption $h_{CV}^m \geq h_c^m$, the velocity v_c is

$$v_c = \sqrt{\frac{2}{M} (h_{CV}^m - h_c^m)}. \tag{10.49}$$

With this result, the mole flow, \dot{N} , through the cross sectional area, A_c , of the vena contracta is

$$\dot{N} = \frac{\varrho}{M} A_c \sqrt{\frac{2}{M} (h_{CV}^m - h_c^m)}. \tag{10.50}$$

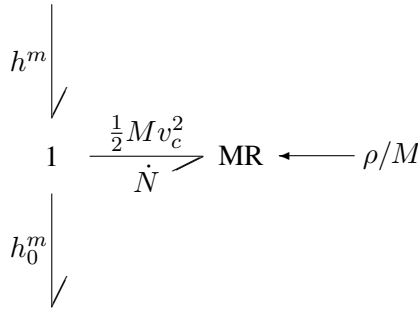


Fig. 10.17 True bond graph of a pneumatic outlet orifice (Beaman and Breedveld, 1988)

Usually, it is assumed that all kinetic energy in the restriction is converted into heat and that the specific molar enthalpy, h_c^m , in the vena contracta equals the ambient molar enthalpy h_o^m . Thus, Equation 10.50 can be considered the constitutive equation of a resistor modulated by the fluid density ρ (Figure 10.17).

In contrast to the pseudo bond graph model of Karnopp, the true bond graph model given by Beaman and Breedveld [2] does not use the ratio of upstream and downstream pressure and the upstream temperature, but instead enthalpy. Therefore, its equation looks simpler. Nonetheless, as Beaman and Breedveld note, it can be shown that both models are equivalent in the case of an ideal gas. From a practical point of view, it may be considered a disadvantage that pressures and temperature as measurable quantities are not directly used in this model.

10.2.3 Further True Bond Graph Approaches to the Modelling of Thermodynamic Systems

According to Equation 10.48, the power of a fluid line through a cross sectional area, A , is

$$(h^m + \frac{1}{2}Mv^2)\dot{N} = (h + \frac{1}{2}v^2)\dot{m} = (\frac{p}{\rho} + u + \frac{1}{2}v^2)\dot{m} . \tag{10.51}$$

The rate of kinetic energy, \dot{E}_{kin} , over the entire cross sectional area, A , is

$$\dot{E}_{kin} = \int_0^A \frac{1}{2}\rho v^2 v d\tilde{A} = \alpha \frac{1}{2}\rho \bar{v}^2 Q , \tag{10.52}$$

where \bar{v} is the mean velocity over the cross sectional area and α , a coefficient that depends on the shape of the velocity profile. Assuming that the enthalpy is constant over the cross section, then with $h_0 := h + \alpha\bar{v}/2$, the so-called *stagnation enthalpy* [10], the total power through a cross sectional area A is the product of two variables,

the stagnation enthalpy h_0 and the mass flow \dot{m} . This result suggests to introduce them as power conjugate variables associated with the bonds in bond graphs of thermodynamic systems with compressible fluids. However, the description of the state of a pure substance requires two independent intensive variables, e.g. pressure and temperature. Taking this into account, Brown [10] associates *two* independent variables with each bond that describe its effort, namely h_0 and the pressure. He calls such a bond a *convection bond* and distinguishes it from a standard bond by an additional dashed line on the effort side. In his representation, h_0 is the proper effort. The additional variable p_0 qualifies the effort, as Brown terms it. The product $p_0 \times \dot{m}$ apparently is not power. The pressure, p_0 , however serves as the effort of that bond with regard to computational causality. That is, in Figure 10.18 the causal stroke indicates that the pressure, p_0 , is determined at port B, while the mass flow \dot{m} is computed at port A. Using convection bonds, Brown represents the compressible fluid flow through a restriction by means of an extension of the power conserving, entropy generating RS element as depicted in Figure 10.19. This figure clearly shows that the stagnation enthalpy, h_0 , and the total power remain constant, while a pressure drop across the orifice occurs. The entropy, however, is not conserved⁵.

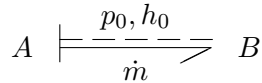


Fig. 10.18 Convection bond (Brown, 1991)

⁵ The rate of entropy generation can be determined in the following manner [10]. Assume a steady flow of an ideal gas through a nozzle. Since the enthalpy remains constant, we write Gibbs' equation in the form

$$d(u^m + pv^m) = dh^m = Tds^m + v^m dp = 0. \quad (10.53)$$

Replacing the molar volume v^m by means of the ideal gas law yields

$$Tds^m + \frac{R}{p} dp = 0. \quad (10.54)$$

If index 1 denotes the upstream side and index 2 the throat of the nozzle, then the solution of Equation 10.54 reads

$$s_2^m - s_1^m = R \ln \left(\frac{p_1}{p_2} \right). \quad (10.55)$$

Thus, the rate of the entropy generation, \dot{S} , is

$$\dot{S} = (s_2^m - s_1^m)\dot{N} = \dot{N}R \ln \left(\frac{p_1}{p_2} \right) > 0. \quad (10.56)$$

The result is in agreement with what we expect. The enthalpy as a function of pressure and entropy remains constant, while the pressure across the nozzle decreases. Consequently, the entropy must increase.

$$\frac{\overline{p_{01}, h_0}}{\dot{m}} \text{ RS } \frac{\overline{p_{02}, h_0}}{\dot{m}}$$

Fig. 10.19 Compressible fluid flow through an orifice (Brown, 1991)

Before considering bond graph modelling of some phenomena in hydraulic systems, another approach to the modelling of thermodynamic systems by means of true bond graphs shall be briefly mentioned.

Instead of using one effort at a single bond that accounts for the different contributions to the total power of incompressible fluids, in 1992, Thoma [35] proposed to use 3-dimensional multibonds. The power conjugate pairs are hydrostatic pressure, p , and volume flow, \dot{V} , temperature, T , and entropy flow, \dot{S} , chemical potential, μ , and mass flow \dot{m} . The product of each pair of variables equals power and the sum of all three products is the product of specific enthalpy times the mass flow or the enthalpy flow \dot{H} . In his approach, Thoma assumes low fluid velocities such that the transported kinetic energy can be neglected. Moreover, since he does not consider phase changes in the fluid, he ignores the chemical potential and indicates this by an activation of the corresponding bond. Thus, this bond becomes a signal representing the mass flow, \dot{m} , only. Thoma also calls his 3-dimensional multibonds convection bonds. The development of convection bond graphs was influenced by previous work of Thoma and Atlan [36]. To the author's knowledge, it was Thoma who also coined the term *convection bond*.

Since restrictions neither store mass nor energy, they are represented by an R element with three bonds for the influx and another three bonds for the efflux of enthalpy (cf. Figure 10.20). Like for the four port pseudo bond graph element introduced by Karnopp (Figure 10.9), Thoma considers a compressible fluid flow and uses standard equations from thermodynamics. For an ideal gas, upstream and downstream pressures and upstream temperature are given by Equations 10.38 and 10.40. Volume flow and convected entropy flow apparently result from the specific volume and the specific entropy by multiplication with the mass flow, \dot{m} , through the restriction (cf. Equations 10.20 and 10.22). For the determination of upstream and downstream specific entropy, see [35].

In bond graph models of hydrostatic systems, the compressibility of the oil in dead spaces or chambers of fixed size is accounted for by a C element attached to the 0-junction representing the chamber pressure and the sum of flows entering into the chamber. In his multibond graph approach, Thoma represents volumes of constant size in the same manner (cf. Figure 10.20). The 0-junction displays the summation of mass flow, volume flow and entropy flows. The latter balance neglects the entropy that appear when mass flows of different temperature mix. These flow variables are inputs into the C element which returns the efforts, viz. the pressure and the temperature in the chamber (integral causality). Again, it is assumed that the fluid in the chamber is well mixed such that, instantaneously, a spatially uniform distribution of pressure and temperature is attained. The chemical potential, μ , is

not computed. The effort quantities are propagated by the 0-junction according to the role of a standard 0-junction. The pressure and the temperature can be computed in the following manner.

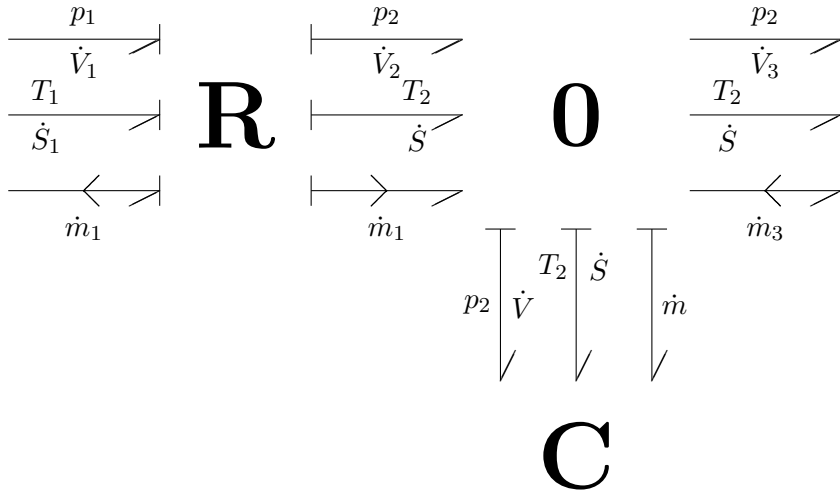


Fig. 10.20 Multibond graph representation of a pneumatic control volume with an inlet restriction according to Thoma, 1992

$$\dot{m} = \dot{m}_1 - \dot{m}_2 \tag{10.57}$$

$$v = \frac{V}{m} \tag{10.58}$$

$$\dot{S} = \dot{S}_2 - \dot{S}_3 \tag{10.59}$$

$$s = \frac{S}{m} \tag{10.60}$$

$$p = f_p(s, v) \tag{10.61}$$

$$T = f_T(s, v) , \tag{10.62}$$

where f_p denotes the relation in Equation 10.40. The function f_T is given by Equation 10.39. Since the control volume is assumed to be of fixed size, the volume flow is not needed for computation of the state of the fluid stored in the control volume.

The C element represents the storage of mass, entropy and volume. Its ports have integral causality. Hence, they provide the input variables of the R element and vice versa.

Like Karnopp’s pseudo bond graphs, the multibond graph representation proposed by Thoma clearly reflects the physical structure of a thermofluid system and uses quantities that are common in thermodynamics. An advantage from a practical

engineering point of view is that the output flows of the basic multiport R element and the output efforts of the multiport C element can be determined by using real data charts or look-up tables.

10.2.4 Bond Graph of a Double Acting Hydraulic Cylinder

In the previous sections, it has been shown how two fundamental phenomena, namely the accumulation of a gas in a control volume and a gas flow through a restriction, can be represented in pseudo bond graphs as well as in true bond graphs. In the following, it is assumed that the fluid is a hydraulic oil. Under the customary assumption of isothermal conditions, a true bond graph of a hydraulic double acting cylinder, a basic component in hydraulic system, will be developed. As usual, bonds representing the transport of thermal energy will be omitted. Strictly speaking, this means that bond graphs represent the storage and the transport of *free* energy. In the following, this difference will not be pointed out any further.

As a matter of fact, hydraulic oil is orders of magnitude less compressible than a gas. However, it not incompressible. Accordingly, in modelling hydraulic systems, it is customary to assume an incompressible flow through fixed and variable restrictions and to account for the oil compliance in chambers or dead spaces. Due to the use of a constant *mean* density, convection of mass and of (free) energy can be described by means of the volume flow contrary to gases. Following the nomenclature frequently used in hydraulics, volume flows will be denoted by the letter Q . Previously, the letter Q denoted an amount of heat following the convention in thermodynamics. In many cases, the dynamic pressure can be neglected in comparison to the hydrostatic pressure. Moreover, differences in height in a hydraulic system mostly are not big enough to contribute a significant gravity term to the hydrostatic pressure. Under these assumptions, the transported free energy can be sufficiently approximated by the hydrostatic energy.

Oil Compliance in a Chamber of Variable Volume

First, consider a chamber of constant volume and let us assume that the kinetic energy inside the control volume and energy losses due to turbulences at the inlet port can be neglected. Under these assumptions, the increase in hydrostatic pressure in the chamber due to volume flow into the chamber can be described by Hooke's law, as discussed in Section 2.7.2. Let ΔV denote the *decrease* of oil volume, V the constant volume of the chamber and B the bulk modulus⁶ of the oil. Then, the resulting *increase* in hydrostatic pressure is

⁶ For a well de-aerated oil, the bulk modulus depending on pressure and temperature is usually replaced with good accuracy by the typical value $B = 1,6 \times 10^{+9} Nm^{-2}$.

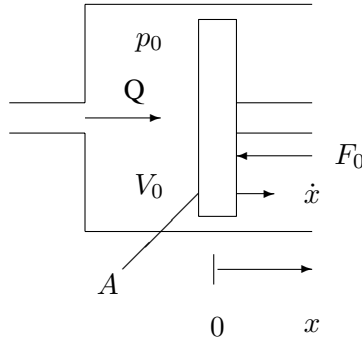


Fig. 10.21 Oil filled chamber of variable volume

$$\Delta p = B \frac{-\Delta V}{V} = \frac{1}{\left(\frac{V}{B}\right)} (-\Delta V) \tag{10.63}$$

(cf. Equation 2.82).

Since Equation 10.63 establishes a relation between the effort, p , and the generalised displacement, $(-\Delta V)$, it is considered the constitutive law of a (linear) 1-port C energy store with the hydraulic capacitance $C := V/B$. In bond graphs of hydraulic or acoustic systems, such a C element is attached to the 0-junction representing the hydrostatic pressure in the chamber under consideration.

If this C element is used for representing oil compliance in a hydraulic cylinder chamber (Figure 10.21), then there is a problem because the oil filled chamber volume is not constant but rather depends on the position, x , of the piston.

Let V_0 denote the volume corresponding to an initial position $x = x_0$ and let A be the area of the piston. Then,

$$\Delta p = \frac{B}{V_0 + Ax} (-\Delta V) . \tag{10.64}$$

Accordingly to Equation 10.64, the factor $(V_0 + Ax)/B$ is sometimes conceived as a variable capacitance $C(x)$ depending on the displacement x . Some authors (see, e.g. [29]) express this dependency in bond graphs of hydraulic systems by adding a signal arrow to the C element indicating the position, x_p , of the piston as reproduced in Figure 10.22. Other authors [14] account for the piston’s position in the constitutive equation, but use the standard 1-port C energy store representation. In the latter case, the bond graph representation does not correspond to the mathematical model.

However, as pointed out in Section 2.5.3, there are no signal controlled energy stores because they violate the principle of energy conservation. Linearisation of Equation 10.64 does not help since piston displacements are not confined to small deviations from a constant position x_0 . A simple obvious remedy is to replace the varying piston position, x , by an average value. To overcome these problems, the

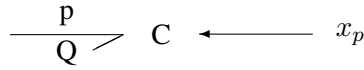


Fig. 10.22 Energetically incorrect representation of an oil filled chamber of variable volume

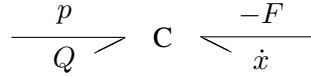


Fig. 10.23 Representation of an oil filled chamber of variable volume by a 2-port C element

author [4] proposed an energy conservative 2-port C element (Figure 10.23) with a hydraulic port and a mechanical port representing chambers of variable volume in hydraulic cylinders.

Constitutive Equations of the 2-Port C-Element of a Variable Chamber

For a derivation of the constitutive equation of the mechanical port of this C element, consider the chamber depicted in Figure 10.21. Let V_0 be the initial chamber volume and p_0 the hydrostatic chamber pressure at time t_0 . An external force F_0 is in equilibrium with the pressure force $A \times p_0$. Now, if a fluid flow enters the oil filled chamber at volume flow Q , then the oil in the chamber is compressed and the chamber pressure, p , increases (Part of this inflow of oil into the chamber may be due to leakage between the two chambers of a hydraulic cylinder through the clearance between piston and container wall). This means that a force $F > F_0$ is acting on the piston. As a result, simultaneously, the chamber volume increases. Let $\tilde{V} := \int_0^t Q d\tau$, $\Delta p := p - p_0$ and $\Delta F := F - F_0$. Then, the change of energy stored in the chamber is

$$dE = (\Delta p)d\tilde{V} - (\Delta F)dx. \quad (10.65)$$

Consider the position of the piston kept virtually fixed for an instant. That is, $dx = 0$. Then, the change of potential energy in the chamber is

$$\begin{aligned} dE &= (\Delta p)d\tilde{V} \\ &= (\Delta p)d(\tilde{V} - Ax) = (\Delta p)d(-\Delta V). \end{aligned} \quad (10.66)$$

Inserting Equation 10.64 and integrating with respect to $(-\Delta V)$, yields for $\Delta E(\tilde{V}, x) := E(\tilde{V}, x) - E_0$, that is, the deviation of the stored energy from its initial value E_0

$$\Delta E(\tilde{V}, x) = \frac{B}{V_0 + Ax} \frac{(\tilde{V} - Ax)^2}{2}. \quad (10.67)$$

Partial differentiation of the stored potential energy with respect to the piston displacement, x , gives the change in the force, ΔF , resulting in the piston's motion.

$$\begin{aligned}
-(\Delta F) &= \frac{\partial(\Delta E)}{\partial x} \\
&= B \left[-\frac{A}{(V_0 + Ax)^2} \frac{(\tilde{V} - Ax)^2}{2} + \right. \\
&\quad \left. \frac{1}{(V_0 + Ax)} (\tilde{V} - Ax)(-A) \right] \\
&= -A \left[\Delta p + \frac{1}{2B} (\Delta p)^2 \right] \tag{10.68}
\end{aligned}$$

Finally, the force F at the mechanical port of the 2-port C energy store is

$$\begin{aligned}
F &= A \left[(p_0 + (\Delta p)) + \frac{(\Delta p)^2}{2B} \right] \\
&= A \left[p + \frac{(\Delta p)^2}{2B} \right]. \tag{10.69}
\end{aligned}$$

Thus, Equations 10.64 and 10.69 are the constitutive equations of a 2-port C element describing the storage of potential energy in a chamber of variable volume. Partial differentiation of both equations prove that they fulfil Maxwell's reciprocity condition. That is, the 2-port C energy store introduced is actually energy conservative.

$$\frac{\partial(-F)}{\partial \tilde{V}} = AB \frac{V_0 + \tilde{V}}{(\tilde{V} + Ax)^2} = \frac{\partial(\Delta p)}{\partial x} = \frac{\partial p}{\partial x} \tag{10.70}$$

A remarkable difference in Equation 10.69 compared to the usual pressure force relation $F = A \times p$ in the case of an incompressible fluid is the quadratic term

$$A \frac{(\Delta p)^2}{2B} = \frac{AB}{2} \left(\frac{\tilde{V} - Ax}{V_0 + Ax} \right)^2.$$

It may be considered a force needed to balance an additional spring force acting on the piston due to the compressibility of the oil (Figure 10.24). For instance, if we consider the special case of no oil inflow, i. e. $\tilde{V} = 0$, we have for the nonlinear spring force

$$F_{spring} = \frac{AB}{2} \left(\frac{-Ax}{V_0 + Ax} \right)^2. \tag{10.71}$$

According to what is to be expected, the spring force takes large values if the oil filled chamber volume tended to zero ($x \rightarrow -V_0/A$).

It is interesting to note that, in practice, numerical values of the spring force are negligible. With the typical value $B = 1,6 \times 10^{+9} \text{ N/m}^2$ for mineral oils, the ratio $\Delta p/(2B)$ is less than 1% for pressure differences up to 320 bar. Thus,

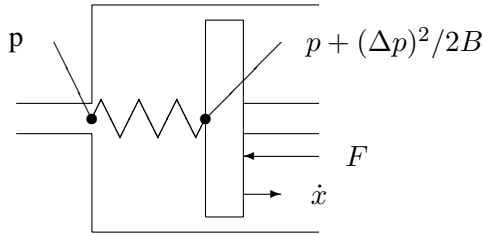


Fig. 10.24 Spring model of oil compressibility

$$F = A \left[p_0 + (\Delta p) \left(1 + \frac{\Delta p}{2B} \right) \right] \quad (10.72a)$$

$$\approx A(p_0 + (\Delta p)) = Ap. \quad (10.72b)$$

That is, the pressure force relation for the incompressible case is also a good approximation for the case of a compressible fluid.

If there are unsolved air bubbles in the oil, then it is well known that for *low* pressures, even a small percentage of unsolved air results in a significant decrease of the bulk modulus. The quadratic term in Equation 10.69 can still mostly be neglected. Let E^* be the elastic modulus of a mixture of oil and air, p_0 the atmospheric air pressure, V_{air_0} the air volume at pressure p_0 and V_{oil} the oil volume at pressure p . Then, under isothermal condition,

$$\frac{E^*}{B} = \frac{1 + \frac{p_0 V_{air_0}}{p V_{oil}}}{1 + \frac{p_0 V_{air_0}}{p V_{oil}} \frac{B}{p}}. \quad (10.73)$$

Assume that there is 1% of unsolved air in the oil, then at a low pressure of 10 bar, $E^*/B \approx 0.39$. For a pressure difference of 10 bar, $\Delta p/(2E^*) \approx 0, 2\%$.

However, if the quadratic term is neglected in Equation 10.69, then Equations 10.64–10.69 no longer fulfill Maxwell's reciprocity condition and cannot be represented by a 2-port C energy store in a bond graph.

Viscous Friction and Leakage in the Clearance between Piston and Cylinder Wall

There are two effects in the small clearance between piston and container wall, namely viscous friction and leakage between chambers (cf. Figure 10.25). Only few authors using bond graphs [1, 4] have considered their interaction. Mostly, they are simply not taken into account in bond graphs of hydraulic cylinders.

As it is well known, viscous friction due to piston movement is described by Newton's law. The shear force is

$$F_N = \eta \frac{\pi D l_p}{c} v_p, \quad (10.74)$$

where η is the dynamic viscosity, D the piston's diameter, l_p the length of the piston, Δr the clearance space, and v_p the piston's velocity. The piston is assumed to be centered in the cylinder. Equation 10.74 establishes a relation between the shear force, F_N , and the piston's velocity, v_p , and thus gives rise to an R element in the bond graph attached to the 1-junction representing the piston's velocity. The volume flow, Q_N , entrained by the piston's movement is

$$\begin{aligned} Q_N &= \int_0^A \dot{x} da = \int_0^{\Delta r} \dot{x}(y) dy \\ &= \pi D \frac{\Delta r}{2} v_p, \end{aligned} \quad (10.75)$$

where A is the annular area between piston and the container wall, and $\dot{x}(y)$ the velocity of a fluid particle at distance y above the piston in radial direction.

The second effect is a laminar fluid flow through the clearance due to the difference of chamber pressures across the piston given by the law of Hagen and Poiseuille.

$$Q_{HP} = \frac{\pi D (\Delta r)^3}{12 \eta l_p} (p_A - p_B) \quad (10.76)$$

This algebraic relation between the difference of the chamber pressures and the volume flow Q_{HP} is taken into account by an R element in a bond graph of a hydraulic cylinder. The law of Hagen and Poiseuille is derived from the balance of friction force and pressure force acting on a cylindrical fluid element of diameter $2y$ and length l .

$$\tau \times \pi \times 2y \times l = (p_A - p_B) \times \pi y^2, \quad (10.77)$$

where τ is the shear tension. Thus, the laminar flow through the clearance exerts on the piston the force

$$F_{HP} = \pi D \frac{\Delta r}{2} (p_A - p_B), \quad (10.78)$$

lowering the friction effect caused by the viscous flow due to the piston's movement. The coefficient $\pi D \Delta r / 2$ relating two efforts in Equation 10.78 is the same as in Equation 10.75, where it relates two flows. This suggests to represent both equations by a TF element of modulus $\pi D \Delta r / 2$.

Assuming a linear superposition of both effects, as indicated in Figure 10.25 by a plus sign between the two velocity profiles, the volume flow Q of the total steady fluid flow through the clearance relative to the container wall is

$$\begin{aligned} Q &= Q_N + Q_{HP} \\ &= \pi D \frac{\Delta r}{2} v_p + \frac{\pi D (\Delta r)^3}{12 \eta l_p} (p_A - p_B). \end{aligned} \quad (10.79)$$

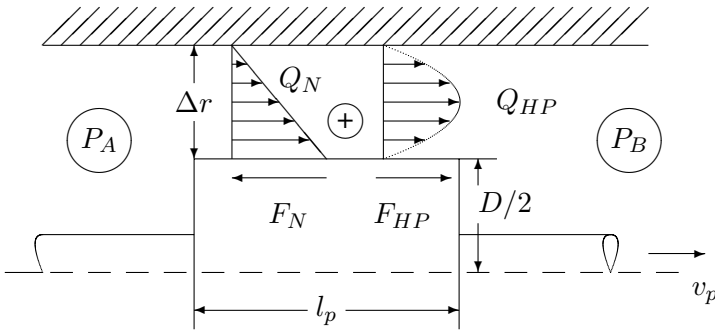


Fig. 10.25 Superposition of viscous flow and laminar flow across the piston

This sum of flows is represented in a bond graph of the hydraulic cylinder by a 0-junction. The balance of forces acting on the piston is

$$m_p \dot{v}_p = F_A - F_B - (-F_{HP} + F_N) - F_{load}, \quad (10.80)$$

where F_A , F_B are the forces acting on the left respectively the right piston land (cf. [17], vol. 1, p. 109).

Bond Graph Model of a Double Acting Hydraulic Cylinder

All considered effects, viz. viscous friction and leakage in the clearance between piston and container wall and oil compliance in the cylinder chambers are eventually represented by the bond graph of Figure 10.26.

The parameters of the linear resistors are $R_v := \eta\pi D l_K / \Delta r$ and $R_{HP} := 12\eta l_K / \pi D (\Delta r)^3$. The causal paths between the mechanical ports of the C elements and the I energy store representing the piston's inertia indicate oscillations in the cylinder chambers. However, as shown above, the nonlinear part of the forces acting on the piston (cf. Equation 10.69) is very small compared to its part due to the hydrostatic pressure. Hence, these oscillations are irrelevant.

10.2.5 Flow Forces in Hydraulic Spool Valves

Concluding this chapter on bond graph modelling of open thermodynamic systems, it is shown how flow induced forces in hydraulic spool valves can be correctly accounted for in a bond graph of the valve.

As it is well known, a spool control orifice is not only a restriction where hydrostatic energy is partly converted into kinetic energy and into some amount of heat. Simultaneously, the fluid flow in the spool valve control orifices forms a jet and exerts a force on the spool. Since servovalves operate as hydraulic amplifiers, it is

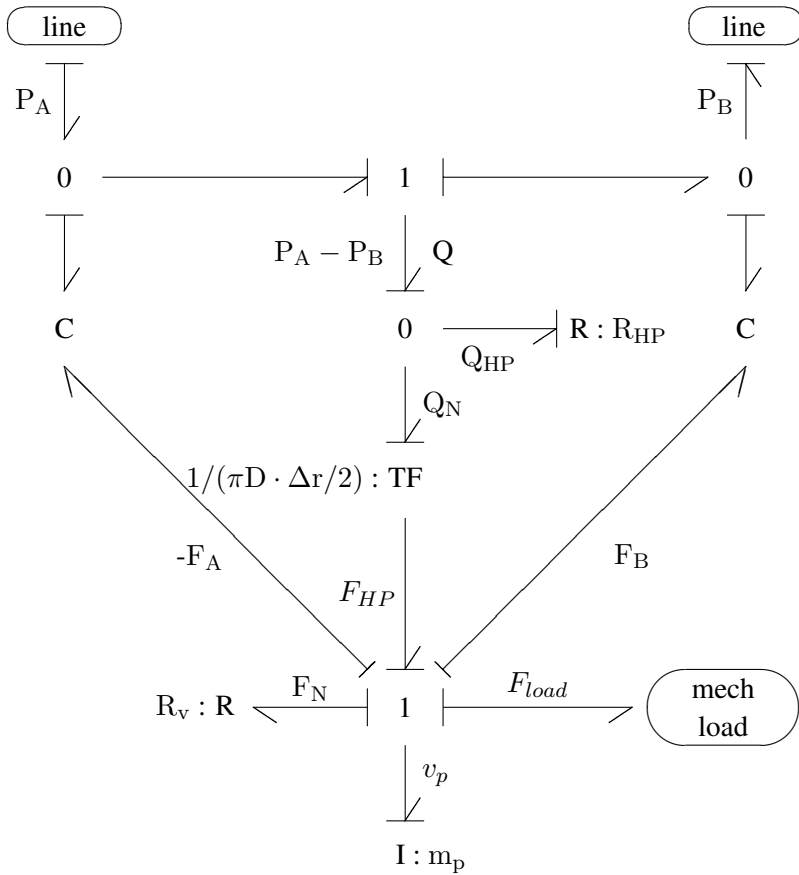


Fig. 10.26 Bond graph model of a hydraulic double acting cylinder (Borutzky, 1993)

important to account for these flow reaction forces in the balance of all forces acting on the spool in order to avoid a cause for instability and to ensure correct operation of the valve. Moreover, servovalves not only control the direction of flows, but also their amount and must meet given accuracy requirements.

Assuming an incompressible fluid flow leaving a valve chamber through a control orifice, conservation of momentum yields for the axial steady part of the flow induced force acting on the spool the well known formula

$$F_{flow} = -\rho \frac{Q^2}{c_c A(x_{sp})} \cos \varepsilon \tag{10.81}$$

[27], where c_c denotes the jet contraction coefficient, $A(x_{sp})$ the cross section area of the orifice modulated by the spool displacement, x_{sp} , and ε the angle between jet axis and spool axis. Assuming that for $x_{sp} = 0$ the orifice under consideration

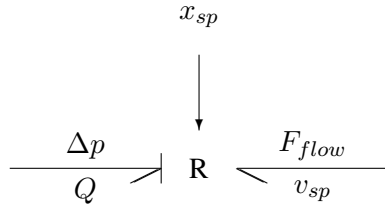


Fig. 10.27 Bond graph representation of a variable control orifice (Rabie and Lebrun, 1981)

is closed, then, for small spool displacements x_{sp} , a typical value of the angle is $\varepsilon = 69^\circ$.

On the other hand, for a turbulent fluid flow through an orifice of a fixed cross section area, the well known square root law

$$Q = c_d A \sqrt{\frac{2}{\rho}} \sqrt{|\Delta p|} \text{sign}(\Delta p) \quad (10.82)$$

holds that can be derived from Bernoulli's energy equation for an incompressible steady flow. The empirical discharge coefficient, c_d , accounts for energy losses. It depends on the geometry of the restriction and of the Reynolds number, Re , which characterises the mode of the flow (Often a constant value for turbulent conditions is adopted. A typical value is $c_d = 0.611$). In Equation 10.82, A is the cross section area of the orifice. The sign function accounts for the direction of the flow.

It is obvious to represent Equation 10.82 in bond graphs by a 1-port resistor that is modulated by the displacement of the valve spool in the case of a valve control orifice. In order to account for the axial steady flow force at variable control orifices, some authors follow an intuitively obvious method of adding a mechanical port to the 1-port hydraulic resistor as reproduced in Figure 10.27 [29]. It is true that the square root law results from Bernoulli's energy equation for a one-dimensional incompressible fluid. Its derivation, however, assumes a steady flow. Hence, the storage of kinetic energy due to the motion of the spool has to be neglected. It can be shown that an ad hoc representation of both equations by a 2-port R element inserted between the 1-junction representing the volume flow through the control orifice and the 1-junction representing the velocity of the spool (Figure 10.28) leads to a model that is inconsistent with the principle of power continuity.

Likewise, for the case of a flapper nozzle subsystem as part of the preamplifier of a two stage electrohydraulic servovalve, Breedveld [7] has shown that the intuitively obvious use of a 2-port R element accounting for the force induced by the fluid flow between nozzle and flapper violates power continuity.

Substituting Equation 10.82 into Equation 10.81 yields

$$F_{flow} = -K_f |\Delta p| x_{sp} , \quad (10.83)$$

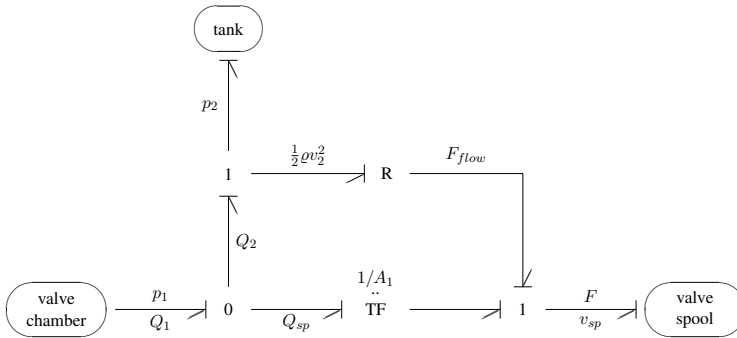


Fig. 10.28 Incorrect bond graph of a spool valve control orifice

where K_f is approximately a constant. Now, in this form, the equation for the axial steady flow force formally looks like the constitutive equation of a negative hydraulic spring of stiffness $K_f|\Delta p|$, which well reflects engineering experience that steady flow forces, independently of the flow direction, always tend to close valve ports. However, since the spring's stiffness depends on the absolute value of the pressure drop, representation by a C element would mean that it is modulated and thus violates the principle of energy conservation.

The two possible views show that a more fundamental approach is needed to come to a correct bond graph model of the interaction between fluid flow and the motion of the spool. The problem with an ad hoc model of spool valve control orifices is that fluid dynamics and mechanical effects are tightly coupled, while bond graph modelling requires a clear separation of energy storage from dissipation of free energy and from power conservative elements.

It might appear tempting to introduce further elements beyond the set of basic bond graph elements in order to represent given equations as relations between port variables of a new element. However, it contradicts the concept of bond graph modelling. Equations 10.82 and 10.83 might give rise to the introduction of an RC element. This, however, would mean giving up a clear separation of dissipation of free energy from energy storage. In view of these difficulties with an ad hoc bond graph representation of equations frequently used in hydraulics, an approach is presented in the following that uses an appropriate control volume, some simplifying though reasonable modelling assumptions and starts from conservation laws for mass, momentum and energy [3].

For the subsequent analysis, a four-way three-position spool valve with four control orifices is chosen as depicted in Figure 10.29 (S denotes the supply port, T the exhaust port, while A and B are load ports). Only for the downstream passage from load port A to the exhaust port T, a bond graph submodel is developed. The other control orifices can be modelled in a similar manner.

The control volume under consideration encompasses the oil filled cylindric region between the spool lands and extends into the vena contracta of the jet up to the

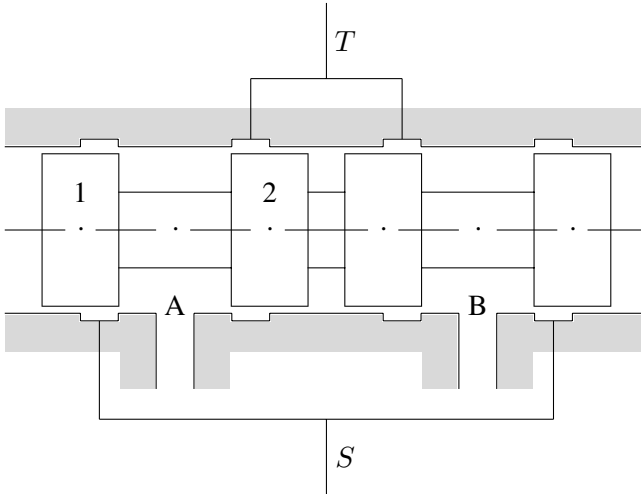


Fig. 10.29 Four-way three-position spool valve

area A_2 . The vector perpendicular on that area is parallel to the jet stream axis (Figure 10.30) In their mathematical modelling of a three-way underlapped hydraulic spool servovalve, Taft and Twill [34] use a similar control volume.

The control volume we are going to consider moves along with the spool and has a fixed size. Since oil enters and leaves the control volume, it is an open system with convection of matter, momentum and energy. Alternatively, the space between the two spool lands could be divided into two non-moving control volumes that vary with the motion of the spool.

In order to simplify the modelling process, we will assume:

- A one-dimensional incompressible fluid
- All energy losses can be neglected
- Isothermal conditions
- The internal energy U of the fluid can be neglected, that is, only kinetic energy is stored in the control volume.
- The fluid flow entering at load port A immediately establishes an average velocity inside the control volume with stream lines parallel, uniform and directed along the axis of the spool. The regions where the oil enters and leaves the control volume are considered small and will be neglected [34].
- The small amount of leakage from the supply port S into the control volume across the spool can be neglected.
- The angle ε of the jet stream at the exhaust port is considered to be constant ($\varepsilon = 69^\circ$).
- The tank is an infinitely large reservoir.

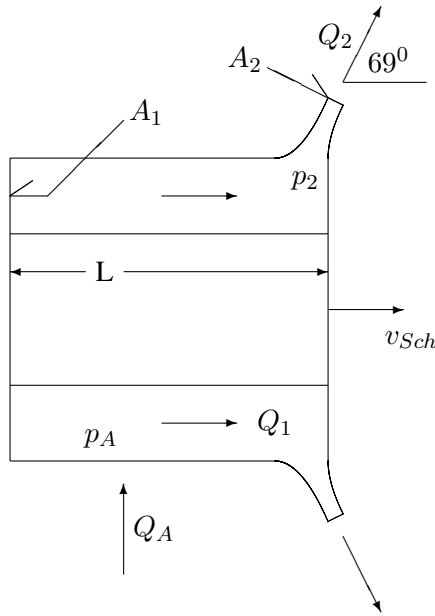


Fig. 10.30 Control volume

Since the flow is assumed to be incompressible and since the control volume is of constant size, the mass balance reduces to a simple continuity equation for the volume flows Q_A and Q_2 .

$$0 = Q_A - Q_2, \tag{10.84}$$

where Q_A is the volume flow entering the control volume at control port A and Q_2 is the flow exiting the control volume at the exhaust port T. Inside the control volume, the entering volume flow Q_A is superimposed by a flow $Q_{sp} := A_1 \times v_{sp}$ due to the motion of the spool, resulting in a volume flow Q_1 .

$$Q_1 = Q_A + Q_{sp} \tag{10.85}$$

Since inside the control volume we assume an average velocity with parallel stream lines parallel uniform and directed along the spool axis, the equality $Q_1 = A_1 \times v_1$ holds, where v_1 is the absolute average velocity inside the control volume with respect to an inertial frame.

The flow of oil through the control volume exerts a force on it. The momentum balance yields

$$\mathbf{F} = \oint_{ACV} \rho \mathbf{v} d\tilde{\mathbf{A}} + \frac{d}{dt} \int_V \rho \mathbf{v} d\tilde{V} + m_{CV} \dot{\mathbf{v}}_{sp}, \tag{10.86}$$

where A_{CV} denotes the surface of the control volume, V its volume and m_{CV} the fluid mass inside the control volume. The first term in the above sum accounts for the change of momentum due to the influx and the efflux of momentum through the control surface. The second term is the rate of change of momentum inside the control volume moving at velocity v_{sp} . The third term accounts for the acceleration of the control volume.

If the influx of momentum is neglected compared to the efflux, then by using Equation 10.85, the axial component of the force acting on the control volume, i. e. the flow force F_{flow} reads

$$\begin{aligned} F_{flow} &= \rho v_2 Q_2 \cos \varepsilon + \frac{d}{dt}(\rho v_A V) + m_{CV} \dot{v}_{sp} \\ &= \rho v_2 Q_2 \cos \varepsilon + (\rho L) \dot{Q}_1 \\ &= \rho v_2 Q_2 \cos \varepsilon + \dot{p}, \end{aligned} \quad (10.87)$$

where $\dot{p} := (\rho L) \dot{Q}_1$.

The rate of change of stored kinetic energy inside the control volume is

$$\begin{aligned} \dot{E}_{kin} &= \frac{d}{dt} \left(\frac{1}{2} m_{CV} v_1^2 \right) \\ &= \dot{p} v_1 \\ &= \left(\frac{\dot{p}}{A_1} \right) Q_1. \end{aligned} \quad (10.88)$$

Finally, by neglecting the dynamic pressure $\rho v_A^2/2$ at load port A in comparison to the dynamic pressure $\rho v_2^2/2$ at the exhaust port T, the energy balance takes the form

$$p_A Q_A - (p_2 + \frac{1}{2} \rho v_2^2) Q_2 + F_{flow} v_{sp} = \left(\frac{\dot{p}}{A_1} \right) Q_1. \quad (10.89)$$

Equations 10.84, 10.87 and 10.89 can be consistently represented in the bond graph of Figure 10.31, where m_{sp} is the mass of the spool and F is the driving force.

In fact, the bond linking both upper 1-junctions display the assumption of an incompressible fluid, i. e., the volume flow Q_A entering the control volume at load port A equals the exiting flow Q_2 in the vena contracta. Thus, the continuity equation for volume flows, Equation 10.84, is taken into account.

The flow of oil leaves the control volume at total pressure $p_2 + \rho v_2^2/2$. If this sub-model is used in a bond graph of a hydraulic system in which efforts are hydrostatic pressures, then the dynamic pressure term $\rho v_2^2/2$ has to be taken into account by an element. For that reason, an S_R element modulated by the position of the spool has been attached to the upper 1-junction. It represents convection of kinetic energy out of the control volume. The fluid flow leaving the control volume at relatively high velocity v_2 gains kinetic energy by a reduction of the hydrostatic pressure. This is well known from Bernoulli's equation applied to an orifice.

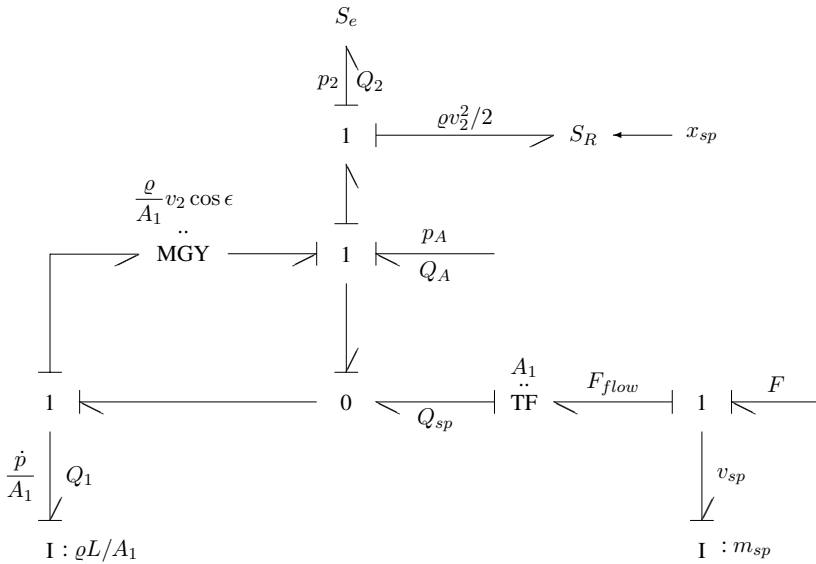


Fig. 10.31 Bond graph of the hydraulic mechanical interaction in a spool valve control orifice (Borutzky, 1992)

Some authors [24] describe this phenomenon by means of a special resistor they call the *Bernoulli-resistor*. However, since it is the conversion of hydrostatic energy into kinetic energy and not the irreversible conversion of energy into heat, we follow Breedveld and use a controlled sink (Breedveld calls this element a Bernoulli sink [7]). However, usually all energy losses are taken into account by an empirical discharge coefficient in the square root law for a steady-state flow through orifices. In this case, the S_R element is to be replaced by a true resistor.

The left 1-junction represents the volume flow, Q_1 , inside the control volume and the momentum balance, Equation 10.87. The modulated gyrator accounts for the efflux of momentum due to the volume flow, Q_2 , exiting the control volume. The I element attached to the 1-junction of the volume flow Q_1 represents the kinetic energy of the oil inside the control volume sitting on the moving spool.

In order to see that the bond graph correctly reflects the principle of energy conservation, consider the power balance at the 1-junction representing the volume flow Q_A .

$$0 = p_A Q_A - (p_2 + \frac{1}{2} \rho v_2^2) Q_2 - \frac{F_{flow}}{A_1} (Q_1 - Q_{sp}) + (\frac{\rho}{A_1} v_2 Q_2 \cos \epsilon) Q_1 \tag{10.90}$$

If the sum of all efforts at the 1-junction of Q_1 , i. e., the momentum balance is used, Equation 10.90 takes the form of the energy balance, Equation 10.89.

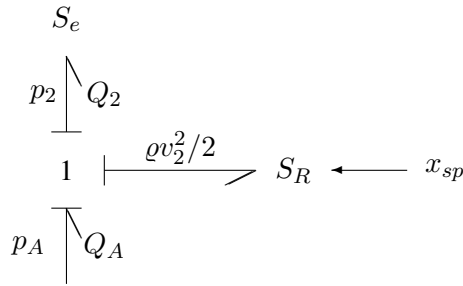


Fig. 10.32 Bond graph of the spool valve control orifice neglecting spool motion

If a steady fluid flow through the control volume is assumed and if the motion of the spool is neglected, that is $\dot{p} = 0$ and $v_{sp} \approx 0$, then the bond graph model reduces to the part depicted in Figure 10.32. The model equations take the simple form

$$Q_A = Q_2 \quad (10.91a)$$

$$F_{flow} = \rho v_2 Q_2 \cos \varepsilon \quad (10.91b)$$

$$0 = p_A Q_A - \left(p_2 + \frac{1}{2} \rho v_2^2 \right) Q_2 . \quad (10.91c)$$

Energy losses are usually taken into account by means of an empirical discharge coefficient in the energy equation, Equation 10.91c. Consequently, the Bernoulli sink modulated by the spool displacement has to be replaced by an R element. Thus, we come back to the representation of a control orifice by means of a controlled resistor considered at the beginning of this section. However, the flow force cannot be represented since the power $F_{flow} \times v_{sp}$ disappears in the quasi static case ($\dot{x}_{sp} \approx 0$). That is, to reach a correct bond graph representation of the fluid mechanical interaction in a spool valve control orifice, the motion of the spool must *not* be neglected.

10.3 Conclusion

In hydraulic, pneumatic and process engineering systems, the exchange of energy between subsystems or components is generally bound to mass flows. That is, not only hydrostatic and kinetic energy are exchanged, but also mass, internal energy and momentum. Due to the fact that the total power transmitted between two power ports is the *sum* of contributions from several energy domains, the most appropriate choice and the use of two variables, of which the product equals the instantaneous power, is not obvious. In the special case of hydrostatic systems (usually isothermal conditions are assumed), the hydrostatic pressure is chosen as an effort variable and the volume flow as a flow. Their product *approximates* the instantaneous power.

Bond graph modelling of hydrostatic systems corresponds to the bond graph modelling of electrical circuits. Of course, elements in each discipline have their domain specific characteristics.

In this chapter, the question has been addressed as to how to represent thermodynamic systems by bond graphs. Since there is no unique answer to this question in the literature, two approaches have been presented and discussed, thereby focussing on some fundamental problems. In both approaches, the energy transport between two locations bound to the fluid flow is not represented by a single bond. One of these two methods, the *pseudo* bond graph approach, drops the fundamental requirement that the product of the two variables associated with each bond must equal the instantaneous amount of transmitted power in favour of more flexibility in the modelling. Consequently, the C storage element is no longer energy conservative. It reflects the *accumulation* of mass and energy.

One approach to bond graph modelling of process engineering systems [15, 20] is to develop separate pseudo bond graphs for the transport of mass and for the transport of thermal energy. Elements in the pseudo bond graph representing the heat transport are modulated by variables from the pseudo bond graph displaying the mass transport in the system. The advantage of this approach is that each of the two phenomena can be conveniently described by means of variables commonly used in thermodynamics, e.g. pressure, mass flow on the one hand and enthalpy and heat flow on the other hand. Couplings between both pseudo bond graphs are due to the fact that parameters of elements in the thermal pseudo bond graph depend on the variables in the pseudo bond graph of mass flows.

In another pseudo bond graph approach introduced by Karnopp, couplings between thermal and hydraulic or pneumatic quantities take place in multiport elements. For representing the accumulation of mass, thermal energy and volume in a gas filled control volume of variable size, Karnopp introduced a 3-port C element with a material (p, \dot{m}) -port, a thermal (T, \dot{U}) and a hydraulic $(-p, \dot{V})$ -port. Since this C element is *not* energy conservative, it is called a *thermodynamic accumulator*. With this accumulator, the stored energy is not a function of the state variables, but a state variable itself. At the hydraulic port, the product of port variables equals power so that this port could be connected via a transformer to a true bond graph of a mechanical subsystem.

The other fundamental phenomenon is the flow of a compressible fluid through a nozzle. Karnopp proposes to represent this effect by means of an R element with two material and two thermal ports with all four of them having conductance causality.

These two elements enable one to develop pseudo bond graphs of pneumatic systems in a clear and systematic manner. Thus, thermal aspects can be conveniently modelled by a pseudo bond graph and can also be linked to a bond graph of a mechanical subsystem via the hydraulic port of the thermodynamic accumulator. From a pragmatic point of view, the advantages make up for the conceptually weak point that the product of variables associated with a bond is not equal to the power across the bond.

On the other hand, as has been demonstrated in particular by Breedveld, thermodynamic systems can also be modelled by *true* bond graphs. Since the starting

point are fundamental physical principles, e.g. Gibbs' equations, it is theoretically more convincing. However, some of the variables used in this approach cannot be directly measured or are less accessible than usual quantities like temperature and pressure. Breedveld also represents a gas filled control volume of variable size by means of 3-port C element. In contrast to Karnopp's thermodynamic accumulator, the product of variables at each port equals power and the element is an energy conservative C energy store. According to Gibb's equation, the supplied power is the sum of three parts, namely thermal power due to convection or conduction, hydraulic power linked with the change of volume and material power due to a mass flow in and out of the control volume. The pairs of power conjugate variables are temperature, T , and the rate of change of entropy, \dot{S} , chemical potential, μ , and mole mass flow, \dot{N} as well as pressure, p , and volume flow, \dot{V} . Out of these six quantities, the entropy flow, \dot{S} , and the chemical potential, μ , for instance, cannot be directly measured.

A compressible fluid flow through an orifice of constant cross sectional area can be simply represented by a 1-port resistor if instead of pressures and temperatures, the enthalpy at the inlet and at the outlet are used. The latter quantities, however, cannot be directly measured.

Regarding hydraulic systems, it is customary to assume an incompressible oil when it flows through orifices and to account for its compressibility in dead volumes or chambers. With regard to the modelling of cylinder chambers, occasionally, this has led to an ad hoc representation by means of a C element modulated by the displacement of the piston. Since such an element violates the energy conservation principle, an energy conservative 2-port C energy store has been proposed in this chapter that accounts for the change of volume due to the motion of the piston. In the constitutive equation for its mechanical port, a term appears that assumes small numerical values in practice. However, if this term is neglected, then the equations for the pressure at the hydraulic port and the force acting on the piston can no longer be represented by an energy conservative 2-port C energy store, but by the artefact of a controlled C element.

The bond graph model developed for a double acting hydraulic cylinder also describes the viscous flow in the clearance between piston and container due to the motion of the piston and accounts for leakage between the cylinder chambers across the piston. Often, this leakage is modelled by a resistor. The superposition of both effects has been considered rarely in the literature.

The chapter concludes with the development of a small true bond graph model of the interaction between the flow of oil through a spool valve control orifice assumed to be incompressible and the motion of the spool. Under some simplifying but reasonable assumptions, the bond graph derived from conservation laws for mass, momentum and energy accounts for the kinetic energy of the fluid flow and the motion of the spool. In contrast, in some bond graph models of hydraulic systems reported in the literature, the 1-port resistor representing energy losses in an orifice is extended in an intuitive and ad hoc manner by a second mechanical port in order to account for the flow force on the spool. However, this representation of formulae,

commonly used in practical analysis of hydraulic control systems, is not consistent with power continuity.

References

- [1] B.W. Barnard. Predicting the dynamic response of a hydraulic system using power bond graphs. Master's thesis, Monash University, Melbourne, Australia, 1973.
- [2] J.J. Beaman and P.C. Breedveld. Physical Modeling with Eulerian Frames and Bond Graphs. *Journal of Dynamic Systems, Measurement and Control*, 110(2):182–188, 1988.
- [3] W. Borutzky. A Dynamic Bond Graph Model of the Fluid Mechanical Interaction in Spool Valve Control Orifices. In P. C. Breedveld and G. Dauphin-Tanguy, editors, *Bond Graphs for Engineers*, pages 229–236. Elsevier, North-Holland, 1992.
- [4] W. Borutzky. An Energetically Consistent Bond Graph Model of a Double Acting Hydraulic Cylinder. In A. Pavé, editor, *Modelling and Simulation 1993, Proc. of the 1993 European Simulation Multiconference*, pages 203–207. SCS Publishing, June 7-9 1993. Lyon, France.
- [5] B. Ould Bouamama, K. Medjaher, A.K. Samantaray, and M. Staroswiecki. Supervision of an industrial steam generator. Part I: Bond graph modelling. *Control Engineering Practice*, 14 (1):71–83, 2006.
- [6] P.C. Breedveld. Integrated Modeling of Physical Systems. to appear.
- [7] P.C. Breedveld. The Thermodynamic Bond Graph Concept Applied to a Flapper Nozzle Valve. In *Proc. 10th IMACS World Congress on Systems Simulation and Scientific Computation*, volume 3, pages 395–397, 1982.
- [8] P.C. Breedveld. *Physical Systems Theory in Terms of Bond Graphs*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1984.
- [9] F.T. Brown. *Engineering System Dynamics*. Marcel Dekker, New York, Basel, 2001. ISBN: 0-8247-0616-1.
- [10] F.T. Brown. Convection Bonds and Bond Graphs. In P.C. Breedveld, editor, *Current Topics in Bond Graph Related Research*, pages 871–886. Journal of The Franklin Institute, 1991.
- [11] F.E. Cellier. *Continuous System Modeling*. Springer-Verlag, New York, Berlin, Heidelberg, 1991. ISBN: 0-387-97502-0.
- [12] F. Couenne, C. Jallut, B. Maschke, P.C. Breedveld, and M. Tayakout. Bond graph modelling for chemical reactors. *Mathematical and Computer Modelling of Dynamic Systems*, 12(2-3): 159–174, 2006.
- [13] P. Dransfield. *Hydraulic Control Systems – Design and Analysis of Their Dynamics*. Springer-Verlag, New York, 1981.
- [14] P. Dransfield and M.K. Teo. Using bond graphs in simulating an electro-hydraulic system. *Journal Franklin Instute*, 308(3):173–184, 1974.
- [15] P.J. Gawthrop and L. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [16] C. Heny, D. Simanca, and M. Delgado. Pseudo-bond graph model and simulation of a continuous stirred tank reactor. *Journal of The Franklin Institute*, 337:21–42, 2000.
- [17] J.L. Shearer J.F. Blackburn, G. Reethof. *Fluid Power Control*, volume 1. Krausskopf-Verlag, Wiesbaden, 1962.
- [18] J.L. Baliño. Galerkin finite element method for incompressible thermofluid flows framed within the bond graph theory. *Simulation Modelling Practice and Theory*, 17 (1):35–49, 2009.
- [19] J.L. Baliño and A.E. Larreteguy and E.F.G. Gandolfo Raso. A general bond graph approach for computational fluid dynamics. *Simulation Modelling Practice and Theory*, 14:884–908, 2006.
- [20] D.C. Karnopp. Pseudo bond graphs for thermal energy transport. *Journal of Dynamic Systems, Measurement, and Control*, 100:165–169, 1978.

- [21] D.C. Karnopp. A bond graph modeling philosophy for thermofluid systems. *Journal of Dynamic Systems, Measurement, and Control*, 100:71–75, 1978.
- [22] D.C. Karnopp. State Variables and Pseudo Bond Graphs for Compressible Thermofluid Systems. *Journal of Dynamic Systems, Measurement, and Control*, 101:201–204, 1979.
- [23] D.C. Karnopp and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1975.
- [24] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: A Unified Approach*. John Wiley & Sons, Inc., New York, 1990.
- [25] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics - Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons Inc., Third edition, 2000. ISBN 0-471-33301-8.
- [26] K. Medjaher, A.K. Samantaray, and B. Ould Bouamama. Bond graph model of a vertical U-tube steam condenser coupled with a heat exchanger. *Simulation Modelling Practice and Theory*, 17 (1):228–239, 2009.
- [27] H.E. Merritt. *Hydraulic Control Systems*. Wiley & Sons, 1967.
- [28] A. Mukherjee and R. Karmakar. *Modelling and Simulation of Engineering Systems through Bondgraphs*. Narosa Publishing House, New Delhi, India, 2000. ISBN: 81-7319-279-0.
- [29] G. Rabie and M. Lebrun. Modélisation par les graphes à liens et simulation d'une servovalve a deux étages. *R.A.I.O. Automatique/Systems Analysis and Control*, 15(2):97–129, 1981.
- [30] A.K. Samantaray, K. Medjaher, B. Ould Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy. Component based modelling of thermo-fluid systems for sensor placement and fault detection. *SIMULATION: Transaction of The Society for Modeling and Simulation International*, 80(8):381–398, 2004.
- [31] A.K. Samantaray, S.K. Goshal, K. Medjaher, and B. Ould Bouamama. Reconfiguration of an industrial steam generator using bond graph modelling. *International Journal of Modelling, Identification and Control*, 2 (2):154–168, 2007.
- [32] S. Scavarda. Systèmes hydrauliques et pneumatiques. In G. Dauphin-Tanguy, editor, *Les bond graphs*, chapter 3, pages 111–159. Hermes Science Publications, 2000.
- [33] A. Schöne. Über abstrakte Modelle technischer Systeme und deren Strukturen. *VDI-Z*, 119 (15/16):753–758, 1977.
- [34] C.K. Taft and J.P. Twill. An analysis of the three-way underlapped hydraulic spool servovalve. *Journal of Dynamic Systems, Measurement and Control*, 100:117–123, 1978.
- [35] J.U. Thoma. Thermofluid Systems by Multi-bondgraphs. *Journal of the Franklin Institute*, 329(6):999–1009, 1992.
- [36] J.U. Thoma and H. Atlan. Network Thermodynamics with Entropy Stripping. *JFI*, 303: 319–328, 1977.
- [37] J.U. Thoma and B. Ould Bouamama. *Modeling and Simulation in Thermal and Chemical Engineering (A Bond Graph Approach)*. Springer-Verlag, Berlin, 2000.
- [38] J.U. Thoma and G. Mocellin. *Simulation with Entropy in Engineering Thermodynamics*. Springer, Berlin, Heidelberg, New York, 2006. ISBN -10 3-540-32798-3.
- [39] B. Willson and A.E. Traver. The Use of Control Volume Analysis and Non-potential Junction Concepts to Model Liquid Piston Engine Dynamics. In *Proc. of the American Control Conference*, pages 1436–1443, 1987.

Chapter 11

Automated Modelling

The previous chapters presented the fundamentals of bond graph methodology and its potential in tackling some basic problems in various application areas, e.g. models of variable structure (Chapter 7), lumped parameter approximation of distributed parameter models (Chapter 9), and open thermodynamic systems (Chapter 10). The questions this chapter attempts to answer are: how can software support bond graph-based physical systems modelling and in which phases of the modelling process can it do so. Before going into details, an important general observation has to be pointed out.

As we know, once a formalised model description is available, it can be transformed fully automatically into an executable simulation program for the problem to be analysed. During model development, formal checks, e.g. with regard to consistency of physical dimensions, can help avoid flaws that otherwise might be overlooked and that could result in strange simulation results. Moreover, some knowledge-based software can provide support to some extent when decisions have to be made during modelling. However, it is the author's belief that the possible potential in automating the modelling process will not result in a software-driven, fully automatic creation of a model that meets the given requirements. Engineering experience, especially in the phases of specification and conceptual system design, assessment of both modelling assumptions and simulation results, and creativity will still remain the essential key to the successful development of a model that is adequate for the problem under consideration. In this sense, the heading of this chapter is *automated* modelling.

In his Ph.D. thesis ([81], p. 50), Top poses the question “*Modelling: art or algorithm?*” and gives the following answer.

We believe to have clarified the source of this dichotomy by identifying the three basic ingredients of the modelling task: specification of requirements, construction of system structure and assessment of the system in terms of explicit requirements. Construction and assessment can largely be automated using the set of explicit requirements, whereas specification entails asking the right questions and is therefore under responsibility of the modeller.

An automated modelling system for the development of parsimonious models that handles modelling assumptions, uses model satisfaction criteria and can revise a

model on the basis of internal considerations is the software package AIM (Artificially Intelligent Modeller) presented by N. Smith in his Ph.D. thesis [72].

11.1 Continuous System Simulation Languages

With the advent of the first (digital) computers, the procedural programming language FORTRAN emerged and was frequently used in subsequent decades to formulate models for the simulation of engineering systems on computers. By following causal paths in small bond graphs, equations can be manually and directly derived from the graph in a systematic manner as demonstrated in Section 3.5 by three small examples. These equations can be transformed with little effort into statements of a FORTRAN subroutine to be compiled and linked with a simulation program. For larger bond graphs, a hierarchical modular model structure can be mapped onto nested calls of subroutines.

Sequential processing of a program on one processor of a computer requires that statements in a computer program are sorted into a computational order that is not determined by the engineering problem or the modelling process. Observing the sequence of statements distracts from modelling considerations and is a potential source of errors. This problem did not appear when a simultaneous block diagram was mapped onto an analog computer.

The next step towards better support of model formulation was the development of *declarative* simulation languages that enable a *simultaneous* problem-oriented description of continuous systems. At this level, a simulation program is needed that *automatically* sorts model equations and transforms them into a program formulated in a procedural programming language, e.g. FORTRAN or C. For these declarative, so-called *Continuous System Simulation Languages* (CSSLs), a committee set up a standard in 1967 [74] that has been supported by numerous simulation programs developed all over the world. These software programs not only support the CSSL standard, but also provide some language extensions. One prominent representative of this class of simulation programs is the program ACSL^{®1} [76] and its *Advanced Continuous System Simulation Language*. This program has been widely used all over the world in academia as well as in industry. Of course, since then many other advanced competitive modelling and simulation programs have emerged. Some of them will be considered in the next section.

Programming of model equations in the simulation language ACSL *directly* from a small causal bond graph is made easier because model equations can go into a so-called DERIVATIVE section in *arbitrary* order (By this method, the author entered bond graph models into ACSL[®] more than 25 years ago, see also Karnopp 1984, [47]). The syntax of ACSL statements is close to that of FORTRAN. The DERIVATIVE section is part of the DYNAMIC section that encompasses all dynamic model equations that must be solved in a time loop integration time step by time step. An

¹ ACSL, acslX, and PowerBlock are registered trademarks of The AEGIS Technologies Group, Inc., 631 Discovery Drive, Huntsville, AL 35806 USA, <http://www.acslx.com>

ACSL model description starts with the keyword `PROGRAM` and is closed by the keyword `END`. In the literature, often the term ACSL program is used.

Using Macros for an ACSL Description of Hierarchical Modular Structured Models

A hierarchical modular structured model description can be achieved by using pre-defined functions, by defining so-called `MACROS` for submodels and by invoking user-defined `FORTRAN` subroutines in ACSL[®]. Calls to `MACROS` can be nested in arbitrary depth. When a preprocessor, called translator, transforms the ACSL description of a model into `FORTRAN` code, calls to `MACROS` are replaced by their corresponding code. `MACROS` enable one to describe block diagrams such that the ACSL description clearly reflects their structure.

Example: ACSL Description of a Shunt Motor Model

In order to see how easy it is to formulate model equations in ACSL directly from a causal bond graph, consider the example of the shunt motor model depicted in Figure 11.1.

At time t_1 , the motor experiences an instantaneous increase of the load moment. As can be seen from the bond graph, the inductance of the field winding has been neglected. From the bond graph of Figure 11.1, we derive the following equations.

$$u_R = R_a \times i_a \quad (11.1a)$$

$$i_f = \frac{1}{R_f} E \quad (11.1b)$$

$$M_R = R_m \times \omega \quad (11.1c)$$

$$\Psi = K \times i_f \quad (11.1d)$$

$$u_a = \Psi \times \omega \quad (11.1e)$$

$$M = \Psi \times i_a \quad (11.1f)$$

$$\frac{di_a}{dt} = \frac{1}{L_a} (E - u_a - u_R) \quad (11.1g)$$

$$\frac{d\omega}{dt} = \frac{1}{J_m} (M - M_R + M_{load}) \quad (11.1h)$$

Figure 11.2 shows the corresponding ACSL description, which is almost self-explanatory. In the section starting with the keyword `INITIAL` parameter values, initial values and parameters controlling the numerical integration of state equations are provided. The `DERIVATIVE` block in the `DYNAMIC` section includes all model equations derived from the bond graph. In a time loop, the simulation program runs through the `DERIVATIVE` section for each time $t \leq T_{stop}$ until the end of the interval $[0, T_{stop}]$ is reached. The `TERMINAL` section may hold assignment statements that need to be performed only once after the simulation run in order to compute output variables or other expressions of interest.

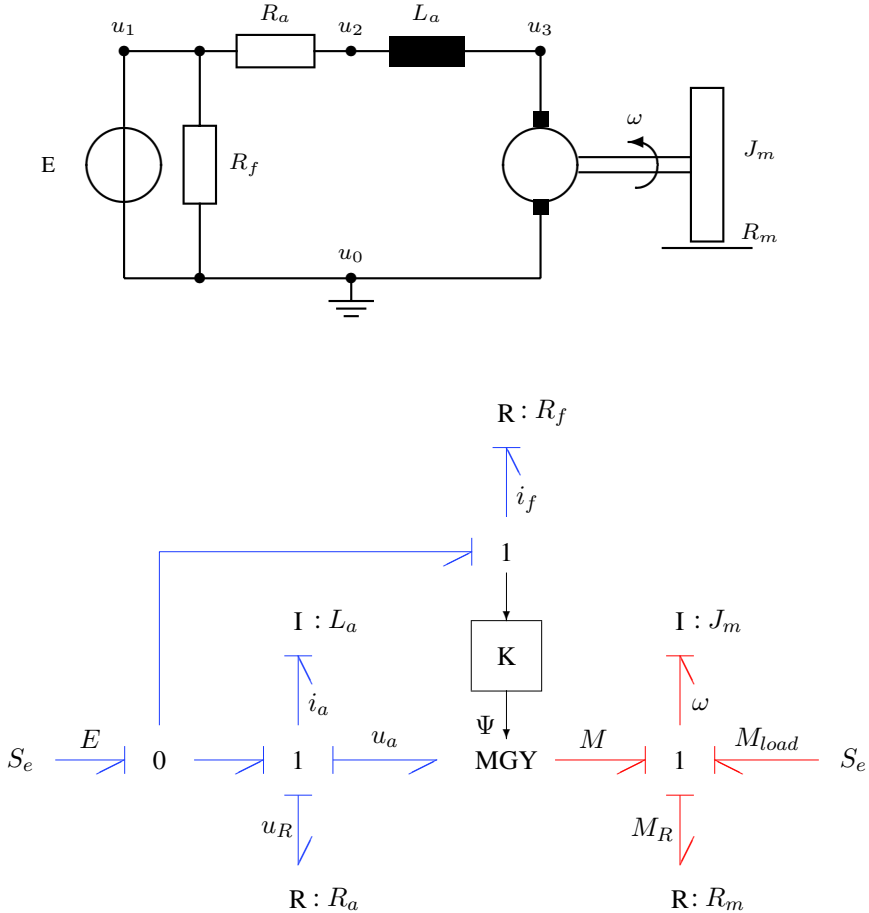


Fig. 11.1 Schematic and causal bond graph of a shunt motor

When the motor model is part of a large overall system model, it can obviously be used as a submodel in order to achieve a clear modular textual description that reflects the structure of a graphical model representation. In ACSL, this is possible by defining the motor model as a MACRO as shown in Figure 11.3.

The aim of introducing the shunt motor model as an ACSL MACRO is to enable its multiple reuse. On the other hand, when an ACSL program is processed by the translator, all MACRO calls are replaced by the model equations. Therefore, all variables that are not passed on with the argument list of the MACRO must be made local in order to ensure a global consistency of variable names. This is achieved with the MACRO REDEFINE directive. Moreover, this MACRO mechanism requires that all element parameters are passed on with the MACRO argument list. If initial

```

PROGRAM Shunt motor
  INITIAL
    CONSTANT E      = 220.0 ! V ! External voltage supply
    CONSTANT Ra     = 0.875 ! Ohm ! Resistance of armature winding
    CONSTANT La     = 0.175 ! Vs/A ! Inductance of armature winding
    CONSTANT Rf     = 5.495 ! Ohm ! Resistance of field winding
    CONSTANT K      = 0.0307 ! Vs/A ! Transduction coefficient
    CONSTANT Jm     = 0.8 ! Nm^2 ! Moment of inertia of the rotor
    CONSTANT Rm     = 0.066 ! Nms ! Rotary dashpot coefficient
    CONSTANT M01    = 100.0 ! Nm ! Load moment
    !
    CONSTANT tstop  = 5.0 ! s
    CONSTANT tz1    = 2.5 ! s
    !
    ! Initial values:
    CONSTANT ia0    = 0.0
    CONSTANT w0     = 0.0
    !
    ALGORITHM IALG = 4
    NSTEPS NSTP = 1
    MAXTERVAL MAXT = 1.0e-3
    MINTERVAL MINT = 1.0e-6
    CINTERVAL CINT = 0.1
  END ! of initial
  DYNAMIC
    DERIVATIVE
      Mload = M01 * STEP(tz1) ! Load moment
      MR = Rm*w ! Friction moment
      ! Resistor of the field winding:
      if = E/Rf
      ! Gyration equations:
      psi = K*if
      ua = psi * w ! w : Angular velocity omega
      M = psi * ia ! ia: Armature current
      ! Resistor of the armature winding:
      uR = Ra * ia
      ! State equations:
      diadt = (1/La) * ( E - ua - uR )
      dwdt = (1/Jm) * ( M + Mload - MR )
      !
      ia = integ (diadt, ia0)
      w = integ (dwdt, w0)
    END ! of derivative
    TERMT( T .GE. TSTOP )
  END ! of dynamic
  TERMINAL
  !
  END ! of terminal
END ! of program

```

Fig. 11.2 ACSL description of the shunt motor model

```

MACRO Motor (E,Mload,i,w,La,Ra,Rf,K,Jm,Rm,ia0,w0)
! input variables : E,Mload
! E : Voltage,
! Mload : Load moment
! output variables : i, w
! i : Current,
! w : Angular velocity
! Parameters : La,Ra,Rf,K,Jm,Rm
! La : Armature inductance
! Ra : Armature resistance
! Rf : Field resistance
! K : Transduction coefficient
! Jm : Total moment of inertia
! Rm : Rotary dashpot coefficient
! internal variables: ia, if,psi,ua,uR,diadt,dwdt,M
! ia : armature current
! if : field current
! initial values: ia0,w0 ! ia0 : initial value of ia
! w0 : initial value of w
MACRO REDEFINE ia,diadt,if,ua,uR,w,dwdt,M
MACRO STANDVAL ia0=0.0,w0=0.0
if = E/Rf
psi = K*if
ua = psi * w
M = psi * ia
uR = Ra * ia
diadt = (1/La) * ( E - ua - uR )
ia = integ(diadt, ia0)
MR = Rm * w
dwdt = (1/Jm) * ( M - MR + Mload )
w = integ(dwdt, w0)
i = if + ia
MACRO END

```

Fig. 11.3 ACSL MACRO for reuse of the shunt motor model

values of state variables are omitted in a MACRO call, then values provided with the MACRO STANDVAL directive are used.

It is true that continuous system simulation languages (CSSLS), e.g. ACSL, enable the description of large models in a hierarchical and modular way. In order to facilitate the ACSL description of schematics of mechanical systems in one-dimensional motion and of bond graphs, Zeid [90] has designed MACROS for basic mechanical components, e.g. masses, springs, dampers and levers as well as for basic bond graph elements.

The Requirement for Causality Assignment at Submodel Ports

The problem, however, is that MACROS require computational causalities to be defined in the model since only so-called *assignment statements* are allowed,

$$\langle Variable \rangle = \langle Expression \rangle, \quad (11.2)$$

in which the expression on the right-hand side is evaluated and assigned to the variable on the left of the equal sign. In the DYNAMIC section of an ACSL description of a model, there must be just one assignment statement for each time dependent variable defining it. This means that the equations in an ACSL program are meant to describe simultaneous relations. Variables, however, are statically divided into input and output variables. This requirement is met by block diagrams and by bond graphs after computational causalities have been eventually assigned. If, however, submodels are connected according to the physical structure of a system (and this is common, e.g. for networks of electrical or hydraulic systems as well as for acausal bond graphs), then variables of submodels can be classified a priori only into those that are local and those needed for interconnection with other submodels. In bond graphs, the latter are the power conjugated port variables. Which of the two variables of a power port assumes the role of an effort such that the conjugate variable becomes the flow depends on the properties of the partner models a submodel under consideration is connected to.

The core of a bond graph model of a hydraulic pump, for instance, is a transformer because the component transforms mechanical energy into hydraulic energy. Internal leakage from the high pressure region to the tank can be accounted for by a resistor. In this case, the hydraulic port of the pump model can have flow causality. The pump model supplies a fluid flow, Q , according to the angular velocity, ω , at which the pump is operated. If oil compliance in the high pressure region is accounted for by a C store, then the hydraulic port of the pump model has got preferred integral causality (Figure 11.4). In the bond graphs of Figure 11.4, V_p denotes the pump's volumetric displacement.

ACSL allows for conditional statements in MACRO definitions. If computational causality at a port is passed as information in the argument list of a MACRO, assignment statements for both possible cases of causality assignment can be provided in the body of the MACRO. From a practical point of view, however, this is convenient only for 1-port elements. The interconnection of submodels according to the physical structure of a system entails two other problems if a CSSL is used.

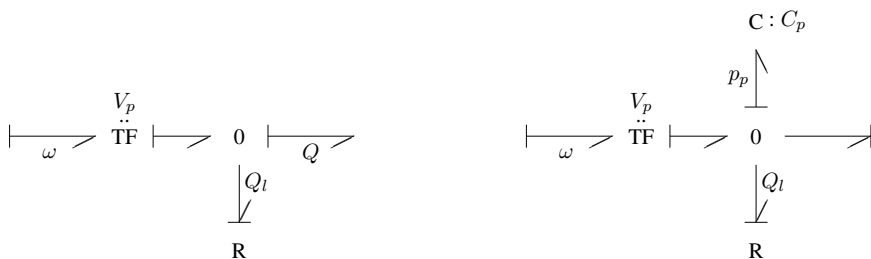


Fig. 11.4 Bond graph models of a constant flow pump with different causalities at its hydraulic port

Algebraic Dependencies Between State Variables

Due to the interconnection of submodels, state variables in different submodels can become algebraically dependent. This happens if, for instance, the pump model on the right side of Figure 11.4 is connected to a line model in which oil compliance has been lumped into two C energy store at the line's inlet and outlet (cf. Figure 9.2). If macros with a static classification of interface variables into inputs and outputs are introduced in order to come to a clear modular description of larger models, then the problem of algebraic dependencies between state variables due to the interconnection of submodels can be circumvented by providing several models of a component with different causalities at their ports. If the hydraulic pump model is to be connected to a line model, then an algebraic dependency between C energy stores in both models could be avoided by either not lumping oil compliance at both ends of the line or by using a joint C element that accounts for oil compliance in the pump and at the line's inlet. The equation of this joint C energy store is then used only in one of the two macros corresponding to the pump and to the line model. If, however, pre-defined macros are taken from a library to build a CSSL description of a model of a large system, then it is difficult to see all the implications that can arise from the choice of macros and their interconnection.

Algebraic Loops

The second problem is that the interconnection of submodels can entail algebraic loops. In the case of dependent state variables as well as in the case of algebraic loops, the equations of the overall system model cannot be transformed into explicit state space form if the algebraic constraints cannot be solved. This, however, was required by early CSSL processing software. To overcome the problems with algebraic loops, the language ACSL provides an `IMPL` operator that enables an iterative solution of implicit algebraic equation in one unknown. Furthermore, after an implementation of the BDF method was included in version 11 of the simulation program ACSL[®] [76], this programs not only accepts assignment statements and implicit algebraic equations in one unknown, but also the general case of sets of differential-algebraic equations. Let \mathbf{x} denote the state vector and \mathbf{z} the vector of algebraic unknowns, then a set of differential-algebraic equations in semi-explicit form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{z}) \quad (11.3a)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{z}) \quad (11.3b)$$

can be formulated in ACSL by means of the integration operator `INTVC` and the `IMPVC` operator denoting the solution of an implicit vector equation by iteration.

$$\mathbf{x} = \text{INTVC}(\mathbf{f}, \mathbf{x}0) \quad (11.4a)$$

$$\mathbf{z} = \text{IMPVC}(\mathbf{g}, \mathbf{z}0), \quad (11.4b)$$

where \mathbf{f} denotes the vector of derivatives and \mathbf{g} the vector of residuals $\mathbf{g}(\mathbf{x}, \mathbf{z})$. In Equation 11.4a, the vector of derivatives is integrated with respect to time, while Equation 11.4b demands an iterative computation of the vector of algebraic unknowns.

Systems of Implicit Differential-Algebraic Systems

In the case of a fully implicit set of DAEs,

$$\mathbf{F}(\dot{\mathbf{x}}, \mathbf{x}, t) = \mathbf{0}, \quad (11.5)$$

the problem can be formulated in ACSL as the integration of a vector $\mathbf{x}\text{dot}$ and the computation of $\mathbf{x}\text{dot}$ by iteration.

$$\mathbf{x} = \text{INTVC}(\mathbf{x}\text{dot}, \mathbf{x}0) \quad (11.6a)$$

$$\mathbf{x}\text{dot} = \text{IMPVC}(\mathbf{F}, 0.0) \quad (11.6b)$$

Via a variable indicating the numerical integration algorithm, its default name is `IALG`, the DASSL code can be selected.

Features of the language ACSL beyond the CSSL standard and their support by the simulation program ACSL certainly meet contemporary requirements that arise from problems in various engineering fields. They are, however, proprietary. If we disregard such extensions, then the considered problems with CSSL descriptions of large hierarchical modular models appear because the CSSL standard only covers assignment statements and the latter are ‘merely’ sorted by the translator of CSSL programs. The more general form of equations

$$\langle \textit{Expression} \rangle = \langle \textit{Expression} \rangle \quad (11.7)$$

or

$$\langle \textit{Expression} \rangle = 0.0, \quad (11.8)$$

expected by solvers of implicit sets of DAEs, are generally not supported.

While CSSLs in general have been confined to the formulation of explicit state space models, solvers for linear implicit systems in which the matrix can be singular have been developed in numerical mathematics and have been implemented in software packages, e.g. ODEPACK [45]. Moreover, the solver DASSL for general implicit DAE systems of index < 2 has found wide application. These solvers expect as input the formulation of model equations in a procedural language to be compiled and linked to a main program that calls the solver. That is, while many simulation programs based on the CSSL standard have been confined to explicit state space models, more advanced solvers from numerical mathematics as the core of simulation programs are lacking the support of model development.

Consequently, modelling and simulation software environments, e.g. Dymola, have been developed that are based on a *modelling language* supporting *non-causal* equations in a mathematical sense. These software programs can internally perform

symbolic reformulation of equations where necessary, and generate a state space model if this is possible. The latter can be output in the simulation language ACSL. If a state space model is not possible, then an index one DAE system to be solved by the DASSL code can be generated.

The model description language excepted by this kind of modern advanced modelling and simulation software can be proprietary like SIDOPS, the underlying language of the 20-sim[®] software [25], or public domain like *Modelica* [56]. The most important difference with regard to simulation languages based on the CSSL standard is that these model description languages support *non-causal* equation formulation. Consequently, beyond equations sorting, symbolic formulae manipulation features must be available in supporting programs. In contrast to simulation programs based on the CSSL standard, such advanced programs not only transform the model description into a formulation that uses a programming language such as C. Moreover, they can transform a non-causal model description into a simulation language, e.g. ACSL. The simulation features of such modelling and simulation software are based on powerful solvers from numerical mathematics.

11.2 Object-Oriented Modelling Languages

When simulation of the dynamic behaviour of engineering systems on (digital) computers came into use, simulation languages were introduced to free model description from the necessity of bringing mathematical equations formulated as FORTRAN assignment statements in a sequential order required by the computer and not induced by the engineering problem under consideration. Since the early days, the ever increasing performance of computers enables simulation of systems of increasing size. To cope with an increasing model size, it became necessary to describe models in a hierarchical and modular way. At the same time, limitations of the CSSL standard established in 1967 and of CSSL based simulation programs became apparent. This has led to the introduction of languages that support a reliable development of large models. In order to point out the importance of the modelling process, the notion *modelling languages* is commonly used nowadays.

Inspired by the object-oriented programming paradigm supporting the development of large software systems [67], modelling languages have come up to support the development of large engineering systems in a way called *object-oriented physical systems modelling*. The characteristics of such a modelling approach have already been explained in Section 1.3. Here, they are briefly recalled.

- Interconnection of Submodels
Submodels are plugged together according to interconnections of their corresponding components in the real engineering system.
- Model Hierarchy
Physical system models are hierarchical and modularly structured.

- **Encapsulation of Knowledge**
Knowledge about a subsystem is encapsulated in the corresponding submodel. Only the part of information about a submodel needed by other submodels can be accessed via well defined interfaces. For instance, the body of a submodel in which its behaviour is described by means of certain characteristics does not need to be known when submodels are connected.
- **Instantiation from Model Classes**
Submodels come into existence by instantiation from generic model classes.
- **Inheritance of Submodel Properties**
If a submodel is used in the declaration of a new submodel, all its properties are inherited by the submodel.

In the following, these characteristics will be considered more closely from a bond graph modelling perspective. The discussion will make use of two model description languages, namely SIDOPS (particularly designed to support bond graph-based physical systems modelling) and *Modelica* (the outcome of an international effort to combine features of present object-oriented modelling languages into a unified language that facilitates exchange and reuse of models). A brief description of modelling languages that explore the concepts of object-oriented physical systems modelling is given in Section 5 of the Modelica Tutorial and Design Rationale, version 1.3 of December 1999².

11.2.1 Connection of Submodels According to the Physical Structure of the System

The interconnection of submodels according to the physical structure of a system, as is common in bond graph modelling as well as in network modelling, means that some interface variables of submodels are set to be equal. For instance, if for several electronic devices a pin is selected and if all these pins are connected to a joint circuit node, then all pins have the same potential and currents add up to zero according to Kirchhoff's current law. Generalised to corresponding power variables in other energy domains, this law holds if, e.g. a hydraulic pump is connected to a line or if in mechanics two components are linked. Forces acting on the connecting point add up to zero and both components have the same velocity in that joint point. As previously explained, it is the connection of two submodels and the properties that decide whether an interface variable can be computed by means of the equations in the body of a submodel or the variable has to be provided by a submodel connected to the submodel. That is, the specific form of equations, Equation 11.2, must be dismissed in favour of the more general form, Equation 11.7. In other words, equations must be declarative or *non-causal*. Therefore, in Modelica, the constitutive equation of a linear 1-port C energy store, for instance, can always be given in the form

² Latest versions of documents on Modelica may be downloaded from <http://www.Modelica.org>

$$C * \text{der}(u) = i, \quad (11.9)$$

independently of the port causality the element assumes in a present context (In Equation 11.9, der denotes the differentiation operator d/dt). Giving another example, Ohm's law can always be specified in the *acausal* form

$$0 = u - R * i, \quad (11.10)$$

where R denotes the resistance. Consequently, equations cannot be sorted immediately into a computational order. Rather, first it must be checked which variable is determined by which equation. This investigation corresponds to causality assignment in bond graph modelling after all submodels have been expanded. If every unknown is related to one single equation, then the transformation of equations into a list of sorted CSSL statements corresponding to a state space model depends on whether subsystems of implicit simultaneous equations are linear in their unknowns and whether the effort for solving them symbolically is acceptable in practice.

11.2.2 Algebraic Loops

In case algebraic loops appear due to the connection of submodels, then the subsystem of simultaneous equations can be symbolically solved, as far as it is linear in its unknowns. In order to avoid long complex symbolic expressions that require considerable storage, auxiliary variables are usually introduced. In Section 5.4, it has been shown how the processing of algebraic constraints can be supported by inserting additional residual sinks into a bond graph indicating *tearing* variables. As a result, the set of algebraic constraints can be reduced to one or several small sets of equations for tearing variables. The latter can be symbolically solved if they are linear in the tearing variables.

11.2.3 Algebraic Dependencies Between State Variables

If state variables in different submodels become dependent due to the connection of submodels, then this means that the overall mathematical model has the form of a DAE, possibly of higher index. As explained in Section 5.2, the index can be reduced by symbolic differentiation of algebraic constraints. Differentiation introduces new derivatives of variables. Consequently, equations determining these variables must be differentiated as well. In the modelling and simulation program Dymola[®] [29], Pantelides' algorithm for determination of a set of consistent initial conditions [61] is used to determine a minimal number of equations that must be differentiated symbolically to reduce the index of the DAE system. When discussing the problem of numerical drift, it has been observed that with regard to stabilisa-

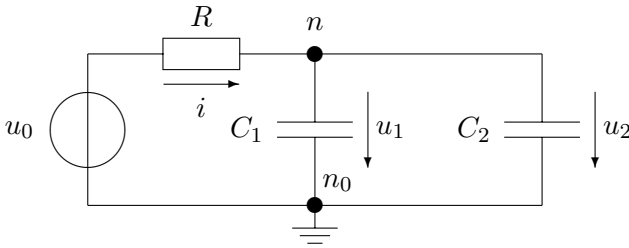


Fig. 11.5 Two capacitors connected in parallel

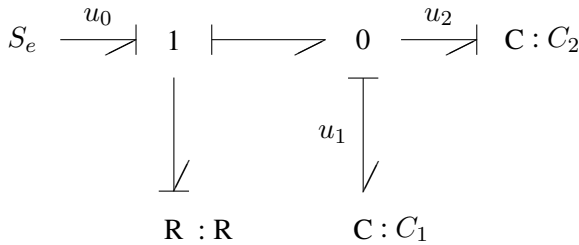


Fig. 11.6 Corresponding bond graph of the circuit in Figure 11.5

tion of the numerical solution, it is reasonable to add the initial constraints to the DAE system of reduced index. This leads to an over-determined set of equations for the Newton-Raphson iteration. As a remedy of this problem, the so-called *dummy derivative method* is applied in the program Dymola[®]. This method conceives the derivatives of some variables as new *independent* algebraic variables. That is, it ignores the fact that they are the derivatives of variables. Thus, it can be achieved that the number of unknowns equals the number of equations. This approach, however, requires a decision on which variables are to be considered state variables.

For illustration of this method, consider the well known example of two capacitors connected in parallel (Figure 11.5). Figure 11.6 shows the corresponding bond graph. The algebraic dependency between the two states \$u_1, u_2\$ is indicated in the bond graph by a causal path between the power ports of the C stores. By looking at the causal bond graph, we get the following equations.

$$i = \frac{1}{R} (u_0 - u_1) \tag{11.11a}$$

$$\dot{u}_1 = \frac{1}{C_1} (i - C_2 \dot{u}_2) \tag{11.11b}$$

$$u_2 = u_1 \tag{11.11c}$$

These equations are an index 2 DAE system with regard to the three unknowns \$u_1, u_2, i\$. If the algebraic constraint Equation 11.11c is differentiated with respect to

time and if the result is added to the equations derived from the bond graph, then we have four equations for three unknowns. In the bond graph, integral causality at the C energy store of element C_1 apparently leads to derivative causality at the second C store. Consequently, if u_2 is not considered a state variable any more and if it is ignored that \dot{u}_2 is the derivative of u_2 , then with $z := \dot{u}_2$, we have four equations for the four unknowns i , \dot{u}_1 , u_2 and z . Since the set of equations

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & C_1 & 0 & C_2 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} i \\ \dot{u}_1 \\ u_2 \\ z \end{bmatrix} = \begin{bmatrix} (u_0 - u_1)/R \\ 0 \\ u_1 \\ 0 \end{bmatrix} \quad (11.12)$$

is linear with a non-singular coefficient matrix, it can be solved symbolically. As a result, the initial DAE system can be transformed into an explicit state equation for the single state variable u_1 . For nonlinear equations, it is generally not possible to reduce an index 1 DAE system into a state space model.

In Section 5.4, the problem of algebraically dependent stores was reduced to algebraic loops between port variables of resistors.

11.3 Bond Graph Modelling from an Object-Oriented Point of View

Bond graphs were introduced by Paynter in as early as 1959 and subsequently elaborated into a methodology by Karnopp and Rosenberg [48] at a time when an object-oriented-approach to physical systems modelling still did not exist. Nevertheless, although notions adopted by object-oriented modelling (OOM) from object-oriented programming (OOP) in software engineering have not been used in bond graph language, features of the object-oriented modelling paradigm can also be found in bond graph modelling. In fact, from a present-day point of view, bond graph modelling may be considered a special form of object-oriented physical systems modelling. This view, shared also, e.g. by Broenink [20], is briefly explained in the following.

- **Objects**

The notion of an object is uncommon in bond graph language. However, models of technical components or of basic physical processes are actually used as objects. From today's point of view, these models represented by nodes in bond graphs may be considered objects.

- **Interconnection of Submodels**

It is an essential feature of bond graphs that submodels are connected according to the way their corresponding engineering components exchange energy. Therefore, the structure of a bond graph shows a strong topological affinity to the physical structure of a system before the graph is simplified. As a consequence, computational causality at the power ports of a submodel depends on the connections of a submodel to other submodels. That is, the internal func-

tional description of the dynamic behaviour of a submodel must be *non-causal*. In order to be able to adjust constitutive equations according to computational causality assigned to the submodel's power ports, functions establishing the constitutive equations must have a unique single valued inverse and must be sufficiently smooth. As a result, non-causal model description has been a feature of bond graph modelling long before it was identified by the OOM paradigm as a need to overcome limitations of CSSLs.

- **Model Hierarchy**
Hierarchical model structuring is a well known concept to cope with large-scale systems. It has a long tradition and is used in various formalisms, e.g. iconic diagrams, networks, block diagrams and in bond graphs. On the hierarchy levels above the level of basic elements, submodels can be represented by words in bond graphs. Such hierarchical bond graphs are called *word bond graphs*. Basic elements are described by means of equations. A submodel above the level of basic elements is described by a bond graph or may also be given by a set of equations as well if it is a behavioural model with no internal structure. *Word bond graphs* support a systematic top-down design of a structured model of a complex system guided by the consideration of the energy exchange between system components. Since bonds connecting power ports can carry information about the reference direction of the energy flow and also information about computational causality, word bond graphs are *not* merely iconic diagrams with icons reduced to an alpha-numeric mnemonic code. Word bond graphs may be regarded rather an intermediate format between an iconic diagram with application specific icons and a mathematical model of the system.
- **Encapsulation of Knowledge**
Submodels in bond graphs representing either system components or elementary physical processes can be accessed only via their interfaces called power ports or signal ports. A storage element, for instance, does not pass information about its state via its power ports to adjacent vertices. That is, the state variable represents encapsulated information (For convenience, in the model description language SIDOPS, a store's state can be accessed via a signal port). Moreover, for interconnection of submodels, the constitutive equations in the body of a submodel description do not need to be known. Bond graph modelling is non-causal. For causality assignment supporting the organisation of a mathematical model, it is sufficient to know the type of an element. The exact form of its constitutive equations is not needed. Hence, what is kept in the body of a submodel description can be viewed as encapsulated.
- **Instantiation from Model Classes**
At the lowest hierarchy level, there is a classification of energy processes into energy storage, power conserving distribution of energy, and transport of energy, transduction of energy into another form, especially transformation into heat. The class of each type of element is denoted by a reserved symbol. A node in a bond graph representing a basic process may be viewed as an instantiation of its corresponding model class. It is a special member of the class it belongs to. Its constitutive equations and parameters characterise the particular submodel and

distinguishes it from other instantiations of the model class. If a vertex of a bond graph represents a component model, most often the latter is an instantiation of a model class from a library augmented by specific parameters.

- **Inheritance of Submodel Properties**

The model super class *store*, for instance, just captures the fundamental properties of passivity of storing a physical quantity, e.g. electrical charge, and of being energy conservative. With regard to the constitutive equations, it is only determined which variables are involved. Moreover, a preferred computational causality may be assigned to its power ports. This causality restriction at a power port only means that one of its two conjugate power variables is related to the integral with respect to time of the other one. The actual functional (linear or nonlinear) dependency however as well as (additional) parameters are specified in the model of an energy store under consideration. While the state is encapsulated information that is not passed via the ports of a storage element, it can be inherited from the superclass *store* for use in the formulation of the store's constitutive equations.

As this brief comparison shows, bond graph based physical systems modelling and object-oriented modelling both have much in common, even though the first approach is much older. Most importantly, both paradigms support

- *domain independent,*
- *hierarchical,*
- *non-causal modelling.*

In bond graph modelling, the concept of computational causalities supports the generation of various forms of mathematical models from a generic bond graph model.

11.4 Describing Bond Graphs in SIDOPS

A model description language particularly designed for supporting bond graph based physical systems modelling is SIDOPS [18]. The acronym stands for *Structured Interdisciplinary Description Of Physical Systems*. It is the underlying textual language of the integrated modelling and simulation environment 20-sim[®] (Twente sim) [20]. In the 20-sim[®] environment, SIDOPS enables the specification of the functional behaviour of a user-defined element or of a component model, e.g. an orifice model in hydraulics, to be added to a collection of predefined library models kept in folders. As to the creation of equation models, SIDOPS is close to common mathematical formulation. The language supports many mathematical functions by means of built-in functions. Its use is intuitive. In addition, the 20-sim[®] equation editor is language sensitive and syntax violations are highlighted while typing. In general, however, model development takes place at a graphical level. For that purpose, the graphical user interface (GUI) of 20-sim[®] supports several graphical formalisms.

In the following, some essential features of SIDOPS will be considered in order to give an impression of the language's potential without the need of explaining too many details of its syntax. For further details of the language, it is referred to the Reference Manual that can be downloaded from the 20-sim website [25].

SIDOPS supports the development of hierarchical, modular structured models and encapsulation of knowledge. For hierarchical modular model description, classes are used. A SIDOPS model class captures all non-changing, permanent properties of a submodel. Initial conditions and parameters are specified by names. When a model class is used (instantiated into a particular submodel), a copy is made and provided with a set of actual parameters. In SIDOPS model classes, the definition of the interface of an object to its outside world is separated from the description of the behaviour. The interface definition part may be followed by a list of subclass definitions.

In the heading of a class definition, a list of static parameters of data type *integer*, so-called *class parameters*, may be specified. This way, it is possible to describe multiport elements without the need for specifying the number of ports in the model class declaration. This is important for the declaration of a class for 0- or 1-junctions, respectively.

The interface declaration part is introduced by the keyword **interface**. In this section, power ports are declared by means of the keyword **ports**, while keywords **inputs**, or **outputs** are used for declaration of signal ports. Constraints with regard to the power flow reference direction or with respect to causality are treated as port attributes specified by keywords (Figure 11.7). An instantiation of a model class can communicate with other models only via the declared interface. The local names of power variables of a port are prefixed by the port's name, viz. <Port-Name>.e or <Port-Name>.f. An exchange of information via global variables is not supported; the only global variable is the time.

In the subclasses declaration section, starting with the keyword **subclasses**, all subclasses used in a class are listed line by line. On each line, the subclass name is followed by a unique name for the copy of that class where the name is local within the class.

The body of a class may contain either so-called *connection lines* that describe the interconnection of subclasses, or equations. In the first case, the class is called a *topological class* and an *equation class* in the second case. Equation classes can have several parts for declaration of parameters, constants and local variables and finally an equation part where constitutive equations are formulated. The latter are declarative and will be rewritten during model processing if necessary.

As an example, the class declaration of a basic 1-port C energy store is given in Figure 11.7, where keywords of the language are typeset in bold. As the example shows, instantiations of the class C1 have got a power port called p and a signal port that allows access to the state of a C store. For the power port, restrictions with regard to computational causality and the reference direction of the power flow can be declared. Preferred integral causality is indicated by the keywords **preferred effort** p. Since a C energy store is a passive element, the half arrow indicating the power reference direction at the adjacent bond is required to point towards the

```

class C1 version 1
# C1: 1-port C energy store with 1-dimensional port
interface
  ports: p
  causality restrictions
  preferred effort p
  orientation restrictions
  fixed in p
  outputs: real state
parameters
  real C
equations
  state = int(p.f)
  p.e = (1/C) * state

```

Fig. 11.7 SIDOPS class declaration of a 1-port C energy store

power port (**fixed in** p). Restrictions at the power ports of 2-port transformers and gyrators can be formulated as shown in Figure 11.8. For both types of elements, one half arrow points towards the element while the other one is oriented away from it (**constraint** powerIn **notequal** powerOut). In the case of a 2-port gyrator, both causal strokes either point towards the element or away from it (**constraint** powerIn **equal** powerOut).

A SIDOPS class declaration of a 1-junction with the number N of ports as a *formal* class parameter is given in Figure 11.9. The causality specification **constraint 1_effort** means that only at one of the N ports the effort variable can be the output variable. The variable out denotes the flow that enters at one port and is the output variable at all remaining $N - 1$ ports. In addition to its N power ports, the 1-junction has a *signal* port at which the variable out can be accessed (cf. Figure 3.5). During model processing,

$$\text{sum} (P[],e) = 0$$

in the **equations** section in Figure 11.9 is expanded into

$$e_i = \sum_{\substack{j=1 \\ j \neq i}}^N \alpha_j \times e_j$$

and

$$\text{identity} (P[],f) = out$$

is translated into

$$\begin{aligned}
 f_j &= f_i \quad j = 1, \dots, N, \quad j \neq i \\
 out &= f_i
 \end{aligned}$$

```

class TF1 version 1
# TF1: Transformer with two one-dimensional power ports
interface
  ports: powerIn, powerOut
  orientation restrictions
    constraint powerIn notequal powerOut
  causality restrictions
    constraint powerIn notequal powerOut
parameters
  real n
equations
  powerOut.f = n * powerIn.f
  powerIn.e = n * powerOut.e

class GY1 version 1
# GY1: Gyrator with two one-dimensional power ports
interface
  ports: powerIn, powerOut
  orientation restrictions
    constraint powerIn notequal powerOut
  causality restrictions
    constraint powerIn equal powerOut
parameters
  real n
equations
  powerOut.e = r * powerIn.f
  powerIn.e = r * powerOut.f

```

Fig. 11.8 SIDOPS class declarations of 2-port transformers and gyrators

```

class one1 (N) version 1
# one1: 1-junction with N one-dimensional bonds attached
# N: number of power ports a-priori not determined
interface
  ports: P[N]
  causality restrictions
    constraint 1 effort P
  outputs: real out
equations
  sum ( P[ ].e ) = 0
  identity ( P[ ].f ) = out

```

Fig. 11.9 SIDOPS model class of a 1-junction with the number of ports as a formal class parameter

[18]. Note that the value of N is now determined by the number of bonds actually connected to the 1-junction. According to the power orientation of the adjacent bonds, the coefficients α_j are equal to either $+1$ or -1 . Index i denotes the bond at which effort e_i is the output and f_i the input after causality assignment to the bond graph of a system under consideration (Figure 11.10).

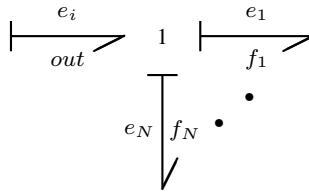


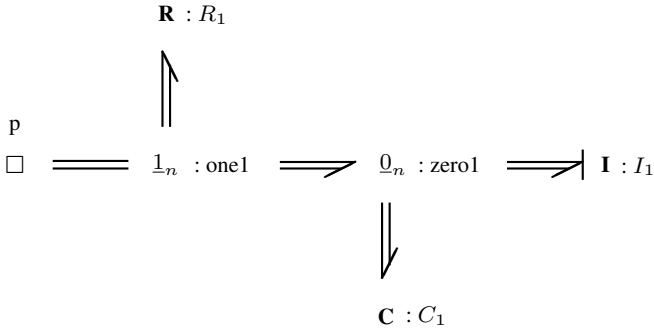
Fig. 11.10 Bond graph 1-junction

For information processing components of a controlled overall system, it is appropriate to describe their behaviour by a set of assignment statements. In order to distinguish assignment statements from equalities in a mathematical sense, SIDOPS has adopted the sign $:=$ from Pascal programming language. However, in a submodel specified by equations, equalities and assignment statements cannot appear mixed.

In a topological class the keyword **structure** starts a list of connection lines. For each subclass in a model class, there is one connection line. Each of them specifies in an intuitive form to which ports of other subclasses the ports of a subclass are connected. That is, these lines correspond with the incidence matrix of a graph. In addition, connections between subclass ports and class ports must be specified. Figure 11.11 shows the body of a SIDOPS class description of a simple bond graph submodel with one power port p as an example. The submodel port p is depicted by the small left-side rectangle \square .

The first connection line in this topological class means that the resistor R_1 is connected to a 1-junction $one1$ by a multibond of dimension n . The orientation of that bond is towards the R element as indicated by the symbol $<$ in the textual description. Causality information can also be coded in an obvious way by means of the symbol $|$ (third line). Apart from the last line, there are exactly as many lines as elements where each connection via a bond appears twice. The last line indicates that there is a connection between the junction $one1$ in the subclass and the port p of the submodel class. Information of the power flow reference direction and the causality at port p are not available. Both attributes are determined when this submodel class is instantiated and connected to another submodel in the bond graph of an overall system model.

In this section, some features of SIDOPS have been outlined. Further developments of this modelling language have been described by Breunese and Broenink [17]. For instance, to help reduce inconsistencies in models, variables and parameters can be annotated with information that can be checked automatically to ensure compatibility in equations. Parameters and constants may be assigned a physical type. For instance, if h denotes a time step, the physical type $\langle\langle\text{time}\rangle\rangle$ can be added as an annotation. If values of variables and parameters must be in a certain range, e.g. a time step apparently must be positive, then this can be specified in a



structure

- R1** <=n= one1
- C1** <=n= zero1
- I1** |<=n= zero1
- one1** (=n= p, =n=> R1, =n=> zero1)
- zero1** (<=n= one1, =n=> C1, =n=>| I1)
- p** =n= one1

Fig. 11.11 Multibond graph and corresponding topological class in SIDOPS

constraints section. Moreover, constants can be annotated with their physical units, e.g. $g = 9.81 \ll m/s^2 \gg$.

Furthermore, models can be described on three levels of abstraction. All three of them may be combined in the description of an overall model. Thus, multiformalism is supported. At the highest level of abstraction, a system model can be described as an interconnection of system components. At the intermediate level of abstraction, the interactions of physical processes, taking place in the components, are modelled. Formalisms for graphical representation can be bond graphs, block diagrams, or iconic diagrams. At the lowest level of abstraction, basic physical processes are to be described by means of equations.

In order to support modelling of digitally controlled time continuous systems, the two functions *sample* and *hold* are available as language constructs. Ultimately, the design of the extension of SIDOPS denoted as SIDOPS+ intends to support the exchange of model information among multiple software tools. With that regard, SIDOPS+ is neither limited to a particular software nor to special application fields. A formal definition of the syntax of the language is given by Breunese in appendix B of his Ph.D. thesis [16].

11.5 Describing Bond Graphs in Modelica

As mentioned in Section 11.2, the modelling language Modelica is the result of an international, especially European effort to combine features of several textual object-oriented modelling languages into a new equation oriented so-called unified object-oriented modelling (OOM) language. A major goal of the design of *Modelica* has been to overcome the limitations of the old CSSL standard with regard to the development of large scale models, to promote the *exchange* of physical system models between various (proprietary) simulation packages and the *reuse* of models, and to introduce by that way a new de facto standard.

The core language specifications were completed in December 2000. Further development of Modelica and of Modelica libraries is organised by the non-profit, non-governmental Modelica Association [56]. The language and some libraries are freely available. The language is supported by the commercial modelling and simulation software Dymola[®] [29, 31] as well as by the open source software OpenModelica [62] developed at the University of Linköping in Sweden.

Some textbooks on object-oriented physical system modelling with Modelica have been published [36, 80]. Besides the software, a tutorial and a user's guide can be downloaded from the web site of the OpenModelica Project [62]. In December 2007, The OpenModelica Consortium was founded supporting the development and the promotion of OpenModelica. As a powerful general-purpose OOM language, Modelica has received much attention in various application areas.

The description of basic bond graph elements and of their interconnection in a textual object-oriented modelling language like Dymola or Modelica is rather straightforward [12, 13, 19, 22], although there are also some restrictions. A general-purpose Modelica bond graph library has been developed by Cellier and Nebot [23].

In the following sections, we will consider features of Modelica relevant for the description of bond graph power ports, of basic bond graph elements and of hierarchical bond graphs [9].

11.5.1 Bond Graph Power Ports and Their Interconnection

With respect to the modelling of energy flows, Modelica relies on the mobility analogy and the concept of generalised networks. Hence, submodel interfaces to the outside world are *pins*, not power ports as in bond graphs. The power variables of an interface are called *across* and *through* variables [33]. They correspond to *efforts* and *flows* as used by bond graph modellers. Correspondences between power variables in multiple domains according to the mobility analogy are given in Table 11.1.

All information needed for the description of a submodel interface is encapsulated in a special model class called **connector**. Contrary to bond graph power ports, connectors in Modelica in addition to power variables may also pass on other quantities like generalised displacements and/or accelerations. Moreover, since Modelica is network oriented, along with the connection of two interfaces, normally the flows

Table 11.1 Correspondence of power variables from different energy domains according to the mobility analogy

Across	Through
Velocity Angular velocity Voltage Pressure Temperature	Force Torque Current Volume flow Entropy flow

```

connector PowerPort    "bond graph power port"
// Version 2:
// Both effort and flow are treated as across variables
  Real e "effort variable";
  Real f "flow variable";
end PowerPort ;

```

Fig. 11.12 Description of a bond graph power port in Modelica

```

connector PowerPort    "bond graph power port"
// Version 1:
  Real e    "effort variable";
  flow Real f "flow variable";
end PowerPort ;

```

Fig. 11.13 Inappropriate declaration of a bond graph power port in Modelica

involved are added up to zero. This is *not* appropriate for bond graphs because (separate from the connection of power ports by bonds) the summation of flows is carried out in a special bond graph element, the 0-junction. However, Modelica allows for the suppression of this summation of flow (or through) variables by introducing two variables of type Real (Figure 11.12), or, in other words, by omitting the keyword **flow** usually associated with the declaration of the flow variables of an interface (Figure 11.13).

This keyword means that with the interconnection of two interfaces, their flow variables are added up to zero. If the prefix **flow** is missing, then corresponding power variables of the two power ports connected together are set to be equal. The summation of flows is not carried out. If, e.g. $M_i.A$ denotes port A of submodel M_i ($i = 1, 2$), then the interconnection of the two power ports $M1.A$ and $M2.A$ is expressed by the statement

```
connect(M1.A, M2.A) .
```

If version 2 of the Modelica declaration of a bond graph power port is used, then the **connect** statement merely represents a port to port connection and may be used to describe the bonds in a bond graph. Otherwise, if version 1 of the Modelica

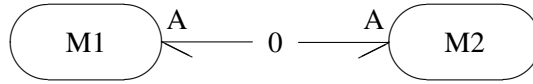


Fig. 11.14 Connection of bond graph power ports if the keyword **flow** is not suppressed in their declaration

declaration of a bond graph power port is used, the **connect** statement is equivalent to the equations

$$\begin{aligned}
 M1.A.e &= M2.A.e \\
 0 &= M1.A.f + M2.A.f,
 \end{aligned}$$

which means that the bonds connected to the ports must be connected by a 0-junction as shown in Figure 11.14.

For the interconnection of more than two power ports, bond graphs provide two types of junctions corresponding to Kirchhoff’s current law or voltage law. Both types of junctions allow for a finite, but *a-priori not determined* number of ports. If a 0- or 1-junction is used in a bond graph, then its number of ports is determined by the number of ports of other elements it is connected to. Based on the concept of generalised networks, the object-oriented modelling language Dymola [30], a major predecessor of Modelica, provides an element called *node* with an undetermined number of interfaces. Thus, it corresponds to a 0-junction in bond graphs. However, a built-in model that corresponds to a bond graph 1-junction and that allows for an undetermined number of ports is not available either in Dymola or in Modelica (A separate element that represents Kirchhoff’s voltage law, extended to across variables, is not needed in generalised networks).

One way to overcome the problem of a missing element corresponding to a bond graph 1-junction with an undetermined number of ports is to replace a 1-junction by a 0-junction and to switch the role of the power variables at each adjacent bond by means of a so-called *symplectic gyrator* (ratio $r = 1$) as has been proposed by Cellier for the language Dymola [22] (Figure 11.15).

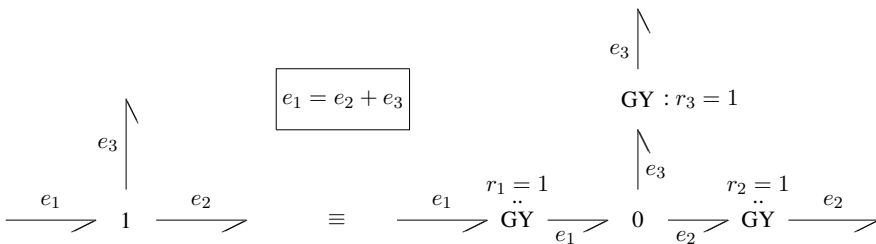


Fig. 11.15 Submodel equivalent to a 3-port 1-junction

```

model One3P "3-port 1-junction"
  PowerPort port1, port2, port3;
equation
  // all flows are equal
  port2.f = port1.f;
  port3.f = port1.f;
  // all efforts add up to zero
  0 = direction(port1) * port1.e +
      direction(port2) * port2.e +
      direction(port3) * port3.e;
end One3P;

```

Fig. 11.16 Modelica class of a 3-port 1-junction

However, especially for bond graph modelling of mechanical systems, such an approach is not functional since a systematic construction of a bond graph model of a mechanical system starts with representing distinct velocities by 1-junctions. Once the bond graph development has been completed, all 1-junctions would have to be replaced by their equivalent according to Figure 11.15. Apparently, this step would introduce many symplectic gyrators that do not correspond to physical phenomena. Hence, by this method, the close affinity between the bond graph and a system schematic as well as the expressiveness of the bond graph are obscured.

Alternatively, a model class may be introduced that depends on the number of bonds connected to the junction. That is, for each number of power ports, a 1-junction model class is defined which may be stored in a library [13], although this approach also is not fully satisfying.

As an example, a Modelica description of a 3-port 1-junction is depicted in Figure 11.16. In the **equation** part of the model `One3P`, **direction** is a built-in function that can be used to account for the orientation of a bond. If the bond starts from `port1` and ends at `port2`, specified by the statement `connect(port1, port2)`, then **direction**(`port1`) returns the value -1 , and **direction**(`port2`) gives the value 1 . A more elegant approach is to define an array of power ports as shown in Figure 11.17. If the model `one` of a 1-junction shown in Figure 11.17 is instantiated, then the default value for the number of adjacent bonds must be replaced by the actual number.

```

one J1 (N=4) ;

```

In order to support 1- and 0-junctions for which the number of power ports is determined by the actual number of adjacent bonds, the dimension of the array of power ports cannot be determined before the development of the bond graph has been completed. If another 1-port element is added to the 1-junction `J1`, the actual number of adjacent bonds, `N`, must be updated.

This problem with bond graph 1- and 0-junctions does not appear in the Modelica description of generalised networks. Consider, for instance, the simple circuit depicted in Figure 11.5 and assume that a class `Pin` has been declared as shown in Figure 11.18. If the class `Pin` is used in the declarations of a resistor and of a

```

model one "1-junction"
  /* N : Number of power ports of a 1-junction,
     depends on the number of adjacent bonds,
     must be at least two.
  */
  parameter Integer N = 2;
  PowerPort ports[N] ;
  Real sum = 0.0;
equation
  for i in 1 : N loop
    // all flows are equal
    ports[i].f := ports[1].f ;
    sum := sum + direction(ports[i]) * ports[i].e ;
  end for;
  // sum of efforts equals zero
  sum = 0 ;
end one ;

```

Fig. 11.17 Modelica class of a 1-junction using an array of power ports

```

model Pin
  Real v "potential of the pin";
  flow Real i ;
end Pin ;

model Resistor
  Pin p, n ; // p: "positive", n: "negative" pin
  Real u "voltage drop across the two pins" ;
  Real i "current through element" ;
  parameter Real R (unit="Ohm") "Resistance"
equation
  0 = p.i + n.i ; // Positive currents into components
  u = p.v - n.v ;
  i = p.i ;
  u = R * i ;
end Resistor ;

```

Fig. 11.18 Modelica class of a pin and its use in a resistor model class

capacitor, then the connection of the three elements R, C1 and C2 according to the circuit of Figure 11.5 is simply expressed by three **connect** statements.

```

connect(R.n, C1.p) ;
connect(R.n, C2.p) ;
connect(C1.p, C2.p) ;

```

Connectors in Modelica versus Ports in Bond Graphs

As has been mentioned at the beginning of this subsection, connectors in Modelica can include any variable appropriate for describing the connection of a submodel interface to another. That is, not only power conjugated variables, but also other quantities may be passed via connectors from one submodel to another. In [32], for instance, the authors declare a connector of a mechanical shaft model that passes an angle (generalised displacement) and a torque (considered as a flow). Moreover, for the description of multibody systems, it is proposed to pass not only power variables, but also displacements and accelerations [60]. With this flexibility in the modelling language, quantities needed in the equations of a submodel can be provided directly. On the other hand, the expressiveness of a graphical model description may be obscured as there is no clear distinction between power and signal ports. Consequently, it may not be fully clear which information is passed on from which submodel. Moreover, by using information passed on in addition to the power variables, it is possible to use (ad hoc) constitutive equations in a submodel which are not in accordance with physical conservation laws. If a connector does not pass on both power variables, power continuity is not ensured by the interconnection mechanism. In that case, in addition to the evaluation of the constitutive equations, power must be calculated inside a submodel and passed to submodels connected to it. Energy flowing into and out of a submodel must be checked.

In contrast, bond graph modelling is more stringent. Power is passed through power ports and signals through signal ports. It is an essential feature of bond graphs that power continuity is expressed explicitly by the junction structure. If signal modulation of elements is used with care, standard bond graph elements and the way they are allowed to pass information help to avoid setting up model equations that may violate the energy balance.

From these observations, it appears that not every Modelica description of a dynamic system can be translated into a bond graph. The description of a bond graph model in Modelica is feasible, although the fundamental 1-junction with an undetermined number of ports so far appears to be insufficiently supported. In the following, it is demonstrated how encapsulation and inheritance can be exploited even in the definition of basic bond graph models.

11.5.2 Basic Bond Graph Elements

In OOM, the concept of *encapsulation of knowledge* and *inheritance* supports the reliable development of submodels to be stored in libraries and reused for developing large scale models. For introduction of a submodel, it is not necessary to rewrite or to modify parts of the definition of another submodel. Joint properties are inherited from a superclass and only features that are special for a submodel to be introduced are added.

```

partial model passiveOnePort
  PowerPort p;
equation
  // ensure that the number of bonds is one
  assert(cardinality(p) == 1, "");
  // ensure that the orientation of the power flow is inward
  assert(direction(p) == 1, "");
end passiveOnePort;

```

Fig. 11.19 Encapsulation of the property of a passive 1-port element

```

partial model energeticOnePort
  extends passiveOnePort;
  Real state "conserved quantity";
  Real rate ;
equation
  // der(state): time derivative of state
  der(state) = rate;
end energeticOnePort;

```

Fig. 11.20 Encapsulation of the energetic property of 1-port energy stores

Both mechanisms can be used favourably even in the definition of the basic bond graph models. For instance, resistors and energy stores share the property of being passive. This suggests the encapsulation of this joint property in a superclass that is inherited in the definition of a resistor or an energy store. In the following, the presentation is confined to 1-port elements for the sake of simplicity. In bond graphs, passivity of resistors and energy stores is accounted for by the convention of an inward positive reference direction of the energy flow at the element port (Energy flows into the port when the product of both power variables e and f is positive). A Modelica class capturing the property of a passive 1-port element is shown in Figure 11.19.

The definition of a model or submodel class starts with the keyword **model**. Since constitutive equations are missing, the model class `passiveOnePort` cannot be instantiated into the model of a passive element. This is indicated in Modelica by the keyword **partial**. In the definition given in Figure 11.19, the keyword **assert** denotes a built-in function that evaluates a Boolean expression and returns an error message in case the value of the expression is false, while the built-in function **cardinality** returns the number of connections to a connector `p`.

The superclass `passiveOnePort` may be used to introduce a subclass `energeticOnePort` in which the energetic property of an energy store is encapsulated in the sense that an energy store conserves a quantity, e.g. charge (Figure 11.20).

The still incomplete model class `energeticOnePort` inherits the properties of the superclass `passiveOnePort` which is expressed by the clause **extends**. The keyword corresponds to **inherit** in the language Dymola. The energetic property can be inherited by the definition of an energy store. According to the type of the energy store (either C- or I-element), the internal variables *rate* and *state* must be

```

model progressiveSpring
// nonlinear one-port C-element
  extends energeticOnePort;
  parameter Real k = 1.0 "spring stiffness";
equation
  rate = p.f;
  p.e = k * (state)^3;
end progressiveSpring;

```

Fig. 11.21 Modelica description of a 1-port C energy store using inheritance

```

partial model PowerConservativeTwoPort
  PowerPort PowerIn, PowerOut ;
equation
  // ensure that the number of adjacent bonds at port PowerIn is one
  assert(cardinality(PowerIn) == 1, " ");
  // ensure an inward reference direction of power at port PowerIn
  assert(direction(PowerIn) == +1, " ");
  // ensure that the number of adjacent bonds at port PowerOut is one
  assert(cardinality(PowerOut) == 1, " ");
  // ensure an outward reference direction of power at port PowerOut
  assert(direction(PowerOut) == -1, " ");
end PowerConservativeTwoPort ;

```

Fig. 11.22 Encapsulation of power conservation in a 2-port element

```

model TF2P "two-port transformer"
  extends PowerConservingTwoPort;
  parameter Real m = 1.0 "modulus";
equation
  PowerIn.e = m * PowerOut.e
  PowerOut.f = m * PowerIn.f
end TF2P;

```

Fig. 11.23 Modelica class of a 2-port transformer

related to the power port variables and the constitutive law of the energy store must be expressed. As an example, the Modelica description of a 1-port C energy store representing a mechanical nonlinear spring is shown in Figure 11.21.

Finally, the property of transformers and gyrators being power conservative may be encapsulated in a joint superclass `PowerConservativeTwoPort` that is inherited in their model definition (Figures 11.22, 11.23).

The Modelica model classes `PowerPort`, `passiveOnePort`, `energeticOnePort` and `PowerConservativeTwoPort` introduced in this subsection are available in a **package** `BondGraph` that is part of the Modelica standard library, where **package** is also a keyword in Modelica. It denotes a special kind of class as does the keyword **connector**.

11.5.3 Computational Causality

Bond graph modelling as well as generalised network based modelling support an interconnection of submodels according to the structure in which corresponding real subsystems are connected. Consequently, model equations must be non-causal. Hence, organisation of equations derived from a graph into a form that is suitable for efficient numerical solution requires symbolic reformulation of some equations or even the symbolic solution of small linear subsystems of equations. The bond graph methodology supports this generation of equation systems by means of the concept of computational causality. It is an essential feature of this concept that all kinds of information can be obtained directly from a causally completed bond graph, including information about the form of a mathematical model, information about structural controllability of a system, or the derivation of transfer functions by means of Mason's loop rule. All this has been demonstrated in previous chapters.

In contrast, the philosophy behind Modelica relies on algorithms that operate on bipartite graphs reflecting the structure of the set of non-causal equations. A first step is to decide which variable is determined by which equations. This problem is called finding an assignment or an output set. The next step is to sort the equations into computational order and to partition the overall set of equations into minimal sets of equations that can be solved simultaneously. This problem is known as finding the strong components in the associated bipartite graph. The result is a lower block triangular structural Jacobian matrix. An efficient algorithm for finding the strong components of a graph has been given by Tarjan [75]. After these algorithms have been applied, it is known in which order equations are to be solved and which equation is to be solved for which variable. Subsets of equations that are linear in their unknowns can be solved symbolically and thus converted into assignment statements. Obviously, this kind of symbolic preprocessing of a set of non-causal equations provides information about causalities and algebraic loops.

Since this processing of model equations can be performed automatically, there is no need for modellers to explicitly specify computational causalities in the Modelica description of a model. Likewise, bond graph software also does not require the modeller to assign causalities because algorithms for causality assignment and propagation have been implemented in the software. If the model is of the form of a set of DAEs, algebraic constraints can be differentiated symbolically with respect to time in order to reduce the differential index or even to convert the model into ODE form. As mentioned above, Pantelides [61] developed an efficient algorithm that can be used to determine which equations are to be differentiated. Mattson and Söderland [52] proposed an algorithm for the selection of state variables, the so-called *dummy derivative method*.

As there are powerful algorithms for fully automatic symbolic processing of a set of non-causal model equations, Modelica does not support computational causalities constraints specified as attributes of interfaces as in SIDOPS. Nevertheless, although the modelling and simulation software Dymola[®] supports automatic selection of state variables, the language Modelica enables control of this selection for various reasons. For instance, as there may be several sets of possible state variables from

a mathematical point of view, some may necessitate inversion of nonlinear characteristics while others do not. Furthermore, an appropriate choice of state variables can lead to a model of less complicated form. Eventually, different choices of state variables can imply that the model has considerably different properties with respect to its numerical computation.

In Modelica, a variable can be assigned an attribute `stateSelect`. Its value indicates whether the variable should be considered a state or not. Possible values of `stateSelect` are

```

always:  do use the variable as a state
prefer:  prefer the variable as a state over those having the value default
default: if no derivative of the variable appears in the model, do not use it as a
         state
avoid:   avoid using the variable as a state in favour of those having the value
         default
never:   do not use variable as state variable

```

The following declaration illustrates how these values can be specified in Modelica.

```
Real v(stateSelect = StateSelect.always);
```

Apparently, if by this way two variables are chosen as states that turn out being algebraically related, then this leads to an error. Furthermore, if several variables have been indicated being possible states, `stateSelect = StateSelect.prefer`, then there is an ambiguity in the selection of states. The value `default` indicates that algebraic variables in a model are never used as states. If, for instance, the kinematic displacement x of a point mass is only used for monitoring purposes, it should not be considered a state. This can be expressed in Modelica in the following way.

```

Real x(stateSelect = StateSelect.never);
Real v = der(x);

```

The operator **der** denotes differentiation with respect to time. Thus, the kinematic displacement x is just the integral of the velocity of a point mass, but not a state variable.

Finally, as demonstrated in the previous subsection, description of basic bond graph elements in Modelica is fairly straightforward. If a model uses ideal switches, computational causalities cannot be statically assigned prior to automatic generation of equations because they dynamically depend on switch states. This has given rise to several proposals in the literature as how to model hybrid systems properly in a bond graph framework (cf. Chapter 7). While in variable causality bond graphs the concept of computational causality loses some of its virtue, variable causality is easily expressed in the textual modelling languages Dymola and Modelica. One way of describing an ideal switch in Modelica is shown in Figure 11.24. In the Modelica model class of an ideal switch, the variable `switchstate` denotes a Boolean signal that controls the switch as depicted in Figure 11.25.

```

model IdealSwitch "ideal switch"
  PowerPort p;
  input Boolean switchstate;
  equation
    0 = if switchstate then p.f else p.e;
  end IdealSwitch;

```

Fig. 11.24 Description of an ideal switch in Modelica

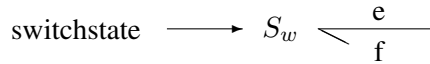


Fig. 11.25 The ideal switch as bond graph element of variable causality

11.5.4 Hierarchical Bond Graphs

Once the basic bond graph elements have been described in Modelica and stored in a bond graph library, it is an obvious step to use them for the description of bond graph models of standard components in an application area and to store those descriptions in a library for that application area. Normally, a number of models of different complexity are provided for standard components accounting for different aspects. This is a way to make models as accurate as needed in a context under consideration and to keep them as simple as possible at the same time. A general-purpose Modelica bond graph library has been developed by Cellier and Nebot and presented in 2005 [23]. It can be downloaded from the home page of the Modelica Association [56]. For hydraulic systems, the description of bond graph models of standard hydraulic devices in Modelica have been briefly presented in [14].

Modelica Description of a Bond Graph Model of a Hydraulic Drive

For illustration of how bond graph models of mechatronic systems can be described in Modelica, consider the example of a controlled hydraulic drive depicted in Figure 11.26. The mechanical load connected to the hydraulic actuator may be an application specific complex mechanical subsystem and therefore has not been specified in this example.

Bond Graph Model and Modelica Description of the Hydraulic Pump

A reasonable bond graph model of the hydraulic pump is shown in Figure 11.27. It accounts for the transformation of mechanical into hydraulic power by means of the power conserving two-port transformer. Moreover, it takes into account the compressibility of the fluid in the outlet port by means of a C energy store attached to the 0-junction and accounts for internal leakage by means of the R element. If a signal input port is added and if the transformer is allowed to be a displacement modulated

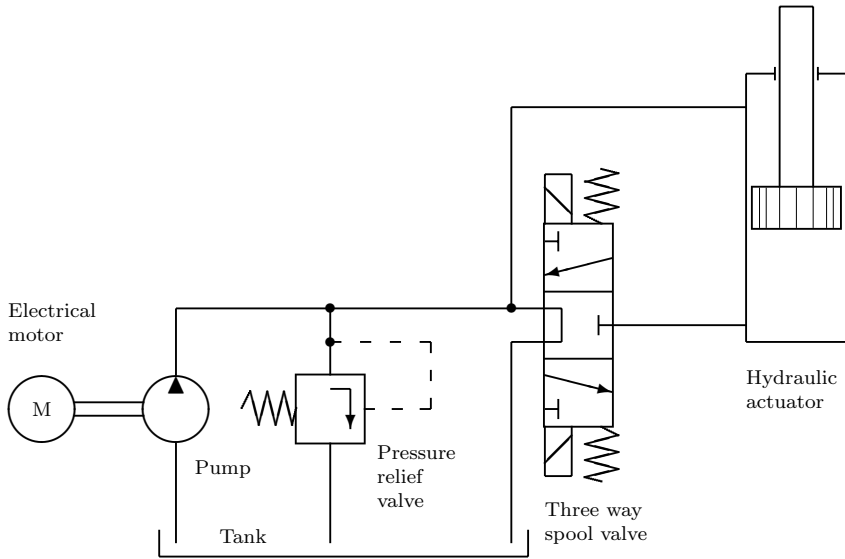


Fig. 11.26 Circuit schematic of a hydraulic drive (Borutzky, 2002)

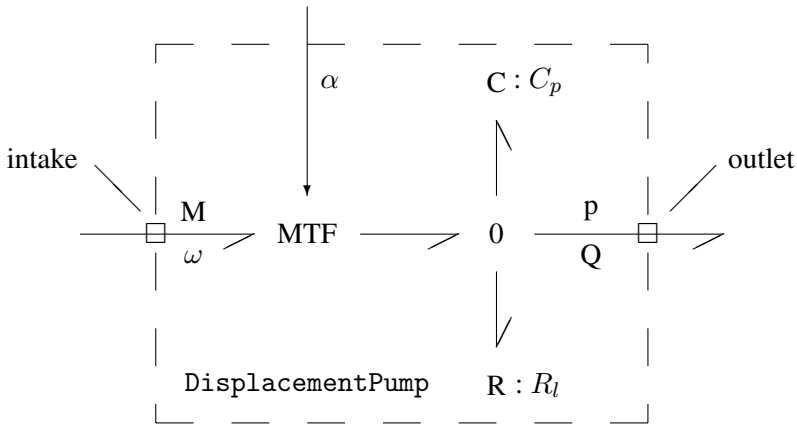


Fig. 11.27 Bond graph model of a (variable) displacement pump

transformer (MTF) as in Figure 11.27, then the model may even represent a variable displacement pump controlled by the angle, α , of inclination of the swashplate. Figure 11.28 shows the Modelica description of a model of the hydraulic resistor in Figure 11.27. It inherits the superclass `passiveOnePort`. A Modelica description of the pump model using the hydraulic resistor model in Figure 11.27 is given in Figure 11.29.

```

model Orifice "hydraulic orifice"
  extends passiveOnePort;
  constant Real rho (final unit="kg/m^3") = 0.85e+3 "fluid density";
  parameter Real alpha=0.61 "flow coefficient (turbulent flow conditions)";
  parameter Real area "cross section of orifice";
  Real Q "volume flow";
  Real dp "pressure difference across the orifice";
equation
  Q = p.f;
  dp = p.e;
  Q = alpha * area * sqrt(2/rho * abs(dp)) * sign(dp);
end Orifice;

```

Fig. 11.28 Modelica description of a hydraulic orifice

```

model DisplacementPump

/* This bond graph model has
- a mechanical power port labeled intake,
- a hydraulic power port labeled outlet which provides the volume flow Q,
- a signal input denoted by alpha controlling the angle of inclination of the swashplate
*/

  PowerPort intake, outlet ;
  input Real alpha;

/* Type and local name of submodels used */
  MTF transformer ;
  zero4P p ; // 0-junction represents the load pressure p
  C Cp ; // accounts for fluid compressibility in the outlet port
  orifice Rl ; // accounts for losses due to internal leakage

/* connectivity of power ports according the bond graph */
equation
  transformer.signalIn = alpha ;
  connect(intake, transformer.PowerIn) ;
  connect(transformer.PowerOut, p.port1) ;
  connect(p.port2, Cp.port1) ;
  connect(p.port3, outlet) ;
  connect(p.port4, Rl.port1) ;
end DisplacementPump ;

```

Fig. 11.29 Modelica description of a bond graph model of a variable displacement pump

Word Bond Graph and Modelica Description of the Hydraulic Drive

Now suppose that bond graph models have been developed for standard hydraulic components, described in Modelica and stored in a library. Moreover, assume that bond graph models described in Modelica are also available for electromechanical devices, e.g. electric motors, and that a bond graph model for the application specific mechanical load of the hydraulic drive has been composed of lower level bond graph models. Then, it is sufficient to translate the system schematic into a hierar-

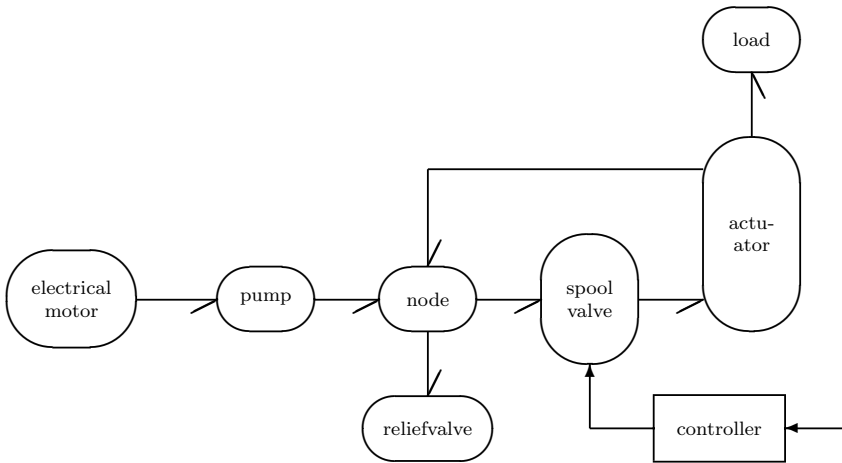


Fig. 11.30 Word bond graph of the controlled hydraulic drive

chical bond graph such that for each submodel, a Modelica description is available from a library. Figure 11.30 shows a *word bond graph* of the controlled hydraulic drive. The word bond graph closely corresponds to the physical structure of the system as depicted in the schematic. A model of the tank and all bonds corresponding to the return lines are missing in the word bond graph due to the fact that the return pressure is usually chosen as a reference. Consequently, the 0-junction representing the return pressure and all adjacent bonds have been eliminated. The signal into the controller is generated from signals provided by sensors which monitor the dynamics of the mechanical subsystem.

The word bond graph is not just another graphical representation, but a first step from a schematic representation towards a non-causal mathematical model. A word bond graph may be viewed as an intermediate format between a schematic that uses (standardised) domain specific icons and a mathematical model. A textual object-oriented description of such a word bond graph is straightforward and rather intuitive in a modelling language like Modelica. The result is given in Figure 11.31.

As in the Modelica description of the bond graph of the variable displacement pump, first, all submodels involved are listed. Each submodel is an instance of a model class given in the first column. The **equation** section describes the connectivity of the submodels such that for each power bond and low power signal connection between the controller and the spool valve, there is a corresponding **connect** statement. What still remains to be described are the sensors and the input of the controller. If all submodels are available from a library, then the equations of the overall system can already be automatically generated by a model processor.

The objective of this section has been to demonstrate how hierarchical bond graph models of multidisciplinary systems can be described in Modelica. Of course alternatively, a generalised network approach can be used. That this, mathemati-

```

model HydraulicDrive
// Submodels of all system components
  ElectricalMotor      electricalmotor;
  ConstantFlowPump    pump;
  ReliefValve          reliefvalve;
  ThreeWayValve        spoolvalve;
  DifferentialCylinder actuator;
  Zero4P               node;
  ControllerType       controller;
  MechLoad             load;
// Connectivity of ports according to the schematic
equation
  connect(electricalmotor.mech, pump.mech);
  connect(pump.hydr, node.port1);
  connect(node.port3, reliefvalve.port1);
  connect(node.port2, spoolvalve.P);
  connect(node.port4, actuator.B);
  connect(spoolvalve.A, actuator.A);
  connect(actuator.mech, load.port1);
  connect(controller.out, spoolvalve.control);
end HydraulicDrive;

```

Fig. 11.31 Modelica description of the word bond graph (Borutzky, 2002)

cal models of components are formulated in Modelica and submodel interfaces are plugged together according to the structure of the circuit schematic given in Figure 11.26. A Modelica description of hydraulic networks is supported by a library developed by Beater [6].

What is the Role of a Textual Modelling Language in the Light of Graphical Model Development?

This section closes with the observation that engineering systems are preferably developed at a graphical level. As a matter of fact, graphical editors being part of powerful graphical user interfaces (GUIs) together with various model libraries support the systematic development of hierarchical modularly structured engineering models and have largely replaced textual simulation languages. Thus, the question might arise as to what role powerful textual model description language play. Some answers to this question might be:

- An object-oriented modelling language like Modelica is an appropriate mean for building non-causal models for various libraries. Inheritance helps avoid inconsistencies and errors. Encapsulation can be used to encapsulate general properties, e.g. being a 1- or 2-port element, being passive, or being energetic. Such general properties can be inherited from a superclass and need not to be formulated again in a special component model allowing for a safe and elegant way of consistently defining bond graph models. Model libraries generally support the

reuse of model knowledge, which is essential to cope with the ever increasing complexity of systems to be designed.

- Languages like Modelica may serve as a uniform format that backs up several graphical formalisms.
- Independent of the graphical formalism in use, modellers will need a textual modelling language for introducing basic component models that are not yet available in one of the libraries and in particular to specify their nonlinear behaviour.

However, is there a need to describe the connectivity of the submodels of a submodel in an object-oriented modelling language if this can be simply expressed in an acausal bond graph being a generic format? Certainly, for composed models, bond graphs can serve as a core representation from which other representations, including time domain models as well as frequency domain models, can be derived. With respect to the exchange of models, it has to be taken into account that not all modelling and simulation programs that support multiple modelling formalisms also support bond graphs. By unifying features of several existing object-oriented modelling languages, Modelica has the potential of serving as an exchange format between various (proprietary) modelling and simulation software.

11.6 Software for Bond Graph Modelling

In the previous section, two modelling languages have been considered with respect to bond graph modelling. In this section, some representatives of software for bond graph modelling falling into three classes will be discussed. Since in many places in the world, software programs have been developed, are being further developed, or are presently under development that can be used directly or indirectly for bond graph modelling, it is neither possible to give a comprehensive survey of relevant programs, nor even to report in an exhaustive manner all features of one of these programs. Due to lack of space, the presentation is confined to some programs. Some others are just referenced. Samantaray has given a list of some bond graph software along with some brief personal assessments [70]. Some links to bond graph software can be found in the bond graph compendium assembled by Cellier [21]. The presentation in this section is not meant to be product oriented. The aim is rather to outline different approaches towards software support of bond graph modelling.

Following the historical evolvement of bond graph software, first, two quite different simulation programs are considered that have been used for a long time by many bond graph model developers.

11.6.1 ENPORT™

The simulation or modelling languages ACSL and Modelica are not bound to a particular modelling methodology. In contrast, while bond graphs devised by Paynter were elaborated into a modelling methodology by his former Ph.D. students Karnopp and Rosenberg, the latter developed the ENPORT® program with the aim of supporting the new bond graph methodology. According to the state-of-the-art at that time, ENPORT® was a simulation program. However, in contrast to CSSL based programs it did not require assignment statements, but accepted as input a non-causal description of the structure of a bond graph. To that end, the undirected edges of an acausal bond graph are enumerated and the incidence matrix of the graph is entered in alphanumeric form as a so-called *line code*.

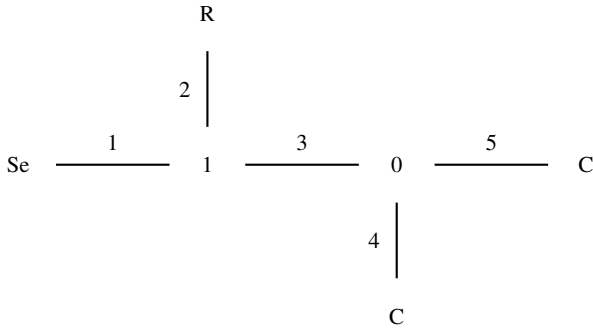
Figure 11.32 shows again the bond graph of the RC circuit in Figure 11.5 and, in addition, the corresponding ENPORT line code. As can be seen from Figure 11.32, the keyword **GRAPH** is followed by a list of fields separated by commas. The first entry in each field is a node of the graph followed by the numbers of its adjacent bonds [66]. This form of model input corresponds to the description of electronic circuits by means of an adjacency matrix. The famous and widely used circuit analysis program SPICE [58], for instance, accepts a so-called net list description of a circuit. Each line in the corresponding alphanumeric input file indicates which element is connected to which nodes in the graph. Of course, with the advent of powerful graphical user interfaces, an alphanumeric net list no longer is the primary input but rather an intermediate format. In addition to a mere description of the connectivity, the ENPORT input format enables one to specify reference directions of the energy flow. This kind of information is started by the keyword **POWER**. In the example, the line below the keyword indicates that bond number 3 is oriented from the 1-junction towards the 0-junction. In a similar way, causality information can be given. However, specification of energy flow reference directions and computational causalities is just an option. If not given, they are automatically assigned according to the rules of Karnopp and Rosenberg.

The information about the structure of a bond graph and the element parameters are enough for ENPORT (initially developed for linear models only) to build the matrices of in the linear state space equations in numerical form. In the case of a linear model, storage ports with derivative causality and causal paths between resistor ports, that is, dependent states and algebraic loops of auxiliary variables, are allowed. Due to the linearity of the junction structure equations and the constitutive equations of the elements, implicit relations can be solved and dependent states can be eliminated.

The starting point for automatically building a linear state space model is a partitioning of the vectors and the matrix in the matrix equation of the junction structure. Figure 11.33 shows the well known partitioning of a bond graph.

Let \mathbf{S} denote the junction structure matrix, then according to the above partitioning of a bond graph, the matrix equation of the junction structure reads

$$[\dot{\mathbf{X}}_i \mathbf{Z}_d \mathbf{D}_{in} \mathbf{V}]^t = \mathbf{S} \cdot [\dot{\mathbf{Z}}_i \mathbf{X}_d \mathbf{D}_{out} \mathbf{U}]^t. \quad (11.13)$$



GRAPH

SE 1, R 2, 1 1 2 3, 0 3 4 5, C 4, C 5.

POWER

3 1 0.

Fig. 11.32 ENPORT line code of a simple bond graph

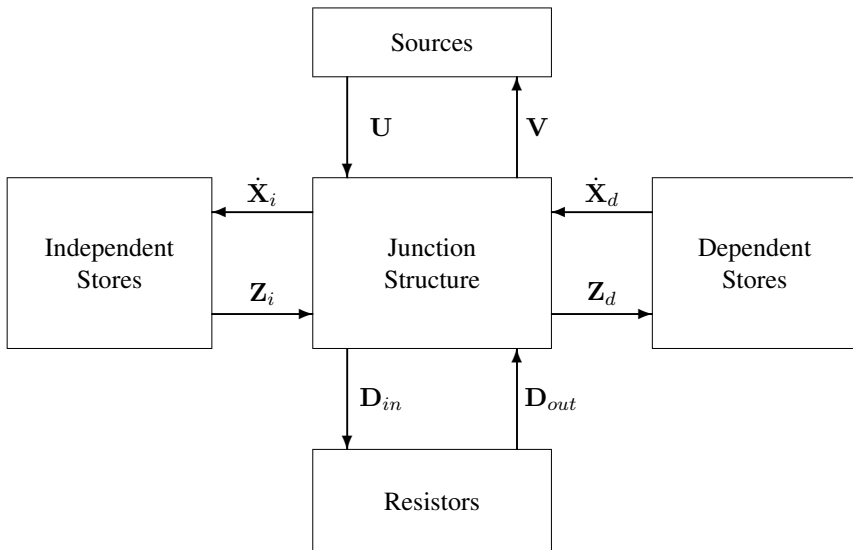


Fig. 11.33 Partitioning of bond graph, (Rosenberg, 1971)

Rosenberg describes this fundamental systematic approach of generating state space equations from a partitioned bond graph for linear models and its extension to non-linear models in detail in [64].

A well known and widely used version of the ENPORT program has been its version 4 [65]. Since then, the program has undergone many modifications. A more recent description of its capabilities in supporting the modelling process is given in [44].

11.6.2 TUTSIM™

In contrast to ENPORT, the purpose of the simulation program THTSIM, developed at the University of Twente, Netherlands, was to enable computation of nonlinear block diagrams on small computers called mini-computers at that time. This program became quite popular in academia and in known industry under the name TUTSIM™ [54]. It was the predecessor of 20-sim®. If every signal block has only one output, a block diagram can be described, writing for each block an equation of the form

$$\text{Output} = \text{Function}(\text{Input}_1, \dots, \text{Input}_n), \quad (11.14)$$

where each input to a signal block apparently must be the output of another signal block. The assignment statement of each block can be written in a short form if all signal block outputs are enumerated by positive numbers, if a negative block number means that the block's output is multiplied by minus one and if the function of each block is indicated by a predefined block type. The assignment statement of a block can then be coded in the form

```
<Block number> <Block type> <Input_1>, ..., <Input_n>.
```

A description of a block diagram in this form has been called a *structure table* in the context of the TUTSIM™ program. A list of lines of the above form captures the structure of a block diagram in alphanumeric form. At the same time, it is a list of assignment statements in a concise form since a block diagram, in essence, is a graphical representation of a system of simultaneous equations. Block diagrams generally do not reflect the physical structure of a system. Their structure can change significantly if the physical structure is slightly modified to account for additional effects.

It has already been demonstrated in Section 3.5 that assignment statements can be directly derived in a systematic and easy manner from a causal bond graph if all stores have got integral causality and if there are algebraic loops. For that purpose, a procedure has been given. According to this procedure, first, the equations of all external sources are written followed by the equations of all resistors. Finally, the constitutive equations of the stores are set up. By back propagation of causal paths, inputs into dependent sources and resistors can be expressed by system inputs and state variables. This is a variant of a procedure presented in as early as 1977 by van Dixhoorn [83]. The key observation of van Dixhoorn has been that such a procedure allows processing of causal bond graphs by block-oriented simulation programs like TUTSIM™ given that the underlying mathematical model is of state space form. Regarding 1- and 0-junctions, the summation of efforts or flows is

transferred into the element that causally dominates the junction. The output of this element is distributed by the junction to all adjacent bonds. As a result, junctions are eliminated.

Example: TUTSIM Structure Table of the Shunt Motor Bond Graph

For illustration of the method, the bond graph of the shunt motor given in Figure 11.1 has been reproduced in Figure 11.34. Instead of assigning variable names to some bonds, all bonds are enumerated as required in ENPORT line code.

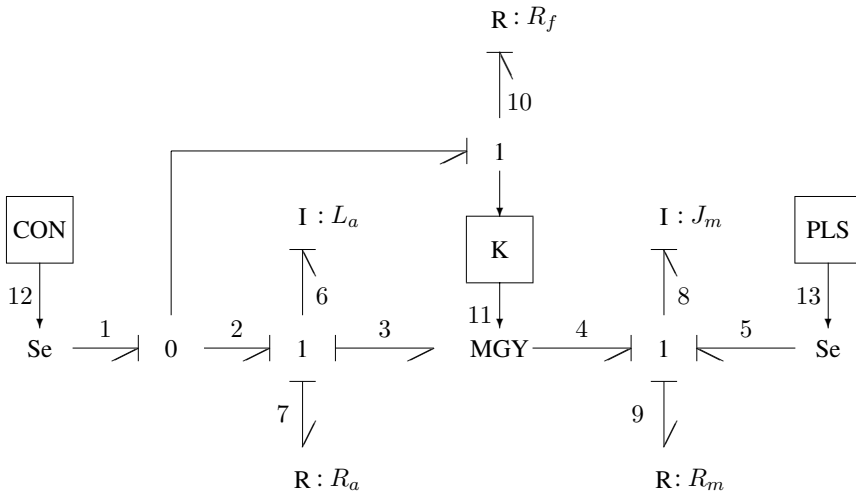


Fig. 11.34 Bond graph of the shunt motor model in Figure11.1 with enumerated bonds

1	SE	12
7	R	6
6	I	1 -7 -3
3	MGY	11 8
4	MGY	11 6
5	SE	13
9	R	8
8	I	4 5 -9
10	R	1
11	GAI	10
12	CON	
13	PLS	

Fig. 11.35 TUTSIM structure table of the causal bond graph in Figure11.34

```

8 I      4 5 -9
6 I      1 -7 -3
7 R      6
9 R      8
12 CON
13 PLS
1 SE    12
5 SE    13
10 R    1
11 GAI  10
3 MGY  11 8
4 MGY  11 6

```

Fig. 11.36 TUTSIM structure table of the causal bond graph in Figure 11.34 sorted into computational order

Figure 11.35 shows the corresponding TUTSIM structure table. The third line in this structure table means that the inputs into the block of type `I` are summed up. According to the constitutive equation of the `I` store, $I: L_a$, with integral causality, the sum of flow variables into the element is integrated and divided by the parameter of the element (cf. Figure 3.10). The output of the `I` element is the effort assigned to the integer 6. Since in TUTSIM blocks have got only one output, two signal blocks are necessary for representing the two-port gyrator (Figure 3.9). In the signal block of type `GAI` (Gain), the input signal is multiplied by a constant coefficient $K > 1$. The block of type `CON` provides a constant system input signal, while the block of type `PLS` provides a pulse.

The TUTSIM™ program checks each line of the structure table with respect to syntax errors immediately after it has been entered. After the input of a structure table has been completed, all blocks are sorted into a *computational order* in which the blocks are computed in each integration time step. Figure 11.36 shows the lines of the structure table sorted into computational order.

Obviously, since storage elements include an integrator which has got an initial value, computation can start with evaluating the outputs of the 1-port stores. The outputs of external sources depend only on time and can be evaluated when needed.

In case there are algebraic loops in the model, they can be broken up by inserting a so-called algebraic delay block, `ADL`, that introduces a delay of one time step. If stores with derivative causality cannot be avoided, then differentiation of variables with respect to time must be performed numerically. As many other simulation programs at that time, TUTSIM™ could only solve explicit state space models.

In contrast to early versions of the ENPORT® program, from the very beginning TUTSIM™ enabled computation of *nonlinear* bond graph models. For many years, it was the preferred simulation program, e.g. of Thoma. In [77], he gives TUTSIM structure tables for a number of bond graph models. However, the program does not support the development of hierarchical models and with regard to the numerical integration of ODEs, only the explicit second order Adams-Bashforth algorithm is available. That is, even if stores with derivative causality or algebraic loops are

removed by inserting parasitic elements into the original model, TUTSIM cannot be used if the resulting ODE system is stiff. As already mentioned, 20-sim[®] has emerged as its powerful successor that meets today's requirements with respect to the development of large complex models as well as with respect to the automatic generation of a mathematical model and its numerical computation.

11.6.3 Bond Graph Preprocessors

It is true that object-oriented modelling languages can be used directly for the development of hierarchical modular models. However, as has already been remarked, they are especially important for the development of component models made available to modellers in libraries. Beyond reliably implemented component models, a graphical user interface (GUI) is needed for building large hierarchical modular models at a graphical level. This can be supported in two different ways.

One option are so-called preprocessors that support model development at a graphical level and can automatically derive a textual model description from the graphical model representation in one of the simulation or modelling languages supported by the preprocessor. The advantage is that a modelling program of one company can be used to produce the input for a simulation program of another company. This way, capabilities of different software products can be exploited. For instance, the simulation program ACSL[®] does not have a graphical interface for bond graphs. However, there is a bond graph preprocessor that can derive equations from the bond graph and output them as assignment statements in ACSL input format.

An alternative are self-contained, integrated modelling and simulation software environments. They use a modelling language, e.g. SIDOPS or Modelica, for a textual intermediate format behind the graphical representation that can be used for various purposes. Such a textual description could, for instance, be converted into another format to interface with other software packages or could be converted into a programming language, e.g. C, or converted into an executable simulation program. In the latter case, obviously, a syntax check of the input processor of a target simulation program is not needed. As an example of an integrated modelling and simulation particularly suited for bond graph modelling, we will consider some features of the 20-sim program.

CAMP-G[®]

One of the first bond graph preprocessors has been the software program CAMP developed by Granda in the framework of his Ph.D. project at the University of California at Davis [43]. Early versions of CAMP required bond graphs to be entered as a line code as in ENPORT[®]. Unlike ENPORT, the preprocessor CAMP sets up the equations for each bond graph node and can output them as an ACSL[®] input file. In the course of further development, a graphical user interface was added so

that bond graphs can be entered graphically. Since that time, the program's name is CAMP-G[®].

As is customary for many graphical editors, the nodes of the graph can be picked, placed on a workspace and connected using a rubber band technique. Moreover, selected parts of the graph can be moved around. When a connection of two ports is drawn, CAMP-G[®] automatically adds the power reference direction and a causal stroke to the bond. If the connection to the new element affects causalities so far assigned to a bond graph under construction, then this is taken into account and causalities are automatically adapted where necessary. That is, rules for power reference directions as well as causality assignment and propagation are implemented in CAMP-G[®]. Consequently, the program can check for causal conflicts, derivative causality at storage ports and for algebraic loops. Bonds are highlighted in red as long as their power reference direction and their causal orientation is still undetermined in an uncompleted bond graph. Derivative causalities at storage ports are also highlighted.

A further development of CAMP-G[®] can output an implicit set of equations in symbolic form for bond graphs with derivative causalities. As it interfaces with MATLAB[®]/Simulink[®], the capabilities of the MATLAB[®] Symbolic Math Toolbox[™] can be exploited in order to eliminate dependent state variables and to produce the matrices of an explicit linear state space equation for linear bond graphs with derivative causalities [41, 42].

For nonlinear elements, the generated ACSL input file must be directly edited for specifying their constitutive equations. The CAMP-G user interface does not provide an equations editor that knows about the syntax of a modelling language and highlights its keywords. Furthermore, if the overall model is a combination of a bond graph with a block diagram fragment, then the equations of the latter must also be added to the generated ACSL file. Alternatively, suppose a linear bond graph model has been developed for an engineering system which is the plant in a control loop, then CAMP-G[®]/MATLAB[®] can generate the matrices of a multiple input- multiple output block (MIMO system) to be used in a block diagram of the engineering system and its closed loop control that can be built in the block diagram editor of Simulink[®]. This way of using a bond graph model as a signal block in a Simulink[®] block diagram is indicated by Figure 11.37. That is, there is no need to transform a linear bond graph into a block diagram for use in a block-oriented simulation program.

On the other hand, CAMP-G[®] neither supports the development of hierarchical bond graph models nor does it allow for a multibond graph representation, e.g. of multibody systems. Furthermore, the model description in ACSL is not optimised in the sense that the DERIVATIVE section should include a minimum of equations because these equations must be reevaluated in each integration time step. Redundant equations of the form `variable_1 = variable_2` due to 0- and 1-junctions are not eliminated, nor are the equations sorted. That is, CAMP-G[®] depends on the capabilities of software packages it interfaces with, such as the ACSL[®] simulation program and MATLAB[®]/Simulink[®].

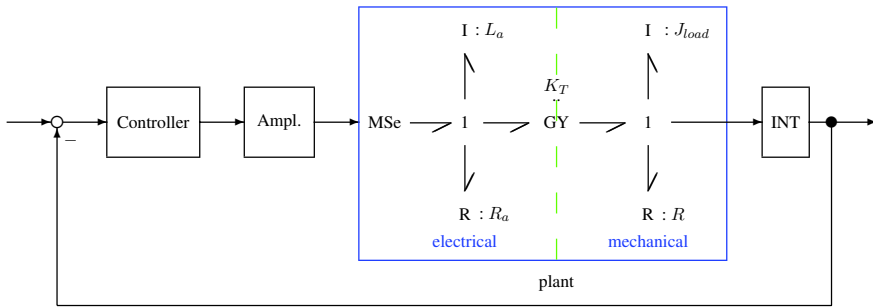


Fig. 11.37 Bond graph model as a signal block in a Simulink block diagram

BAPS[®]

Another program with a graphical front end that enables the development of hierarchical modular bond graph models at a graphical level and that can output a textual description in various simulation languages is BAPS[®], developed by Ruzicka at the Technical University of Vienna [68] in the framework of his Ph.D. project. Bond graphs created by means of the graphical front end BAPSDRAW[®] are first converted into a textual format. At this level, the structure of a bond graph (sub)model is given as a line code. Moreover, nonlinear constitutive equations as well as nonlinear relations between parameters can be specified in the language BAPS. The program BAPS[®] then performs causality assignment checks for algebraic loops and generates an optimised simulation model in ACSL or in one of the other supported simulation languages.

ARCHER

Finally, we shall mention the modelling and analysis program ARCHER [4] developed at l'École Centrale de Lille, France. Regarding its purpose and its capabilities, it is more than a bond graph preprocessor for a simulation program. It is mentioned here because it is a software that supports bond modelling at a graphical level, but does not include simulation routines. It rather enables the generation of transfer functions in symbolic form directly from linear bond graphs and can set up state space models in a symbolic form.

A remarkable feature of ARCHER is that it enables the analysis of models with respect to structural controllability and structural observability. In ARCHER, some essential research results of the bond graph modelling group at l'École Centrale de Lille have been implemented.

11.6.4 Bond Graph Toolboxes

If a software program does not understand bond graphs, then an obvious step is to develop an appropriate software tool that bridges the gap between the modelling method and the program. This approach apparently is not confined to simulation programs. For instance, a large number of so-called toolboxes for various application areas have been developed that enhance the features of the basic mathematics software package MATLAB® [8]. These powerful toolboxes, along with the easy interactive use of the software and its support in visualisation of results, have essentially contributed to the wide acceptance and use of this program in academia and industry. Since then, the simulator Simulink® has been added to MATLAB®; the combination of both software allows the user to analyse and simulate dynamic systems in various engineering domains.

BondLab

In view of the immense popularity of MATLAB®/Simulink®, it is no surprise that at ESAT of the Catholic University of Leuven, Belgium, Minten, Vranckx, De Moor and Vanderwalle developed a MATLAB® toolbox called BondLab [55] that aims to support the bond graph-based design of mechatronic systems at a graphical level. It can generate linear and nonlinear state space models in symbolic form and output them as a MATLAB *.m file. Also, transfer functions can be derived in symbolic form. The graphical user interface allows direct access to simulation and visualisation facilities. Animation of energy flows on the bond graph is also available.

Bond Graph Toolbox

A similar approach is to provide software tools for graphical entry of bond graphs, for causality assignment, for equation generation in a toolbox along with some auxiliary routines for graphical representation of simulation results and to exploit the comprehensive symbolic and numerical features of classical algebra systems, e.g. Mathematica®³ [89], for an interactive symbolic analysis of bond graph-based models and for reduction of symbolic equations to state space form. One of the first modelling environment of this kind has been the *Bond Graph Toolbox* developed by Nolan at the University College of Galway, Ireland [59]. In a Mathematica session, this toolbox allows the user to interactively generate the state matrix as well as transfer functions in symbolic form for linear systems and to analyse them, e.g. with respect to stability. In the case of nonlinear systems, the equations derived from the bond graph can be reduced to state space form given the mathematical model corresponding to the causal bond graph is of that form. For simulation, it can be output as a FORTRAN routine. The latter one can be linked with a library routine for numerical integration of ODEs. For an object-oriented modelling approach, it

³ Mathematica® is a registered trademark of Wolfram Research, Inc., 100 Trade Center Drive, Champaign, IL 61820-7237, USA, <http://www.wolfram.com>

is essential that model equations are non-causal. Consequently, processing of a textual model description in a modelling language, e.g. SIDOPS, Dymola or Modelica, also requires symbolic reformulation of equations. However, in the model processor for one of these languages, this is done automatically, while an algebra system like Mathematica[®] allows for an interactive analysis and reformulation of equations.

The Mathematica Bond Graph Toolbox

A bond graph tool box for Mathematica has been developed by N. Venuti [84]. This extension to Mathematica enables the user to graphically enter small bond graphs in the Mathematica symbolic environment. For creating and editing a bond graph, a floating palette is available. Relying on the power of Mathematica, this toolbox offers some advanced features as multiport stores, support of multibonds, symbolic transformation, analysis with respect to structural observability, removal of algebraic loops in linear systems, and handling of derivative causality.

Inspired by the work of Nolan, another toolbox called MTT has been developed by Gawthrop [37] at the University of Glasgow, Scotland. The underlying philosophy of this toolbox will be discussed in Section 11.6.6.

11.6.5 Integrated Modelling and Simulation Environments

An alternative to bond graph preprocessors for simulation programs and mathematics packages is to integrate appropriate support of modelling, simulation and animation in a self-contained software environment. In this subsection, the software packages 20-sim[®], SYMBOLS Shakti^{TM4} and MS1^{®5} shall be considered as state-of-the-art examples of such integrated software environments.

Another powerful integrated modelling and simulation environment introduced in 2003 by Damić and Montgomery [26] is the software package BondSim[®]. It supports systematic hierarchical bond graph modelling and simulation of mechatronic systems. A special component port concept enables component models to be plugged together according to the structure in which the components of a real system are interconnected. Aiming at a separation of model development and model processing, BondSim[®] relinquishes the concept of computational causality. Mathematical models are automatically generated in the form of DAE systems and are numerically computed by means of a sophisticated solver. To that end, integrated symbolic computer algebra methods are used for the generation of Jacobian matrices in symbolic form.

Furthermore, in 2002, Raczynski presented the modelling and simulation software system PASION[®] for continuous, discrete and hybrid processes [63]. This

⁴ SYMBOLS ShaktiTM is a trademark of HighTech Consultants, STEP, I.I.T. Kharagpur - 721 302, India, <http://www.htcinfo.com>

⁵ MS1 is a registered trademark of Lorenz Simulation SA, Rue Jacob-Macoy, 89, B-4000 Liège, Belgium, <http://www.lorsim.be>

software system makes use of an object-oriented, Pascal related simulation language that is translated to Pascal and includes a module for graphical input of bond graph models, automatic causality assignment and equations generation.

20-sim[®]

The modelling and simulation software environment 20-sim[®] has emerged from the Ph.D. project of Broenink [18] at the University of Twente, Netherlands. Based on a completely new design, it has become a powerful next generation successor of the well known block-oriented TUTSIM[™] simulation program also developed at Twente University. Tested, improved and enhanced by a number of new features over two decades, 20-sim[®] has evolved into a reliable, mature and intuitive to use modelling and simulation software environment particularly suited for bond graph modelling of mechatronic systems. The software has been presented, e.g. in [20]. For the latest features, see the home page of the program [25].

With regard to modelling, 20-sim[®] supports the development of hierarchical models and multiple formalisms for model representation. That is, for representation of models or submodels, either iconic diagrams, bond graphs, block diagrams, equations, or a combination of these formalisms can be used.

According to the philosophy of separating the interface definition of a submodel from its body, the 20-sim graphical user interface provides a window for specification of ports and parameters of a submodel and a window for entering its implementation either as a graph or as a textual description. The latter case is supported by a SIDOPS language sensitive editor. Moreover, an icon representing the submodel can be designed by means of a drawing editor. The hierarchy of submodels is displayed as a tree-like structure in another window. By clicking on the submodel icons in this hierarchy tree, the user can switch between the implementations of different submodels. Figure 11.38 shows a screen shot of the 20-sim 4.0 editor.

The large implementation window in the middle shows the bond graph of the shunt motor model previously described in ACSL (cf. Figure 11.1). The bottom left window includes the Interface tab and the Icon tab. In Figure 11.38, the Interface tab has been selected. It shows that the selected (sub-)model, in this case, the MGY element, has got two power ports p_1 and p_2 and a signal port r . Double clicking on one of the ports p_1 , p_2 , or r opens the Interface Editor and enables one to modify the port's attributes (Figure 11.39).

Double clicking on the icon MGY on the Icon tab opens the Icon editor. The latter enables one to design an icon for a submodel. In addition, the Icon editor provides icons that are standard in various engineering disciplines, e.g. for components in electrical or hydraulic networks.

The long tall window on the left side comprises the Model tab and the Library tab. The Model tab provides a survey of the model hierarchy and enables one to browse through this hierarchy. In the case of the shunt motor example model, there is only one level of hierarchy. Clicking on one of the items in the list of elements results in a display of its equations in the large middle window. The Library tab enables one to browse the comprehensive 20-sim model library organised as part of the Windows

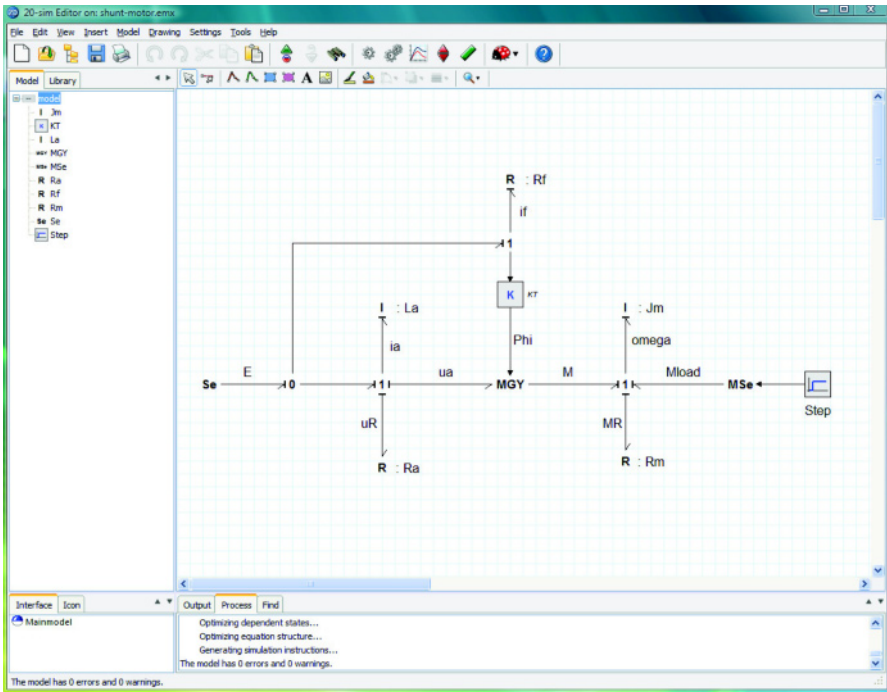


Fig. 11.38 Screen shot of the 20-sim 4.0 editor

file system. This library provides models for many standard components in various engineering disciplines, e.g. electrical, mechanical, hydraulic, thermal and control engineering.

The bottom window displays messages. In case of the shunt motor example, the item “Check complete model” in the pull down menu “model” has been clicked. By consequence, 20-sim® has checked whether the model is correct with regard to the rules of the chosen modelling methodology and has performed some optimisation steps. The results of these steps are communicated to the model developer in the Process tab of the bottom window. In this case, no error messages nor warnings have been produced.

Like the program BONDYIN [34, 85], designed for bond graph modelling of multibody systems, 20-sim® supports multibond graphs at a graphical level. In SIDOPS, functions on vectors and matrices are available, allowing a formulation that is close to usual mathematical notation. Recall that equations in a section introduced by the keyword **equation** are not assignment statements. If equations are meant to be assignment statements, the symbol ‘:=’ must be used.

Development of hierarchical models at a graphical level means that submodels are taken from one of the model libraries and are used in a higher level submodel or must be build if not available in a library.

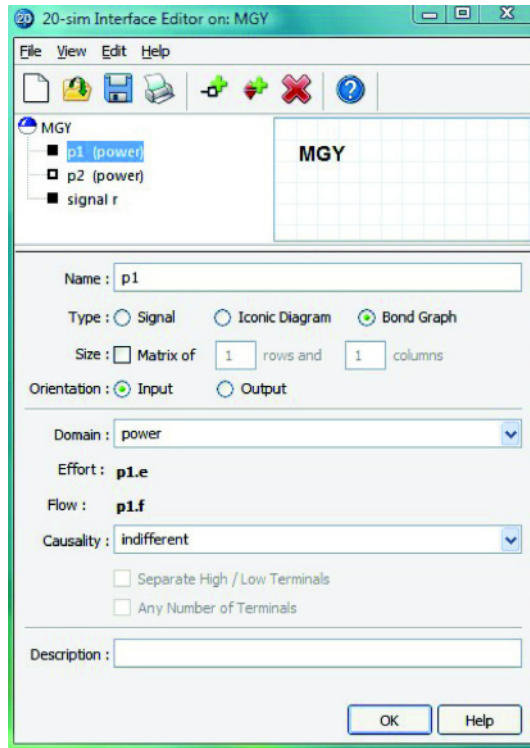
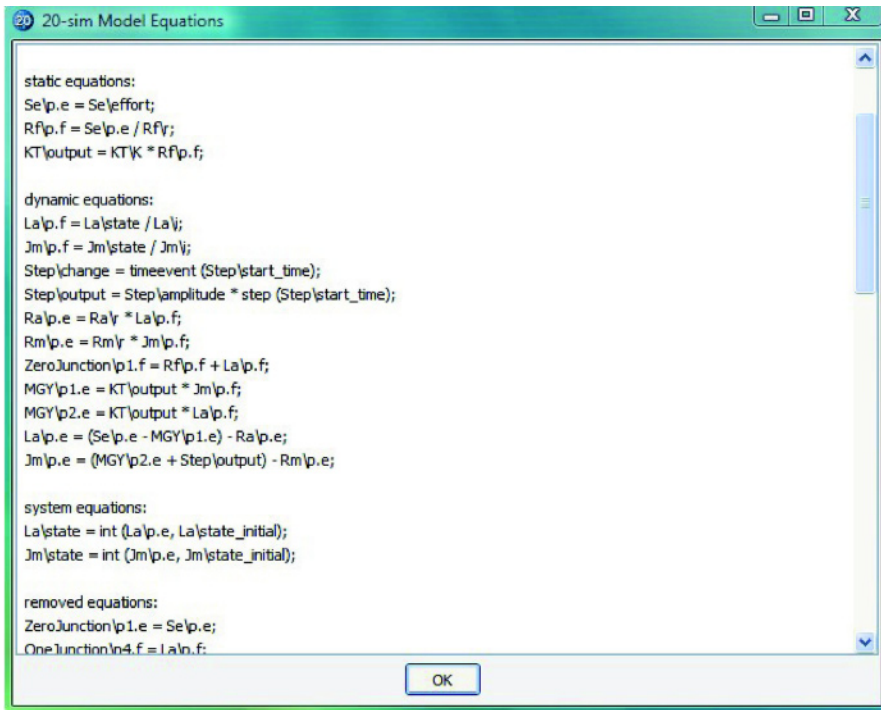


Fig. 11.39 20-sim 4.0 Interface editor

In the development of a bond graph model, power reference directions and causalities are automatically assigned as in CAMP-G[®] whenever two ports are connected. If causalities already assigned are affected by a new port to port connection, then causalities are adapted where necessary. Derivative causalities are highlighted in red. Causalities can also be set or changed by the user. Since 20-sim[®] has symbolic manipulation capabilities and can solve algebraic loops numerically by iteration and includes an implementation of the BDF method for numerical integration of DAE systems, the default is that in cases of dependent states or algebraic loops, only a warning is issued. Normally, 20-sim[®] will silently try to solve algebraic loops symbolically and will try to solve differential equations for dependent states. Moreover, redundant equations, as they are introduced by 0- and 1-junctions and by connecting ports of different submodels, are eliminated.

Finally, the structure of the set of equations is optimised. In order to speed up the simulation, all equations that do not affect the rates of state variables are taken out of the section of differential and algebraic equations to be evaluated in a loop during an integration step. Clearly, this set of dynamic equations should be as small as possible. Once this model processing has been completed, the generated set of equations can be output in symbolic form for inspection or for use in another pro-



```

20-sim Model Equations
static equations:
Se\p.e = Se\effort;
Rf\p.f = Se\p.e / Rf\;
KT\output = KT\K * Rf\p.f;

dynamic equations:
La\p.f = La\state / La\;
Jm\p.f = Jm\state / Jm\;
Step\change = timeevent (Step\start_time);
Step\output = Step\amplitude * step (Step\start_time);
Ra\p.e = Ra\ * La\p.f;
Rm\p.e = Rm\ * Jm\p.f;
ZeroJunction\p1.f = Rf\p.f + La\p.f;
MGY\p1.e = KT\output * Jm\p.f;
MGY\p2.e = KT\output * La\p.f;
La\p.e = (Se\p.e - MGY\p1.e) - Ra\p.e;
Jm\p.e = (MGY\p2.e + Step\output) - Rm\p.e;

system equations:
La\state = int (La\p.e, La\state_initial);
Jm\state = int (Jm\p.e, Jm\state_initial);

removed equations:
ZeroJunction\p1.e = Se\p.e;
OneJunction\p4.f = La\p.f;
OK

```

Fig. 11.40 20-sim 4.0 popup window with equations derived from the bond graph of the shunt motor model

gram. Figure 11.40 shows a 20-sim 4.0 popup window with equations derived from the bond graph of the shunt motor model. As can be clearly seen, the set of equations is structured into sections.

Furthermore, models or submodels can be converted into ANSI C-code by means of a built-in C-code generator for use in other programs. For instance, for a submodel selected in a bond graph, C-code can be generated for a Simulink® S-function as indicated by Figure 11.41.

Finally, a built-in compiler can generate executable simulation code. That is, 20-sim® is a self-contained modelling and simulation environment. The availability of a compiler or further software for post-processing of results is not required. Since model entry, model processing and simulation code generation are integrated into one software environment, it is possible to perform some error checking already during model development. Moreover, users can easily switch between model modification and model processing. The possibility of connecting submodels at a graphical level according to the physical structure of a system, the support of creating hierarchial models, automatic causality assignment and generation of a simulation model enables the reliable development of engineering system models in industrial projects. Furthermore, since the simulator is integrated into this advanced software

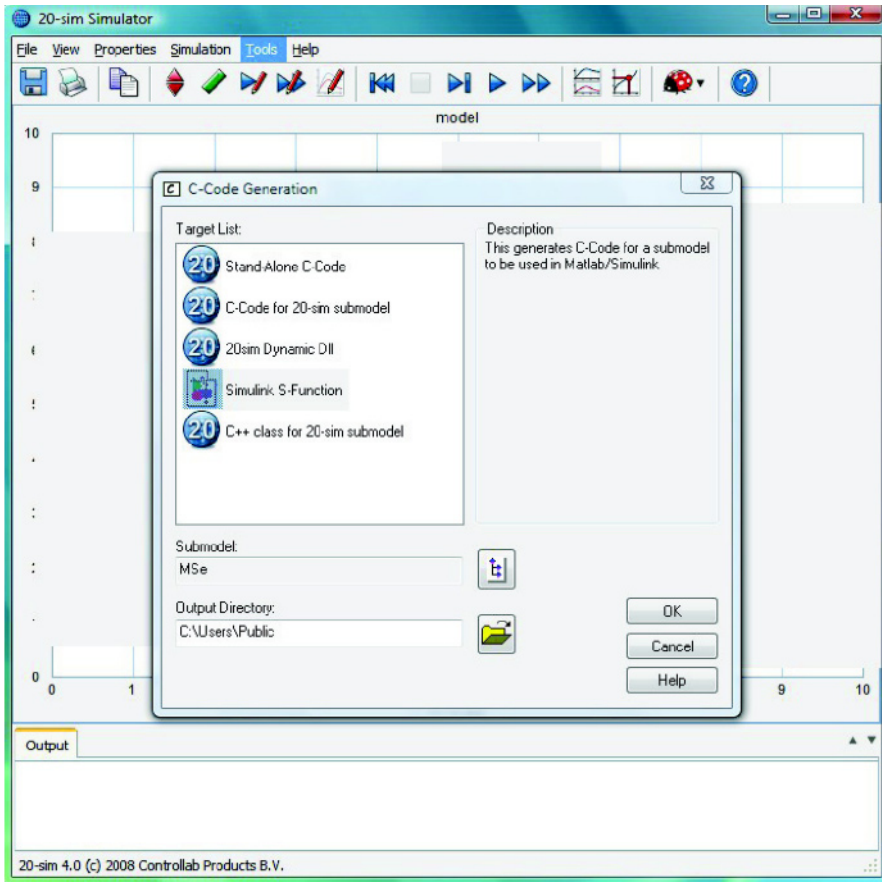


Fig. 11.41 20-sim 4.0 Realtime Toolbox window for generating C-code

environment, it is easy to switch between experiments on the model and modification of a model.

20-sim has been used for bond graph modelling and simulation in a number of books, e.g. [78, 79, 86]. In Chapter 12, 20-sim 3.2 has been used for most of the case studies.

SYMBOLS Shakti™

Another major state-of-the-art integrated modelling and simulation environment is SYMBOLS Shakti™ [24] (The abbreviation stands for *System Modeling by BOND-graph Language and Simulation*). This suite of software modules has been developed by A. Mukherjee and A. Samantaray and has emerged from more than two decades of research and teaching at the Indian Institute of Technology in Karagpur.

SYMBOLS Shakti™ and 20-sim® share many features even if both software environments differ with respect to some conceptual aspects and with respect to implementation details. SYMBOLS also supports hierarchical modelling of multi-energy domain systems and allows for several representation formalisms that can be mixed. The software comes with a large number of submodels for various engineering domains. The developers of SYMBOLS call them capsules. We will come back to this notion below. Submodels can also be connected according to the structure of the engineering system to be modelled. Bond graph elements can be picked from an object window left from the drawing area of the graphical editor. Ports are connected just by straight lines. That is, bond graph modelling in SYMBOLS starts with *undirected* bond graphs. By clicking the corresponding three buttons, bonds in the graph are numbered, assigned power reference directions and causal strokes. After an integrity check of the bond graph, equations can be derived in a reduced form (state space form) or in an optimised form that makes use of auxiliary variables. The result is displayed in the equations window below the drawing area.

Algebraic loops and derivative causality at storage ports are tolerated. Algebraic loops lead to a warning when the software is required to derive equations. The user is requested to use one of two methods provided for dealing with algebraic loops. In the case of derivative causalities, the integrity check of the bond graph issues a warning and asks the user to specify which element parameters are constant because this information is used to reduce the complexity of equations by setting derivatives of selected terms to zero.

Like in CAMP-G element, parameters are not assumed constant. After equations have been derived from a causally augmented bond graph, nonlinear relations between parameters as well as nonlinear constitutive element equations can be specified in C++ notation in an expression window below the drawing area. That is, unlike in 20-sim, transformers and gyrators do not have a signal port. In fact, they are conceived as two-port elements. The time varying transformer modulus or the gyrator ratio can be displayed as an annotation. Likewise, nonlinear sources are not represented as modulated sources. The result is a convenient concise bond graph representation that is not distracted by block diagram fragments. On the other hand, algebraic relations can be constructed in the expression window that are not explicitly depicted in a combined bond graph - block diagram representation. A dependency check may help to maintain consistency in the set of expressions by removing all unused variables in a scope and by sorting variables.

For illustration, Figure 11.42 shows a screen shot of the SYMBOLS editor Bondpad with a bond graph of the shunt motor model. The little pink dot on bond number 11 denotes a detector which has been configured to provide the flow f_{11} (as a global variable). When the software is commanded to derive equations from the bond graph in reduced form, it also expresses this non-state variable by means of inputs and states in addition to the state equations. In this simple example, f_{11} only depends on the supply voltage, that is, $f_{11} = 1/R_{11} * SE1$. Now, the modeller can use this signal variable for modulating the ratio of the gyrator by adding the equation $r = K * f_{11}$ in the expressions window (The sun symbol left to variable r means that it is a global variable).


```

[BASICSTATES=2]
P6
P10
[BASICSTATES]
[USERSTATES=1]
f11
[USERSTATES]
[STATEEQUATIONS]
SE1 = 220; // Volt
Mload=1;
SE8 = Mload*step(t,5,500);
M10=0.8; // mechanical inertia
R9=0.066; // mechanical friction
R5=0.875; // resistance of armature winding
M6=0.175;
R11=5.4945; // resistance of field winding
K=0.0307; // motor torque constant
r=K*U[0];
dY[0]=SE1-R5*Y[0]/M6-r*Y[1]/M10;
dY[1]=SE8-R9*Y[1]/M10+r*Y[0]/M6;
U[0]=1/R11*SE1;
[STATEEQUATIONS]
[USEREQUATIONS]
SE1 = 220; // Volt
Mload=1;
SE8 = Mload*step(t,5,500);
M10=0.8; // mechanical inertia
R9=0.066; // mechanical friction
R5=0.875; // resistance of armature winding
M6=0.175;
R11=5.4945; // resistance of field winding
K=0.0307; // motor torque constant
r=K*U[0];
U[0]=1/R11*SE1;
[USEREQUATIONS]

```

Fig. 11.43 Part of the SYMBOLS simulation code derived from the bond graph of the shunt motor model

by power reference directions and causal strokes. Ports, called *glue ports*, constitute a *causal* interface to the outside world. According to two possible power flow orientations and two possible causal assignments, the SYMBOLS capsule concept distinguishes four different types of glue ports. They are denoted effort input glue ports, flow input glue ports, effort output glue ports and flow output glue port. With regard to causality, an effort input glue port apparently is equivalent to a flow output glue port. However, while the power flow reference direction is oriented towards a flow output glue port, it points away from an effort input glue port.

Non-causal modelling now is supported by *generic* capsules. These are sets of equivalent capsules. According to causality requirements partner elements or capsules impose on the glue ports of a capsule, the proper capsule from the group of equivalent capsules is loaded that fits the causality requirements at its glue ports. Moreover, members of a generic capsule can have different numbers of states, equations and parameters as long as they have an equal number of external glue ports.

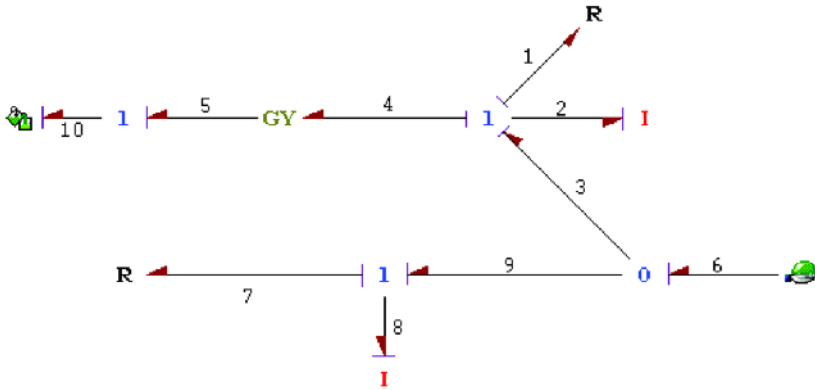


Fig. 11.44 SYMBOLS library capsule of a DC shunt motor

The loading of a proper member of a generic capsule is transparent to the model developer. Obviously, this approach of enabling submodels to be connected together according to the structure of their corresponding real system components' connections may require quite a number of hidden library capsules. However, especially for nonlinear components, this approach can be more practical than the reformulation of constitutive equations according to imposed causality requirements. In addition, it offers the flexibility of using different graph structures and different states, provided all capsules have the same number of external glue ports.

Example: SYMBOLS Library Capsule of a DC Shunt Motor

As an example, Figure 11.44 shows the capsule of a DC shunt motor that comes with the SYMBOLS library for the electrical domain. The DC shunt motor capsule in Figure 11.44 accounts for the inductance of the field winding (bond 8). The glue port at bond 6 is an effort input glue port and allows connection to a voltage supply. The port at bond 10 is an effort output glue port. It provides the motor moment acting on the mechanical load. The transparent mechanism of selecting the proper member of a generic capsule according to causal boundary conditions of a submodel is not confined to capsules in libraries that come with the SYMBOLS software. It also extends to user-defined generic capsules.

Structural Analysis with SYMBOLS

Another feature among the many others of this advanced software supporting hierarchical non-causal modelling of hybrid systems, simulation and control analysis is the capability for structural analysis. This facility, being an essential feature of

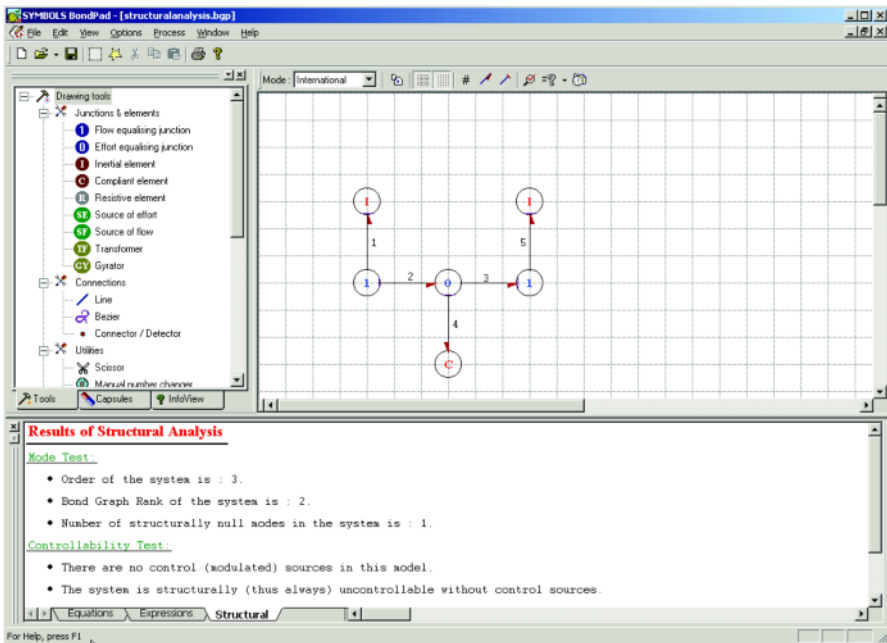


Fig. 11.45 System with one null mode

the program ARCHER, has also been implemented in SYMBOLS. Figure 11.45 shows the bond graph of the small example of two oscillating masses connected by a spring previously considered in Section 3.6. The result of the structural analysis is partly shown in the window below the drawing area. It confirms what has been demonstrated in Section 3.6. Figure 11.46 shows the result of the structural analysis of another example that has been considered in Section 6.4 (cf. Figure 6.16).

Furthermore, the software program SYMBOLS™ includes a powerful control analysis module that enables the derivation of transfer functions in symbolic form from a bond graph. Recently, a module has been added to SYMBOLS so that the program does not only support the development of bond graph models and the design of a control, but also bond graph model-based fault detection and isolation [15].

Some features of SYMBOLS are presented and illustrated by means of examples in the appendix of [57].

MSI®

We conclude this section on integrated modelling and simulation environments by briefly considering essential features of the modelling and simulation software MSI® [49, 51] developed by F. Lorenz, Liège, Belgium, in cooperation with EDF (Electricité de France) and with the support of INSA de Lyon (Institut National des Sciences Appliquées de Lyon).

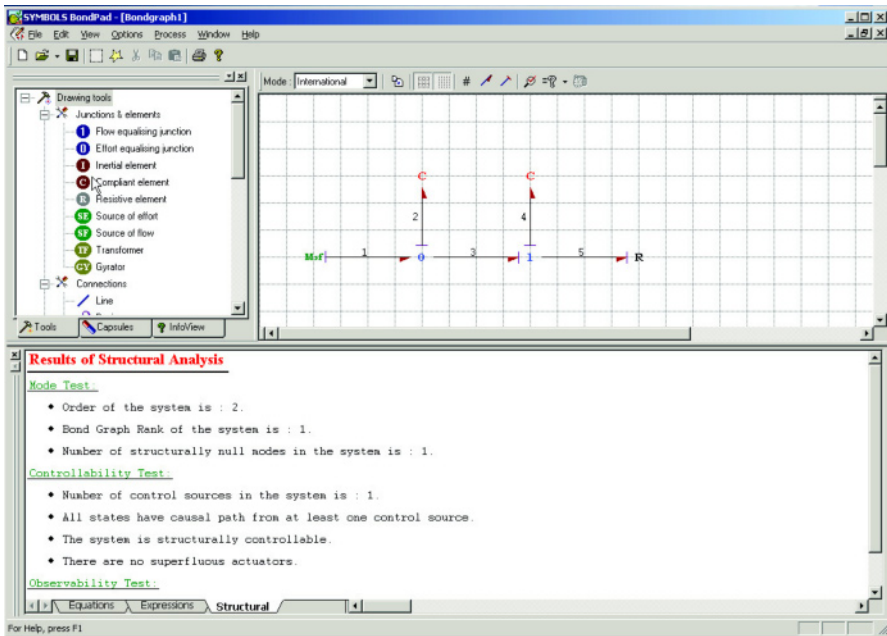


Fig. 11.46 Structurally controllable system

- **Multiformalism**
For developing models of system components, several graphical description formalisms can be used. MS1 supports bond graphs, block diagrams, linear graphs and equation models. On the textual level, model description in the Neutral Model Format, NMF [69] is supported by a language sensitive editor. Submodels can be described directly on the algorithmic level either in FORTRAN or C. However, FORTRAN or C code is not parsed, but just passed to be compiled and linked with a chosen solver.
- **Hierarchical Models**
Component models described in different forms can be connected graphically in order to build an overall system model. The different description forms are automatically transformed into an internal format.
- **Causality Assignment**
If a bond graph model of a component has been completed, computational causalities are automatically assigned. Moreover, by clicking on the corresponding buttons, the user can have the software check for topological loops and signal paths between input and and output variables.
- **Topological Loops**
A feature of MS1[®] to be pointed out is that these topological loops and signal paths can be highlighted on the bond graph (cf. Figure 11.47).

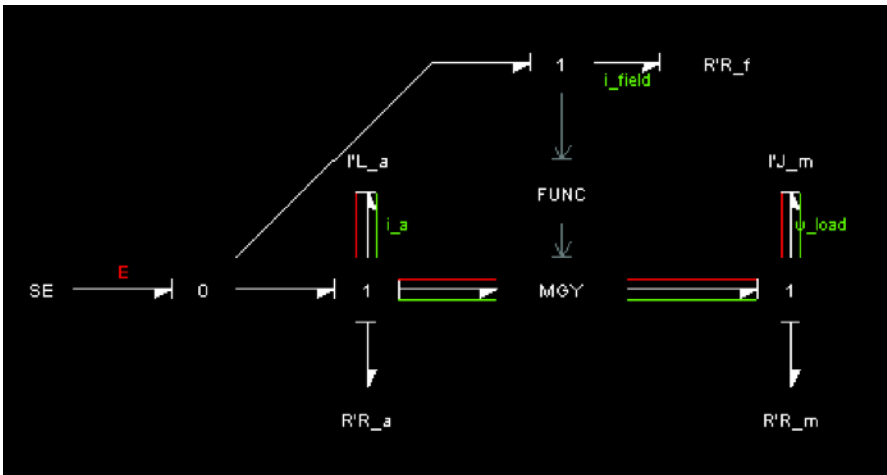


Fig. 11.47 Highlighting of the 2nd order topological loop between the two stores in the bond graph of the shunt motor model

Once all topological loops and signal paths between input and output variables have been determined, their corresponding gains can be used in Mason’s rule in order to derive transfer functions directly from a causal bond graph (cf. Section 6.2). Surprisingly, the user cannot have MS1 (version 5.2c) provide the transfer function between a selected output and a selected input variable.

Figure 11.48 shows one of the three loops in the bond graph of the DC motor model considered in Section 6.2 (cf. Figure 6.8). By clicking on the corresponding button the next loop is displayed.

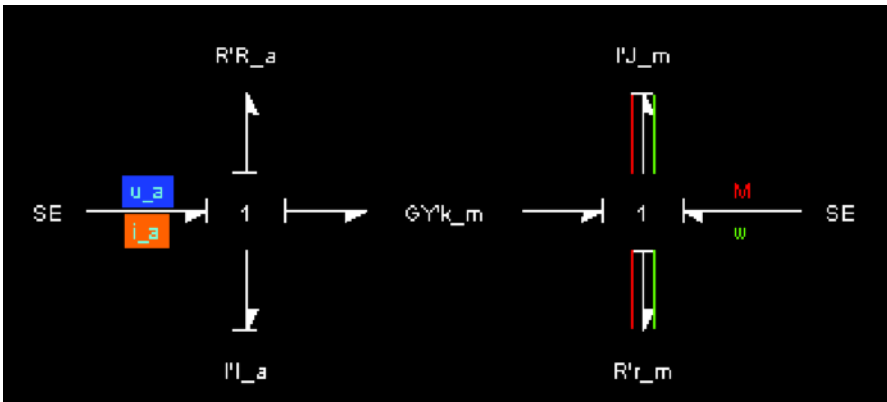


Fig. 11.48 Highlighting of one of the three topological loops in the bond graph of the DC motor model (cf. Figure 6.8)

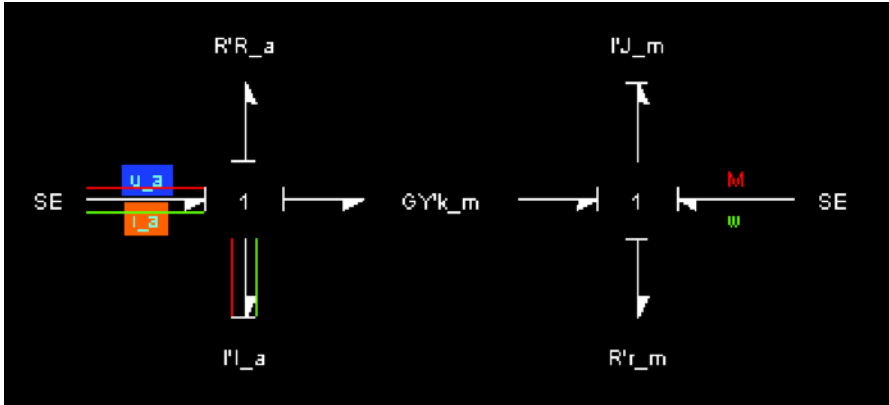


Fig. 11.49 Highlighting of the topological path between input and output variable (cf. Figure 6.8)

Figure 11.49 depicts the topological path between the input variable, u_a , and the output variable, the current i_a , in accordance with Figure 6.8).

- Bicausal Bond Graphs

Furthermore, MS1[®] supports the concept of bicausality.

- Equations generation

Equations derived from a causal bond graph can be output in the input language of one of the commercial solvers ACSL[®], Maple[™]⁶ and MATLAB[®], or in C code as a S-function to be used by Simulink[®]. In addition, MS1 offers the option to generate a Modelica description of a model so that it can be further processed, e.g. in the OpenModelica software [62].

If there are dependent stores in the bond graph or causal paths between resistive ports, then the mathematical model is formulated according to the features of the target input language. If, for instance, ACSL[®] is chosen, then algebraic loops are taken into account by means of the IMPL operator provided by the language ACSL ([76], Section 4.46, p. 74). Figure 11.50 shows the highlighting of a zero-order causal path between two inertias.

The zero-order causal path between the two inertias means that there is an implicit algebraic equation for the effort of the inertia with derivative causality. In the generated ACSL[®] output file, this is expressed by means of the IMPL operator (Figure 11.51). Alternatively, Figure 11.52 shows the Maple[™] input file generated by MS1[®].

- Animation of Energy Flows

MS1 was one of the first modelling and simulation environments to provide the user with the ability to animate energy flows directly on a bond graph. To that end, a contour is drawn on each bond. Its shape is similar to that of two aircraft wings sitting on the bond. Its height perpendicular to the bond indicates the ef-

⁶ Maple[™] is a trademark of Waterloo Maple Inc., 615 Kumpf Drive, Waterloo, Ontario, Canada N2V 1K8, <http://www.maplesoft.com>

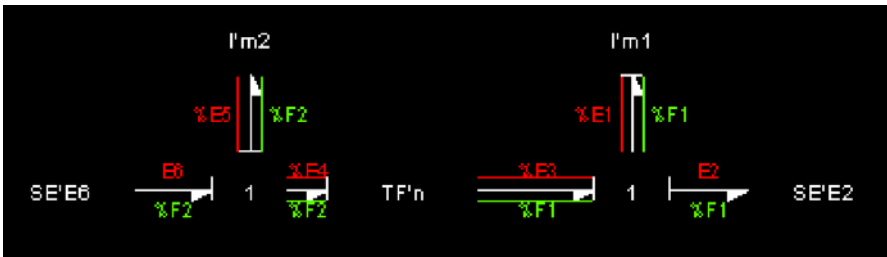


Fig. 11.50 Highlighting of the topological loop between the two inertia I:m1 and I:m2

```

DERIVATIVE ! NEW

! FIXED VARIABLES
  E2 = 1.0
  E6 = 1.0

! ALGEBRAIC VARIABLES
  F3 = P2/1.0
  F1 = P1/1.0
  F2 = F1*1.0

! IMPLICIT ZONE - BEGIN
  E5 = IMPLC(F3 - (F2), 0.0)
! IMPLICIT ZONE - END

  E4 = E6 - E5
  E3 = E4 * 1.0
  E1 = E3 - E2

! INDEPENDENT STATE VARIABLES
  P2 = INTEG(E5, G2)
  P1 = INTEG(E1, G1)

END ! DERIVATIVE NEW

```

Fig. 11.51 Derivative section of the ACSL file generated by MS1

fort and its width parallel to the bond represents the flow. Thus, the area inside this contour measures the instantaneous power along the bond. Moreover, the contour expresses the direction of the energy flow. If the instantaneous power is positive, the contour points into the direction given by the half arrow of the bond. In addition, a convex or concave shaped contour is drawn near each energy store indicating the stored energy. In the case of a linear energy store, the contour is either an upside or downside triangle [50].

```

# MAPLE MODEL : new
# MS1 V5.0d8 - licence 01S07-1-E - Thu Feb 19 12:27:12 2004
#
# ----- DECLARATIONS -----
P1:='P1':
E2:='E2':E6:='E6':F1:='F1':F4:='F4':F2:='F2':F5:='F5':P2:='P2':E4:='E4':
E3:='E3':E1:='E1':F6:='F6':F3:='F3':E5:='E5':
#
# ----- EQUATIONS -----
#
E2 = 1.0
E6 = 1.0
F1 = P1/1.0
F4 = F1
F2 = F4*1.0
F5 = F2
P2 = F5*1.0
E5 = D(P2)
E4 = E6-E5
E3 = E4*1.0
E1 = E3-E2
F6 = F2
F3 = F1
D(P1) = E1

```

Fig. 11.52 Maple input file generated by MS1

With regard to its capability of generating code for several simulation programs and to interface with computer algebra systems, MS1[®] resembles bond graph pre-processors like CAMP-G. There are also other features inspired by CAMP-G[®]. For instance, by double clicking on a bond graph element, its constitutive equations are displayed in a popup window. However, unlike the CAMP-G, MS1 supports development of hierarchical models and multi-formalism. Components of one and the same system can be described using different formalisms. On the other hand, the MS1 user interface enables the start of a chosen simulator in a *transparent* way, displaying results and animating energy flows on bond graphs. From this point of view, MS1 can be considered an integrated modelling and simulation environment.

11.6.6 Transformation Between Different Model Description Forms

The development of models of engineering systems serves different purposes. Simulation of the dynamic behaviour of a system is only one aim among others, although it is a very frequent one. Other objectives for model development are, for instance, system identification, the design of a controller or an observer of a dynamic system.

The graphical representation of models often depends on conventions in an engineering domain. In mechanics, it is common to use free body schematics, in control

theory block diagrams and in electrical engineering networks are used. Software supporting the modelling process must accept at least one form of a graphical representation and be able to convert it into other forms of representation. The required target form depends on the respective task. For time domain analysis, a system of ODEs and algebraic constraints, if necessary, is needed, while control analysis needs the coefficients of state space matrices or transfer functions in symbolic form.

Section 2.7 gives two procedures, one for mechanical systems and the other for non-mechanical systems, that enable the user to construct in a systematic manner a bond graph from a schematic. As demonstrated in Chapter 6, not only equations for simulations can be derived from a causally completed bond graph, but also transfer functions, equations of an inverse system as well as information about structural properties, viz. structural controllability and structural observability. Moreover, in Section 3.2 we noticed that causal bond graphs can be transformed into block diagrams if necessary. Thus, it is obvious to use bond graphs as a *generic* model description form, from which other forms of representation can be generated according to the requirements of an actual task.

MTT

This philosophy of physical systems modelling has led to the development of a set of *Model Transformation Tools*, MTT, [40] by Gawthrop at the University of Glasgow [37]. With this approach model development is considered an order of transformation steps from one model representation to another. One of these model representation forms are bond graphs. They take the role of a *core* model representation.

The implementation of this toolbox follows the usual UNIX philosophy of developing software with powerful capabilities by assembling available software tools dedicated to specific tasks. In MTT, model transformations are performed by software tools that are called in script files of the Bourne shell. Each model representation has two attributes. One of them is an abstract representation form that may be an acausal bond graph or a DAE system. The other attribute is an implementation language. First, graphical entry of an acausal bond graph is made possible by the drawing program *xfig* available as a UNIX utility. This graphical entry then is transformed into a textual description in the language *fig* and further into a description in the language Prolog. By means of rules formulated in Prolog, the initially acausal bond graph can be transformed into a causal one. Given a causal bond graph and constitutive element equations formulated in the input language of the formula manipulation package REDUCE [2], a set of DAEs can be generated in the language of REDUCE and further converted into the input format of the program Simulink®.

Another option is to derive the matrices of a linearised descriptor equation in symbolic form from the system of DAEs and to output them in MATLAB® input format. To that end, symbolic differentiation is performed in REDUCE. Eventually, transfer functions in symbolic form can be generated from the symbolic matrices of the linearised descriptor equation (cf. Section 6.2). The formulation of the system of DAEs as well as the matrices of the linearised descriptor equation in REDUCE can

be converted into L^AT_EX format for display in usual mathematical form on a monitor or for output in a document.

The software tools included in the set of MTT enable the transformation between various model representations and thus allow for different modelling views. The use of a model representation, however, is left to an independent program that can process the model representation appropriate for a given task. That is, simulation can be performed by means of Simulink[®], while MATLAB[®] can be used for a frequency domain analysis. The concept and the implementation of MTT, however, allow for using other programs as well. To that end, corresponding transformation tools must be added to MTT.

One might argue that the execution of transformation tools from a toolbox by means of UNIX shell scripts is less appealing to a model developer than the use of an integrated modelling and simulation environment. However, MTT, as a result of a research project, is a non-commercial software that is mainly built on GNU tools. Instead of MATLAB[®], the free GNU program Octave [1] could be used. Furthermore, MTT is open for inclusion of other tools, e.g. of new powerful numerical solvers.

Beyond the concept of transforming model representations, MTT offers the following features. It supports

- hierarchical bond graphs,
- hybrid modelling of systems by providing switched I and C components [39], and
- the concept of bicausalities introduced by Gawthrop [38].

MAX

In the previous consideration of different types of software packages, an obvious question has been whether and how the software supports bond graph modelling. Actually, modelling of engineering systems mostly starts with a schematic that is more intuitive than a strictly formalised and unambiguous graphical representation. The purpose is to express relations between objects at a *conceptual* level. These can stand for a physical effect, a basic building block, or a system component composed of building blocks. A schematic may be considered a graphical representation of a so-called *Ideal Physical Model* (IPM) in which physical phenomena are described under some *idealised* assumptions. This idealisation includes, for example, that physical effects are considered isolated by disregarding other involved effects and by assuming that they can be spatially lumped.

In schematic system representations, icons are interconnected that may depend on the application field. Moreover, not every engineering domain uses standardised symbols as in electrical engineering (IEEE, 1987, [46]). Such (application specific) symbols are usually called *icons*. Graphs in which nodes are represented by icons are called *iconic diagrams* (see, for instance, [27]). Although symbols in schematics are not standardised in all application domains, they are quite common in the early conceptual design phase of the modelling process. A computer aided analysis of a model, however, requires an unambiguous formalised description. The latter can

take the form of a bond graph. In Section 2.7, two procedures have been given for systematically converting a schematic of a mechanical and of a non-mechanical system into a bond graph.

The modelling environment MAX (Modelling and Analysis eXpert) for the design of mechatronic systems [27, 28] was developed in a research project around 1994. To the author's knowledge, the software has not been further developed since then. MAX supports both levels of abstraction, that is, ideal physical models as well as bond graphs by corresponding editors [27, 82]. Automatic transformation from one representation form into the other ensures consistency between both forms. That is, not only the transformation from a schematic into a bond graph is supported, but also its back transformation. If a model is changed by using one of the two representation forms, changes are automatically and consistently taken into account in the other representation form. This is achieved by the use of an internal core representation.

The model developer can start at the conceptual level and, supported by a browser, select submodels from a library according to given specifications. First, one may attempt to achieve a design solution with simple submodels that partially comply with given requirements. After assessment of simulation results, the modeller must decide which submodels shall account for which further properties. This way, a model can be developed that is just as complex as necessary in order to meet the requirements of a given design task. Accounting for further effects step-by-step and removing others that have proven irrelevant in contrast to expectations helps to avoid unnecessarily complex models. However, there is no knowledge-based assistance by the software in deciding which of the submodels of a system component available in the library should be chosen to meet the given requirements. It is the modeller's experience that is required to control the selection of submodels of appropriate complexity. As is well known, accounting for some effects in submodels expected to be relevant for the overall system dynamics can lead to long simulation runs and even to numerical problems. It is the art of modelling to single out such effects and to meet the given system requirements.

Once submodels have been chosen according to the given specifications, a bond graph can be considered an intermediate format in the transformation of an ideal physical model into a mathematical model.

The *Bond Graph Toolbox* provides special routines for converting a system schematic having the form of an electrical circuit into a bond graph. These routines can be called in Mathematica[®].

In MTT, bond graphs are created by means of the drawing tool *xfig*. Likewise, iconic diagrams could be created. What is needed is a transformation tool that converts a textual description of the iconic diagram in the language *fig* into the *fig* description of the corresponding bond graph.

CAMBAS

The automated modelling environment CAMBAS (Computer Aided Model Building Automation System) developed by Louca and Stein at the University of Michigan supports two levels of abstraction [73].

On the top or component level corresponding to the level of word bond graphs, components with a fixed number of ports can be connected to interact with other components. Internally, components are assigned a certain number of models of increasing complexity. Stein and Louca use the term *template*. On the second hierarchical lower element level, each component has a certain bond graph implementation.

At the top level of abstraction, components are graphically represented by rectangular blocks and their power ports by dots. Components are displayed in different colours according to the energy domain they belong to. Line segments between ports indicate energy flows between them. Building system models by means of components is supported by a library of components, where each component is assigned a certain number of bond graph models. According to their functionality, components are grouped into classes. For each component, information about its characteristic features is available. An editor allows the creation of a bond graph model of a component not available in the library. If the appropriate model of a subsystem is a distributed parameter model, then it is approximated by a lumped parameter model based on the modal analysis method. Users can build a model by taking system components, e.g. a DC motor, a hydraulic pump, a pressure relief valve, etc., from the library and by connecting them. The result, in the first place, is a word bond graph model where the complexity of the models of the components still needs to be defined.

In the modelling environment MAX, a browser helps to navigate through the library. However, it is up to the user to decide which submodels of which complexity are appropriate to ensure that the overall system model complies with given specifications with regard to the system dynamics. The essential feature of the template based approach implemented in CAMBAS is that the software attempts an algorithmic solution to the synthesis problem. Wilson and Stein call component models of minimal complexity that ensure a required system behaviour, *proper models* [88]. To have the software algorithmically determine these proper models, the user only has to specify the range that includes natural frequencies of the system. This means that models must be linear which is of some disadvantage. However, the automatic iterative determination of component models of minimal complexity with respect to certain criteria is certainly an extraordinary feature of CAMBAS. A description of the *Model Order Deduction Algorithm* (MODA) implemented in CAMBAS is given in [88]. In order to see how the model of a component changes during the iterative determination of the proper model, the bond graph of the component can be displayed by the graphical editor. Finally, the modeller can have the software output the bond graph of the overall system model of minimal complexity.

11.7 Exchange and Reuse of Bond Graph Models

The previous sections of this chapter clearly demonstrate that there are modern modelling languages as well as a whole spectrum of software from mathematical programs, preprocessors for simulation programs to advanced integrated modelling and simulation environments that can support bond graph-based physical system modelling of multi-energy domain engineering systems in different ways. Sophisticated modelling and simulation environments usually support multiformalism and come with model libraries for a number of application areas. Some of these programs, e.g. 20-sim use an underlying modelling description language such as SIDOPS, while others support the formulation of user defined component models in a higher programming language such as C++ (SYMBOLS).

In the realm of object-oriented modelling, the modelling language Modelica has emerged from international efforts of unifying features of several object-oriented modelling languages and to support the exchange of non-causal reusable models. As shown in Section 11.5, Modelica can be used for describing bond graph models also, though it is based on the generalised network paradigm. Accordingly, in 2005, Nebot and Cellier presented a general Modelica library for bond graph modelling [23]. However, it appears that bond graphs are not widely appreciated as a modelling formalism in the object-oriented modelling realm, nor is Modelica much used as a language for the description of bond graph models.

As a matter of fact, bond graph methodology along with various supporting software and model libraries are used for modelling and design of engineering systems in different places in academia and in industry worldwide. However, to the author's knowledge, little effort has been undertaken so far to support the exchange and reuse of bond graph models, to share the associated engineering knowledge and to avoid redevelopment of bond graph models of building blocks that have been already developed elsewhere. Bond graph modellers have widely agreed on conventions of drawing bond graphs, but a widely accepted definition of a format for the description of bond graph models that enables the exchange of bond graph models between different software for bond graph modelling and the reuse of engineering models is still missing.

In contrast, XML, the eXtensible Markup Language [87], has become a popular language for data modelling in various areas including scientific and engineering disciplines. The language, widely adopted by the computer industry, enables one to define the content of a document and to separate the content from its presentation. There are parsers that can validate rules defined in so-called Document Type Definitions (DTDs) or in XML schemas, respectively. Moreover, there are languages and transformation tools that can be used to extract information from an XML document in a form required for further processing by a target application software. The relevance of XML for modelling and simulation, in general, has been briefly pointed by Fishwick in 2002 [35].

Inspired by the use of XML in the computer industry, the author proposed an XML based schema for bond graphs[10] in the 2005 International Conference on Bond Graph Modelling and Simulation (ICBGM' 05), while Bevan in the same

conference outlined another data model for bond graphs that he and his co-authors created for use in their Next-Generation Transformation Tools (NTT) project [7] following the open source software environment *Model Transformation Tools* (MTT) [5, 40]. In the following, the concept of using XML as a basis of a format that can support the exchange and reuse of bond graph models is illustrated. For a more detailed presentation see [11].

11.7.1 Useful XML Features for the Description and Processing of Bond Graph Models

For the reconstruction of a bond graph from a given textual description and for its further processing, it is essential to be able to identify objects and relationships between them, viz. to identify bond graph elements, ports and bonds. XML enables one to tag items with meta-information of unique meaning in a given context and to formally describe the data in a document and the document's structure. As a result, objects and their relations in a physical system model can be mapped onto XML elements. In contrast, if a software program would output the information about a bond graph in a given internal data format, then reconstruction of the bond graph by another software, clearly, would be difficult and inefficient without any meta information. Information about an object in the context of bond graph methodology enclosed by a starting and an ending XML tag will be called an XML element. Such XML elements, the order in which they can appear, and how they can be nested inside each other are defined either in a so-called Document Type Definition (DTD) or in an XML schema. That is,

- XML enables a formal description of the content of a document.
- XML documents can be validated against the rules defined in a DTD or in an XML schema.
- XML meta-information enables one to perform a *content oriented*, application specific processing of XML documents. Consequently, export and import software tools can be written that transform internal data formats into XML and vice versa.
- Furthermore, the Extensible Stylesheet Language (XSL) and the Extensible Stylesheet Language for Transformation (XSLT) enable one to select information in an XML document, to specify the formatting and to output it such that it can be processed by another application. For this process, there is an XSL processor that reads directives from a stylesheet as to how the logical structure of an XML structure is to be converted into a presentation structure.
- XML, XSL and XSLT have been standardised by the W3 Consortium. Moreover, there are a number of freely available XML programming interfaces that can be used to reduce the effort of developing application specific XML processors.

The XML features outlined above suggest that XML is ideally suited for the definition of an exchange format for bond graph models that can be processed auto-

matically. As XML documents are ASCII files, in principle, they can be read and manipulated by humans also. However, they tend to increase considerably in length.

11.7.2 Information that an Exchange Format for Bond Graphs Should Capture

Present day bond graph software provide a graphical front end that supports a graphical construction of bond graph models. With regard to the definition of an exchange format, it has to be taken into account how model developers are supported and what information is stored.

- The hierarchy in composed bond graph models and the use of submodels is handled differently in general.
- Should an exchange format provide information about power reference directions and causal strokes? This information is added automatically by some bond graph editors.
- Some bond graph software provide an equations editor for specifying nonlinearities, while other bond graph programs rely on an ordinary text editor or the features of a target application software. The preprocessor CAMP-G[®], for instance, does not allow one to specify nonlinearities. It is expected that this is done in MATLAB[®].
- Some advanced modelling environments provide a modelling language, while others require specification of nonlinearities in a programming language such as C++.
- Some bond graph software does not support multibond graphs.

These observations suggest that a format for exchange and reuse of bond graph models should include as much information as possible organised in such a manner that it can be easily and efficiently accessed. Moreover, a truly efficient exchange format should be flexible enough so that requirements emerging from its application in different fields can be accounted for by further extensions without the need of a redesign. As a result, an XML based exchange format for bond graph models should capture

- the hierarchy of a composed model,
- its topological structure and that of its composed submodels, i.e. which nodes of which type are connected in which way,
- information about the graphical representation to enable a bond graph to be re-drawn by another editor or by a browser,
- constitutive relations of elements and of submodels with no internal structure,
- names, numerical values and physical units of parameters and initial conditions to allow for reproduction of simulation results,
- meta-information about the model and its submodels including submodel name, version, revision date, revision history and bug-fixes, submodel authors' names, and a short description where and how to use a (sub-)model.

According to an object-oriented modelling approach, the description of the interface of a model component (submodel, or lowest-level element) to its outside world should be separately formulated from that of the components body.

Depending on its purpose, bond graph related software will not need all of the above information. Clearly, a bond graph preprocessor, e.g. CAMP-G[®], will not need numerical values of parameters and initial conditions in order to output equations for processing by the Symbolic Math Toolbox[®] of MATLAB[®]. On the other hand, for an exchange of models between self-contained bond graph modelling and simulation environments, however, all kinds of information listed above is needed. Furthermore, capabilities of software, not especially designed for supporting bond graphs, e.g. mathematical software such as MATLAB[®], Scilab [71], or Mathematica[®], are also used for processing bond graph models in practice. Such software programs need equations in their mathematical input language, but not the information about the graphical representation of a bond graph. To that end, an import software tool that derives the equations of a mathematical model from the XML based description of a bond graph and outputs them so that they can be processed by a mathematics program, e.g. Scilab, will need information about power orientation and computational causalities. In contrast, it is the graphical representation of a bond graph that is essential for an efficient (distant) communication between model developers.

Finally, the use and the incorporation of submodels developed elsewhere into a model under construction demands for giving credits to submodel authors and requires good documentation of each submodel. This is especially true in the development of large complex models in an industrial environment where project management and future maintenance of models is of importance.

11.7.3 A Schema for an XML Based Description of Combined Bond Graph and Block Diagram Models

Taking into account the previous general considerations, the author developed a schema for the description of models composed of a bond graph and block diagrams (The bond graph represents the engineering system or process, block diagram parts display its control). Inspired by the acronym MathML, an XML application for the description of mathematical equations, the novel markup language has been termed BGML. In the following, the BGML schema is briefly illustrated. A more detailed presentation including information on how the schema is processed in an experimental software environment is given in [11].

The schema has been graphically designed by means of the powerful free software Altova XMLSpy^{®7} [3]. Hence, subsequent illustrations of parts of the schema make use of the way the software presents relations between XML items. Fig-

⁷ XMLSPY is a registered trademark of Altova GmbH, Rudolfsplatz 13a/9, A-1010 Wien, Austria/EU, <http://www.altova.com>

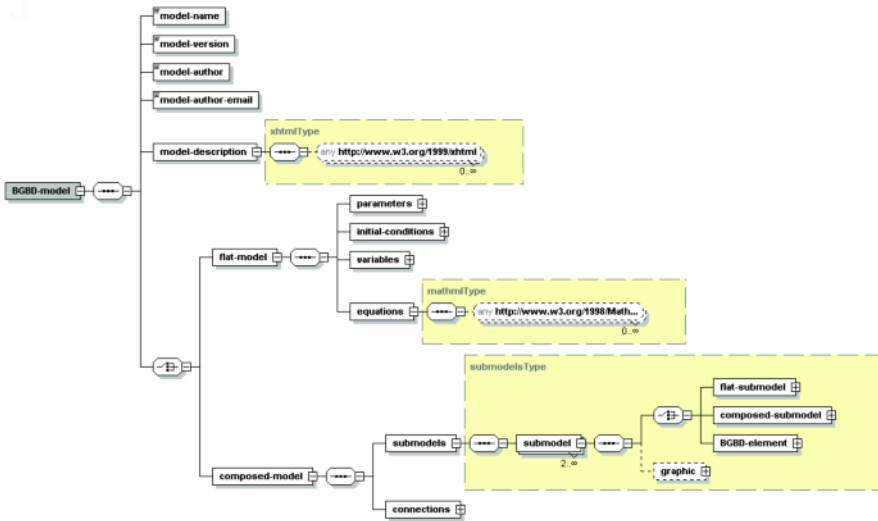


Fig. 11.53 Root of the BGML schema and its successors

Figure 11.53 shows the root XML element of the schema labeled *BGBD-model* and its immediate successors.

The connector between the root element and the perpendicular branch marked with a line of dots arranging XML elements in an XML schema is called *sequence*. It means that the starting tag of the root XML element *BGBD-model* must be followed by the XML elements *model name*, *model version* etc. in exactly that order. That is, the root element of the XML structure is a container element that includes the bond graph - block diagram (BGBD) model as well as meta-information about this model. The other type of connector between the most left perpendicular branch and the XML element *flat model*, *composed-model* respectively is called *choice*. It means that the XML element *model-description* may be followed either by an XML child of type *flat-model* or of type *composed-submodel*. In this context, it has to be recalled that XML elements must not overlap. A starting tag inside an XML element must be followed by its corresponding closing tag inside that element. It is a prerequisite of XML documents that they are so-called well-formed.

In BGML, the description of a bond graph - block diagram (submodel) must start with general information about the model (submodel) such as model name, model version etc. The XML element *model-description* accounts for any information about the purpose of the model, model assumptions, model features, etc. This information is expected to be given in XHTML format. The cardinality attribute (0...∞) attached to the ellipsis with a dashed line and the string *any http://www.w3.org/1999/xhtml* means that there may be no such XML element of type *xhtmlType* or any number.

The model itself may be either a set of equations or be composed of submodels. In the first case, we call the model a flat model and describe its equations in

MathML. The equations are preceded by a listing of parameters, initial conditions and variables in that order. In the case of a hierarchical model, it is assumed that there are at least two submodels and at least one connection. Submodels, in turn, may be either flat submodels, composed submodels, or either a bond graph or a block diagram element. Finally, each submodel has a graphical representation. In order to be able to check the validity of XML descriptions of bond graphs against the schema without having to specify equations in MathML, the admissible minimum number of equations has been set to zero. Also, to ease a manual development of XML descriptions of bond graph models for test purposes, information about the graphical representation of nodes in a graph and specification of parameters may be omitted. In contrast, the cardinality attribute associated with the XML element *submodel* enforces that there are at least two submodels in a composed submodel. Such requirements may be easily introduced into the XML schema and relaxed for convenience at a graphical level by using an appropriate tool, e.g. XMLSpy®.

If a submodel is a composed submodel, then it must have at least one port to the outside world. This port can be either a power or a signal port. In case of a signal port, a further distinction can be made. Signal generators as well as ordinary junctions only have outgoing signals (*signal-generator-port*), monitors, or signal detectors allow for ingoing signals only (*signal-detector-port*). Furthermore, modulated elements only accept an ingoing signal at their signal port. Connections between ports may be either bonds or signals (activated bonds). They have an orientation from a source port to a target port. In the trivial case, a composed submodel may contain a single submodel connected to the submodel’s port having no further ports. This has been taken into account by the cardinality attribute of the *choice* connector following the XML element *connections* in Figure 11.54.

Children of the XML element *BGBD-element* can be either bond graph or block diagram elements. Bond graph elements may be classified into junctions, modulated elements and such elements that are neither junctions nor modulated elements, viz.

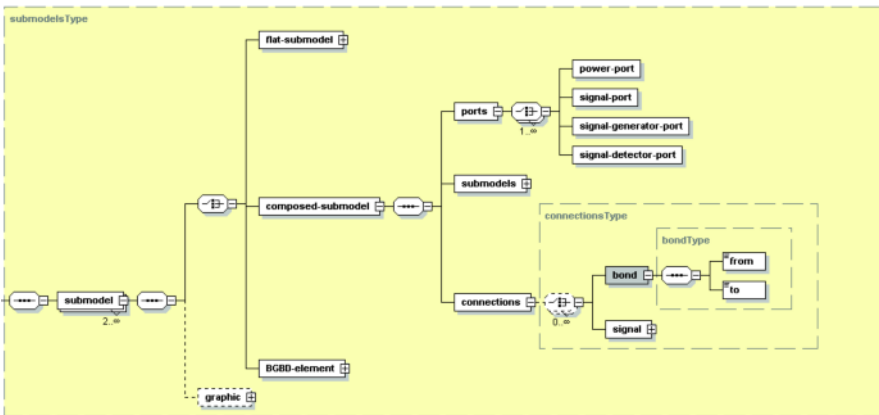


Fig. 11.54 The XML element *composed-submodel* and its successors

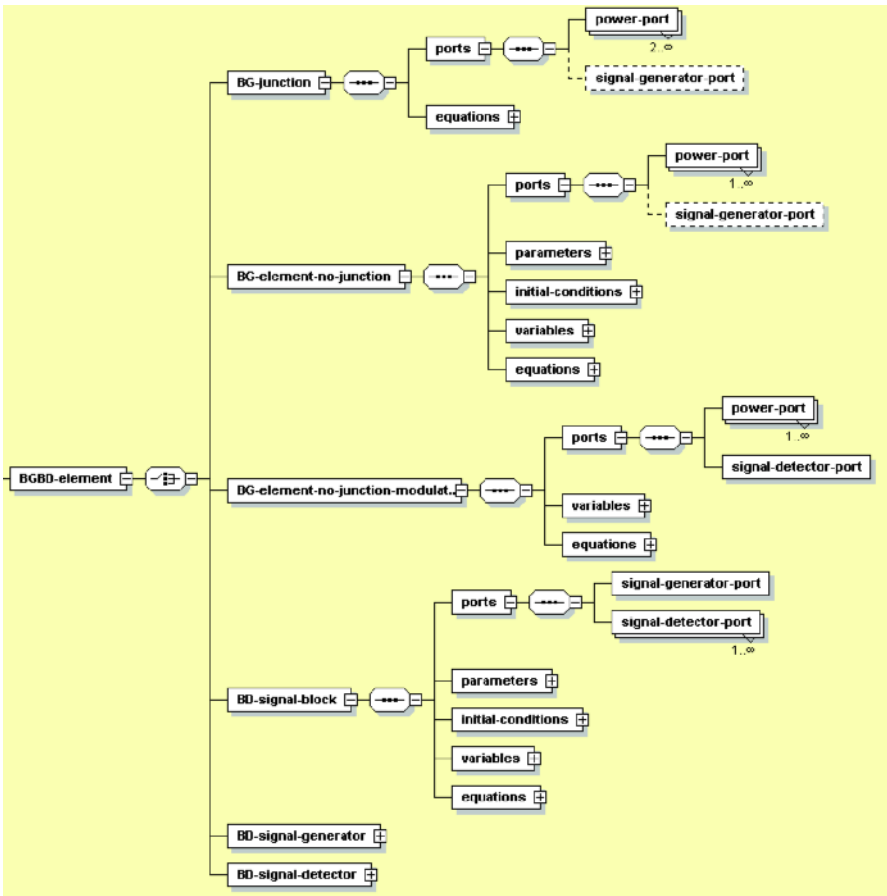


Fig. 11.55 Part of the XML schema covering bond graph and block diagram elements

a member of the set $\{Sf, Se, C, I, R, TF, GY\}$. Subsequent Sections 11.7.4 and 11.7.5 consider how pseudo bond graphs and controlled junctions can be included. First, Figure 11.55 shows the part of the XML schema that covers bond graph and block diagram elements.

Clearly, a bond graph junction must have at least two power ports. This is reflected by the cardinality attribute of XML element *power-port* in the most upper part of Figure 11.55. As the common effort of some 0-junctions or the common flow of some 1-junctions is needed as an input signal into a signal port, the XML description of a junction may have an optional part describing this port with an outgoing signal (*signal-generator-port*). Bond graph elements that are no junctions and are not modulated must have at least one power port. For energy storage elements, the energy variable may be an outgoing signal, e.g. in the modelling language SIDOPS. If a bond graph element is modulated, it must have a port for an ingoing signal

```

<xs:element name="BG-element-no-junction-modulated">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ports">
        <xs:complexType mixed="false">
          <xs:sequence>
            <xs:element name="power-port" type="power-portType" maxOccurs="unbounded"/>
            <xs:element name="signal-detector-port" type="signal-portType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="parameters" type="parametersType"/>
      <xs:element name="variables" type="variablesType"/>
      <xs:element name="equations" type="mathmType"/>
    </xs:sequence>
    <xs:attribute name="type" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="MR"/>
          <xs:pattern value="MSe"/>
          <xs:pattern value="MSf"/>
          <xs:pattern value="MGY"/>
          <xs:pattern value="MTF"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

```

Fig. 11.56 Listing of the XML element *BG-element-no-junction-modulated*

(*signal-detector-port*). Finally, block diagram elements that are no signal generators nor signal detectors must have at least one port for an ingoing signal and one port for an outgoing signal.

The classification of elements along with requirements associated with the type of an element and implemented in the schema helps to exchange bond graph models that are consistent with the rules of the bond graph methodology because if a BGML description of a bond graph model does not pass the validation against the schema, it is rejected. A bond graph description that is syntactically correct with regard to the bond graph language can even be more enforced by the fact that XML elements may have attributes for which admissible patterns can be specified. For instance, if a bond graph element is modulated, its type identifier should be a member of the set {MSf,MSf,MR,MTF,MGY}. The software Altova XMLSpy[®] automatically transforms the graphically designed XML schema into XML. Figure 11.56 shows the listing of the XML element *BG-element-no-junction-modulated* and its use of the attribute *type* capturing the list of admissible bond graph element types.

As can be seen, modulated storage elements are not admissible as they do not exist in *true* bond graphs. Furthermore, the XML element *BG-element-no-junction-modulated* makes use of the XML element *parameters* which is of type *parameterType*. Admissible attributes of this type are *name*, *value*, *physical unit*. Specification of this kind of information can be either enforced or released by setting *use* to *required* or *optional*. It is up to the software processing this information whether it is correct with regard to syntax and sematic. Finally, Figure 11.57 shows how a power port can be described in XML.

```

<xs:complexType name="power-portType" mixed="false">
  <xs:complexContent mixed="false">
    <xs:extension base="portType">
      <xs:attribute name="domain" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="electric"/>
            <xs:pattern value="hydraulic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="causality" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="indifferent"/>
            <xs:pattern value="fixed effort out"/>
            <xs:pattern value="fixed flow out"/>
            <xs:pattern value="preferred effort out"/>
            <xs:pattern value="preferred flow out"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Fig. 11.57 Listing of the XML complex type *power-port*

The complex type *power-port* uses the complex type *power-portType* which is an extension of the type *portType* and may have an optional attribute *domain*. For reasons of space limitation, not all admissible strings, e.g. electric, hydraulic, etc. have been listed. Another important attribute is *causality*. Specification of causality is required as some target software needs this information but does not assign causalities itself. The choice of admissible strings for the value of the causality attribute has been inspired by the modelling language SIDOPS. For instance, a port of an effort source naturally has a fixed effort out causality, while a port of C element has a preferred effort out causality. As can be seen from Figure 11.57, the string *indifferent* is also admissible, allowing one to describe non-causal bond graphs. The latter attribute value can be used if a BGML description of a bond graph is passed to an import software tool for an advanced modelling and simulation environment such as 20-sim which automatically assigns reference directions and computational causalities when loading a bond graph description in the internally used format. The causality attribute *indifferent* is also useful for a BGML description of non-causal component models stored in a library. Attributes of XML types enable one to define properties of modelling objects locally without having the need for introducing further XML elements. As a result, the structure of the schema can be kept

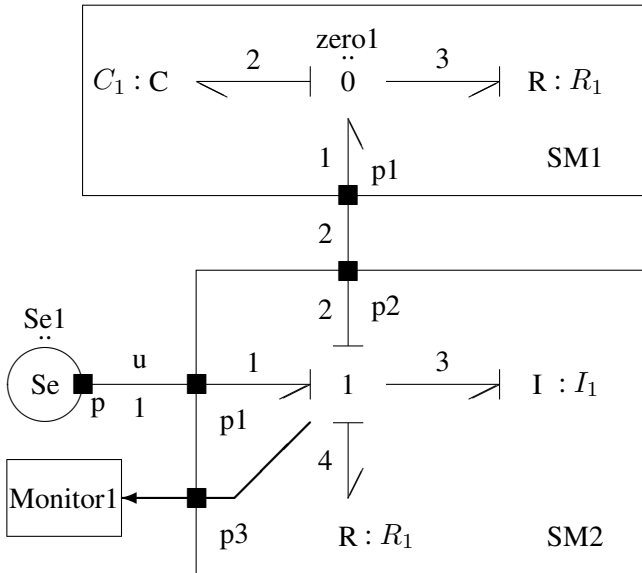


Fig. 11.58 Example of a simple hierarchical bond graph

simpler which, in turn, can facilitate the development of transformation tools. Only the XML element *BGDB-model*, the XML complex types *submodel*, *port*, *power-port*, *signal-port*, *equations* and some other types, e.g. *initial-conditions*, are defined globally.

Example: A Simple Hierarchical Bond Graph Model and its BGML Description

XML descriptions, although human readable in principle, tend to be quite long with many nested elements. Nevertheless, for illustration, an outline of the BGML description of the simple hierarchical bond graph in Figure 11.58 shall be given. In the bond graph of Figure 11.58, black squares denote submodel ports. Their names as well as the enumeration of bonds is relative to the submodel they belong to. Figure 11.59 shows a coarse outline of the associated BGML description.

As the example bond graph is a composed model, the listing of top-level submodels is followed by a listing of their interconnections. The first submodel reduces to the bond graph element of type *Se*. The second submodel in that listing is a composed submodel. Hence, first its ports constituting the interface to the outside world are listed. The description of the submodel's interface must be followed by a listing of all submodels and a subsequent listing of all connections inside that submodel. For clarity and space limitations, only the description of the *I* element is outlined. The description of submodel *SM1* and of connections have been skipped. The coordinates associated with the name of a submodel are needed for positioning the name as an annotation in a drawing of the bond graph. As can be seen, BGML also

```

<model-name>SampleBG</model-name>
<composed-model>
<submodels>
  <submodel name="Se" x-coord="100" y-coord="200">
    <BGDB-element>
      <BG-element-no-junction type="Se">
        <ports>
          <power-port name="p" id="1" orientation="out" dimension="1" causality="fixed effort out"/>
        </ports>
      </BG-element-no-junction>
    </BGDB-element>
  </submodel>
  <submodel name="SM2" x-coord="200" y-coord="200">
    <composed-submodel>
      <ports>
        <power-port name="p1" id="1" orientation="in" dimension="1" x-coord="100" y-coord="200" domain="power"
causality="preferred flow out"/>
        <power-port name="p2" id="2" orientation="out" dimension="1" x-coord="200" y-coord="100"
domain="power" causality="preferred flow out"/>
        <signal-port name="p3" id="1" orientation="out" dimension="1" x-coord="300" y-coord="300"/>
      </ports>
      <submodels>
        <submodel name="I1" x-coord="300" y-coord="200">
          <BGDB-element>
            <BG-element-no-junction type="I">
              <ports>
                <power-port name="p" id="1" orientation="in" dimension="1" causality="preferred flow out"/>
                <signal-port name="state" id="1" orientation="out" dimension="1"/>
              </ports>
            </BG-element-no-junction>
          </BGDB-element>
        </submodel>
      </submodels>
    </composed-submodel>
  </submodel>
</submodels>
<connections>
  <bond id="1" dimension="1" effort="E">
    <from>Se|p</from>
    <to>SM2|p1</to>
  </bond>
  <signal id="1" dimension="1">
    <from>SM2|p3</from>
    <to>MONITOR1|sdp</to>
  </signal>
</connections>
</composed-model>
</BGDB-model>

```

Fig. 11.59 Skeleton of the BGML description of the sample hierarchical bond graph in Figure 11.58

supports multibonds. Ports as well as bonds and signals may have a dimension > 1 . Furthermore, bonds and signals may have a numerical identifier and annotations. In order to distinguish port names at a hierarchy level, they are preceded by the name of the submodel they belong to. For instance, at the top level of the hierarchy, bond 1 is points from port p of the effort source to p1 of submodel SM2, while there is a signal with the identifier 1 from port p3 of submodel SM2 to the signal detector port, sdp, of submodel MONITOR1.

In the XML listing of Figure 11.59, equation sections have also been skipped also. The formulation of equations in MathML makes the listing very long.

11.7.4 Pseudo Bond Graphs in BGML

So far, an XML description of models composed of true bond graphs and block diagrams have been considered. One of the advantages of the BGML schema is that rules of the bond graph methodology have been implemented. That is, violations can be detected while an XML description of a bond graph model is validated against the schema. For example, modulated storage elements are not admissible bond graph element types for the XML element *BG-element-no-junction-modulated*. For the description of thermo-fluid phenomena, however, pseudo bond graphs have proven useful and appropriate. As pseudo bonds do not carry power variables modulated storage elements, violating the principle of energy conservation may be admissible. Therefore, in the BGML schema, the choice compositor `<xsd:choice>` below the XML element *BGBD-element* allows for a subtree with the root *Pseudo-BG-element* (Figure 11.60). Admissible pattern for the type of this XML element are all bond graph elements including MC- and MI-storage elements. Clearly, pseudo bond graph elements must have at least one port of type *pseudo-power-portType*. They may also have signal ports with an inward orientation accounting for signals modulating the element (*signal-detector-port*). The XML element *pseudo-power-port* has an attribute *domain*. Admissible patterns in this case may be *pseudo* or *pseudothermal*. Some pseudo bond graph elements, e.g. the thermal accumulator introduced by Karnopp may have a true power port. This extension to the BGML schema ensures that modulated storage elements are allowed only in pseudo bond graph models. As power ports do have an attribute *domain*, software for processing an XML document that contains a bond graph model can check if the domain of two power ports connected by a bond have the same admissible pattern.

11.7.5 Controlled Junctions in BGML

As to the abstraction of instantaneous changes between modes in a dynamic system and its appropriate description in bond graph models, one of the extensions to bond graph methodology proposed by P. Mosterman are controlled junctions (cf. Section 7.2.2). In its on-mode, a controlled junction behaves as a normal junction. In its off-mode, a controlled 0-junction acts as an effort source that provides a vanishing effort. Accordingly, in its off-mode, a controlled 1-junction acts as a flow source of value zero. This ideal switching behaviour of controlled junctions is governed by a *local* control algorithm for each junction. Clearly, in a simulation program, ideal switching requires appropriate changes to the control of numerical integration, e.g. restart and proper re-initialisation. In contrast to ordinary junctions, controlled junctions must have a signal port with *inward orientation*. One way to account for controlled junctions in the BGML schema is to add an XML element *BG-junction-controlled* to the compositor `<xd:choice>` below the XML element *BGBD-element* in the BGML tree (Figure 11.61). This new XML element has an attribute *type* with admissible patterns XZERO and XONE. The local algorithm controlling a junction

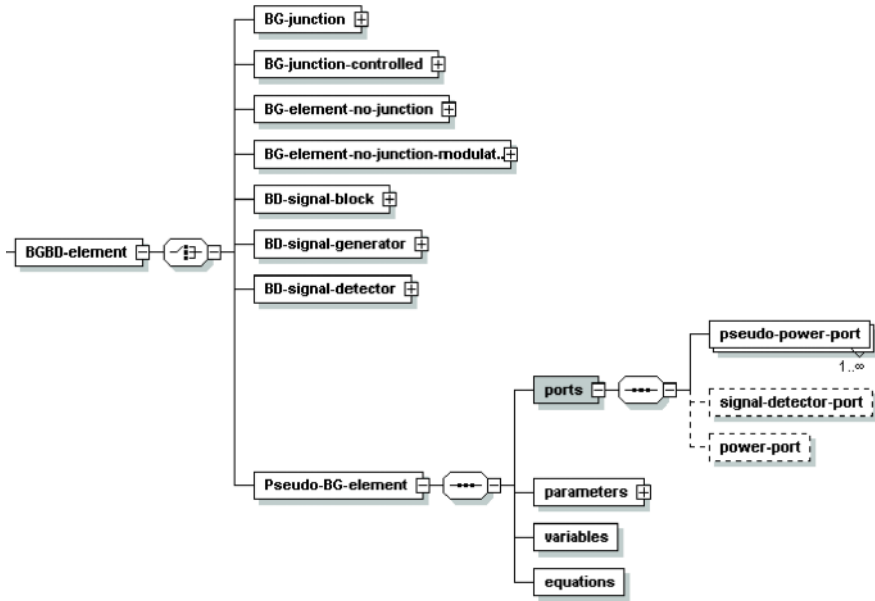


Fig. 11.60 Accounting for pseudo bond graph elements in the BGML schema

can be mapped onto a block diagram with signal inputs from other parts of the bond graph and a signal output switching the junction.

11.7.6 Supporting the Exchange and Reuse of Submodels

The previous discussion implicitly focussed on an XML description and the exchange of *entire* bond graph models. However, with the development of large complex models, submodels are likely to come from various designers or groups of model developers in academia and industry. Consequently, it is important that an XML schema for bond graph models not only accounts for meta-information about an entire model, but also for meta-information about each submodel in order to support project management and future maintenance of large models under construction. In order to facilitate the exchange of submodels and the (distant) co-operation of designers, meta-information about submodels should include information such as the name of the submodel, version, revision date, revision history, bug-fixes, the submodel authors' names and a short description of where and how to use a submodel.

Figure 11.62 shows the XML elements in the container element *submodel* that capture meta-information about a submodel. The XML element *model-description* of type *XHTMLType* can hold any information about a submodel. XHTML has been

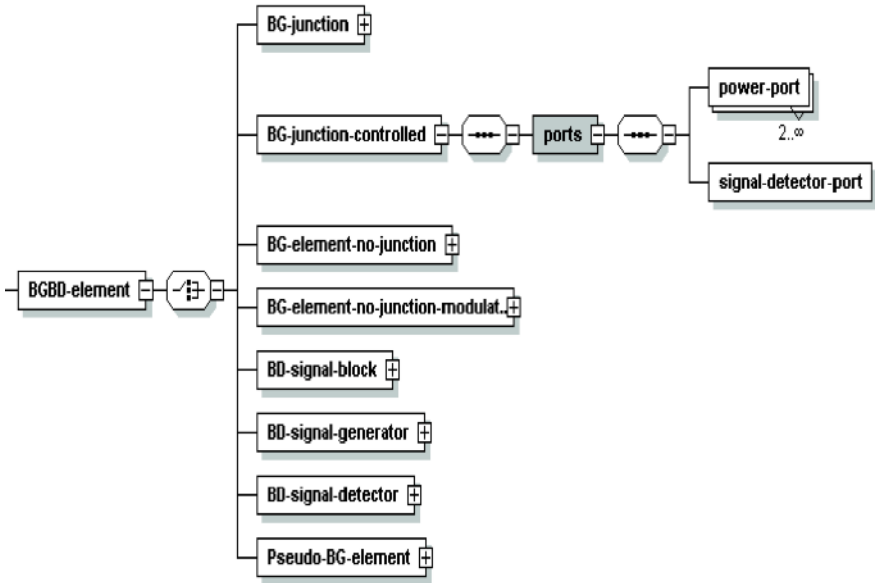


Fig. 11.61 Controlled bond graph junctions in the BGML schema

chosen as a format because some modelling software environments generate model documentation in HTML and because it has become common to use HTML for the exchange of documentation and to view it by means of a web-browser. Meta-information about a submodel could be structured in more detail by means of further XML elements. Note that these XML elements do not guarantee that meta-information about a submodel, in fact, is provided. They can be empty. The schema only requires that these XML elements occur in the required order. As a submodel can reduce to a basic bond graph or block diagram element, the BGML schema requires that meta-information or at least their XML tags are even provided for each basic element.

To support the exchange of submodels or the building of XML based component model libraries (cf. Section 11.7.8), the XML element *submodel* is a *global* element. That is, besides the XML element *BGBD-model*, it can be the root of a tree of XML elements. Consequently, an XML document may contain just the description of a submodel and it can be validated against the BGML schema.

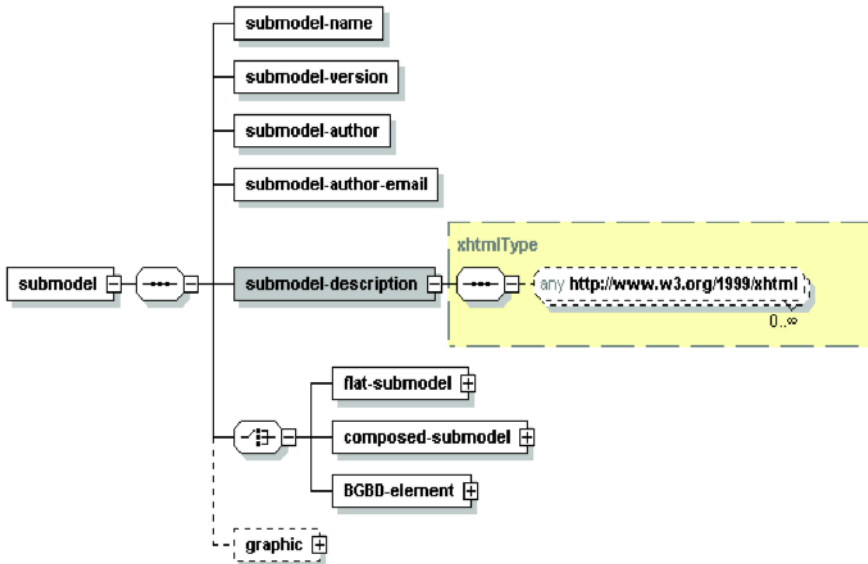


Fig. 11.62 XML elements capturing meta-information about a submodel

11.7.7 Transforming the BGML Description of a Bond Graph Model into a Target Language

Transformation from One XML Description into Another

If there is an XML description of a bond graph (sub-)model, it can be transformed into another XML format that conforms to a different schema by means of an open source implementation of the XSLT processor, e.g. Saxon-B, or Xalan. Such software processors transform a source XML tree into an XML target tree. In this transformation, the processing of each node of the source tree can be controlled by directives in an XSL stylesheet. For instance, in BGML, the type of a bond graph element is an attribute of the XML element introduced for a class of bond graph elements, e.g. elements that are not a junction and that are not modulated. In another XML description of bond graph models, the type could be an XML element called *BG-elementType*. To achieve such a conversion, an XSL stylesheet would contain the lines

```
<xsl:output method="xml"
    encoding="ISO-8859-1" indent="yes"/>
...
<BG-elementType>
  <xsl:value-of select="@type"/>
</BG-elementType> .
```

Due to different schemas, XML descriptions of a model may look quite different and an XSL stylesheet controlling the conversion can be lengthy and complex. However, there are tools that support a *visual* design of XSL stylesheets, e.g. in the free Home Edition of Altova XMLSpy® package [3]. The corresponding code is automatically generated.

Transformation from BGML into SIDOPS

In order to demonstrate the usefulness of an XML description of bond graph models according to the introduced BGML schema as an exchange format and the feasibility of transformations, a prototype of a stand alone software processor has been developed and implemented in Java [53] that can transform an XML description of a bond graph model according to a first working BGML schema into the modelling language SIDOPS (cf. Section 11.4). An implementation of a transformation from BGML into SIDOPS is facilitated in the sense that the transformation does not need to account for power bond orientations and causalities at power ports. The program 20-sim, when loading a SIDOPS description of a bond graph, automatically adds this information. Moreover, 20-sim is able to automatically derive a set equations from the bond graph.

Transformation into a Mathematical Script Language

Furthermore, a comprehensive BGML description of a bond graph model can serve not only as an exchange format between different bond graph software, but can be transformed so that software that is not designed to support bond graph modelling can use the transformed information as well. For instance, the equations of a bond graph model can be extracted from its BGML description and can be organised so that the result can be an input into the open source mathematical software Scilab [71]. In contrast to a transformation from BGML to SIDOPS, the transformation from BGML into the mathematical input language of Scilab is a more challenging task. While the constitutive relations of all elements are available in MathML, an *ordered* set of equations for the overall model being the body of a Scilab function has to be derived. Of course, for this task, information about orientations of bonds and computational causalities at power ports is needed. One of the features of Scilab useful for simulation is that it provides the most advanced mathematical solvers for systems of Ordinary Differential Equations (ODEs) or differential-algebraic Equations (DAEs). Consequently, a transformation from BGML into Scilab mathematical language could always produce a set of DAEs. For some details of both transformations see [11].

Transformation by Means of the Query Language XQuery

Another approach to a transformation of a BGML description into a target language to be mentioned in this most likely incomplete list of options is based on the use of the query language XQuery [87]. It has also been defined by the W3 Consortium

and is supported by some software tools, e.g. the free Altova XMLSpy® Home Edition. XQuery is a functional query language that uses expressions and strictly uses types. It can conveniently retrieve information from an XML document. For a depth-first search on the tree underlying an XML document, it uses XPATH. That is, all descendants of a node are visited first. A node of this tree contains a component of the XML document. XQuery can combine information from different sources and restructure it in order to create a new result. To demonstrate the usefulness of XQuery for the transformation of BGML descriptions of bond graph models, a prototype of an XQuery script has been written that enables the transformation from BGML into SIDOPS. Figure 11.63 depicts an outline of the SIDOPS description of the hierarchical sample bond graph in Figure 11.58 generated by means of this XQuery script.

As can be seen from the bond graph of Figure 11.58, the interface of submodel SM2 has three ports, while the interface of SM1 has one. These ports are called plugs in SIDOPS. Furthermore, an arrow \Rightarrow denotes an interconnection between two power ports with a reference orientation indicated by the arrow head, while an arrow \rightarrow denotes an interconnection between signal ports. Port P3 of submodel SM2 and the port of monitor1 are signal ports.

11.7.8 XML Based Bond Graph Component Model Libraries

As mentioned in Section 11.7.6, a file may only contain the BGML description of a submodel, which can be validated against the BGML schema. By consequence, not only the exchange and reuse of submodels is supported, but also the building of XML based Bond Graph component model libraries organised as an ordinary file system (For comparison, in the 20-sim software environment, model libraries are organised as directories of the Windows file system and models can be accessed by a library browser). The XML description of a submodel chosen from an XML based component model library can be added to the list of submodels of the calling *composed* submodel either when the icon of the library submodel is picked and placed onto the working space of the graphical editor, or when the *composed* submodel or the overall model is saved to disk and its XML description is generated.

For illustration, Figure 11.64 shows an outline of the BGML description of a constant hydraulic flow source library submodel, of which the bond graph is depicted in Figure 11.65. Note that the BGML description of a library submodel for internal leakage has been copied into the description of the constant displacement flow pump submodel. Both submodels can be kept in separate files being entities of an XML Bond Graph component model library. They can be validated against the BGML schema and can be passed for exchange and reuse. Finally, a librarian software can be designed that can help model developers navigate through component model features and find a file accordingly.

```

model 0 0
type Mainmodel
end;
implementation bg
submodels
Se 100 200
MONITOR1 300 200
SM2 200 200
plug p1 100 200 ;
plug p2 200 100 ;
plug p3 300 300 ;
One1 200 200
I1 300 200
R1 200 300
connections
p1 => One1\1
One1\2 => p2
One1\3 => I1\p
One1\4 => R1\p
end;
implementation_end;
SM1 200 100
plug p1 200 300 ;
Zero1 200 200
C1 100 200
R1 300 200
connections
p1 => Zero1\1
Zero1\2 => C1\p
Zero1\3 => R1\p
end;
implementation_end;
end;
connections
Se\p => SM2\p1
SM2\p2 => SM1\p1
SM2\p3 -> MONITOR1\input
end;
implementation_end;

```

Fig. 11.63 Outline of a SIDOPS description of the sample bond graph in Figure 11.58 generated by an XQuery script

```

<?xml version="1.0" encoding="UTF-8"?>
<submodel name="FlowPump" x-coord="200" y-coord="200"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\BG-XML\BGmodels4-5.xsd">
  <submodel-name>FlowPump</submodel-name>
  <submodel-version/>
  <submodel-author/>
  <submodel-author-email/>
  <submodel-description/>
  <composed-submodel>
    <ports>
      <power-port name="outlet" orientation="out" causality="fixed flow out">
      </power-port>
      <signal-detector-port name="signal-in" orientation="in">
      </signal-detector-port>
    </ports>
    <submodels>
      <!-- resistor for internal leakage -->
      <submodel name="Rleak">
        <submodel-name/>
        <submodel-version/>
        <submodel-author/>
        <submodel-author-email/>
        <submodel-description/>
        <flat-submodel type="R">
          <ports>
            <power-port name="p" orientation="in" causality="fixed flow out">
            </power-port>
          </ports>
          <parameters/>
          <initial-conditions/>
          <variables/>
          <equations/>
        </flat-submodel>
      </submodel>
    </submodels>
    <connections>
      <signal id="1" dimension="1">
        <from>signal-in</from>
        <to>MSf1\in</to>
      </signal>
      <bond id="2" dimension="1">
        <from>MSf1\p</from>
        <to>ZER01\p1</to>
      </bond>
      <bond id="2" dimension="1" flow="Q">
        <from>ZER01\p2</from>
        <to>Rleak\p</to>
      </bond>
      <bond id="3" dimension="1" flow="Q1">
        <from>ZER01\p3</from>
        <to>outlet</to>
      </bond>
    </connections>
  </composed-submodel>
</submodel>

```

Fig. 11.64 Outline of a BGML description of a constant displacement flow pump library submodel

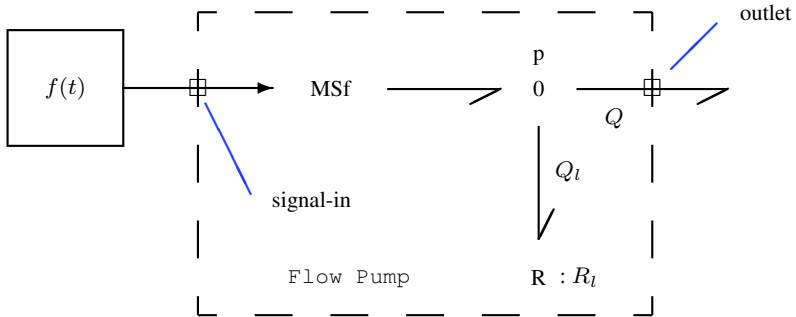


Fig. 11.65 Constant displacement flow pump

11.8 Conclusion

In this chapter, we considered simulation languages at the algorithmic level, contemporary higher level model description languages and software packages with respect to their support of bond graph based physical system modelling. The investigation has revealed the following results.

It is one option to formulate equations derived from non-hierarchical causal bond graphs in a simulation language based on the CSSL standard in order to have a general purpose simulation program process the simulation model. For small models, equations can be derived manually from the causal bond graph by following causal paths and formulated directly in a CSSL. Bond graph preprocessors and some modelling environments, e.g. MS1[®], can perform this step automatically. The MACRO feature of a descriptive CSSL and its ability to include procedures written in a programming language can be used to capture the hierarchical and modular structure of large models.

A major disadvantage of directly using a simulation language of CSSL type, however, is that equations must be already causal. What the translator of a simulation program essentially does is to sort the assignment statements into computational order. This limitation of CSSLs to causal equations hampers the development of reusable submodels and their connection according to the physical structure of an overall system. To that end, equations must be non-causal and some equations in some submodels must be reformulated in order to meet requirements due to the interconnection of submodels.

It is this disadvantage of CSSLs that has been overcome by object-oriented modelling languages. These languages support the definition and non-causal description of objects and their connection according to the way their corresponding real system components are connected. Consequently, this approach requires that equations are not only sorted into computational order, but are symbolically reformulated also where necessary. The concepts of object-oriented modelling languages provide the mechanisms that are needed for the reliable development of large hierarchical mod-

ular models. In 1967, when the CSSL standard was defined, these mechanisms did not yet exist.

The paradigm of object-oriented modelling has given rise to the introduction of a number of new languages with similar features and has led to the definition of the unified object-oriented modelling language *Modelica*. This language aims to support model reuse and exchange between (proprietary) modelling and simulation packages.

The developers of the modelling and simulation environment SYMBOLS Shakti™ argue that the use of languages supporting non-causal modelling and consequently the need for symbolic manipulation of equations is appropriate for linear components or subsystems. In order to have nonlinear component models fit into causal boundaries imposed by other submodels the component is connected to, the SYMBOLS Shakti™ developers grouped several causal versions of a submodel into an object they call capsule and implemented a mechanism that transparently for the user selects the proper version of a submodel according to the causal requirements at the ports of a component model. Clearly, more complex component models stored in libraries require capsules comprising a number of hidden causal submodel instances.

While the model description language SIDOPS has been especially designed to support bond graph based physical system modelling, Modelica could be used as well, although the language has been developed with generalised networks in mind. While engineers concerned with the design of a system prefer graphical model representations over a textual description, the use of textual modelling languages mainly serve

- as an internal description form in modelling and simulation environments,
- as an implementation form for models grouped in domain specific libraries that come with a modelling and simulation software,
- as an expected format for the exchange of models between proprietary modelling and simulation tools.

As to the support of bond graph modelling at a graphical level, there are different approaches. An obvious idea is to have a bond graph preprocessor with a graphical user interface generate a textual description in a CSSL. An early implementation of this approach has been the program CAMP-G®. It can automatically derive equations from a bond graph entered at a graphical level and output them in the simulation language ACSL. A similar approach has been the development of bond graph toolboxes for software packages supporting mathematics, e.g. Mathematica® and MATLAB®. Finally, the idea of transforming a bond graph entered at a graphical level into the textual input of a target software tool has been extended by Gawthrop. By considering a bond graph of a model as a *core* representation from which others can be derived, he provided an open set of model transformation tools.

The advantage of these approaches is that powerful features of different existing software packages can be used for bond graph modelling. For instance, Mathematica® or the Symbolic Math Toolbox™ of MATLAB® can be used for setting up transfer functions in symbolic form. To that end, a bond graph preprocessor

or a toolbox requires a proper software interface. Depending on the design of the user interface, the access to other tools can be transparent for the bond graph model developer as it is, for instance, in MS1[®].

An attractive alternative to the use of a set of different (proprietary) software tools are today's integrated modelling and simulation software environments. They cover hierarchical, non-causal, multi-domain modelling in a comprehensive manner, they support *multi-formalism* including multibond graphs and allow the user to easily switch between graphical model entry, model processing, simulation experiments and model modification. Users of these software environments have to be concerned with non-causal equations only if they want to define a new building block to be added to a model library. Generation of the simulation code uses formula manipulation where necessary. Nonlinear algebraic relations are solved numerically. In 20-sim[®], the development of models at the equation level is supported by the modelling language SIDOPS. Some of the most advanced up-to-date modelling and simulation environments have been considered in this chapter.

As a disadvantage, one might consider that users are confined to the capabilities an integrated software environment offers. For instance, in general, it is not possible to add a new advanced integration method from a mathematical library. On the other hand, most present integrated modelling and simulation environments offer a number of interfaces to other vendor's software packages, e.g. to MATLAB[®].

Beyond the graphical entry and modification of bond graph models, some software packages support modelling at a *conceptual* level. To that end, the software MAX, developed in a research framework, provides a bidirectional transformation between schematic representations using domain specific icons and bond graphs. Once a model has been designed at a conceptual level, the design engineer can navigate through a model library to find submodel implementations that meet given specifications.

With regard to simulation run times, it is advisable to start with submodels of reasonably low complexity and to account for further properties in some submodels if suggested by simulation results. The modelling software CMBAS attempts an algorithmic solution to the synthesis problem. It uses model order reduction algorithms to determine for each component a model of minimal complexity among available models of a component such that all natural frequencies of the system are within a specified frequency range. While advanced (commercial) software for hierarchical modelling and multiple description formalisms along with libraries have become state-of-the-art, it appears that software aimed at the support of the early stages of the modelling process have been more a subject of research projects. Some results of these projects have been briefly addressed in this chapter.

References

- [1] Octave. URL <http://www.gnu.org/software/octave/>.
- [2] REDUCE. URL <http://www.reduce-algebra.com/>.

- [3] Altova GmbH. URL <http://www.altova.com>.
- [4] A. Azmani and G. Dauphin-Tanguy. Archer: a program for computer aided modelling and analysis. In P. C. Breedveld and G. Dauphin-Tanguy, editors, *Bond Graphs for Engineers*, pages 263–278. Elsevier, North-Holland, 1992.
- [5] D. Ballance, G. Bevan, P.J. Gawthrop, and D. Diston. Model Transformation Tools (MTT): The Open Source Bond Graph Project. In J.J. Granda and F.E. Cellier, editors, *2005 International Conference on Bond Graph Modeling, and Simulation (ICBGM 2005)*, pages 123–128. SCS Publishing, 2005. Simulation Series, volume 37, no 1, ISBN: 1-56555-287-3.
- [6] P. Beater. HyLib – Library of hydraulic components. URL <http://www.modelon.se>.
- [7] G. Bevan. Next-Generation Transformation Tools for Scalable Integrated System Modelling. In J.J. Granda and F.E. Cellier, editors, *2005 International Conference on Bond Graph Modeling, and Simulation (ICBGM 2005)*, pages 143–148. SCS Publishing, 2005. Simulation Series, volume 37, no 1, ISBN: 1-56555-287-3.
- [8] A. Biran and M. Breiner. *MATLAB for Engineers*. Addison Wesley, 1999. URL <http://www.Mathworks.com>.
- [9] W. Borutzky. Bond Graphs and Object Oriented Modelling – A Comparison. *Proc. Instn Mech. Engrs, Part I, Journal of Systems and Control Engineering*, 216(1):21–33, 2002.
- [10] W. Borutzky. Exchange and Reuse of Bond Graph Models based on XML. In J.J. Granda and F.E. Cellier, editors, *2005 International Conference on Bond Graph Modeling, and Simulation (ICBGM 2005)*, pages 129–136. SCS Publishing, 2005. Simulation Series, volume 37, no 1, ISBN: 1-56555-287-3.
- [11] W. Borutzky. BGML – a novel XML format for the exchange and the reuse of bond graph models of engineering systems. *Simulation Modelling Practice and Theory*, 14(7):787–808, 2006.
- [12] W. Borutzky. Relations between bond graph and object-oriented physical systems modeling. In J.J. Granda and F.E. Cellier, editors, *ICBGM'99, 4th International Conference on Bond Graph Modeling and Simulation*, pages 11–17. SCS Publishing, 1999. Simulation Series, volume 31, no. 1, ISBN: 1-56555-155-9.
- [13] W. Borutzky and F.E. Cellier. Tearing algebraic loops in bond graphs. *TRANSACTIONS of the SCS*, 13(2):102–115, June 1996.
- [14] W. Borutzky, B. Barnard, and J.U. Thoma. Describing bond graph models of hydraulic components in Modelica. *Mathematics and Computers in Simulation*, 53(4–6):381–387, 2000.
- [15] B. Ould Bouamama, A.K. Samantaray, K. Medjaher, M. Staroswiecki, and G. Dauphin-Tanguy. Model builder using functional and bond graph tools for fdi design. *Control Engineering Practice*, 13(7):875–891, 2005.
- [16] A.P.J. Breunese. *Automated support in mechatronic systems modeling*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1996.
- [17] A.P.J. Breunese and J.F. Broenink. Modeling mechatronic systems using the SIDOPS+ language. In J.J. Granda and G. Dauphin-Tanguy, editors, *1997 International Conference on Bond Graph Modeling, and Simulation (ICBGM'97)*, pages 301–306. SCS Publishing, 1997. Simulation Series, volume 29, no 1, ISBN: 1-56555-103-6.
- [18] J.F. Broenink. *Computer-Aided Physical-Systems Modeling And Simulation: a bond-graph approach*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1990.
- [19] J.F. Broenink. Bond-graph modeling in Modelica. In W. Hahn, A. Lehmann, W. Borutzky, and H. Ziegler, editors, *Simulation in Industry, 9th European Simulation Symposium 1997, ESS'97*, pages 137–141, 1997. Passau, Germany.
- [20] J.F. Broenink. 20-sim software for hierarchical bond-graph/block-diagram models. *Simulation Practice and Theory*, 7(5–6):481–492, 1999. URL <http://www.20sim.com>.
- [21] F.E. Cellier. Bond graph compendium. URL www.inf.ethz/personal/fcellier/BondGraphs/bg.html.
- [22] F.E. Cellier. Hierarchical nonlinear bond graphs: a unified methodology for modeling complex physical systems. *SIMULATION*, 58(4):230–248, April 1992.
- [23] F.E. Cellier and A. Nebot. The Modelica Bond Graph Library. In *Proc. 4th International Modelica Conference*, pages 57–65, 2005. volume 1, Hamburg, Germany.

- [24] HighTech Consultants. URL www.htcinfo.com.
- [25] Controllab Products. 20-sim the power in modeling. URL <http://www.20sim.com>.
- [26] V. Damić and J. Montgomery. *Mechatronics by Bond Graphs: An Object-Oriented Approach to Modelling and Simulation*. Springer-Verlag, 2003.
- [27] T.J.A. de Vries. *Conceptual design of controlled electro-mechanical systems*. PhD thesis, Univ. of Twente, Enschede, The Netherlands, 1994.
- [28] T.J.A. de Vries, A.P.J. Breunese, and P.C. Breedveld. MAX: a mechatronic model building environment. In J. Sharpe and V. Oh, editors, *Computer Aided Conceptual Design, Proc. 1994 Lancaster International Workshop on Engineering Design CACD'94*, pages 299–318, 1994.
- [29] Dynasim. URL <http://www.Dynasim.se>.
- [30] H. Elmqvist. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1978. Report CODEN: LUTFD2/(TFRT-1015)/1-226/(1978).
- [31] H. Elmqvist, D. Brück, and M. Otter. *Dymola – User’s Manual*. Dynasim AB, Lund, Sweden, 1996. URL <http://www.Dynasim.se>.
- [32] H. Elmqvist, S.E. Mattson, and M. Otter. Modelica – the new object oriented modeling language. In *Proc. of the 12th European Simulation Multiconference, ESM'98*, 1998. Manchester, UK, June 1998.
- [33] H. Elmqvist et al. *ModelicaTM – A Unified Object-Oriented Language for Physical Systems Modeling*. URL <http://www.Modelica.org>.
- [34] J. Félez, C. Vera, I. San José, and R. Cacho. BONDYN: A Bond Graph Based Simulation Program for Multibody Systems. *Trans. ASME, Journal of Dynamic Systems, Measurement and Control*, 112:717–727, 1990.
- [35] P. Fishwick. The Art of Modeling: XML. *Modeling & Simulation*, 1(4):5, 2002.
- [36] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley & Sons, 2004. ISBN: 0-471-47163-1.
- [37] P.J. Gawthrop. MTT: Model Transformation Tools. In F.E. Cellier and J.J. Granda, editors, *ICBGM'95, International Conference on Bond Graph Modeling and Simulation*, volume 27(1) of *Simulation Series*, pages 197–202. SCS Publishing, 1995.
- [38] P.J. Gawthrop. Bicausal Bond Graphs. In F.E. Cellier and J.J. Granda, editors, *ICBGM'95, International Conference on Bond Graph Modeling and Simulation*, volume 27(1) of *Simulation Series*, pages 83–88. SCS Publishing, 1995.
- [39] P.J. Gawthrop. Hybrid Bond Graphs Using Switched I and C Components, . URL <http://www.mech.gla.ac.uk/~peterg>.
- [40] P.J. Gawthrop. MTT Home page, . URL <http://www.mech.gla.ac.uk/~peterg/software/MTT/>.
- [41] J.J. Granda. The CAMP-G/MATLAB-Simulink Computer Generated Solution of Bond Graph Derivative Causality. In J.J. Granda and F.E. Cellier, editors, *2003 International Conference on Bond Graph Modeling and Simulation (ICBGM '03)*, pages 163–171. SCS Publishing, 2003. *Simulation Series*, volume 35, no. 2, ISBN: 1-56555-257-1.
- [42] J.J. Granda. The CAMP-G/MATLAB Solution of Algebraic Loops in Dynamic Systems represented by Bond Graph Models. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 2005 International Conference on Bond Graph Modeling and Simulation (ICBGM' 05)*, volume 37(1) of *Simulation Series*, pages 115–122. SCS, 2005.
- [43] J.J. Granda. *Computer-aided modelling program (CAMP): a bond graph preprocessor for computer-aided design and simulation of physical systems using digital simulation languages*. PhD thesis, Univ. of California, Davis, 1982.
- [44] M. Hales and R.C. Rosenberg. ENPORT Model Builder: An Improved Tool for Multiport Modeling of Mechatronic Systems. In J. J. Granda and G. Dauphin-Tanguy, editors, *2001 International Conference on Bond Graph Modeling (ICBGM'01), Proc. of the 2001 Western Simulation Multiconference*, pages 152–157. SCS Publishing, January 7-11 2001. *Simulation Series*, volume 33, no. 1, ISBN: 1-56555-221-0.
- [45] A.C. Hindmarsh. ODEPACK A Systemized Collection of ODE Solvers. Preprint UCRL-88007, Lawrence Livermore National Laboratory, August 1982.

- [46] IEEE. *Electrical and electronics graphical symbols and reference designations*. IEEE, New York, U.S.A., 2nd edition edition, 1987. ISBN 471-63456-5.
- [47] D.C. Karnopp. Direct Programming of Continuous System Simulation Languages Using Bond Graph Causality. *Transactions of the Society for Computer Simulation*, 1(1):49–60, 1984.
- [48] D.C. Karnopp and R.C. Rosenberg. *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics*. MIT Press, Cambridge, MA, 1968.
- [49] F. Lorenz and P. Erhard. MS1 A multi-formalism modelling and simulation environment. In *Proc. 9th European Simulation Symposium, ESS'97*, pages 192–196, Oct. 19–22 1997. Passau.
- [50] F. Lorenz and M. Goffart. Direct Representation of Simulation Results on the Bond Graph Model. In J.J. Granda and G. Dauphin-Tanguy, editors, *1997 International Conference on Bond Graph Modeling, and Simulation (ICBGM'97)*, pages 20–25. SCS Publishing, 1997. Simulation Series, volume 29, no 1, ISBN: 1-56555-103-6.
- [51] Lorenz Simulation. MS1. URL <http://www.lorsim.be/Default.htm>.
- [52] S.E. Mattsson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal of Scientific and Statistical Computing*, 14(3):677–692, 1993.
- [53] F. Mayer. Transformation of an XML based format for Bond Graphs – Concept and implementation of a prototype of a software processor. Bachelor's thesis, Dept. of Computer Science, Bonn-Rhein-Sieg University of Applied Sciences, 2004. (in German).
- [54] J.W. Meerman. *TUTSIM on IBM PC COMPUTER User's Manual*. Meerman Automation, Postbus 154, 7160 AC Neede, The Netherlands, 1989.
- [55] W. Minten, S. Vranckx, B. De Moor, and J. Vandewalle. BondLab, a Matlab based GUI for Bond Graph Modeling. *Journal A*, 38 (3):11–15, 1997.
- [56] Modelica Association. Modelica and the Modelica Association. URL <http://www.modelica.org>.
- [57] A. Mukherjee, R. Karmakar, and A.K. Samantaray. *Bond Graph in Modeling, Simulation and Fault Identification*. I.K. International Publishing House, New Delhi, India, 2006. ISBN: 81-88237-96-5.
- [58] L.W. Nagel. *SPICE2: A computer program to simulate semiconductor circuits*. PhD thesis, University of California, Berkeley, 1975. Memorandum No. ERL-MS20.
- [59] P.J. Nolan. Symbolic and Algebraic Analysis of Bond Graphs. *Journal of the Franklin Institute*, 328(5/6):1027–1046, 1991.
- [60] M. Otter and H. Elmqvist. Energy flow modeling of mechatronic systems via object diagrams. In *2nd Mathmod Vienna, IMACS Symposium on Mathematical Modeling*, pages 705–710, 1997.
- [61] C.C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM, Journal of Scientific and Statistical Computation*, 9:213–231, 1988.
- [62] Programming Environment Laboratory, Department of Computer and Information Science, Linköping University, Sweden. The OpenModelica Project. URL <http://www.ida.liu.se/labs/pelab/modelica/OpenModelica.html>.
- [63] S. Raczynski. The PASION Simulation System. *Simulation News Europe*, 35/36:24–28, 2002. URL <http://www.raczynski.com/pn/pn.htm>.
- [64] R.C. Rosenberg. State-Space Formulation for Bond Graph Models of Multiport Systems. *Journal of Dynamic Systems, Measurement, and Control*, pages 35–40, March 1971.
- [65] R.C. Rosenberg. *A User's Guide to ENPORT-4*. Wiley-Interscience, 1974.
- [66] R.C. Rosenberg. *ENPORT-5 User's Manual*. A.H. Case Center, College of Engineering, Michigan State University, Jan. 1981.
- [67] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1991. ISBN: 0-13-630054-5.

- [68] R. Ruzicka. BAPS – Bondgraph Analyse und Programm Synthese. In J. Halin, editor, *Simulationstechnik, Informatik-Fachberichte 150*, pages 82–91. Springer-Verlag, 1987. URL <http://www.simutech.at/english/Index.html>.
- [69] P. Sahlin, A. Bring, and E.F. Sowell. The Neutral Model Format for Building Simulation. Technical report, Department of Building Sciences, The Royal Institute of Technology, Stockholm, June 1996. URL <http://www.equa.se/dncenter/nmfre302.pdf>. Version 3.02.
- [70] A.K. Samantaray. Bond Graph Modeling, Simulation and Control Software. URL <http://www.bondgraph.info/software.html>.
- [71] Scilab Consortium. Scilab. URL <http://www.scilab.org/>.
- [72] N. Smith. *Reducing the Need for Assumptions in the Automated Modelling of Physical Systems*. PhD thesis, De Montfort University, Milton Keynes, UK, 1998.
- [73] J. L. Stein and L. S. Louca. A template-based modeling approach for system design: Theory and implementation. *TRANSACTIONS of The Society for Computer Simulation International*, 13(2):87–101, 1995.
- [74] J.C. Strauss, D.C. Augustin, M.S. Fineberg, B.B. Johnson, R.N. Linebarger, and F.H. Sanson. The SCi Continuous System Simulation Language (CSSL). *SIMULATION*, pages 281–303, Dec. 1967.
- [75] R.E. Tarjan. Depth First Search and Linear Graph Algorithms. *SIAM J. of Computing*, 1: 146–160, 1972.
- [76] The AEGis Technologies Group Inc. acslX Language Reference Guide Version 2.5 March 2009. URL <http://www.acslx.com>.
- [77] J.U. Thoma. *Simulation by Bondgraphs – Introduction to a Graphical Method*. Springer-Verlag, New York, 1990.
- [78] J.U. Thoma and B. Ould Bouamama. *Modeling and Simulation in Thermal and Chemical Engineering (A Bond Graph Approach)*. Springer-Verlag, Berlin, 2000.
- [79] J.U. Thoma and G. Mocellin. *Simulation with Entropy in Engineering Thermodynamics*. Springer, Berlin, Heidelberg, New York, 2006. ISBN -10 3-540-32798-3.
- [80] M. Tiller. *Introduction to Physical Modeling with Modelica*. Kluwer Academic Publishers, Boston, MA, 2001. ISBN: 0-7923-7367-7.
- [81] J. Top. *Conceptual Modelling of Physical Systems*. PhD thesis, Twente University of Technology, Enschede, Niederlande, 1993.
- [82] J. van Dijk and P.C. Breedveld. Automated mechatronic system modelling. In R. Vichnevetsky and J.J.H. Miller, editors, *Proc. 13th IMACS World Congress on Computation and Applied Mathematics*, pages 1088–1090, 1991. Dublin, Ireland.
- [83] J.J. van Dixhoorn. Simulation of bond graphs on minicomputers. *Journal of Dynamic Systems, Measurement and Control*, pages 9–14, March 1977.
- [84] N. Venuti. Bond graph empowered by Mathematica. In J. J. Granda and G. Dauphin Tanguy, editors, *Proc. of the 2001 International Conference on Bond Graph Modeling and Simulation (ICBGM'01)*, pages 171–176, 2001.
- [85] C. Vera, F. Aparicio, J. Félez, and I. San José. Bond Graph simulation with the BONDYN program. In J.J. Granda and F.E. Cellier, editors, *International Conference on Bond Graph Modeling, ICBGM'93, Proc. of the 1993 Western Simulation Multiconference*, pages 151–159. SCS Publishing, January 17-20 1993. Simulation Series, volume 25, no. 2, ISBN: 1-56555-019-6.
- [86] M. Vergé and D. Jaume. *Modélisation structurée des systèmes avec les Bond Graphs*. Edition Technip, 2003. ISBN: 2-7108-0838-2.
- [87] The World Wide Web Consortium W3C. URL <http://www.w3.org/>.
- [88] B.H. Wilson and J.L. Stein. An algorithm for obtaining minimum-order lumped-parameter models of distributed and discrete systems. In *Proc. of the 1992 ASME Winter Annual Meeting, Symposium on Automated Modeling*, pages 47–58, Nov. 1992. Anaheim, CA., ASME Book No. G00747, New York.
- [89] Wolfram Research. Mathematica. URL <http://www.wolfram.com/>.
- [90] A. Zeid. Simple Simulation Models for Complex Systems. *Transactions of the SCS*, 6(4): 241–264, 1989.

Chapter 12

Applications

The previous chapters demonstrate the capabilities of the bond graph methodology in tackling various basic engineering problems and how various software programmes support this methodology. In this chapter, application of the methodology in various engineering fields is illustrated by considering a number of small case studies. Some of these examples have been analysed elsewhere in the literature without making use of bond graphs. Bond graph models of further examples can be found in text books on bond graph modelling published in various languages and in many research papers. As to text books, readers are referred to, e.g. [13, 17, 21, 29], just to mention a few. In most of the following case studies, the integrated modelling and simulation environment 20-sim[®][7], version 3.2, has been used. The open source mathematical software package Scilab [26] and the root finding version LSODAR of the ODE solver LSODA [14, 23] as part of Scilab has been used for the example of a clutch (Section 12.6) and for the example of a quarter vehicle (Section 12.7).

12.1 Inverted Pendulum

In feedback control of engineering systems, the inverted pendulum is often chosen as an example in order to show how a controller can be designed for stabilising an inherently unstable system. The equations of motion are usually set up directly by considering forces and moments in a free body diagram. In [13], Gawthrop and Smith gave a library model for the planar motion of a rigid rod and adapted it to the case of a simple inverted pendulum hinged to a cart. In [29], Vergé and Jaume applied the general multibond graph approach to modelling rigid multibody systems. In this section, it is shown how easily a bond graph model can be developed by graphically representing velocity constraints derived from geometric constraints. From the completed bond graph with all I energy storage elements in derivative causality, equations of motion are derived in the form of Lagrange equations of the second kind.

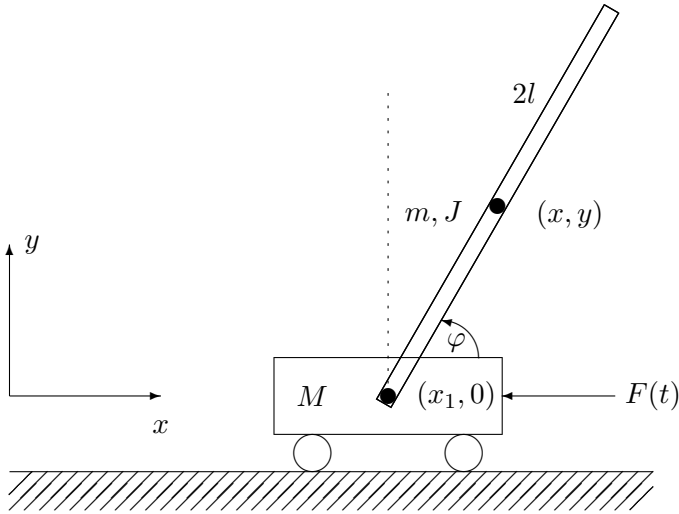


Fig. 12.1 Schematic of an inverted pendulum

Development of a Bond Graph Model

Figure 12.1 shows a schematic of the inverted pendulum with a uniform rigid rod of length $2l$, mass m , moment of inertia J about its centre of mass hinged to a rigid cart of mass M . The cart is pushed by a force $F(t)$.

The schematic provides the following two geometric relations.

$$x = x_1 + l \cos \varphi \quad (12.1a)$$

$$y = l \sin \varphi \quad (12.1b)$$

Differentiation with respect to time gives

$$\dot{x} = \dot{x}_1 - l\dot{\varphi} \sin \varphi = (-l \sin \varphi)\dot{\varphi} + \dot{x}_1 \quad (12.2a)$$

$$\dot{y} = l\dot{\varphi} \cos \varphi = (l \cos \varphi)\dot{\varphi}. \quad (12.2b)$$

The velocity constraints 12.2a–12.2b can be represented by the bond graph of Figure 12.2.

A bond graph of the inverted pendulum is obtained by simply adding effort sources, I elements and the resistor $R : b$ representing friction between the wheels of the cart and the ground. For simplicity, a linear friction characteristic is assumed. The completed bond graph is depicted in Figure 12.3.

Note that derivative causality has been assigned to all I elements. The two flow sources with a vanishing effort into the source are artificial flow sources introduced by Karnopp [16]. They resolve the causal conflicts at the 1-junctions they are attached to and indicate the generalised coordinates x_1 and φ (cf. Section 4.10).

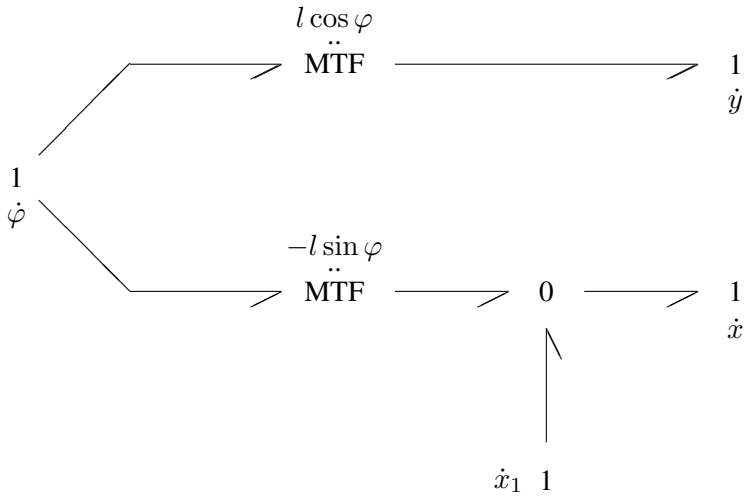


Fig. 12.2 Bond graph representation of the velocity constraints 12.2a–12.2b

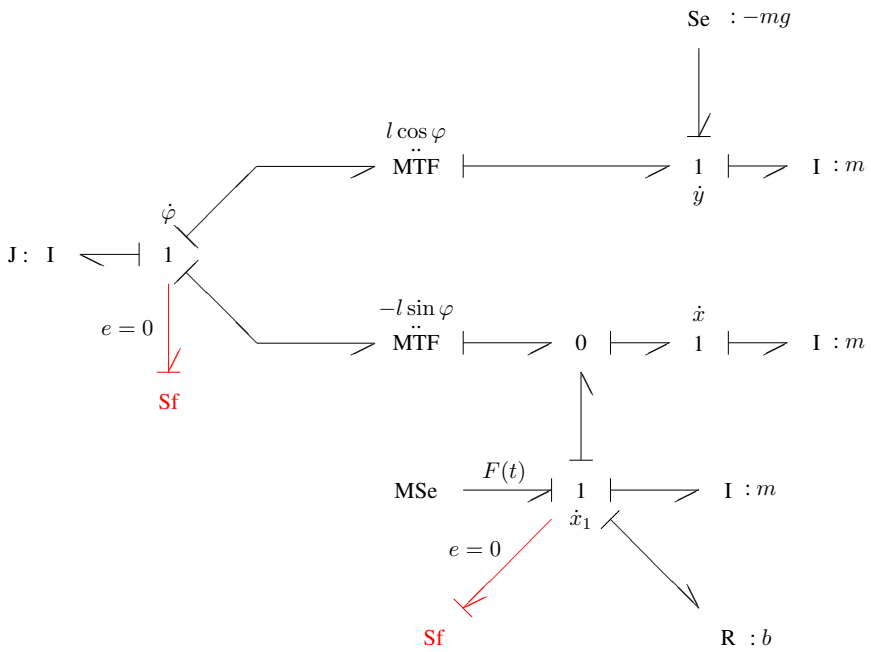


Fig. 12.3 Bond graph of the inverted pendulum

Deriving Lagrange Equations from the Bond Graph

Clearly, the system has two degrees of freedoms. Summing up all moments at the left 1-junction representing $\dot{\varphi}$ gives one of the two Lagrange equations of the second kind.

$$\begin{aligned} 0 &= J\ddot{\varphi} + l \cos \varphi (m\ddot{y} + mg) + (-l \sin \varphi) m\ddot{x} \\ &= J\ddot{\varphi} + ml^2 \cos \varphi (\ddot{\varphi} \cos \varphi - \dot{\varphi}^2 \sin \varphi) + mgl \cos \varphi \\ &\quad - ml \sin \varphi [\ddot{x}_1 - l(\ddot{\varphi} \sin \varphi + \dot{\varphi}^2 \cos \varphi)] \\ &= (J + ml^2)\ddot{\varphi} - ml \ddot{x}_1 \sin \varphi + mgl \cos \varphi \end{aligned} \quad (12.3)$$

Summation of all forces at the 1-junction representing \dot{x}_1 gives the Lagrange equation for the second degree of freedom.

$$\begin{aligned} 0 &= F(t) - M\ddot{x}_1 - m\ddot{x} - b\dot{x}_1 \\ &= F(t) - M\ddot{x}_1 - m[\ddot{x}_1 - (\ddot{\varphi} \sin \varphi + \dot{\varphi}^2 \cos \varphi)] - b\dot{x}_1 \\ F(t) - b\dot{x}_1 &= (M + m)\ddot{x}_1 - ml \ddot{\varphi} \sin \varphi - ml \dot{\varphi}^2 \cos \varphi \end{aligned} \quad (12.4)$$

Deriving Equations of Motion from the Lagrangian

For comparison, the equations of motion 12.3–12.4 shall be deduced also from the Lagrangian of the inverted pendulum. As the system has two degrees of freedom, the following equations are to be formed

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_1} \right) - \frac{\partial L}{\partial x_1} = F(t) - b\dot{x}_1 \quad (12.5a)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}} \right) - \frac{\partial L}{\partial \varphi} = 0, \quad (12.5b)$$

where $L := T - V$ is the difference of the kinetic energy, T , and the potential energy V .

The kinetic energy is

$$\begin{aligned} T &= \frac{1}{2}M \dot{x}_1^2 + \frac{1}{2}m [(\dot{x}_1 - l\dot{\varphi} \sin \varphi)^2 + (l\dot{\varphi} \cos \varphi)^2] + \frac{1}{2}J \dot{\varphi}^2 \\ &= \frac{1}{2}(M + m) \dot{x}_1^2 + \frac{1}{2}m (-2l \dot{x}_1 \dot{\varphi} \sin \varphi + l^2 \dot{\varphi}^2) + \frac{1}{2}J \dot{\varphi}^2. \end{aligned} \quad (12.6)$$

The potential energy is

$$V = -mgl \sin \varphi. \quad (12.7)$$

Hence,

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_1} \right) &= \frac{d}{dt} [(M + m) \dot{x}_1 - ml \dot{\varphi} \sin \varphi] \\ &= (M + m) \ddot{x}_1 - ml(\ddot{\varphi} \sin \varphi + \dot{\varphi}^2 \cos \varphi) \end{aligned} \quad (12.8)$$

and

$$\frac{\partial L}{\partial x_1} = 0. \quad (12.9)$$

Substitution of Equations 12.8 and 12.9 into Equation 12.5a gives Equation 12.4.

Furthermore,

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}} \right) &= \frac{d}{dt} [-mgl \dot{x}_1 \sin \varphi + ml^2 \dot{\varphi} + J \dot{\varphi}] \\ &= (J + ml^2) \ddot{\varphi} - ml(\ddot{x}_1 \sin \varphi + \dot{x}_1 \dot{\varphi} \cos \varphi) \end{aligned} \quad (12.10)$$

and

$$\frac{\partial L}{\partial \varphi} = -ml \dot{x}_1 \dot{\varphi} \cos \varphi - mgl \cos \varphi. \quad (12.11)$$

Finally, substitution of Equations 12.10 and 12.11 into Equation 12.5b gives Equation 12.3.

Transfer Function of the Inverted Pendulum

Now, let $\varphi := \pi/2 + \phi$. For small ϕ , viz. small deviations from the vertical position of the rod, the equations of motion 12.3–12.4 can be linearised.

$$(J + ml^2) \ddot{\phi} - ml \ddot{x}_1 - mgl \phi = 0 \quad (12.12a)$$

$$(M + m) \ddot{x}_1 - ml \ddot{\phi} = F(t) - b \dot{x}_1 \quad (12.12b)$$

Laplace transform of the linearised equations of motions gives the transfer function

$$\frac{\mathcal{L}\phi}{\mathcal{L}F} = \frac{mls^2}{(\tilde{M}\tilde{J} - m^2l^2)s^4 + b\tilde{J}s^3 - mgl\tilde{M}s^2 - bmgls}, \quad (12.13)$$

where $\tilde{M} := M + m$ and $\tilde{J} := J + ml^2$.

As can be seen from the transfer function of Equation 12.13, there is a pole-zero cancellation at the origin. Furthermore, if the parameter values of Table 12.1 [12] are used, the transfer function has one positive real pole, $p_1 = 5.087$, in the right-half of the s-plane. This is in agreement with the fact that the open-loop system is unstable. Figure 12.4 shows a root locus plot and Figure 12.5 depicts a Nyquist plot of the uncontrolled cart-pendulum system.

Table 12.1 Parameters of the inverted pendulum

Parameter	Value	Units	Meaning
M	0.7429	kg	Cart mass
m	0.21	kg	Mass of the rigid rod
$2l$	0.61	m	Pendulum length
b	7.19	Ns/m	Friction between cart wheels and ground

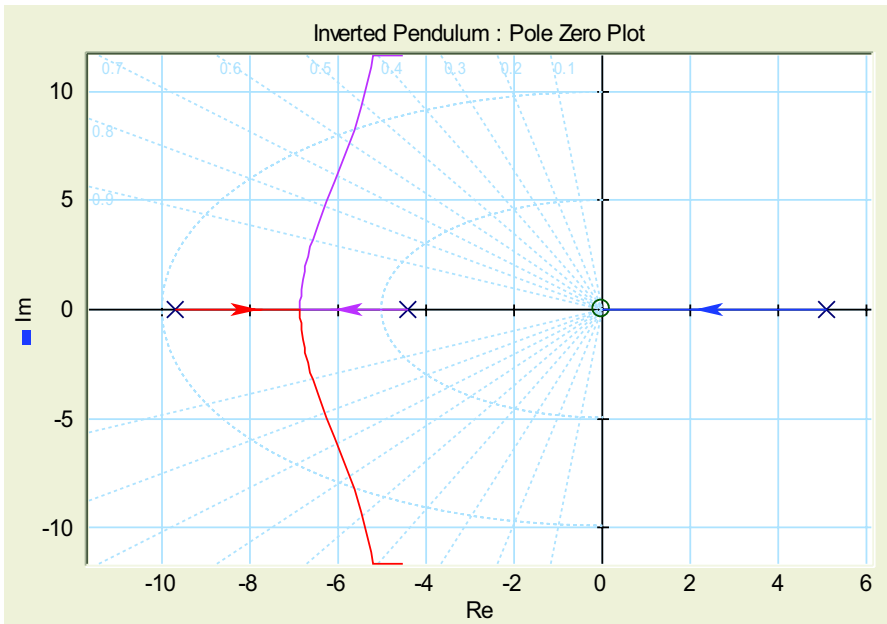


Fig. 12.4 Root locus plot of the uncontrolled cart-pendulum system

If $G(s)$ denotes the open-loop transfer function and $H(s)$ the transfer function of the feedback component then according to the Nyquist stability criterion, $H(s)$ has to be designed so that the Nyquist plot of $G(s)H(s)$ has one anti-clockwise encirclement of the point $-1 + j 0$. As a result, the number of unstable closed-loop poles is zero. As the purpose of this section has been the development of a bond graph model of the inverted pendulum and the derivation of Lagrange equations of motion from the bond graph, the design of a controller for stabilisation of the unstable inverted pendulum is not considered. The control of the unstable inverted pendulum example has been addressed, for instance, in the textbook of F. Brown [3] (Guided Problem 8.2) and in the online tutorials authored by B. Messner and D. Tilbury [19].

12.2 Shunt Motor

In Chapter 11, a bond graph model of a shunt motor has been used as a reference example for illustration of various aspects. In this section, a small simulation study is carried out to determine the motor's dynamic response to a sudden increase of the load torque. For convenience, the schematic and the bond graph model are reproduced from Chapter 11 (Figures 12.6 and 12.7).

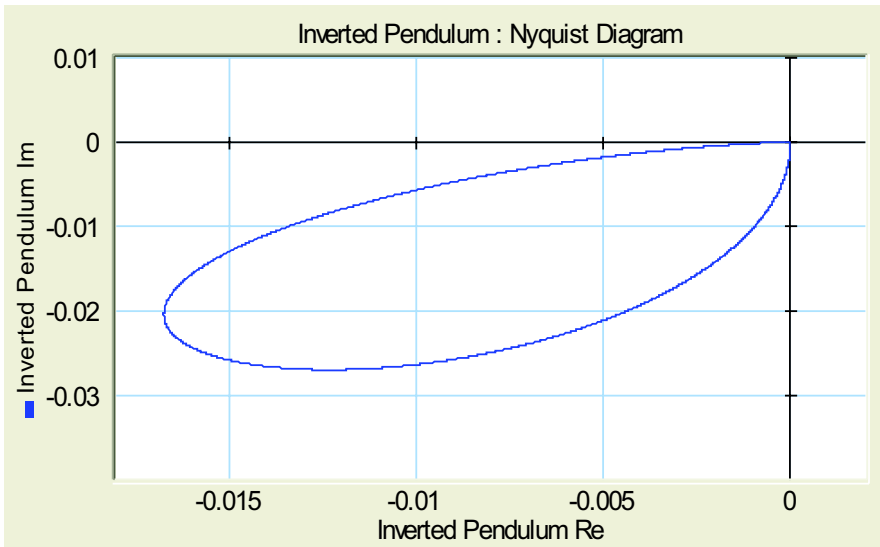


Fig. 12.5 Nyquist plot of the uncontrolled cart-pendulum system

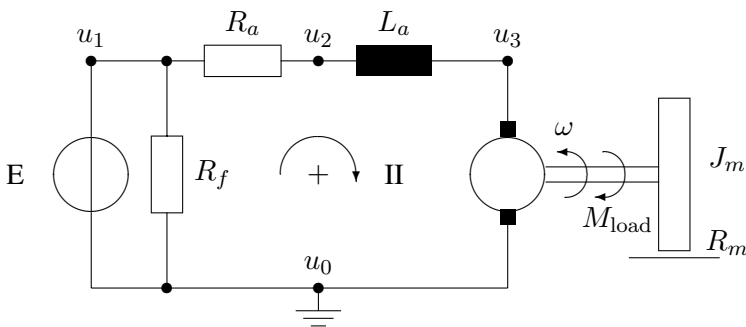


Fig. 12.6 Schematic of a shunt motor

From the causal bond graph in Figure 12.7, the following equations can be derived.

$$i_f = \frac{1}{R_f} E \tag{12.14a}$$

$$u_R = R_a \times i_a \tag{12.14b}$$

$$M_R = R_m \times \omega \tag{12.14c}$$

$$\Psi = K \times i_f \tag{12.14d}$$

$$u_a = \Psi \times \omega \tag{12.14e}$$

$$M = \Psi \times i_a \tag{12.14f}$$

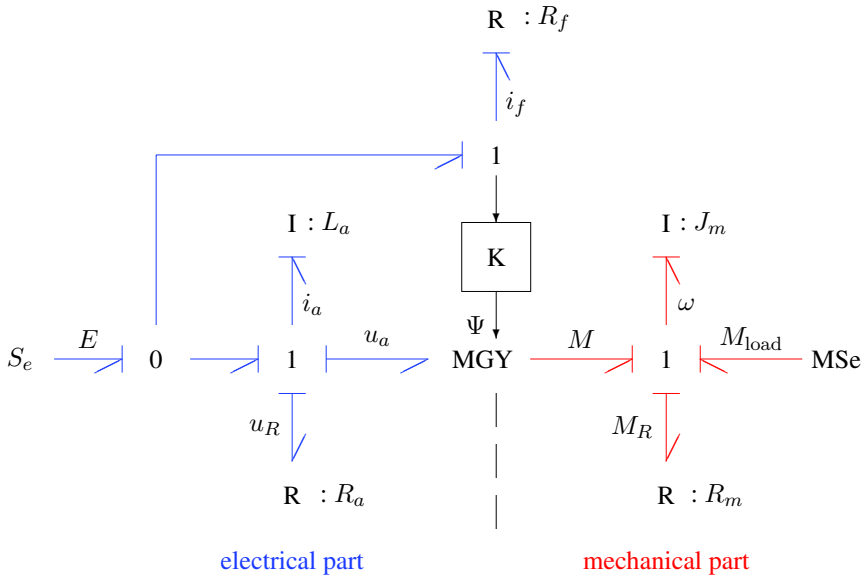


Fig. 12.7 Bond graph model of a shunt motor

$$\frac{di_a}{dt} = \frac{1}{L_a} (E - u_a - u_R) \tag{12.14g}$$

$$\frac{d\omega}{dt} = \frac{1}{J_m} (M - M_R + M_{load}) \tag{12.14h}$$

For the simulation study, it is assumed that the motor is driven by a constant voltage source. Its value is $E = 220\text{ V}$. First, the motor’s idling performance is analysed. Then, after 2.5 s , the motor is subjected to an immediate jump of the load torque to a constant value of 100 Nm . The parameters used for the simulation are listed in Table 12.2.

Table 12.2 Parameters of the simulation study

Parameter	Value	Units	Meaning
E	220	V	Voltage supply
R_f	5.495	Ω	Resistance of the field winding
R_a	0.875	Ω	Resistance of the armature winding
R_m	0.066	$Nm.s$	Friction coefficient
K	0.0307	$V.s/A$	$\Psi = K \times i_f$
L_a	0.175	H	Self-inductance of the armature winding
J_m	0.8	$Nm.s^2$	Moment of inertia of the flywheel
M_{load}	100	Nm	Load torque effective for $t \geq 2.5\text{ s}$

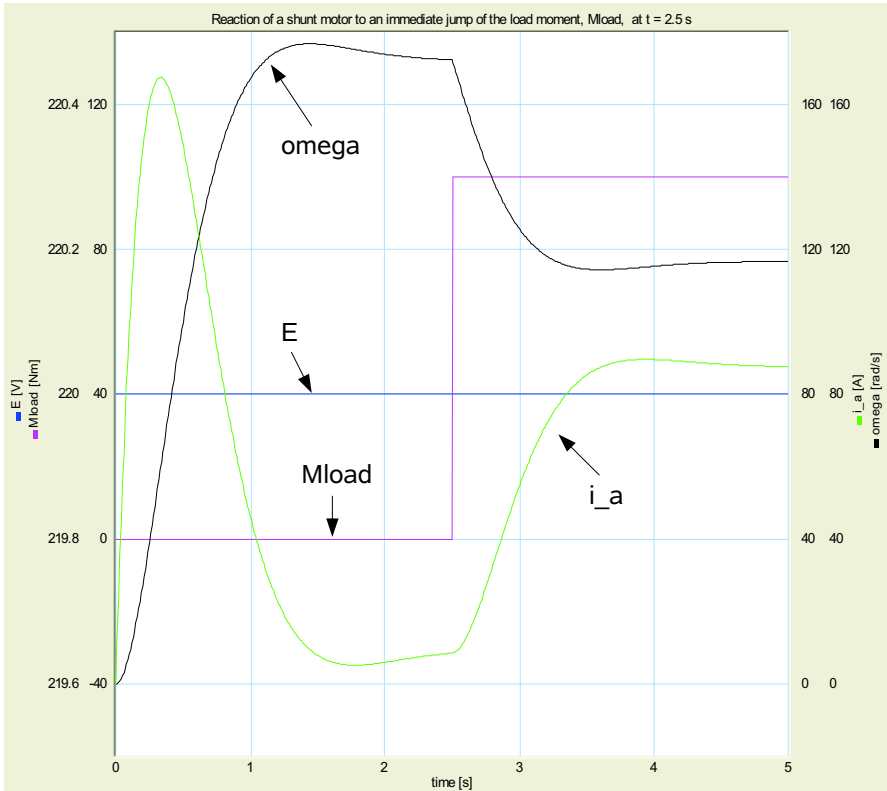


Fig. 12.8 Time evolution of the shaft velocity and the armature current

Simulation Results

Figure 12.8 shows the time evolution of the angular velocity, ω , of the motor shaft and of the current, i_a , through the armature winding. After switching on the voltage supply of the motor, the angular velocity rises and approaches a steady idle speed value. During the rise time of the angular velocity, the current consumption reaches a maximum value and peaks off to low values when the angular velocity is around its steady state value.

Some algebra on the dynamic equations results in the following formulae for the steady state values i_a^0 and ω^0 .

$$i_a^0 = \frac{R_f}{K E} R_m \omega^0 \tag{12.15a}$$

$$E = \left(\frac{K E}{R_f} + R_a \frac{R_f}{K E} R_m \right) \omega^0 \tag{12.15b}$$

With the parameters from Table 12.2, i_a^0 and ω^0 take the numerical values

$$i_a^0 = 9.257 \text{ A} \quad (12.16a)$$

$$\omega^0 = 172.4 \text{ 1/s} , \quad (12.16b)$$

which verifies the values obtained by simulation.

When the constant load torque becomes effective at $t = 2.5 \text{ s}$, then the angular velocity drops and the current consumption rise to new steady state values, as to be expected.

12.3 A Machine with an Unbalanced Rotor

An unbalanced mass has a feedback on the motor of a shaft-driven machine and can cause unwanted vibrations and noise of a machine mounted on springs. A motor mounted on springs as sketched in Figure 12.9 has been analysed by Christ in his dissertation in as early as 1966 [6].

As the differential equations are nonlinear, Hoffmann has chosen this example for a MATLAB[®]/Simulink[®] simulation that starts from given differential equations [15]. Such a spring mounted vertically moving machine is also briefly considered in the textbook by Brown ([3], Example 6.8), where a very simple bond graph is given.

Development of a Bond Graph Model

Similar to the case of the previous example of an inverted pendulum, in this section, the development of a bond graph model starts from considering the position of the unbalanced mass. Differentiation of its coordinates in a global frame with respect to time, again, provides velocity constraints that can be represented in a bond graph fragment. This bond graph for the kinematic constraints can be easily extended into a full dynamic model from which the equations of motion can be derived.

In a global frame, the position of the unbalanced mass, m_2 , has the coordinates

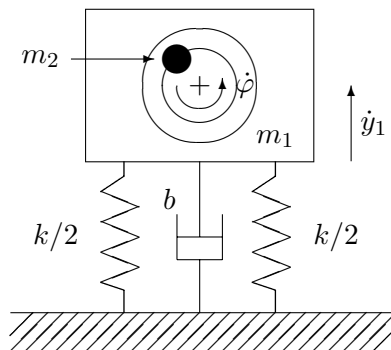


Fig. 12.9 Schematic of a machine with an unbalanced rotor

$$x_2 = 0 + e \sin \varphi \tag{12.3a}$$

$$y_2 = y_1 + e \cos \varphi . \tag{12.3b}$$

Differentiation with respect to time gives the velocity constraints

$$\dot{x}_2 = (e \cos \varphi) \dot{\varphi} \tag{12.4a}$$

$$\dot{y}_2 = \dot{y}_1 + (-e \sin \varphi) \dot{\varphi} . \tag{12.4b}$$

As for the rod of the inverted pendulum, the velocity constraints 12.4a–12.4b can be represented by the bond graph of Figure 12.10.

The bond graph fragment of Figure 12.10 is extended into a dynamic model of the machine by adding inertia elements to the 1-junctions, a C and an R element for the spring-damper pair and effort sources accounting for gravitational forces. Figure 12.11 shows the resulting bond graph. The machine is driven by a DC motor with constant excitation providing a torque M_m . A bond graph of the motor is depicted in Figure 12.12.

Derivation of Model Equations from the Bond Graph

Summation of all forces at the 1-junction representing \dot{y}_1 gives

$$\begin{aligned} m_1 \ddot{y}_1 &= -m_1 g - b \dot{y}_1 - k y_1 - (m_2 g + m_2 \ddot{y}_2) \\ m_1 \ddot{y}_1 + b \dot{y}_1 + k y_1 &= -m_2 \ddot{y}_2 - (m_1 + m_2) g . \end{aligned} \tag{12.5}$$

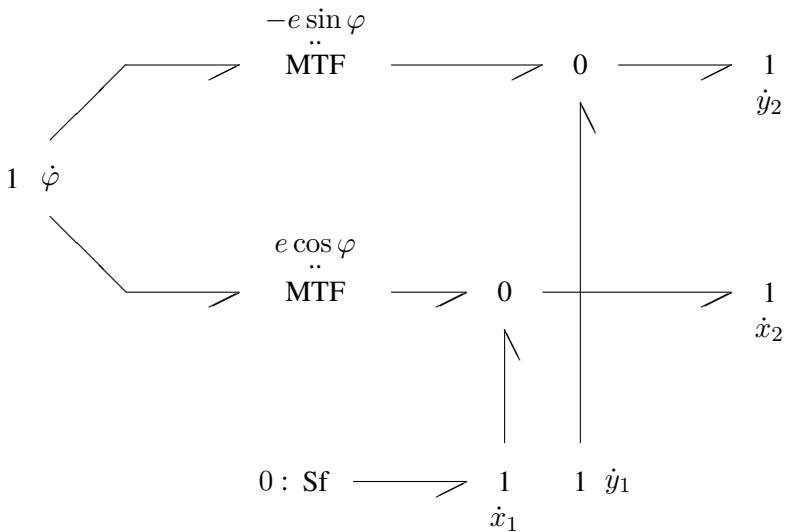


Fig. 12.10 Bond graph representation of the velocity constraints 12.4a–12.4b

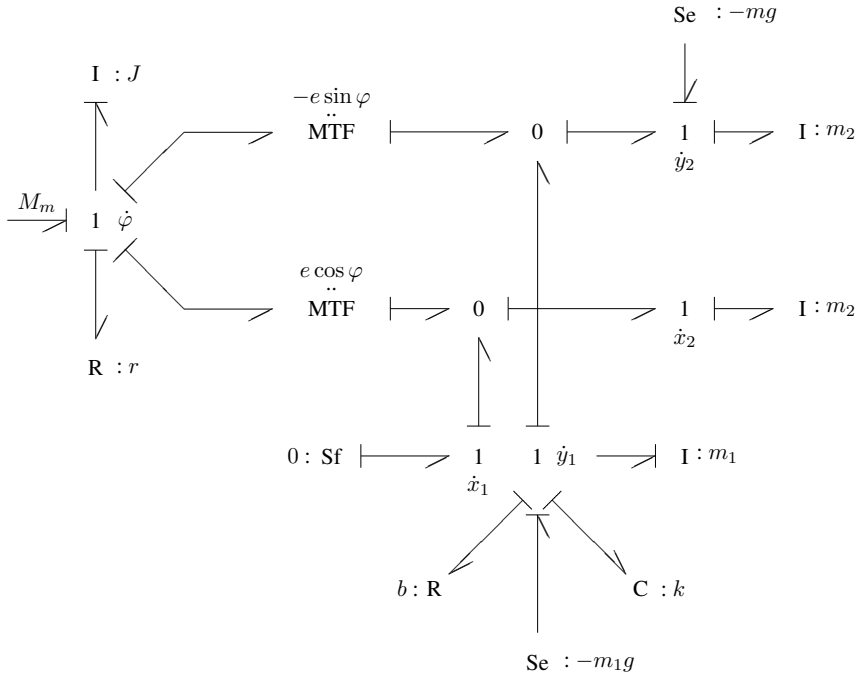


Fig. 12.11 Bond graph of the machine with an unbalanced mass

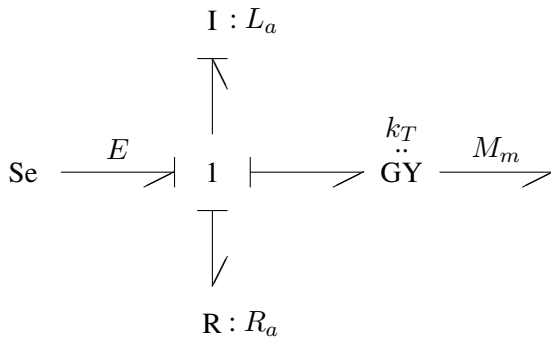


Fig. 12.12 Bond graph of the DC motor driving the machine with an unbalanced mass

The sum of flows at the upper 0-junction reads

$$\dot{y}_2 = (-e \sin \varphi) \dot{\varphi} + \dot{y}_1 . \quad (12.6)$$

The derivative causality of the I element attached to the 1-junction of \dot{y}_2 requires differentiation of Equation 12.6. Substitution of the result into Equation 12.5 yields the equation for the vertical motion of the machine.

$$(m_1 + m_2) \ddot{y}_1 + b\dot{y}_1 + ky_1 = m_2 e (\varphi^2 \cos \varphi + \ddot{\varphi} \sin \varphi) - (m_1 + m_2)g \quad (12.7)$$

If the system is at rest, then the initial position is

$$y_{10} = -\frac{(m_1 + m_2)g}{k} . \quad (12.8)$$

If there is no unbalance with eccentricity e , then natural frequency of the undamped oscillation is

$$\omega_0 = \sqrt{\frac{k}{m_1 + m_2}} . \quad (12.9)$$

The unbalanced mass m_2 with the eccentricity e causes the vertical excitation force

$$F_e = m_2 e (\varphi^2 \cos \varphi + \ddot{\varphi} \sin \varphi) . \quad (12.10)$$

Summation of all moments at the 1-junction representing $\dot{\varphi}$ yields

$$M_m = J \ddot{\varphi} + r \dot{\varphi} + (-e \sin \varphi)(m_2 \ddot{y}_2 + m_2 g) + (e \cos \varphi) m_2 \ddot{x}_2 . \quad (12.11)$$

Again, summation of all flows at the 0-junctions yields \dot{y}_2 and \dot{x}_2 . Derivative causality at both right-hand side I elements of mass m_2 requires differentiation with respect to time. After substitution of these time derivatives, the balance of moments reads

$$M_m = (J + m_2 e^2) \ddot{\varphi} + r \dot{\varphi} - m_2 e (\ddot{y}_1 + g) \sin \varphi . \quad (12.12)$$

From the bond graph of the DC motor in Figure 12.12, the following two equations are derived.

$$M_m = k_T i_a \quad (12.13a)$$

$$E = L_a \frac{di_a}{dt} + R i_a + k_T \dot{\varphi} \quad (12.13b)$$

Table 12.3 Parameters of the machine with an unbalanced mass

Parameter	Value	Units	Meaning
m_1	80	kg	Mass of the bed and the rotor
m_2	20	kg	Unbalanced mass
k	1600	N/m	Spring stiffness
b	4	Ns/m	Friction coefficient for translational motion
r	100	Nms	Friction coefficient for rotation
J	9.8	Nms ²	Moment of inertia
e	0.1	m	Eccentricity of m_2
L_a	1.0	H	Inductance of the armature winding
R_a	0.1	Ω	Resistance of the armature winding
k_T	50	Nm/A	Torque constant of the motor
E	100	V	Voltage applied to the motor

Simulation of the Machine with an Unbalanced Rotor

For simulation, the parameter values in Table 12.3 [15] have been used. If there is no eccentricity, then the rotor speed in steady state, ω_m , is

$$\omega_m = E / (R_a \frac{r}{k_T} + k_T) . \quad (12.14)$$

With the given parameter values, the angular velocity takes the value $\omega_m = 1.992$ rad/s. The vertical vibration due to the unbalanced mass causes the angular velocity of the motor to oscillate around this mean value, as Figure 12.13 shows.

Furthermore, in case there is no eccentricity, the torque provided by the motor in steady state is

$$M_m = r \times \omega_m . \quad (12.15)$$

This steady state value is $M_m = 199.2$ Nm. As ω_m oscillates around a mean value, so does the motor torque due to the vibration of the spring-mass system caused by the unbalanced mass m_2 (cf. Figure 12.14). Figure 12.15 shows the time evolution of the vertical oscillation, y_1 , of the bed due to the unbalance.

Frequency Analysis of the Machine with an Unbalanced Rotor

A FFT of the time evolution of y_1 (cf. Figure 12.16) reveals that the vertical excitation force at mean angular frequency, $\omega_m = 1.992$ rad/s, stimulates the resonant frequency, $\omega_0 = 4$ rad/s of the undamped spring-mass system, which is undesired.

If y_1 is replaced by $y_1 + y_{10}$ in Equation 12.7, then, by observing the expression for the excitation force F_e , Equation 12.10, the Laplace transform of Equation 12.7 gives

$$\frac{\mathcal{L}y_1}{\mathcal{L}F_e} = \frac{1}{(m_1 + m_2)s^2 + bs + k} \quad (12.16)$$

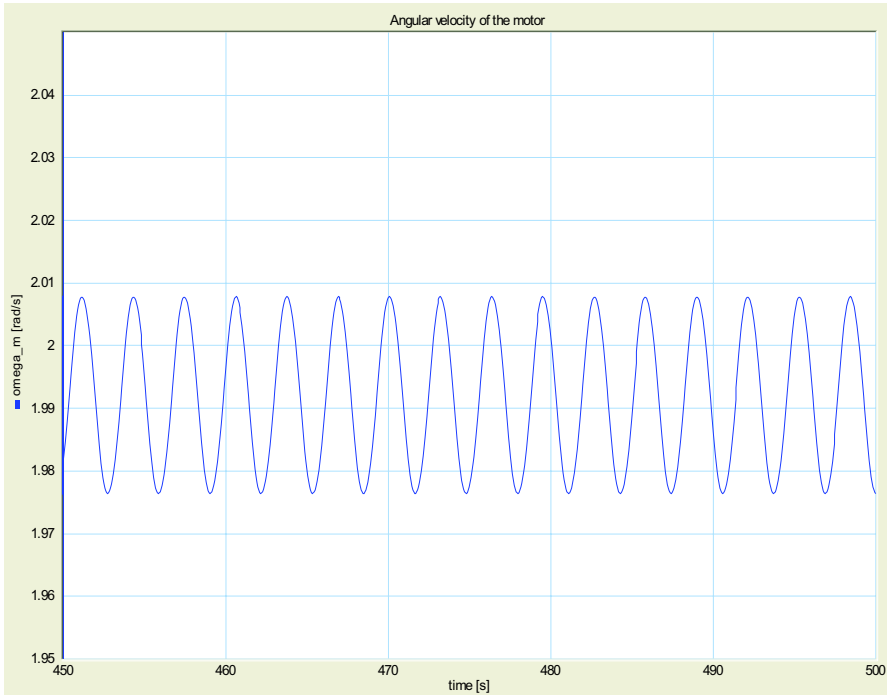


Fig. 12.13 Angular velocity of the motor

in case of a constant excitation frequency $\dot{\varphi} = \omega_m = \text{const.}$

Given the parameter values of Table 12.3, Figure 12.17 depicts the Bode plot of the transfer function of Equation 12.16. As can be seen, for excitation frequencies well above the natural frequency of the spring-mass system, the amplitude of the vertical oscillation rapidly decreases and the phase takes the constant value of -180° degrees due to the system's inertia.

Reducing the value of the stiffness k of the spring supporting the bed from 1600 N/m to 160 N/m results in a natural frequency, $\omega_0 = 1.25$ rad/s of the undamped spring-mass system that is below the mean excitation frequency of the motor. Figure 12.18 shows the oscillation of the bed in case of a reduced spring stiffness. A FFT of the time evolution of y_1 shows that in steady state, the frequency of the vertical oscillation adapts to the excitation frequency (cf. Figure 12.19).

12.4 An Electronic Balance with Displacement Compensation

Figure 12.20 depicts a conceptual schematic of an electronic balance with displacement compensation. If there is no load on the weighing scale, then the gravitational

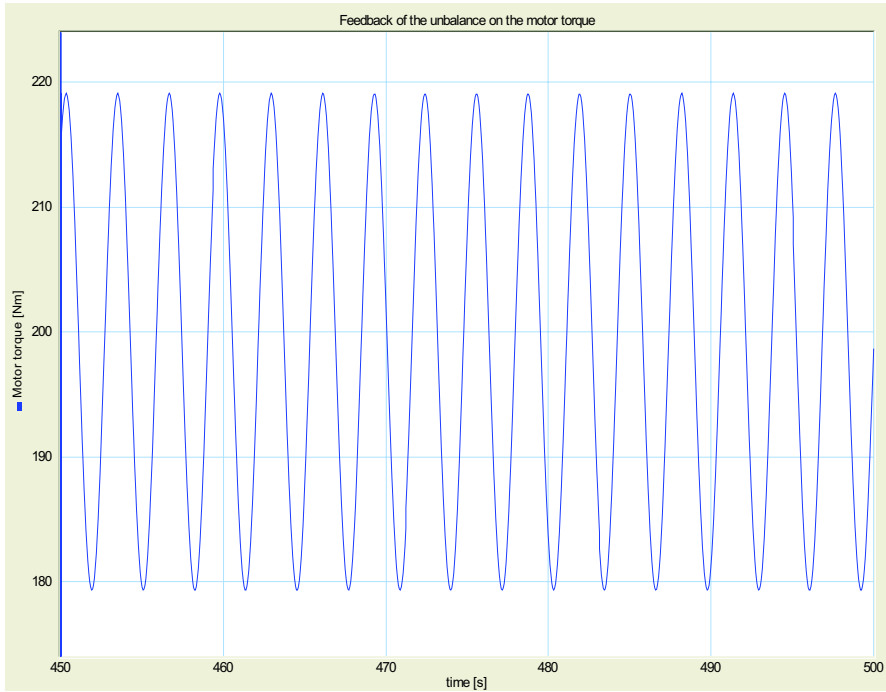


Fig. 12.14 Feedback of the unbalance on the motor torque

force of the scale pan and the spring force will be in equilibrium, defining the set point of the weighing scale's displacement set point $y = 0$. No voltage, E , is applied to the plunger coil and no current is flowing through the coil. Now, a load, m , causes a deviation from this set point. A current through the plunger coil causes an electromagnetic force that lifts the scale pan back into its initial position. The current needed to generate the electromagnetic force for compensation of the scale pan's displacement or the voltage applied to the coil can serve as a measure of the load. The electromechanical energy conversion in the coil can be represented by a gyrator.

Development of a Bond Graph Model

Construction of a bond graph model inspired by the topology of the schematic in Figure 12.20 is straightforward. Figure 12.21 shows the result. The lower 1-junction represents the current through the plunger coil. The attached R and I element account for its resistance and self-inductance. The upper 1-junction represents the velocity of load and scale pan against a spring and a damper force of the suspension. Finally, the Se source accounts for the gravitational force of the total mechanical load.

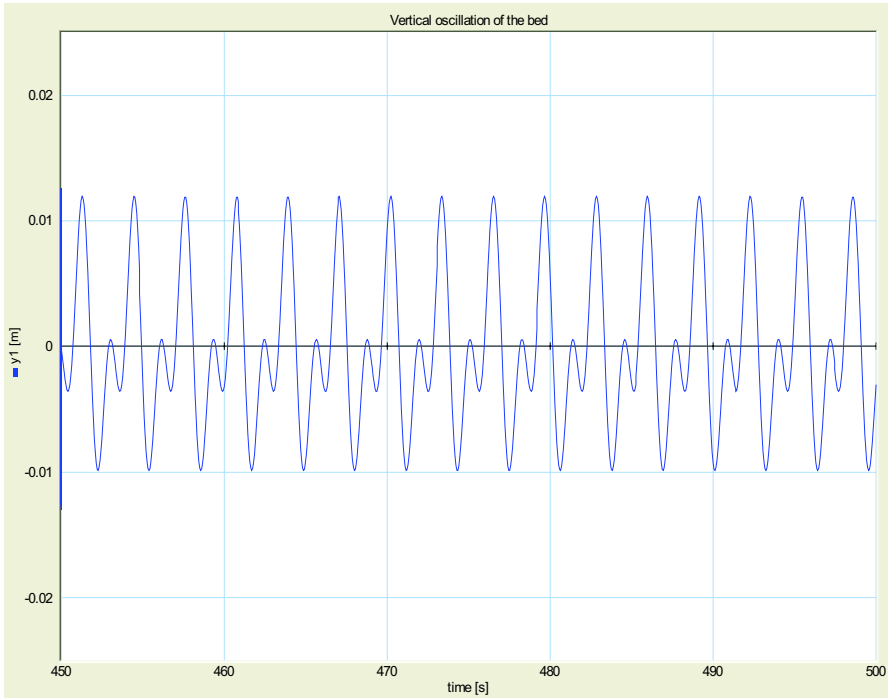


Fig. 12.15 Vertical oscillation y_1 of the bed due to the unbalance

The velocity of the load is sensed and integrated. The deviation from the set point is input into a controller. The output signal of the controller is fed into an amplifier with saturation. The amplifier's output is a voltage that is applied to the coil.

The sum of all efforts at the upper 1-junction yields the equation of motion for the mechanical part of the scale with respect to the equilibrium position.

$$-mg + Ti = (m + m_b)\ddot{y} + r\dot{y} + ky \quad (12.17)$$

The sum of all efforts at the lower 1-junction results in an equation for the dynamic behaviour of the electrical part.

$$E(t) = L \frac{di}{dt} + Ri + Tj \quad (12.18)$$

The rules for assigning half arrows to the bonds of a bond graph help to ensure that signs in the model equations derived from the bond graph are consistent. This consistency is not automatically ensured if a free body diagram is used for the mechanical part of an electromechanical system and a conventional network for the electrical part. If a load m is put on the scale at some time, t , then, in steady state, the voltage

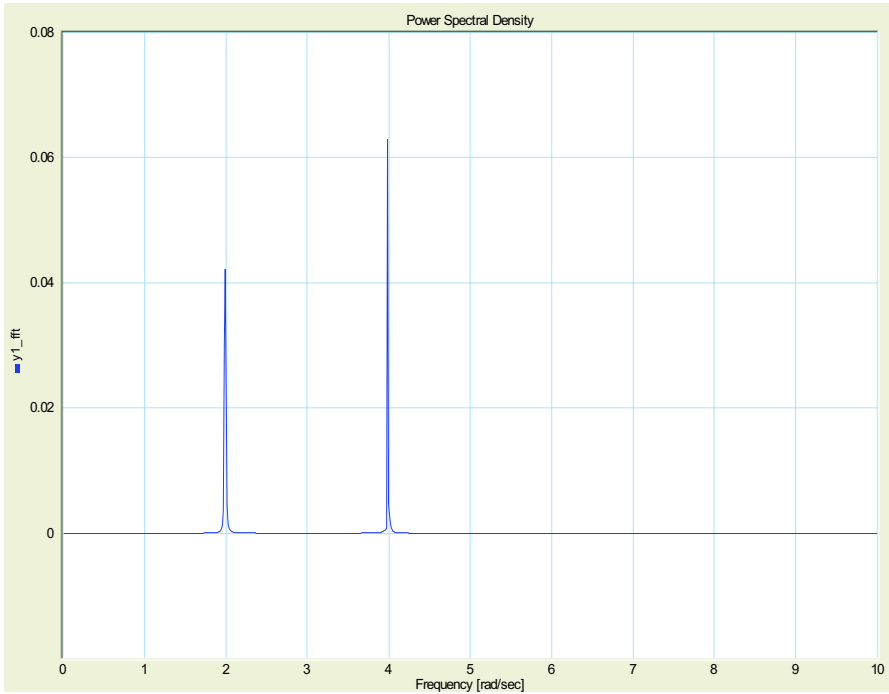


Fig. 12.16 Power spectral density of the vertical oscillation y_1

$$E_s = R \frac{mg}{T} \quad (12.19)$$

is needed to generate an electromagnetic force that compensates for the scale pan's displacement,

$$y_s = -\frac{mg}{k} . \quad (12.20)$$

In order to achieve a compensation of the scale's displacement, a PID controller is chosen.

Finally, putting a load on the scale at some time means that the mass of the scale pan is instantaneously increased. However, a rigid body with a time varying mass cannot be represented by a bond graph I element. That is, the I element in Figure 12.21 has to be replaced by a functional block implementing an equation of the form

$$p.f = \text{int}(p.e) / (mb + m * \text{step}(tstart)) . \quad (12.21)$$

In this equation, $p.f$ and $p.e$ denote the effort and flow variables of port p . The effort is integrated by the function `int` and the function `step` produces a unity step at time $tstart$. The mass of the scale pan is mb which is increased by the mass m of the load.

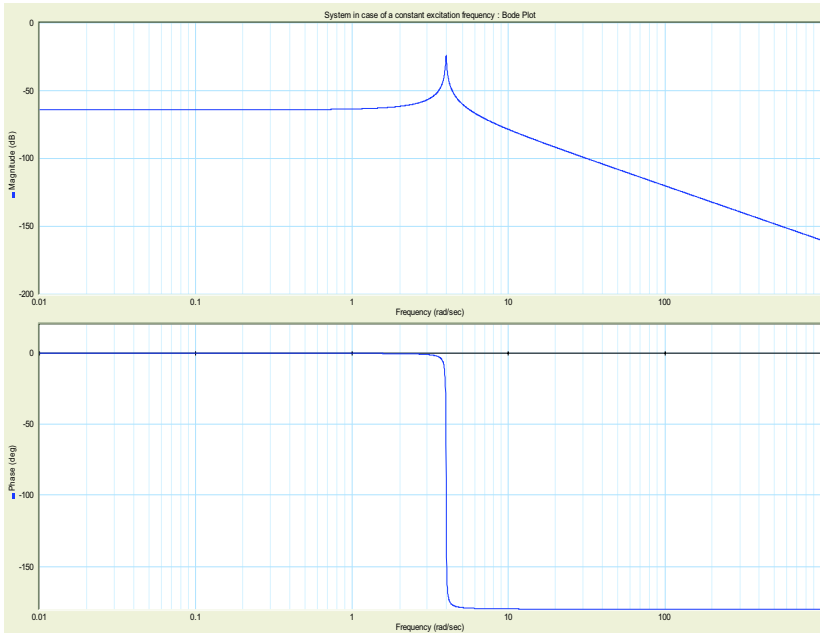


Fig. 12.17 Bode plot of the vertical oscillation case of a constant excitation frequency

Simulation of the Electronic Balance with Displacement Compensation

For simulation of the dynamic behaviour, the parameters given in Table 12.4 have been used [25].

The transfer function of the PID controller is used in the form

$$U(s) = K \left[1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{N} s} \right] E(s), \quad (12.22)$$

Table 12.4 Parameters of the controlled balance

Parameter	Value	Units	Meaning
R	1	Ω	Resistance of the coil
L	20	mH	Self-inductance of the coil
T	5	Vs	Transductance of the coil
k	1500	N/m	Stiffness of the spring
r	8.5	Ns/m	Damping coefficient
mb	0.03	kg	Mass of the scale pan
m	0.05	kg	Mass of the load

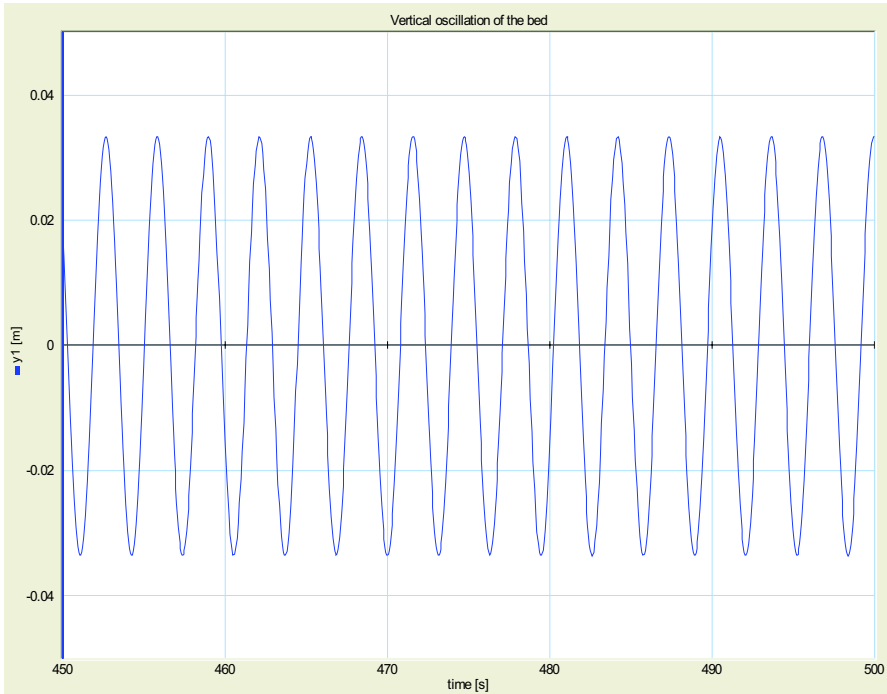


Fig. 12.18 Vertical oscillation y_1 of the bed in case of reduced spring stiffness

where $s \in \mathbb{C}$. In Equation 12.22, E and U denote the Laplace transforms of input and output. As can be seen, the derivative part is approximated. Table 12.5 lists the controller parameters.

Given the parameters in Table 12.4, the steady state value of the voltage needed to compensate the scale’s displacement according to Equation 12.19 is

$$E_s = 1 \times \frac{0.05 \times 9.81}{5} = 98.1 \text{ mV} . \tag{12.23}$$

If this voltage is not applied to the coil, then the load of $m = 0.05 \text{ kg}$ would cause a displacement of

Table 12.5 Parameters of the PID controller

Parameter	Value	Units	Meaning
K	400		Proportional gain
Ti	0.05	s	Integral time constant
Td	0.01	s	Derivative time constant
N	20		Derivative gain limitation

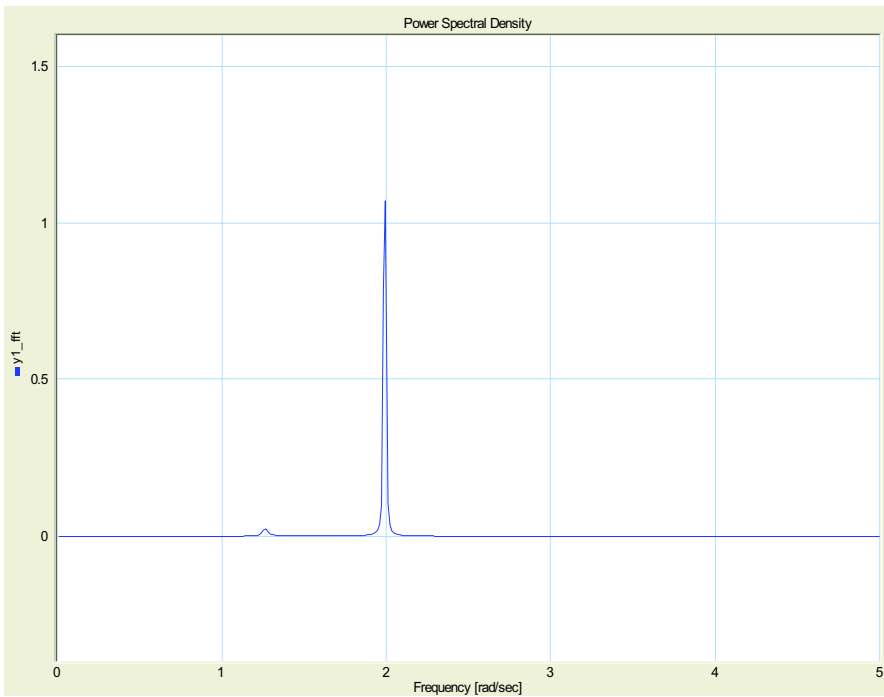


Fig. 12.19 Power spectral density of the vertical oscillation y_1 in case of reduced spring stiffness

$$y_s = -\frac{0.05 \times 9.81}{1500} = 0.33 \text{ mm} \quad (12.24)$$

according to Equation 12.20.

Figure 12.22 shows the time evolution of the current and the scale's displacement due to an instantaneous increase of the load from 0 to 0.05 kg at $t = 0.2 \text{ s}$. As can be seen from Figure 12.22, the displacement, in fact, is compensated within about 0.4 s . When the mechanical load jumps from 0.03 kg to 0.08 kg at $t = 0.2 \text{ s}$, then the current just starts from 0 A , and the scale pan is lowered by about 0.2 mm . When this displacement is reduced to zero at about $t = 0.6 \text{ s}$, the current actually reaches the value of 98.1 mA , which is necessary for generating the electromagnetic force that compensates the gravitational force of the load.

As an instantaneous increase in the load does not only mean a disturbance of the gravitational load force but also the movement of a heavier body, the dynamic behaviour during compensation is different for different loads with respect to frequency and damping. Figure 12.23 depicts the time evolution of the displacement and the current in the coil for a four times heavier load.

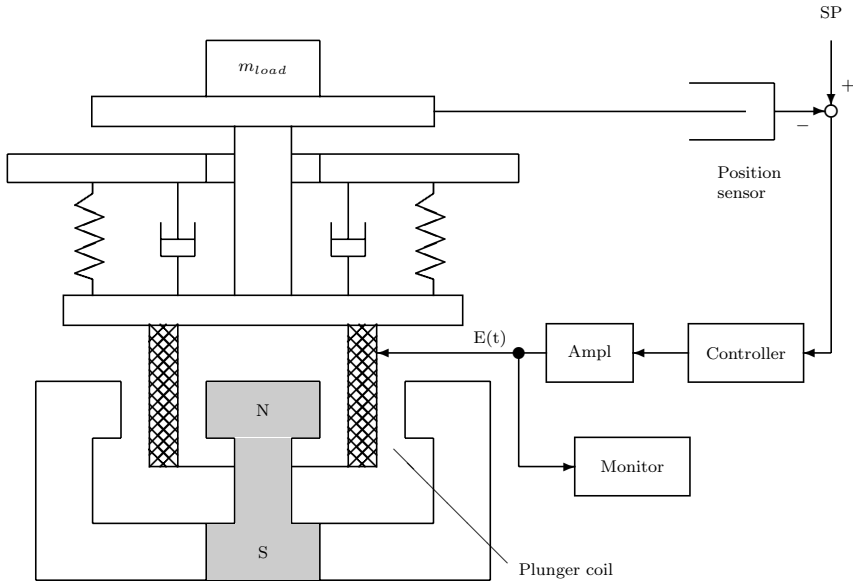


Fig. 12.20 Conceptual schematic of an electronic balance with displacement compensation (cf. [15])

12.5 A Piezoelectric Seismometer

This section addresses the bond graph modelling of a piezoelectric transducer and its use in a seismometer. A piezoelectric crystal can be considered an electromechanical transducer that mechanically behaves like a spring and electrically like a capacitor, and both effects are coupled. That is, a force imposed on the crystal produces results in a (small) voltage drop across the device and vice versa. Piezoelectric crystals are used, e.g. for sensing vibrations, for converting pressures into voltages, or as actuators in hydraulic control valves.

Bond Graph Model of the Piezoelectric Crystal

In the following, a one-dimensional model is considered. That is, it is assumed that mechanical stress or strain is applied in only one direction and the electric field lines are perpendicular to the parallel conductive surfaces of cross-sectional area A . Let x_0 denote the distance of these surfaces for the unbiased crystal. Let S denote the mechanical strain, $\Delta x = x_0 S$, the relative displacement of the parallel conductive surfaces, σ , the tensile stress, $F = A\sigma$, the associated force, E_m , the mechanical modulus of elasticity, E , the electric field strength, $u_p = x_0 E$, the applied voltage, D the electric displacement, ε , the dielectric constant and d_ε , the piezoelectric coupling. The constitutive relations of a piezoelectric crystal are assumed to be linear.

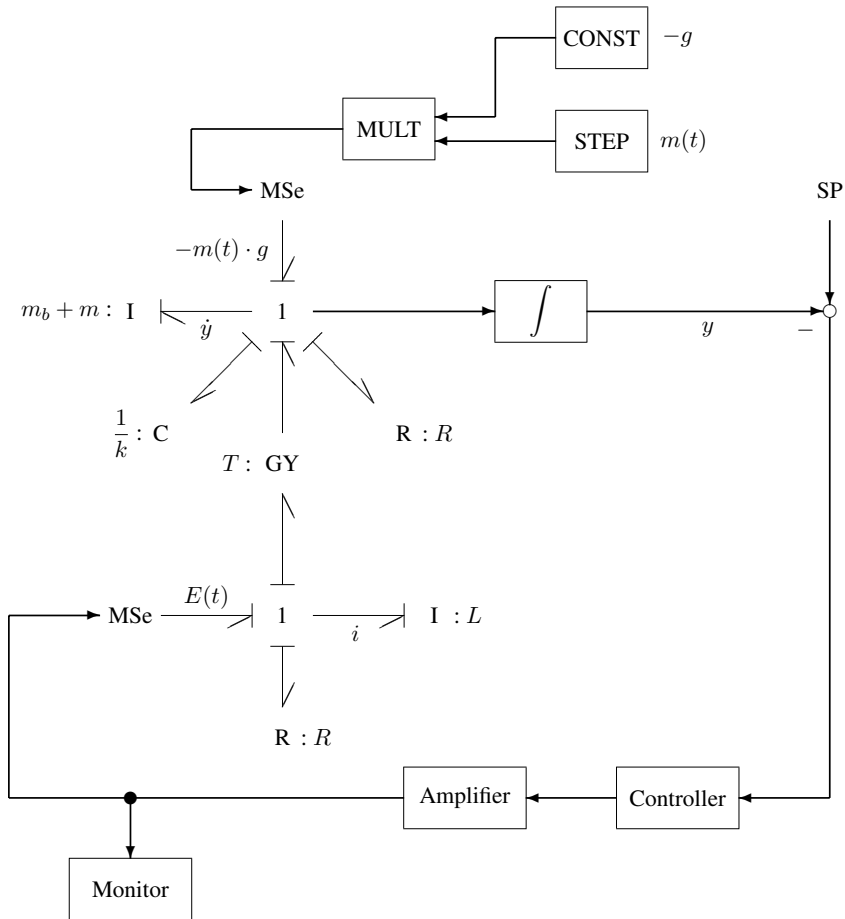


Fig. 12.21 Bond graph model of the electronic balance with displacement compensation

A commonly known form is

$$\begin{bmatrix} S \\ D \end{bmatrix} = \begin{bmatrix} 1/E_m & d_\epsilon \\ d_\epsilon & \epsilon \end{bmatrix} \begin{bmatrix} \sigma \\ E \end{bmatrix}. \tag{12.25}$$

Given the introduced quantities, the constitutive relations can be rewritten as

$$\begin{bmatrix} \Delta x \\ q \end{bmatrix} = \begin{bmatrix} 1/k_p & d_\epsilon \\ d_\epsilon & C_p \end{bmatrix} \begin{bmatrix} F \\ u_p \end{bmatrix}, \tag{12.26}$$

where $k_p = EA/x_0$ denotes the mechanical stiffness of the piezoelectric crystal and $C_p = \epsilon A/x_0$, its electrical capacitance.

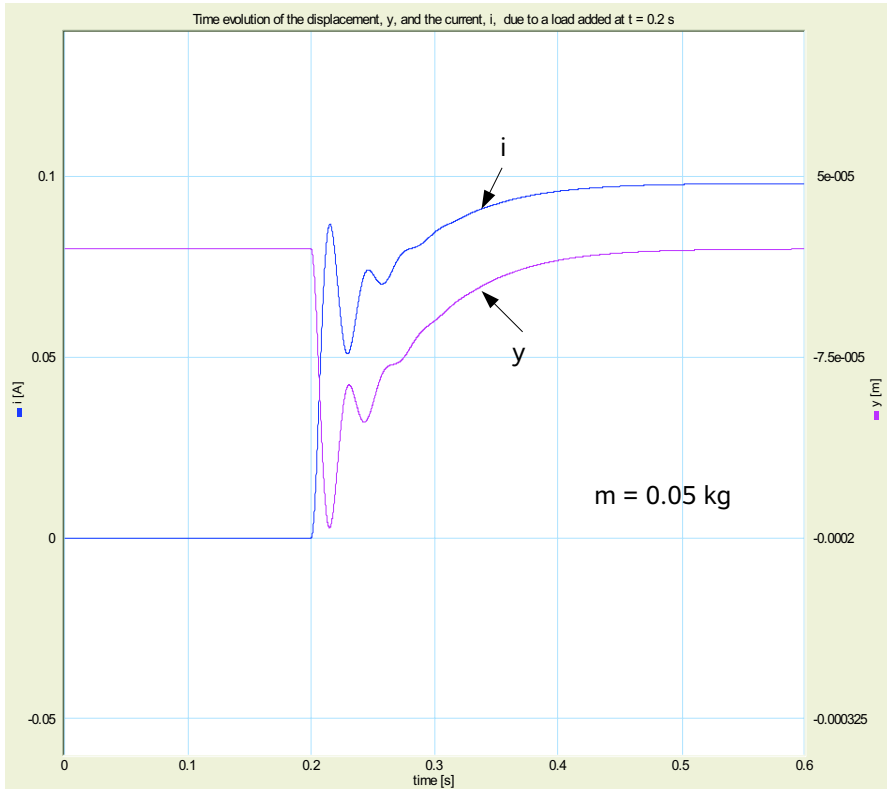


Fig. 12.22 Time evolution of the displacement and the current due to a load added at $t = 0.2\text{ s}$

If Equation 12.26 is solved for the vector $[F^t u_p]^t$, then the result can be read as the constitutive relation of a linear energy conservative 2-port C field in integral causality (cf. Figure 12.24). The linear 2-port C field can be decomposed as depicted in Figure 12.25, where $\Delta := c_p/k_p - d_\varepsilon^2$.

Bond Graph Model of the Piezoelectric Seismometer

In the following, the bond graph of Figure 12.25 is used in the development of a model of a piezoelectric seismometer as sketched in Figure 12.26.

The casing of the piezoelectric seismometer is placed on the ground, or, e.g., on a surface of a machine where vibrations $\dot{u}(t)$ are to be sensed. Inside the case, a piezoelectric crystal is attached to the case. A seismic mass, m , on top of the crystal is fixed to the case by a spring of stiffness k . The piezoelectric crystal reacts to the relative motion, $\dot{y}_r := \dot{y} - \dot{u}$, between the case and the seismic mass. It produces a small current that is fed into an amplifier to be modelled in a third step.

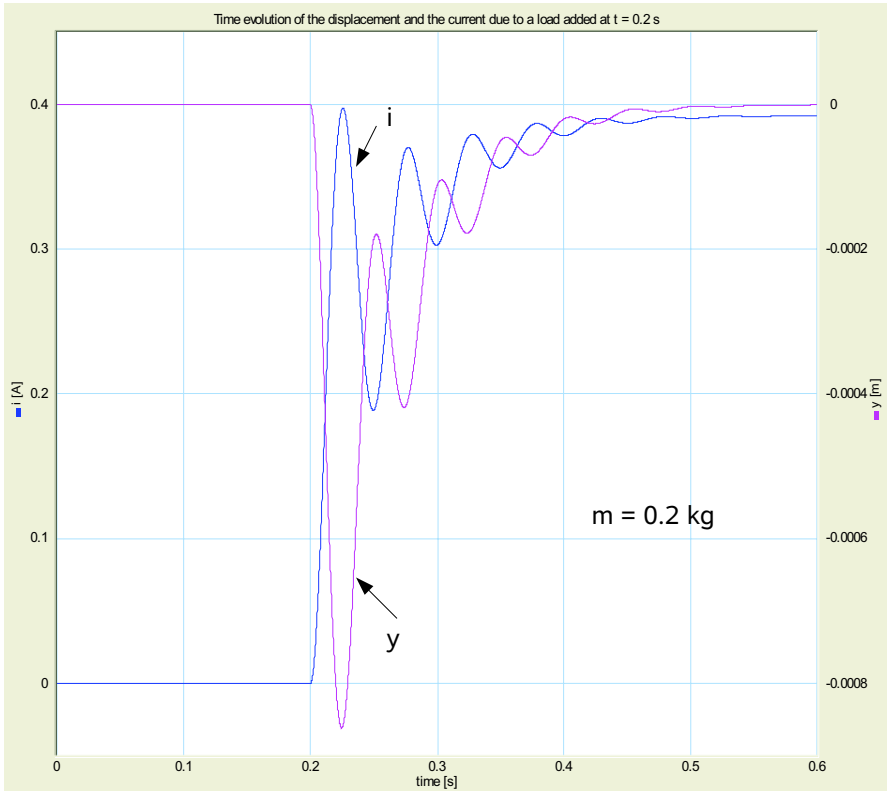


Fig. 12.23 Time evolution of the displacement and the current due to a four times heavier load added at $t = 0.2s$

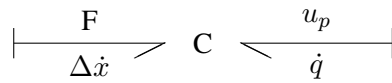


Fig. 12.24 Representation of a piezoelectric crystal by a 2-port C field

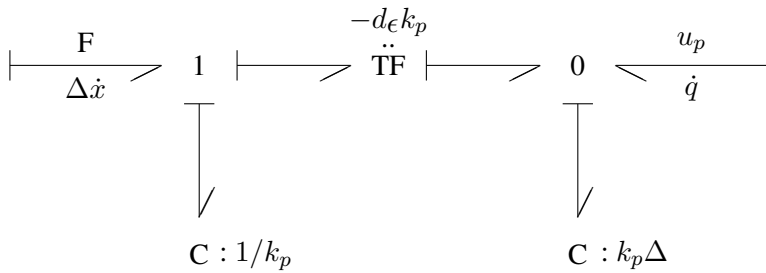


Fig. 12.25 Decomposition of the linear 2-port C field

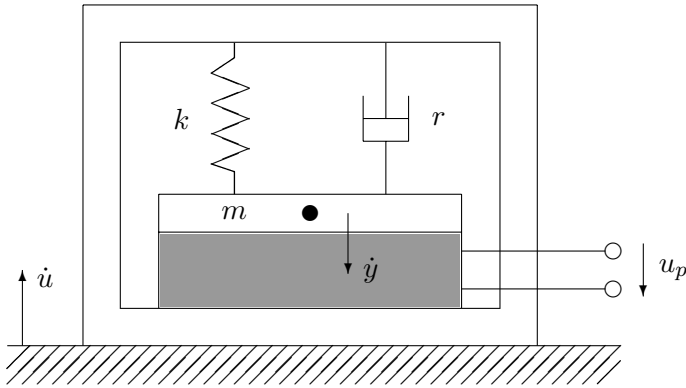


Fig. 12.26 Piezoelectric seismometer

The bond graph modelling of the seismometer is straightforward. The result is depicted in Figure 12.27. Displacements are relative to the position where gravitational force of the seismic load and the spring force of the crystal are in equilibrium.

From the bond graph of Figure 12.27, the following equation of motion is derived for the mechanical part of the piezoelectric sensor.

$$m\ddot{y}_r + r\dot{y}_r + (k + k_p)y_r = -m\ddot{u} - d_\epsilon k_p u_p \tag{12.27}$$

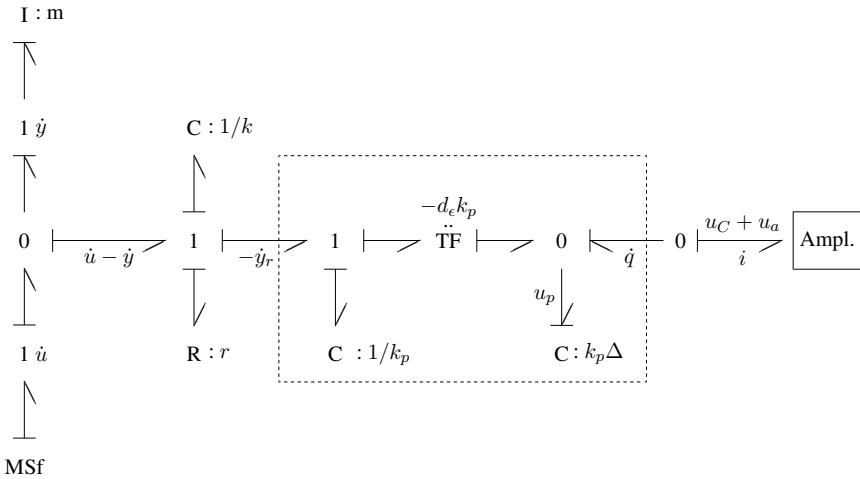


Fig. 12.27 Bond graph of the piezoelectric seismometer

Bond Graph Model of a Charge Amplifier

The current \dot{q} , generated by the piezoelectric crystal, is fed into a charge amplifier of which a circuit diagram is shown in Figure 12.28. Figure 12.29 shows a bond graph of the charge amplifier.

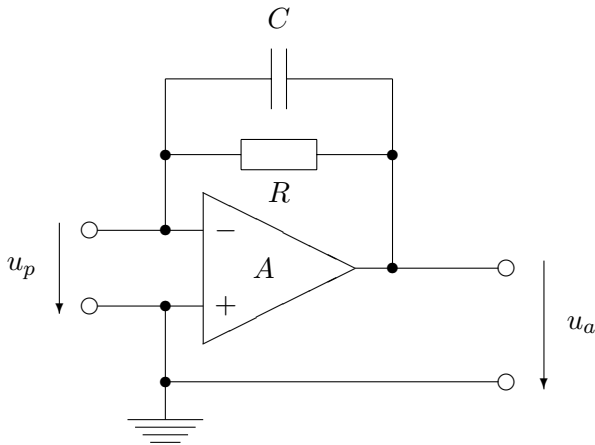


Fig. 12.28 Charge amplifier

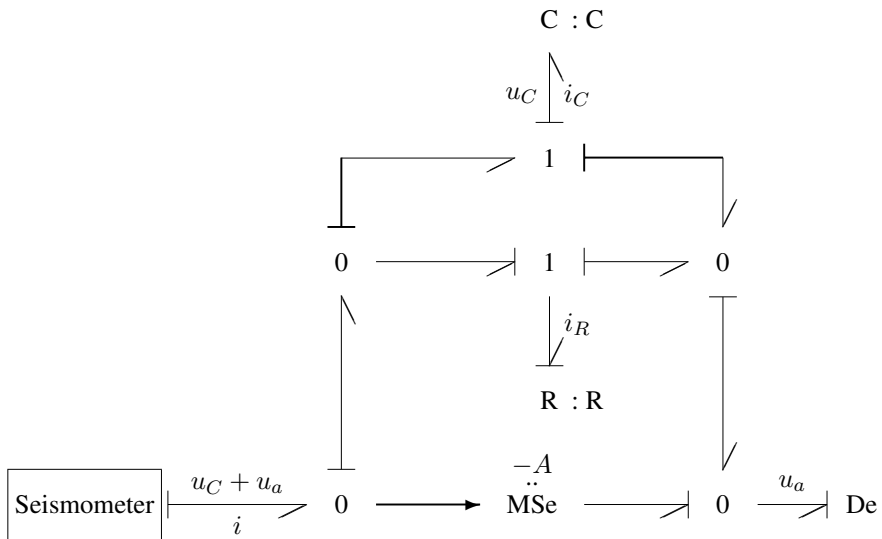


Fig. 12.29 Bond graph of the charge amplifier in Figure 12.28

From the bond graph of Figure 12.29, a relation between the voltage u_a at the amplifier's output and the current $i = -\dot{q}$ into the charge amplifier can be derived.

$$C\left(1 + \frac{1}{A}\right)\dot{u}_a + \frac{1}{R}\left(1 + \frac{1}{A}\right)u_a = \dot{q} \quad (12.28)$$

For $A \gg 1$, Equation 12.28 reduces to

$$RC\dot{u}_a + u_a = R\dot{q}. \quad (12.29)$$

Furthermore, the constitutive equation of the modulated voltage source in the bond graph of the amplifier (Figure 12.29) is

$$u_a = (-A)(u_C + u_a). \quad (12.30)$$

Hence, as $u_p = u_C + u_a$ (cf. Figure 12.27), a high value of the amplification, A , entails a small voltage, u_p , across the electrical terminals of the piezoelectric crystal.

$$u_p = u_a + u_C = -\frac{1}{A}u_a \approx 0 \quad (12.31)$$

Consequently,

$$-\dot{q} = (-d_\epsilon k_p)(-\dot{y}_r) \quad (12.32)$$

and

$$m\ddot{y}_r + r\dot{y}_r + \underbrace{(k + k_p)}_k y_r = -m\ddot{u}. \quad (12.33)$$

Frequency Analysis of the Piezoelectric Seismometer-Amplifier System

Combining the Laplace transforms of Equations 12.29, 12.31, and 12.33 yields the transfer function

$$\begin{aligned} \frac{\mathcal{L}u_a}{\mathcal{L}\ddot{u}} &= \frac{-\frac{m}{k}}{\underbrace{\frac{m}{k}}_{1/\omega_0^2} s^2 + \underbrace{\frac{r}{k}}_{2\zeta/\omega_0} s + 1} \underbrace{\left(-\frac{d_\epsilon k_p}{T}\right)}_T \underbrace{\frac{Rs}{RC}}_\tau s + 1 \\ &= \frac{\frac{TR}{\omega_0^2} s}{\left[\left(\frac{s}{\omega_0}\right)^2 + \left(\frac{2\zeta}{\omega_0}\right) s + 1\right] [\tau s + 1]}. \end{aligned} \quad (12.34)$$

For a Bode plot of the transfer function (Equation 12.34), the parameters in Table 12.6 have been used (cf. [15]). The frequency domain behaviour of the seismometer-amplifier system is shown in Figure 12.30.

Table 12.6 Parameters of the seismometer-amplifier system

Parameter	Value	Units	Meaning
f_0	10^4	1/s	$\omega_0 = 2\pi f_0$: eigenfrequency of the undamped seismometer
T	10^8	As/m	$T = d_e k_p$: Transduction coefficient of the piezoelectric crystal
ζ	0.5		Damping coefficient of the seismometer
R	1	Ω	Resistance (cf. Figure 12.28)
τ	5	s	$\tau = R \times C$: Time constant of the amplifier

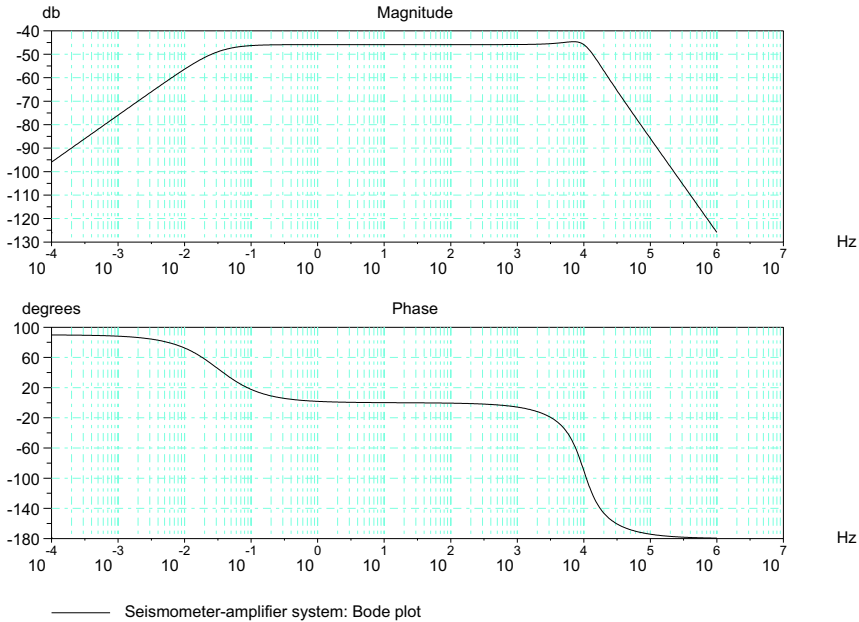


Fig. 12.30 Bode plot of the seismometer-amplifier system

As can be seen from Figure 12.30, the system can sense accelerations of the ground almost up to the eigenfrequency, f_0 , of the undamped seismometer. The low frequency behaviour of the system is limited by the high pass filter characteristic of the charge amplifier and its corner frequency $\omega_c = 0.2$ 1/s.

12.6 Engagement of a Clutch

In Chapter 7, some approaches to a bond graph representation of variable structure models have been discussed. If one wants to keep the structure of the bond graph and the computational causality invariant for all system modes, then one option is

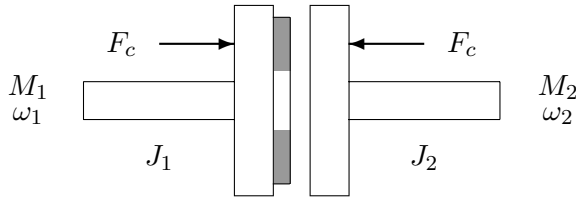


Fig. 12.31 Schematic of a clutch

to use sinks of fixed causality. At the advent of a discrete event, they impose an effort or a flow such that there is an instantaneous state transition and the new state conditions are met (cf. Section 7.1.4). In the following, this approach is used to model the engagement of a clutch. Another option is to model the clutch by an ideal switch of variable causality as has been done by Buisson in [4] (cf. also Section 7.2). In [25], this example has been used for a MATLAB®/Simulink® simulation.

Figure 12.31 shows a schematic of the clutch. Clearly, as long as the clutch is disengaged, the two disks rotate with their own angular velocity. When they get into contact, the two of them encounter a friction moment M_R . This friction moment reduces the angular velocity, ω_1 , on the drive side and simultaneously accelerates the power-takeoff side against a possible load moment M_2 . If the value of the contact force, F_c , is high enough, then the angular velocities of the disks will converge. At one point in time, they stick together and will continue to rotate as one single body with one and the same angular velocity ω . That is, there are two system modes.

Bond Graph Model of the Clutch

The approach in Section 7.1.4 leads to the bond graph in Figure 12.32.

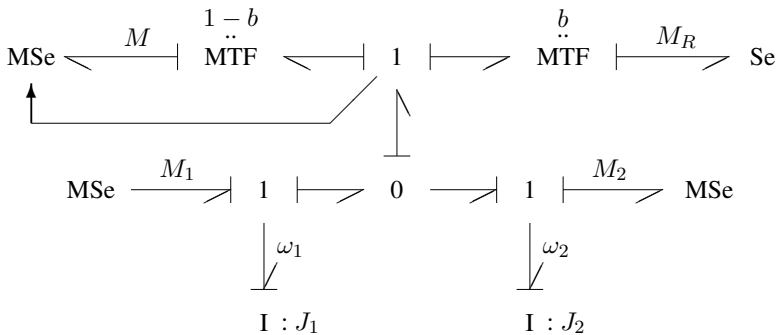


Fig. 12.32 Bond graph of the clutch with invariant structure and invariant computational causality

In the bond graph of Figure 12.32, M_1 and M_2 denote the moment on the drive side and the load moment on the power-takeoff side respectively. According to Section 7.1.2, switches have been modelled by modulated transformers. As long as there is a difference between the two angular velocities on the drive side and on the power-takeoff side, the modulus b equals 1 and both rotating disks encounter a friction moment, M_R , proportional to the contact force F_c . As in [25], it is assumed that the initial angular velocity of the engine is $\omega_{10} = 200 \text{ rad/s}$, while the transmission side starts from $\omega_{20} = 0 \text{ rad/sec}$. When the increasing angular velocity ω_2 equals ω_1 , then the two disks stick together and the modulus b of the right-hand side MTF is set to zero. Consequently, the upper left part of the bond graph becomes active. It provides a moment M that ensures that the two disks stick together and rotate with one and the same angular velocity ω . Note that in both system modes, the two inertia elements are invariably in integral causality. However, when the two disks rotate with a common angular velocity, the set of equations is a DAE system as there is no differential equation for the moment, M , ensuring that the angular velocity difference remains zero.

Simulation of the Clutch Behaviour

Given the parameter values in Table 12.7 [25], simulation (by means of Scilab) yields the time evolution of the angular velocities depicted in Figure 12.33.

Analytical Evaluation of the Clutch Model

Due to the simplifying assumptions, the problem can also be analytically solved. Hence, essential values obtained by simulation can be checked. For $\omega_2 < \omega_1$ ($b = 1$), the following two equations for the inertia elements are easily derived from the bond graph.

$$\dot{\omega}_1 = \frac{1}{J_1} (M_1 - M_R) \quad (12.35a)$$

$$\dot{\omega}_2 = \frac{1}{J_2} (M_R - M_2) \quad (12.35b)$$

Table 12.7 Parameter values used for simulation of the clutch engagement

Parameter	Value	Units	Meaning
M_1	200	Nm	Driving torque
M_2	0	Nm	Load moment
J_1	1	kgm^2	Moment of inertia on the drive side
J_2	5	kgm^2	Moment of inertia on the power-takeoff side
F_c	5000	N	Contact force
k	0.38	m	$M_R = k \times F_c$
$\omega_1(0)$	200	rad/s	Initial angular velocity on the drive side
$\omega_2(0)$	0	rad/s	Initial angular velocity on the transmission side

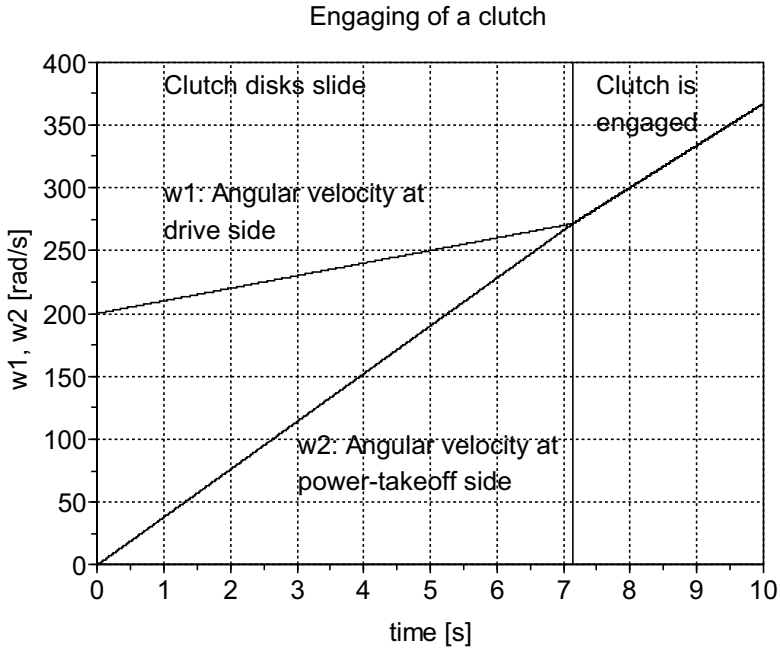


Fig. 12.33 Angular velocities of the clutch disks

Their integration yields

$$\omega_1(t) = \frac{M_1 - M_R}{J_1} t + \omega_{10} \quad (12.36a)$$

$$\omega_2(t) = \frac{M_R - M_2}{J_2} t + \omega_{20} . \quad (12.36b)$$

If the value of the contact force, F_c , is sufficiently high, the clutch disks stick together at some time point $t = t_1$ and continue to rotate with one and the same angular velocity ω . Equality of both angular velocities gives for t_1

$$t_1 = \frac{\omega_{10} - \omega_{20}}{\frac{M_2 - M_R}{J_2} + \frac{M_R - M_1}{J_1}} . \quad (12.37)$$

The parameter values in Table 12.7 result in the value $t_1 = 7.143 \text{ s}$. The value of the common angular velocity is $\omega = 271.43 \text{ rad/s}$. For $t > t_1$, the angular acceleration $\dot{\omega}$ reads

$$\dot{\omega} = \frac{M_1}{J_1 + J_2} . \quad (12.38)$$

The numerical value is $\dot{\omega} = 33.3 \text{ rad/s}^2$.

For $t > t_1$ ($b = 0$), the DAE system derived from the bond graph reads

$$\dot{\omega}_1 = \frac{1}{J_1} (M_1 - M) \quad (12.39a)$$

$$\dot{\omega}_2 = \frac{1}{J_2} (M - M_2) \quad (12.39b)$$

$$0 = \omega_1 - \omega_2 . \quad (12.39c)$$

After differentiation of the algebraic constraint with respect to time, solution of the resulting ODE system yields for the moment M

$$M = \frac{J_2}{J_1 + J_2} M_1 + \frac{J_1}{J_1 + J_2} M_2 . \quad (12.40)$$

Given the parameter values in Table 12.7, the value is $M = 166.7 \text{ Nm}$. Hence, for $t \geq t_1$, the descriptor vector, $[\omega_1, \omega_2, M]^t$, has the initial conditions $[271.43, 271.43, 166.7]^T$. Location of the time point t_1 and continuation of the simulation with the correct initial conditions can be performed by the root finding version of the widely used numerical integration codes DASSL [2] or ODEPACK [14]. These solvers are part of, for instance, the open source mathematical software package Scilab [26].

12.7 Dry Friction in a Suspension Strut of a Car

Another example in which different system modes can be distinguished is the stick-slip effect in a suspension strut of a car. For the analysis of this effect, the widely used simple quarter vehicle model depicted in Figure 12.34 is considered.

As long as there is slip friction effective in the strut, the mass of the chassis, m_c , and the mass of the wheel, m_w , move up and down with different displacements x_c and x_w . If stick friction is effective, there is a holonomic constraint

$$x_w - x_c = \text{const.} \quad (12.41)$$

and the view can be taken that one body with the combined mass is moving up and down. In this mode, one degree of freedom has gone. Once the break value F_H of the sticking force has been overcome, the system returns into the slip mode and the suppressed degree of freedom is available again. In [18], Kölsch and Ostermeyer account for the sticking mode and the switching between slip and sticking mode by a modification of the constraint force between two bodies in the equations of motion.

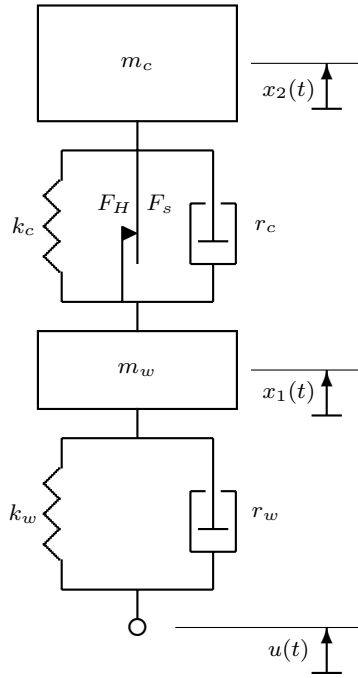


Fig. 12.34 Schematic of a quarter vehicle

Bond Graph Model of the Quarter Vehicle Accounting for Stick-Slip Friction

In the following, the more general bond graph approach using sinks of fixed computational causality described in Section 7.1.4 is applied.

Figure 12.35 shows a bond graph of the quarter vehicle accounting for the slip and the sticking mode. In the bond graph model of Figure 12.35, the modulated effort sink provides the constraint force λ . The value of the transformer modulus b accounts for the system mode. For $b = 1$, the sticking force F_{stick} is enabled and, simultaneously, the slip friction force F_{slip} is disabled and vice versa for $b = 0$. That is, according to the system mode, one part of the bond graph model is switched on and another one is switched off. Note that computational causalities remain fixed independent of the system mode.

Derivation of Model Equations from the Bond Graph

The mathematical model to be derived from the bond graph depends on the system mode. For the slip mode, it is a set of explicit ODEs. For the sticking mode, it is a DAE system of index 2 that can analytically reduced to an explicit ODE system

$$\lambda = k_c(x_2 - x_1) + r_c(\dot{x}_2 - \dot{x}_1) - k_w \frac{m_c}{m_w + m_c}(x_1 - u) - r_c \frac{m_c}{m_w + m_c}(\dot{x}_1 - \dot{u}) . \quad (12.43)$$

In slip mode ($b = 0$), the friction force F is

$$F = F_{\text{slip}} = F_s \text{sign}(v_1 - v_2) . \quad (12.44)$$

Simulation of the Stick-Slip Problem Using Scilab and LSODAR

Computation of this hybrid model requires the location of times at which the system mode changes from sliding to sticking or vice versa. This suggests the use of the root finding version of an ODE solver. For this case study, Scilab and the ODE solver LSODAR have been used.

Formulation of the model equations as a Scilab function is straightforward. The result is shown in Figure 12.36. The Scilab function `f` is an argument of the Scilab function `ode` which is an interface to the ODE solvers in the software package ODEPACK.

```
// Scilab function of a quarter vehicle
// Suspension with dry friction

function [dy] = f(t,y)

v1      = y(1)
v2      = y(2)
x1      = y(3) // vertical position of the chassis
x2      = y(4) // vertical position of the wheel

u       = ramp(t,t0,t1,u1) // unevenness of the road surface
du      = pulse(t,t0,t1)

Fslip   = Fs*sign(v2 - v1)

lambda  = kc*(x2 - x1) + rc*(v2 - v1) - c1*(x1 - u) - c2*(v1 - du)

mwdv1   = -(rw + rc)*v1 + rc*v2 - (kw + kc)*x1 + kc*x2 + kw*u + rw*du
         + (1-b)*Fslip - b*lambda
dv1     = mwdv1 / mw

mcdv2   = rc*v1 - rc*v2 + kc*x1 - kc*x2 - (1-b)*Fslip + b*lambda
dv2     = mcdv2 / mc

dy1     = dv1
dy2     = dv2
dy3     = v1
dy4     = v2

dy      = [dy1;dy2;dy3;dy4]

endfunction
```

Fig. 12.36 Equations of the quarter vehicle model as a Scilab function

Figure 12.37 shows the Scilab script for a simulation run on the quarter vehicle model. In the Scilab script of Figure 12.37, the meaning of variables has been explained by inline comments. The root finding capability of the ODE solver is used to locate the time points t_c at which the system mode may change from sticking to slipping or vice versa. If the current mode is sticking ($b = 1$), then integration is performed up to a time point for which $|\lambda| - F_H = 0$ holds. Otherwise, if the current mode is sliding ($b = 0$), then integration stops at a time point for which $v_1 = v_2$. Both conditions are formulated in the function \mathfrak{g} . If such a time point t_c has been located, it must be checked if the conditions for a change in the system mode are met. If the current mode is sticking and the absolute value of the constraint force reaches the breakaway level, F_H , of the friction force F , then both bodies can slip. On the contrary, if both bodies slide with a common velocity, then there is potential risk for sticking. Both bodies will stick if the absolute value of lambda falls below the slip level, F_s , of the friction force F . The system mode is taken into account by the value of the modulus b , which must be changed appropriately. This is done in the function `change` depicted in Figure 12.38. In this manner, integration proceeds from one discrete event to the next until the final simulation time has been reached.

For simulation runs, parameter values given in [18] have been used (Table 12.8). As to the velocity excitation of the road, $\dot{u}(t)$, it is assumed that the quarter vehicle moves up a ramp (Figure 12.39). Furthermore, it is assumed that the strut is in stiction mode at $t = 0$. The initial conditions for the simulation run are

$$v_1(0) = v_2(0) = 0 \quad (12.45)$$

and

$$x_1(0) = x_2(0) = 0. \quad (12.46)$$

The simulation result in Figure 12.40 shows that immediately after the start, the system changes from initial sticking into the slipping mode, which lasts until about 1.43 s. During this period, there are two short time intervals in which sticking occurs. As can be seen from Figure 12.40, in fact, the difference between the vertical velocities of the wheel, v_1 , and of the chassis, v_2 , vanishes during the sticking periods. Figure 12.41 shows the time history of the vertical displacements of the wheel, x_1 , and of the chassis (x_2). According to Figure 12.40, for $t > 1.43$ s, the wheel

Table 12.8 Parameter values used for simulation of the quarter vehicle

Parameter	Value	Units	Meaning
m_w	100	<i>kg</i>	Mass of the wheel
r_w	126	<i>Ns/m</i>	Damping coefficient for the tyre
k_w	395×10^3	<i>N/m</i>	Stiffness of the tyre
r_c	1500	<i>Ns/m</i>	Damping coefficient of the suspension
k_c	39.5×10^3	<i>N/m</i>	Stiffness of the suspension
F_H	500	<i>N</i>	Breakaway level of the friction force F
F_s	500	<i>N</i>	Slip level of the friction force F
m_c	1000	<i>kg</i>	Mass of the chassis

```

// run quarter vehicle model

%ODEOPTIONS=[2,0,0,%inf,0,2,1000,12,5,0,-1,-1];
tf = 5.0; // [sec] tf: final time of the integration
ng = 1;
b = 1; // initial mode is sticking
//
// 'root': LSODAR from ODEPACK is called
// x0: vector of initial conditions
// t0: initial start time
// f: right-hand side of the set of ODEs: ydot = f(t,y)
// g: integration of ydot = f(t,y) is performed
// until g(t,y) = 0 holds for one component of y
// rd(1): time at which integration stops
// sol: matrix, each row >= 2 contains a component of y at times <= rd(1)
//
[sol,rd] = ode('root',x0,t0,tf,f,ng,g);
m = size(sol); // m(2): number of columns
tc = rd(1);
//
bvector=b*ones(1,m(2)); // contains system mode at times <= tc
//
// check if the system mode changes at tc:
//
b = change(sol);
//
// continue the integration as long as tc <= tf:
//
while tc <= tf
//
// use values at tc as new initial conditions xc:
//
xc = [sol(2,m(2));sol(3,m(2));sol(4,m(2));sol(5,m(2))];
//
// continue the integration until the solution crosses
// the surface g(t,y) = 0:
//
[xsol2,rd2] = ode('root', xc,tc,tf,f,ng,g);
//
m2 = size(xsol2);
bvector2 = b*ones(1,m2(2));
bvector = [bvector,bvector2];
//
sol = [sol,xsol2];
m = size(sol);
tc = rd2(1);
//
// check if the system mode changes at tc:
//
b = change(sol);

```

Fig. 12.37 Scilab script for a simulation run on the quarter vehicle model

```

// Function change() is called at time point tc
// at which integration has stopped because the conditions
// specified in function g() are met.
// According to the result of the check below function change()
// returns an update of the system mode indicator b used in the
// script that calls change().
// Fs: slip level of the friction force F

function [b] = change(sol)
v1 = sol(2,m(2))
v2 = sol(3,m(2))
x1 = sol(4,m(2))
x2 = sol(5,m(2))

u = ramp(tc,t0,t1,u1)
du = pulse(tc,t0,t1)

lambda = kc*(x2 - x1) + rc*(v2 - v1) - c1*(x1 - u) - c2*(v1 - du)

mode = b; // current system mode
if (mode == 0 & abs(lambda) <= Fs) then b = 1; end
if mode == 1 then b = 0; end

endfunction

```

Fig. 12.38 Function change changing the system mode indicator b appropriately

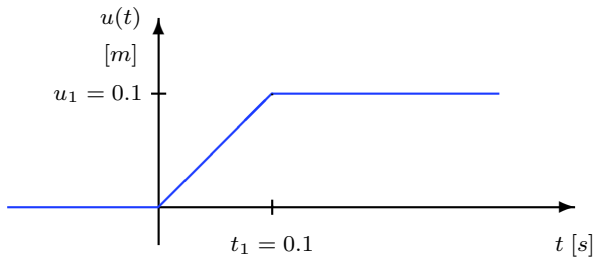


Fig. 12.39 Roadway unevenness u over time

and the chassis stick together and oscillate with a common frequency of about 3 Hz and a very small amplitude up and down. This oscillation is lightly damped because of the low damping coefficient of the tyre.

Finally, Figure 12.42 displays the time evolution of the constraint force λ and of the slip friction force $F_{\text{slip}} = F_s \text{sign}(v_1 - v_2)$. The simulation results presented in this case study agree with those given in [18].

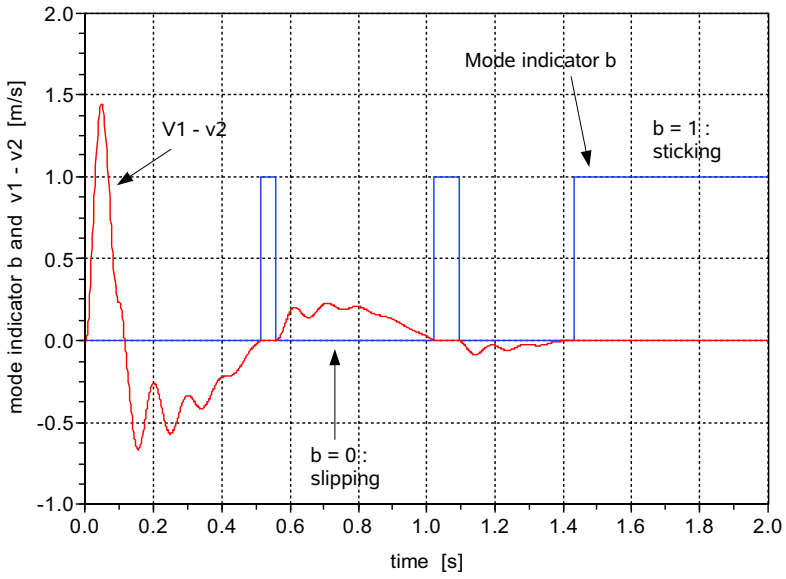


Fig. 12.40 System mode indicator b and relative velocity between wheel and chassis

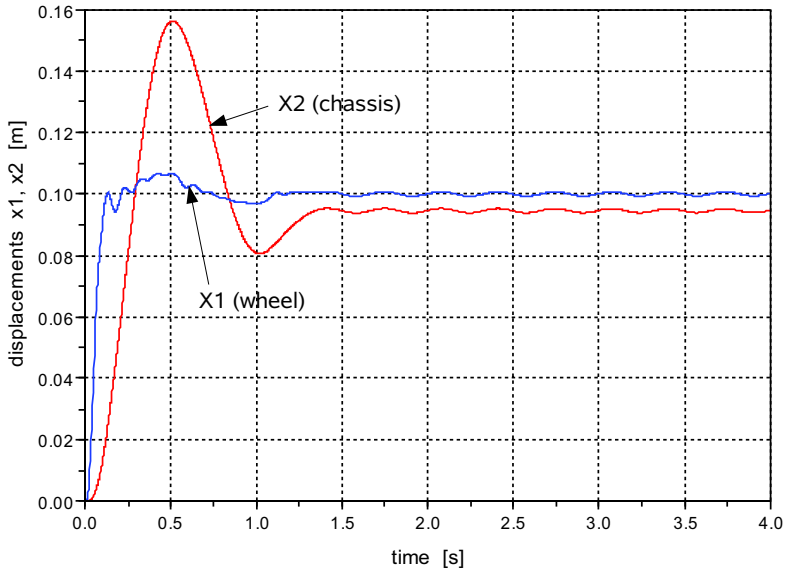


Fig. 12.41 Vertical displacements of the wheel and of the chassis

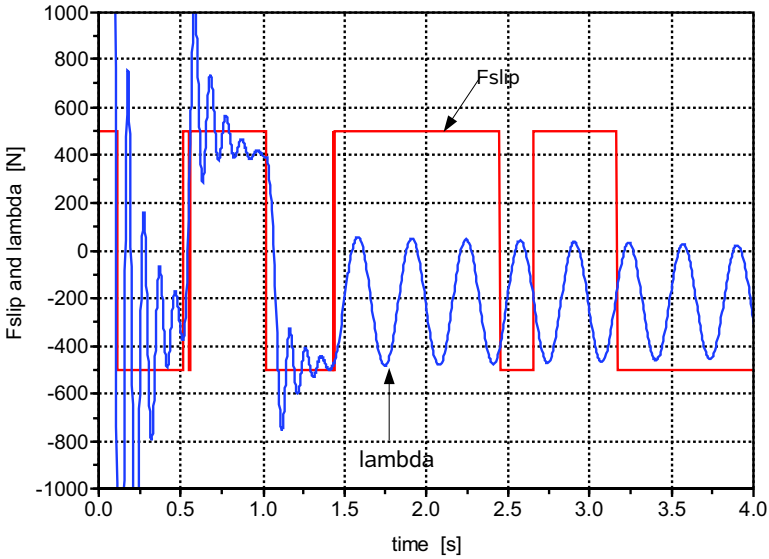


Fig. 12.42 Constraint force λ and slip friction force F_{slip}

12.8 A Buck Converter

Chapter 7 discusses several approaches to a representation of variable structure models. As bond graphs are based on the energy exchange between system components taking place in time periods not equal to zero, they are best suited for continuous time models. Accordingly, Section 7.3 proposes to describe discrete system states and transitions between them by a Petri net and to develop a bond graph model for each system mode. For illustration of this approach and in order to apply bond graph modelling to an electronic circuit beyond passive RLC networks, a DC-DC buck converter as displayed in Fig 12.43 is studied.

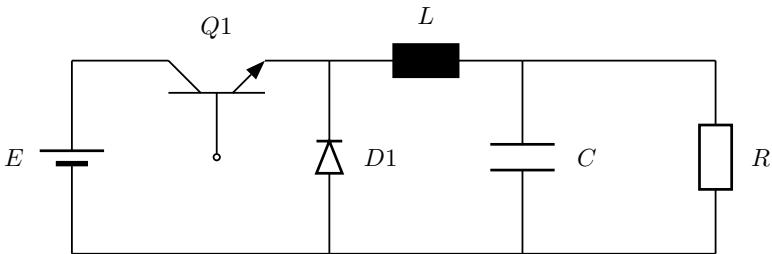


Fig. 12.43 Schematic of a buck converter

A DC-DC buck converter is a well known power electronic device [5]. Its purpose is to reduce a DC input voltage. The use of switching elements enables one to reduce the energy consumption in comparison to linear regulators. Due to its function and the use of switching elements, it is also called step-down switch mode power supply. As the device is superior to linear voltage regulators, they are used in applications where size and power dissipation matters, e.g. for the low voltage power supply of processors in laptop computers, or in rechargers.

In the context of bond graph modelling of power electronic circuits, buck converters have been considered by several authors [1, 10, 11, 20]. The transistor $Q1$ and the diode D are usually modelled as switches with an ON-resistance. In [10, 11], Garcia-Gomez uses a unique bond graph model for all switch modes and represents the switches by means of a modulated transformer and a resistor (cf. Figure 7.6, Section 7.1.2).

System Modes of the Buck Converter

If the transistor and the diode are considered as switches, theoretically, four system modes as listed in Table 12.9 can be distinguished.

When the transistor switch is in ON state (closed), a current is flowing through the inductor into the load composed of the resistor R and the capacitor C in parallel, and the inductor stores energy as highlighted in Figure 12.44. In this system mode, the diode (displayed in grey in Figure 12.44) is off because it is reverse biased. This system mode is known as load state.

Table 12.9 Theoretical switch state combinations

System mode	Transistor	Diode
1	ON	OFF
2	OFF	ON
3	ON	ON
4	OFF	OFF

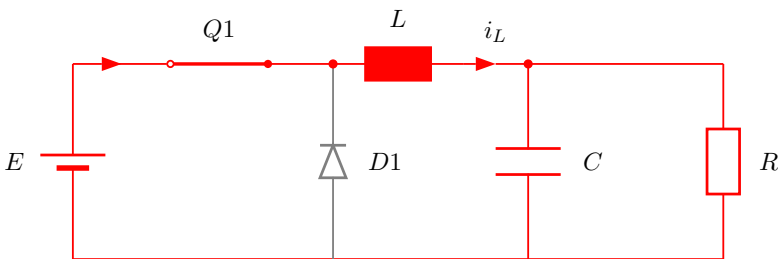


Fig. 12.44 Buck converter in mode 1 (transistor switch on and diode off)

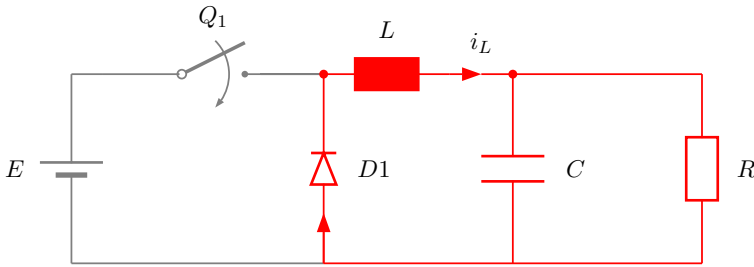


Fig. 12.45 Buck converter in mode 2 (transistor switch off and diode on)

In system mode 2, the transistor switch is off (Figure 12.45). The diode is forward biased and the energy stored in the inductor discharges into the load. This mode is called free wheel mode.

If the ON resistance of both switches is neglected, then the third switch state combination (both switches are closed) would entail a short-circuit of the voltage source and consequently a disfunction of the circuit. That is, this switch state combination can be discarded. This is reflected by a causal conflict in the bond graph of Figure 12.46.

In the fourth system mode (both switches are off), the current through the coil has become zero and the coil does not store any magnetic energy. While the inductor remains empty, the energy of the capacitor discharges via the load resistor (Figure 12.47). In the literature, this state is sometimes called the rest state. In Figure 12.47, there is no current in the part of the circuit displayed in grey. In the following, it is assumed that the transistor is switched on and off periodically and that there are no time periods in which the current through the inductor remains zeros. In this case, the buck converter is said to operate in continuous mode. Accordingly, the circuit toggles between the two system modes 1 and 2. This is captured in the simple Petri net of Figure 12.48. In the Petri net, T denotes the duty cycle of the

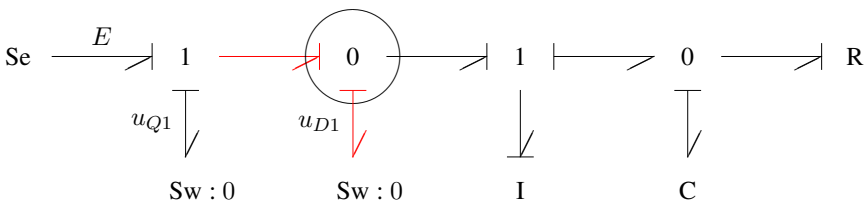


Fig. 12.46 Bond graph of the buck converter in case both switches are on

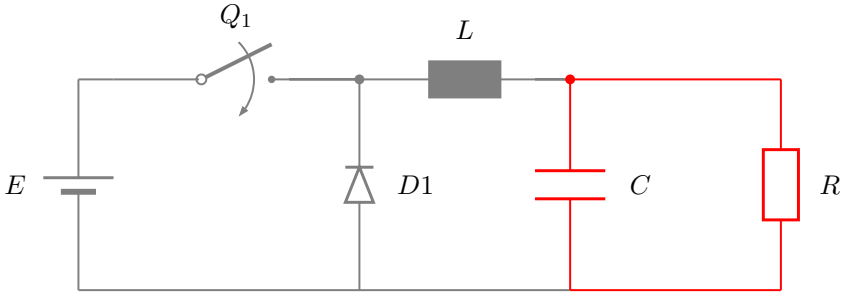


Fig. 12.47 Buck converter in mode 4 (both switches off)

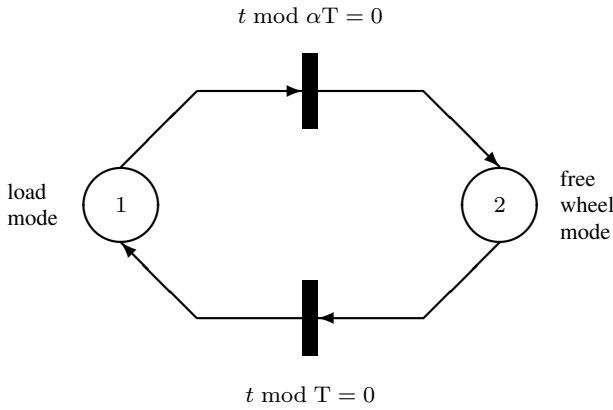


Fig. 12.48 Petri net for a periodically switched buck converter

signal switching the transistor on and off. The transistor is on for the period αT ($0 < \alpha < 1$), while it is off during the period $(1 - \alpha) T$ (Figure 12.49).

Bond Graph Models of the Buck Converter in Load Mode and in Free Wheel Mode

For both system modes, the construction of a bond graph is straightforward. Figures 12.50 and 12.51 show the results.

A straightforward way to simulate the behaviour of the buck converter is to alternate between the computation of the two models. For the buck converter under study, a look at the equations derived from the bond graph models shows that both models can be combined into one unique bond graph displayed in Figure 12.52. In the bond graph of Figure 12.52, the transformer modulus m is the OnOff-function

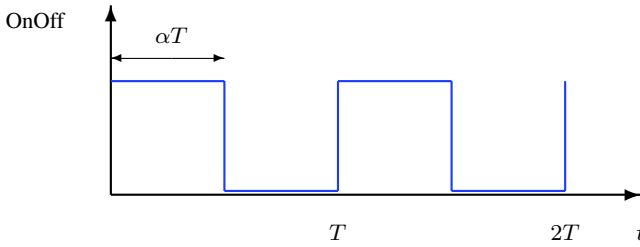


Fig. 12.49 Signal switching the transistor on and off

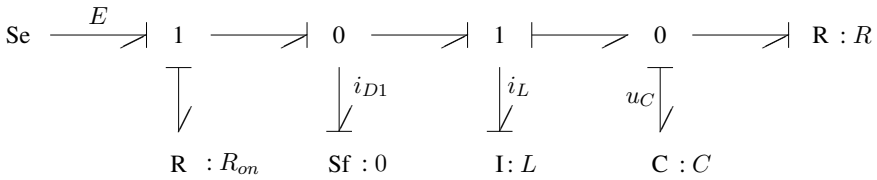


Fig. 12.50 Bond graph of the buck converter in load mode

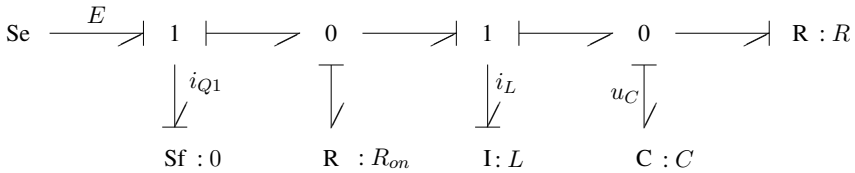


Fig. 12.51 Bond graph of the buck converter in free wheel mode

in Figure 12.49. This modulus toggles the computation between the sets of model equations for the two system modes.

Simulation of the Buck Converter Dynamic Behaviour

The simulation study uses the parameters listed in Table 12.10. Figure 12.53 displays the time evolution of the current, i_L , through the inductor as well as its mean value \bar{i}_L over one duty cycle. Accordingly, Figure 12.54 shows the time evolution of the voltage drop, u_C , across the capacitor as well as its mean value \bar{u}_C .

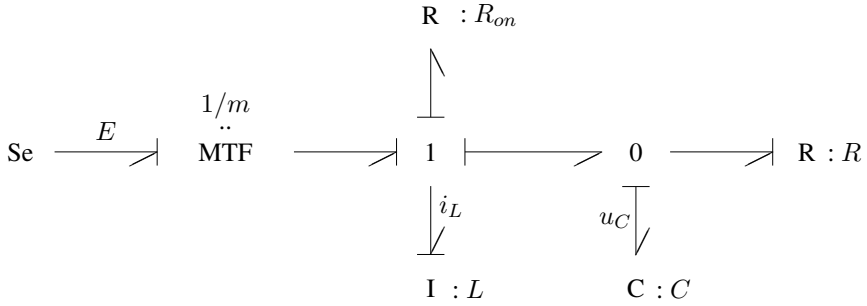


Fig. 12.52 Unique bond graph of the buck converter for system modes 1 and 2

Table 12.10 Parameters used in the simulation study

Parameter	Value	Units
E	100	V
R_{on}	0.1	Ω
L	50	mH
C	50	μF
R	50	Ω
T	10^{-3}	s
α	0.7	-

Verification of Simulation Results

As the circuit under study is rather simple, simulation results can be checked against analytical results. From the bond graph of Figure 12.52, the following two state equations can be derived.

$$\frac{di_L}{dt} = \frac{1}{L} [mE - R_{on} i_L - u_C] \tag{12.47a}$$

$$\dot{u}_C = \frac{1}{C} [i_L - \frac{1}{R} u_C] \tag{12.47b}$$

If the ON resistance R_{on} is neglected and if mE is replaced by the mean value αE , then for $t \rightarrow \infty$, the voltage drop across the capacitor takes the value

$$\bar{u}_C = \alpha E = 0.7 \times 100 \text{ V} = 70 \text{ V} . \tag{12.48}$$

Accordingly, for $t \rightarrow \infty$, the current through the inductor takes the value

$$\bar{i}_L = \frac{1}{R} \bar{u}_C = \frac{1}{50} 70 \text{ A} = 1.4 \text{ A} . \tag{12.49}$$

The mean values ($t \rightarrow \infty$) obtained from simulation are in good agreement with these values.

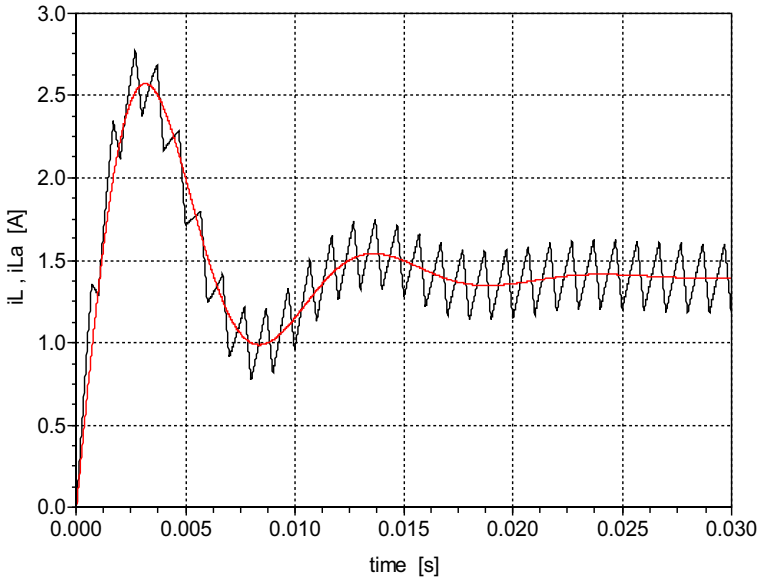


Fig. 12.53 Time evolution of the current through the inductor

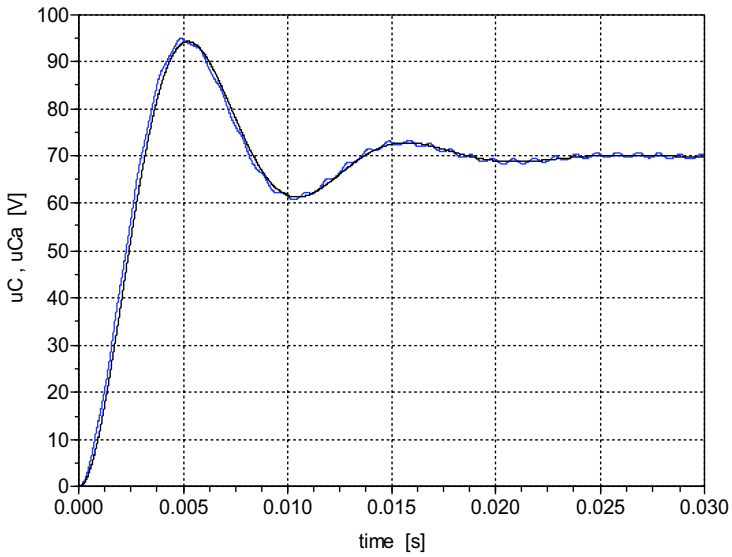


Fig. 12.54 Time evolution of the voltage drop across the capacitor

Furthermore, as Figure 12.53 shows, there is a significant ripple on the waveform of the inductor current. If the ON resistance of the switches is neglected and if the voltage across the capacitor is replaced by the mean value αE , then Equation 12.47a reads

$$\frac{di_L}{dt} = \frac{1}{L}[mE - \alpha E] = \frac{m - \alpha}{L}E. \quad (12.50)$$

That is, when the circuit is in load state ($m = 1$), the current through the inductor rises linearly while it falls linearly in the free wheel state ($m = 0$). For $m = 1$, the value of the slope is

$$\frac{di_L}{dt} = \frac{1.0 - 0.7}{50 \times 10^{-3}} \times 100 \text{ A/s} = 600 \text{ A/s}. \quad (12.51)$$

For $m = 0$, the inclination is

$$\frac{di_L}{dt} = \frac{-0.7}{50 \times 10^{-3}} \times 100 \text{ A/s} = -1400 \text{ A/s}. \quad (12.52)$$

As can be seen from an enlargement of the ripple depicted in Figure 12.55, simulation results agree well with these values.

According to [24], the amplitude $\Delta I/2$ of the ripple on the waveform of the inductor current i_L is

$$\Delta I = \frac{\alpha E(1 - \alpha)}{L}T. \quad (12.53)$$

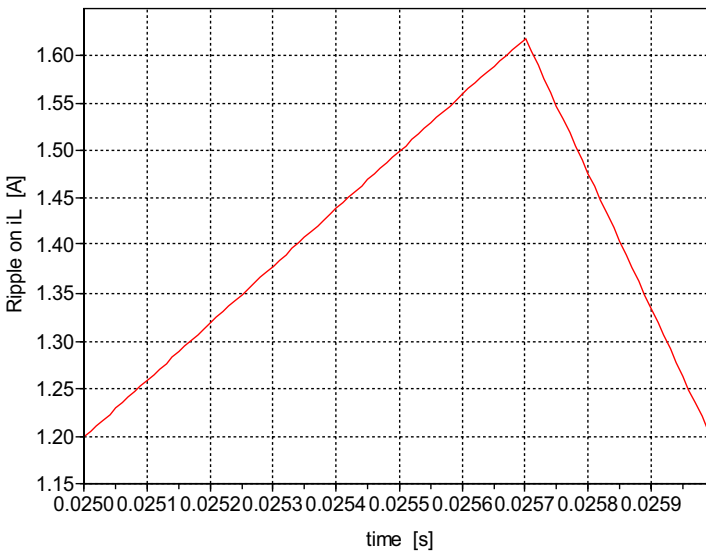


Fig. 12.55 Enlargement of the ripple on the waveform of the inductor current

With the parameters in this case study, the amplitude is 0.21 A. As the mean value is 1.4, values of the ripple should be within the range

$$1.19 A = (1.40 - 0.21) A \leq i_L \leq (1.40 + 0.21) A = 1.61 A .$$

This can be verified by inspection of Figure 12.55.

The amplitude $\Delta V/2$ of the ripple on the waveform of u_C is

$$\Delta V = \frac{\Delta I}{8C} \times T \tag{12.54}$$

[24]. The parameters in this case study give the result $\Delta V = 1.05 V$. The enlargement of the ripple on the waveform of u_C well confirms this value (Figure 12.56). The ripple on the waveform of u_C oscillates around the mean value of 70 V with an amplitude of about 0.5 V.

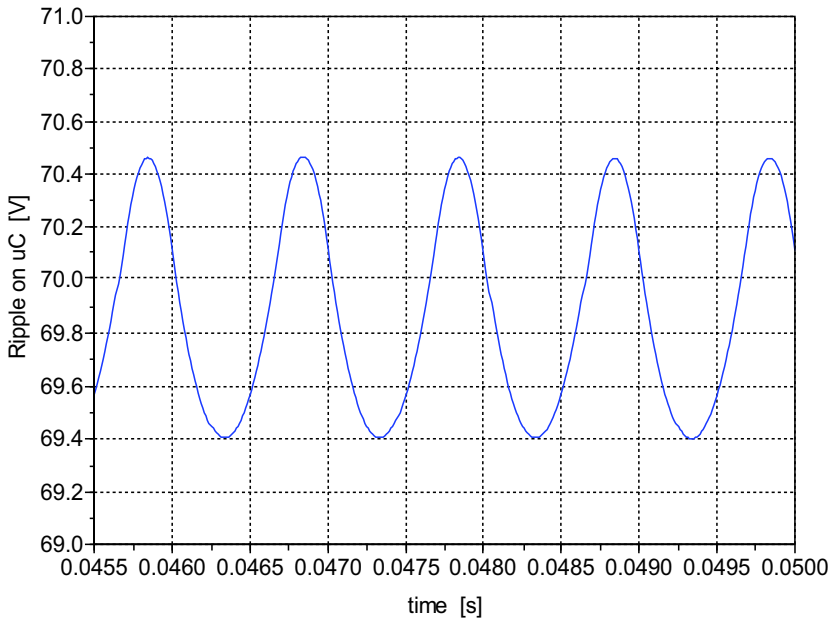


Fig. 12.56 Enlargement of the ripple on the waveform of the voltage drop across the capacitor

12.9 A Two Degrees of Freedom Rotary Joint Manipulator

The concise representation of multibond graphs supports the systematic development of bond graph models of multibody systems. Library models of a freely moving rigid body with hinges and various types of joints can be assembled in the same way the bodies and the joints of the real system are connected. What needs to be taken into account are transformations between body fixed reference frames represented by multiport transformers in a multibond graph.

For illustration, a multibond graph of a part of the Stanford arm has been given in Chapter 8). In the following, it is assumed that the prismatic joint is locked, resulting in a simpler rotary joint manipulator with two degrees of freedom. Figure 12.57 depicts a schematic of this manipulator which may be considered a part of the industrial PUMA robot.

As can be seen from Figure 12.57, body 1 simply rotates on its y_1 axis, while body 2 moves in three dimensions by rotation on an axis through point A that is parallel to the z_1 axis. This axis in turn rotates around the y_1 axis. This type of manipulator has also been considered by Gawthrop and Smith in [13] and by Vergé and Jaume in [29]. In this case study it is shown that the standard form of robot equations [8]

$$\mathbf{M}(\Theta)\ddot{\Theta} + \mathbf{V}(\Theta, \dot{\Theta}) + \mathbf{G}(\Theta) = \tau, \quad (12.55)$$

can be directly derived from the multibond graph of the robot in all derivative causality. In Equation 12.55, the vectors Θ , $\dot{\Theta}$, $\ddot{\Theta}$ denote the position, the velocity, and the accelerations of the joints. $\mathbf{M}(\Theta)$ is the $n \times n$ mass matrix, $\mathbf{V}(\Theta, \dot{\Theta})$ the $n \times 1$ vector of centrifugal and Coriolis terms, and $\mathbf{G}(\Theta)$ is an $n \times 1$ vector of gravity terms. Torques are combined into the vector τ .

In this study, the Lagrange equations have been manually derived from the multibond graph, reformulated as a set of four ODEs and coded as a Scilab function [26]. With software packages supporting multibond graphs, the equations of motion can be automatically derived.

Multibond Graph of the Rotary Joint Manipulator

Since there is no translation, the multibond graphs of the two rigid bodies simplify (cf. to the multibond graph of a freely moving rigid body with two hinge points in Figure 8.16). The robot's base, body 0, does not move at all. As a result, Figure 12.58 gives a multibond graph of the rotary joint manipulator.

Derivation of the Standard Form of Robot Equations from the Multibond Graph

In the multibond graph of Figure 12.58, the vector ω_1^1 (ω_2^2) denotes the angular velocity of body 1 (body 2) with respect to reference frame 0 expressed in coordinates of reference frame 1 (reference frame 2). Actually,

$$\omega_1^1 = \omega_1^0 = [0 \ \dot{\Theta}_1 \ 0]^T \quad (12.56)$$

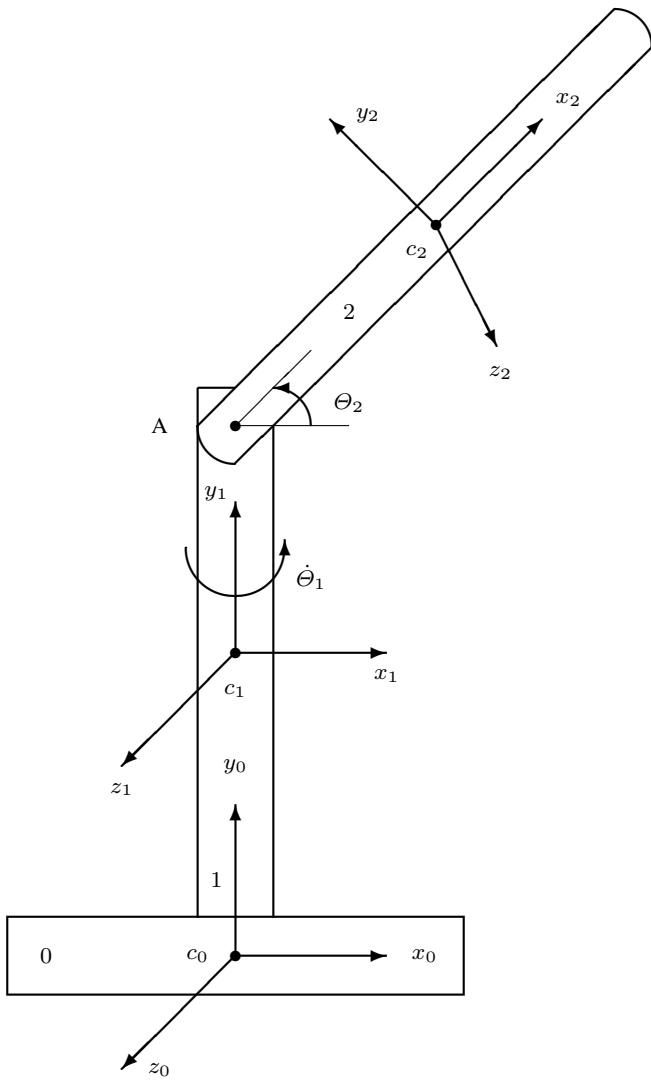


Fig. 12.57 Schematic of a two degrees of freedom rotary joint manipulator

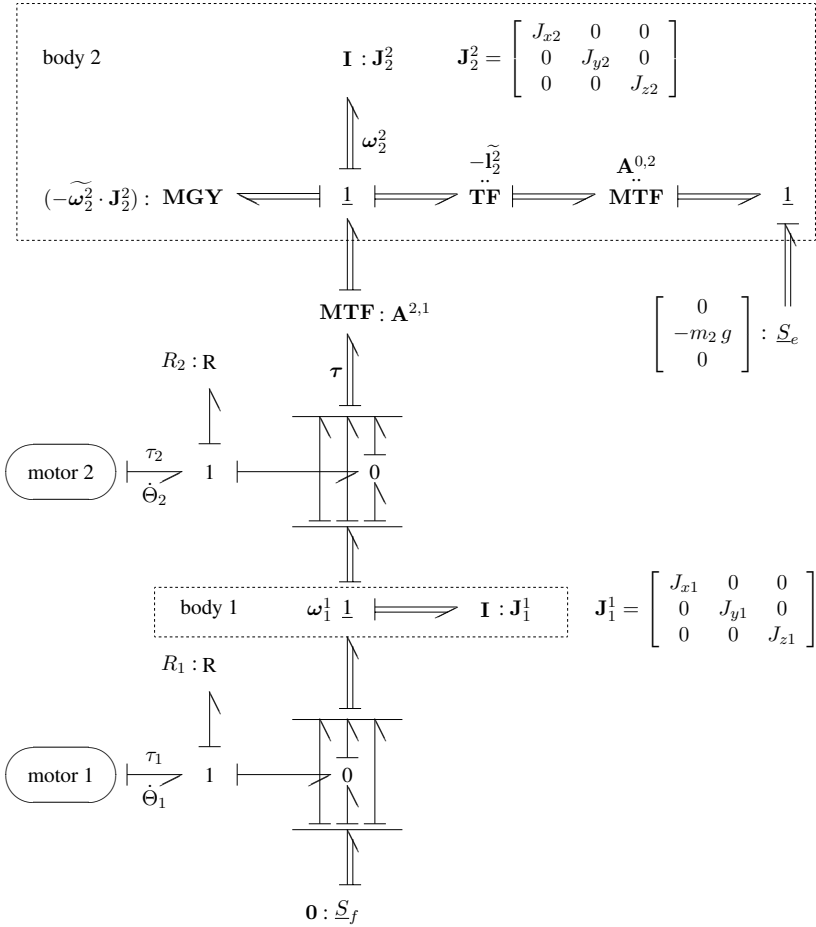


Fig. 12.58 Multibond graph of the two degrees of freedom rotary joint manipulator

where $\dot{\theta}_1$ denotes the angular velocity of motor 1. Matrix \mathbf{J}_1^1 (\mathbf{J}_2^2) denotes the inertia matrix of body 1 (body 2) with respect to the reference frame sitting in the centre of gravity of body 1 (body 2). Both matrices are diagonal as the axes of the body fixed reference frames are parallel to the principal axes of inertia.

The modulated multiport transformer $\mathbf{MTF} : \mathbf{A}^{2,1}$ represents the transformation between the two reference frames of body 1 and body 2. Accordingly, the angular velocity of body 1 with respect to reference frame 0 expressed in coordinates of reference frame 2 reads

$$\omega_1^2 = \underbrace{\begin{bmatrix} \cos \Theta_2 & \sin \Theta_2 & 0 \\ -\sin \Theta_2 & \cos \Theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}^{2,1}} \omega_1^1 = \begin{bmatrix} \dot{\Theta}_1 \sin \Theta_2 \\ \dot{\Theta}_1 \cos \Theta_2 \\ 0 \end{bmatrix}. \quad (12.57)$$

Let $\omega_2^2 = [\omega_{x2} \ \omega_{y2} \ \omega_{z2}]^T$. Then according to the multibond graph in Figure 12.58,

$$\begin{bmatrix} \omega_{x2} \\ \omega_{y2} \\ \omega_{z2} \end{bmatrix} = \mathbf{A}^{2,1} \begin{bmatrix} 0 \\ \dot{\Theta}_1 \\ \dot{\Theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{\Theta}_1 \sin \Theta_2 \\ \dot{\Theta}_1 \cos \Theta_2 \\ \dot{\Theta}_2 \end{bmatrix}. \quad (12.58)$$

In the same way, the matrix $\mathbf{A}^{0,2}$ relates the absolute velocity of the centre of gravity of body 2, expressed in coordinates of the body fixed reference frame to the same velocity expressed in coordinates of the inertial frame sitting in the centre of gravity of body 0.

$$\mathbf{A}^{0,2} = \begin{bmatrix} \cos \Theta_2 & -\cos \Theta_1 \sin \Theta_2 \sin \Theta_1 \\ \sin \Theta_2 & \cos \Theta_2 & 0 \\ -\sin \Theta_1 \cos \Theta_2 & \sin \Theta_1 \sin \Theta_2 & \cos \Theta_1 \end{bmatrix} \quad (12.59)$$

Furthermore, l_2 denotes the distance of the centre of gravity of body 2, c_2 , from the hinge point A. The vector from the origin of reference frame 2 to hinge point A expressed in coordinates of frame 2 is $\mathbf{l}_2^2 = [-l_2 \ 0 \ 0]^T$ and $\tilde{\mathbf{l}}_2^2$ denotes the skew symmetric matrix generated by this vector. Consequently, according to the multibond graph in Figure 12.58, the moment acting on body 2 caused by the force of gravity is

$$m_2 g \begin{bmatrix} 0 \\ 0 \\ l_2 \cos \Theta_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & l_2 \\ 0 & -l_2 & 0 \end{bmatrix}}_{(-\tilde{\mathbf{l}}_2^2)^T} \underbrace{\begin{bmatrix} c \Theta_2 & s \Theta_2 & -s \Theta_1 c \Theta_2 \\ -c \Theta_1 s \Theta_2 & c \Theta_2 & s \Theta_1 s \Theta_2 \\ s \Theta_1 & 0 & c \Theta_1 \end{bmatrix}}_{(\mathbf{A}^{0,2})^T} \begin{bmatrix} 0 \\ m_2 g \\ 0 \end{bmatrix}, \quad (12.60)$$

where $c\theta := \cos \theta$ and $s\theta := \sin \theta$.

The moment acting on body 2 contributed by the Eulerian junction structure is

$$\begin{aligned} -\tilde{\omega}_2^2 \mathbf{J}_2^2 \omega_2^2 &= - \begin{bmatrix} 0 & \omega_{z2} & -\omega_{y2} \\ -\omega_{z2} & 0 & \omega_{x2} \\ \omega_{y2} & -\omega_{x2} & 0 \end{bmatrix} \underbrace{\begin{bmatrix} J_{x2} & 0 & 0 \\ 0 & J_{y2} & 0 \\ 0 & 0 & J_{z2} \end{bmatrix}}_{:= \mathbf{J}_2^2} \begin{bmatrix} \omega_{x2} \\ \omega_{y2} \\ \omega_{z2} \end{bmatrix} \\ &= \begin{bmatrix} J_{z2} \omega_{y2} \omega_{z2} & -J_{y2} \omega_{y2} \omega_{z2} \\ J_{x2} \omega_{x2} \omega_{z2} & -J_{z2} \omega_{x2} \omega_{z2} \\ J_{y2} \omega_{x2} \omega_{y2} & -J_{x2} \omega_{x2} \omega_{y2} \end{bmatrix}. \end{aligned} \quad (12.61)$$

Let $\boldsymbol{\tau} = [\tau_{x1} \ \tau_{y1} \ \tau_{z1}]^T$. Then, summation of moments at the 1-junction of ω_2^2 and transformation across the multiport transformer $\mathbf{MTF} : \mathbf{A}^{21}$, gives

$$\boldsymbol{\tau} = (\mathbf{A}^{2,1})^T (\mathbf{J}_2^2 \dot{\omega}_2^2 + (-\widetilde{\omega}_2^2) \mathbf{J}_2^2 \omega_2^2 + m_2 g \begin{bmatrix} 0 \\ 0 \\ l_2 \cos \Theta_2 \end{bmatrix}). \quad (12.62)$$

Finally, summation of torques at the 1-junction of $\dot{\Theta}_2$ yields

$$\tau_2 = R_2 \dot{\Theta}_2 + \tau_{z1}. \quad (12.63)$$

Expanding this equations by using Equations 12.62, 12.61, 12.58 and 12.57 gives one of the two Lagrange equations describing the motion of the rotary joint manipulator.

$$\tau_2 = R_2 \dot{\Theta}_2 + \underbrace{J_{z2}}_{=: m_{22}} \ddot{\Theta}_2 - \underbrace{(J_{x2} - J_{y2})(\sin \Theta_2 \cos \Theta_2)}_{=: C_{21}} \dot{\Theta}_1^2 + \underbrace{m_2 g l_2 \cos \Theta_2}_{=: g_{21}} \quad (12.64)$$

The second Lagrange equation is obtained in the same manner. Summation of torques on the 1-junction of $\dot{\Theta}_1$ gives

$$\tau_1 = R_1 \dot{\Theta}_1 + J_{y1} \ddot{\Theta}_1 + \tau_{y1}. \quad (12.65)$$

After evaluation of Equation 12.62, the torque τ_{y1} becomes

$$\begin{aligned} \tau_{y1} &= [\sin \Theta_2 \ \cos \Theta_2 \ 0] \left(\begin{bmatrix} J_{x2} \dot{\omega}_{x2} \\ J_{y2} \dot{\omega}_{y2} \\ J_{z2} \dot{\omega}_{z2} \end{bmatrix} \right. \\ &\quad \left. + \begin{bmatrix} J_{z2} \omega_{y2} \omega_{z2} - J_{y2} \omega_{y2} \omega_{z2} \\ J_{x2} \omega_{x2} \omega_{z2} - J_{z2} \omega_{x2} \omega_{z2} \\ J_{y2} \omega_{x2} \omega_{y2} - J_{x2} \omega_{x2} \omega_{y2} \end{bmatrix} + m_2 g \begin{bmatrix} 0 \\ 0 \\ l_2 \cos \Theta_2 \end{bmatrix} \right) \\ &= J_{x2} \dot{\omega}_{x2} \sin \Theta_2 + J_{y2} \dot{\omega}_{y2} \cos \Theta_2 \\ &\quad + (J_{z2} - J_{y2}) \omega_{y2} \omega_{z2} \sin \Theta_2 \\ &\quad + (J_{x2} - J_{z2}) \omega_{x2} \omega_{z2} \cos \Theta_2. \end{aligned} \quad (12.66)$$

Finally, observing Equation 12.58, the second Lagrange equation reads

$$\begin{aligned} \tau_1 &= R_1 \dot{\Theta}_1 + \underbrace{(J_{y1} + J_{x2} \sin^2 \Theta_2 + J_{y2} \cos^2 \Theta_2)}_{m_{11}} \ddot{\Theta}_1 \\ &\quad + \underbrace{(2(J_{x2} - J_{y2}) \sin \Theta_2 \cos \Theta_2)}_{b_{11}} \dot{\Theta}_1 \dot{\Theta}_2. \end{aligned} \quad (12.67)$$

That is, Equation 12.55 takes the form

$$\underbrace{\begin{bmatrix} m_{11} & 0 \\ 0 & m_{22} \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \ddot{\Theta}_1 \\ \ddot{\Theta}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} b_{11} \\ 0 \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} \dot{\Theta}_1 \dot{\Theta}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ c_{21} & 0 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \dot{\Theta}_1^2 \\ \dot{\Theta}_2^2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ g_{21} \end{bmatrix}}_{\mathbf{G}(\boldsymbol{\Theta})} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} R_1 \dot{\Theta}_1 \\ R_2 \dot{\Theta}_2 \end{bmatrix}. \quad (12.68)$$

Notice that the entries in the matrices \mathbf{B} and \mathbf{C} are zero when either $\Theta_2 = 0$ or $\Theta_2 = \pi/2$.

Simulation of the Robot’s Motion

For a simulation of the robot’s motion, the parameters listed in Table 12.11 have been used [29]. In this table, c_i denotes the centre of gravity of body i . Given the parameters in Table 12.11, Figure 12.59 displays the time evolution of the angular velocities ω_1, ω_2 and of the angle Θ_2 . As can be seen from Figure 12.59, with increasing time, the angular velocity of body 1 takes a steady state value of about 0.66 rad , ω_2 becomes zero and consequently, takes a constant value of about -0.1 . These values can be verified. Under the conditions $\omega_1 = \text{const.}$ and $\omega_2 = 0$, the dynamic Equation 12.67 reduces to

$$\omega_1 = \frac{\tau_1}{R_1} = 1.0/1.5 = 0.66 \text{ rad/s}. \quad (12.69)$$

When $\Theta_2 = \text{const.}$, Equation 12.64 reduces to a nonlinear algebraic equation for Θ_2 .

$$\tau_2 = m_2 g l_2 \cos \Theta_2 - (J_{x2} - J_{y2})(\sin \Theta_2 \cos \Theta_2) \dot{\Theta}_1^2 \quad (12.70)$$

A numerical solution of Equation 12.70 is $\Theta_2 = -0.0999574 \text{ rad}$.

Table 12.11 Parameter values used for simulation of the robot’s motion

Parameter	Value	Units	Meaning
J_{y1}	6	kgm^2	Moment of inertia of body 1 with respect to c_1
J_{x2}	5	kgm^2	Moment of inertia of body 2 with respect to c_2
J_{y2}	10	kgm^2	Moment of inertia of body 2 with respect to c_2
J_{y2}	10	kgm^2	Moment of inertia of body 2 with respect to c_2
m_2	40	kg	Mass of body 2
l_2	0.5	m	Distance c_2 - pivot point A (cf. Figure 12.57)
R_1	1.5	Nms/rad	Friction between body 0 and body 1
R_2	2.0	Nms/rad	Friction in pivot point A
τ_1	1.0	Nm	Torque acting on body 1
τ_2	195	Nm	Torque acting on body 2

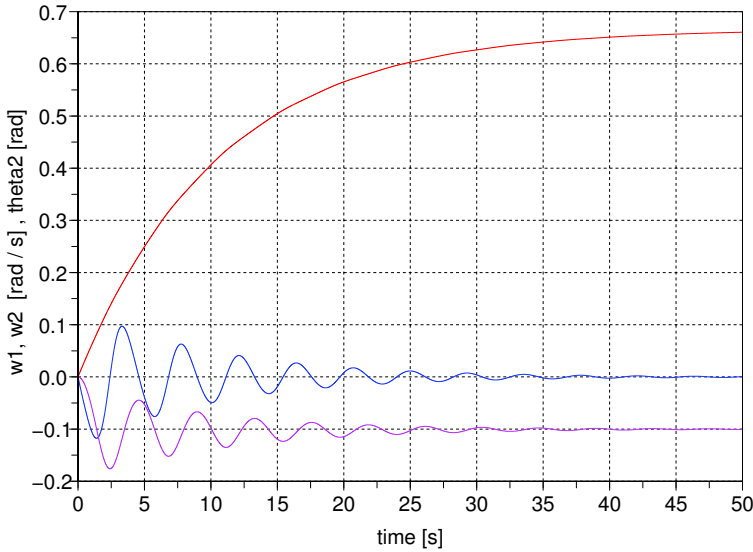


Fig. 12.59 Time evolution of the angular velocities ω_1 , ω_2 and of the angle Θ_2

12.10 Fluid Level Control in a Three Tank System

One of the usual tasks in process engineering systems is to control the fluid level in tanks. As an example, the system of three coupled tanks depicted in Figure 12.60 is considered. It is assumed that

- the fluid flow is one way from left to right,
- the fluid inertia can be neglected,
- there is a uniform hydrostatic pressure at the bottom of each tank,
- isothermal conditions apply,
- hydraulic power can be approximated by the product of hydrostatic pressure and volume flow.

In this example, the task is to control the fluid level in the last right-hand side tank. To that end, the level is sensed and fed into a PID controller that controls the fluid flow supply into the first tank as shown in Figure 12.60. In [15], Hoffmann starts from the equations and presents a MATLAB[®]/Simulink[®] simulation of this example.

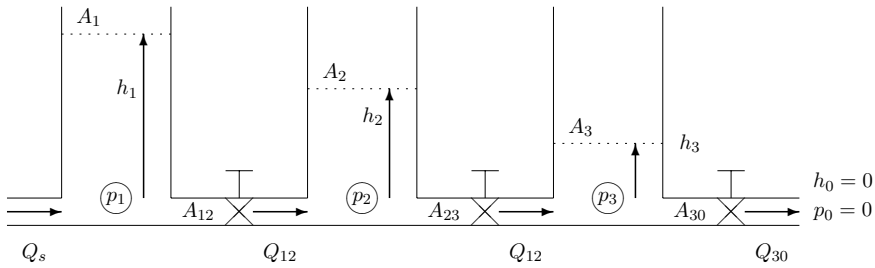


Fig. 12.60 Schematic of a three tank system

Bond Graph Model of the Three Tank System

Bond graph modelling of this hydraulic plant is straightforward. The valves in the pipes connecting the tanks are modelled by resistors. Their constitutive relation is given by Bernoulli’s law. The pressure at the bottom of the tanks is represented by a 0-junction and the storage of potential energy in the tanks is taken into account by a C element. Accordingly, Figure 12.61 represents a bond graph model of the three tank system.

Let A_{ij} be the cross section area of the valve between pressures p_i and p_j . According to Bernoulli’s law, the volume flow Q_{ij} through the valve equals

$$\begin{aligned}
 Q_{ij} &= A_{ij} \operatorname{sign}(\Delta p_{ij}) \sqrt{\frac{2}{\rho} |\Delta p_{ij}|} \\
 &= A_{ij} \operatorname{sign}(\Delta h_{ij}) \sqrt{2g |\Delta h_{ij}|} .
 \end{aligned}
 \tag{12.71}$$

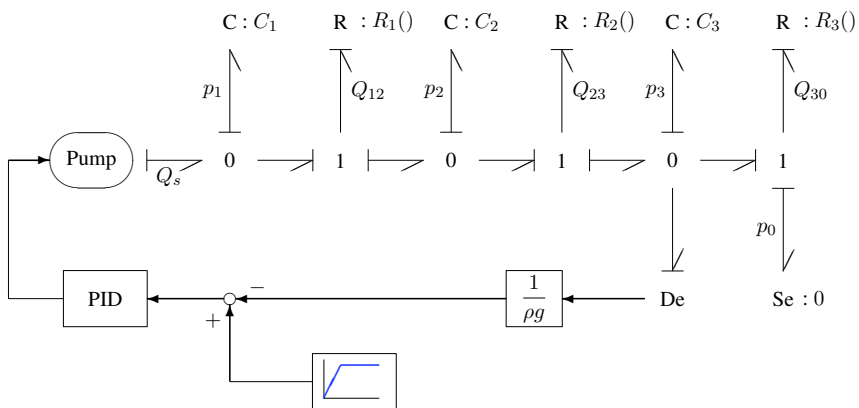


Fig. 12.61 Bond graph of the three tank system

Dynamic Equations of the Three Tank System

The dynamic equations are immediately derived from the bond graph of Figure 12.61.

$$\dot{p}_1 = \frac{1}{C_1} (Q_s - Q_{12}) \quad (12.72a)$$

$$\dot{p}_2 = \frac{1}{C_2} (Q_{12} - Q_{23}) \quad (12.72b)$$

$$\dot{p}_3 = \frac{1}{C_1} (Q_{23} - Q_{30}) \quad (12.72c)$$

An equal cross section area for all three valves implies that in steady state

$$h_2 = 2h_3 \quad (12.73a)$$

$$h_1 = 3h_3 \quad (12.73b)$$

(as depicted in Figure 12.60) and

$$Q_s = Q_{12} = Q_{23} = Q_{30} . \quad (12.74)$$

Adopting the values $Q_s = 1 \text{ m}^3/\text{s}$ and $A_{12} = 0.1 \text{ m}^2$ results in the steady state fluid levels

$$h_1 = 15.29 \text{ m} , \quad h_2 = 10.19 \text{ m} , \quad h_3 = 5.097 \text{ m} . \quad (12.75)$$

Simulation of the Uncontrolled Three Tank System

Simulation of the uncontrolled system's behaviour confirms these steady values (cf. Figure 12.62 and Figure 12.63). Numerical values used for simulation of the uncontrolled systems are listed in Table 12.12.

Simulation of the Controlled Three Tank System

The transfer function of the PID controller is used in the form

Table 12.12 Parameters for simulation of the uncontrolled three tank system

Parameter	Value	Units	Meaning
$A_1 = A_2 = A_3$	1.999	m^2	Cross section area of the tanks
$A_{12} = A_{23} = A_{30}$	0.1	m^2	Cross section area of the valves
ρ	780	kg/m^3	Fluid density
Q_s	1	m^3/s	Volume flow of the supply

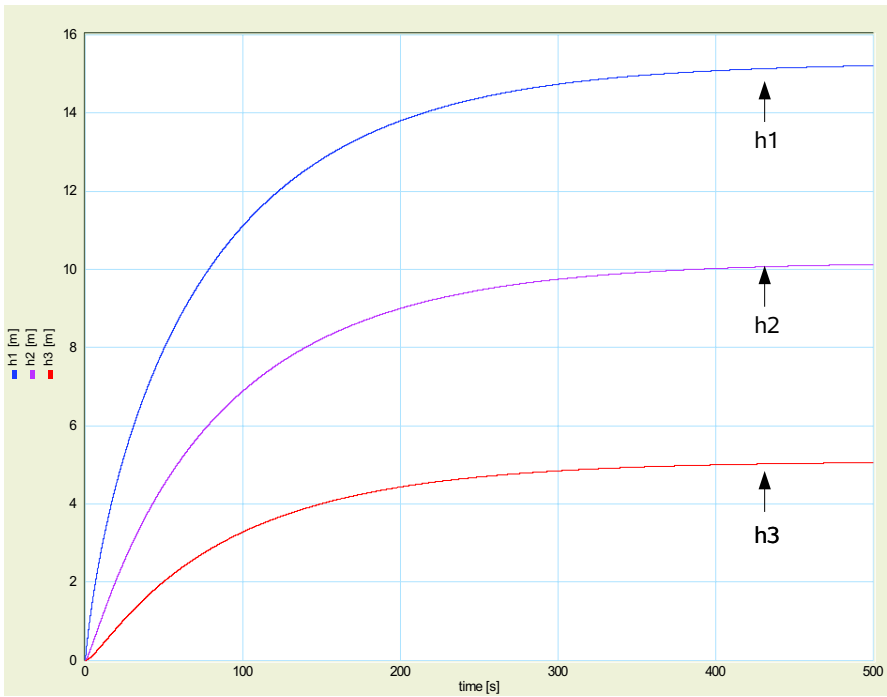


Fig. 12.62 Time evolution of fluid levels in the uncontrolled three tank system due to a step of Q_s

$$U(s) = K \left[1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{N} s} \right] E(s), \quad (12.76)$$

where $s \in \mathbb{C}$ and E and U denote the Laplace transforms of the input error and the controller output. Table 12.13 gives the parameters of the PID controller.

If details of the hydraulic power supply subsystem are known, a bond graph can be developed for the submodel called pump in the bond graph of Figure 12.61. Otherwise, it may be appropriate and sufficient to approximate the pump’s delay in response to an immediate step in the controller signal by a first order lag signal element and to account for saturation of the volume flow Q_s by a satura-

Table 12.13 Parameters of the PID controller

Parameter	Value	Units	Meaning
K	5		Proportional gain
Ti	50	s	Integral time constant
Td	3	s	Derivative time constant
N	20		Derivative gain limitation

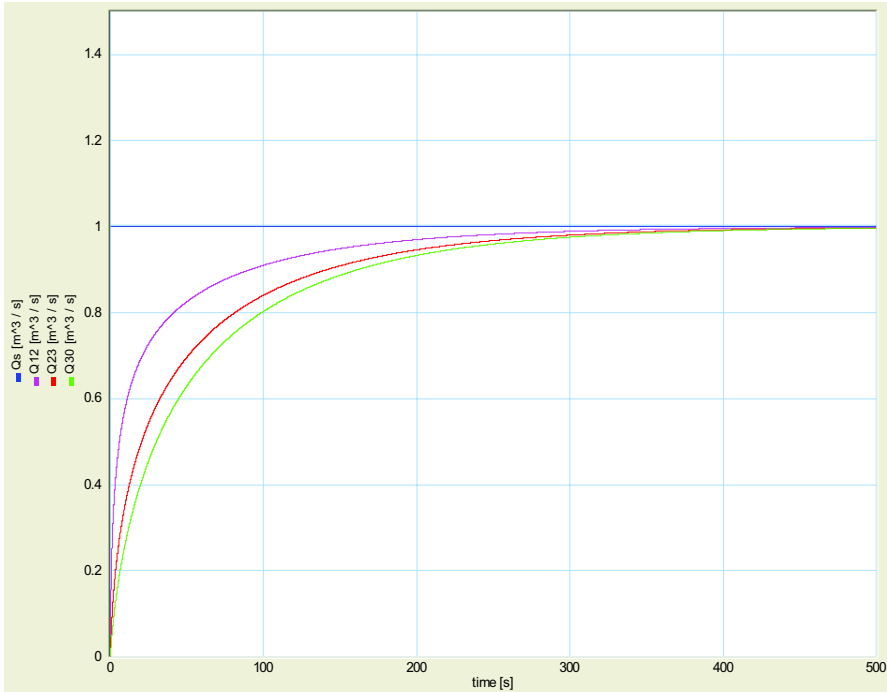


Fig. 12.63 Time evolution of volume flows in the uncontrolled three tank system

tion signal block. The first order lag element is described by a transfer function $G(s) := k/(\tau s + 1)$, where k denotes the proportional gain and τ the time constant. Parameters of these two signal blocks are given in Table 12.14.

Figure 12.64 shows the time history of the fluid levels in the tanks if all of them are initially empty and if the set point for the fluid level in the third tank is 1 m. Again, in steady state, Equations 12.73a–12.73b hold. Figure 12.65 shows the dynamics of the volume flows in the controlled system.

According to Equation 12.71 and given the set point of 1 m for the fluid level h_3 , a steady state value of

$$Q_{30} = A_{30} \sqrt{2g} = 0.1 \sqrt{2 \times 9.81} = 0.4429 \text{ m}^3/\text{s} \tag{12.77}$$

Table 12.14 Parameters of the first order lag element and the saturation block

Parameter	Value	Units	Meaning
k_p	1		Proportional gain
τ	1	s	Time constant
min	0	m^3/s	lower bound
max	2.5	m^3/s	upper bound of the saturation block

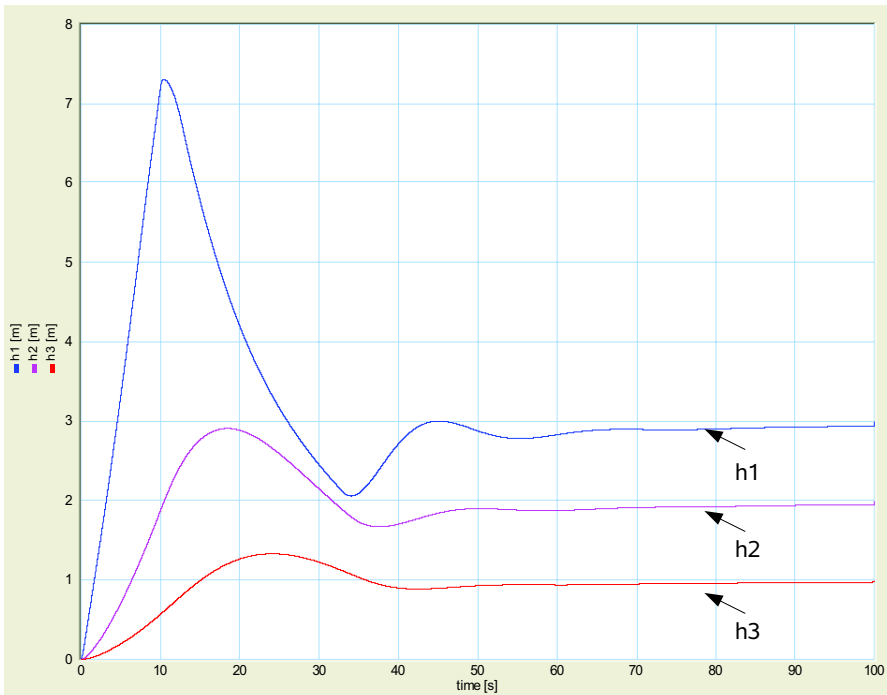


Fig. 12.64 Time evolution of fluid levels in the controlled three tank system

is obtained. This and Equation 12.74 is confirmed by the simulation results in Figure 12.65. Figure 12.66 shows the history of the volume flows in case the set point for the fluid level in tank 3 rises linearly to its value of 1 m within a time interval of 10 s . The simulation results displayed in Figure 12.66 agree with those given in [15].

For further reading on bond graph modelling of controlled hydraulic systems, refer to the textbook of Dransfield [9].

12.11 Fault Detection in a Hydraulic Two Tank System

This section illustrates the bond graph model-based approach to FDI introduced in Section 6.8 by application to the simple hydraulic two tank system displayed in Figure 6.12 and reproduced in Figure 12.67 for the sake of convenience. It is assumed that the pressures in both tanks are measured. For simplicity, only the mass flow is considered. Associated thermal convection is not taken into account.

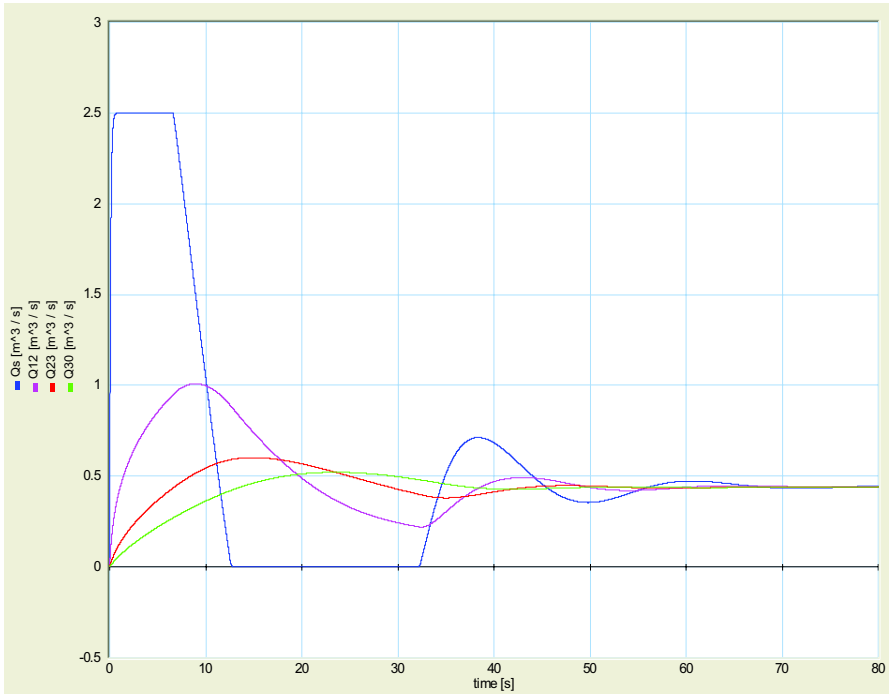


Fig. 12.65 Dynamics of the volume rates in the controlled three tank system

Bond Graph Models of the Two Tank System Coupled by Residual Sinks

Figure 12.68 shows a behavioural model of the process subject to faults (lower part of Figure 12.68) coupled to a model of the faultless process (upper part of Figure 12.68) by modulated effort sources and residual flow sinks. The measuring of the tank pressures has been taken into account by effort detectors (De-elements) attached to the 0-junctions. A fault such as leakage from a tank can be introduced into the model of the real process by switching the modulated flow sinks, MSf, attached to the 0-junctions of the tank pressures on and off.

A partial blockage of a valve results in a reduction of the valve's parameter $k = c_d A_V(t) \sqrt{2/\rho}$. Hence, k is a function of time, t , that takes into account the way in which the valve blocks. This may take place abruptly or progressively. Accordingly, the valves are represented by modulated resistors in the model of the *faulty* process. As a result, the model of the real process differs from that of the faultless model by these elements allowing, for user introduced faults.

In the integrated model, all energy stores take preferred integral causality. Due to the residual sinks, the underlying mathematical model is a DAE system.

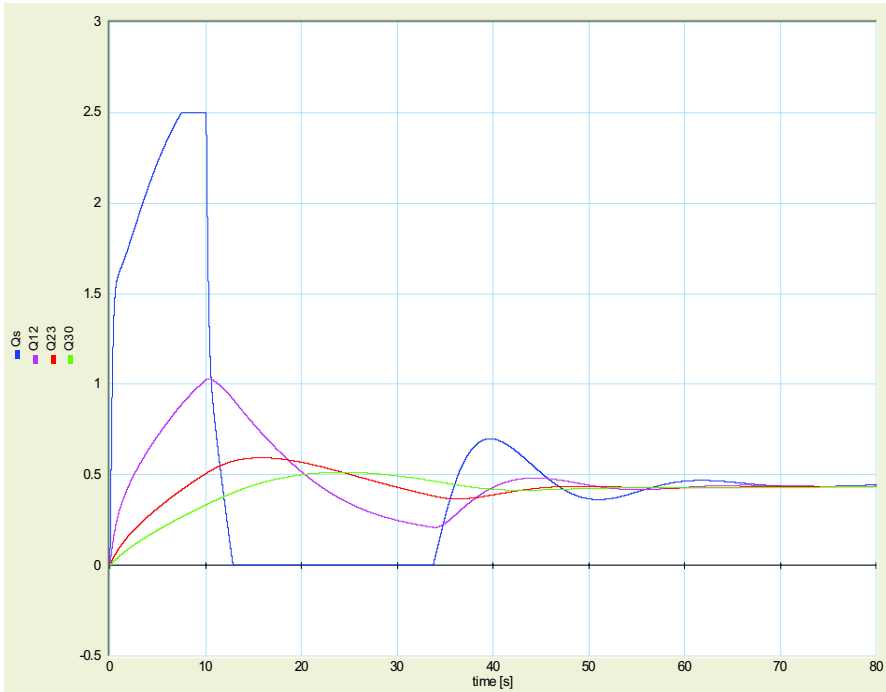


Fig. 12.66 History of the volume rates in the controlled three tank system in case the controller's set point rise is limited

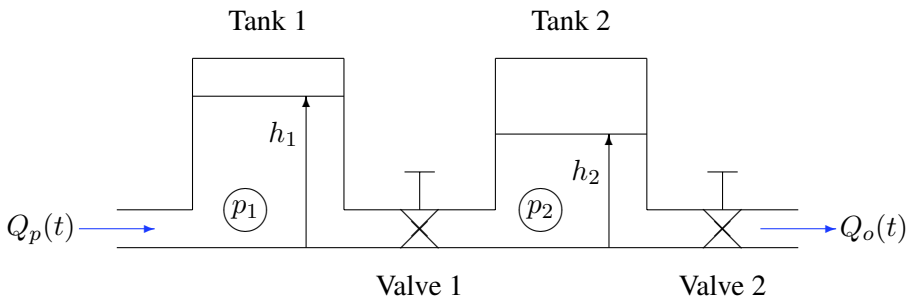


Fig. 12.67 Schematic of a hydraulic two tank system reproduced from Figure 6.12

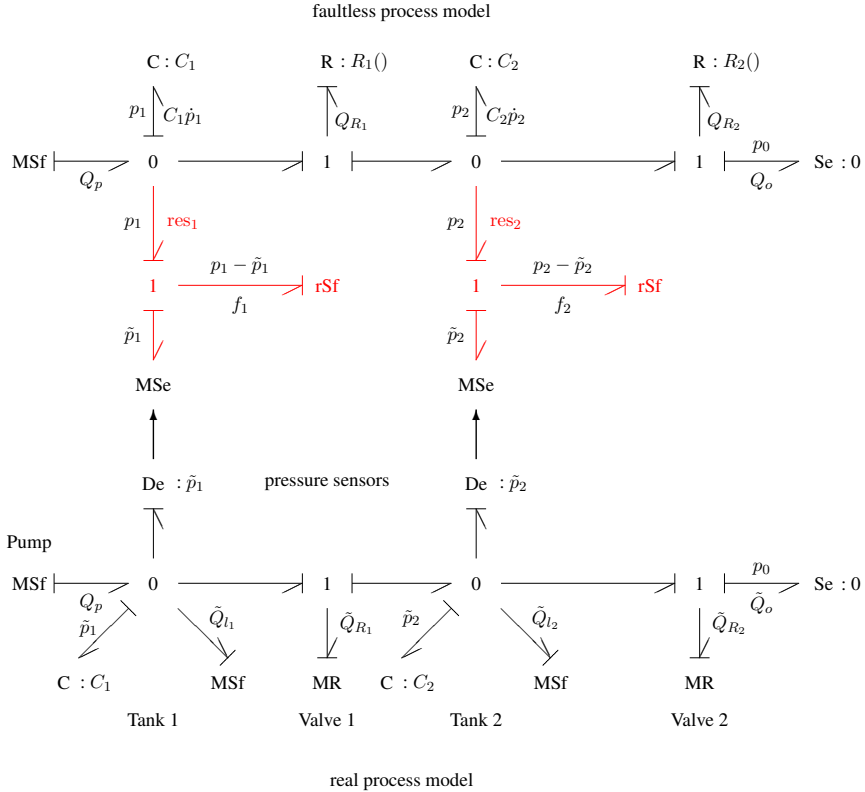


Fig. 12.68 Coupling of a faulty process model (lower part) to a faultless process model (upper part) by means of residual flow sinks

The DAE System Derived from the Coupled Bond Graphs

Derivation of model equations from the causal bond graph in Figure 12.68 is straightforward. In this case study, they have been formulated in Scilab's mathematical input language and stored in a script to be read by Scilab. The Scilab function of the DAE system of the overall model to be passed in a call to the solver DASSL is displayed in Figure 12.69.

The application of the solver DASSL requires that all equations are written in implicit form. In the Scilab script, the residuals $r(i)$, $i = 1, \dots, 6$, are not to be confused with the residuals of the ARRs to be computed. The latter variables are denoted f_1, f_2 . In the behavioural model of the real engineering process, perturbed power variables are denoted by a name that starts with the letter t standing for tilde.

With $\mathbf{x}_p := [p_1, p_2, \tilde{p}_1, \tilde{p}_2]^T$ and $\mathbf{w} := [f_1, f_2]$ as components of a descriptor vector, the DAE system of the example, in fact, is a semi-explicit DAE of the form of Equation 6.105. Its index is 2. In this example, the matrices in Equation 6.106

```

// Scilab function including the DAE system of the two tank models
// coupled by two residual flow sinks
function [res, ires] = daesys(t,x,xdot)

// components of the descriptor vector x = [x_p, w]:
// tank pressures (faultless process model):
p1 = x(1)
p2 = x(2)
// tank pressures (faulty process model):
tp1 = x(3)
tp2 = x(4) // x_p := [ p1,p2,tp1,tp2 ]
// residuals:
f1 = x(5)
f2 = x(6) // w := [ f1, f2 ]

// time derivatives of the components of the descriptor vector:
dp1 = xdot(1)
dp2 = xdot(2)
dtp1 = xdot(3)
dtp2 = xdot(4)
df1 = xdot(5)
df2 = xdot(6)

// system inputs: volume flow of the feed pump
Qp = Flow*pulse(t,tstart,tstop)

// no leakage from the two tanks of the real process:
tQ11 = 0.0
tQ12 = 0.0

// volume flows through the valves:
QR1 = orifice(AV1,p1,p2)
QR2 = orifice(AV2,p2,p0)

// partial blockage of valve 1 for 50.0s <= t <= 60.0s:
tQR1 = (1.0 - pulse2(t,50.0,60.0,0.8))*orifice(AV1,tp1,tp2)
tQR2 = orifice(AV2,tp2,p0)

// continuity equations for the tanks in both submodels:
r(1) = Qp - QR1 - C1*dp1 - f1 // p1
r(2) = QR1 - QR2 - C2*dp2 - f2 // p2
r(3) = Qp - tQR1 - tQ11 - C1*dtp1 // tp1
r(4) = tQR1 - tQR2 - tQ12 - C2*dtp2 // tp2

// equations of the residual flow sinks:
r(5) = p1 - tp1 // f1
r(6) = p2 - tp2 // f2

ires = 0 // indicator of successful computation of r
endfunction

```

Fig. 12.69 Scilab function including the DAE system of the two tank models coupled by residual flow sinks

take the form

$$\frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_p} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad (12.78)$$

and

$$\frac{\partial \mathbf{f}_1}{\partial \mathbf{w}} = \begin{bmatrix} -1/C_1 & 0 \\ 0 & -1/C_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (12.79)$$

Consequently,

$$\det \begin{pmatrix} \frac{\partial \mathbf{f}_2}{\partial \mathbf{x}_p} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{w}} \end{pmatrix} = \det \begin{bmatrix} -1/C_1 & 0 \\ 0 & -1/C_2 \end{bmatrix} = \frac{1}{C_1 C_2} \neq 0. \quad (12.80)$$

Simulation of the Faultless System Behaviour

The numerical solution of a DAE system requires a consistent set of initial conditions for the components of the descriptor vector and their time derivatives. For the consistent initialisation of a DAE system, the algorithm of Pantelides [22] can be used. To facilitate the specification of a consistent set of initial conditions, it is assumed that the two tanks are empty at initial time $t = 0$ and that the pump delivers a constant volume flow, Q_p , for the time period $10.0 \text{ s} \leq t \leq 40.0 \text{ s}$. That is, the empty tanks are filled for 30 s . Thereafter, they discharge at a rate depending on how much the valves are open. Figure 12.70 depicts the undisturbed dynamic behaviour. The parameters of the hydraulic two tank system are given in Table 12.15.

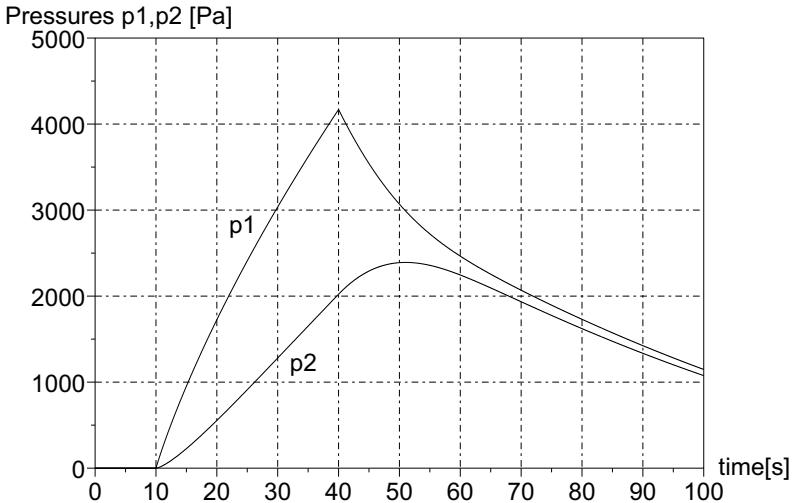


Fig. 12.70 Time history of the tank pressures in faultless operation mode

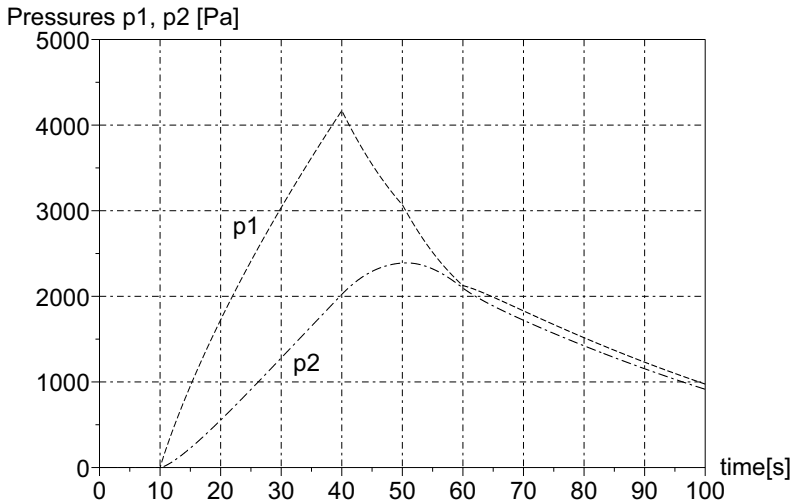
Table 12.15 Parameters of the hydraulic two tank system

Parameter	Value	Units	Meaning
g	9.81	m/s^2	Gravitational acceleration
ρ	780	kg/m^3	Oil density
c_d	0.61		Discharge coefficient
$A_{T_1} = A_{T_2}$	0.153	m^2	Cross sectional area of the tanks
$C_1 = C_2 = A_{T_1}/(\rho g)$		m^3/Pa	Capacitances of the tanks
A_{V_1}	$0.2 \cdot 10^{-2}$	m^2	Cross sectional area of valve 1
A_{V_2}	$0.1 \cdot 10^{-2}$	m^2	Cross sectional area of valve 2
Flow	$0.5 \cdot 10^{-2}$	m^3/s	Volume flow of the pump
Q_l	$0.1 \cdot 10^{-2}$	m^3/s	Leakage from tank 1
p_0	0.0	Pa	Pressure of the environment

Study of Fault Scenarios

In this case study, two types of faults are considered, namely leakage from the tanks and partial blockage of the valves. As a first fault scenario, a constant leakage flow from tank 1 is assumed to be effective for the time period $50 s \leq t \leq 60 s$, while the two tanks discharge. As a result, the pressures in the tanks decrease at a higher rate during this time period. Figure 12.71 shows the time history of the tank pressures in the case of a leakage from tank 1.

Leakage from tank 1 corresponds to a decrease of the area of its bottom. According to the fault signature matrix of Table 6.5, residual res_1 is affected, while residual

**Fig. 12.71** Time history of the tank pressures in the case of a leakage from tank 1

res_2 is not. Figures 12.72 and 12.73 depicting the residuals f_1 and f_2 validate this expectation. Note that at $t = 60 s$, the leakage from tank 1 abruptly stops. The system abruptly returns to normal mode operation. Accordingly, residual f_1 abruptly drops to zero.

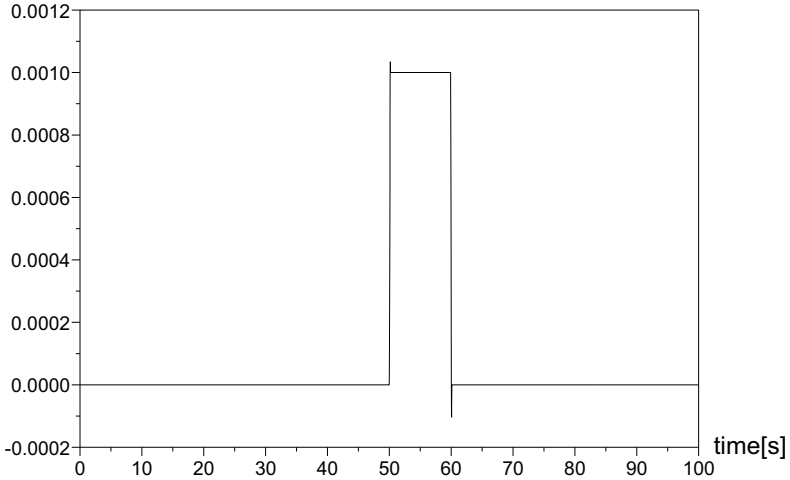


Fig. 12.72 Residual f_1 in the case of a leakage from tank 1 during the time interval $50 s \leq t \leq 60 s$

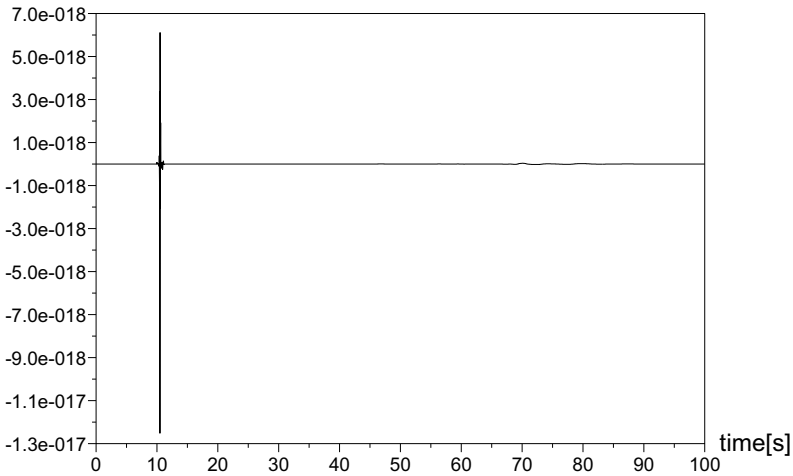


Fig. 12.73 Residual f_2 in the case of a leakage from tank 1

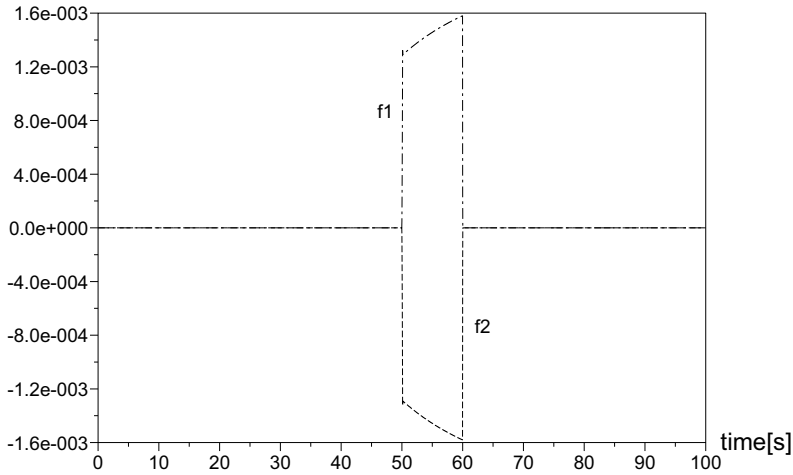


Fig. 12.74 Residuals f_1 and f_2 in the case of partial blockage of the valve 1 during the time interval $50\text{ s} \leq t \leq 60\text{ s}$

As a second fault scenario, partial blockage of the valve between the two tanks is assumed to be effective during the time interval $50\text{ s} \leq t \leq 60\text{ s}$ after the constant flow pump has been switched off. Consequently, the pressure in tank 1 decreases at a lower rate, while the pressure in tank 2 decreases at a higher rate. Its outlet is not affected, but its inlet is choked. According to the fault signature matrix 6.5, both residuals should be sensitive to this type of fault. This is verified by Figure 12.74.

12.12 Heated Stirred Tank

Bond graph modelling of thermal systems is the subject of a book by Thoma and Bouamama [27]. In this section, the simple example of a single heated stirred tank considered in Section 10.1.1 is taken as a subject of a small modelling and simulation study (cf. [13, 29]). The schematic of the tank and a pseudo bond graph model are redisplayed in Figure 12.75 and Figure 12.76.

The following assumptions apply.

- The fluid flow is one way from left to right.
- The mass flow can be considered incompressible; inertia effects can be neglected.
- The mass flow \dot{m}_i and the temperature T_i at the tank inlet are constant.
- There is a uniform hydrostatic pressure at the bottom of the tank.
- The fluid in the tank is heated. The heat is supplied at constant rate. Stirring ensures a spatially uniformly distributed temperature T .
- Heat losses to the ambient and the heat capacity of the tank wall can be neglected.

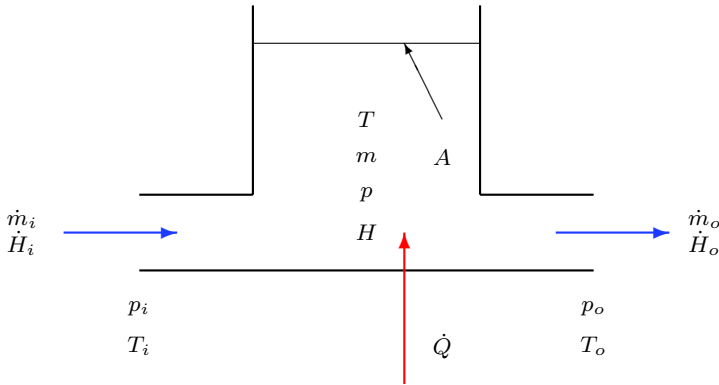


Fig. 12.75 Heated stirred tank

Constitutive Relations of the Elements

As the pressure at the hydraulic inlet resistor only affects the (ideal) source of hydraulic power supply, it can be omitted. The constitutive relation (CR) of the hydraulic outlet resistor is

$$\dot{m}_o = k_2 \sqrt{p}. \quad (12.81)$$

The constitutive relation of the hydraulic capacitor is

$$p = \frac{1}{C_h} m, \quad (12.82)$$

where $C_h = A/\rho$.

The thermal capacitor's constitutive relation is

$$T = \frac{1}{C_{therm}} H, \quad (12.83)$$

where $C_{therm} = c \times m$.

The constitutive relation of the thermal resistors is

$$\dot{H}_{index} = c \dot{m}_{index} T_{index}, \quad (12.84)$$

where index either denotes the inlet or the outlet resistor.

Table 12.16 gives the parameters of the simulation study (cf. [29]). The fluid flow enters the tank at a mass flow of 4 kg/s. At time instance $t = 1500$ s, this value increases to 4.8 kg/s. That is,

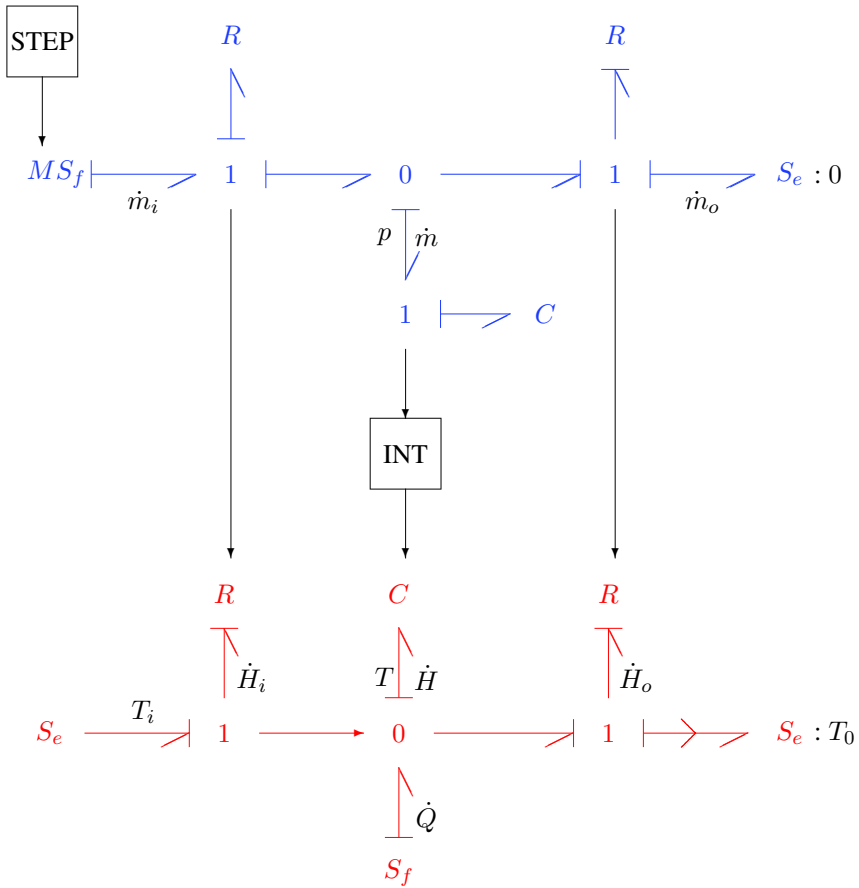


Fig. 12.76 Pseudo bond graph of the heated stirred tank

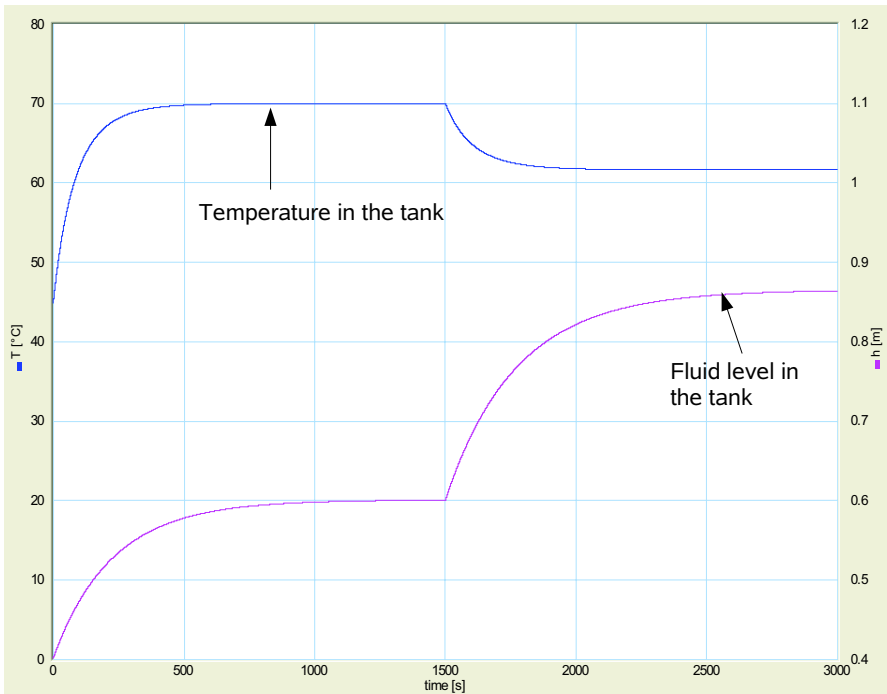
$$\dot{m}_i = \begin{cases} 4.0 \text{ kg/s} & 0 \leq t < 1500 \text{ s} \\ 4.8 \text{ kg/s} & t \geq 1500 \text{ s} \end{cases} \quad (12.85)$$

Simulation Results

Figure 12.77 shows the time evolution of the temperature and the fluid level in the tank. As can be seen from Figure 12.77, the fluid level as well as the temperature in the tank increase due to an inflow at constant mass flow and due to constant heating of the fluid. Then, due to the immediate increase of the mass flow at time $t = 1500 \text{ s}$ and a continued unchanged constant heating of the fluid, the fluid level further increases while the temperature in the tank decreases.

Table 12.16 Parameters of the simulation study

Parameter	Value	Units	Meaning
g	9.81	m/s^2	gravitational acceleration
ρ	800	kg/m^3	Fluid density
c_p	200	$J/(kg \text{ } ^\circ C)$	specific heat
A	1.0	m^2	Cross section area of the tank
k_2	5.824×10^{-2}	$\sqrt{kg \text{ } m}$	Coefficient of the hydraulic outlet resistor
T_i	20	$^\circ C$	Inlet temperature
\dot{Q}	40	kW	Heat flow
$h(0)$	0.4	m	Initial fluid level in the tank
$T(0)$	45	$^\circ C$	Initial temperature in the tank
$H(0)$	2.88×10^6	J	Initial enthalpy in the tank

**Fig. 12.77** Time evolution of the temperature and the fluid levels in the tank

The steady state values of fluid level and temperature in the tank obtained by simulation can be easily manually verified. Summation of mass flows at the upper 0-junction of the bond graph in Figure 12.76 gives

$$\dot{p} = \frac{1}{C_h} (\dot{m}_i - \dot{m}_o)$$

$$\begin{aligned}
 &= \frac{1}{C_h} (\dot{m}_i - k_2 \sqrt{p}) \\
 &= \frac{1}{C_h} (\dot{m}_i - k_2 \sqrt{\rho g h}) .
 \end{aligned} \tag{12.86}$$

In steady state, the time derivative of the pressure vanishes. Hence,

$$h = \frac{\dot{m}_i^2}{k_2^2 \rho g} . \tag{12.87}$$

Using numerical values from Table 12.16 gives $h(t = 3000 \text{ s}) = 0.865 \text{ m}$.

The steady state value of the temperature in the tank is obtained accordingly. Summation of enthalpy flows at the lower 0-junction of the bond graph in Figure 12.76 in steady state gives

$$0 = \frac{1}{C_{therm}} [c_p \dot{m}_i T_i - c_p \dot{m}_o T + \dot{Q}] . \tag{12.88}$$

Hence,

$$T = T_i + \frac{1}{c_p \dot{m}_o} \dot{Q} . \tag{12.89}$$

The numerical result is $T(t = 3000 \text{ s}) = 61.66 \text{ }^\circ\text{C}$.

12.13 A Counterflow Heat Exchanger

This section illustrates how the pseudo bond graph approach can be conveniently used for modelling open thermodynamic systems. The example under study is a simple counterflow heat exchanger as is depicted in Figure 12.78. In [25], Scherf directly sets up the equations for a counterflow heat exchanger and uses them for a MATLAB[®]/Simulink[®] simulation. In this case study, the parameters given by Scherf are used. A bond graph model of a heat exchanger has also been presented by Thoma and his co-authors [27, 28]. It makes use of a non-standard element they call HEXA (Heat Exchanger).

The simple counterflow heat exchanger in Figure 12.78 can be considered as a tube carrying the cooling water with a counterflow hot oil stream passing through an inner tube. Both tubes have an inlet and an outlet. That is, each tube can be viewed as a control volume with a mass inflow and a mass outflow and can be presented by a pseudo bond graph similar to the one of a heated stirred tank in Section 10.1.1.

For simplicity, it is assumed that

- a one-dimensional concentrated parameter model is appropriate,
- no mass is accumulated in both tubes,
- hydraulic losses can be neglected in both tubes, and
- the wall of pipe enclosing the inner tube is perfectly insulating.

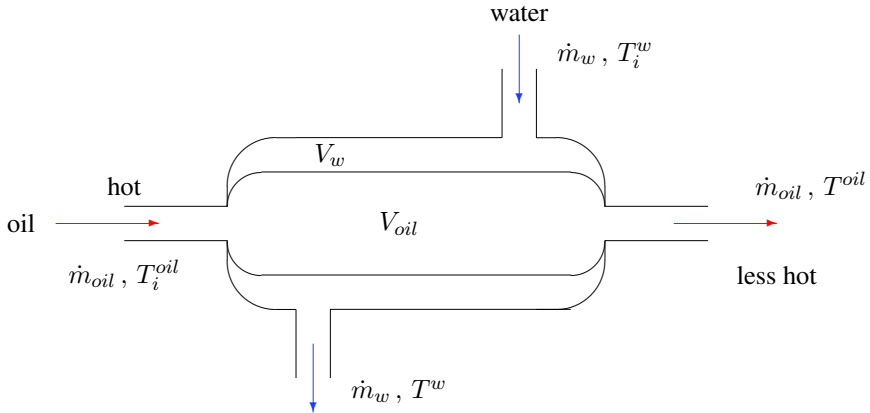


Fig. 12.78 Schematic of a counterflow heat exchanger

A Pseudo Bond Graph Model of the Counterflow Heat Exchanger

According to the pseudo bond graph model of a heated stirred tank (cf. Figure 10.5), Figure 12.79 shows a pseudo bond graph of the heat exchanger. The upper part of the pseudo bond graph model represents the outer tube with the cooling water. The lower part of same structure models the inner pipe carrying the hot oil stream. The hydraulic part of both submodels reduces to a 1-junction of the mass flow as mass accumulation and hydraulic losses have been neglected. Both submodels are coupled by a modulated resistor that accounts for the heat conduction from the hot inner pipe to the enclosing pipe. This R element is modulated by the upstream temperatures of water and oil. Its constitutive equations are

$$\Delta T = \frac{(T_i^{oil} - T^w) - (T^{oil} - T_i^w)}{\ln(T_i^{oil} - T^w) - \ln(T^{oil} - T_i^w)} \quad (12.90a)$$

$$\dot{H}^{oil} = k A \Delta T \quad (12.90b)$$

$$\dot{H}^w = k A \Delta T, \quad (12.90c)$$

where k denotes the thermal conductance coefficient (assumed to be constant) and A is the surface of the inner pipe effective in the heat exchange.

As pointed out in Section 10.1.1, the enthalpy flow into and out of each pipe does not depend on a temperature difference, but on the upstream temperature. Therefore, the bond with the downstream temperature is activated. Furthermore, the C element in the thermal part of both submodels is modulated, which is no problem in a pseudo bond graph.

The cooling water entering the heat exchanger is provided through a valve. In order to ensure a given constant temperature of the oil at the outlet, the actual tem-

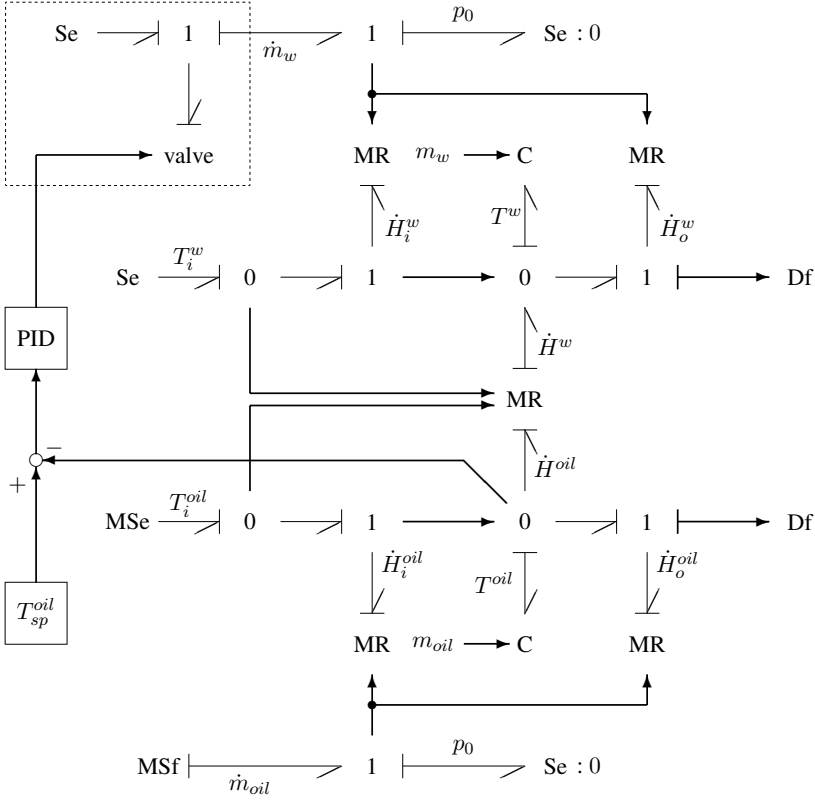


Fig. 12.79 Pseudo bond graph of the counterflow heat exchanger

perature is measured and compared with a set point value. The difference is fed into a PID controller that controls the opening of the water inlet valve. If details of the valve are known, a bond graph model can be developed for the valve. In this study, the limitation of the opening is taken into account by a signal saturation block. The dynamics of the valve opening are modelled by a first order lag signal block.

From the pseudo bond graph in Figure 12.79, the equations describing the dynamics of the heat exchanger are easily derived by summing up flows at the right-hand side 0-junctions. They are, in fact, power balances. The enthalpy flow entering the C element reads

$$\begin{aligned}
 c_{oil} m_{oil} \dot{T}^{oil} &= \dot{H}_i^{oil} - \dot{H}^{oil} - \dot{H}_o^{oil} \\
 &= c_{oil} \dot{m}_{oil} T_i^{oil} - k A \Delta T - \\
 &\quad c_{oil} \dot{m}_{oil} T^{oil}
 \end{aligned}
 \tag{12.91a}$$

Table 12.17 Parameters used for simulation runs

Parameter	Value	Units	Meaning
m_{oil}	75	kg	Oil mass in the heat exchanger
m_w	100	kg	Mass of water in the heat exchanger
A	5	m^2	Surface of inner pipe effective in heat conductance
k	85	$J/(s m^2 K)$	Thermal conductance coefficient
c_{oil}	1600	$J/(kgK)$	Specific heat of the oil at constant volume
c_w	4200	$J/(kgK)$	Specific heat of the water at constant volume
\dot{m}_{oil}	500	kg/h	Oil mass flow
T_i^{oil}	120	$^{\circ}C$	Oil temperature at the inlet
T_i^w	10	$^{\circ}C$	Water temperature at the inlet

$$c_w m_w \dot{T}^w = c_w \dot{m}_w T_i^w + k A \Delta T - c_w \dot{m}_w T^w, \quad (12.91b)$$

where c_{oil} and c_w denote the specific heat at constant volume of the oil and the water.

Simulation of the Uncontrolled Counterflow Heat Exchanger

In order to see whether the model correctly reflects the dynamic behaviour of the heat exchanger, first, the uncontrolled system has been simulated. To that end, the parameters in Table 12.17 (cf. [25]) have been used.

If in steady state the outlet temperatures are to be $T_{oil} = 33.8^{\circ}C$ and $T_w = 43.0^{\circ}C$ respectively, then for the cooling water flow, the required mass flow is $\dot{m}_w = 0.1382 kg/s$. Starting from this steady state, the oil mass flow is increased by 10% at $t = 1000 s$. Furthermore, at $t = 5000 s$, the temperature of the oil entering the heat exchanger rises from $5^{\circ}C$ to $125^{\circ}C$.

Figure 12.80 shows the step responses of the uncontrolled systems. Both events cause an increase in the outlet temperatures as to be expected. As can be seen, the dynamics of the increase are different. At $t = 1000 s$ the outlet temperature of the oil rises much faster than the one of the water.

Simulation of the Controlled Counterflow Heat Exchanger

The transfer function of the PID controller is

$$U = K \left[1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{N} s} \right] E, \quad (12.91c)$$

where $s \in \mathbb{C}$. E denotes the Laplace transform of the error into the controller and U the Laplace transform of the controller output. The parameters of the controller have been adopted from [25] and are given in Table 12.18.

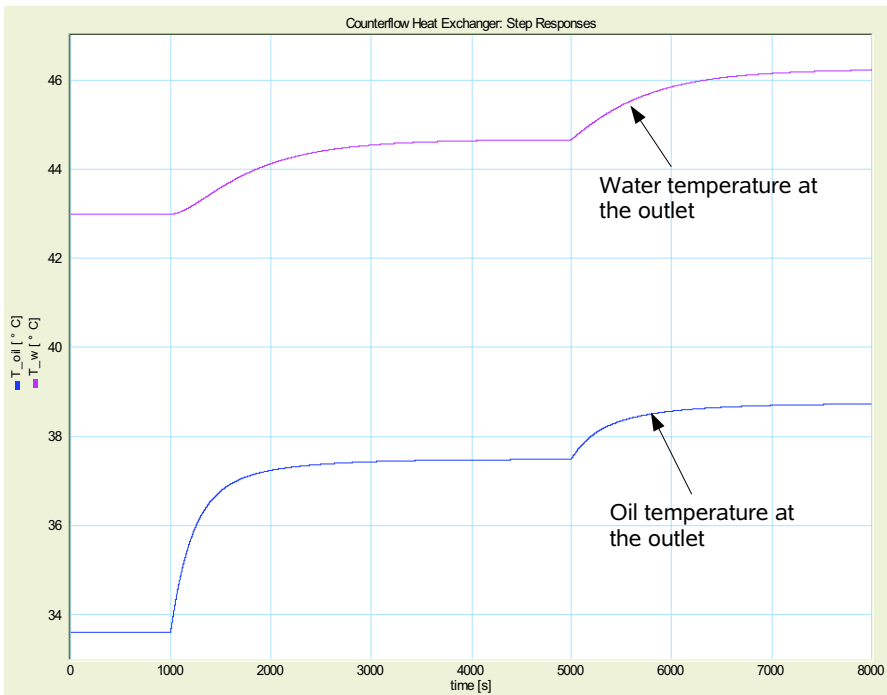


Fig. 12.80 Step responses of the uncontrolled system

Table 12.18 Parameters of the PID controller

Parameter	Value	Units	Meaning
K	-150		Proportional gain
Ti	630	s	Integral time constant
Td	63	s	Derivative time constant
N	63		Derivative gain limitation

As been mentioned above, the major characteristics of the valve have been modelled by a signal saturation block followed by a first order lag signal block with the transfer function $G(s) := k/(\tau s + 1)$. The minimum and the maximum output bound of the saturation block are 0 and 100 respectively. The proportional gain of the first order lag block is $k = 0.005$ so that the output of the valve is limited to 0.5 kg/s . The time constant of the first order lag block is $\tau = 1 \text{ s}$.

The initial value of the controller’s internal integrator, $uI_0 = 27.7$, is the valve’s opening that corresponds to the steady state value of the mass flow of the water flow, $\dot{m}_w = 0.1385 \text{ kg/s}$, required to maintain a steady state oil temperature $T_{oil} = 33.6 \text{ }^\circ\text{C}$ at the outlet.

Figure 12.81 depicts the oil temperature at the outlet of the controlled heat exchanger and the mass flow of the cooling water flow. At $t = 1000 \text{ s}$, the set point of

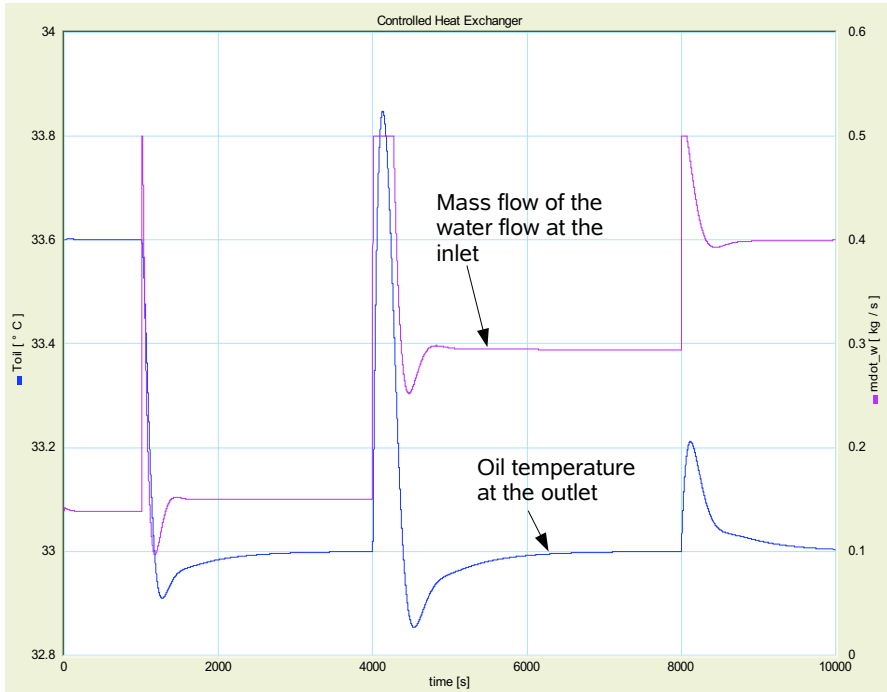


Fig. 12.81 Oil outlet temperature and mass flow of the cooling water flow for the controlled system

the oil outlet temperature instantly drops from 33.6°C to 33.0°C . At $t = 4000\text{ s}$, the mass flow of the entering oil is increased by 10% and at $t = 8000\text{ s}$, the temperature of the entering oil rises by 5°C . In all three cases, after a more or less significant overshoot, the oil outlet temperature returns to the given set point value. The time evolution of the mass flow of the cooling water flow clearly shows the saturation of the valve.

12.14 Conclusion

In this chapter, bond graph methodology has been used in a number of small elaborated case studies. The aim has been to show that, in fact, bond graph modelling can cover the whole range of engineering applications from mechanical systems to electromechanical systems, robots, hydraulic and to open thermodynamic systems in a unified manner. A further objective of this chapter has been to illustrate the application of different features of bond graph modelling methodology.

- As to mechanical systems, an intuitive and convenient modelling approach is to start by considering geometric relations, deriving kinematic constraints from

them and augmenting their bond graph representation so that a model of the system dynamics is obtained.

- Furthermore, classical Lagrange equations, e.g. for robots can be derived directly and systematically from a multibond graph representation. Of course, if simulation of the dynamic behaviour is the purpose and if a standard DAE solver is to be used then these second order ODEs are to be transformed into a set of first order ODEs.
- Three case studies, the engagement of a clutch (Section 12.6), dry friction in a suspension strut of a car (Section 12.7) and a DC-DC buck converter (Section 12.8) have been chosen to show how *hybrid* systems including discrete events can be modelled and simulated in bond graph framework.
- As to hydraulic systems, in general, it is sufficient and appropriate to use a hydrostatic approach. That is, the hydrostatic pressure is chosen as an effort and the amount of hydraulic power is approximated by the product of hydrostatic pressure and volume flow. This assumption has been used in the fluid level control of a three tank system.
- The bond graph model-based approach to FDI presented in Section 6.8 has been illustrated by means of the often used hydraulic two tank system.
- Finally, two small examples including the well known heated stirred tank problem (Section 12.12) and a counterflow heat exchanger (Section 12.13) have been chosen to illustrate the practical use of the pseudo bond graph approach.

In this presentation of a number of small case studies from various disciplines, emphasis has been on bond graph modelling. It has not been the aim to show how existing software can support bond modelling and simulation of the examples. There is one exception with the consideration of dry friction in a suspension leg. For this example, it has been shown in some detail how such a hybrid model, actually, can be computed by means of the open source mathematical software Scilab and the use of an ODE solver with root finding capability.

As to large bond graph models of complex systems, it is clearly beneficial to have some software available that supports a hierarchical modelling approach, provides component model libraries, can automatically set up model equations and can manipulate them symbolically before numerical solution is performed. For models of small up to medium size, an ordered set of model equations can also be derived manually from a causal bond graph in a systematic manner and can be directly formulated, e.g. in the script language of Scilab.

References

- [1] B. Allard, H. Morel, Ph. Lautier, and J.M. Retif. Bond graphs for averaged modeling of power electronic converters. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 1997 International Conference on Bond Graph Modeling and Simulation*, volume 29(1) of *Simulation Series*, pages 201–206. SCS, 1997.
- [2] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, 1996.

- [3] F.T. Brown. *Engineering System Dynamics*. Marcel Dekker, New York, Basel, 2001. ISBN: 0-8247-0616-1.
- [4] J. Buisson. Analysis and Characterisation of Hybrid Systems with Bond Graphs. In *1993 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 264–269, 1993.
- [5] Y.H. Chin and D. Hollander. A DC to DC Converter for Notebook Computers Using HDT-MOS and Synchronous Rectification. Technical Report AN 1547/D, Motorola Inc., 1995.
- [6] H. Christ. *Stationärer und instationärer Betrieb eines federnd gelagerten, unwuchtigen Motors*. PhD thesis, Technische Hochschule Karlsruhe, 1966.
- [7] Controllab Products. 20-sim the power in modeling. URL <http://www.20sim.com>.
- [8] J.J. Craig. *Introduction to Robotics Mechanics & Control*. Pearson Prentice Hall, Upper Saddle, New Jersey, USA, 2005.
- [9] P. Dransfield. *Hydraulic Control Systems – Design and Analysis of Their Dynamics*. Springer-Verlag, New York, 1981.
- [10] J. Garcia-Gomez. *Approche bond graph pour la modélisation des effets thermiques dans les composants de commutation en lectronique de puissance*. PhD thesis, Université des Sciences et Technologies de Lille, Lille, France, 1997.
- [11] J. Garcia-Gomez, G. Dauphin-Tanguy, and Ch. Rombaut. Average bond graph models of dc/dc power converters. In J.J. Granda and F.E. Cellier, editors, *Proc. of the 1999 International Conference on Bond Graph Modeling and Simulation*, volume 31(1) of *Simulation Series*, pages 338–343. SCS, 1999.
- [12] P.J. Gawthrop and E. Ronco. A Sensitivity Bond Graph Approach to Estimation and Control of Mechatronic Systems. Technical Report CSC-99018, Centre for Systems and Control, Univ. of Glasgow, Faculty of Engineering, Dec 1999.
- [13] P.J. Gawthrop and L. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [14] A.C. Hindmarsh. ODEPACK A Systemized Collection of ODE Solvers. Preprint UCRL-88007, Lawrence Livermore National Laboratory, August 1982.
- [15] J. Hoffmann. *MATLAB und Simulink – Beispielerorientierte Einführung in die Simulation dynamischer Systeme*. Addison-Wesley, 1998.
- [16] D.C. Karnopp. Lagrange’s Equations for Complex Bond Graph Systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 99(4):300–306, December 1977.
- [17] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics - Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons Inc., Fourth edition, 2005. ISBN: 0-471-70965-4.
- [18] D. Kölsch and G.P. Ostermeyer. Coulombsche Reibung in der Fahrzeugsimulation. *Automobil-Industrie*, 5:385–388, 1992.
- [19] B. Messner and D. Tilbury. Control Tutorials for Matlab. URL www.engin.umich.edu/group/ctm/.
- [20] H. Morel, B. Allard, H. Elomari, K. Ammous, D. Bergogne, and A. Ammous. Causality Analysis And State Initialization. In J.J. Granda and G. Dauphin-Tanguy, editors, *Proc. of the 2001 International Conference on Bond Graph Modeling and Simulation*, volume 33(1) of *Simulation Series*, pages 91–95. SCS, 2001.
- [21] A. Mukherjee, R. Karmakar, and A.K. Samantaray. *Bond Graph in Modeling, Simulation and Fault Identification*. I.K. International Publishing House, New Delhi, India, 2006. ISBN: 81-88237-96-5.
- [22] C.C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM, Journal of Scientific and Statistical Computation*, 9:213–231, 1988.
- [23] K. Radhakrishnan and A.C. Hindmarsh. Description and use of LSODE, the Livermore solver for ordinary differential equations. Technical Report UCRL-ID-113855, Lawrence Livermore National Laboratory, December 1993.
- [24] V. Ramaswamy. Interactive Power Electronics Online Course. URL <http://www.powerdesignersusa.com/InfoWeb/ressources/pe.html/>.
- [25] H.E. Scherf. *Modellbildung und Simulation dynamischer Systeme*. Oldenbourg, 2004.

- [26] Scilab Consortium. Scilab. URL <http://www.scilab.org/>.
- [27] J.U. Thoma and B. Ould Bouamama. *Modeling and Simulation in Thermal and Chemical Engineering (A Bond Graph Approach)*. Springer-Verlag, Berlin, 2000.
- [28] J.U. Thoma and G. Mocellin. *Simulation with Entropy in Engineering Thermodynamics*. Springer, Berlin, Heidelberg, New York, 2006. ISBN -10 3-540-32798-3.
- [29] M. Vergé and D. Jaume. *Modélisation structurée des systèmes avec les Bond Graphs*. Edition Technip, 2003. ISBN: 2-7108-0838-2.

Chapter 13

Overall Conclusion and Outlook

The objective of this book has been to present the variety of features, the potential and the state-of-the-art of the bond graph modelling methodology in a comprehensive way. To keep the size of the book manageable, some topics had to be excluded. In the beginning of bond graph modelling, it was obvious that some books pointed out relations between bond graph modelling and more traditional modelling approaches. Since then, a number of textbooks on bond graph modelling have emerged from the community of bond graph modellers and this book, among them, is exclusively devoted to the various facets of bond graph methodology.

One of the essential features of bond graph methodology is that it supports the early phase of *conceptual* modelling.

- Word bond graphs are able to account for and describe physical effects and their relations first of all in a *qualitative* manner.
- Submodel templates, to be specified in more detail, can be connected according to the way corresponding real subsystems or real system components are linked.
- The stepwise systematic refinement of word bond graphs enables the development of hierarchical modular models. Submodels can be built up either from basic bond graph elements, or by choosing submodels from engineering domain specific libraries. Once a hierarchical modular model has been composed, several approaches can be used to remove unnecessary complexity and to come up with a proper model of reduced complexity that is sufficiently accurate for a given engineering task.
- The interconnection of bond graph submodels according to the physical structure of a system anticipates one aspect of object-oriented modelling, namely that equations describing a submodel must be *non-causal* and that the classification of variables into input and output variables results from the interconnection of submodels.

As a consequence of the latter aspect, many up-to-date software packages supporting bond graph modelling perform some formulae manipulation during model processing *before* simulation code is generated. Chapter 11 demonstrated how object-

oriented modelling languages can be used for the description of hierarchical modular bond graph models.

Another essential and powerful feature of bond graph methodology is the concept of *computational causality*. It appears that this is sometimes underestimated by those who prefer a network approach and who rely on the symbolic preprocessing and the numerical solution of DAE systems. Actually, (automatic) assignment of computational causalities provides considerable insight into features of the model as demonstrated in Chapters 3–6.

- There are several causality assignment procedures. Depending on the way computational causalities are assigned and depending on the model, either state space models, equations in descriptor form, or Lagrange equations can be derived.
- Before writing any equations, inspection of a causally completed bond graph can provide information about the form of the mathematical model. If it will take the form of a DAE system, inspection of causal paths can reveal information about its structural index as has been shown by van Dijk. By considering causal paths and their associated topological loops, modellers can decide whether dependent stores or algebraic loops can be avoided by reasonable modifications of the models, or if the model should be left for numerical solution by a DAE solver. Up-to-date advanced modelling and simulation software can handle these cases to some extent automatically by applying some symbolic formulae manipulation where possible and by performing numerical iteration where necessary. Consequently, depending on the settings, such software programs mostly will just issue warnings instead of error messages.
- Chapter 6 shows that a bond graph, in fact, can serve as a *core* model representation from which not only equations for time domain analysis can be derived, but also transfer functions, equations of the inverse system, parameter sensitivities, or state equations for robustness study. Moreover, inspection of causal bond graphs can reveal information about structural properties, e.g. structural controllability and structural observability. Furthermore, bond graph modelling can support model-based fault detection and isolation.

In Chapter 5, it is shown that identification of tearing variables in causal bond graphs can support the partitioning of large sets of algebraic constraints into a number of coupled smaller sets of equations. If the equations are linear in the tearing variables, they can be solved symbolically. As a result, a set of ODEs instead of a DAE system has to be solved. If the resulting ODE system is supposed not to be stiff, an explicit integration method can be chosen. Otherwise, at least the advantage remains that at each time, the set of equations to be solved by Newton-Raphson iteration has been reduced.

The aim of including the abstraction of discontinuous system behaviour into bond graph modelling has led to a number of approaches. None of them has prevailed as a standard method so far. Some authors use the ideal switch as another basic bond graph element and dismiss the concept of fixed causalities assigned once before a simulation run. Consequently, after each discrete event, when a switch state has

changed, the impact of the causality change at this switch on the rest of the bond graph must be checked [1, 21].

In contrast, Section 7.3 proposes to use (for each system mode given by a certain configuration of all switch states) a bond graph model and to represent the different system modes and transitions between by an associated Petri net. A disadvantage of this and related approaches using Petri nets or finite state machines is that the number of possible combinational system modes and transitions between them significantly increases with the number of switches. However, actually not all combinational modes are physically feasible. Thus, the task remains to identify all physical feasible system modes and transitions between them. For each system mode, a bond graph model with standard bond graph elements can be set up in a usual way. As a result, a system is not represented by one single bond graph, but by a number of bond graphs and a global Petri net. The approach is feasible and of advantage if the number of elements is small that give rise to an approximation of their fast transients by an instantaneous discontinuous change of state.

A possible alternative is to use a single bond graph in which some sources at the advent of an instantaneous discontinuous change of the system state assume values such that some degrees of freedom are either switched off or are reestablished. The consideration of several approaches in Chapter 7 shows that the abstraction of instantaneous discontinuous changes between states can be included into a bond graph modelling framework. Yet, some authors consider the concept of computational causalities oversold with regard to hybrid system models.

Publications of Sueur and Dauphin-Tanguy as well as those of other authors have shown that inspection of causally completed bond graphs of models with *invariant* structure can provide information about structural properties such as structural observability and structural controllability which are prerequisites of the observability and controllability of a system.

Samantaray, Bouamama and Staroswiecki as well as other authors have shown how bond graph modelling can be used for fault detection and isolation (FDI). On the other hand, several approaches have been proposed on how to deal with the abstraction of instantaneous state transitions in a bond graph framework. There is ongoing research to extend the analysis with regard to structural properties [20] as well as advances regarding fault diagnosis to hybrid systems models [7, 15, 16].

Furthermore, bond graph methodology, in fact, has the potential for a broad application across different engineering disciplines and allows for a unified description of multidisciplinary systems. This becomes evident by the chapters on modelling multibody systems with rigid bodies, distributed parameter models and on open thermodynamic systems and by the small case studies from various disciplines in Chapter 12.

Today's concurrent system design increasingly considers entire complex controlled systems. Consequently, systems to be designed are inevitably multidisciplinary. Thus, methodologies supporting modelling of multidisciplinary dynamic systems become more and more important. From this point of view, bond graph methodology, although almost 50 years old, is receiving new additional attention.

It is true that there are well known software programs, e.g. Adams[®], for the computation of mechanical structures and other powerful software for modelling and simulation of hydraulic systems, or Spice for electronic circuit simulation. However, these programs require domain specific model descriptions and account for components from other domains only in a limited way. On the other hand, if a mechanical structure with parts to be represented by a distributed parameter model is driven by hydraulic actuators, then the overall system can be represented uniformly by a bond graph. Such a model then can be graphically entered and processed in modelling and simulation environments like SYMBOLS Shakti[™] or 20-sim[®].

In Chapter 8, first, multibond graph modelling of multibody systems as proposed by Bos is presented. That is, translational motion of bodies assumed to be rigid is referred to an inertial frame, while the rotation of a body is described with reference to a body fixed frame. Moreover, bond graph modelling of multibody systems can also be used for the joint coordinate method in order to generate a minimal set of equations of motion for systems with tree-like structure having no kinematic loops.

In Chapter 10 on open thermodynamic systems, two details in hydraulic components have been considered that sometimes have given rise to an ad hoc description in the literature. It has been shown that the energetically incorrect description of an oil filled variable chamber of a hydraulic cylinder by means of a C element controlled by the displacement of the piston can be replaced by an energy conservative 2-port C energy store. Moreover, under some simplifying assumptions, a bond graph model of the hydraulic mechanical interaction of the jet stream with the valve in a spool valve control orifice has been developed that is consistent with conservation laws for mass, momentum and energy.

This monograph on bond graph methodology aims at a comprehensive and self contained presentation while at the same time concentrating on a number of topics in order to deal with them in sufficient depth. Beyond the considered topics, there are a number of other engineering domains in which bond graph methodology can be applied as well as areas in which bond graph modelling is a subject of ongoing research. For instance, bond graph modelling of the complex fluid mechanical, biochemical processes in sewage plants is considered a subject of further research. As Chapter 10 on open thermodynamic systems has shown, there are different possible ways to represent intrinsic interactions between mass, energy and momentum. In a 1993 conference paper, Brooks and Cellier [3] expressed their confidence that “bond graphs do point the way towards the nature of the requisite research”. Since then, further results have been published (see for instance [4, 14, 23], or [12], Chapter 8 on process systems, or the recent textbook of Samantaray and Bouamama on model-based process supervision [22]).

Further application areas that are beyond the scope of this book on bond graph methodology are, e.g. chemical reaction kinetics or qualitative reasoning in the overlap of artificial intelligence and engineering disciplines. In his 1991 textbook on continuous system simulation, Cellier [5] devoted an entire chapter to bond graph-based modelling of chemical reaction kinetics.

Regarding bond graph modelling of biophysical systems, an article of Oster, Perelson and Katchalsky [17] was published in as early as 1973. One of the lat-

est publications in this field is the contribution of V. Diaz and her co-workers [8] in the 2009 special issue on bond graph modelling of the journal *Simulation Modelling Practice and Theory* [2]. By connecting a 3D model of the mitral valve to a bond graph model of the left ventricle, the authors demonstrate that the bond graph methodology is not only suited for the development and the analysis of mechatronic engineering systems, but also has the potential to even support physiological modelling with complex mechanical, hydraulic and biochemical interactions. Although the bond graph model is rather simple and modelling in this area certainly faces considerable challenges, the paper shows that the bond graph methodology is a promising approach to the development of lumped parameter component models for physiological systems.

Besides engineering systems such as industrial robots and all kinds of mobile robots including walking machines, a subject of further research and progress in bond graph modelling is expected to be the human body musculoskeletal system. In this area, Selk Ghafari et. al. presented a paper in 2007 [13].

Xia, Linkens and Bennett, for instance, published on the integration of qualitative reasoning and bond graph modelling [26].

Concluding this monograph on bond graph methodology, we come back to Professor Paynter and the vision he expressed in the preface of the proceedings of the first international conference on bond graph modelling in 1992.

I remain convinced that BG models will play an increasingly important role in the upcoming century, applied to chemistry, electrochemistry and biochemistry, fields whose practical consequences will have a significance comparable to that of electronics in this century.

Textbooks on thermal and chemical engineering have been authored by Thoma and Bouamama in 2000 [23] and Thoma and Mocellin in 2006 [24]. Examples of the latest publications in the area of chemistry and electrochemistry are the Ph.D. thesis of Franco [11], the paper of Esperilla and his coworkers on bond graph modelling of lead-acid batteries [9], the article of Couenne and her coauthors on bond graph modelling of chemical reactors [6], and the Ph.D. thesis of Vijay on true bond graph modelling, simulation and control of a fuel cell [25].

Another field in which bond graph modelling is anticipated to play an increasing role is medical engineering as there is an increasing use of micro-scale mechatronic devices. Recently, Pirvu and her coworkers applied bond graph modelling in the redesign of a hand held medical instrument that is used for the diagnosis of ear diseases [19]. In [18], bond graph modelling has been applied in the design of a behind-the-ear hearing aid. The near future is likely to show further applications of bond graph modelling in medical engineering.

Beyond miniaturised multi-energy domain systems in medical engineering, a further potential area for application of the bond graph methodology are micro-electro-mechanical systems (MEMS) (see for instance [10]).

References

- [1] C.D. Beers, E.J. Manders, G. Biswas, and P.J. Mosterman. Building efficient simulation from hybrid bond graph models. In *Proc. of the 2nd IFAC Conf. on Analysis and Design of Hybrid Systems*, pages 71–76, Alghero, Italy, 2006.
- [2] W. Borutzky, editor. *Special Issue Bond Graph Modelling*, volume 17/1 of *Simulation Modelling Practice and Theory*, 2009. Elsevier.
- [3] B.A. Brooks and F.E. Cellier. Modeling of a distillation column using bond graphs. In J.J. Granda and F.E. Cellier, editors, *Proc. of the International Conference on Bond Graph Modeling, ICBGM' 93*, pages 315–320. SCS, SCS Publishing, Jan. 17–20 1993. La Jolla, CA.
- [4] F.T. Brown. Non-iterative evaluation of multiphase thermal compliances in bond graphs. *Proc. Instn Mech. Engrs, Part I, Journal of Systems and Control Engineering*, 216(1):13–19, 2002.
- [5] F.E. Cellier. *Continuous System Modeling*. Springer-Verlag, New York, Berlin, Heidelberg, 1991. ISBN: 0-387-97502-0.
- [6] F. Couenne, C. Jallut, B. Maschke, P.C. Breedveld, and M. Tayakout. Bond graph modelling for chemical reactors. *Mathematical and Computer Modelling of Dynamic Systems*, 12(2-3): 159–174, 2006.
- [7] M. Daigle, X. Koutsoukos, and G. Biswas. *Hybrid Systems: Computation and Control*, volume 4981/2008 of *Lecture Notes in Computer Science*, chapter An Integrated Approach to Parametric and Discrete Fault Diagnosis in Hybrid Systems, pages 614–617. Springer-Verlag Berlin Heidelberg, 2008. ISBN 978-3-540-78928-4.
- [8] V. Diaz-Zuccarini, D. Rafirou, J. LeFèvre, D.R. Hose, and P.V. Lawford. Systematic Modelling and Computational Physiology: The Applications of Bond Graph Boundary Conditions for 3D Cardiovascular Models. *Simulation Modelling Practice and Theory*, 17/1:125–136, 2009.
- [9] J.J. Esperilla, J. Félez, J. Maroto, and J.M. Cabanellas. A model for simulating a lead-acid battery using bond graphs. *Simulation Modelling Practice and Theory*, 15(1):82–97, 2007.
- [10] Z. Fan, J. Wang, S. Achiche, E. Goodman, and R.C. Rosenber. Structured synthesis of mems using evolutionary approaches. *Appl. Soft Comput.*, 8(1):579–589, 2008. ISSN 1568-4946.
- [11] M.A.A. Franco. *Un modèle physique multiéchelle de la dynamique électrochimique dans une pile à combustible à électrolyte polymère – Une approche Bond Graph dimension infinie*. PhD thesis, L' Université Claude Bernard – Lyon 1, France, 2005.
- [12] P.J. Gawthrop and L. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall International (UK) Limited, Hemel Hempstead, 1996. ISBN: 0-13-489824-9.
- [13] A. Selk Ghafari, A. Meghdari, and G.R. Vossoughi. Modeling of human lower extremity musculoskeletal structure using bond graph approach. In *Proc. ASME International Mechanical Engineering and Exposition*, Seattle, Washington, USA, 2007. on CD-ROM.
- [14] J. Greifeneder and F.E. Cellier. Modeling Multi-Phase Systems Using Bond Graphs. In J.J. Granda and G. Dauphin-Tanguy, editors, *2001 International Conference on Bond Graph Modeling (ICBGM'01)*, *Proc. of the 2001 Western Simulation Multiconference*, pages 285–291. SCS Publishing, January 7-11 2001. Simulation Series, Vol. 33, No. 1, ISBN: 1-56555-221-0.
- [15] P.J. Mosterman. *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis*. PhD thesis, Vanderbilt University, Nashville, TN, USA, 1997.
- [16] S. Narasimhan and G. Biswas. *Hybrid Systems: Computation and Control*, volume 2289/2002 of *Lecture Notes in Computer Science, Hybrid Systems: Computation and Control*, chapter An Approach to Model-Based Diagnosis of Hybrid Systems, pages 308–322. Springer-Verlag Berlin, Heidelberg, New York, 2002. ISBN 3-540-43321-X.
- [17] G.F. Oster, A.S. Perelson, and A.K. Katchalsky. Network thermodynamics: dynamic modelling of biophysical systems. *Q. Rev., Biophys.*, 6:1–134, 1973.

- [18] M.A. Perry, M.A. Atherton, R.A. Bates, and H.P. Wynn. Bond Graph Based Sensitivity and Uncertainty Analysis Modelling for Micro-Scale Multiphysics Robust Engineering Design. *Journal of the Franklin Institute*, 345(3):282–292, 2008.
- [19] A. Pirvu, G. Dauphin-Tanguy, and Ph. Kubiak. Automatic Adaptation of a Bond Graph Model to Transfer Function Specifications. *Simulation Modelling Practice and Theory*, 17/1: 257–270, 2009.
- [20] A. Rahmani and G. Dauphin-Tanguy. Structural analysis of switching systems modelled by bond graph. *Mathematical and computer Modelling of Dynamical Systems*, 12 (2-3):235–247, 2006.
- [21] I. Roychoudhury, M. Daigle, G. Biswas, X. Koutsoukos, and P.J. Mosterman. A Method for Efficient Simulation of Hybrid Bond Graphs. In J. J. Granda and F. E. Cellier, editors, *Proc. of the 2007 International Conference on Bond Graph Modeling and Simulation*, volume 39 (1), pages 177–184, 2007.
- [22] A.K. Samantaray and B. Ould Bouamama. *Model-based Process Supervision – A Bond Graph Approach*. Advances in Industrial Control. Springer, London, 2008. ISBN 978-1-84800-158-9.
- [23] J.U. Thoma and B. Ould Bouamama. *Modeling and Simulation in Thermal and Chemical Engineering (A Bond Graph Approach)*. Springer-Verlag, Berlin, 2000.
- [24] J.U. Thoma and G. Mocellin. *Simulation with Entropy in Engineering Thermodynamics*. Springer, Berlin, Heidelberg, New York, 2006. ISBN -10 3-540-32798-3.
- [25] P. Vijay. *Modelling, simulation and control of a solid oxide fuel cell system: a bond graph approach*. PhD thesis, Indian Institute of Technology, Kharagpur, 2009.
- [26] S. Xia, D.A. Linkens, and S. Bennett. Integration of Qualitative Reasoning and Bond Graphs: an engineering approach. In P.C. Breedveld and G. Dauphin-Tanguy, editors, *Bond Graphs for Engineers*, pages 323–332. North-Holland, 1992.

Glossary

In this glossary, some of the notions introduced in the text are listed in alphabetical order along with their explanation.

Acausal bond graph An acausal bond graph is a directed bond graph. That is, reference directions for the energy flows have been defined but not computational causalities. p. 94

Activated bond A bond is called activated if one of its power conjugated variables is set zero for all time instances. p. 95

Admissible numerical realisation of a structural matrix Let $[\mathbf{A}]$ be a structural matrix. A matrix \mathbf{A} is called an admissible numerical realisation of $[\mathbf{A}]$ (for short $\mathbf{A} \in [\mathbf{A}]$) if all entries of indeterminate value are replaced by entries of particular numerical values. p. 240

Algebraic loop If the variables of a topological loop dependent algebraically on themselves, viz. no integration with respect to time is involved, then the topological loop is called an *algebraic loop*. p. 132

Bipartite graph In a bipartite graph, variables as well as equations are represented by nodes. If a variable appears in an equation, their corresponding nodes are connected by an undirected edge. p. 90

Bond loop or mesh A closed cascade of bonds is called a *bond loop*, a *mesh*, or a general mesh. p. 109

Causal bond graph A directed bond graph is called a causal or causally completed bond graph if a decision with regard to computational causality has been made for each bond expressed by a perpendicular causal stroke added to one end of each bond. p. 94

Causal conflict of type 1 In this case, there is no bond that determines the common variable of the junction. At a 1-junction, the causal stroke of all adjacent bonds point towards the 1-junction. At a 0-junction, the causal stroke at all adjacent bonds is pointing away from the junction. p. 164

Causal conflict of type 2 and degree k If there are $k + 1$ bonds that want to determine the common variable of a junction, then there is a causal conflict of type 2 and degree k . That is, at a 1-junction not one, but $k + 1$ flows are input to the junction, while at a 0-junction, $k + 1$ instead of only one single effort are input to the junction.
p. 164

Causal loop If the bonds of a causal path only connect elements of the junction structure and if the causal path is closed, then it is called a *causal loop*.
p. 110

Causal loop gain The gain of a causal loop is the loop gain of the two topological loops of opposite orientation associated of the causal loop.
p. 149

Causal mesh A *causal mesh* is a closed causal path with an odd number of gyrators.
p. 110

Causal path A sequence of bonds from one power port of an element to a power port of another element is called a causal path if there is no 2-port gyrator in between and if all bonds have their causal stroke at the same end.

A cascade of bonds between two power ports with a gyrator in between is called a causal path if all bonds on one side of the gyrator have their causal stroke at the same end, while all bonds on the other side of the gyrator have their causal stroke on the opposite end. That is, the gyrator switches the direction of efforts on one of its sides.
p. 109

Complete state controllability The state $\mathbf{x}(t)$ is said to be controllable at time $t = t_0$ if there exists a piecewise continuous input function \mathbf{u} that causes the state vector to move to any final value $\mathbf{x}(t_f)$ in a finite time $t_f > t_0$. If each state \mathbf{x}_0 is controllable, then the system is said to be *completely state controllable*.
p. 241

Complete observability A system is said to be completely observable if for each initial state $\mathbf{x}(t_1)$ at time instant t_1 , there is a time $t_2 > t_1$ such that $\mathbf{x}(t_1)$ can be found from the known input vector $\mathbf{u}(t)$ and the output vector $\mathbf{y}(t)$ measured over the interval of time $[t_1, t_2]$.
p. 245

Dimension of the state vector It is assumed that kinematic displacements are not needed to describe the dynamic behaviour of a system. Then, the dimension of the state vector equals the number of I and C ports.
p. 119

Directed bond graph A bond graph is called a directed bond graph if a half arrow has been added to each bond indicating the positive reference direction of the energy flow across the bond.
p. 29

Disjoint causal paths Causal paths that do not share any bonds are called *disjoint*.
p. 109

Environmental elements All elements that do not belong to the general junction structure are called *environmental elements*
p. 42

External bond A bond is called an external bond if it connects a 0- or 1-junction to a power port of an element that does not belong to the simple junction structure. p. 35

Flat loop A topological loop is called a *flat loop* if both opposite signals of each bond being part of the causal path or causal loop are involved in the signal flow loop. p. 132

General junction structure A bond graph with nodes of type 0,1, (M)TF, (M)GY is called General Junction Structure (GJS). p. 42

Index of a matrix A quadratic matrix \mathbf{M} is called *nilpotent* if there is positive integer k such that $\mathbf{M}^k = \mathbf{0}$.

If \mathbf{M} is a nilpotent matrix, then the smallest positive integer ν for which $\mathbf{M}^\nu = \mathbf{0}$ and $\mathbf{M}^{\nu-1} \neq \mathbf{0}$ is called the *index of nilpotency*. p. 135

Influence coefficient The influence coefficient of a junction structure node is the ratio of the output variable to the input variable for a particular signal flow loop fragment associated with *two* adjacent bonds of opposite causal orientation. p. 148

Internal bond A bond is called an internal bond if it connects a 0- or 1-junction to another 0- or 1-junction. p. 35

Internal modulation If a bond graph element is modulated by a power variable, then it is said to be *internally modulated*. p. 158

Inverse system Let S denote a system; let \mathbf{u} be the vector of all inputs to S and \mathbf{y} the vector of outputs of S . Then, the *inverse* system S_I is the system that provides the signals \mathbf{u} as a response $\mathbf{y}_I = \mathbf{u}$ to the input signals $\mathbf{u}_I = \mathbf{y}$. p. 235

Junction structure A bond graph in which bonds connect only nodes that instantaneously transfer or distribute power (without energy storage or conversion into heat) is called Junction Structure (JS). p. 30

Loop gain of a topological loop The loop gain of a topological loop is the product of all influence coefficients. p. 149

multiport A bond graph node is called a *multiport* if it has more than one port. p. 21

Open loop A topological loop that uses only one of the opposite signals of some or all bonds in a causal path is called an *open loop*. p. 132

Order of the set of differential equations The order of the set of differential equations is equal to the number of *independent* state variables. p. 119

Order of the model It is assumed that kinematic displacements are not needed to describe the dynamic behaviour of a system.

If integral causality has been assigned as preferred causality to the power ports of storage elements, then the order of the model is the number of power ports of energy stores with integral causality. p. 119

Order of a topological loop The order of a topological loop denotes the number of remaining integrators involved in the causal path. If there is no remaining integration in the causal path, then the topological loop is called a zero-order loop. p. 132

Power port The connection points of a bond graph node that enable the energy exchange with other nodes across a power bond are called power ports. p. 20

Simple junction structure A bond graph is called a simple junction structure or Kirchhoff junction structure, if each node is either a 0- or 1-junction. p. 35

Simple even (odd) mesh A simple mesh is called even (odd) if an even (odd) number of its bonds has the same energy flow reference direction in a clock-wise or counter-clockwise sense. p. 110

Simple mesh A bond loop is called a *simple mesh* if it includes no transformers, no gyrators, and no 2-port energy stores or 2-port resistors. In other words, a simple mesh is a loop of bonds that interconnect alternately 0- and 1-junctions. p. 109

State variable, System state Suppose there are p inputs u_1, \dots, u_q to a dynamic system and n intermediate variables x_1, \dots, x_n . Moreover, physical laws may yield n differential equations

$$\dot{x}_i(t) = f_i(x_1(t), \dots, x_n(t); u_1(t), \dots, u_p(t)) \quad i = 1, \dots, n.$$

Let $t_0 \geq 0$ an arbitrary time point. For all times $t \geq t_0$, values $u_1(t), \dots, u_p(t)$ of all p system inputs may be known. Then, n intermediate variables $x_1(t), \dots, x_n(t)$ are called *state variables* if they are uniquely determined for all times $t > t_0$ provided their initial values $x_1(t_0), \dots, x_n(t_0)$ are given. The set of all values $x_1(t), \dots, x_n(t)$ at a time instant $t \geq 0$ is called the *state* of a system. p. 104

Strong (weak) causal determination of a junction A bond imposes a *strong* causal determination on a junction J it is connected to if one of its power conjugate variables determines the variable common to all remaining adjacent bonds. Otherwise, the bond gives a *weak* causal determination to the junction. p. 145

Structurally complete state controllability A linear time-invariant MIMO system with matrices \mathbf{A} and \mathbf{B} is said to be *structurally* completely state controllable if there is at least one numerical admissible realisation \mathbf{A} of the structural matrices $[\mathbf{A}]$ and one admissible realisation \mathbf{B} of $[\mathbf{B}]$ for which the system is completely state controllable. p. 241

Structurally complete observability A linear time-invariant MIMO system with matrices \mathbf{A} and \mathbf{C} is said to be *structurally* completely observable if there are numerical admissible realisations $\mathbf{A} \in [\mathbf{A}]$ and $\mathbf{C} \in [\mathbf{C}]$ for which the system is completely observable. p. 246

Structural matrix A *structural* matrix $[\mathbf{A}]$ is determined by the number and the position of its non-zero entries. The latter are considered of indeterminate value and independent of each other. They are denoted by the symbol $*$. p. 240

Topological loop A topological loop is a signal flow loop along a causal path or a causal loop. The causal path must not begin or end at an ideal source. p. 131

Topological path A topological path is a part of a topological loop. It is a signal flow graph fragment that represents bond variables and constitutive relations being part of a causal path. p. 132

Undirected bond graph An undirected bond graph is an undirected graph whose vertices denote subsystems, components, or basic elements, while the edges called (power) bonds represent the instantaneous energy flows between nodes. p. 17

Weighted junction structure A bond graph with nodes of type 0,1, (M)TF is called a Weighted Junction Structure (WJS). p. 42

Word bond graph A bond graph is called a word bond graph if its vertices represent subsystems or components and are denoted by a word or an alphanumeric symbol. p. 20

Index

20-sim® 516–519

A

Accumulator 427
 thermodynamic accumulator 432
Across variable 24, 28
ACSL®
 DERIVATIVE 470
 DYNAMIC 470
 IMPVC 477
 MACRO 471
 TERMINAL 471
Activity metric 294
Actuator 60
Adams® 385
Algebraic loop 105, 134, 160–164, 476, 480,
 506
Analogy
 classical 23
 mobility- 23
Analytical Redundancy Relations 285
ANSYS® 407
ARCHER 514
Array
 of bond graph elements 355–356
Artificial flow source 169

B

BAPS® 513
BDF 180, 181
Bernoulli's energy equation 458
BGML 539
Bicausality 278
Bipartite graph 92
Block diagram 9–10

Bond

activated 97
bicausal 278
convection 446
external 35
internal 35
multibond 355

Bond graph

acausal 96
bicausal 277–283
causal 96
directed bond graph 29
hybrid 338
incremental 251–273
latent part 330
multibond graph 354–365
of a hydraulic cylinder 457
pseudo- 63–65
simplifications 83
systematic construction 66–81
uncertainty 273–277
undirected bond graph 17
word bond graph 20
Bond graph preprocessors 511–514
Bond graph rank 125
Bond loop 111
BondSim® 516
BONDYN 374, 518
Boolean matrix 242
Bouncing ball problem 314–316, 323, 338
Bridge circuit
 pseudo bond graph 438
 with double acting pneumatic cylinder
 437
Buck converter 603–611

C

- C-field 404, 412
- CAMBAS 534
- CAMP-G® 512
- Capacitor
 - thermal 47
- Causal analysis 92
- Causal conflict 100
 - of type 1 166
 - of type 1 and degree k 166
- Causal loop 112
- Causal path 111
 - between independent and dependent storage port 144–147
 - between resistive ports 131–136
 - closed 147–152
 - removal of causal paths between resistors 142
 - zero-order 135
- Causal stroke 96
- Causality
 - assignment 98–100, 103–106
 - Bicausality 278
 - derivative 99
 - integral 99
 - Lagrange 168–173
 - relaxed 164–168
 - strong 211
 - weak 211
- Charge amplifier 589–590
- Chemical potential 439
- Clutch 591–595
- Computational causality
 - assignment 103
 - notion 94
 - representation 95
 - variable 331–346
- Computational order 480, 510
- Computational structure 9
- Condition number
 - iteration matrix 183
- Constraints
 - geometric 563
 - holonomic 328
 - kinematic 376
- Controllability
 - complete state controllability 243
 - matrix 243
 - structural complete controllability 243
- Coordinate transformation 369
- Coulomb friction *see* Friction, dry

D

- DAE *see* differential-algebraic equations
 - DASSL 181, 221, 330, 345
 - DC motor 40, 258, 281, 527
 - Delta subnetwork of resistors 155
 - Descriptor form 139
 - linearised 146
 - Detectors 62
 - fictitious (virtual) 62, 290
 - differential-algebraic equations 91
 - Differential-algebraic system
 - differential index 138
 - differential index of semi-state systems 139
 - index determination 140
 - index of nilpotency 137
 - local index 138
 - perturbation index 216
 - reduction of the index 184–191
 - Diode
 - ideal 322
 - Discrete event 317, 340, 341, 346
 - Disjoint causal paths 111
 - Dissipators 49–56
 - Dummy derivatives 189
 - Dymola® 92
- E**
- Effort 22
 - Electrical circuit
 - bond graph 75
 - with an isolating transformer 74
 - Energy
 - co-energy 44
 - free 49
 - Helmholtz free energy 444
 - internal 76, 425, 432, 438, 444, 461
 - Energy sources 59–63
 - Energy stores 42–49
 - dependent 109
 - independent 109
 - with parameters of small value 141–144
 - ENPORT™ 506–507
 - GRAPH 506
 - line code 506
 - Enthalpy 429
 - flow 429, 435
 - stagnation 446
 - Entropy 47
 - Equations
 - constitutive 31
 - residual equations 199

Euler's equation 360
 Euler-Bernoulli beam 394
 Eulerian control volume 425
 Eulerian Junction Structure (EJS) 360

F

Fault detection and isolation (FDI) 284
 Fault isolation 288–290
 Fault signature matrix 288
 Field 357
 energy storage field 42
 implicit field 364
 Finite element method 408–412
 Finite State Machine 338
 Flow 22
 Flow forces in spool valves 456–464
 Fourier's law 53
 Friction
 dry 324–331, 595–601

G

Galerkin's weighted residual method 415
 Generalised
 constraint forces 376
 coordinates 168, 376
 displacement 28
 momentum 28
 Gibbs' fundamental equation 439
 Graph
 bipartite 92
 Gyration
 symplectic 47
 two-port 38
 Gyristor 171

H

Hagen and Poiseuille law 455
 Hamiltonian 192
 equations 192
 Heat exchanger 635–640
 Heated stirred tank 631–635
 Hooke's law 450
 HYBrSIM 340
 Hydraulic drive 79
 bond graph 80
 word bond graph 79

I

I-field 412
 implicit Runge-Kutta methods 184

Incremental bond graph 251–273

Index

 local index 138
 of a matrix 137
 of nilpotency 137
 perturbation index 216–218
 reduction 184–191

Inertias

 dependent 110, 372

Integrated modelling and simulation

 environments 515–530

J**Joint**

 coordinates 376
 prismatic joint 370, 371
 revolute joint 370, 371

Junction

 0-junction 32
 1-junction 33
 controlled 337–340, 345, 547–548
 inactive 297
 switched power junction 313–316

Junction structure 30

 Eulerian Junction Structure (EJS) 360
 general 42
 simple 35
 weighted 42

Junctions 31–35**K**

Kinematic displacement 110

Kirchhoff

 current law 32
 voltage law 34

L**Lagrange**

 causalities 168–173
 equations of the second kind 168, 563
 multipliers 192, 323, 328

Lagrangian 171

Legendre transformation 192, 445

Linear fractional transformation 271

Loop

 algebraic loop 105, 134, 160–164, 476, 480, 506
 bond loop 111
 causal loop 112
 causal loop of unity loop gain 155
 flat loop 134

Mason's loop rule 233
 open loop 134
 order of a topological loop 134
 topological loop 133
 LSODAR 344

M

Maple™ 197, 299
 Mason's loop rule 233
 Mass matrix 409, 410
 Mathematica® 515
 MATLAB® 92, 251, 257, 260
 MAX 533
 Maxwell's reciprocity condition 358
 Memristors 56–59
 Mesh 111
 simple even mesh 111
 simple mesh 111
 Method
 dummy derivative method 481
 finite element method 407–418
 Galerkin's method 415
 joint coordinate method 374–385
 of relaxed causalities 164–168
 MIMO system 229, 243
 Modal
 mass 396
 oscillator 397, 398, 401, 402
 oscillators 401
 stiffness 396
 Modal analysis 394–406
 mode shapes 395
 Model
 distributed parameter model 391–421
 generic model 400, 420
 hybrid 598, 641
 lumped parameter model 22, 391
 partitioning 294
 reduction 144, 294
 simplification 295
 state space 115
 Modelica® 490–505
 Modelling
 object-oriented 5–8
 physical systems 19
 Modulation
 internal 160
 MS1® 525–530
 MTT 515, 531
 Multiport 21
 elements 356–365

N

NASTRAN® 407
 Newton-Euler's equation of motion 376
 Nodicity property 158
 Null modes 126

O

Object-oriented modelling languages
 478–505
 Observability
 complete observability 247
 matrix 247
 structural complete observability 248
 ODASSL 189
 ODEPACK 477, 595, 598
 ODEs *see* System, ordinary first order
 differential equations
 One-junction 33
 array of 1-junctions 355
 Onsager's reciprocity 364
 Order
 of the model 121
 of the set of differential equations 121
 topological loop 134

P

Pantelides algorithm 187
 Parameter sensitivity 251–262
 PASION® 516
 Pendulum
 inverted 563–568
 mathematical 185–191
 Petri net 333, 343
 Piezoelectric crystal 584–586
 Piezoelectric seismometer 584–591
 Port
 multiport 21
 power 20
 signal 21
 Power variable 22
 Pressure
 hydrostatic 38, 425
 total 76
 Pseudo bond graph 63–65
 Puma robot 612–617

R

RADAU code 184
 Reciprocity
 Maxwell's 358

Onsager's 364
 REDUCE 532
 Reference directions
 of energy flows 82
 Residual 285
 equations 199
 Residual sink 200, 202, 290–293
 Resistive bond graph model
 of a C energy store 208
 of an I energy store 208
 Resistor *see* Dissipators
 RLC circuit 172
 bond graph 173
 Robot
 multibond graph model 373
 with three degrees of freedom 372
 Robustness 263
 RS Element 52–56

S

Scilab 598, 612
 Semi-explicit form 182
 Semi-state
 systems 139
 variables 139
 Sensitivity matrix 251
 Sensors 62–63
 Sequential Causality Assignment Procedure
 (SCAP) 103
 Series motor
 simplified bond graph 161
 Shape functions 408
 Shunt motor 568–572
 ACSL description 473
 causal bond graph 40, 472
 SIDOPS
 interface 485
 subclasses 485
 Signal flow graph 10–11, 231–233
 Simulation languages 470–478
 Simulink® 92, 513, 527, 572
 Slider crank mechanism 145
 bond graph 146
 Source-sensor 63, 241, 279
 Sparse-matrix solver 198, 220
 Standard interconnection form 271–273
 State space form 115
 canonical form 267
 semi-explicit 374
 State variable 106
 algebraic state variable 166
 dependent state variable 109, 121
 semi- 139

Steady state 226–229
 Stick-slip friction 324–327
 Stiffness matrix 410, 417
 Strong causal determination of a junction
 147, 340
 Structural matrix 242
 admissible numerical realisation 242
 Structure
 physical 17
 Switch
 ideal 318
 real 311
 switch state 310
 Symbolic Math Toolbox™ 251, 538
 SYMBOLS Shakti™ 520–525
 System
 closed system 444
 holonomic 193
 hybrid 340, 499, 525, 641
 inverse system 237
 multidisciplinary engineering 19
 open system 425
 ordinary first order differential equations
 92

T

Tearing 197–216
 Three tank system 618–623
 Through variable 24, 28
 Toolboxes 514–515
 Transfer functions 229–237
 Transformation tools 530–535
 Transformer
 displacement modulated 145, 385
 two-port 36
 Translator 471
 TUTSIM™ 507–511
 computational order 510
 structure table 508

U

Uncertainty bond graph 273–277
 Unity Loop Gain 155–160

V

Variable
 across 24
 co-energy variable 109
 energy variable 28
 extensive 438
 intensive 439

power variable 22
 semi-state 139
 state 106–111
 tearing variable 198, 199
 through 24
 Virtual inertia 171
 Voltage follower 117
 bond graph 118
 macro model 118

W

Weak causal determination of a junction 147
 Wheatstone bridge

bond graph of a hydraulic bridge 133
 hydraulic bridge with double acting cylinder
 132

X

XML 536
 schema 537

Z

Zero-junction 32
 Zero-order causal path 135